Diplomarbeit

Medizinische Bildsegmentierung mittels Fuzzy Logik

von

Markus Pils

Wintersemester 1995/96

Fachhochschule Darmstadt
Fachbereich Informatik

Referent: Prof. Dr. W.-D. Groch
Korreferent: Prof. Dr. W. Kestner

extern durchgeführt am:
Zentrum für Graphische Datenverarbeitung (ZGDV)
Darmstadt

Computer Graphics in Darmstadt

unter Betreuung von
Dipl.-Ing. Christoph Busch

Fachhochschule Darmstadt

Fachbereich Informatik

Referent: Prof. Dr. W.-D. Groch
Korreferent: Prof. Dr. W. Kestner

Aufgabenblatt

Diplomarbeit im Wintersemester 1995/96

Thema: „Medizinische Bildsegmentierung mittels Fuzzy Logik"

Motivation:

In der Medizinischen Informatik fallen Volumendaten an, die durch bildgebende Verfahren wie Computer Tomographie (CT) oder Kernspin Tomographie (MRT) erzeugt werden. Diese Daten bieten eine wertvolle Unterstützung im medizinischen Diagnoseprozess. Durch Verfahren der Bilddatenanalyse ist es möglich, automatisch einzelne Gewebestrukturen (z.B. Gehirntumore) in den Bildern zu segmentieren. Als Ergebnis liegt ein klassifizierter 3D-Volumendatensatz vor.
Durch den Einsatz unterschiedlicher Merkmalsextraktionsverfahren im Auswerteprozess werden zu einem Eingangsbild mehrere klassifizierte Bilder erzeugt, die nicht immer identische Resultate aufweisen.

Aufgabenstellung:

Im Rahmen dieser Arbeit soll ein Konzept erstellt und realisiert werden, mit dessen Hilfe klassifizierte Bilddaten miteinander verknüpft werden können. Dazu soll vorrangig die Methode der *Fuzzy Logik* (unscharfe Logik) betrachtet werden. Die Möglichkeiten einer logischen Verknüpfung sind zu untersuchen, wobei insbesondere die Anatomie der untersuchten Daten zu berücksichtigen ist. Zum Test des Verfahrens stehen Tomographiedaten des menschlichen Gehirns zur Verfügung.

Teilaufgaben:

- Literaturstudium (Theorie zur Fuzzy Logik, neuronale Netzwerke, projektbezogene Literatur)
- Einarbeitung in die Software des medizinischen Projekts KAMEDIN
- Erstellung eines Konzepts zur Bilddatenverknüpfung mittels Fuzzy Logik
- Realisierung
- Analyse und kritische Bewertung
- Ausarbeitung und Dokumentation

Die Anfertigung der Ausarbeitung soll parallel zur Bearbeitung der anderen Teilaufgaben erfolgen.

Hard - und Softwareumgebung

- Workstation
- Betriebssystem UNIX
- Programmiersprache C/C++
- Oberfläche OSF-Motiv

Ausgabe der Arbeit: 13. Oktober 1995

Bearbeiter: Markus Pils

Eidesstattliche Erklärung

Hiermit erkläre ich an Eides Statt, daß ich die vorliegende Diplomarbeit selbständig und allein mit den angegebenen Hilfsmitteln angefertigt habe.

Darmstadt, den 01. März 1996

1 Theorie der unscharfen Mengen

1.1 Einleitung und Motivation

Die Fuzzy Logik hat in den letzten Jahren in den verschiedensten Bereichen der Wissenschaft an Bedeutung gewonnen. Obwohl die Theorie der unscharfen Mengen auf größtenteils mathematisch gesicherten Erkenntnissen beruht, hat es sehr lange gedauert, bis die Fuzzy Logik allgemein akzeptiert wurde und auch im wissenschaftlichen Bereich Einzug nahm. Daß dies erst so spät geschah, hat sicherlich mehrere Ursachen. Zum einen hat der aus dem Englischen entlehnte Begriff *Fuzzy* , der sich zu *schwammig* oder *wattig* übersetzen läßt, im wissenschaftlichen Bereich lange Zeit für ein gewisses Unbehagen gesorgt, ist man dort doch bestrebt, die Dinge möglichst greifbar und exakt zu behandeln. Auf der anderen Seite haben heutzutage viele Felder der wissenschaftlichen Forschung eine derart hohe Komplexität erreicht, daß eine mathematisch exakte Beschreibung von Sachverhalten mit den Mitteln der herkömmlichen Mathematik oft nur noch schwer möglich ist, so daß in letzter Zeit verstärkt nach neuen Methoden gesucht wurde, den wachsenden Anforderungen gerecht zu werden. Im Kontext dieser Situation haben neuronale Netze und Fuzzylogik schlagartig an Boden gewonnen. Beide Gebiete ermöglichen erstmals die Modellierung und Steuerung extrem komplizierter Prozesse, ohne daß dafür eine genaue Kenntnis der mathematischen Hintergründe erforderlich wäre. Dies ist übrigens eine Eigenschaft, die dem menschlichen Gehirn noch nie Schwierigkeiten bereitet hat, so kann ein Mensch z.B. Fahrradfahren, ohne die den Vorgang beschreibende Differentialgleichung zu kennen.

Die computergestützte Auswertung von Bilddaten hat im Bereich der medizinischen Röntgentomographie Meilensteine gesetzt. Durch den Einsatz immer besserer Klassifikationsverfahren bei der Auswertung können eventuell vorhandene Krankheiten schon in einem sehr frühen Stadium erkannt und diagnostiziert werden.

Mit der vorliegenden Arbeit wird ein Konzept vorgestellt, die Zuverlässigkeit bestehender, auf neuronalen Netzwerken basierenden parametrischen Segmentierungsverfahren weiter zu erhöhen, indem relevante Bilddaten unter Einsatz von Fuzzy Logik selektiv miteinander verknüpft und klassifiziert werden.

Doch zunächst sollen die Grundlagen von Fuzzy Logik erläutert werden.
Prof. Lotfi A. Zadeh entwickelte bereits in den sechziger Jahren an der Universität Berkeley in Kalifornien die Theorie der unscharfen Mengen. Seine ursprüngliche Idee bei der Entwicklung der Theorie der Fuzzy Sets bestand darin, den Begriff der „charakteristischen Funktion“ (Sprungfunktion) einer gewöhnlichen Menge dahingehend zu erweitern, nicht nur zwischen Zugehörigkeit und Nicht-Zugehörigkeit der Elemente einer vorgegebenen Referenzmenge zu einer interessierenden Untermenge zu unterscheiden, sondern auch graduelle Zugehörigkeit einzubeziehen.

Mit Hilfe der unscharfen Mathematik lassen sich nicht nur die klassischen, scharfen Zustände *ja/nein*, *wahr/falsch* oder *0/1* berechnen, sondern auch viele Zwischenstufen. Wohlgemerkt

berechnen, denn am Ende von Operationen mit unscharfen Mengen steht wieder ein exakter Wert. Fuzzy Logik wird im Kontext *mehrwertiger Logiken* gesehen und ermöglicht es Computern, mit vagen, toleranzbehafteten Aussagen umgehen zu können, was in etwa auch der menschlichen Denkweise entspricht.
Unscharfe Mathematik ist keine Verschlechterung der klassischen Mathematik, sondern eine Erweiterung. Sie ermöglicht es, menschliche Alltagserfahrung und Relationen wie *sehr*, *etwas* oder Zustände wie *kalt, warm, heiß*, *schnell* und *langsam* mathematisch exakt und elegant in den Griff zu bekommen.

1.2 Scharfe Mengen

Betrachtet man einen beliebigen Zahlenkörper, z.B. die Menge $\mathbb{N}$ der natürlichen Zahlen, so läßt sich die benötigte Teilmenge in Form einer Liste von Zahlen darstellen. Die Relation „alle Zahlen kleiner als 10“ wird durch folgende Menge repräsentiert:

$$A = \{1,2,3,4,5,6,7,8,9\} \subseteq \mathbb{N}. \qquad \textbf{(1.1)}$$

Eine weitere Möglichkeit zur Darstellung von Teilmengen ist die Einführung eines Prädikats, beispielsweise „kleiner als 10“. Eine Zahl $n \in \mathbb{N}$ gehört genau dann zur Menge A, wenn die Prädikatenbedingung erfüllt ist.

$$A = \{n \in \mathbb{N} \mid n \text{ kleiner als } 10\} \qquad \textbf{(1.2)}$$

Oft bedient man sich einer *Sprungfunktion* bzw. *charakterististischen Funktion* zur Darstellung einer Menge. Dabei werden alle Elemente, für die die Prädikatenbedingung zutrifft, auf die Zahl 1 abgebildet, alle restlichen Elemente aus $\mathbb{N}$ auf die 0.

$$I_A: \quad \mathbb{N} \rightarrow \{0,1\} \qquad \textbf{(1.3)}$$

$$I_A(n) = \begin{cases} 1 & \textit{falls n kleiner 10} \\ 0 & \textit{sonst} \end{cases}$$

Mit dem Konzept einer scharfen Menge läßt sich z.B. die Zugehörigkeit der Menge „alle Menschen älter als 45 eindeutig beschreiben.

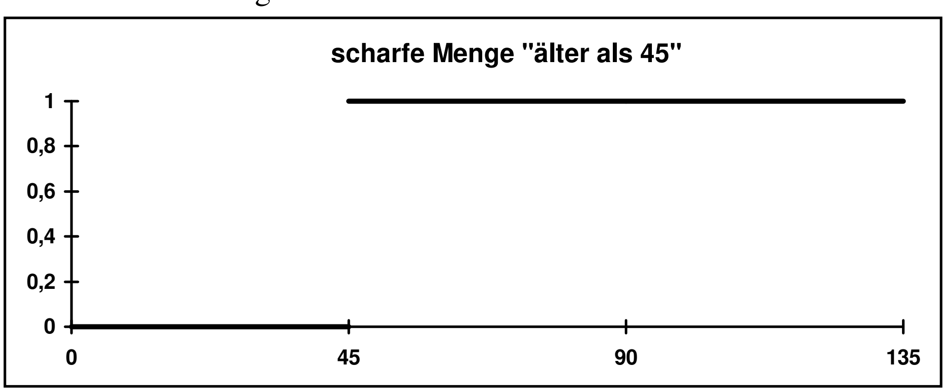


Abbildung 1.1 **Darstellung einer *scharfen* Menge**

1.3 Unscharfe Mengen

Mit der unscharfen Menge wird der *Cantorsche Mengenbegriff* um einen sogenannten Zugehörigkeitsgrad erweitert. Während in der klassischen Mathematik ein bestimmtes Element in einer Menge enthalten ist oder nicht, kann in einer unscharfen Menge ein Element auch nur zu einem gewissen Grad enthalten sein.

Ersetzt man nun den linguistischen Term „älter als 45“ aus Abbildung 1.1 durch „alle alten Menschen“, so widerspricht die mathematisch sicherlich richtige Definition doch der menschlichen Anschauung.

Ein 44 Jahre alter Mensch würde danach schon nicht mehr zur Menge der alten Menschen gehören, während eine 45 Jahre alte Person - definitionsgemäß - schon zu den alten Menschen zählt. Die Aussage hätte mit Sicherheit einen höheren Wahrheitsgehalt wenn man sagen würde, die Eigenschaft *alt* trifft auf die betreffende Person nur zu einem gewissen Grad zu.

Eine mögliche Lösung des Problems stellt die Verallgemeinerung der Definition der charakteristischen Funktion dar, und zwar indem man an Stelle der Abbildung auf die beiden Elemente {0,1}, eine stetige Abbildung auf das Intervall [0,1] definiert (siehe Abbildung 1.2). Dies führt zu folgender Beschreibung eines Fuzzy Sets.

Definition: *Es sei G eine Grundmenge und μ_X eine Funktion von G in das Intervall [0,1] der reellen Zahlenachse:*

$$\mu_X: G \to [0,1]. \qquad \textbf{(1.4)}$$

Die Menge

$$X = \{(x, \mu_X(x)) | x \in G\} \qquad \textbf{(1.5)}$$

heißt Fuzzy Set (unscharfe Menge, Fuzzy Menge) über G.

(Die Definition wurde entnommen aus [HABER95] S.247)

1.4 Linguistische Variablen

In der natürlichen Sprache werden Zustände oder Gegebenheiten meist nicht exakt, sondern nur vage beschrieben. Man sagt nicht, die Temperatur beträgt 5,25 ^{0}C, sondern es ist ziemlich kalt. Anstelle quantitativer Angaben treten in der natürlichen Sprache qualitative Zustandsbeschreibungen wie *niedrig, mittel, hoch, gut, schlecht, ..., usw*.
In der Fuzzy Logik gibt es sogenannte *linguistische Variablen*, deren Elemente aus verschiedenen Zuständen oder Ausprägungen bestehen. In der Literatur wird anstelle von *„Ausprägung“* häufig der Begriff *Fuzzy Term* verwendet. Eine Linguistische Variable kann beispielsweise die Terms *kalt, warm* und *heiß* annehmen. Ein Fuzzy Term läßt sich wiederum als Fuzzy Set darstellen.

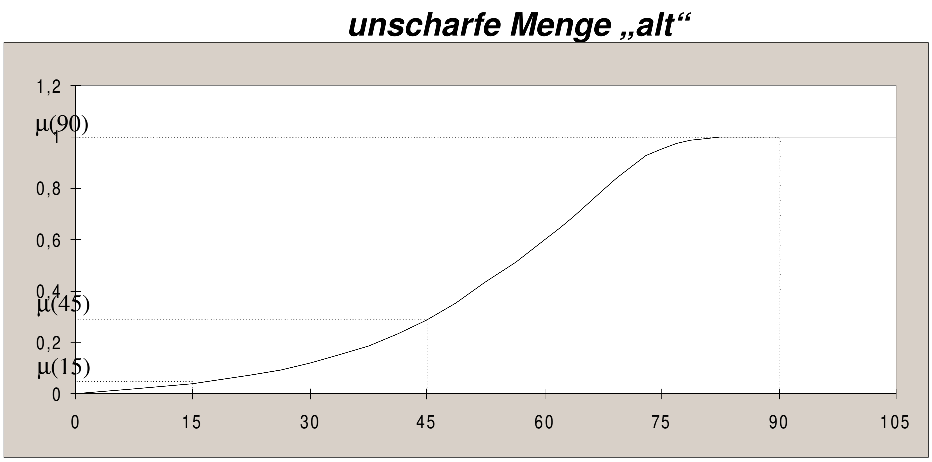


Abbildung 1.2 **Beispiel für ein Fuzzy Set**

Abbildung 1.2 beschreibt eine Zugehörigkeitsfunktion der linguistischen Variable „alt“ zum Fuzzy Set μ. Danach entspricht die Zugehörigkeit eines 90 jährigen zur linguistischen Variablen „alt“ 1 (μ(90) = 1), während eine 45 Jahre alte Person eine Zugehörigkeit zur linguistischen Variablen „alt“ (μ(45) = 0,3) von ungefähr 0,3 hat. Ein 15 jähriger hat einen Zugehörigkeitsgrad von 0,05 (μ(15)=0,05).

Die formelle Repräsentation einer Zugehörigkeitsfunktion geschieht häufig intuitiv, weil in vielen Anwendungen kein mathematisches Modell existiert, welches eine eindeutige Interpretation dieser unscharfen Daten zuläßt.

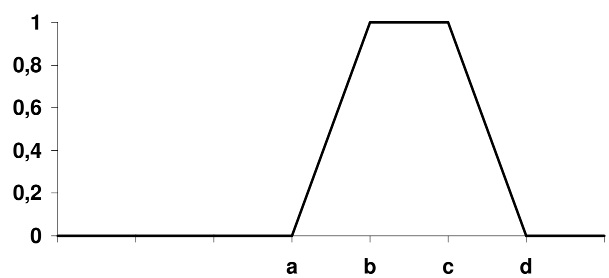
1.5 Repräsentationsformen für Fuzzy Sets

Es gibt verschiedene Möglichkeiten zur Bestimmung der Zugehörigkeitsfunktion eines Fuzzy Sets. Enthält die Menge X nur endlich viele Elemente, so kann man das Fuzzy Set μ der Menge X eindeutig definieren, indem man jedem Element $x \in X$ einen Zugehörigkeitsgrad μ(x) zuordnet. Ist die Anzahl der Elemente in X sehr groß oder nicht abzählbar, so beschreibt man μ(x) besser stückweise (Abbildung 1.3), oder als parametrische Funktion (Abbildung 1.4).

Beispiel: Es sei $X = \mathbb{R}$ und $a < b < c < d$,
ein linguistischer Ausdruck könnte beispielsweise durch folgende Funktion dargestellt werden.:

$$\mu_{a,b,c,d}(x) = \begin{cases} 0 & \textit{falls} \quad x < a \\ \frac{x-a}{b-a} & \textit{falls} \quad a \le x < b \\ 1 & \textit{falls} \quad b \le x \le c \\ \frac{x-d}{c-d} & \textit{falls} \quad c < x \le d \\ 0 & \textit{falls} \quad x > d \end{cases}$$

graphische Darstellung eines Fuzzy Sets

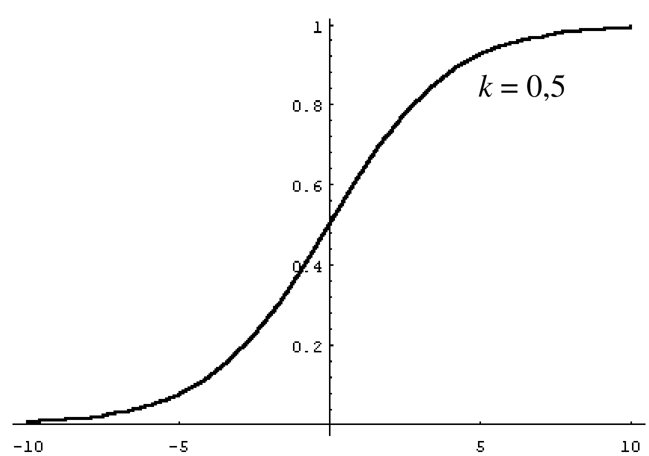


Formale Repräsentation (1.6) Abbildung 1.3 Graphische Darstellung

Die in Abbildung 1.4 dargestellte parametrische Sigmoid Funktion verwendet den Parameter *k* zur Einstellung der „Schärfe" der Funktion. Die Sigmoid Funktion ist stetig differenzierbar, und ihre Funktionswerte liegen zwischen 0 und 1. Mit zunehmender Schärfe ($k \to \infty$) nimmt die Funktion die Eigenschaften einer charakteristischen Sprungfunktion an.

$$y = S(x) = \frac{1}{1 + e^{-kx}}$$

mit $x \in [-\infty,\infty]$



Formale Repräsentation (1.7) Abbildung 1.4 Ausschnitt einer sigmoiden Funktion S(x)

1.6 Operationen mit Fuzzy Sets

Wie herkömmliche Mengen lassen sich auch Fuzzy Sets mathematisch miteinander verknüpfen. Hier stellt sich natürlich die Frage, wie man die klassischen, mengentheoretischen Grundoperationen auf Fuzzy Sets übertragen kann. Es wird vorausgesetzt, daß eine auf zwei Fuzzy Sets $\mu,\mu' \in F(X)$ anzuwendende Operation elementweise zu berechnen ist. Ein Operator kann erst dann als solcher akzeptiert werden, wenn er den aus der mehrwertigen Logik bekannten t-Norm Bedingungen entspricht. Diese können auf Seite 73 nachgelesen werden. Die im weiteren Verlauf verwendeten Operatoren erfüllen alle die t-Norm Bedingungen.

Die elementaren Operatoren, die Zadeh vorgeschlagen hat, sind der Minimumoperator für die Schnittmenge von zwei Fuzzy Sets $C = A \cap B$

$$\mu_{(C)}(\mathbf{x}) = \min\{\mu_A(\mathbf{x_1}), \mu_B(\mathbf{x_2})\} \quad x \in X, x_1 \in X_1, x_2 \in X_2 \qquad \textbf{(1.8)}$$

und der Maximumoperator für die Vereinigung von zwei Fuzzy Sets $C = A \cup B$

$$\mu_{(C)}(\mathbf{x}) = \max\{\mu_A(\mathbf{x_1}), \mu_B(\mathbf{x_1})\} \quad x \in X, \; x_1 \in X_1, x_2 \in X_2 \qquad \textbf{(1.9)}$$

sowie das Komplement C eines Fuzzy Sets A

$$\mu_{(C)}(\mathbf{x}) = 1 - \mu_A(\mathbf{x}) \qquad x \in X \qquad \textbf{(1.10)}$$

Alle Grundmengen X, X_1, X_2 bestehen aus den Elementen des Intervalls [0,...,1].

Graphisch können Zadehs Verknüpfungsoperationen wie folgt dargestellt werden:

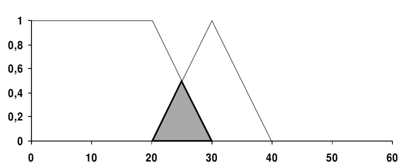


Abbildung 1.5

Schnittmenge zweier Fuzzy Sets

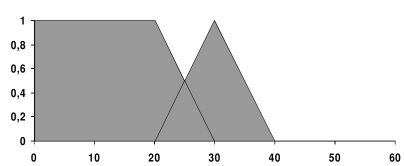


Abbildung 1.6

Vereinigung zweier Fuzzy Sets

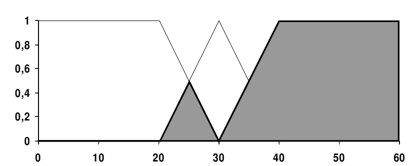


Abbildung 1.7

Negation von Abbildung 1.6

Die grauen Flächen stellen dabei die Ergebnismengen der verschieden Operatoren dar. Schnittmenge, Vereinigung und Komplement korrespondieren dabei mit den aus der binären Logik bekannten Operationen *UND*, *ODER* und *NICHT*. Die Rolle des UND bzw. ODER Operators muß allerdings nicht unbedingt von Zadehs MIN und MAX Operatoren übernommen werden. Es gibt eine Vielzahl weiterer Operatoren, die ein ähnliches Verhalten aufweisen und für das ein oder andere Problem z.T. besser geeignet sind.

Weitere Fuzzy Operatoren:

- **Algebraisches Produkt** (ap) und **algebraische Summe** (as)

$$ap(\mu_A(x), \mu_B(x)) = \mu_A(x) * \mu_B(x) \qquad \textbf{(1.11)}$$

$$as(\mu_A(x), \mu_B(x)) = \mu_A(x) + \mu_B(x) - \mu_A(x) * \mu_B(x) \qquad \textbf{(1.12)}$$

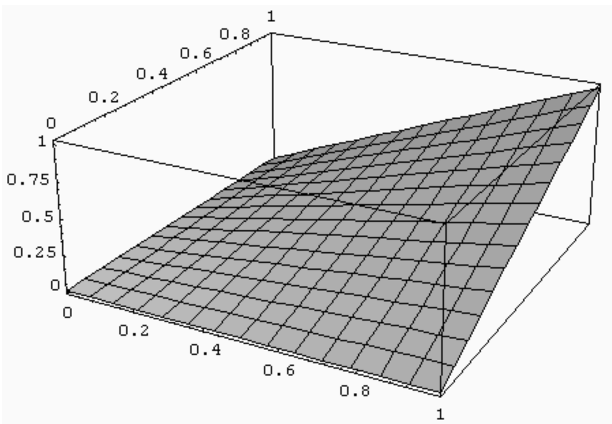


Abbildung 1.8 - Algebraisches Produkt

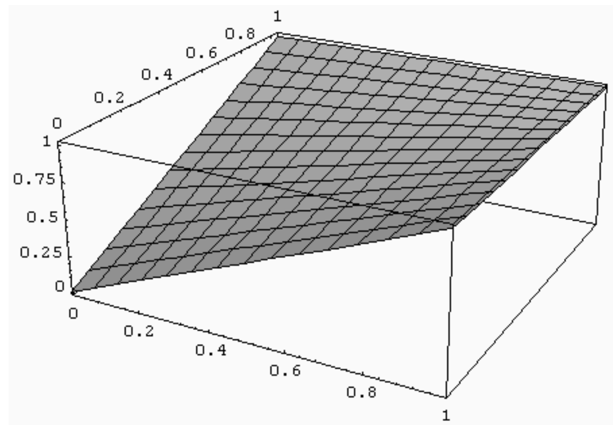


Abbildung 1.9 - Algebraische Summe

- **Hamacher Produkt** hp und **Hamacher Summe** hs

$$hp(\mu A(x), \mu B(x)) = (\mu A(x) * \mu B(x)) /(\mu A(x) + \mu B(x) - \mu A(x) * \mu B(x)) \qquad \textbf{(1.13)}$$

$$hs(\mu A(x), \mu B(x)) = (\mu A(x) + \mu B(x) - 2*\mu A(x)* \mu B(x)) /(1 - \mu A(x) * \mu B(x)) \qquad \textbf{(1.14)}$$

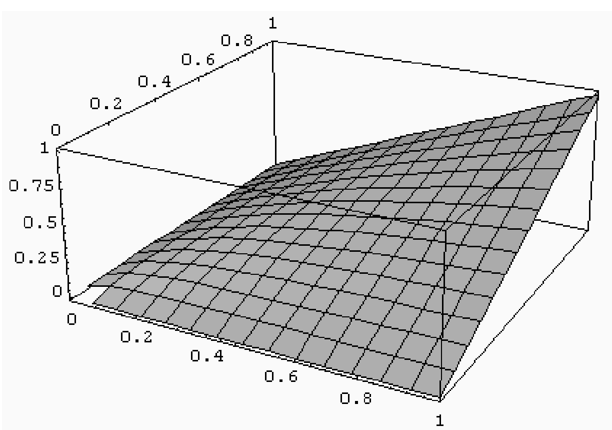


Abbildung 1.10- Hamacher Produkt

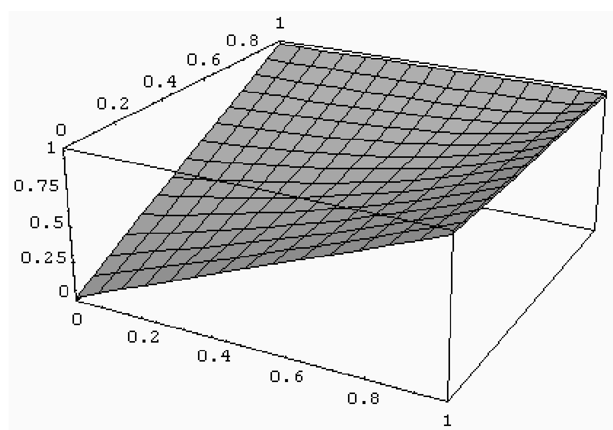


Abbildung 1.11 - Hamacher Summe

Jeder dieser Operatoren hat seine Berechtigung, denn er löst ein Problem mehr oder weniger gut. Eine gesicherte Theorie, wann welcher Operator einzusetzen ist existiert bisher nicht. Für die aus der binären Logik bekannten Werte 1 und 0 verhalten sich alle hier vorgestellten

Operatoren wie die hinreichend bekannten Binäroperatoren *UND* und *ODER* .Weitere wichtige Fuzzy Operatoren werden im Anhang beschrieben.

1.7 Der Begriff der unscharfen Zahl

Anschaulich kann eine unscharfe Zahl als ein Fuzzy Set betrachtet werden, als Intervall, in dessen Mitte die Zahl selbst (z.B. der Meßwert) liegt und dessen Breite durch die Toleranzen bestimmt wird. Da es keinen formalen Unterschied zwischen Fuzzy Zahlen und Fuzzy Sets gibt, werden Fuzzy Zahlen bei der graphischen Darstellung durch eine senkrechte Linie in der Intervallmitte (Abbildung 1.12) gekennzeichnet.

Ein Thermometer soll beispielsweise eine Temperatur von 37 °C anzeigen (Abbildung 1.12), seine Toleranz beträgt ±10 %. Die Temperaturangabe läßt sich als unscharfe Zahl darstellen im Intervall von [33,3°C;40,7°C]. Je exakter das Meßergebnis ist, desto höher ist auch der Zugehörigkeitsgrad μ des Meßwertes.

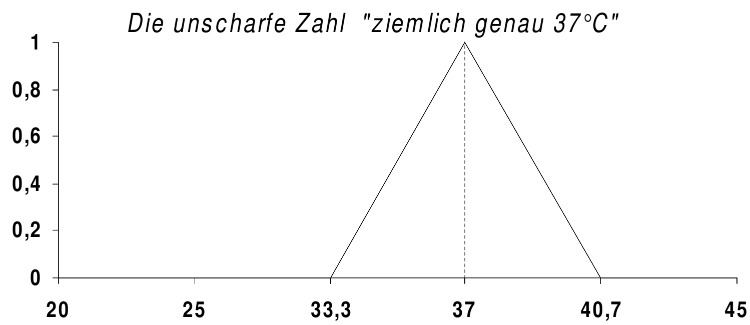


Abbildung 1.12 Darstellung einer unscharfen Zahl

Je größer die Toleranzangabe desto breiter ist das Intervall. Bei der Wahl der Zugehörigkeitsfunktion haben sich in der Technik symmetrische Dreiecksfunktionen (Abbildung 1.12) als besonders praktisch erwiesen, da Berechnungen mit stückweise linearen Funktionen auf dem Computer vergleichsweise einfach realisierbar sind. Es sind aber auch andere Verläufe gebräuchlich, beispielsweise gaußsche Glockenkurven der Form $\mu_{a,m}(x) = \exp(-a(x-m)^2)$, $a > 0$, $m \in \mathbb{R}$.

Aus der Sicht der mathematischen Modellierung bietet sich zur Darstellung von Toleranzen das Intervall $[x - \varepsilon, x + \varepsilon]$ an, auf dem mit den üblichen Methoden der Fehlerrechnung operiert wird. Dieser Ansatz hat allerdings den Nachteil, daß $x - \varepsilon$ und $x + \varepsilon$ intuitiv als scharfe Grenzen angesehen werden.

Bei Verknüpfungsoperationen zwischen Fuzzy Zahlen und Fuzzy Sets verwendet man die üblichen Fuzzy Operatoren. Da unscharfe Zahlen durch unscharfe Mengen repräsentiert werden, können Mengenoperationen zwischen zwei Fuzzy Sets ohne Einschränkung der Allgemeinheit auf Verknüpfungen zwischen unscharfen Zahlen und unscharfen Mengen übertragen werden.

1.8 Singletons

Singletons stellen die einfachste Form von Fuzzy Sets dar. Zugehörigkeitswerte $\mu(x)$ ungleich Null sind nur für bestimmte Abszissenwerte x_i definiert. Die unscharfen Mengen werden dabei in Form von Linien beschrieben, deren Höhen den Zugehörigkeitsgrad $\mu(x_i)$ bestimmen.

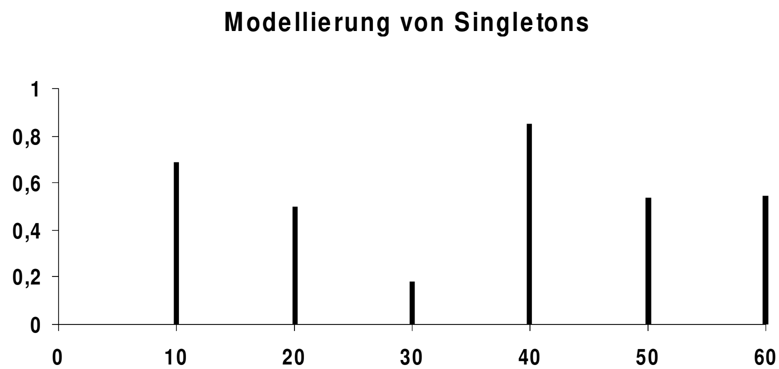


Abbildung 1.13 Darstellung von Singleton Sets

Singletons sind immer dann sinnvoll, wenn die auf der Abszisse abzutragenden Eingangswerte nur bestimmte, diskrete Werte annehmen können. Zwischenwerte $x \neq x_i$ nehmen den Zugehörigkeitsgrad $\mu(x) = 0$ an.

1.9 Fuzzy Regeln und Expertensysteme

Im täglichen Leben wird Vagheit bewußt in Kauf genommen, um dadurch die Kommunikation zu erleichtern und Informationen auf den für gezieltes Handeln wesentlichen Anteil zu beschränken. So ist beispielsweise die Verwendung des Adjektivs „schnell“ zur Charakterisierung der Fortbewegung eines entgegenkommenden Fahrzeugs völlig ausreichend, um ein notwendiges Ausweichmanöver einzuleiten. Die exakte Geschwindigkeit des entgegenkommenden Fahrzeugs spielt in dieser speziellen Situation eine untergeordnete Rolle.

Zur Modellierung vager, unpräziser und unsicherer Informationen ist neben dem Erzeugen von Fuzzy Sets, das Aufstellen eines Satzes von Verarbeitungsregeln notwendig, die es ermöglichen, bei bestimmten Konstellationen der Eingangsvariablen, entsprechende Folgeaktionen - wie z.B. das Durchführen eines Ausweichmanövers - einzuleiten. Diese Regeln beruhen meist auf Erfahrungen. Auf diese Weise kann Wissen direkt, ohne den Umweg über die Bildung abstrakter, theoretischer Modelle, umgesetzt werden.

Unter einem Fuzzy Expertensystem versteht man ein System, das anstelle boolescher Logik eine Sammlung von Zugehörigkeitsfunktionen und Regeln verwendet. Fuzzy Expertensysteme können sogar Prozesse regeln, die bislang nicht automatisch zu regeln waren.

Die Regeln in einem Fuzzy Expertensystem haben meist folgende Form:

$$\begin{array}{l} \textit{wenn } X_1 = A_{1,1} \textit{ Operator ... Operator } X_n = A_{n,1} \rightarrow Y_1 = B_1 \\ \vdots \\ \textit{wenn } X_1 = A_{1,k} \textit{ Operator ... Operator } X_n = A_{n,k} \rightarrow Y_k = B_k \end{array}$$

wobei alle X_1 bis X_n linguistische Bezugsvariablen sind und Y_1 bis Y_k die linguistischen Ergebnisvariablen. $A \in \{A_{1,1}, \ldots ,A_{n,k}\}$ und $B \in \{B_1, \ldots ,B_k\}$ sind Fuzzy Terms von $X_{i \ldots} X_k$ bzw. $Y \in \{Y_1, \ldots ,Y_k\}$ und stellen Referenzen auf ihre Zugehörigkeitsfunktionen dar. $B \in \{B_1, \ldots ,B_k\}$ ist der jeweilige Gesamtzugehörigkeitswert, der sich aus einer Regel berechnet. Als Operator kommt jeder den t-Norm Bedingungen genügende Fuzzy Operator in Frage, beispielsweise Zahdehs *Max* oder *Min* Operator, dabei können bei den Regeln problemabhängig durchaus verschiedene Operatoren eingesetzt werden.

Beispiel :

Regel 1: *WENN Gemüse = Tomate und Farbe = grün DANN Reifegrad = unreif.*
Regel 2: *WENN Reifezeit = kurz oder Reifezeit = mittel DANN Reifegrad = unreif.*

Gemüse	= Tomate	: 1	Gemüse	= Tomate	: 1
Farbe	= grün	: 0.8	Farbe	= grün	: 0.8
Reifezeit	= kurz	: 0.6	Reifezeit	= mittel	: 0.4

Anhand der Zugehörigkeitsfunktionen wurden folgende Zugehörigkeitsgrade ermittelt:

Dem Reifegrad *unreif* für Regel 1 wird der Zugehörigkeitsgrad min(1 ,0.8) = 0,8 zugeordnet.
Dem Reifegrad *unreif* für Regel 2 wird der Zugehörigkeitsgrad max(0.4 ,0.6) = 0,6 zugeordnet.

Für UND und ODER gelten Zadehs MIN und Max Operatoren.

Für den Fall, daß mehrere Regeln den Zugehörigkeitsgrad des selben Terms einer linguistischen Variablen bestimmen, muß eine Funktion den endgültigen Zugehörigkeitsgrad ermitteln. In diesem Fall verwendet man häufig das arithmetische Mittel ,oder einen Fuzzy Operator.

Eine Verknüpfung weist der Resultatvariablen eine Zugehörigkeitsfunktion zu. Die meisten Programme lassen pro Fuzzy Regel auch mehrere Schlußfolgerungen zu. Die Menge aller Regeln in einem Fuzzy Expertensystem nennt man Regel- oder Wissensbasis. Im Gegensatz zu normalen Expertensystemen erfordern Fuzzy Expertensysteme keine konsistente Regelbasis, außerdem kann auch Teilwissen in die Regelbasis mit aufgenommen werden.

1.10 Unscharfes Schließen

Nachdem die Grundlagen zur Darstellung von Fuzzy Sets und verschiede Verknüpfungsoperatoren vorgestellt wurden, soll nun ein Konzept vorgestellt werden, wie man auf der Grundlage von Fuzzy Sets und einer oder mehrerer Eingangsvariablen, aussagekräftige Prämissen erhalten kann.

Unter unscharfem Schließen versteht man den Vorgang, der anhand von Eingangsvariablen, einer Regelbasis und einer Menge von Fuzzy Sets, Aussagen bzw. Schlüsse auf eine oder mehrere Ergebnisvariablen zuläßt, mit denen man im weiteren Verlauf Entscheidungen treffen, oder einen Prozeß steuern kann.

Der Vorgang des unscharfen Schließens kann in die drei Teilprozesse Fuzzifizieren, Inferenz und Defuzzifizieren untergliedert werden.

1.10.1 FUZZIFIZIEREN

Die in einem Verarbeitungsvorgang auftretenden Eingangsgrößen a_1, ... ,a_n sind im allgemeinen scharfe Zahlenwerte. Beim Fuzzifizieren wird für eine scharfe Eingangsgröße a_i anhand der den Fuzzy Sets zugrunde liegenden Relationsvorschriften für jeden Term einer linguistischen Variablen der Zugehörigkeitsgrad μ_T (Übereinstimmungsmaß) ermittelt. Die Menge aus Zugehörigkeitswerten μ_T repräsentiert somit die scharfe Eingangsvariable a_i.

Beispiel 1.1: Die Temperatur einer Badewanne soll geregelt[1] werden.

Das Badewasser habe eine Temperatur von 34°C.
Gegeben seien die vier Fuzzy - Terms *kalt, kühl, warm, heiß*.
Die Mengenzugehörigkeitsfunktionen können aus dem Schaubild für die Linguistische Variable „Temperatur“ (Abbildung 1.14) entnommen werden.

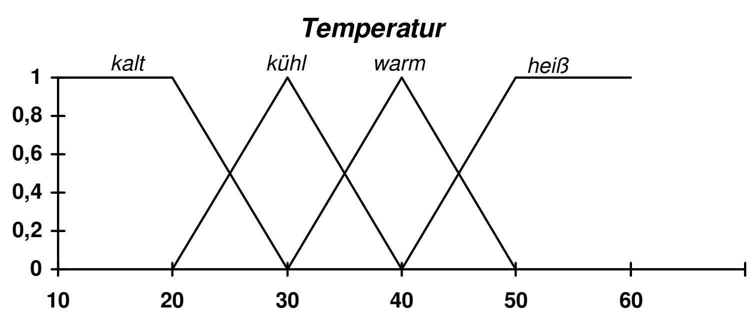


Abbildung 1.14 Darstellung einer linguistischen Variablen mit vier Terms

[1] Das Beispiel soll dem Verständnis über die Arbeitsweise von Fuzzy Logik dienen. Es wurde bewußt nur eine Eingangs- und Ergebnisvariable gewählt. Der für eine Regelung im eigentlichen Sinne notwendige Soll/Ist - Vergleich wurde weggelassen.

Der Fuzzifizierungsvorgang geschieht folgendermaßen. Es wird für jeden Term anhand der Zugehörigkeitsfunktionen für die „scharfe" Eingangsvariable Badetemperatur ein Zugehörigkeitswert ermittelt. Dies erfolgt durch Abtragen des Temperaturwertes an der Abszisse. Anhand der Schnittpunkte mit den Zugehörigkeitsfunktionen kann für jeden Term ein Zugehörigkeitswert durch Ablesen des Ordinatenwertes in Höhe des Schnittpunktes ermittelt werden.
Der Übersichtlichkeit wegen zeichnet man die Zugehörigkeitsfunktionen der einzelnen Terms, die eine Linguistische Variable annehmen kann, in das selbe Koordinatensystem. Eine Zugehörigkeitsfunktion ist auf dem gesamten Bereich ihrer Wertemenge X definiert. Erhält man beim Ermitteln des Zugehörigkeitswertes lediglich einen Schnittpunkt mit der Abszisse, so liegt der Zugehörigkeitswert außerhalb des Bereiches in dem die Zugehörigkeitsfunktion einen Wert ungleich Null liefert, der Zugehörigkeitswert μ ist dann Null.

Die Fuzzy Sets erzeugen für eine Badetemperatur von 34°C folgende Zugehörigkeitswerte:

$$\mu_{kalt}\,(34°C) = 0$$
$$\mu_{kühl}\,(34°C) = 0.6$$
$$\mu_{warm}(34°C) = 0.4$$
$$\mu_{heiß}\,(34°C) = 0$$

Jedem Term wurde nun ein Grad der Zugehörigkeit zu den verschieden Mengen zugewiesen. Anders formuliert: Der Fuzzifizierungsvorgang bestimmt, zu welchem Grad ein bestimmter Sachverhalt (Meßwert, Gegenstand, etc.) gewisse Eigenschaften erfüllt.

Dabei ist zu beachten, daß es sich beim Zugehörigkeitsgrad nicht um eine Wahrscheinlichkeit handelt. Eine Wahrscheinlichkeit sagt aus, wie wahrscheinlich oder unwahrscheinlich ein bestimmtes Ereignis eintrifft. Sie sagt aber nichts über einen tatsächlichen Zustand aus.
Der Zugehörigkeitsgrad ist aber eine reale Eigenschaft eines betrachteten Elements, das diese Eigenschaft auch tatsächlich besitzt. Aus den Zugehörigkeitswerten einer linguistischen Variablen läßt sich ohne weiteres wieder der korrekte Wert der „scharfen" Eingangsvariablen berechnen.

Für die Zugehörigkeitsgrade in Beispiel 1.1 könnte eine sehr grobe Look-Up-Tabelle etwa wie folgt aussehen:

Temperatur [°C]	10	15	20	25	30	35	40	45	50	55	60
μ_{kalt}	1	1	1	0,5							
$\mu_{kühl}$				0,5	1	0,5					
μ_{warm}						0,5	1	0,5			
$\mu_{heiß}$								0,5	1	1	1

Tabelle 1.1 Look-Up-Tabelle mit Zugehörogkeitsgraden

Die leeren Felder haben einen Zugehörigkeitswert zum entsprechenden Fuzzy Set $\mu = 0$.

1.10.2 INFERENZ

Der Inferenzprozeß schafft die Verbindungen zwischen Fuzzy Regeln und linguistischen Variablen und umfaßt im wesentlichen zwei Verfahrensschritte. Im ersten Verfahrensschritt werden sämtliche vorhandenen Fuzzy Regeln unter Zugrundelegung einer Funktion f_1 ausgewertet. Aus jeder Regel erhält man als Ergebnis einen Zugehörigkeitsgrad μ. Da es möglich ist, daß zwei oder mehrere verschiedene Regeln existieren, deren Ergebnisse sich auf den selben Term B_i einer linguistischen Zielvariablen Y beziehen, müssen diese Zugehörigkeitswerte mit einer Funktion f_2, die im Folgenden auch Inferenzoperator genannt wird, noch einmal miteinander verknüpft werden, denn letztendlich kann jedem Term B_i der Zielvariablen nur ein einziger Zugehörigkeitswert zugeordnet werden.
Der zweite Verfahrensschritt befaßt sich mit der Modifikation der Zugehörigkeitsfunktionen einer linguistischen Zielvariablen, anhand der im ersten Schritt für die einzelnen Terms ermittelten Zugehörigkeitswerte. Die Modifikation der Fuzzy Sets erfolgt durch eine weitere Funktion f_3.

In Schaubild (Abbildung 1.15) stellt sich der Inferenzprozeß schematisch wie folgt dar:

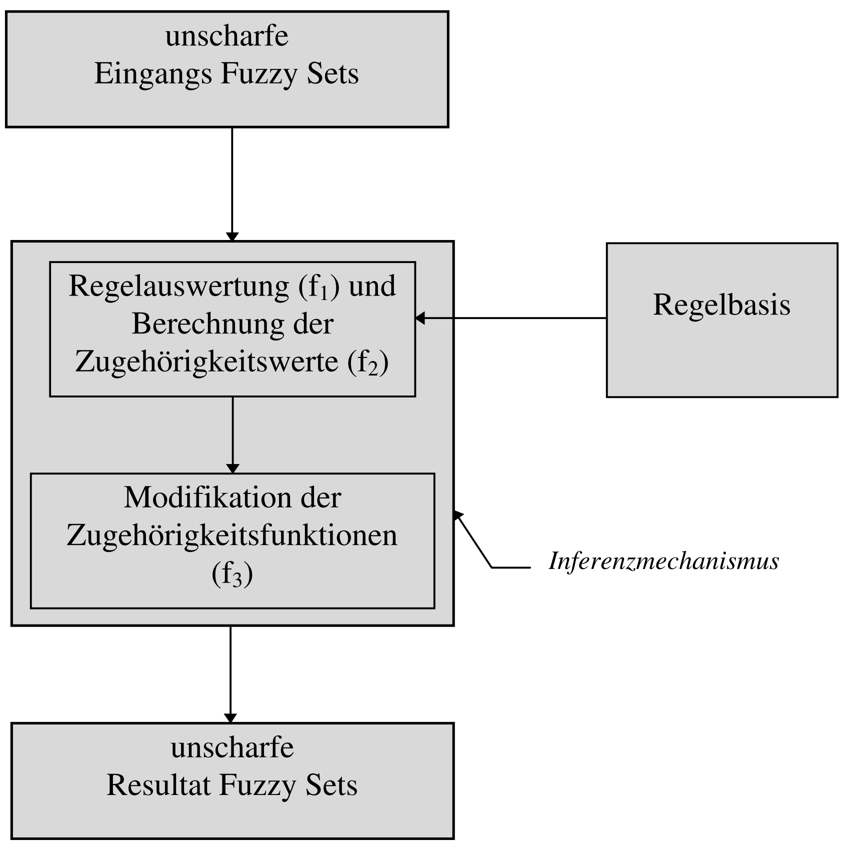


Abbildung 1.15 Der Inferenzprozeß

Beispiel 1.1 **(Fortsetzung)**

Gegeben sei folgender Verlauf der Zugehörigkeitsfunktionen des zulaufenden Wassers:

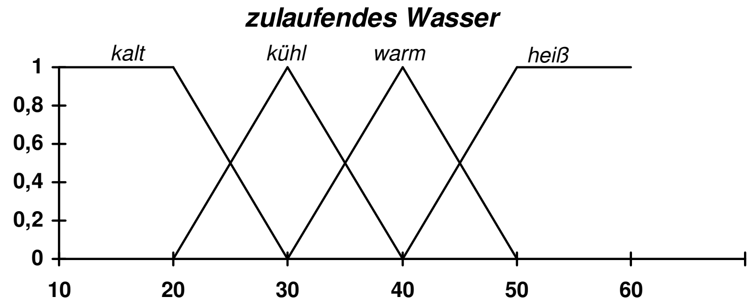


Abbildung 1.16 Fuzzy Sets des zulaufenden Wassers

Außerdem seien in einer Regelbasis noch folgende Inferenzregeln vereinbart:

Inferenzregeln:

WENN Badewasser IS kalt DANN zulaufendes Wasser heiß.
WENN Badewasser IS kühl DANN zulaufendes Wasser warm.
WENN Badewasser IS warm DANN zulaufendes Wasser kühl.
WENN Badewasser IS heiß DANN zulaufendes Wasser kalt.

Mittels der Inferenzregeln kann von den unscharfen Eingangsvariablen auf die Resultatvariablen geschlossen werden. Die Inferenzregeln sind hierbei nicht als Implikationen zu verstehen, die als Ergebnis einen Wert Null oder Eins zurückliefern, sondern sie wirken sich auf die Resultatvariablen nur so weit aus, wie der aus ihnen ermittelte Gesamtzugehörigkeitswert aus den Eingangsvariablen.

Für das angeführte Badewasserbeispiel erhält man für die Resultatvariable *zulaufendes Wasser* folgende Zugehörigkeitsrade:

μ_{kalt} (zulaufendes Wasser) = $\mu_{heiß}$ (Badewasser) = 0
$\mu_{kühl}$(zulaufendes Wasser) = μ_{warm} (Badewasser) = 0,6
μ_{warm}(zulaufendes Wasser) = $\mu_{kühl}$ (Badewasser) = 0,4
$\mu_{heiß}$ (zulaufendes Wasser) = μ_{kalt} (Badewasser) = 0

In einem komplizierteren Beispiel mit mehreren Eingangsvariablen wäre es durchaus denkbar, daß weitere Regeln existieren, die den selben Zugehörigkeitsert eines Terms beeinflussen.
z.B.: *WENN AUSSENTEMPERATUR IS niedrig DANN zulaufendes Wasser IS warm.*
Diese zusätzliche Regel hätte zur Folge, daß für μ_{warm} zwei unterschiedliche Zugehörigkeitswerte ermittelt würden. Da jedem Term nur einen Zugehörigkeitswert zugeordnet werden kann, müssen die verschiedenen Zugehörigkeitswerte eines Terms durch

eine geeignete Funktion f_2 zu einem einzigen Zugehörigkeitswert zusammengefaßt werden. Hier bietet sich - je nach Problemstellung - der Maximum Operator oder das arithmetische Mittel an (Abbildung 1.17).
Zur Bestimmung des „endgültigen“ Zugehörigkeitswertes $\mu(B)$ eines Fuzzy Terms wird mittels einer Funktion f_2 aus allen für den Term ermittelten Zugehörigkeitswerten $f_2(\mu_1 \ldots \mu_n)$ ein Gesamtzugehörigkeitswert $\mu(B)$ ermittelt. Handelt es sich bei der Funktion f_2 um einen Fuzzy Operator, so gelten die t- und t-Conormbedingungen. Die Funktion f_2 ließe sich dann auch als Verkettung von Operatoren darstellen.

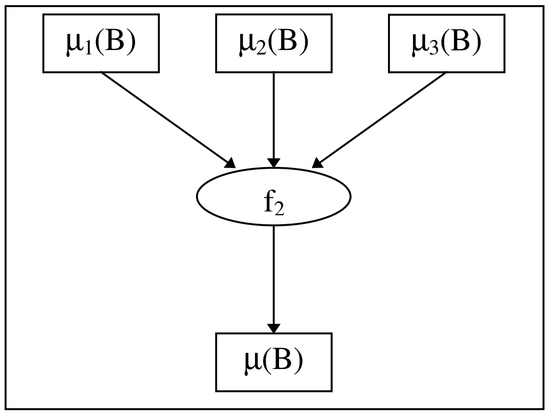


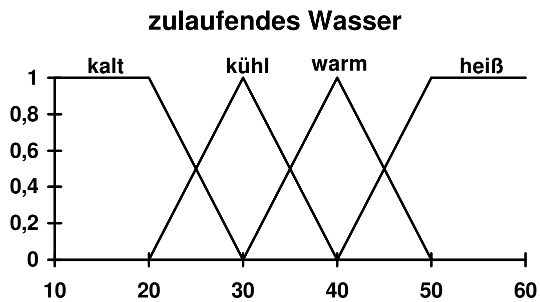
Abbildung 1.17 Verknüpfung von Zugehörigkeitswerten

Nachdem im ersten Teilprozeß die Zugehörigkeitswerte zu den einzelnen Terms ermittelt wurden, werden im zweiten Teilprozeß die Zugehörigkeitsfunktionen der Resultatsets implizit mit Funktionen $f_3(\mu(B),x_i)$ modifiziert. $\mu(B)$ bezeichnet hierbei den für einen Fuzzy Term berechneten Zugehörigkeitswert.

Bei der das Resultat Fuzzy Set modifizierenden Funktion f_3 handelt es sich wieder um einen Fuzzy Operator. Die Modifikation erfolgt dermaßen, daß beispielsweise bei Verwendung von Zadehs Minimum Operator, jedes einzelne Element x_i des Fuzzy Sets X durch das Funktionsergebnis $MIN(\mu(B),x_i)$ ersetzt wird. Anschaulich betrachtet entspricht dies einem Abschneiden (clipping) der Zugehörigkeitsfunktion in Höhe des für sie ermittelten Zugehörigkeitswertes. Wird als modifizierende Funktion f_3 das arithmetische Produkt verwendet, so entspricht dies einer Skalierung der Zugehörikeitsfunktion mit dem für sie ermittelten Zugehörigkeitswert.

Wurde als Inferenzoperator der MAX Operator verwendet (f_2 = MAX), und als modifizierende Funktion f_3 Zadehs MIN Operator, so nennt man den gesamten Inferenzvorgang Max/Min Methode. Analog nennt man den Inferenzvorgang Max/Produkt Methode, wenn als modifizierende Funktion f_3 der arithmetische Produktoperator eingesetzt wird.

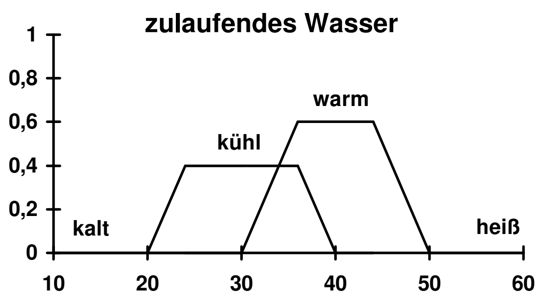
Abbildung 1.18 zeigt das Resultat Fuzzy Set des zulaufenden Wasser vor der Modifikation der aus den Fuzzy Regeln ermittelten Zugehörigkeitswerten. Abbildung 1.19 und Abbildung 1.20 zeigen die Fuzzy Sets nach der Modifikation mit der Max/Min bzw. Max/Produkt Methode.



Max/Produkt-Methode

Abbildung 1.18

Max/Min-Methode



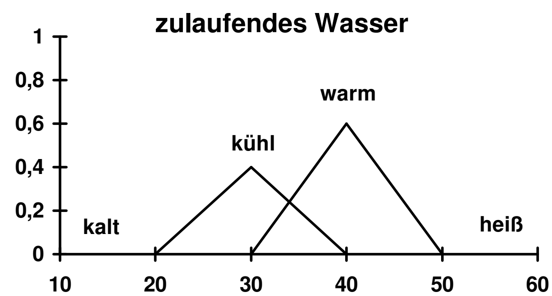


Abbildung 1.19 - Resultatsets (Clipping)

Abbildung 1.20 - Resultatsets (Skalierung)

Mit dem Anpassen der Zugehörigkeitsfunktionen für alle linguistischen Variablen ist der Inferenzvorgang beendet. Die unscharfen Ergebnisvariablen werden durch die modifizierten Zugehörigkeitsfunktionen repräsentiert. Die so gewonnenen Resultat Fuzzy Sets können als Zwischenergebnis dienen, die durch zusätzliche Fuzzy Regeln weitere Fuzzy Sets modifizieren, oder man leitet direkt den Defuzzifizierungsvorgang ein, der aus den Fuzzy Sets einer linguistischen Variablen wieder ein scharfes Ergebnis berechnet.

1.10.3 DEFUZZIFIZIEREN

Bisher werden alle Ein- und Ausgangsgrößen durch unscharfe linguistische Variablen dargestellt. Bei der praktischen Anwendung möchte man jedoch oft aus numerischen Fakten wieder eine numerische Größe berechnen. Dazu ist eine Umwandlung vom unscharfen Fuzzy Wert zu einer scharfen Größe notwendig.

Die einfachste Möglichkeit der Defuzzifizierung erhält man, indem man den größten Zugehörigkeitswert $Z_m = \sup\{\max\{A_1, \ldots, A_i\}\}$ aller Fuzzy Sets einer linguistischen Variablen bestimmt. Der Schnittpunkt einer den Punkt Z_m durchlaufenden Orthogonalen mit der Abszisse ergibt dabei das scharfe Ergebnis. Diese sogenannte *Mean of Maximum* Methode (MOM) eignet sich allerdings nur für Probleme, an deren Ende ein qualitatives Ergebnis steht, für andere Aufgaben, z.B. Klassifizierungen ist das (scharfe) Ergebnis zu ungenau.

Eine sehr häufig verwendete Methode zur Defuzzifizierung ist die Schwerpunktmethode *Center of Gravity* (COG). Dabei bildet der Schwerpunkt der vom Fuzzy Set umschlossenen Flächen das Ergebnis.

$$Y_L = \frac{\int_{x \in Y} \mathbf{x} * \mu_{x1,\ldots,xn}(\mathbf{x})\mathbf{dx}}{\int_{x \in Y} \mu_{x1,\ldots,xn}(\mathbf{x})\mathbf{dx}} \quad \textbf{(1.15)}$$

Y_L : x - Koordinate des Flächenschwerpunktes
$\mu_{x1,\ldots,xn}$: Funktion der Flächenkurve
n : Anzahl Terms der lingusistischen Ergebnisvariable
$x \in Y$: Elemente x der Ergebnisvariablen Y

Das Verfahren eignet sich besonders gut, wenn die numerische Größe aus vielen Regeln berechnet werden soll. Überlappende Bereiche werden von der Flächenformel allerdings nicht berücksichtigt. Folgende Schwerpunktformel zur Defuzzifizierung gewichtet zusätzlich die Teilschwerpunkte der Kurvenflächen.

$$Y_L = \frac{\sum_{i=1}^{n} x_i A_i g_i}{\sum_{i=1}^{n} A_i g_i} \quad \textbf{(1.16)}$$

Y_L : x - Koordinate der Lösung
x_i : x - Koordinate des einzelnen Teilschwerpunktes
A_i : Flächeninhalt der einzelnen unscharfen Menge
g_i : individueller Gewichtsfaktor der einzelnen unscharfen Menge
i : Laufvariable über die Terms
n : Anzahl Terms der linguistischen Ergebnisvariable

Für das Badewasserbeispiel wurden bei Verwendung der Max/Produkt Methode (Abbildung 1.18) für die vier Fuzzy Sets folgende Werte ermittelt:

$x_{kalt} = 0$	$A_{kalt} = 0$
$x_{kühl} = (20+30+40) / 3 = 30$	$A_{kühl} = (40 - 20) * 0.4 /2 = 4$
$x_{warm} = (30+40+50) / 3 = 40$	$A_{warm} = (50 - 30) * 0.6 /2 = 6$
$x_{heiß} = 0$	$A_{heiß} = 0$
x-Koordinaten der Teilschwerpunkte	**Teilflächen der einzelnen unscharfen Mengen**

Die errechneten Schwerpunkte der Fuzzy Sets werden - unabhängig vom individuellen Gewichtsfaktor g_i - durch ihre Flächeninhalte gewichtet und aufsummiert n : **Anzahl Terms der linguistischen Ergebnisvariable** (1.16). Ein Resultat Fuzzy Set mit großem Flächeninhalt geht stärker ins Gesamtergebnis ein, als ein Resultat Fuzzy Set mit kleinem Flächeninhalt. Die Teilschwerpunkte (x_i,y_i) können sehr einfach durch Vektoraddition der Eckkoordianten und anschließendem Normalisieren berechnet werden. Der gewichtete Gesamtschwerpunkt der Abszissenkoordinaten ergibt sich dann zu:

$$Y_L = \frac{30*4+40*6}{4+6} = 36 .$$

Der individuelle Gewichtsfaktor g_i ist 1. Der Defuzzifizierungsmechanismus hat für die Temperatur des zulaufenden Wassers den scharfen Wert 36°C berechnet.
Im vorliegenden Beispiel konnten die Flächen aus der bekannten Flächenformel für Dreiecke berechnet werden. Allgemein kann eine aus diskreten Koordinatenpaaren bestehende Zugehörigkeitsfunktion als Polygonzug aufgefaßt werden, der zur Ermittlung des Flächeninhalts, unter Beachtung der Vektorumlaufrichtung, in eines oder mehrere konvexe Polygone zerlegt wird. Die Gesamtfläche der Zugehörigkeitsfunktion erhält man durch Aufsummieren der konvexen Teilflächen.

Eine allgemeine Formel zu Berechnung der Fläche S eines konvexen Polygons läßt sich leicht aus der Determinante dreier Vektorkoordinaten ableiten und lautet für n Punkte $P_1, \dots, P_n$ folgendermaßen:

$$S = \frac{1}{2}\left[(x_1-x_2)(y_1+y_2) + (x_2-x_3)(y_2+y_3) + \dots + (x_{n-1}-x_n)(y_{n-1}+y_n) + (x_n-x_1)(y_n+y_1)\right]$$

$$P_i = (x_i, y_i) \qquad \textbf{(1.17)}$$

Die durch Fuzzy Logik ermittelten Ergebnisse sind stark abhängig von den Fuzzy Regeln und der Definition der Zugehörigkeitsfunktionen. Wird Fuzzy Logik zur Regelung oder Steuerung von Systemen eingesetzt, so entfällt ein großer Teil der Arbeit auf das Anpassen der Zugehörigkeitsfunktionen und die Wahl der Fuzzy Regeln.

1.11 Aufbau eines Fuzzy Controllers

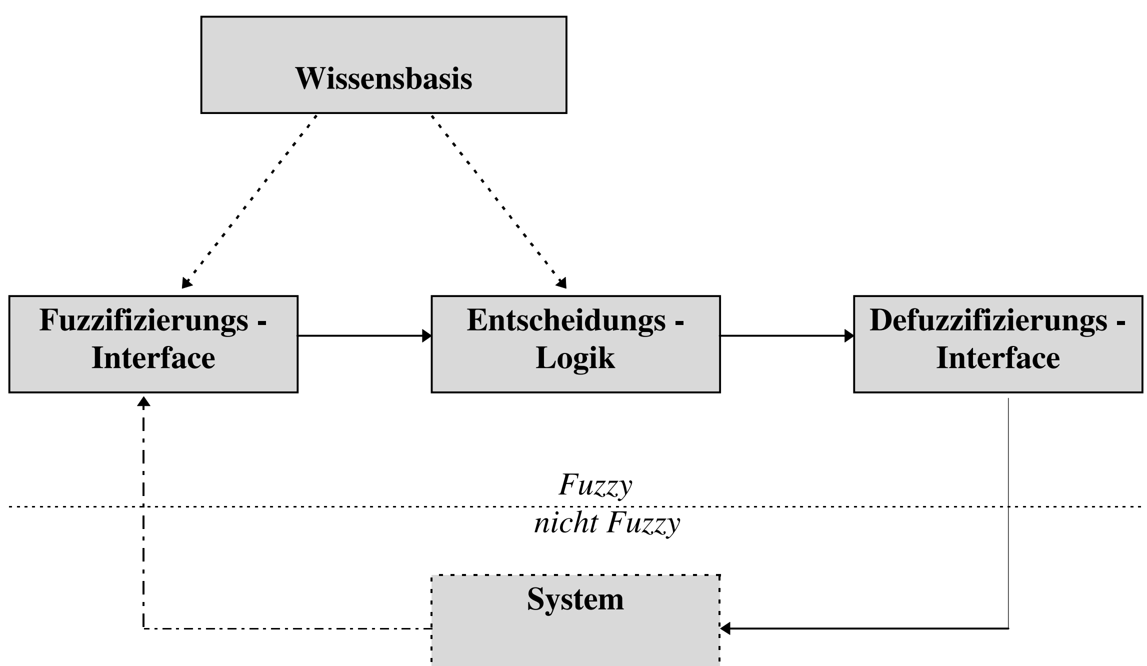


Abbildung 1.21

Ein Fuzzy Controller hat die Aufgabe ein System mittels Fuzzy Logik zu regeln oder steuern. Das in Abbildung 1.21 dargestellte System besteht aus folgenden Komponenten:

- *Fuzzifizierungs Interface:*

  Das Fuzzifizierungs Interface nimmt den scharfen Eingangswert auf und bildet ihn anhand der Zugehörigkeitsfunktionen ab, so daß am Ende für jedes Fuzzy Set ein Zugehörigkeitsgrad angegeben werden kann, deren Gesamtheit eine scharfe Eingangsvariable repräsentiert.

- *Wissensbasis:*

  Die Wissensbasis enthält im Wesentlichen eine Ansammlung von linguistischen Kontrollregeln, die die Arbeitsweise des Fuzzy Controllers festlegen. Neben den Kontrollregeln kann eine Wissensbasis noch weitere Informationen enthalten, beispielsweise Wertebereiche und zusätzliche Transformationsvorschrifen der Meß- und Stellgrößen, sowie die Definitionen der Zugehörigkeitsfunktionen und eventuell erforderliche Normierungs-vorschriften.

- *Entscheidungslogik:*

  Die Entscheidungslogik ist der Kern des Fuzzy Controllers und erzeugt aus den Eingangsgrößen und der Regelbasis die entsprechenden Resultatgrößen. Dem Entscheidungsprozeß liegt der Inferenzmechanismus für unscharfe Mengen zugrunde.

- *Defuzzifizierungs Interface:*

  Das Defuzzifizierungs Interface berechenet aus den unscharfen Resultatinformationen wieder eine scharfe Resultatgröße, die als Stellwert für weitere Steuerungs - oder Regelungsprozesse verwendet werden kann.

1.12 Ansatz einer Fuzzy Regelung nach Mamdani

Bei diesem Ansatz formuliert der Experte sein Wissen in Form von linguistischen Regeln. Zunächst werden die linguistischen Terme festgelegt, die in den Fuzzy Regeln auftreten können. Es müssen daher für jede der Wertemengen $X_1,...,X_n$ für die Eingangsgrößen und Y für die Stellgröße geeignete linguistische Terme wie ungefähr Null, positiv klein, positiv mittel, usw. festgelegt werden.
Für eine mathematische Modellierung muß dann jedem dieser linguistischen Terme ein Fuzzy Set zugeordnet werden. Mit dem Namen der Terme assoziert man auf diese Weise die zum Term gehörenden Fuzzy Sets.

Um geeignete Regeln für die Wissensbasis des Fuzzy - Reglers aufstellen zu können, wird zunächst jede der Mengen der Eingangsgrößen $X_1,...,X_n$ und Y_j mit Hilfe von Fuzzy Mengen

„partitioniert“. Dazu werden auf der Menge X_1 p verschiedene Fuzzy Mengen mit einem linguistischen Term assoziiert. Oft verwendet man hierzu symmetrische Dreiecksfunktionen.

Jede der Eingangsgrößen X_i wird auf diese Art und Weise in verschiedene Fuzzy Sets unterteilt, dabei kann die Anzahl der Partitionierungen für verschiedene X_i durchaus unterschiedlich groß sein.

Beispiel 1.2:

Ein freistehender Stab soll in der Schwebe gehalten werden.

Ein Meßsystem ermittelt ständig Winkel θ und Winkelgeschwindigkeit $\dot{\theta}$. Für das Stabbalance-Problem wird eine Regelbasis mit 19 Regeln verwendet. Es handelt sich hierbei um ein Beispiel, das tatsächlich an der Universität Braunschweig realisiert wurde. Für die Eingangs und Resultat Fuzzy Sets wurden stückweise lineare, symmetrische Dreiecksfunktionen verwendet, wie sie schon im Badewasserbeispiel beschrieben wurden. Nachdem die Partitionierungen der Mengen X_1, ..., X_n, Y_1, ... , Y_k festgelegt sind, wird anschließend die Regelbasis spezifiziert, die Kontrollregeln der Form

Rule 1: *wenn* $X_1 = A_{1,1}$ and ... and $X_n = A_{n,1} \rightarrow Y_1 = B_1$

. . . . .

. . . . .

Rule k: *wenn* $X_1 = A_{1,k}$ and ... and $X_n = A_{n,k} \rightarrow Y_k = B_k$

besteht.

Die Regelbasis für das Stabbalance-Problem sieht für die beiden Eingangsvariablen tabellarisch folgendermaßen aus:

$\dot{\theta}$ \ θ	ng	nm	nk	eN	pk	pm	pg
ng			**pk**	**pg**			
nm				**pm**			
nk	**nm**		**nk**	**pk**			
En	**ng**	**nm**	**nk**	**eN**	**pk**	**pm**	**pg**
pk				**nk**	**pk**		**pm**
pm				**nm**			
pg				**ng**	**nk**		

Termbezeichnungen

ng : negativ groß
nm : negativ mittel
nk : negativ klein
eN : etwa Null
pk : positiv klein
pm : positiv mittel
pg : positiv groß

Tabelle 1.2 Regelbasis für das Stab Balance Problem

Quelle: [KRU93]

Auffällig ist hierbei, das nicht notwendigerweise für jedes Paar von Eingangsgrößen $(\theta, \dot{\theta})$ eine Regel existieren muß. Im obigen Beispiel wurden nur für solche Eingangsgrößen Regeln definiert, für die der Stab auch in der Schwebe gehalten werden kann. Anschließend erfolgt der Defuzzifizierungsvorgang mittels MOM (mean of maximum) oder COG (center of gravity) .

Der Eintrag in der zweiten Zeile von Tabelle 1.2 ist beispielsweise als die Regel

if θ **is** *etwa Null* **and** $\dot{\theta}$, **is** *negativ mittel* **then** Y **is** *positiv mittel.*

zu verstehen.

Dabei sind $A_{1,1},\ldots, A_{n,k}$ linguistische Terme, die den Fuzzy - Mengen $\mu_{1,1},\ldots, \mu_{n,k}$ gemäß den Partitionierungen der Mengen $X_1, \ldots , X_n$ bzw. $Y_1, \ldots , Y_k$ entsprechen. Diese Kontrollregeln sollen im Falle des Ansatzes von Mamdani nicht als Implikationen, sondern im Sinne einer stückweise definierten Funktion verstanden werden.

2 Konzept einer Anwendung auf das gestellte Segmentierungsproblem.

Das medizinische Bildverarbeitungssystem Kamedin stellt eine Reihe von Bildverarbeitungs-algorithmen zur Verfügung, mit denen es möglich ist, radiologische Bilddaten zu analysieren und weiterzuverarbeiten. Hierbei handelt es sich vorwiegend um Volumendaten des menschlichen Gehirns, die bei der Kernspintomographie anfallen. Die Auswertung der Tomographiedaten erfolgt automatisch. Verschiedene, trainierte, Segmentierungsalgorithmen auf der Basis neuronaler Netzwerke segmentieren dabei die Gewebeteile in ihre verschiedenen Gewebeklassen. Die Volumendaten liegen als Sequenz von voneinander unabhängigen Schichtbildern vor.
Das Hauptinteresse bei der Segmentierung gilt dabei der Erkennung von pathologischem Gewebe, d.h. das System soll krankes Gewebe von gesundem Gewebe unterscheiden können. Ziel dieser Arbeit ist es, die durch die Segmentierungsalgorithmen bereitgestellten Resultatbilder (Abbildung 2.1) mittels Fuzzy Logik so miteinander zu verknüpfen, daß ein besseres Segmentierungsergebnis entsteht, als es die einzelnen Algorithmen leisten könnten, d.h.

- Die Resultatbilder sollten nach der Verknüpfung mit einem Fuzzy Tool eine möglichst hohe statistische Übereinstimmung mit den von Experten klassifizierten Testgebieten[2] haben.
- Die Bilder sollen rauscharm sein und die Klassen aus möglichst großen, zusammen-hängenden Gebieten bestehen, wobei diese Aufgabe größtenteils von einer Bildnachverarbeitung (Postprocessing) übernommen wird.

Die Zahl der betrachteten Gewebeklassen richtet sich nach dem medizinischem Interesse und unterscheidet derzeit die acht verschiedenen Klassen *Tumor, Liquor, Oedem, Knochen, Fett, graue Gehirnsubstanz, weiße Gehirnsubstanz und Hintergrund*.

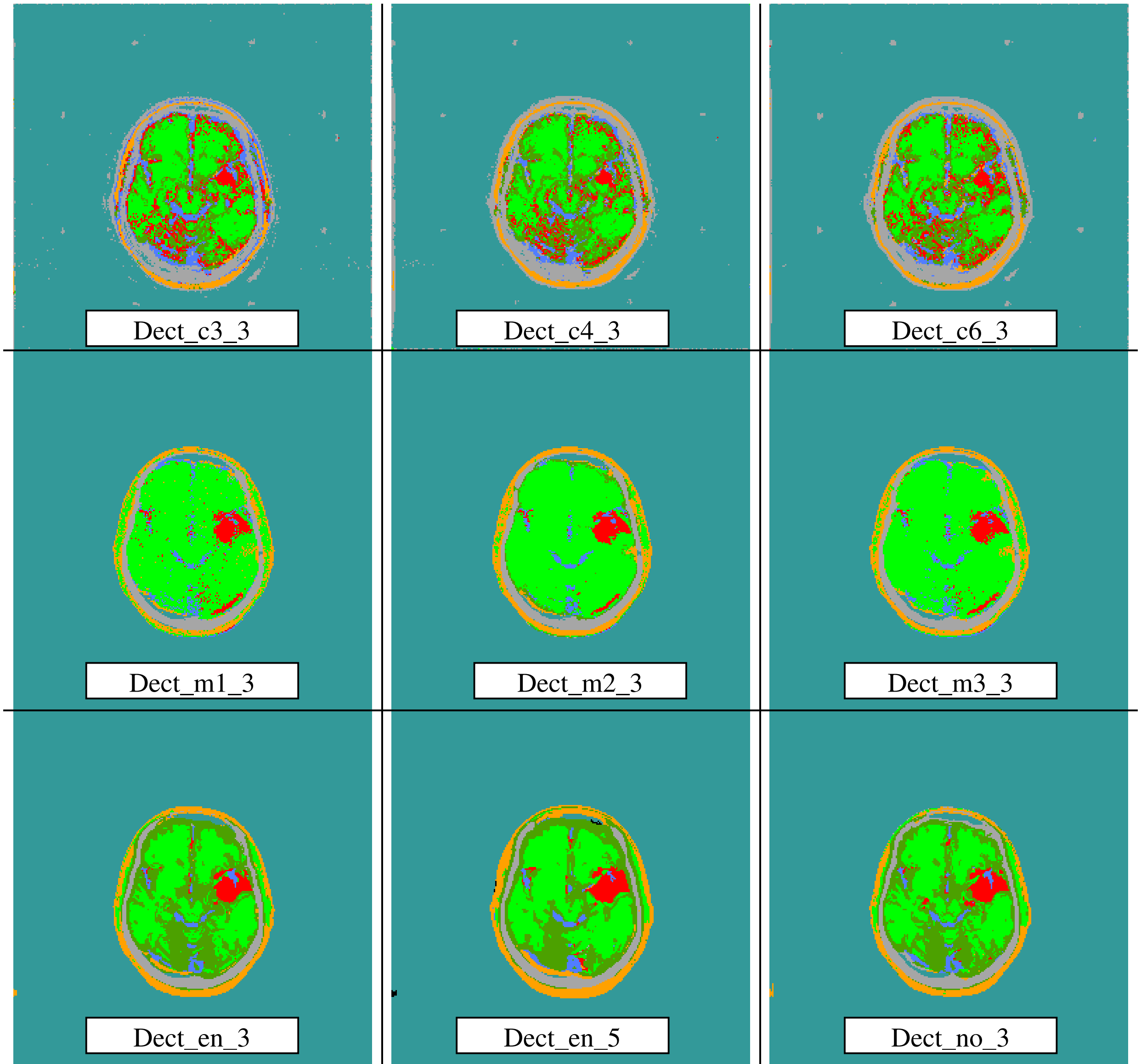
Repräsentativ für alle Schichtbilder der Originalsequenzen wurde Schicht 10 der Gehirnsequenz *da55* ausgesucht und die Klassifikationsergebnisse der derzeit 11 Segmentierungsalgorithmen Abbildung 2.1 grafisch dargestellt. Jede Farbe entspricht in der dateiinternen Darstellung einem Index, mit dem die „erkannte" Gewebeklasse assoziiert wird.

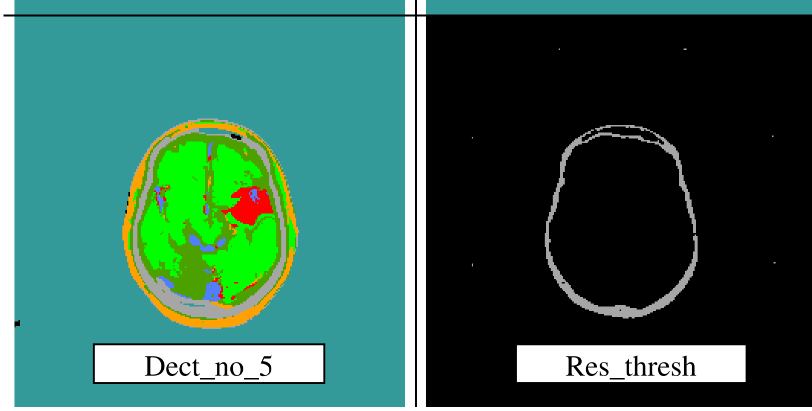
Farbe	Index	Gewebeart	Farbe	Index	Gewebeart
rot	1	Tumor	Orange	5	Fett
blau	2	Liquor	Weiß	6	weiße Substanz
(n.v.)[3]	3	Oedem	Grau	7	graue
grau	4	Knochen	Cyan	8	Hintergrund

Tabelle 2.1 Farbzuordnungstabelle zur Gewebedarstellung

[2] Die Testgebiete liegen digital, in Form von ROI's (Regions of Interest) vor. Sie enthalten u.a. Gewebebereiche, die von Ärzten klassifiziert wurden und dienen als Referenzdaten.
[3] Von der Gewebeklasse Oedem gibt es bisher keine Testdaten. Eine Klassifizierung ist momentan nicht möglich und deshalb in den Resultatsequnzen (Abbildung 2.1) nicht zu finden.

2.1 Klassifizierte Resultatbilder einer Schicht





Bei den 11 Schnitten handelt es sich jeweils um Schicht Nr. 10 der Sequenz da55. Die Bearbeitung erfolgte durch die in den Bildunterschriften aufgeführten Segmentierungsalgorithmen. Die Bildsätze existiern für jede Schicht und für alle Bildssequenzen.

Abbildung 2.1 Überblick der klassifizierten Eingangssequenzen

Neben den Originalaufnahmen und Resultatbildern der Segmentierungsalgorithmen stehen außerdem noch sogenannte ROI's (Regions of Interest) zur Verfügung, dabei handelt es sich vorwiegend um Gebiete des menschlichen Gehirns, die von Ärzten untersucht und klassifiziert wurden. Das damit vorhandene Expertenwissen kann in Form von Regeln in eine Fuzzy Verknüpfungsstrategie mit einbezogen werden. Durch Vergleichen der Untersuchungsergebnisse mit den Klassifikationsebnissen der einzelnen Algorithmen kann die Trefferquote und damit die Qualität der Algorithmen beurteilt werden.

Für eine Bewertung der vorhandenen Segmentierungsalgorithmen und der Fuzzy Resultatsequenz ist eine statistische Auswertung der Segmentierungsergebnisse erforderlich, die in Kapitel 3 beschrieben wird.

Für die Verknüpfung der Resultatsequenzen wurde ein Fuzzy Controller entwickelt, mit dem es möglich ist, die Resultatsequenzen auf der Basis von Fuzzy Logik sinnvoll zu verknüpfen und anschließend auszuwerten. Am Ende soll einen Fuzzy Controller mit einer „optimalen" Regelbasis zur Verfügung stehen, der ein möglichst perfektes Segmentierungsergebnis erzielt werden kann.

Die bisher betrachteten Beispiele bestanden immer aus Fuzzy Terms, hinter denen eine lineare Ordnung stand. So galt beispielsweise im Badewasserbeispiel die Relation kalt < kühl < warm < heiß. Das Mischen von kaltem und warmem Wasser würde bei der Defuzzifizierung einen Wert ergeben, der in hohem Maße dem zwischen *kalt* und *warm* liegenden Term *kühl* zugeordnet werden könnte.
Die in den Segmentierungsergebnissen vorkommenden Zahlenwerte, die einen Index für die jeweils segmentierte Gewebeart darstellen, sind nicht linear geordnet. Eine einzige linguistische Variable „Gewebeart", welche die linguistischen Terms der verschieden Gewebeindizes aufnehmen kann, würde somit sicherlich keine korrekten Segmentierungsergebnisse liefern.

Zur Vereinfachung des Problems soll deshalb zunächst ein modifizierter Mamdani Controller zur Erkennung von nur einer einzigen Gewebeart herangezogen werden. Es stellt sich hier die Frage, wie die Eingangsvariablen und Ergebnissvariablen und die Zugehörigkeitsfunktionen auszusehen haben.

Jeder Algorithmus liefert zwei Informationen, zum einen den ermittelten Farbindex, zum anderen - indirekt, nach vorheriger statistischer Auswertung der ROI's - wie zuverlässig das Resultat eines bestimmten Algorithmus für die Erkennung der betreffenden Gewebeklasse ist. Diese Zuverlässigkeit (Trefferwahrscheinlichkeit) kann für jede Gewebeklasse in Form von Singletons implementiert werden, deren Höhen die klassenspezifischen Trefferwahrscheinlichkeiten repräsentieren (Siehe Abbildung 2.2). Singletons eignen sich deshalb besonders gut, weil die Menge der möglichen Werte für eine Eingangsvariable aus ganzzahligen Klassenzugehörigkeitsindizes des betreffenden Segmentierungsalgorithmus besteht. Es erweist sich daher als sinnvoll, anstelle von Farbindizes, die Trefferwahrscheinlichkeiten der Segmentierungsalgorithmen für die einzelnen Gewebeklassen zu fuzzifizieren.

Der Fuzzifizierungsmechanismus eines einfachen Fuzzy Controllers zur Erkennung der verschiedenen Gewebeklassen könnte beispielsweise wie in Abbildung 2.2 aussehen. Das Singletone Set wurde eigens auf die Trefferwahrscheinlichkeit des Segmentierungsalgorithmus, im betrachteten Beispiel DECT_C4_3, angepaßt. Hat der Segmentierungsalgorithmus ein Rasterelement klassifiziert, so dient es als (scharfer) Eingangswert für den Fuzzy Mechanismus. In Abbildung 2.2 liefert der Algorithmus den Index „2“, der für die Gewebeklasse *Liquor* steht. Bei der Fuzzifizierung wird nun für jede Gewebeklasse der betreffenden linguistischen Eingangsvariablen ein Zugehörigkeitswert ermittelt, der hier trivialerweise für alle Klassen außer Liquor den Zugehörigkeitswert Null liefert. Der erkannten Gewebeklasse Liquor wird ein Zugehörigkeitswert μ in Höhe der für die Gewebeklasse ermittelten Trefferwahrscheinlichkeit zugeordnet.

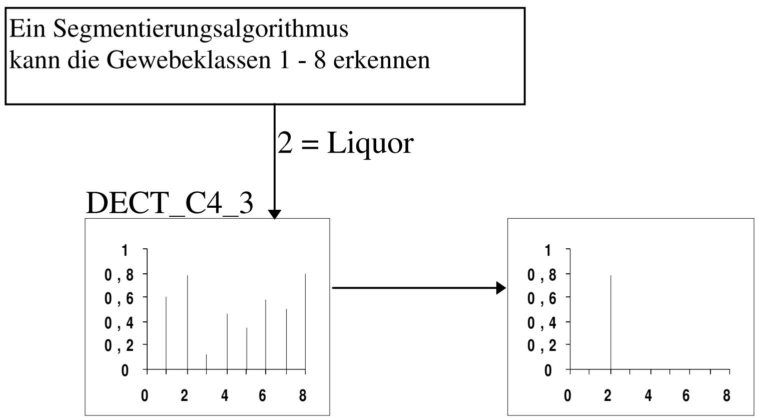


Abbildung 2.2 - Fuzzifizierung mittels Singletons

Der Inferenzvorgang gestaltet sich nun folgendermaßen:
Wenn das Ereignis DECT_C4_3 = 2 Einfluß auf das Fuzzy Resultatbild nehmen soll, so muß mindestens eine Regel existieren die dem Ereigneis eine weitere Aktion zuordnet. Dies geschieht in Form einer Fuzzy Regel R_1:

R_1 : *if DECT_C4_3 is Liquor* *then KLASSE_LIQUOR is hoch*
R_2 : *if DECT_C4_3 is Tumor* *then KLASSE_TUMOR is hoch*
...

Dabei wird das Fuzzy Set *hoch* der linguistischen Variablen (z.B. *KLASSE_LIQUOR*) mit einer Funktion $f_3(\mu(B)_i,x_i)$ entsprechend der Max/Min oder Max/Produkt Methode (Kapitel 1, Seite 22) modifiziert. Der Inferenzvorgang umfaßt die Auswertung aller in der Regelbasis vorhandenen Regeln.

Die linguistische Variable *KLASSE_LIQUOR* enthält z.B. drei Fuzzy Sets, die durch die linear geordneten Terms *gering, mittel, hoch* repräsentiert werden. Die einzelnen Sets werden - wie im Badewasserbeispiel (Kapitel 1) - durch Dreiecksfunktionen beschrieben. Analog der Variablen *KLASSE_LIQUOR* existieren - je nach Anzahl maximal möglicher Klassen - noch weitere linguistische Variablen (*KLASSE_TUMOR*, *KLASSE_KNOCHEN* , ...), die die selbe Struktur aufweisen wie die gerade angeführte Variable *KLASSE_LIQUOR*.

Anschließend werden die Resultat Fuzzy Sets den einzelnen linguistischen Variablen (*KLASSE_TUMOR*, *KLASSE_LIQUOR KLASSE_KNOCHEN* ,...) defuzzifiziert.
Für jede Resultatklasse gibt es eine linguistische Resultatvariable, aus deren Benennung die zu assoziierende Gewebeklasse abgeleitet werden kann, d.h. es existieren genausoviele linguistische Resultatvariablen wie Gewebeklassen. Als Ergebnis nach der Defuzzifizierung erhält man für jede linguistische Resultatvariable einen scharfen Zahlenwert, der aussagt, in welchem Maße das Ergebnis der betreffenden Klasse zugeordnet werden kann. Liefert der verwendete Segmentierungalgsorithmus als Eingangsvariable beispielsweise den Wert (2 = Liquor), so erzeugt die linguistische Variable KLASSE_LIQUOR nach dem Defuzzifizierungsvorgang den höchsten scharfen Resultatwert. Voraussetzung ist natürlich, daß die Regelbasis entsprechend sinnvoll gewählte Zuordnungsregeln enthält und daß alle Resultatvariablen gleichartig aufgebaute Zugehörigkeitsfunktionen enthalten, denn nur so können die scharfen Zahlenwerte untereinander verglichen und interpretiert werden.
Mit dem Maximum über alle scharfen Resultatwerte kann die Klasse mit der höchsten Wahrscheinlichkeit (in Bezug auf Übereinstimmung mit der Realität) identifiziert werden.
Setzt man nur einen einzigen Segmentierungsalgorithmus ein, so werden die Klassifizierungsergebnisse (die die Eingangsvariablen bei der Fuzzifizierung darstellen) mit dem Fuzzy Ergebnis exakt übereinstimmen.

Wesentlich interessanter ist der Einsatz mehrerer Resultatsequenzen von verschiedenen Segmentierungsalgorithmen, die zum Teil unterschiedliche Ergebnisse in den Eingangswerten aufweisen können (Abbildung 2.3 Verknüpfungsmechanismus zur Gewebesegmentierung). Jedem Segmentierungsalgorithmus werden - entsprechend der Trefferquote für die Segmentierungsqualität der verschiedenen Klassen - eine Menge von Singleton Sets zugeordnet. Die Menge der Resultat Fuzzy Sets bleibt unverändert. Besonders, wenn viele Fuzzy Regeln existieren, erzeugen die unterschiedlichen Singleton Sets Zugehörigkeitswerte, mit denen u.a. auch mehrmals gleiche Terms der Resultat Fuzzy Sets modifiziert werden sollen. Nach dem Inferenzvorgang modifiziert maximal ein einziger Zugehörigkeitswert $\mu(B)$ (der beim Inferenzvorgang aus den unterschiedlichen Zugehörigkeitsvariablen gebildet wird) das betreffende Fuzzy Set. In der Aussagenlogik würde man von sich widersprechenden Regeln sprechen, im vorliegenden Fall wird aus den unterschiedichen Zugehörigkeitsgraden der betreffenden Regeln das arithmetische Mittel gebildet.
Anschließend erfolgt der Defuzzifizierungsvorgang über die Resultat Fuzzy Sets. Ein Auswahlmechanismus (Maximum) über die scharfen Resultatwerte kann dann die *richtige* Ergebnisklasse identifizieren.

Die Blockdarstellung in Abbildung 2.3 enthält die schon erläuterte Fuzzifizierungs- bzw. Verknüpfungsstrategie für mehrere Eingangsvariablen. Jeder Eingangsvariablen ist ein auf den Segmentierungsalgorithmus abgestimmtes Singleton Set zugeordnet, so daß der daraus ermittelte Zugehörigkeitsgrad die Trefferwahrscheinlichkeit des Segmentierungsalgorithmus für betrachtete Klasse widerspiegelt. Die in der Regelbasis festgelegten Fuzzy Regeln verknüpfen die Eingangszugehörigkeitswerte der Singleton Sets mit den verschiedenen Fuzzy Operatoren selektiv und ordnen die so entstandenen Ergebniszugehörigkeitswerte den Resultatklassen zu.

Damit jedes Resultat Fuzzy Set mit dem entsprechenden Zugehörigkeitswert modifiziert wird, wird auch für jede Klasse nach der Defuzzifizierung ein scharfer Resultatwert erzeugt. Der höchste scharfe Resultatwert unter den verschiedenen Gewebeklassen gibt an, daß das Segmentierungsergebnis auf die betreffende Klasse mit der größten Wahrscheinlichkeit zutrifft. Auf diese Art und Weise kann jedes Rasterelement klassifiziert werden. Voraussetzung ist selbstverständlich, daß die Resultatpixel der verschiedenen Segmentierungsalgorithmen in ihrer Lage exakt übereinstimmen. Dies ist jedoch bei den vorhandenen Datensequenzen sichergestellt.

Blockdarstellung zur Verknüpfung der Segmentierungsergebnisse mittels Fuzzy Logik

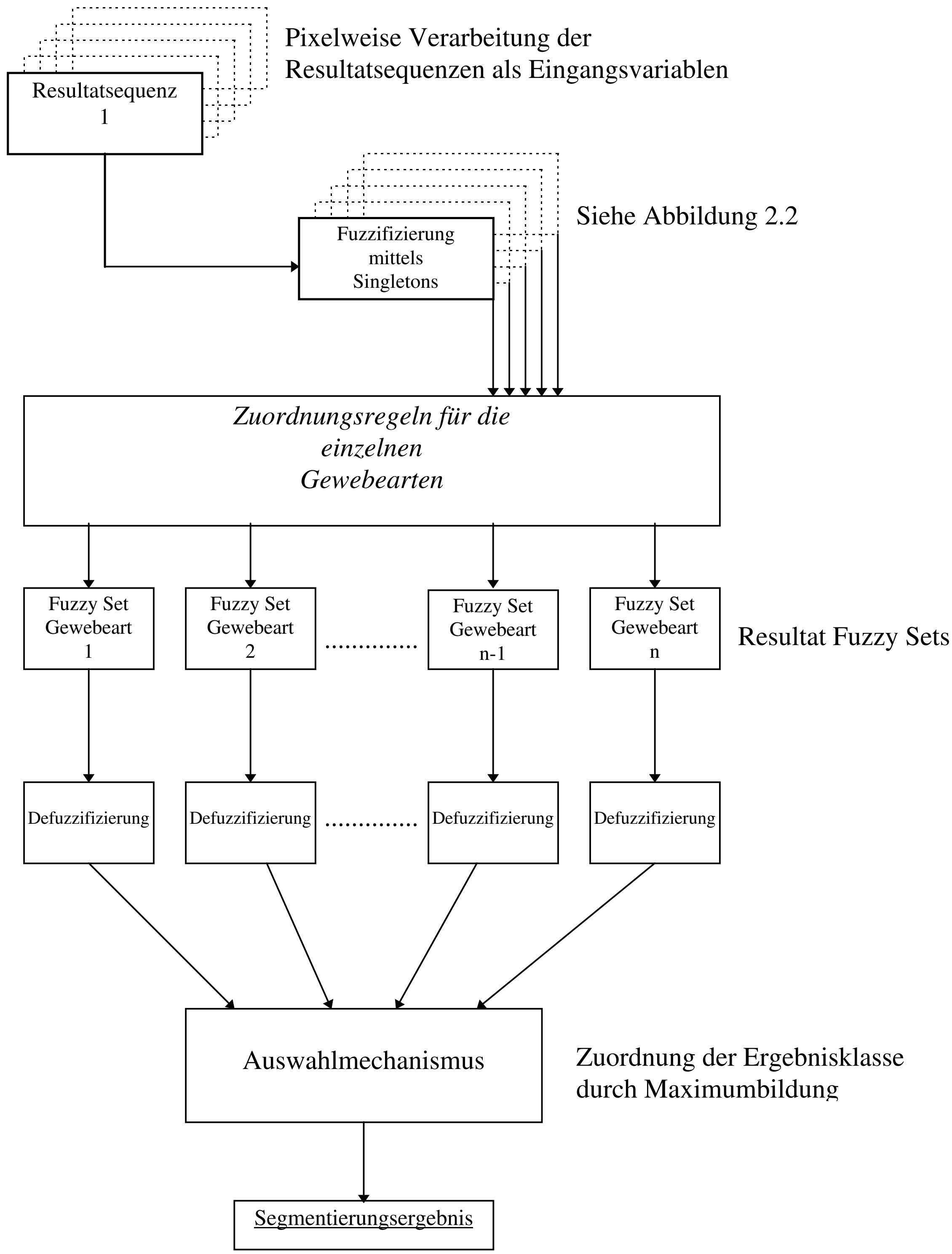


Abbildung 2.3 Verknüpfungsmechanismus zur Gewebesegmentierung

2.1.1 Diskussion über die Form der Fuzzy Sets einer Resultatvariablen und der verwendeten Inferenzmethode

Das durch die Fuzzy Verknüpfung erzeugte Fuzzy Resultatbild ist abhängig von:

- der Qualität der Segmentierungsalgorithmen
- dem Aussehen der den liguistischen Variablen zugrunde liegenden Fuzzy Sets
- der Qualität der Fuzzy Regeln
- der Wahl der Operatoren beim Inferenzvorgang
- dem Defuzzifizierungsmechanismus

Da die Anzahl der Resultatvariablen mit der Anzahl der Gewebeklassen übereinstimmt, und eine Gewichtung der Zugehörigkeitswerte im Vorfeld über Singleton Sets stattfindet, sollten alle Resultat Fuzzy Sets der unterschiedlichen linguistischen Variablen gleich aussehen, ansonsten lassen sich nach einer Defuzzifizierung keine Aussagen mehr machen, welcher Klasse das Segmentierungsergebnis zuzuordnen ist. Es wurden deshalb zunächst die bewährten Dreiecksfunktionen verwendet, die die Eingangswerte linear auf ihre Zugehörigkeitswerte abbilden. Eine linguistische Resultatvariable enthält die drei Fuzzy Terme *niedrig, mittel* und *hoch*.

Bei der verwendeten Inferenzmethode hat sich bei empirischen Untersuchungen gezeigt, daß die Arithmetik/Produkt Methode für das gestellte Segmentierungsproblem am besten geeignet ist. Dies kann auch logisch begründet werden: Liefern mehrere Fuzzy Regeln Zugehörigkeitswerte für einen bestimmten Term einer linguistischen Variablen, so kommt der Inferenzoperator zum Einsatz. Würde man als Inferenzoperatoren Zadehs Max oder Min Operatoren wählen, so würde nur die eine Fuzzy Regel mit dem größten bzw. kleinsten Zugehörigkeitswert den Resultatterm modifizieren. Bei den zur Gewebesegmentierung eingesetzten Fuzzy Regeln soll jede einzelne Regel zur Verbesserung des Segmentierungsergebnisses beitragen, so daß eventuelle Überbewertungen einer Regel durch andere Regelergebnisse kompensiert werden können. Eine lineare Kompensation kann durch den Einsatz des arithmetischen Mittels erreicht werden. Beim Verwenden multiplikativer Operatoren, wie zum Beispiel dem arithmetische Produkt werden, je mehr Regeln für eine Zugehörigkeitsfunktion existieren, die Zugehörigkeitswerte bei der Produktbildung immer kleiner, so daß Regeln für andere Terms indirekt höher bewertet werden, was ebenfalls nach der Defuzzifizierung zu Fehlklassifikationen führen kann. Das arithmetische Mittel hat sich in dieser Hinsicht als am zuverlässigsten erwiesen.

Für die, die Fuzzy Sets modifizierende Funktion f_3 wurde die Produkt Methode ausgewählt. Da die Zugehörigkeitsfunktionen der die Gewebeklassen repräsentierenden linguistischen Variablen exakt gleich aussehen, spielt es keine Rolle ob der Minimumoperator (Clipping) oder der Produktoperator (Skalierung) als modifizierende Funktionen eingesetzt werden. Nach dem Defuzzifizierungsvorgang entstehen zwar andere (scharfe) Zahlenwerte, es ändert sich allerdings nicht die Reihenfolge, die man durch ordnen der Zahlen nach ihrer Größe erhalten würde. Da für die Klassifikation nur der größte Zahlenwert eine Rolle spielt, ist die Wahl des modifizierenden Operators nicht relevant. Eine Skalierung der Zugehörigkeitsfunktionen benötigt allerdings weniger Rechenzeit als eine Modifikation durch Clipping. Es wurde deshalb als modifizierende Funktion der Produkt Operator ausgewählt.

Ein Fuzzy Regelsatz für vier Segmentierungsalgorithmen könnte beispielsweise folgendermaßen aussehen:

Beispiel 2.1 Ein einfacher Fuzzy Regelzatz zur Gewebesegmentierung

if DECT_C6_3 tumor DECT_M1_3 tumor AND DECT_C4_3 tumor DECT_M2_3 tumor AND AND then TUMOR hoch
if DECT_C6_3 knochen DECT_M1_3 knochen AND DECT_C4_3 knochen DECT_M2_3 knochen AND AND then KNOCHEN hoch
if DECT_C6_3 fett DECT_M1_3 fett AND DECT_C4_3 fett DECT_M2_3 fett AND AND then FETT hoch
if DECT_M2_3 liquor DECT_M1_3 liquor AND then LIQUOR hoch

if DECT_M3_3 grau DECT_M2_3 grau AND DECT_M1_3 grau OR then grau mittel			
if DECT_M3_3 weiss	then WEISS	mittel	GRAU mittel
if DECT_M1_3 tumor	then tumor	mittel	
if DECT_M2_3 liquor	then liquor	mittel	
if DECT_M1_3 fett	then fett	mittel	
if DECT_M3_3 knochen	then knochen	mittel	
if DECT_M3_3 back	then back	mittel	

(Die linguistischen Eingangsvariablen DECT_C6_3, DECT_M1_3, DECT_C4_3, DECT_M2_3 wurden entsprechend der Namen ihrer Segmentierungsalgorithmen benannt. Die in Kleinbuchstaben geschriebenen Gewebesorten sind die Terme, welche in Verbindung mit der entsprechenden linguistischen Variablen einen Zugehörigkeitswert liefern. Mit den in Großbuchstaben geschriebenen Gewebesorten werden die linguistischen Resultatvariablen für die entsprechende Gewebeart assoziiert.)

Bei dem für die Verknüpfungen entwickelten Fuzzy Tool werden die Fuzzy Regeln in klammerfreier polnischer Notation (UPN) dargestellt. Fuzzy Operatoren sind zwar assoziativ und kommutativ, dadurch wird es jedoch möglich, verschiedene Operatoren in einer Regel zu verwenden. Bei UPN steht der Operator immer hinter den Operanden. Für ein näheres Verständnis von UPN sei an dieser Stelle auf die ausreichend vorhandene Fachliteratur verwiesen.

Man beachte, daß die durch viele AND Operatoren (arithmetisches Produkt) verknüpften Terme die Fuzzy Sets *hoch* beeinflussen. Es gibt relativ wenig Pixel, bei denen jeder der verwendeten Resultatsequenzen das selbe Segmentierungsergebnis aufweist. Sollte dies jedoch der Fall sein, so kann mit sehr hoher Wahrscheinlichkeit angenommen werden, daß es sich um das richtige Segmentierungsergebnis handelt. Die Terms *hoch* werden aus dem Grund modifiziert, weil sie bei der Defuzzifizierung stärker in das Gesamtergebnis eingehen, als beispielsweise die Terme *mittel* oder *niedrig*.

Da leider nur sehr wenig Pixel sicher erkannt werden können, erscheint es sinnvoll, weitere „tolerantere“ Regeln zu generieren, deren Segmentierungseigenschaften zwar geringer sind, die jedoch erheblich mehr Pixel klassifizieren. Diese Regeln modifizieren die Terme *mittel* der betreffenden Resultatvariablen. Die einzelnen Regeln werden somit in verschiedene Qualitätsklassen unterteilt, deren Zuordnung analog der linearen Ordnung in den Resultat Fuzzy Sets erfolgt. Eine Fuzzy Regel kann auch mehrere Fuzzy Sets modifizieren, dies wird daran ersichtlich, daß hinter „*then*“ auch mehr als ein Ausdruck, bestehend aus linguistischer Variable und Fuzzy Term, stehen kann.

2.2 Vergleich mit anderen Verknüpfungsstrategien

Es gibt im wesentlichen zwei Methoden der pixelweisen Verknüpfung mehrerer ähnlicher Eingangsbilder zu einem Resultatbild. Dies ist zum einen die binäre Verknüpfung, bei der das Resultatpixel der Klasse zugeordnet wird, die am häufigsten in den Eingangsvariablen enthalten ist, und zum anderen die gewichtete Verknüpfung, bei der die verschiedenen Eingangssequenzen und deren einzelne Klassen individuell gewichtet werden können und das Resultatpixel der Klasse zugeordnet wird, die beim klassenweisen Aufaddieren der Gewichte den höchsten Wert aufweist.

Mit der vorgestellten Fuzzy Verknüpfungsstrategie sind durch die freie Wahl der Zugehörigkeitsfunktionen und Fuzzy Regeln beide Methoden realisierbar. Verwendet man als Eingangszugehörigkeitsfunktionen anstelle von Dreiecksfunktionen Rechteckfunktionen der Höhe 1, so werden sämtliche Eingangsvariablen auf die Zugehörigkeitswerte 0 oder 1 abgebildet und das Fuzzy Tool arbeitet binär. Ebenso verhalten sich sämtliche Fuzzy Operatoren für die Werte 0 und 1 genauso wie die hinreichend bekannten booleschen Operatoren.

Für eine gewichtete Verknüpfung ähnlicher Eingangsbilder können die in den Singletons definierten Höhen der einzelnen Klassen als Gewichte der Eingangsvariablen interpretiert werden, die mit den entsprechenden Fuzzy Regeln auf die Resultatvariablen abgebildet werden können.

Der wesentliche Vorteil bei Verknüpfungen auf Basis von Fuzzy Logik ist die freie Wahl der Zuordnungsstrategie, die in Form von Regeln und der verwendeten Fuzzy Operatoren individuell auf das gestellte Problem angepaßt werden kann. Insbesondere können durch entsprechende zusätzliche Fuzzy Regeln bestimmte Muster in den Eingangsvariablen erkannt und in den Resultatvariablen entsprechend berücksichtigt werden. Außerdem besteht die Möglichkeit kompensierende Regeln zu entwerfen, die eventuelle Überbewertungen durch andere Regeln ausgleichen können. Da die Regelbasis sehr schnell sehr groß und kompliziert werden kann, steigt die Komplexität und damit auch die Wahrscheinlichkeit, daß informationstrennende Regeln mit für das Segmentierungsergebnis negativen Seiteneffekten erzeugt werden können. Um dem vorzubeugen werden Methoden benötigt, mit denen die Regelerzeugung systematisch durchgeführt werden kann. In Kapitel 4 werden zwei Methoden der Regelgenerierung vorgestellt. Fuzzy Systeme sind sehr rechenintensiv, da für jedes zu klassifizierende Pixel anhand sämtlicher Eingangszugehörigkeitsfunktionen Zugehörigkeitswerte erzeugt werden müssen, anschließend muß die gesamte Regelbasis abgearbeitet werden und beim Defuzzifizieren müssen mehrere flächengewichtete Schwerpunkte berechnet werden. Fuzzy Logik sollte aus dem Grund nur für Probleme eingesetzt werden, bei denen herkömmliche Berechnungsverfahren nicht die gewünschten Ergebnisse liefern.

3 Datenanalyse

Ein Großteil dieser Arbeit beschäftigt sich mit der Verknüpfung von Segmentierungsergebnissen auf der Basis von Fuzzy Logik. Dabei werden die Resultatbilder der verschiedenen Segmentierungsalgorithmen auf der Grundlage von in Regeln gefaßtem Expertenwissen pixelweise miteinander verknüpft. Es werden also nicht direkt die bei der Tomographie entstandenen Originalbilder[4] ausgewertet, sondern die Resultatbilder der zur Verfügung stehenden Segmentierungsalgorithmen. Die grundlegenden Prinzipien der Arbeitsweise dieser Algorithmen soll im folgenden kurz erläutert werden.

Um möglichst gute Segmentierungsergebnisse zu erhalten, ist es notwendig genügend Informationen über das einzelne Pixel bereitzustellen, so daß eine sichere Zuordnung ermöglicht wird. Alle verwendeten Segmentierungsalgorithmen nutzen dabei die Methoden der *texturorientierten Bildanalyse*. Der Begriff *texturorientierte Bildanalyse* beinhaltet dabei folgende zu klärenden Aspekte: Zum einen die texturorientierte Vorgehensweise, d.h. die pixelweise Betrachtung von Objekten in Rasterbildern, wobei außer dem zu untersuchenden Bildpunkt dessen lokale Umgebung zu betrachten ist. Zum anderen beinhaltet die Bildanalyse die Verfahren, welche die zu untersuchenden Pixel einerseits in Gruppen ähnlicher Eigenschaften zusammenfassen (Clusteranalyse) und andererseits zu jedem Pixel eine Aussage bezüglich seiner Klassenzugehörigkeit treffen (Klassifikation).
Eine wichtige Beziehung besteht zwischen Merkmalsextraktion und Klassifikation. Je schwieriger es fällt, selbst durch komplexe Verfahren der Merkmalsextraktion die Daten zu beschreiben, desto größer werden die Anforderungen an den eingesetzten Klassifikator. Bei den zu untersuchenden Gewebedaten gestaltet sich die Merkmalsextraktion trotz des Vorhandenseins multidimensionaler Daten als extrem schwierig. Als Klassifikatoren kommen die beiden bewährten neuronalen Netztypen Backpropagation und Kohonen Feature Map zum Einsatz.

Ein neuronales System besteht aus einer Menge von Verarbeitsungseinheiten (auch Units oder Neuronen genannt) sowie aus der Netzwerkstruktur, über die die Elemente miteinander verbunden sind und sich gegenseitig beeinflussen können. Die Struktur wird dargestellt durch eine Konnektions- oder Gewichtmatrix W bei der jeder einzelne Eintrag das Gewicht der Verbindung eines Neurons zu einem anderen angibt. Dabei wird nach der Merkmalstrennung der entstandene Merkmalsvektor auf die Eingabeneuronen übertragen. Nach der Merkmalsübertragung bestimmt eine sogenannte Aktivierungsfunktion F für jedes Neuron einen Aktivierungszustand zu einem bestimmten Zeitpunkt, anschließend liefert eine Outputfunktion f die Abhängigkeit vom aktuellen Grad der Aktivierung, anhand der eine Klassifizierung des betrachtenen Eingangspixels möglich ist. Für ein grundlegendes Verständnis über Topologie und Arbeitsweise neuronaler Netzwerke wird auf die Fachliteratur verwiesen, beispielsweise [KOHON84].

[4] Kapitel 5 befaßt sich mit einer Klassifikation der Originaldaten auf Fuzzy Basis

3.1 Klassifikatoren

Tabelle 3.1 soll einen groben Überblick zur verwendeten Art der Merkmalsextraktion und der Texturgröße geben. Die Angabe zur Texturgröße ist vierdimensional (*x,y,b,z*). Der Parameter *b* enthält die Anzahl der verwendeten Bildkanäle einer multidimensionalen Sequenz. Zur Zeit besteht eine Originalsequenz maximal aus drei Bildkanälen (CT,PW,T2).

<u>Für die Merkmalsextraktion werden verwendet:</u>

Verfahren 1. Ordnung : statistische Momente (MOMENT)
Verfahren 2. Ordnung : Cooccurrence Matrizen (COOC), Texturenergie (ENERGY)
und Verfahren der impliziten Merkmalsextraktion.

Algorithmus	Alias	Merkmalsextraktion	Texturgröße
dect_c3_3	ALG0	COOC	3*3*2*3
dect_c4_3	ALG1	COOC	3*3*2*3
dect_c6_3	ALG2	COOC	3*3*2*3
dect_m1_3	ALG3	MOMENT	3*3*2*3
dect_m2_3	ALG4	MOMENT	3*3*2*3
dect_m3_3	ALG5	MOMENT	3*3*2*3
dect_en_3	ALG6	ENERGY	3*3*2*3
dect_en_5	ALG7	ENERGY	5*5*2*3
dect_no_3	ALG8	implizite Merkmalsextraktion	3*3*2*3
dect_no_5	ALG9	implizite Merkmalsextraktion	5*5*2*3
res_thresh	ALG10	lineare Separierung	-

Tabelle 3.1 Merkmalsextraktion und Texturgröße der eingesetzten Algorithmen

Bei *res_thresh* handelt es sich um einen Algorithmus der eine lineare Separierung des Knochens aus den CT Bändern der Originalsequenzen durch einen unteren und oberen Schwellwert in der Houndsfieldeinheit (Signalintensität) vornimmt. Dieses einfache Segmentierungsverfahren, das lediglich einen definierten Bereich von Houndsfieldeinheiten des Originalbildes auf den Klassencode für Knochen in den Resultatsequenzen abbildet, liefert naturgemäß für Knochen sehr gute Erkennungsraten. Alle weiteren Gewebetypen werden von dem Verfahren nicht erkannt. Nähere Informationen zum Verfahren können aus [WEBB95] entnommen werden.

Alle hier aufgeführten Bildsegmentierungsalgorithmen gewinnen die zur Segmentierung notwendige trennende Information aus Originalsequenzen und erzeugen einkanalige Resultatsequenzen (b=1), die als Eingabedaten zur Verknüpfung durch Fuzzy Logik zur Verfügung stehen.

3.2 Statistische Auswertung der Resultatsequenzen

Zur Bewertung der Segmentierungsalgorithmen wurde ein Auswerteprogramm erstellt, mit der man die in den ROI's gespeicherten Trainingsdaten mit den in den Resultatsequenzen gespeicherten Klassifikationsergebnissen der unterschiedlichen Segmentierungsalgorithmen vergleichen kann.

Eine statistische Erhebung über die Resultatsequenzen ist aus folgenden Gründen erforderlich:

- Die Eingangsvariablen eines Fuzzy Verknüpfungstools sollen mit den statistisch ermittelten Erwartungswerten gewichtet werden.

- Zum Erstellen von Regeln in einer Regelbasis ist eine möglichst genaue Kenntnis der Resultatsequenzen und Trefferwahrscheinlichkeiten der Segmentierungsalgorithmen notwendig.

- Durch Vergleiche der vom Fuzzy Tool erzeugten Klassifikationsdaten mit den in den ROI's gespeicherten Daten kann die Qualität des Verknüpfungstools beurteilt werden.

Da die Algorithmen verschiedene Gewebearten unterschiedlich gut erkennen, wurden die Erkennungsquoten für die unterschiedlichen Gewebeklassen jeweils getrennt aufgeführt.
Die Spalten in Tabelle 3.2 enthalten die Klassifikationsergebnisse des in der Spaltenüberschrift betrachteten Segmentierungsalgorithmus. Die Zeilen enthalten die Anzahl der richtig erkannten Pixel einer Gewebeklasse. Es wurden nicht die gesamten anatomischen Daten untersucht, sondern lediglich die Testgebiete, denn nur diese lassen einen Vergleich mit den berechneten Klassifikationsergebnissen zu. Die Spaltenüberschrift „Original“ stellt keinen Algorithmus dar, sondern sie enthält die vom Experten klassifizierten Testdaten und stellt damit Referenzwerte zur Verfügung.

Tabelle 3.2 Richtig erkannte Pixel (Anzahl)

Algorithmus / Klasse	Original (real)	dect_ c3_3	dect_ c4_3	dect_ 6_3	dect_ m1_3	dect_ m2_3	dect_ m3_3	dect_ en_3	dect_ en_5	dect_ no_3	dect_ no_5	res_ thresh
Tumor	2284	1325	1415	1539	2130	2143	2167	2057	2134	2084	2094	0
Liquor	1551	1407	1424	1383	1100	1261	1228	1462	1443	1472	1432	0
Oedem												
Knochen	7035	5679	6314	6359	6674	6685	6743	6947	6944	6948	6934	6997
Fett	3603	3257	3281	3191	2439	3195	3152	3448	3488	3415	3502	0
graue Sub.	1576	712	1131	974	0	143	10	1185	1311	1166	1265	0
weiße Sub.	5884	5396	5417	5445	5805	5868	5867	5607	5691	5661	5774	0
Hintergrund	27628	26818	27173	26977	27620	27623	27627	27628	27626	27628	27628	0
Summe	***49561***	***44594***	***46155***	***45868***	***45768***	***46918***	***46794***	***48334***	***48637***	***48374***	***48629***	***6997***

Für die Klasse Oedem existieren derzeit keine ROI's mit Testdaten. Die Klasse wurde jedoch der Vollständigkeit halber mit aufgeführt. In Tabelle 3.3 werden die Segmentierungsergebnisse prozentual bezüglich den Trainingsdaten aus den ROI's (Tabelle 3.2 Spalte Original = 100 %) dargestellt. Die jeweils besten Klassenergebnisse wurden hervorgehoben.

Tabelle 3.3 richtig erkannte Pixel (Prozentual)

Algorithmus / Klasse	dect_ c3_3	dect_ c4_3	dect_ 6_3	dect_ m1_3	dect_ m2_3	dect_ m3_3	dect_ en_3	dect_ en_5	dect_ no_3	dect_ no_5	res_ thresh
Tumor	58,01%	61,95%	67,38%	93,26%	93,83%	**94,88%**	90,06%	93,43%	91,24%	91,68%	
Liquor	90,72%	91,81%	89,17%	70,92%	81,30%	79,17%	94,26%	93,04%	**94,91%**	92,33%	
Oedem											
Knochen	80,72%	89,75%	90,39%	94,87%	95,02%	95,85%	98,75%	98,71%	98,76%	98,56%	**99,46%**
Fett	90,40%	91,06%	88,57%	67,69%	88,68%	87,48%	95,70%	96,81%	94,78%	**97,20%**	
graue Sub.	45,18%	71,76%	61,80%	0,00%	9,07%	0,63%	75,19%	**83,19%**	73,98%	80,27%	
weiße Sub.	91,71%	92,06%	92,54%	98,66%	**99,73%**	99,71%	95,29%	96,72%	96,21%	98,13%	
Hintergrund	97,07%	98,35%	97,64%	99,97%	99,98%	100,00%	100,00%	99,99%	100,00%	**100,00%**	
gesamt	***89,98%***	***93,13%***	***92,55%***	***92,35%***	***94,67%***	***94,42%***	***97,52%***	***98,14%***	***97,60%***	***98,12%***	***14,12%***

Tabelle 3.3 spiegelt nicht die wirkliche Qualität der Klassifikationsergebnisse wieder. So erkennt Algorithmus dect_m2_3 (Abbildung 3.1) die Klasse *weiße Gehirnsubstanz* zwar zu 99,73 % [5868 Pixel] richtig, gleichzeitig klassifiziert der selbe Algorithmus aber auch in 31,15 % [1883 Pixel] der Fälle Gewebe als weiße Gehirnsubstanz, in denen dies nicht der Fall ist (Tabelle 3.4 und Tabelle 3.5). Der Algorithmus ordnet der Klasse insgesamt mehr Pixel zu [5868+1883 = 7751], als in Wirklichkeit vorhanden sind [real: 5884]. Die prozentualen Ergebnisse für richtig und falsch erkannte Pixel addieren sich lediglich im Gesamtergebnis über alle Klassen zu 100 %, nicht jedoch für die einzelnen Klassen.

Tabelle 3.4 falsch erkannte Pixel (Anzahl)

Algorithmus / Klasse	Original	dect_ c3_3	dect_ c4_3	dect_ 6_3	dect_ m1_3	dect_ m2_3	dect_ m3_3	dect_ en_3	dect_ en_5	dect_ no_3	dect_ no_5	res_ thresh
Tumor	0	820	395	509	404	262	314	66	75	63	82	0
Liquor	0	1031	514	561	209	149	166	206	150	178	188	0
Oedem												
Knochen	0	1023	792	1025	142	130	114	63	45	59	42	312
Fett	0	389	385	499	167	91	106	42	41	41	47	0
graue Sub.	0	563	714	510	0	105	3	394	307	337	235	0
weiße Sub.	0	783	531	523	2793	1833	2002	452	306	507	338	0
Hintergrund	0	358	75	66	78	73	62	4	0	2	0	0
Summe	***0***	***4967***	***3406***	***3693***	***3793***	***2643***	***2767***	***1227***	***924***	***1187***	***932***	***42564***

Tabelle 3.5 falsch erkannte Pixel (Prozentual)

Algorithmus / Klasse	dect_ c3_3	dect_ c4_3	dect_ 6_3	dect_ m1_3	dect_ m2_3	dect_ m3_3	dect_ en_3	dect_ en_5	dect_ no_3	dect_ no_5	res_ thresh
Tumor	35,90%	17,29%	22,29%	17,69%	11,47%	13,75%	2,89%	3,28%	2,76%	3,59%	
Liquor	66,47%	33,14%	36,17%	13,48%	9,61%	10,70%	13,28%	9,67%	11,48%	12,12%	
Oedem											
Knochen	14,54%	11,26%	14,57%	2,02%	1,85%	1,62%	0,90%	0,64%	0,84%	0,60%	4,43%
Fett	10,80%	10,69%	13,85%	4,64%	2,53%	2,94%	1,17%	1,14%	1,14%	1,30%	
graue Sub.	35,72%	45,30%	32,36%	0,00%	6,66%	0,19%	25,00%	19,48%	21,38%	14,91%	
weiße Sub.	13,31%	9,02%	8,89%	47,47%	31,15%	34,02%	7,68%	5,20%	8,62%	5,74%	
Hintergrund	1,30%	0,27%	0,24%	0,28%	0,26%	0,22%	0,01%	0,00%	0,01%	0,00%	0,00%

Für eine Verknüpfung der Segmentierungsergebnisse mit dem Fuzzy Tool sollen vorrangig die Algorithmen verwendet werden, welche die qualitativ besten Segmentierungsergebnisse hervorbringen. D.h. es sollen möglichst viele Pixel richtig erkannt werden, gleichzeitig soll der

Algorithmus möglichst wenig Fehler machen. Ein Qualitätsmaß, das beide Aspekte berücksichtigt soll dabei die Auswahl erleichtern. In Abbildung 3.1 wird die Verteilung richtig und falsch erkannter Pixel in Form von Mengen dargestellt.

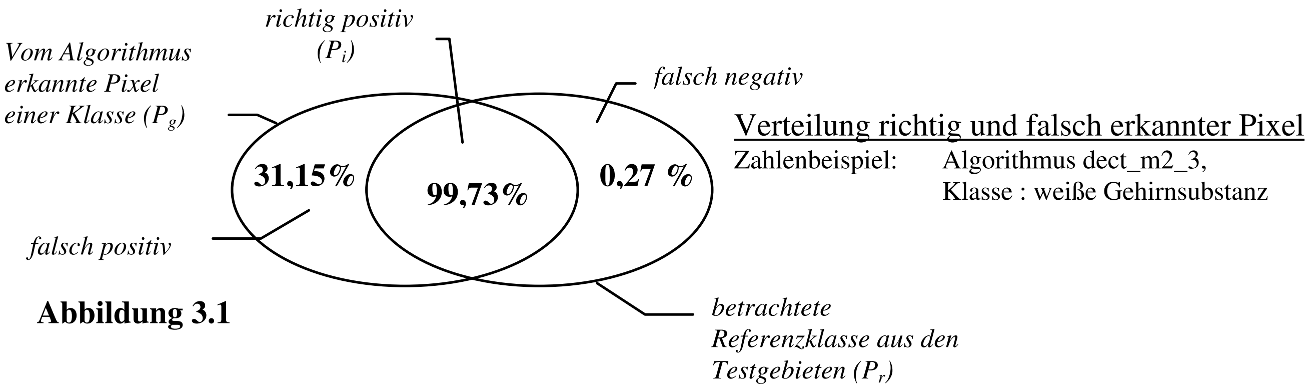


Abbildung 3.1

Die Rasterelemente mit der Bezeichnung *richtig positiv* weisen sowohl in der Menge der Testdaten, als auch in der Menge der vom Algorithmus erkannten Daten die gleiche Klassenzugehörigkeit auf.

Rasterelemente mit der Bezeichnung *falsch negativ* sind zwar in den Testgebieten der betrachteten Gewebeklasse zugeordnet, werden aber vom Algorithmus falsch klassifiziert

Rasterelemente mit der Bezeichnung *falsch positiv* werden vom Algorithmus der betrachteten Klasse zugeordnet, sind aber in den Testgebieten anders klassifiziert.

Sowohl *falsch positiv* als auch *falsch negativ* bewertete Pixel erzeugen nicht nur Fehler in der betrachteten Klasse, sondern die Fehlklassifikation wirkt sich bei der Betrachtung einer anderen Klasse auf diese ebenfalls negativ aus. Aus den Tabellen Tabelle 3.4 und Tabelle 3.5 können Rückschlüsse gezogen werden, wie groß die Klassenfehler der unterschiedlichen Segmentierungsalgorithmen sind.

Genauso interessant ist das Wissen, wenn eine Fehlklassifikation auftrat, bei welcher Klasse der betreffende Fehler gemacht wurde. In Kapitel 6 und im Anhang befinden sich sogenannte Konfusionsmatrizen, aus denen genau hervorgeht, welcher Klasse im Fehlerfall das entsprechende Pixel zugeordnet wurde. Anhand dieser Information kann der auf neuronalen Netzwerken basierende Klassifikator eines Segmentierungsalgorithmus weiter verbessert werden, ebenso können einzelne Fuzzy Regeln der Regelbasis entsprechend angepaßt und verbessert werden.

Aus der Mengendarstellung (Abbildung 3.1) wurde eine Formel (3.1) abgeleitet, die als Qualitätsmaß dienen soll. Das Qualitätsmaß $\boldsymbol{Q_i}$ kann folgendermaßen berechnet werden:

$$\boldsymbol{Q_i = \sqrt{\frac{P_i}{P_g}} * \sqrt{\frac{P_i}{P_r}} = \sqrt{\frac{P_i^2}{P_g * P_r}}} \qquad \textbf{(3.1)}$$

P_i : *richtig positive* Pixel
P_g: Gesamtzahl der erkannten Pixel einer Klasse
P_r: tatsächlich richtige Pixel (ROI Ergebnis)
Q_i: Qualitätsindex

Das Qualitätsmaß wird maximal, wenn der Algorithmus alle erkannten Pixel einer Klasse richtig klassifiziert ($P_i/P_g = 1$) und keine Pixel „übersehen“ werden ($P_i = P_r$). Ein solches Ergebnis liefert zum Beispiel Algorithmus *dect_no_5* bei der Hintergrundklassifikation ($\boldsymbol{Q_i} = 1 \equiv 100$ %).
Der Quotient P_i/P_g drückt das Verhältnis von richtig klassifizierten und Gesamtzahl der vom Algorithmus erkannten Pixel aus, P_i/P_r drückt das Verhältnis von richtig klassifizierten und tatsächlich richtigen Pixeln einer Klasse aus. Interpretiert man das Produkt der beiden Terme als Wahrscheinlichkeit, so läßt sich dieses als Trefferquote oder Qualitätsmaß interpretieren. Die Trefferquote ermöglicht einen Vergleich der einzelnen Klassifikationsergebnisse untereinander. Tabelle 3.6 enthält das Qualitätsmaß für die verschiedenen Klassen.
Für den Spezialfall daß ein Algorithmus genauso viele Pixel einer Klasse ermittelt hat wie in den Testgebieten ausgewiesen ($P_g = P_r$), erhält man als Trefferquote - nach auflösen des Wurzelzeichens - die tatsächliche Wahrscheinlichkeit (P_i / P_r) dieser Klasse.

Die durch das Qualitätsmaß errechneten Trefferquoten spielen quantitativ eine untergeordnete Rolle, wichtig ist lediglich, daß die verschiedenen Segmentierungsergebnisse der jeweiligen Algorithmen untereinander vergleichbar werden.

Je größer das Qualitätsmaß, desto höher ist die Wahrscheinlichkeit, daß das Klassifikationsergebnis der Realität entspricht.

Vergleichbare Lösungen zum Sachverhalt wurden in der Literatur nicht gefunden. In Statistikbüchern, z.B. [LEHN92], findet man jedoch theoretische Ansätze zu den Themen *Nullhypothese, Parameterschätzung, Fehler erster und zweiter Art*.

Algorithmus Klasse	dect_ c3_3	dect_ c4_3	dect_ 6_3	dect_ m1_3	dect_ m2_3	dect_ m3_3	dect_ en_3	dect_ en_5	dect_ no_3	dect_ no_5	res_ thresh
Tumor	59,86%	69,59%	71,16%	88,54%	91,44%	91,03%	93,41%	95,01%	94,11%	93,93%	0,00%
Liquor	72,36%	82,13%	79,65%	77,20%	85,27%	83,51%	90,90%	91,80%	92,02%	90,34%	0,00%
Oedem											
Knochen	82,71%	89,30%	88,23%	96,38%	96,55%	97,09%	98,93%	99,03%	98,96%	98,98%	97,58%
Fett	89,86%	90,28%	87,51%	79,60%	92,85%	92,00%	97,23%	97,82%	96,78%	97,93%	0,00%
graue Sub.	50,23%	66,33%	63,69%		22,87%	6,99%	75,12%	82,10%	75,76%	82,27%	0,00%
weiße Sub.	89,49%	91,57%	91,89%	81,61%	87,17%	86,22%	93,91%	95,80%	93,97%	96,28%	0,00%
Hintergrund	97,87%	99,04%	98,69%	99,84%	99,86%	99,89%	99,99%	100,00%	100,00%	100,00%	0,00%

Tabelle 3.6 Qualitätsmaß der Klassifikationsergebnisse

4 Generierung von Fuzzy Regeln

Dieses Kapitel befaßt sich mit der Umsetzung von Wissen in konkrete Fuzzy Regeln. Dabei muß anhand eines vorliegenden Musters von Zugehörigkeitswerten aus den Resultatbildern die Entscheidung getroffen werden, welcher Klasse das betreffende Pixel zugeordnet wird. Es werden zwei Modelle zur Lösung von unscharfen Entscheidungsproblemen vorgestellt, zum einen ein rein statistischer Ansatz, zum anderen ein Modell, bei dem der aus der Informationstheorie bekannte ID3 Algorithmus von Quinlan [QUIN95] verwendet wird, der auf Seite 46 *ff* beschrieben wird. Ein Fuzzy System sollte idealerweise das gleiche Entscheidungsverhalten wie ein Experte aufweisen. Der Experte ist dabei - im speziellen Fall der Gewebesegmentierung - der untersuchende Arzt, dessen Untersuchungsergebnisse in maschinenlesbarer Form als ROI's auf einem Datenträger abgelegt wurden. Die Untersuchungsergebnisse aus den ROI's liegen nur für einen Teil der vorhandenen Gewebedaten vor. Die Fuzzy Regeln werden als allgemeingültig und generalisierungsfähig betrachtet und sollen auf alle vorhandenen Gewebedaten angewendet werden.

In den Resultatbildern sind die Klassifikationsergebnisse der verschiedenen Algorithmen gespeichert. Die Resultatbilder enthalten vollständige, allerdings nicht notwendigerweise richtige Klassenzugehörigkeitsinformationen und dienen als Eingangsdaten zur Weiterverarbeitung mit Fuzzy Logik. Anhand der Fuzzy Sets und Fuzzy Regeln soll für jedes Pixel die Entscheidung getroffen werden, welcher Gewebeklasse es zuzuordnen ist.

4.1 Statistische Regelerzeugung

Dieser Abschnitt befaßt sich mit der Generierung von Fuzzy Regeln, basierend auf den statistischen Ergebnissen. Da die aus Kapitel 3 gewonnene statistische Information zur Erzeugung von Fuzzy Regeln verwendet wird, wird im weiteren Verlauf von *statistischer Regelerzeugung* gesprochen.

Die Erzeugung von Fuzzy Regeln erfordert eine sehr gute Kenntnis der vorhandenen Eingangsdaten. Aus dem in Kapitel 3 eingeführten Qualitätsmaß (**Tabelle 3.6**) können Aussagen zur Klassifikationsgüte gemacht werden. Aus diesen Aussagen lassen sich direkt Fuzzy Regeln ableiten.

Dabei werden die Klassensieger im Qualitätsmaß beispielsweise in Regeln der Form:

IF dect_en_5 = tumor THEN POSSIBILITÄT_TUMOR is mittel
IF dect_no_3 = liquor THEN POSSIBILITÄT_LIQUOR is mittel
IF dect_en_5 = knochen THEN POSSIBILITÄT_KNOCHEN is mittel
IF dect_no_5 = fett THEN POSSIBILITÄT_FETT is mittel
IF dect_no_5 = graue_substanz THEN POSSIBILITÄT_GRAUSUB is mittel
IF dect_no_5 = weiße_substanz THEN POSSIBILITÄT_WEISSUB is mittel
IF dect_no_5 = hintergrund THEN POSSIBILITÄT_HINTERGRUND is mittel

gefaßt.

Der prinzipielle Nachteil bei der Verwendung der Klassensieger unter den verschiedenen Segmentierungsalgorithmen ist, daß bei einer rein logischen Verknüpfung von Gewebedaten verschiedener Segmentierungsalgorithmen gleicher Klassen bzgl. der statistischen Häufigkeit eine Fehlermultiplikation stattfindet. Dieses Problem kann in der Fuzzy Logik durch Hinzufügen kompensierender Regeln verringert werden, macht sich aber dennoch bemerkbar.

Eine weitere Strategie bei der Generierung einer Regelbasis erhält man, indem man den in der Gesamtstatistik über alle Klassen besten Algorithmus in Form von Einfachstregeln direkt ins Fuzzy Resultatbild übernimmt und damit z.B. die Resultatterms *mittel* modifiziert.

IF dect_en_5 = tumor	*THEN KLASSE_TUMOR*	*is mittel*
IF dect_en_5 = liquor	*THEN KLASSE_LIQUOR*	*is mittel*
IF dect_en_5 = knochen	*THEN KLASSE_KNOCHEN*	*is mittel*
IF dect_en_5 = fett	*THEN KLASSE_FETT*	*is mittel*
IF dect_en_5 = graue_substanz	*THEN KLASSE_GRAUSUB*	*is mittel*
IF dect_en_5 = weiße_substanz	*THEN KLASSE_WEISSUB*	*is mittel*
IF dect_en_5 = hintergrund	*THEN KLASSE_HINTERGRUND*	*is mittel*

Würden keine weiteren Fuzzy Regeln hinzukommen, und würde auch nur das Resultatbild dieses einen Algorithmus als Eingangsvariable dienen, so wäre das Fuzzy Resultatbild identisch mit dem Eingangsresultatbild. Es kommen allerdings noch weitere, „bessere" Regeln hinzu. Dies ist auch der Grund, warum bisher nur die Terme *mittel* modifiziert wurden und nicht die Terme *hoch*. Da alle weiteren Regeln UND Verknüpfungen enthalten und diese erst einen hohen Gesamtzugehörigkeitswert zurückliefern, wenn beide Eingangsprämissen einer Regel hohe Zugehörigkeitswerte aufweisen, kann man erwarten, daß UND verknüpfte Regeln weniger Fehler produzieren und damit besser klassifizieren als Einfachstregeln. Aus diesem Grunde werden alle weiteren Regeln die Terme *hoch* modifizieren, da diese bei der Defuzzifizierung einen größeren Einfluß auf das Gesamtergebnis haben als die Terme *mittel*. Falls eine der UND veknüpften Regeln einen hohen Zugehörigkeitswert liefert, wirkt sie sich im Resultat Set dominant aus und kompensiert eventuelle Fehler der weniger einflußreichen Regeln der Terme *mittel*.

if dect_no_3 = tumor	*dect_no_5 = tumor*	*AND THEN KLASSE_TUMOR*	*hoch*
if dect_no_3 = liquor	*dect_no_5 = liquor*	*AND THEN KLASSE_LIQUOR*	*hoch*
if dect_no_3 = knochen	*dect_no_5 = knochen*	*AND THEN KLASSE_KNOCHEN*	*hoch*
if dect_no_3 = fett	*dect_no_5 = fett*	*AND THEN KLASSE_FETT*	*hoch*
if dect_no_3 = graue_substanz	*dect_no_5 =graue_substanz*	*AND THEN KLASSE_GRAUSUB*	*hoch*
if dect_no_3 = weiße_substanz	*dect_no_5 =weiße_substanz*	*AND THEN KLASSE_WEISSUB*	*hoch*
if dect_no_3 = hintergrund	*dect_no_5 = hintergrund*	*AND THEN KLASSE_HINTERGRUND*	*hoch*

Die so gewonnenen Fuzzy Resultatbilder sind - wie ein ROI Vergleich zeigt - statistisch tatsächlich besser als jedes der verwendeten Eingangsresultatbilder. Allerdings bleibt der Einfluß des den Terms *mittel* zugrundeliegenden Resultatbildes unverkennbar. Charakteristische Fehler des dominierenden Resultatbildes bleiben zudem erhalten. Der Hauptnachteil der statistischen Regelerzeugung besteht allerdings darin, daß zwar gute Fuzzy Resultatbilder erzeugt werden können, andererseits kann man sich niemals sicher sein, ob das so gewonnene Resultatbild schon *optimal* ist, oder ob die Regelbasis durch empirische Untersuchung noch weiter verbessert werden kann. In der Praxis zeigt sich, daß ein auf rein statistischen Erwägungen basierendes *intuitives* finden von Regeln sehr zeitaufwendig sein kann, eine Aufgabe, die mit Hilfe eines Computers sicherlich schneller und besser gelöst werden könnte.

4.2 Automatische Regelerzeugung mit Quinlan's ID3 Algorithmus

Die Qualität der Resultatbilder ist maßgeblich von den Fuzzy Regeln abhängig. Dazu ist eine sehr genaue Kenntnis der vorhandenen Eingangsdaten notwendig, die über eine rein statistische Auswertung hinausgeht.

Besonders interessant ist die Frage, ob es eventuelle Muster in den Eingangsvariablen gibt, bei denen eine bestimmte Klassenzugehörigkeit in der Resultatvariablen mit sehr hoher Wahrscheinlichkeit zu erwarten ist. Der Mensch ist ohne weiteres in der Lage, sofern die Datenmengen klein sind, Zusammenhänge oder Muster zu erkennen und sie als Regeln zu formulieren. In Anbetracht der beträchtlichen Datenmengen (ca. 2,5 MB ROI Daten) soll diese Erkennung automatisiert werden.

Zur Extraktion des Expertenwissens aus den ROI Daten bietet sich der aus der Informationstheorie bekannte ID3-Algorithmus von Quinlan an. Dabei baut das ID3 Verfahren einen Entscheidungsbaum auf, in dessen Knoten sich jeweils eine bestimmte Teilmenge der gegebenen Beispiele befindet. Ein Knoten heißt dabei homogen, wenn alle darin vorkommenden Attribute die gleiche Expertenmeinung aufweisen.

Beispiel 4.1 - Entscheidungsfindung mit dem ID3 Algorithmus

Als Beispiel sollen die Trainingsdaten für Kauf eines Autos dienen, mit den Attributen Leistung, Ausstattung und Preis.

Attribut	mögliche Ausprägungen
Leistung	hoch, mittel, niedrig
Ausstattung	gut, schlecht
Preis	billig, teuer

Tabelle 4.1 - Attributausprägungen

Die Trainingsdaten sehen folgendermaßen aus:

PKW	Leistung	Ausstattung	Preis	Entscheidung
Auto 1	**hoch**	**gut**	**billig**	**ja**
Auto 2	**hoch**	**schlecht**	**teuer**	**nein**
Auto 3	**niedrig**	**schlecht**	**billig**	**ja**
Auto 4	**mittel**	**gut**	**billig**	**nein**
Auto 5	**mittel**	**schlecht**	**billig**	**nein**
Auto 6	**hoch**	**schlecht**	**billig**	**ja**
Auto 7	**mittel**	**schlecht**	**teuer**	**nein**
Auto 8	**hoch**	**gut**	**teuer**	**nein**

Tabelle 4.2 - Entscheidungstabelle für Autokauf

Der Interessent hat für acht Autos eine Entscheidung getroffen, die als Expertenmeinung bezeichnet wird. Mit Hilfe des ID3 Algorithmus sollen nun die Regeln, die bei dem Experten zur Kaufentscheidung geführt haben, festgestellt werden.

Dies läßt sich durch folgenden Algorithmus erreichen:

- Nehme ein beliebiges Blatt des Baumes und bezeichne es als das aktuelle Blatt. Bei dem beliebigen Blatt kann es sich auch um die Wurzel des Baumes handeln, die sämtliche Trainingsbeispiele und Attribute enthält, der Algorithmus.

- Überprüfe ob das betrachtete Blatt homogen ist. Ein Blatt ist dann homogen, wenn es nur noch Testbeispiele enthält, die bei der Entscheidung (Tabelle 4.2) die selbe Ausprägung aufweisen. Falls das aktuelle Blatt inhomogen ist, bilde die Menge der Attribute abzüglich derer, die beim Durchlaufen des Baumes von der Wurzel bis zum aktuellen Blatt vorher schon verwendet wurden. Wähle das Attribut mit dem geringsten Informationsgehalt nach folgendem Kriterium aus:

$$I_{(K)} = -\sum_{i=1}^{m} p_i * \log_m(p_i)$$

m: Anzahl der Ausprägungen des betrachteten Attributs
p_i : Wahrscheinlichkeit p einer Ausprägung i eines Attributs
p_i = (Anzahl$_i$ / Gesamtzahl der Beispiele)
$I_{(K)}$: Informationsgehalt eines Knotens

Entropie nach Shannon [SHANNON]

Für Attribute mit den beiden Ausprägungen (ja/nein) würde sich der Informationsgehalt eines Knotens wie folgt berechnen:

$I_{(K)} = - p * \log_2(p) - q * \log_2(q)$,

wobei *p* dem relativen Anteil mit der Entscheidung *ja* und *q* dem relativen Anteil mit der Entscheidung *nein* entspricht. Haben alle Attribute jeweils nur zwei Ausprägungen, so entsteht ein Binärbaum. Die maximale Höhe des Baumes entspricht der Anzahl der verschiedenen Attribute.

- Führe am aktuellen Blatt eine Aufspaltung bezüglich des ausgewählten Attributs durch. Dies geschieht dadurch, daß für jede Ausprägung des Attributs ein neues Blatt erzeugt wird.

- Führe das Verfahren so lange durch, bis alle Blätter des Baumes homogen sind oder die Menge der Attribute, welche die maximale Höhe des Baumes begrenzen, erschöpft ist.

Ein mit diesem Verfahren erzeugter Entscheidungsbaum enthält das gesuchte Expertenwissen, welches in Regeln dargestellt werden kann. Ein Entscheidungsbaum zur Extraktion des Expertenwissens beim Autokoauf
(Beispiel 4.1) und den Trainingsdaten (Tabelle 4.2) kann folgendermaßen aussehen:

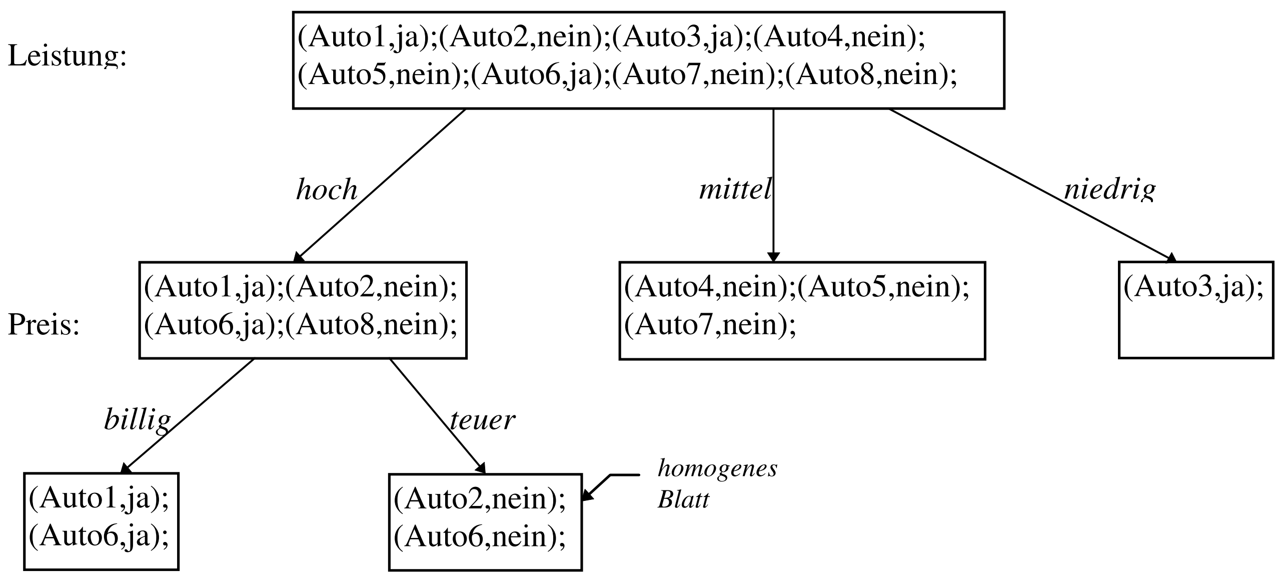


Abbildung 4.1

Im vorliegenden Entscheidungsbaum wurde das Attribut *Ausstattung* nicht mehr benötigt, weil alle Blätter des Baumes Baum schon nach der zweiten Hierarchieebene vollständig homogen sind. Anstelle von Preis, hätte in der zweiten Ebene auch Ausstattung stehen können, da beide Attribute nach Shannons Entropiekriterium den selben Informationsgehalt haben. Die Reihenfolge der Attribute nach dem Entropiekriterium erzeugt einen kleinstmöglichen Baum mit möglichst vielen Testgebieten in den homogenen Blättern. Eine andere Reihenfolge der Attribute kann ebenfalls zur Informationstrennung verwendet werden, es werden dann unter Umständen mehr kleinere homogene Blätter gebildet. Die das Expertenwissen enthaltenden Regeln können direkt aus der topologischen Struktur des Baumes abgelesen werden. Dabei wird jeder Weg von der Wurzel bis zu einem Blatt in eine Regel transformiert. Jede Ebene repräsentiert ein Attribut (linguistische Variable), nach dessen Ausprägungen die Testdaten sortiert und den nachfolgenden Blättern entsprechend zugeordnet werden. Wurden sämtliche Attribute verwendet und der Baum enthält immer noch inhomogene Blätter, so ist eine vollständige Informationstrennung mit den zugrundeliegenden Attributen nicht möglich

Aus dem Baum in Abbildung 4.1 können folgende Regeln ermittelt werden:

1: if Leistung = hoch	*AND Preis = billig*	*THEN Entscheidung = ja*
2: if Leistung = hoch	*AND Preis = teuer*	*THEN Entscheidung = nein*
3: if Leistung = mittel		*THEN Entscheidung = nein*
4: if Leistung = niedrig		*THEN Entscheidung = ja*

Das Verfahren läßt sich in ähnlicher Weise auf das gestellte Segmentierungsproblem anwenden.
Dabei entsprechen die jeweiligen Attribute den verschiedenen linguistischen Variablen, deren Ausprägungen die Bezeichnungen der einzelnen Fuzzy Sets (Terms) annehmen können. Ein Attribut enthält als Beispielmenge die Ergebnisse eines Segmentierungsalgorithmus, der die Ausprägungen *nicht klassifiziert, Tumor, Liquor, Oedem, Knochen, Fett, graue Substanz, weiße Substanz und Hintergrund* haben kann. Die theoretische maximale Anzahl der Knoten errechnet sich aus der Anzahl der derzeit verfügbaren Segmentierungsalgorithmen (11) und der Anzahl Ausprägungen (8) $8^{11} \approx 8{,}5 * 10^9$, wobei Algorithmus Nr. 10 (res_thresh) nur zur Erkennung von Knochen geeignet ist und damit auch die theoretisch maximale Anzahl der Knoten geringer wird.
Der Algorithmus kann unter folgenden Voraussetzungen im Zusammenhang mit Fuzzy Logik zur Erzeugung von Regeln eingesetzt werden:

- Die Attribute und die entsprechende Expertenmeinung bestehen immer aus Mengen mehrerer fest vorgegebener, aufzählbarer Zustände wie *ja, nein* oder *rot, grün, gelb* usw.

- Der ID3 Algorithmus geht von konsistenten, widerspruchsfreien Daten aus. Die von den Segmentierungsalgorithmen erzeugte „Expertenmeinung“ spiegelt das reale Ergebnis jedoch nicht mit absoluter Sicherheit, sondern nur mit einer bestimmten Wahrscheinlichkeit wieder. Der ID3 Algorithmus muß deshalb dahingehend erweitert werden, daß ein Blatt schon dann als homogen betrachtet wird, wenn der relative Anteil der gewünschten Klassenausprägung eine vorher festgelegte Grenze nicht unterschreitet. Ansonsten würde der Algorithmus in den seltensten Fällen homogene Blätter produzieren. Das Zulassen von Fehlertoleranzen impliziert natürlich auch, daß die erzeugten Fuzzy Regeln bei der Generalisierung ebenfalls gewisse Fehler produzieren können.

- Der Baum und die damit verbundenen Terminierungszeiten des Algorithmus können sehr groß werden. Es empfiehlt sich deshalb, bei großen Datenmengen mit vielen Ausprägungen die Anzahl der Attribute zu begrenzen, dadurch wird gleichzeitig die Höhe des Baumes begrenzt und somit die exponentiell steigende Ausführungszeit reduziert.

Der modifizierte ID3 Algorithmus wurde implementiert und erzeugt bei einer Toleranz von 1%
für homogene Bereiche folgende Regeln:

```
if DECT_NO_5 back THEN back big
if RES_THRESH knochen          DECT_NO_5 knochen             AND THEN
knochen       big
if DECT_NO_5 knochen           DECT_NO_5 back  AND THEN back  big
if DECT_M2_3 fett              DECT_NO_3 fett  AND THEN fett  big
if DECT_M2_3 grau              DECT_NO_3 fett  AND THEN fett  big
if DECT_M2_3 weiss             DECT_EN_3 tumor               AND THEN
tumor         big
if DECT_C4_3 tumor             DECT_EN_3 grau  AND DECT_NO_3 grau
DECT_EN_5 grauAND AND
     THEN grau    big
if DECT_C4_3 grau              DECT_EN_3 grau  AND DECT_NO_3 grau
DECT_EN_5 grauAND AND
     THEN grau    big
if DECT_C6_3 weiss             DECT_EN_3 grau  AND DECT_NO_3 weiss
DECT_EN_5 weiss   AND AND
     THEN weiss    big
if DECT_C4_3 grau              DECT_EN_3 weissAND DECT_NO_3 grau
DECT_EN_5 grauAND AND          DECT_NO_5      grau AND THEN grau big
if DECT_C6_3 weiss             DECT_EN_3 weiss               AND
DECT_NO_3 weiss DECT_EN_5 weiss   AND AND        DECT_NO_5       weiss
AND THEN weiss big
```

Die Regeln werden in polnischer Notation erzeugt und können direkt in die Regelbasis des Verknüpfungstools übernommen werden.
Homogene Bereiche, die aus 30 oder weniger Pixeln bestehen, wurden nicht berücksichtigt, da sich bei kleinen homogenen Clustern die Frage stellt, in wieweit die daraus resultierenden Regeln noch allgemeingültig zur Gewebesegmentierung verwendet werden können.
In der Praxis hat sich eine Beschränkung der bei der Regelgenerierung verwendeten Attribute auf fünf als sinnvoll erwiesen. Die Ausführungszeit beträgt dabei ca. 25 Minuten. Tests mit einer größeren Anzahl von Attributen haben ergeben, daß zwar wesentlich mehr homogene Blätter erzeugt werden, die meisten dieser Blätter enthielten allerdings nur ein oder zwei Testbeispiele und wurden aus Gründen der eingeschränkten Genealisierungsfähigkeit nicht weiter beachtet.

Bei der automatischen Regelgenerierung wurden nicht für alle Klassen Regeln erzeugt. Es konnten beispielsweise keine Regeln zur Erkennung der Klasse Liquor erzeugt werden, weil das Homogenitätskriterium bis zur fünften Ebene nicht erfüllt war. Für die Klasse Tumor existiert zwar eine Regel, diese umfaßt aus den ROI Daten jedoch nur einen geringen Teil der vorhandenen Tumorpixel (273 von 2284) - hier muß die Regelbasis weiterhin nach statistischen Methoden ergänzt werden.

Da die Fuzzy Regeln ausschließlich den UND Operator zum erzeugen homogener Bereiche verwenden ist es offensichtlich, daß es Pixel in den Schichtbildern geben wird, die nicht „erkannt“ werden. Solche Pixel werden nicht klassifiziert und haben den Zugehörigkeitswert Null. Nicht klassifizierte Pixel werden der Sonderklasse „*STAT*“ zugeordnet, im Fuzzy

Resultatbild werden sie schwarz dargestellt. Generell enthalten Regeln, die immer die selben Terms ansprechen einen Großteil der Testgebiete, da die Information durch die Segmentierunsalgorithmen zum Teil schon getrennt wurde. (Beispiel: `if DECT_C4_3 fett DECT_NO_3 fett AND DECT_EN_3 fett DECT_EN_5 fett AND AND THEN fett mid`)
Bei der Auswertung wurden die verschiedenen UND Operatoren (Einstein Product, arithmetic Product, bounded Difference, Hamacher Product, Min-Operator) getestet und die Ergebnisse mit den ROI Daten verglichen. Die Erkennungsraten (Qualitätsmaß) waren allerdings auf zwei Nachkommastellen genau identisch. Im vorliegenden Fall scheint die Wahl des die UND Verknüpfung realisierenden Operators von untergeordneter Bedeutung zu sein.

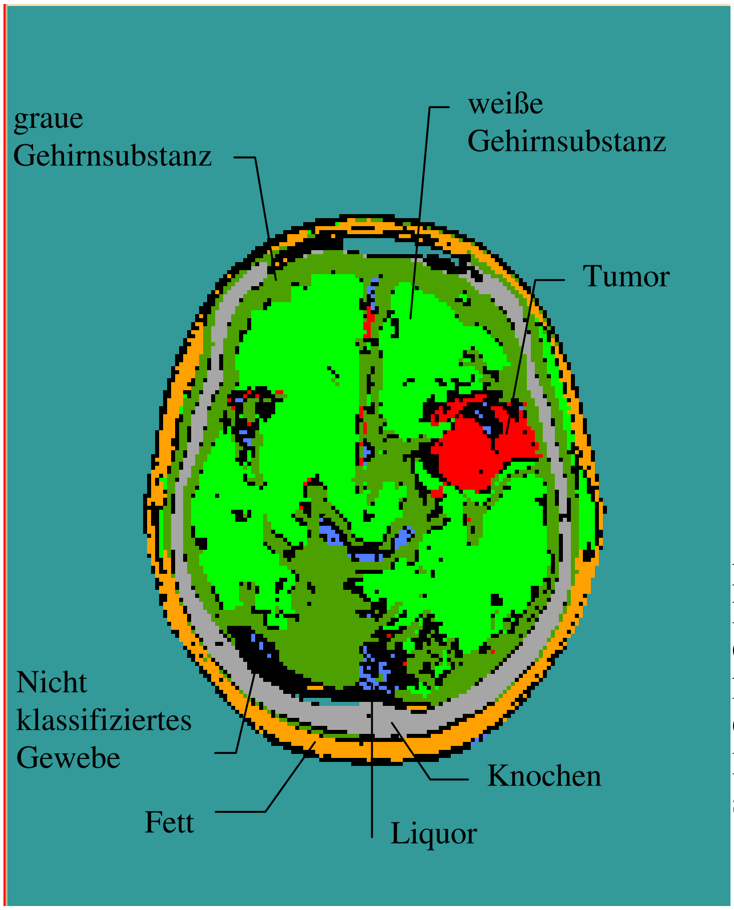
Um die Zahl der insgesamt nicht klassifizierten Pixel zu verringern, wurden das Verfahren in einem zweiten Durchgang auf alle bisher nicht klassifizierten Pixel noch einmal angewendet.
Die Toleranzvorgabe für homogene Bereiche wurde auf 2,5 % erhöht.
Die Regeln sehen folgendermaßen aus:

```
if DECT_C6_3     tumor      DECT_EN_3   tumor AND THEN tumor mid
if DECT_C4_3     weiss      DECT_NO_5   liquor      AND THEN liquor mid
if DECT_C4_3     weiss      DECT_NO_3   fett  AND THEN fett mid
if DECT_C4_3     liquor     DECT_NO_5   weiss AND DECT_EN_3 liquor AND
THEN liquor mid
if DECT_C6_3     fett       DECT_EN_3   tumor AND DECT_NO_5 tumor AND THEN
tumor mid
if DECT_C4_3     fett       DECT_NO_3   weiss AND DECT_EN_3 fett AND THEN
fett mid
if DECT_C4_3     weiss      DECT_NO_3   weiss AND DECT_EN_3 fett AND THEN
fett mid
if DECT_NO_3     liquor     DECT_C6_3   weiss AND DECT_EN_3 grau DECT_EN_5
grau AND AND
     THEN grau mid
if DECT_C4_3     fett       DECT_NO_3   fett  AND DECT_EN_3 fett DECT_EN_5
fett AND AND
     THEN fett mid
if DECT_C4_3     weiss      DECT_NO_5   weiss AND DECT_EN_3 liquor
DECT_EN_5 liquor AND AND    THEN liquor mid
if DECT_NO_3     weiss      DECT_C6_3   grau  AND DECT_EN_3 weiss
DECT_EN_5 grau AND AND      DECT_NO_5   grau  AND THEN grau mid
```

Um die Zahl der nicht klassifizierten Pixel weiter zu reduzieren wurden weitere Regeln „von Hand“ hinzugefügt. Dazu wurden größere Cluster mit wenig Fehlern zusätzlich in die Regelbasis mit aufgenommen.

```
if DECT_M1_3 grau            DECT_M2_3 grau    AND DECT_M3_3 grau   AND
THEN grau      min
if DECT_M2_3 tumor           DECT_M3_3 tumor   AND DECT_EN_3 tumor  AND
THEN tumor     min
if DECT_EN_5 grau            DECT_NO_5 grau    AND THEN grau  min
if DECT_EN_5 weiss           DECT_NO_5 weiss   AND THEN weiss min
```

Mit den Fuzzy Regeln wurde folgendes Fuzzy Resultatbild erzeugt:



Es fällt auf, daß hauptsächlich in den Randbereichen zwischen den Gewebeklassen viele nicht-klassifizierte Pixel liegen. Offensichtlich gestaltet sich hier die Informationstrennung besonders schwierig.

Abbildung 4.2 Fuzzy Resultatbild

Diskussion des Resultabildes

Bei dem Resultatbild handelt es sich um Schicht Nr. 10 der Sequenz da55. Das Fuzzy Resultatbild weist trotz guter statistischer Übereinstimmung mit den Testgebieten noch unerwünschte Fehler auf:

- Die zusammenhängenden Gebiete einer Klasse werden durch einzelne Pixel anderer Klassen gestört. Diese können allerdings im Rahmen eines Postprocessings beseitigt werden.

- Auch die an den Texturgrenzen häufig auftretenden nichtklassifizierten Pixel können durch das Postprocessing beseitigt werden. Im hinteren Schädelbereich existiert allerdings auch ein Gebiet nichtklassifizierter Pixel, das sich auch bei einer Nachverarbeitung nicht entfernen läßt.

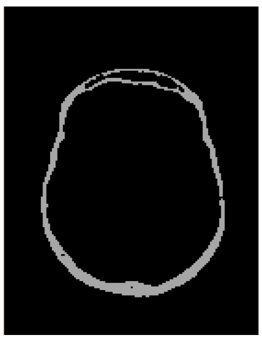
- Die den Schädel begrenzenden Gewebeteile Knochen und Fett sind unterbrochen. Wünschenswert wäre es, wenn sie zusammenhängend wären.

- Im Bereich der Ohren wird zwischen Knochen und Fett Gewebe den Klassen weiße und graue Gehirnsubstanz zugeordnet. Tatsächlich handelt sich hierbei um eine Gewebeart, die keiner der bisher betrachteten Klassen zugeordnet werden kann. In den Resultatsequenzen weisen diese Gebiete allerdings eine gewisse Übereinstimmung auf, so daß sich der Effekt auch im Fuzzy Resultatbild bemerkbar macht. Eine vollständige Beseitigung ist nicht möglich, jedoch läßt sich der Bereich verkleinern, indem man für Fett und Knochen *tolerantere* Regeln zuläßt.

4.3 Anpassung und Optimierung der Regelbasis

Im Folgenden wurde die Regelbasis nach zusätzlichen, visuellen Gesichtspunkten abgeändert. Die Regeln zur Erkennung von Knochen wurden ersetzt durch die Regel:

- *if RES_THRESH knochen then knochen big*



Dabei wird zur Segmentierung von Knochen einzig und allein der Algorithmus *res_thresh* verwendet, der eigens für die Erkennung von Knochen entwickelt wurde. Im Qualitätsmaß schneidet *res_thresh* schlechter ab als beispielsweise *dect_no_5, dect_no_3 oder dect_en_5.* Allerdings liefert dieser Algorithmus bis auf wenige Ausnahmen ein vollständig zusammenhängendes anatomisches Bild des Schädelknochens.
Das als Knochen klassifizierte Gewebe geht zu 100 % in das Fuzzy Resultatbild ein.

Abbildung 4.3 Schichtbild eines Schädels

Algorithmus *dect_no_5* liefert die statistisch besten Resultate bei der Erkennung von Fettgewebe. Die die Terms *big* modifizierenden Regeln zur Erkennung von Fettgewebe wurden dabei ersetzt durch die Regel:

- *if DECT_NO_5 fett then fett big*

Auch diese Regel geht zu einem hohen Anteil in das Fuzzy Resultatbild ein. Es existieren bei den Termen *mid* noch zwei weitere Fuzzy Regeln, die das Resultatbild ebenfalls beeinflussen. Auch hier wurde auf Kosten der Klassifikationsqualität dem Fettgewebe einen größeren Einfluß ins Gesamtbild eingeräumt. Der Preis hierfür ist, daß einzelne Pixel - hauptsächlich graue und weiße Gehirnsubstanz - fälschlicherweise zu Fettgewebe klassifiziert werden. Abbildung 4.4 zeigt ein Fuzzy Resultatbild, das aus einer Regelbasis erzeugt wurde, deren statistische Trefferquote aufgrund visueller Erwägungen geringfügig vermindert wurde. Knochen und Fettgewebe sind zusammenhängend und umgeben das Schädelinnere. Die Gehirnregionen sind augenscheinlich von kleinen als Fettgewebe klassifizierten Bereichen durchsetzt, was anatomisch gesehen äußerst selten vorkommt.

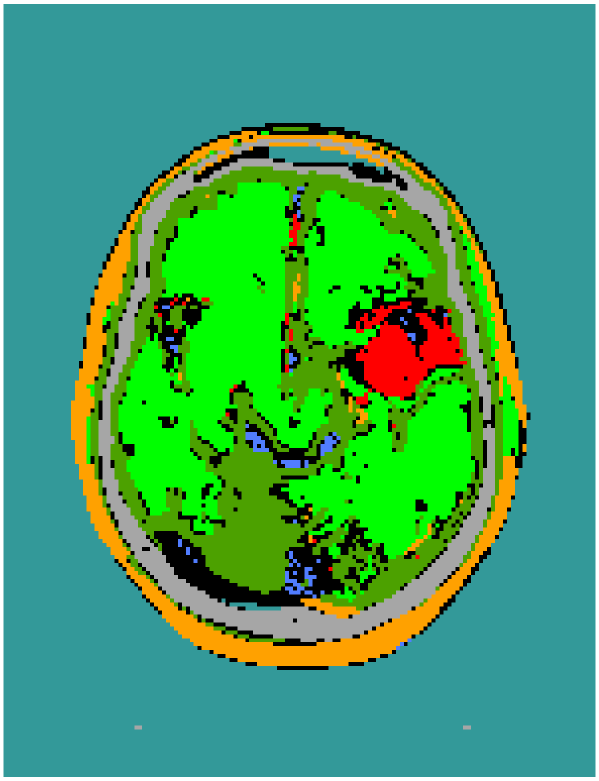


Abbildung 4.4 - Fuzzy Resultatbild mit angepaßter Regelbasis

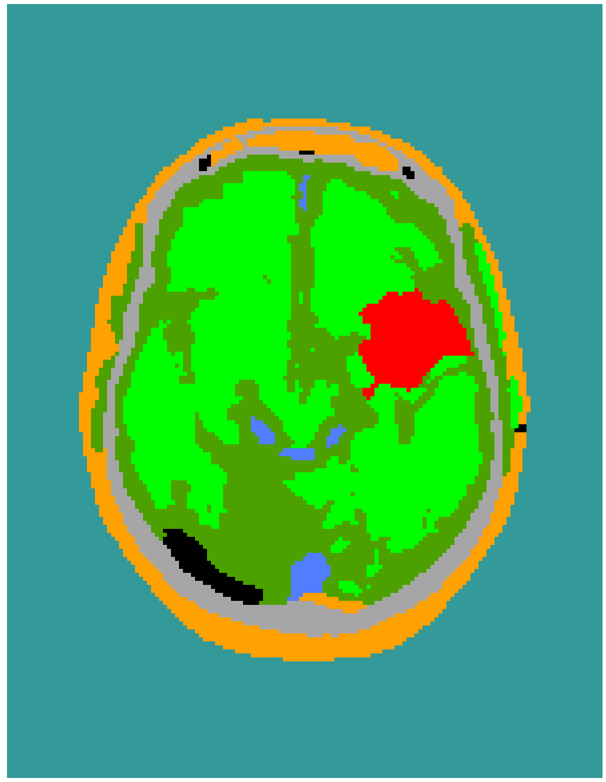


Abbildung 4.5 - Durch Postprocessing nachbearbeitetes Bild

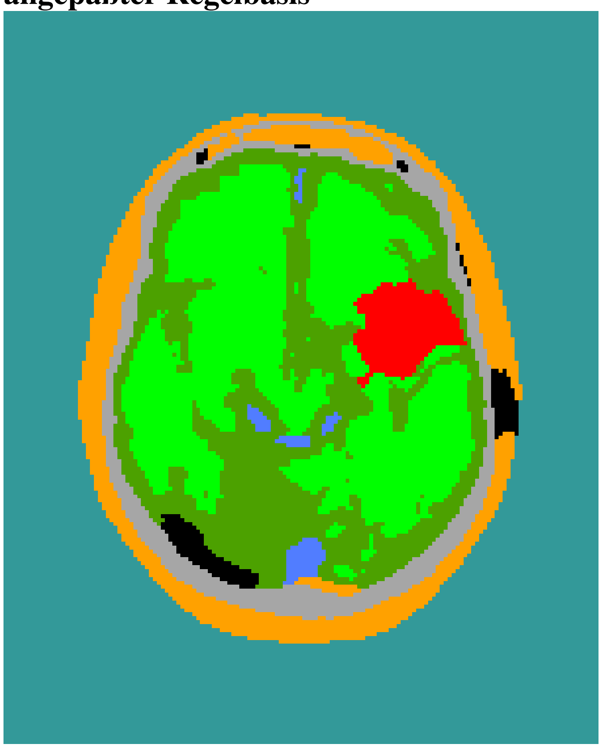


Abbildung 4.6 - Zusätzlicher Nachbearbeitungsschritt im Postprocessing.

Die Fuzzy Resultatbilder wurden nicht maßstabsgetreu abgebildet

4.4 Postprocessing

Zur Nachbearbeitung der Fuzzy Sequenzen wurde auf ein schon bestehendes Tool PPP (Post Processor Parser) zurückgegriffen. Es handelt sich hierbei um eine Methodik morphologische Operationen direkt auf farbkodierte Bilder auszuführen, wobei semantische Zusammenhänge wie z.B. die Anatomie berücksichtigt werden. Einzelne fehlerhaft klassifizierte Rasterpunkte können durch eine Nachbearbeitung des Bildes (Postprocessing) korrigiert werden (Abbildung 4.5,4.6) , so daß sich die Fehler im endgültigen Bild kaum mehr bemerkbar machen. Die im Ohrenbereich falsch klassifizierten Weichteilregionen zwischen Knochen und Fettgewebe spielen für medizinische Befunde eine untergeordnete Rolle. Für die Fuzzy Resultatsequenzen wurde durch Einfügen eines zusätzlichen Nachbearbeitungsschrittes das falsch klassifizierte Gewebe im Ohrenbereich kosmetisch korrigiert. Anstelle einer kosmetischen Korrektur des Bildes wäre es allerdings sinnvoller, generell eine neue Gewebeklasse einzuführen und die Klassifikationsfähigkeit der verschiedenen Segmentierungsalgorithmen zu erweitern - dies erfordert allerdings größeren Entwicklungsaufwand.

Die wesentlichen Operationen beim Postprocessing sind sensitive Dilatationen und Erosionen, die klassenweise und texturabhängig auf das zu bearbeitende Resultatbild angewendet werden können. „Sensitive“ Dilatation bedeutet, daß sich eine Farbe ausdehnen kann, sie aber dabei auf keinen Fall bestimmte vorgegebene geschützte Farben überschreiben darf. Im ersten Bearbeitungsschritt werden Hintergrundpixel im Schädelbereich entfernt, anschließend werden graue und weiße Substanz außerhalb des Knochens und im Abstand von fünf Pixeln vom Hintergund gelöscht, dadurch werden die Fehlklassifikationen im Ohrenbereich entfernt Ein weiterer Bearbeitungsschritt entfernt einzelne Fehlklassifikationen durch eine Umgebungsanalyse. Durch Dilatation und Erosion werden größere zusammenhängende Bereiche erzeugt, anschließend wird das Bild geglättet und kleinere undefinierte Bereiche mit Umgebungsfarben aufgefüllt.

5 Segmentierung mittels Fuzzy Logik aus Originalsequenzen

Dieses Kapitel befaßt sich mit der Bildsegmentierung aus den Originaldaten. Bei der Gehirntomographie wird eine Sequenz von Schichtbildern erzeugt, die übereinandergelegt ein dreidimensionales Abbild des betrachteten Kopfes ergeben. Um möglichst viel Informationen auf die Bilder zu übertragen, werden pro Schicht mehrere radiologische Aufnahmen aus jeweils unterschiedlichen Frequenzspektren erzeugt. Da verschiedene Gewebearten die Röntgenstrahlen unterschiedlich absorbieren, trägt jede Aufnahme andere Schichtinformationen, die die Merkmalsextraktion bei der Auswertung wesentlich vereinfachen, sofern sichergestellt ist, daß die Bilddaten geometrisch exakt übereinstimmen.

5.1 Untersuchung des vorhandenen Datenmaterials

Bei dem vorliegenden Datenmaterial besteht jede Schicht aus drei Einzelbildern, mit einer Grauwertauflösung von 12 Bit, d.h. es gibt 4096 unterschiedliche Grauwerte. Die Abbildungen zeigen exemplarisch eine CT, PW und T2 Wichtung von Schicht Nr. 10 der Sequenz da60.

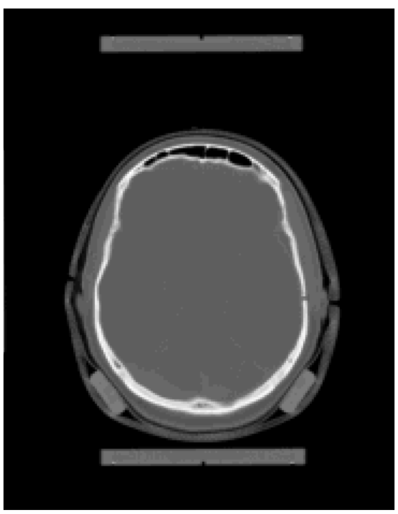


Abbildung 5.1 CT Bild

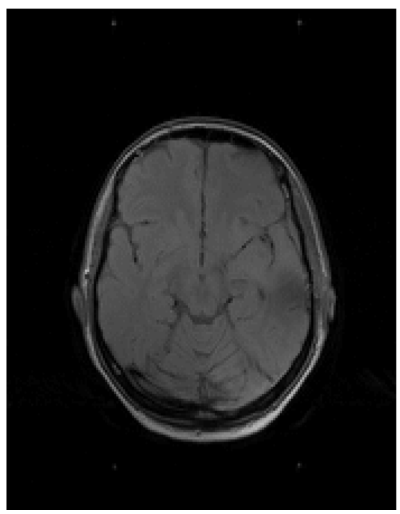


Abbildung 5.2 PW Bild

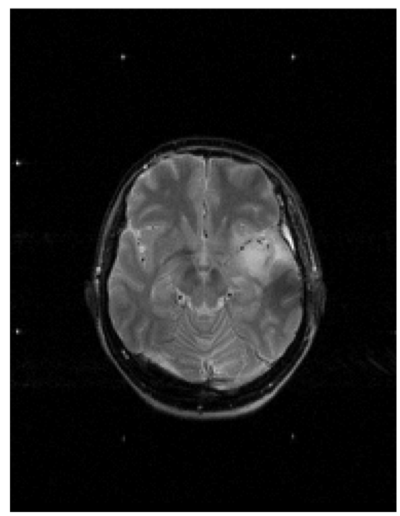


Abbildung 5.3 T2 Bild

Die Verknüpfung und Segmentierung der Bilder soll, wie in Kapitel 4, ebenfalls durch Fuzzy Logik erfolgen. Da es sich hierbei nicht mehr um algorithmisch vorklassifizierte Daten handelt, existiert kein klassenbezogenes Qualitätsmaß, so daß eine Fuzzifizierung mittels Singeltons ausscheidet. Vielmehr bietet es sich hier an, für jedes Grauwertbild ein Histogramm zu erstellen, das als Einganszugehörigkeitsfunktion für Fuzzy Verknüpfungen dienen kann.
In den Abbildungen wurden für die Testgebiete in den ROI's Histogramme für sämtliche Gewebeklassen erstellt. Die Histogramme wurden aus dem gesamten vorhanden Datenmaterial erstellt, d.h. es wurden sämtliche verfügbaren Gehirnsequenzen klassenweise zu Histogrammen verschmolzen. Da die Gehirnsequenzen unter nahezu idealen

Aufnahmebedingungen erstellt wurden, sind Kontrast und Intensität der Bilder gleich, so daß eine direkte Überlagerung möglich ist und keine Wischeffekte auftreten, die die Histogramme unbrauchbar machen würden. Das Vorhandensein der Testdaten bietet hier einen enormen Vorteil, da so für jede Klasse eine vollständige Verteilungsfunktion - unabhängig von störenden Nebeneffekten anderer Klassen - angegeben werden kann.

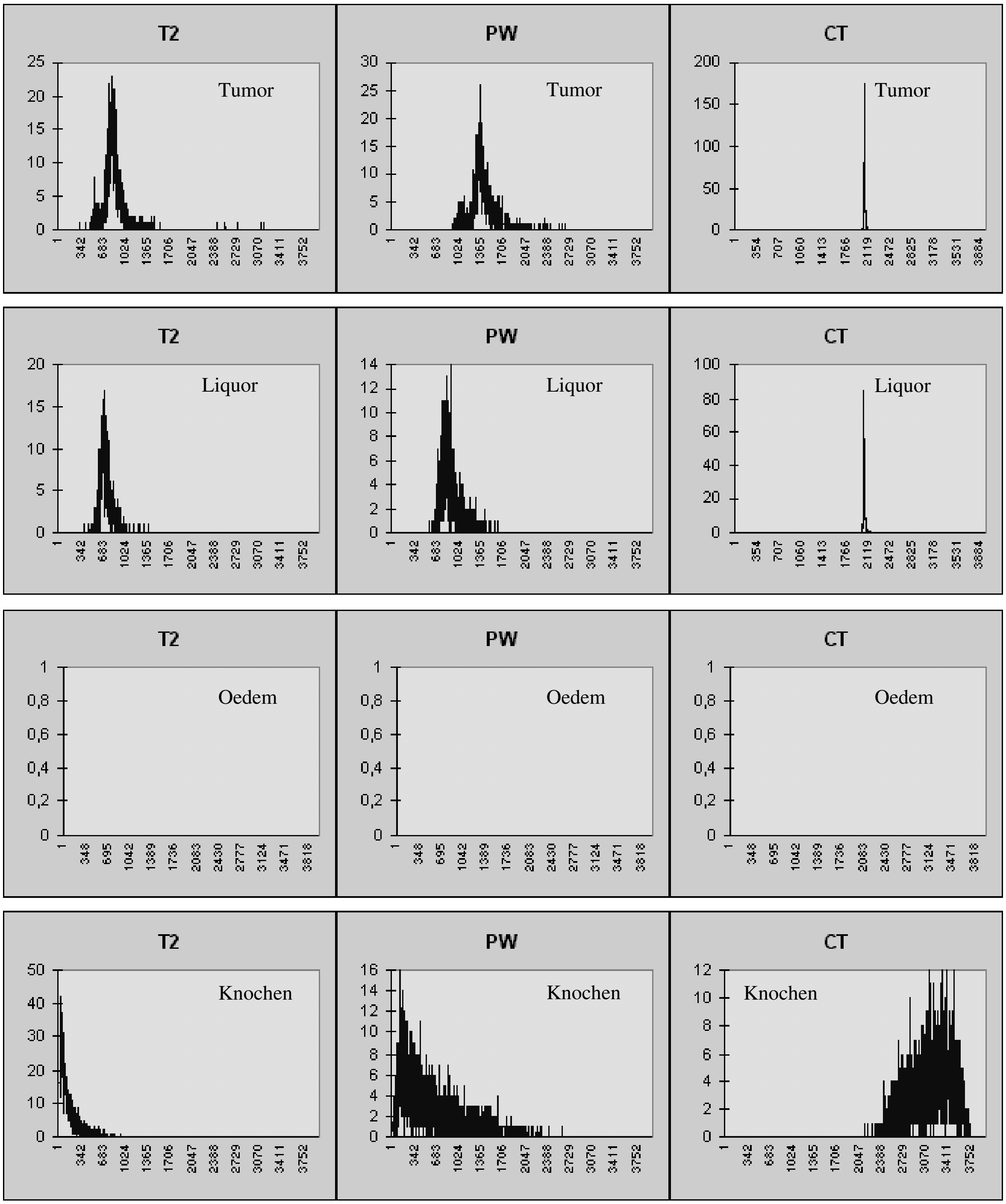


Abbildung 5.4 Klassenspezifische Histogramme aus den Testgebieten

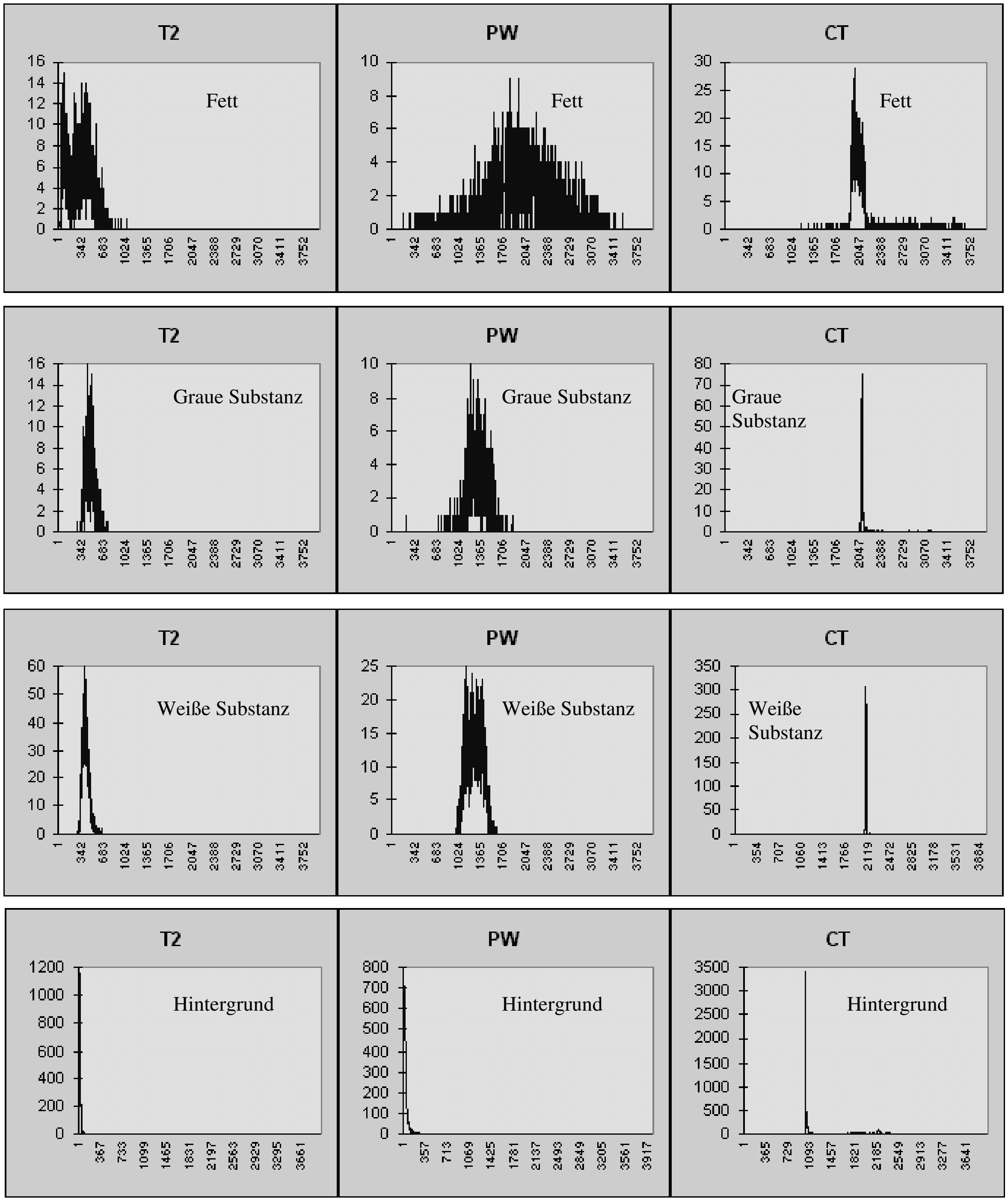


Abbildung 5.5 Klassenspezifische Histogramme aus den Testgebieten (Fortsetzung)

Da sich die Histogramme aus verschiedenen Gehirnsequenzen zusammensetzen, sind sie generalisierungsfähig, d.h. eine weitere, unter ähnlichen Bedingungen erzeugte Gehirnsequenz würde vergleichbare Ergebnisse liefern. Die Histogramme enthalten somit die zur Merkmalsextraktion notwendige Information, dabei sind - wie aus den Diagrammen ersichtlich - keine Testdaten für die Klasse *Oedem* vorhanden. Grobe Aussagen zu den Eigenschaften der

Klassen kann man schon durch reines Betrachten machen, so unterscheidet sich beispielsweise Knochen allein im CT Diagramm von allen anderen Gewebesorten, was auch im CT Grauwertbild (Abbildung 5.1) deutlich sichtbar ist. Natürlich lassen sich noch viel mehr Aussagen zu den Histogrammdaten machen, die Interpretation und Auswertung der Daten sollen jedoch automatisch durch das Fuzzy Tool von statten gehen.

Abbildung 5.6 zeigt exemplarisch ein detaillierteres Histogramm der Gewebeklasse *Knochen* im T2 Spektrum. Dabei wurde der grobe Verlauf der Kurve durch mehrere sich an den Enden berührende Linien nachgebildet. Jede Punktkoordinate einer Unstetigkeitsstelle kann dabei als Teil eines die Zugehörigkeitsfunktion definierenden Polygonzuges interpretiert werden, der so skaliert wurde, daß der maximale Ordinatenwert gerade eins ist. Diese Zugehörigkeitsfunktionen wurden für alle Histogramme erstellt und dienen der Fuzzifizierung der Eingangsdaten, die im weiteren Verlauf erklärt wird.

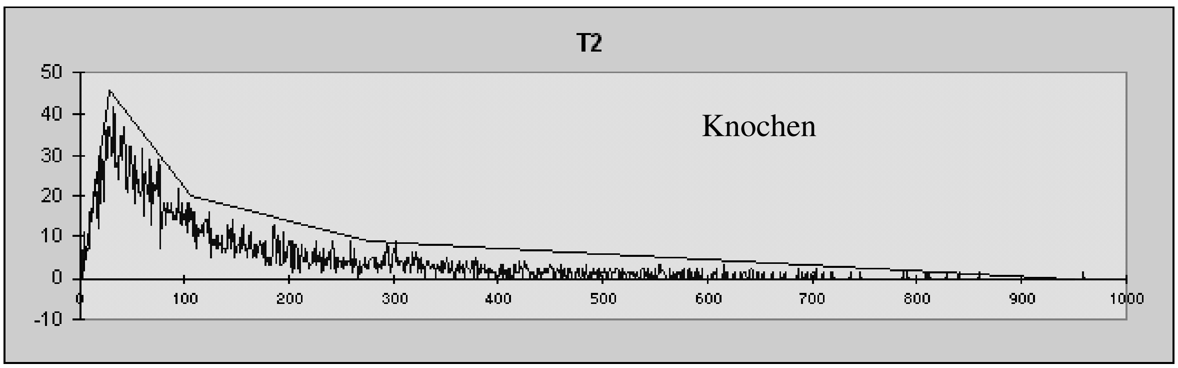


Abbildung 5.6 Angleichung einer Zugehörigkeitsfunktion

5.2 Fuzzifizierungsstategien zur Segmentierung aus Originaldaten

Jede zu untersuchende Sequenz liefert für jedes Pixel drei Grauwerte von den verschiedenen Eingangsbildern (CT, PW, T2), welche die drei scharfen Eingangswerte für das Fuzzy Verknüpfungstool bilden. Entsprechend der drei scharfen Eingangswerte existieren auch drei linguistische Variablen mit jeweils die acht verschiedenen Gewebeklassen repräsentierenden Zugehörigkeitsfunktionen, die dem jeweiligen Verlauf der Histogramme angeglichen wurden. Beim Fuzzifizieren werden dann für sämtliche Gewebeklassen die Zugehörigkeitswerte berechnet. Ein Regelsatz überträgt die Zugehörigkeitswerte unter Berücksichtigung der verschiedenen Fuzzy Operatoren auf acht Resultat Fuzzy Sets, mit denen die einzelnen Gewebeklassen assoziiert werden und die in Form und Aufbau identisch mit den bereits in Kapitel 4 verwendeten Resultat Fuzzy Sets sind. Durch Maximumbildung nach der Defuzzifizierung erfolgt die Klassifikation wieder entsprechend dem größten Resultatwert der verschiedenen Klassen.

Ein Median Operator , der eine 3*3 Umgebung des aktuellen Schichtbildes berücksichtigt übergibt dem Fuzzy Tool den Grauwert inklusive der Umgebungsinformationen.

Es kann folgendes Verknüpfungsschema angegeben werden:

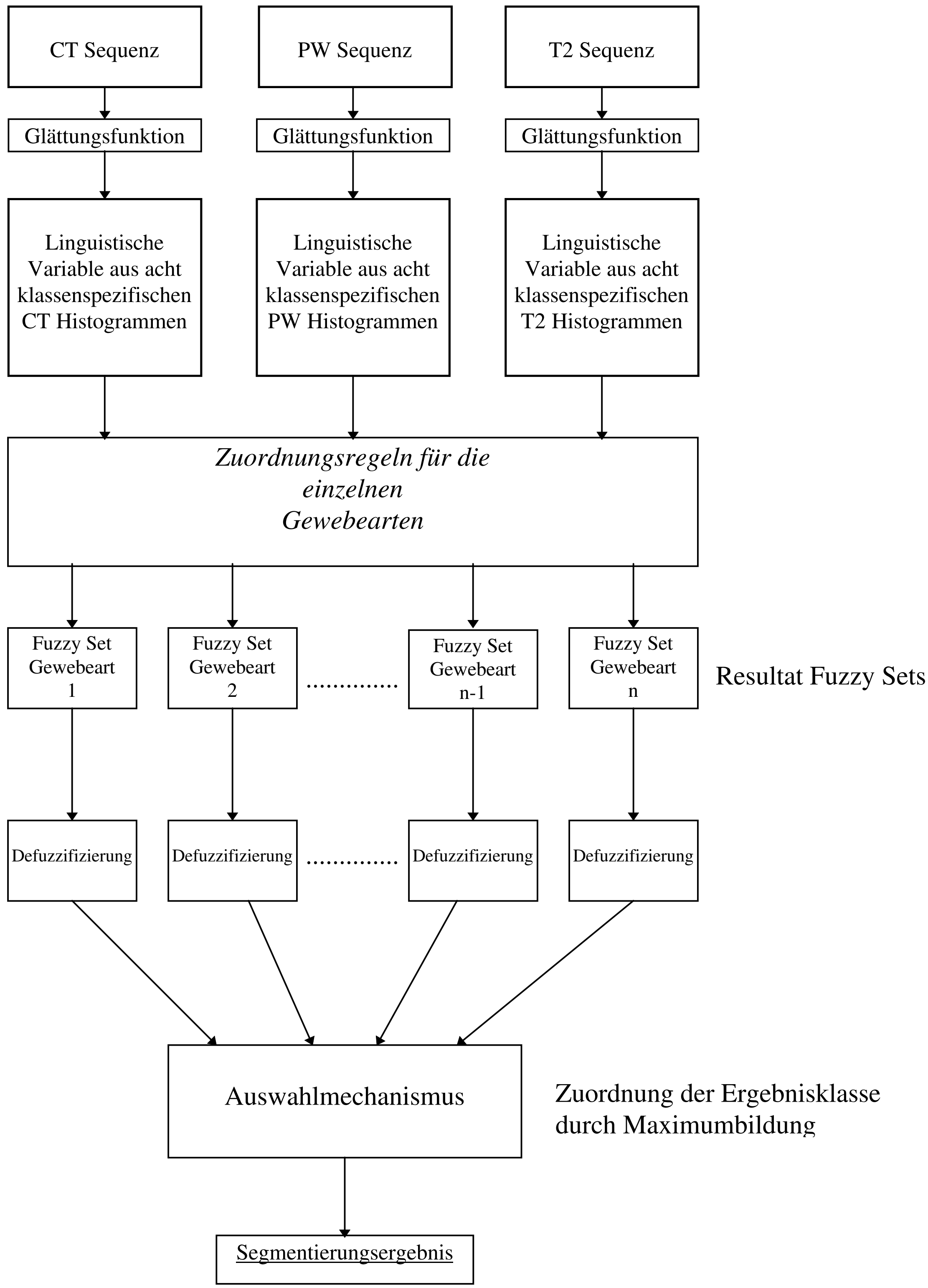


Abbildung 5.7 Fuzzifizierung aus Originaldaten

Die Regelbasis wurde von Hand erstellt, indem die Zugehörigkeitswerte jeder Gewebeklasse untereinander mit dem Einstein Produkt Operator verknüpft wurden. In Versuchen konnte durch anschließenden Datenvergleich mit den Testdaten festgestellt werden, daß das Einstein Produkt bessere Werte liefert als beispielsweise das arithmetische Produkt oder der Min Operator. Die drei linguistischen Eingangsvariablen haben die Wichtungsnamen der Ausgangsbilder (CT, PW,T2), gefolgt von den Termen mit den Histogramm-zugehörigkeitsfunktionen, aus denen die Zugehörigkeitswerte berechnet werden können. Die linguistischen Resultatvariablen erhalten die Namen der mit ihnen assoziierten Gewebeklassen.

```
if PW PW_TUMOR   T2 T2_TUMOR   EP CT CT_TUMOR   EP then TUMOR   big
if PW PW_LIQUOR  T2 T2_LIQUOR  EP CT CT_LIQUOR  EP then liquor  big
if PW PW_KNOCHEN T2 T2_KNOCHEN EP CT CT_KNOCHEN EP then KNOCHEN big
if PW PW_FETT    T2 T2_FETT    EP CT CT_FETT    EP then FETT    mid
if PW PW_GRAU    T2 T2_GRAU    EP CT CT_GRAU    EP then weiss   mid
if PW PW_WEISS   T2 T2_WEISS   EP CT CT_WEISS   EP then grau    mid
if T2 T2_BACK    CT CT_BACK    EP then BACK                     big
```

5.3 Diskussion des direkt aus Originalsequenzen entstandenen Fuzzy Resultatbildes

Das Resultatbild zeigt Schicht 10 der Sequenz da55. Es wurde ausschließlich durch an die Histogrammdaten angepaßte Fuzzy Sets erzeugt. Die Qualität des Bildes ist schlechter als bei einer Verknüpfung direkt aus Resultat-sequenzen. Die Klassifikation erfolgt ausschließlich durch die angepaßten Histogrammfunktionen. Würde man noch weitere Klassifkatoren einsetzen und diese ebenfalls in die Struktur des Fuzzy Klassifikationsmechanismus mit einbauen, so ließe sich das Ergebenis mit Sicherheit noch verbessern. Außerhalb des Schädels sind zum Teil Utensilien der Röntgenapparatur mit abgebildet, diese müssen in einem weiteren Nachbearbeitungsschritt entfernt werden. Dem hier gezeigten Schädel liegt ein leicht erweiterter Fuzzy Regelsatz zugrunde, der im Anhang nachgelesen werden kann.

Abbildung 5.8 Ein direkt aus Originalsequenzen segmentiertes Bild

Sieht man von den außerhalb des Schädels liegenden Teilen der Röntgenapparatur ab, die fälschlicherweise als Knochen klassifiziert wird, so wird Knochen sehr gut erkannt. Dies liegt vor allem daran, daß Knochen im CT Bild eindeutig zu identifizieren ist. Diese Eigenschaft macht sich auch der zur Segmentierung von Knochen eingesetzte Algorithmus Res_thresh zunutze. Auch die Trennung von Tumor und Liquor ist noch einigermaßen akzeptabel. Allerdings sind an den Liquor signalisierenden blauen Stellen grundsätzlich sehr viele Tumor Fehlklassifikationen zu finden. Eine Trennung zwischen weißem und grauem Gehirngewebe ist nicht möglich, da die Histogramme in allen drei Eingangsbändern sehr ähnlich aussehen. Hier versagt das Klassifikationsverfahren. Fettgewebe wurde zwar erkannt, ist aber im Resultatbild stark unterrepräsentiert. Das Resultatbild kann mit den schon vorhandenen Segmentierungsalgorithmen nicht konkurrieren, dennoch bietet die verwendete Regelbasis noch sehr viele Möglichkeiten der Verbesserung des Bildes. Es liegt nahe, daß mit einer modifizierten Form des ID3 Algorithmus eventuell Regeln erzeugt werden können, die eine weitaus bessere Informationstrennung zulassen. Eine weitere Möglichkeit der Verbesserung des Segmentierungsverfahrens könnte erreicht werden, wenn zu den drei Eingangsvariablen die Umgebungspixel aus den Originalbildern zusätzlich als Eingangsvariablen verwendet würden und in den Fuzzy Regeln ebenfalls zur Informationstrennung benutzt werden könnten. Mit der vorgestellten Segmentierungsstrategie lassen sich durch Verbesserungen der Regelbasis und hinzufügen weiterer informationstrennender Maßnamen mit Sicherheit in relativ kurzer Zeit wesentliche Verbesserungen bei der Segmentierungsqualität erzielen. Mit dem vorgestellten Ergebnis konnte gezeigt werden, daß eine Segmentierung auf Basis von Fuzzy Logik möglich ist und sich mit relativ einfachen Mitteln durchaus gute Ergebnisse erzeugen lassen.

6 Zusammenfassung der Ergebnisse und Ausblick

In diesem Kapitel werden die wesentlichen Untersuchungsergebnisse der durch Fuzzy Logik segmentierten Resultatsequenzen vorgestellt und diskutiert.

6.1 Bewertung der Segmentierungsergebnisse aus Resultatsequenzen

Zur Präsentation der Ergebnisse wurden die Resultatsequenzen von vier ausgewählten Fuzzy Regelbasen mit den in den ROI Daten gespeicherten Testergebnissen verglichen und ausgewertet. Die Tabellen 6.1 bis 6.4 enthalten sogenannte Konfusionsmatrizen, deren Diagonale klassenweise die Anzahl der richtig klassifizierten Rasterpunkte enthält. Die Spaltensumme ergibt die Gesamtzahl der in den ROI's gespeicherten Pixel einer Klasse. Im Idealfall, bei einer perfekten Segmentierung, enthält das Diagonalelement 100 % der Rasterpunkte einer Klasse und alle restlichen Klassen der betrachteten Spalte sind Null. Im Anhang sind die hier aufgeführten Konfusionsmatrizen noch einmal zahlemäßig (Anzahl Pixel) aufgeführt, außerdem findet man dort die Matrizen für sämtliche Segmentierungsalgorithmen. Konfusionsmatrizen haben sich bei der betrachteten Arbeit bewährt, da nicht nur die richtig erkannten Pixel betrachtet werden können, sondern es geht aus ihnen genau hervor welchen Klassen bei Fehlklassifikationen die entsprechenden Pixel zugeordnet wurden. Die Regelbasis *Pure* ist die einfachste Regelbasis. Sie enthält lediglich für jede Klasse eine Zuordnungsregel und keine Fuzzy Operatoren. Die Fuzzy Resultatsequenz enthält jeweils die nach dem Qualitätsmaß gewichteten Pixel der Klassensieger der unterschiedlichen Sequenzen. Bei der Regelbasis *Dominant* wurde der beste Segmentierungsalgorithmus direkt auf die Resultatterms *mid* abgebildet. Weitere UND Verknüpfungen auf die Resultatterms *hoch* sollen das Fuzzy Resultatbild weiter verbessern. Bei den Regelbasen *F_and* und *Visual* wurden die Regeln automatisch, unter Zuhilfenahme des ID3 Algorithmus erzeugt, dabei wurden bei *Visual* einige Regeln von Hand geändert um dem Fuzzy Resultatbild auch visuellen Erwägungen wie z.B. der Forderung nach einem zusammenhängenden Schädelknochen gerecht zu werden.

Allen Regelbasen liegen die selben Eingangs- und Resultatzugehörigkeitsfunktionen zugrunde. Die einzelnen Regelbasen sind im Anhang auf Seite 102 beschrieben.

Pure	STAT	Tumor	Liquor	Oedem	Knochen	Fett	Grau	Weiß	Hintergrund
STAT	**0,00%**	0,26%	0,84%	0,00%	0,11%	0,00%	0,06%	0,00%	0,00%
Tumor	0,00%	**93,43%**	4,77%	0,00%	0,00%	0,00%	0,00%	0,02%	0,00%
Liquor	0,00%	6,30%	**93,29%**	0,00%	0,00%	0,00%	0,19%	0,00%	0,00%
Oedem	0,00%	0,00%	0,00%	**0,00%**	0,00%	0,00%	0,00%	0,00%	0,00%
Knochen	0,00%	0,00%	0,06%	0,00%	**98,71%**	1,11%	0,25%	0,00%	0,00%
Fett	0,00%	0,00%	0,13%	0,00%	0,53%	**97,14%**	0,00%	0,00%	0,00%
Grau	0,00%	0,00%	0,84%	0,00%	0,65%	1,05%	**80,08%**	1,87%	0,00%
Weiß	0,00%	0,00%	0,06%	0,00%	0,00%	0,69%	19,42%	**98,11%**	0,00%
Hintergrund	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	**100,00%**

Tabelle 6.1 Konfusionsmatrix einer Verknüpfung nach Klassensiegern

Dominant	STAT	Tumor	Liquor	Oedem	Knochen	Fett	Grau	Weiß	Hintergrund
STAT	**0,00%**	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
Tumor	0,00%	**92,38%**	4,71%	0,00%	0,00%	0,00%	0,06%	0,00%	0,00%
Liquor	0,00%	7,53%	**93,10%**	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
Oedem	0,00%	0,00%	0,00%	**0,00%**	0,00%	0,00%	0,00%	0,00%	0,00%
Knochen	0,00%	0,00%	0,00%	0,00%	**98,56%**	1,05%	0,25%	0,00%	0,00%
Fett	0,00%	0,00%	0,13%	0,00%	0,60%	**97,20%**	0,00%	0,00%	0,00%
Grau	0,00%	0,04%	1,68%	0,00%	0,84%	1,05%	**80,27%**	1,87%	0,00%
Weiß	0,00%	0,04%	0,39%	0,00%	0,00%	0,69%	19,42%	**98,13%**	0,00%
Hintergrund	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	**100,00%**

Tabelle 6.2 Konfusionsmatrix Verknüpfung mit einem dominanten Algorithmus

Visual	STAT	Tumor	Liquor	Oedem	Knochen	Fett	Grau	Weiß	Hintergrund
STAT	**0,00%**	9,37%	27,01%	0,00%	0,28%	0,14%	6,73%	3,40%	0,00%
Tumor	0,00%	**89,84%**	3,61%	0,00%	0,00%	0,03%	0,00%	0,05%	0,00%
Liquor	0,00%	0,79%	**67,70%**	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
Oedem	0,00%	0,00%	0,00%	**0,00%**	0,00%	0,00%	0,00%	0,00%	0,00%
Knochen	0,00%	0,00%	0,00%	0,00%	**99,46%**	5,86%	0,95%	0,00%	0,31%
Fett	0,00%	0,00%	0,06%	0,00%	0,01%	**92,73%**	0,00%	0,00%	0,00%
Grau	0,00%	0,00%	1,55%	0,00%	0,24%	0,61%	**79,44%**	0,78%	0,00%
Weiß	0,00%	0,00%	0,06%	0,00%	0,00%	0,64%	12,88%	**95,77%**	0,00%
Hintergrund	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	**99,69%**

Tabelle 6.3 Konfusionsmatrix automatischer Regeln u.a. nach visuellen Kriterien

F_and	STAT	Tumor	Liquor	Oedem	Knochen	Fett	Grau	Weiß	Hintergrund
STAT	**0,00%**	9,37%	27,08%	0,00%	0,65%	4,64%	6,73%	3,40%	0,00%
Tumor	0,00%	**89,84%**	3,61%	0,00%	0,00%	0,03%	0,00%	0,05%	0,00%
Liquor	0,00%	0,79%	**67,70%**	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
Oedem	0,00%	0,00%	0,00%	**0,00%**	0,00%	0,00%	0,00%	0,00%	0,00%
Knochen	0,00%	0,00%	0,00%	0,00%	**98,34%**	1,05%	0,25%	0,00%	0,00%
Fett	0,00%	0,00%	0,00%	0,00%	0,36%	**93,31%**	0,00%	0,00%	0,00%
Grau	0,00%	0,00%	1,55%	0,00%	0,65%	0,53%	**80,14%**	0,78%	0,00%
Weiß	0,00%	0,00%	0,06%	0,00%	0,00%	0,44%	12,88%	**95,77%**	0,00%
Hintergrund	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	**100,00%**

Tabelle 6.4 Konfusionsmatrix mit automatisch erzeugten Regeln

Die Auswertung der Fuzzy Resultatsequenzen brachte Überraschungen mit sich, so schnitten die mit dem ID3 Algorithmus erzeugten automatischen Regeln *F_and* und *Visual* schlechter ab, als die einfachsten Regelbasen *Pure* und *Dominant*. Interessanterweise erzeugte die trivialste Regelbasis das beste Segmentierungsergebnis (98,24 % richtig erkannnte Pixel) und lag damit besser als der im Gesamtergebnis beste Segmentierungsalgorithmus (dect_en_5 - 98,14%). Dennoch stellt sich angesichts der kleinen Differenzen im Gesamtergebnis die Frage, ob sich der zusätzliche Aufwand bei der Erkennung wirklich lohnt. Dazu ist zu sagen, daß die vorliegenden Segmentierungsalgorithmen allein schon extrem gut klassifizieren und die nie erreichbare Trennschärfe von 100 % beim besten Einzelalgorithmus nur um 1,86 % verfehlt wurde. Eine Steigerung der Trennschärfe um 0,1 % kann deshalb durchaus als Erfolg gewertet werden. Allerdings kann die Verbesserung nicht dem Einsatz der Fuzzy Logik zugeschrieben werden, da eine klassenbezogene Gewichtung der Eingangssequenzen auch mit weniger

Aufwand durchgeführt werden kann. Die eigentliche Verbesserung brachte hier die Verwendung des Qualitätsmaßes, dessen Ergebnisse als Gewichtungsfaktoren eingesetzt, eine sehr gute, vielleicht die bestmögliche Gewichtung zwischen den zu verknüpfenden Sequenzen ermöglicht. Der Einstatz des ID3 Algorithmus zur Informationstrennung hat beim vorliegenden Segmentierungsproblem wider Erwarten nicht die besten Resultate hervorgebracht. Die Hauptursache dafür scheinen jedoch die schon sehr guten Segmentierungsalgorithmen als Eingangsdaten für das Fuzzy Logik Tool zu sein. Da die Informationstrennung nur bis zur fünften Iterationsebene durchgeführt wurde und auch nur homogene Blätter mit vielen Testbeispielen zur Regelgenerierung verwendet wurden, ist die klassenbezogene Trennschärfe bei einer normalen Gewichtung nach dem Qualitätsmaß höher, als die durch den ID3 Algorithmus erzeugten Regeln. Der ID3 Algorithmus ist sicherlich ein Weg, bis zu einer gewissen Grenze, automatische Regeln mit sehr hoher Trennschärfe zu erzeugen. Im vorliegenden Fall lag die Qualitätsvorgabe etwas zu hoch. Möglicherweise kann die Informationstrennung durch größere Iterationstiefen und kleineren Toleranzvorgaben beim Homogenitätskriterium noch weiter verbessert werden. Der ID3 Algorithmus macht vor allem bei der Erzeugung von Regeln für die Klasse *Liquor* Schwierigkeiten, da das Homogenitätskriterium für zu wenig Blätter erfüllt wird. Betrachtet man sich die Histogramme der Testklassen, oder auch die Konfusionsmatritzen, so fällt auf, daß sich Tumor und Liquor stark überlagern und eine Trennung sehr schwer möglich ist. Dies spiegelt sich auch im Klassifikationsergebnis von *F_and* und *Visual* wieder. Es wurden jeweils zwei Ergebnisse aufgeführt. Das Ergebnis (Gesamt) drückt die Anzahl richtig klassifizierter Pixel bezüglich der Testgebiete aus. Das zweite Ergebnis enthält den Durchschnitt der Klassenergebnisse. Da ein großteil der klassifizierten Pixel Hintergrund ist, und dieser zu 100 Prozent erkannt werden kann, sieht die Gesamtstatistik sehr viel besser aus als wenn man die einzelnen Gewebeklassen im Schnitt betrachtet.

	Pure	Dominant	F_and	Visual
Tumor	93,43%	92,38%	89,84%	89,84%
Liquor	93,29%	93,10%	67,70%	67,70%
Oedem				
Knochen	98,71%	98,56%	98,34%	99,46%
Fett	97,14%	97,20%	93,31%	92,73%
Graue Substanz	80,08%	80,27%	80,14%	79,44%
Weiße Substanz	98,11%	98,13%	95,77%	95,77%
Hintergrund	100,00%	100,00%	100,00%	99,69%
Gesamt	**98,24%**	**98,18%**	**96,66%**	**96,59%**
Durchschnitt	**94,93%**	**94,23%**	**89,30%**	**89,23%**

Tabelle 6.5 Bewertungstabelle der verknüpften Resultatbilder

Für die Regelbasen *Dominant*, *F_and* und *Visual* wurden die Sequenzen erneut nach einem Postprocessing ausgewertet. Das Postprocessing führt zu einer weiteren Verbesserung des Klassifikationsergebnisses. Vor allem die von nichtklassifizierten Pixeln durchdrungene Gewebeklasse Liquor wird durch die generalisierende Reklassifikation der Nachbearbeitung bei den durch automatische Regeln generierten Sequenzen stark verbessert.

	Dominant	F_and	Visual
Tumor	95,79%	95,94%	95,94%
Liquor	91,84%	77,28%	77,28%
Oedem			
Knochen	98,53%	98,43%	99,60%
Fett	98,44%	96,13%	93,33%
Graue Substanz	78,95%	84,43%	83,94%
Weiße Substanz	98,20%	98,58%	98,58%
Hintergrund	100,00%	100,00%	100,00%
Gesamt	**98,44%**	**98,04%**	**97,98%**
Durchschnitt	**94,53%**	**92,97%**	**92,67%**

Tabelle 6.6 Bewertungstabelle nach dem Postprocessing

Die Tabellen 6.7 bis 6.9 stellen die Konfusionsmatrizen der nachbearbeiteten Sequenzen dar.

Dominant	STAT	Tumor	Liquor	Oedem	Knochen	Fett	Grau	Weiß	Hintergrund
STAT	0,00%	0,00%	0,00%	0,00%	0,00%	0,26%	0,00%	0,00%	0,00%
Tumor	0,00%	95,79%	5,50%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
Liquor	0,00%	4,18%	91,84%	0,00%	0,00%	0,00%	0,31%	0,00%	0,00%
Oedem	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
Knochen	0,00%	0,00%	0,00%	0,00%	98,53%	1,08%	0,25%	0,00%	0,00%
Fett	0,00%	0,00%	0,00%	0,00%	0,82%	98,44%	0,00%	0,00%	0,00%
Grau	0,00%	0,00%	2,21%	0,00%	0,65%	0,00%	78,95%	1,80%	0,00%
Weiß	0,00%	0,03%	0,45%	0,00%	0,00%	0,00%	20,49%	98,20%	0,00%
Hintergrund	0,00%	0,00%	0,00%	0,00%	0,00%	0,22%	0,00%	0,00%	100,00%

Tabelle 6.7 Konfusionsmatrix der Sequenz Dominant nach einem Postprocessing

F_and	STAT	Tumor	Liquor	Oedem	Knochen	Fett	Grau	Weiß	Hintergrund
STAT	**0,00%**	4,03%	9,18%	0,00%	0,18%	2,05%	0,31%	0,23%	0,00%
Tumor	0,00%	**95,94%**	3,74%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
Liquor	0,00%	0,00%	**77,28%**	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
Oedem	0,00%	0,00%	0,00%	**0,00%**	0,00%	0,00%	0,00%	0,00%	0,00%
Knochen	0,00%	0,00%	0,00%	0,00%	**98,43%**	1,08%	0,25%	0,00%	0,00%
Fett	0,00%	0,00%	0,00%	0,00%	0,65%	**96,13%**	0,00%	0,00%	0,00%
Grau	0,00%	0,00%	8,78%	0,00%	0,74%	0,00%	**84,43%**	1,19%	0,00%
Weiß	0,00%	0,03%	1,02%	0,00%	0,00%	0,00%	15,01%	**98,58%**	0,00%
Hintergrund	0,00%	0,00%	0,00%	0,00%	0,00%	0,75%	0,00%	0,00%	**100,00%**

Tabelle 6.8 Konfusionsmatrix der Sequenz F_and nach einem Postprocessing

Visual	STAT	Tumor	Liquor	Oedem	Knochen	Fett	Grau	Weiß	Hintergrund
STAT	**0,00%**	4,03%	9,18%	0,00%	0,00%	0,55%	0,12%	0,23%	0,00%
Tumor	0,00%	**95,94%**	3,74%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
Liquor	0,00%	0,00%	**77,28%**	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
Oedem	0,00%	0,00%	0,00%	**0,00%**	0,00%	0,00%	0,00%	0,00%	0,00%
Knochen	0,00%	0,00%	0,00%	0,00%	**99,60%**	5,90%	0,93%	0,00%	0,00%
Fett	0,00%	0,00%	0,00%	0,00%	0,10%	**93,33%**	0,00%	0,00%	0,00%
Grau	0,00%	0,00%	8,78%	0,00%	0,29%	0,00%	**83,94%**	1,19%	0,00%
Weiß	0,00%	0,03%	1,02%	0,00%	0,00%	0,00%	15,01%	**98,58%**	0,00%
Hintergrund	0,00%	0,00%	0,00%	0,00%	0,00%	0,22%	0,00%	0,00%	**100,00%**

Tabelle 6.9 Konfusionsmatrix der Sequenz Visual nach einem Postprocessing

6.2 Bewertung der Segmentierungsergebnisse aus Originalsequenzen

Seghist1	STAT	Tumor	Liquor	Oedem	Knochen	Fett	Grau	Weiß	Hintergrund
STAT	**0,00%**	0,60%	6,91%	0,00%	1,89%	6,81%	1,73%	0,05%	12,24%
Tumor	0,00%	**86,63%**	14,42%	0,00%	0,00%	0,28%	2,21%	0,10%	0,00%
Liquor	0,00%	0,38%	**76,70%**	0,00%	0,00%	0,09%	0,29%	0,00%	0,01%
Oedem	0,00%	0,00%	0,00%	**0,00%**	0,00%	0,00%	0,00%	0,00%	0,00%
Knochen	0,00%	0,26%	0,09%	0,00%	**98,11%**	2,04%	2,21%	0,15%	21,22%
Fett	0,00%	5,44%	0,09%	0,00%	0,00%	**80,31%**	1,88%	0,94%	0,03%
Grau	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	**0,00%**	0,00%	0,00%
Weiß	0,00%	6,69%	1,80%	0,00%	0,00%	10,05%	91,67%	**98,76%**	0,02%
Hintergrund	0,00%	0,00%	0,00%	0,00%	0,00%	0,43%	0,00%	0,00%	**66,49%**

Tabelle 6.10 Konfusionsmatrix zur Auswertung von Originaldaten

Die Konfusionsmatrix spiegelt durch Zahlen wieder, was bei der Betrachtung des Resultatbildes (Abbildung 5.8) schon erwähnt wurde. Zwar sind die aus den Resultatsequenzen gewonnenen Bilder besser als die Bilder, die direkt aus den Originaldaten erzeugt wurden, dennoch werden auch bei der Segmentierung aus Originalsequenzen z.T. sehr gute Ergebnisse erzielt. Dadurch, daß gewisse Teile des Tomographen als Knochen klassifiziert wurden, die in den Testgebieten eindeutig zum Hintergrund gehören, wird die Klasse HINTERGRUND in der Gesamtstatistik schlechter bewertet, als sie wirklich ist. Durch eine entsprechende Vorverarbeitung lassen sich die störenden Nebeneffekte aus dem Bild beseitigen und damit würde die Erkennungsquote für die Klasse Hintergrund noch einmal steigen, da dann nur noch wenige Pixel fälschlicherweise als Knochen klassifiziert würden. Ein Algorithmus zur Erkennung von Knochen, der ähnlich gute Ergebnisse liefert wie der Algorithmus Res_thresh wäre mit dem vorhandenen Fuzzy Tool ohne weiteres realisierbar, indem beispielsweise alle Regeln außer jene, die zu Klassifizierung von Knochen eingesetzt werden, die Terme der Sonderklasse STAT modifizieren. Bedenkt man, daß bei den schon vorhanden Segmentierungsalgorithmen ein erheblicher Aufwand bei der Merkmalsextraktion und einer anschließenden Klassifikation durch ein neuronales Netz stattfindet, so sind die Resultate, die sich mit dem Segmentierungsalgorithmus auf Basis von Fuzzy Logik erzielen lassen erstaunlich gut. Lediglich eine Trennung von grauer und weißer Gehirnsubstanz ist mit der angewandten Klassifikationsmethode nicht möglich.

6.3 Ausblick

Der Umgang mit medizinischen Daten von Patienten ist ethisch gesehen nicht unproblematisch, handelt es sich hier doch um Menschen, die ein schweres Schicksal zu erleiden haben. Dennoch ist für die Lösung des Problems eine wissenschaftliche Sichtweise - unabhängig vom einzelnen Menschen - unabdingbar. Die Gewebeklassifikation auf Basis von Tomographiedaten kann nur als unterstützende Maßname im medizinischen Diagnoseprozeß verwendet werden. Dennoch kann durch eine Verbesserung der Tumorerkennung die ärztliche Diagnose genauer gestellt werden als Früher.
Desweitern läßt sich bei genauer Kenntnis der Tumorkonturen das Bestrahlungspolygon auf ein Minimum reduzieren, so daß die Patienten schonender und doch effektiver behandelt werden können als das noch vor wenigen Jahren der Fall war. Da es heute nur wenige ausgebildete Spezialisten gibt, die in der Lage sind, die Tomographierohdaten direkt auszuwerten, kann durch eine Automatisierung die Früherkennung beschleunigt werden und im Falle eines positiven Befundes früher als bisher, tumorbekämpfende Maßnamen eingeleitet werden, was mitunter Leben retten kann.

Die Verwendung der Fuzzy Logik im Bereich der Bildverarbeitung ist ein relativ neues Gebiet auf dem noch nicht allzuviele Erfahrungen gesammelt wurden. Die vorliegende Arbeit hat gezeigt, daß der Ansatz der Fuzzy Logik zur Bildsegmentierung sinnvoll ist und daß sich schon mit einfachen Methoden der Merkmalstrennung gute Segmentierungsgebnisse erzielen lassen.

Die Fuzzy Logik genießt zurecht den Ruf etwas undurchschaubar zu sein, denn trotz daß das mathematische Gerüßt auf festem Boden steht, fehlt es doch an Konzepten, die die Handhabung von Fuzzy Logik vereinfachen. Richtlinien zur Wahl der Fuzzy Sets gibt es keine, denn das Aussehen der Mengen ist abhängig vom Problem. Zur Erstellung von Regeln wurden zwei Konzepte vorgestellt, eines auf statistischer Basis und ein baumorientiertes Auswahlverfahren unter Nutzung des ID3 Algorithmus, durch die es möglich wurde die Regelgenerierung zu systematisieren. Ein weiteres Problem der Fuzzy Logik stellt die Auswahl des für das Problem am besten geeignetsten Fuzzy Operator dar. Es ist durchaus möglich, bei Verwendung anderer Operatoren und Regeln, nach entsprechender Anpassung der Regelbasis, zwei Fuzzy Systeme zu erzeugen, die bei der Entscheidungsfindung ähnlich gut arbeiten. Dennoch, trotz einiger Schwierigkeiten bei der Handhabung erweist sich Fuzzy Logik als sehr zuverlässiges und stabiles Logiktool dessen Leistungsfähigkeit zweifelsohne die reine Binärlogik übertrifft. Zadeh schlägt selbst vor, Fuzzy Logik als festen Bestandteil in gängige Anwendungsentwicklungssysteme zu integrieren.

Fuzzy Logik sollte andererseits nicht überbewertet werden, denn es handelt sich eben nur um Logik, die mit unscharfen Sachverhalten besser umzugehen vermag als Binärlogik. Im Rahmen dieser Arbeit hat sich auch gezeigt daß Fuzzy Logik nur in Verbindung mit der Anwendung von weiteren Verfahren der Bildverarbeitung vernünftig eingesetzt werden kann. So waren beispielsweise die Ergebenisse bei der Bildsegmentierung aus Originalsequenzen im Vergleich zu anderen Verfahren lediglich durchschnittlich. Mit einer Vorverarbeitung der Bilder, beispielsweise eine Zerlegung in homogene Cluster oder eine Mustererkennung könnte man die Qualität der Ergebnissequenzen mit Sicherheit noch weiter verbessern. Dies zeigt auch die Anwendung der im ersten Teil vorverarbeiteten Resultatsequenzen, bei deren Verknüpfung es möglich war, fast durchgängig homogene Bereiche für die verschiedenen Gewebearten zu erhalten.

Die Neuartigkeit der Technik und die sich Laufe ergebenden Probleme machten diese Arbeit sowohl im Rahmen der Implementierung als aus auch im konzeptionellen Teil sehr interessant.

7 Anhang

7.1 Fuzzy Operatoren

Eigenschaften von Operatoren

Die grundlegenden Funktionen der Mengenlehre Schnitt, Vereinigung und Komplement können ebenfalls als Fuzzy Sets dargestellt werden. Man erwartet daß die Schnittmenge $\mu \cap \mu'$ zweier Fuzzy Sets $\mu,\mu' \in F(X)$ elementweise berechnet werden können, d.h. es gibt eine Funktion $\prod$: $[0,1]^2 \rightarrow [0,1]$ so daß gilt $\mu'(x) = \prod(\mu(x),\mu'(x))$.

Der Schnittmengenoperator muß zusätzlich die t-norm Bedingungen erfüllen (7.2.7). Neben den schon vorgestellten „einfachen“ Operatoren gibt es noch eine Reihe kompensatorischer Operatoren. Kompensatorische Operatoren verkörpern kein reines „und“ und kein reines „oder“ sondern liefern als Ergebnis ein Kompromiß zwischen diesen beiden Operatoren , meist in Abhängigkeit von einer weiteren Variablen, die die Tendenz vorgibt.

Kompensatorische Operatoren sind für das unscharfe Schließen und bei der Entscheidungsfindung außerordentlich wichtig. Die wichtigsten kompensatorischen Operatoren sind Hamacher - Intersection - Union, Yager - Intersection - Union und der Gamma - Operator.

- **Drastisches Produkt** dp und **Drastische Summe** ds

$$\mathbf{dp}(\mu_A(\mathbf{x}), \mu_B(\mathbf{x})) = \begin{Bmatrix} \min\{\mu_A , \mu_B\} & \textbf{falls} & \max\{\mu_A , \mu_B\} = 1 \\ 0 & \textbf{sonst} & \end{Bmatrix} \qquad \textbf{(7.1)}$$

$$\mathbf{ds}(\mu_A(\mathbf{x}), \mu_B(\mathbf{x})) = \begin{Bmatrix} \max\{\mu_A , \mu_B\} & \textbf{falls} & \min\{\mu_A , \mu_B\} = 0 \\ 1 & \textbf{sonst} & \end{Bmatrix} \qquad \textbf{(7.2)}$$

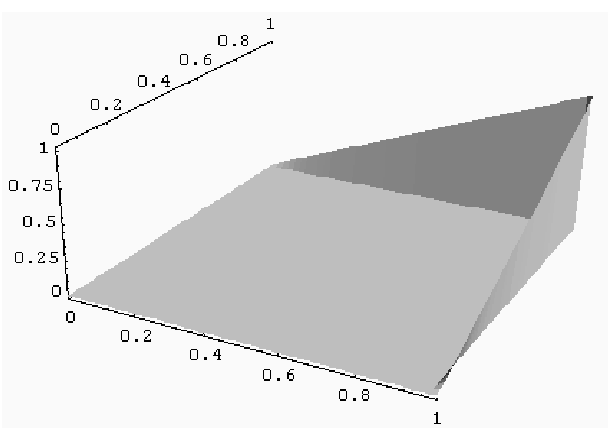


Abbildung 7.1 - Drastisches Prudukt

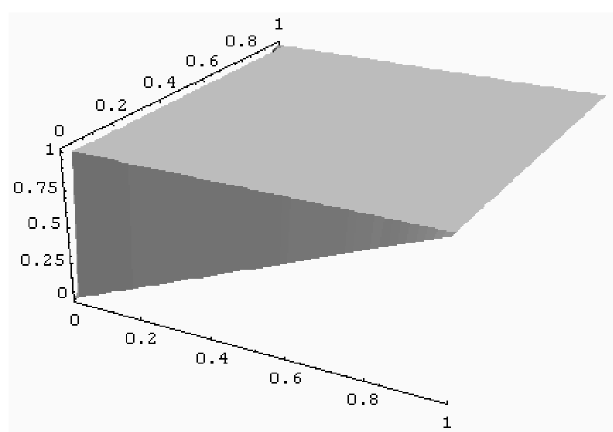


Abbildung 7.2 - Drastische Summe

- Bounded difference **bd und** bounded sum **bs**

$$\mathbf{bd(\mu_A(x), \mu_B(x)) = max(0, \mu_A(x) + \mu_B(x) -1)} \qquad (7.3)$$

$$\mathbf{bs(\mu_A(x), \mu_B(x)) = min(1, \mu_A(x) + \mu_B(x))} \qquad (7.4)$$

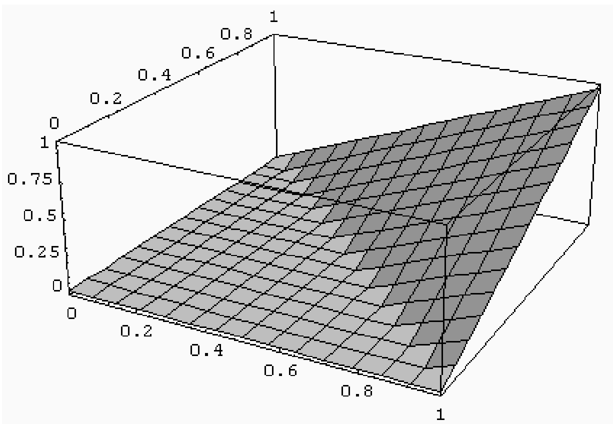


Abbildung 7.3 - bounded difference

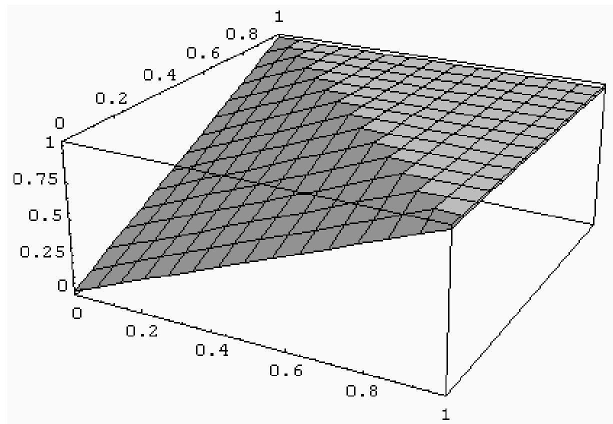


Abbildung 7.4 - bounded sum

- **Einstein Produkt** (ep) und **Einstein Summe** (es)

$$ep(\mu_A(x), \mu_B(x)) = (\mu_A(x) * \mu_B(x)) /(2 - [\mu_A(x) + \mu_B(x) - \mu_A(x) * \mu_B(x)]) \qquad \mathbf{(7.5)}$$

$$es(\mu_A(x), \mu_B(x)) = (\mu_A(x) + \mu_B(x)) /(1 + \mu_A(x) * \mu_B(x)) \qquad \mathbf{(7.6)}$$

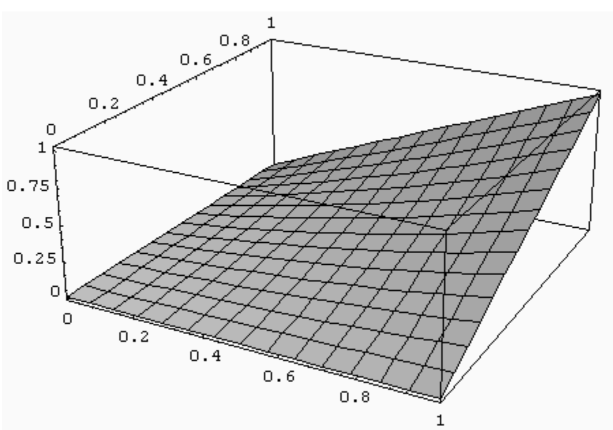


Abbildung 7.5 - Einstein Produkt

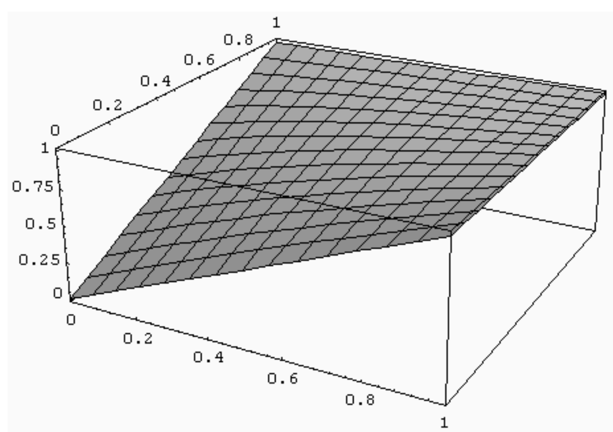


Abbildung 7.6 - Einstein Summe

7.2 Begriffserklärungen:

7.2.1 charakteristische Funktion:

Im Zusammenhang mit Fuzzy Logik versteht man unter einer charakteristischen Funktion eine Sprungfunktion, die jedem Element $x < x_0$ einer linear geordneten Menge X einen Wert $a \in X$ zuordnet und jedem Element $x \geq x_0$ einen Wert $b \in X$ zuordnet.

7.2.2 Fuzzy Set:

Ein Fuzzy Set ist eine Menge die jedem Element $x \in X$ einen Zugehörigkeitsgrad $\mu \in [0,1]$ zuordnet. Das selbe Element x kann - im Gegensatz zur zweiwertigen Logig - auch Element einer weiteren Menge X' sein. Unter $\mu(X)$ versteht man die Menge aller Zugehörigkeitsgrade $\mu \in [0,1]$ eines Fuzzy Sets. Der Wert $\mu(x)$ bezeichnet dabei ein Element aus einer Menge $\mu(X)$.

7.2.3 Fuzzy Term:

Ein Fuzzy Term drückt das gleiche aus wie ein Fuzzy Set. Im Rahmen dieser Arbeit wird der Begriff dann verwendet, wenn mehr auf den Aspekt der Namensgebung eingegengen wird, ansonsten wird Fuzzy Set oder Zugehörigkeitsfunktion verwendet.

7.2.4 Linguistische Variable:

Eine linguistische Variable drückt einen unscharfen Zustand aus. Dieser Zustand wird repräsentiert durch mehrere Mengenzugehörigkeitsgrade, die jeweils linguistische Begriffe repräsentiern. Die linguistischen Begriffe einer linguistischen Variablen sind linear geordent.

7.2.5 mehrwertige Logik:

Unter mehrwertiger Logik versteht man allgemein eine erweiterte Logik, die neben den klassischen Zuständen Null und Eins noch weitere Zustände kennt.

7.2.6 ROI:

ROI ist eine Abkürzung für Regions of Interest. In dieser Arbeit bezeichnen sie Testgebiete von Gehirnregionen, die einer festen Gewebeklasse zugeordnet sind. Sie dienen als Referenz und Vergleichsdaten zur Beurteilung der Klassifikationseigenschaften von Segmentierungsalgorithmen

7.2.7 t-Norm Bedingungen:

Eine Funktion $\top : [0,1]^2 \rightarrow [0,1]$ heißt t-Norm, wenn die Bedingungen

(i)	$\top(a,1) = a$	*(Neutrales Element bezüglich 1)*
(ii)	$a \leq b \Rightarrow \top(a,c) \leq \top(b,c)$	*(Monotonie)*
(iii)	$\top(a,b) = \top(b,a)$	*(Kommutativität)*
(iv)	$\top(a, \top(b,c)) = \top(\top(a,b),c)$	*(Assoziativität)*

erfüllt sind.

Eine Funktion $\perp$ nennt man t-Conorm, wenn $\perp$ kommutativ, assoziativ, monoton fallend und ihr neutrales Element Null ist.

Für alle Paare von t-Normen und t-Conormen gelten zusätzlich die DeMorganschen Gesetze. Für das spezielle Operatorpaar min,max gilt außerdem noch das Distributivgesetz.

7.2.8 Zugehörigkeitsfunktion:

Eine Zugehörigkeitsfunktion bildet Eingangsdaten auf ein Intervall im Bereich [0 ... 1] ab. Fuzzy Sets können durch Zugehörigkeitsfunktionen beschrieben werden.

7.2.9 Zugehörigkeitsgrad:

Der Zugehörigkeitsgrad legt fest, wie stark ein Element einer bestimmten Menge angehört. Ein Zugehörigkeitswert von $\mu = 1$ bedeutet, daß das Element vollständig in der betrachteten Menge enthalten ist, während bei Zugehörigkeitsgraden kleiner eins das betrachtet Element nur eine graduelle Zugehörigkeit zu einer bestimmten Menge hat.

7.3 Mengendefinition nach Cantor:

Unter einer Menge verstehen wir jede Zusammenfassung ***M*** von bestimmten, wohlunterschiedenen Objekten ***m*** unserer Anschauung oder unseres Denkens (welche die „Elemente“ von ***M*** genannt werden) zu einem Ganzen.

7.4 Konfusionsmatrizen der verschiedenen Segmentierungsalgorithmen

dect_c3_3	STAT	Tumor	Liquor	Oedem	Knochen	Fett	Grau	Weiß	Hintergrund
STAT	**0**	0	0	0	0	0	0	0	0
Tumor	0	**1325**	50	0	97	49	300	324	0
Liquor	0	198	**1407**	0	604	98	130	1	0
Oedem	0	0	0	**0**	0	0	0	0	0
Knochen	0	14	34	0	**5679**	136	29	0	810
Fett	0	146	10	0	210	**3257**	21	2	0
Grau	0	214	48	0	84	56	**712**	161	0
Weiß	0	387	2	0	3	7	384	**5396**	0
Hintergrund	0	0	0	0	358	0	0	0	**26818**

Tabelle 7.1 Algorithmus dect_c3_3

dect_c4_3	STAT	Tumor	Liquor	Oedem	Knochen	Fett	Grau	Weiß	Hintergrund
STAT	**0**	0	0	0	0	0	0	0	0
Tumor	0	**1415**	44	0	23	9	123	196	0
Liquor	0	235	**1424**	0	195	8	68	1	7
Oedem	0	0	0	**0**	0	0	0	0	0
Knochen	0	3	18	0	**6314**	258	65	0	448
Fett	0	70	4	0	292	**3281**	19	0	0
Grau	0	200	61	0	136	47	**1131**	270	0
Weiß	0	361	0	0	0	0	170	**5417**	0
Hintergrund	0	0	0	0	75	0	0	0	**27173**

Tabelle 7.2 Algorithmus dect_c4_3

dect_c6_3	STAT	Tumor	Liquor	Oedem	Knochen	Fett	Grau	Weiß	Hintergrund
STAT	**0**	0	0	0	0	0	0	0	0
Tumor	0	**1539**	57	0	21	25	146	260	0
Liquor	0	254	**1383**	0	169	29	102	1	6
Oedem	0	0	0	**0**	0	0	0	0	0
Knochen	0	2	57	0	**6359**	262	59	0	645
Fett	0	55	18	0	343	**3191**	83	0	0
Grau	0	123	36	0	78	95	**974**	178	0
Weiß	0	311	0	0	0	0	212	**5445**	0
Hintergrund	0	0	0	0	65	1	0	0	**26977**

Tabelle 7.3 Algorithmus dect_c6_3

dect_m1_3	STAT	Tumor	Liquor	Oedem	Knochen	Fett	Grau	Weiß	Hintergrund
STAT	**0**	0	0	0	0	0	0	0	0
Tumor	0	**2130**	290	0	3	10	79	22	0
Liquor	0	64	**1100**	0	10	1	131	1	2
Oedem	0	0	0	**0**	0	0	0	0	0
Knochen	0	0	1	0	**6674**	110	31	0	0
Fett	0	0	0	0	89	**2439**	**16**	56	6
Grau	0	0	0	0	0	0	0	**0**	0
Weiß	0	90	160	0	197	1027	1319	5805	**0**
Hintergrund	0	0	0	0	62	16	0	0	**27620**

Tabelle 7.4 Algorithmus dect_m1_3

dect_m2_3	STAT	Tumor	Liquor	Oedem	Knochen	Fett	Grau	Weiß	Hintergrund
STAT	**0**	0	0	0	0	0	0	0	0
Tumor	0	**2143**	205	0	2	1	49	5	0
Liquor	0	44	**1261**	0	3	0	102	0	0
Oedem	0	0	0	**0**	0	0	0	0	0
Knochen	0	0	0	0	**6685**	114	16	0	0
Fett	0	0	0	0	81	**3195**	5	0	5
Grau	0	0	2	0	82	10	**143**	11	0
Weiß	0	97	83	0	117	276	1260	**5868**	0
Hintergrund	0	0	0	0	65	7	1	0	**27623**

Tabelle 7.5 **Algorithmus dect_m2_3**

dect_m3_3	STAT	Tumor	Liquor	Oedem	Knochen	Fett	Grau	Weiß	Hintergrund
STAT	**0**	0	0	0	0	0	0	0	0
Tumor	0	**2167**	231	0	3	3	64	13	0
Liquor	0	30	**1228**	0	10	3	123	0	0
Oedem	0	0	0	**0**	0	0	0	0	0
Knochen	0	0	0	0	**6743**	79	35	0	0
Fett	0	0	0	0	91	**3152**	11	3	1
Grau	0	0	0	0	2	0	**10**	1	0
Weiß	0	87	92	0	143	351	1329	**5867**	0
Hintergrund	0	0	0	0	43	15	4	0	**27627**

Tabelle 7.6 Algorithmus dect_m3_3

dect_en_3	STAT	Tumor	Liquor	Oedem	Knochen	Fett	Grau	Weiß	Hintergrund
STAT	**0**	0	0	0	0	0	0	0	0
Tumor	0	**2057**	62	0	0	1	0	3	0
Liquor	0	203	**1462**	0	0	0	3	0	0
Oedem	0	0	0	**0**	0	0	0	0	0
Knochen	0	0	0	0	**6947**	43	20	0	0
Fett	0	0	0	0	42	**3448**	0	0	0
Grau	0	18	27	0	40	35	**1185**	274	0
Weiß	0	6	0	0	2	76	368	**5607**	0
Hintergrund	0	0	0	0	4	0	0	0	**27628**

Tabelle 7.7 Algorithmus dect_en_3

dect_en_5	STAT	Tumor	Liquor	Oedem	Knochen	Fett	Grau	Weiß	Hintergrund
STAT	**0**	0	0	0	0	0	0	0	0
Tumor	0	**2134**	74	0	0	0	0	1	0
Liquor	0	150	**1443**	0	0	0	0	0	0
Oedem	0	0	0	**0**	0	0	0	0	0
Knochen	0	0	1	0	**6944**	40	4	0	0
Fett	0	0	1	0	38	**3488**	0	0	2
Grau	0	0	29	0	52	34	**1311**	192	0
Weiß	0	0	3	0	1	41	261	**5691**	0
Hintergrund	0	0	0	0	0	0	0	0	**27626**

Tabelle 7.8 Algorithmus dect_en_5

dect_no_3	STAT	Tumor	Liquor	Oedem	Knochen	Fett	Grau	Weiß	Hintergrund
STAT	**0**	0	0	0	0	0	0	0	0
Tumor	0	**2084**	60	0	0	0	0	3	0
Liquor	0	175	**1472**	0	0	0	3	0	0
Oedem	0	0	0	**0**	0	0	0	0	0
Knochen	0	0	0	0	**6948**	50	9	0	0
Fett	0	0	0	0	41	**3415**	0	0	0
Grau	0	17	13	0	44	43	**1166**	220	0
Weiß	0	8	6	0	0	95	398	**5661**	0
Hintergrund	0	0	0	0	2	0	0	0	**27628**

Tabelle 7.9 Algorithmus dect_no_3

dect_no_5	STAT	Tumor	Liquor	Oedem	Knochen	Fett	Grau	Weiß	Hintergrund
STAT	**0**	0	0	0	0	0	0	0	0
Tumor	0	**2094**	81	0	0	0	1	0	0
Liquor	0	188	**1432**	0	0	0	0	0	0
Oedem	0	0	0	**0**	0	0	0	0	0
Knochen	0	0	0	0	**6934**	38	4	0	0
Fett	0	0	5	0	42	**3502**	0	0	0
Grau	0	1	27	0	59	38	**1265**	110	0
Weiß	0	1	6	0	0	25	306	**5774**	0
Hintergrund	0	0	0	0	0	0	0	0	**27628**

Tabelle 7.10 Algorithmus dect_no_5

res_thresh	STAT	Tumor	Liquor	Oedem	Knochen	Fett	Grau	Weiß	Hintergrund
STAT	**0**	2284	1551	0	38	3392	1561	5884	27542
Tumor	0	**0**	0	0	0	0	0	0	0
Liquor	0	0	**0**	0	0	0	0	0	0
Oedem	0	0	0	**0**	0	0	0	0	0
Knochen	0	0	0	0	**6997**	211	15	0	86
Fett	0	0	0	0	0	**0**	0	0	0
Grau	0	0	0	0	0	0	**0**	0	0
Weiß	0	0	0	0	0	0	0	**0**	0
Hintergrund	0	0	0	0	0	0	0	0	**0**

Tabelle 7.11 Algorithmus res_thresh

Prozentuale Darstellung der Konfusionsmatrizen

dect_c3_3	STAT	Tumor	Liquor	Oedem	Knochen	Fett	Grau	Weiß	Hintergrund
STAT	**0,00%**	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
Tumor	0,00%	**58,01%**	3,22%	0,00%	1,38%	1,36%	19,04%	5,51%	0,00%
Liquor	0,00%	8,67%	**90,72%**	0,00%	8,59%	2,72%	8,25%	0,02%	0,00%
Oedem	0,00%	0,00%	0,00%	**0,00%**	0,00%	0,00%	0,00%	0,00%	0,00%
Knochen	0,00%	0,61%	2,19%	0,00%	**80,72%**	3,77%	1,84%	0,00%	2,93%
Fett	0,00%	6,39%	0,64%	0,00%	2,99%	**90,40%**	1,33%	0,03%	0,00%
Grau	0,00%	9,37%	3,09%	0,00%	1,19%	1,55%	**45,18%**	2,74%	0,00%
Weiß	0,00%	16,94%	0,13%	0,00%	0,04%	0,19%	24,37%	**91,71%**	0,00%
Hintergrund	0,00%	0,00%	0,00%	0,00%	5,09%	0,00%	0,00%	0,00%	**97,07%**

Tabelle 7.12 Algorithmus dect_c3_3

dect_c4_3	STAT	Tumor	Liquor	Oedem	Knochen	Fett	Grau	Weiß	Hintergrund
STAT	**0,00%**	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
Tumor	0,00%	**61,95%**	2,84%	0,00%	0,33%	0,25%	7,80%	3,33%	0,00%
Liquor	0,00%	10,29%	**91,81%**	0,00%	2,77%	0,22%	4,31%	0,02%	0,03%
Oedem	0,00%	0,00%	0,00%	**0,00%**	0,00%	0,00%	0,00%	0,00%	0,00%
Knochen	0,00%	0,13%	1,16%	0,00%	**89,75%**	7,16%	4,12%	0,00%	1,62%
Fett	0,00%	3,06%	0,26%	0,00%	4,15%	**91,06%**	1,21%	0,00%	0,00%
Grau	0,00%	8,76%	3,93%	0,00%	1,93%	1,30%	**71,76%**	4,59%	0,00%
Weiß	0,00%	15,81%	0,00%	0,00%	0,00%	0,00%	10,79%	**92,06%**	0,00%
Hintergrund	0,00%	0,00%	0,00%	0,00%	1,07%	0,00%	0,00%	0,00%	**98,35%**

Tabelle 7.13 Algorithmus dect_c4_3

dect_c6_3	STAT	Tumor	Liquor	Oedem	Knochen	Fett	Grau	Weiß	Hintergrund
STAT	**0,00%**	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
Tumor	0,00%	**67,38%**	3,68%	0,00%	0,30%	0,69%	9,26%	4,42%	0,00%
Liquor	0,00%	11,12%	**89,17%**	0,00%	2,40%	0,80%	6,47%	0,02%	0,02%
Oedem	0,00%	0,00%	0,00%	**0,00%**	0,00%	0,00%	0,00%	0,00%	0,00%
Knochen	0,00%	0,09%	3,68%	0,00%	**90,39%**	7,27%	3,74%	0,00%	2,33%
Fett	0,00%	2,41%	1,16%	0,00%	4,88%	**88,57%**	5,27%	0,00%	0,00%
Grau	0,00%	5,39%	2,32%	0,00%	1,11%	2,64%	**61,80%**	3,03%	0,00%
Weiß	0,00%	13,62%	0,00%	0,00%	0,00%	0,00%	13,45%	**92,54%**	0,00%
Hintergrund	0,00%	0,00%	0,00%	0,00%	0,92%	0,03%	0,00%	0,00%	**97,64%**

Tabelle 7.14 Algorithmus dect_c6_3

dect_m1_3	STAT	Tumor	Liquor	Oedem	Knochen	Fett	Grau	Weiß	Hintergrund
STAT	**0,00%**	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
Tumor	0,00%	**93,26%**	18,70%	0,00%	0,04%	0,28%	5,01%	0,37%	0,00%
Liquor	0,00%	2,80%	**70,92%**	0,00%	0,14%	0,03%	8,31%	0,02%	0,01%
Oedem	0,00%	0,00%	0,00%	**0,00%**	0,00%	0,00%	0,00%	0,00%	0,00%
Knochen	0,00%	0,00%	0,06%	0,00%	**94,87%**	3,05%	1,97%	0,00%	0,00%
Fett	0,00%	0,00%	0,00%	0,00%	1,27%	**67,69%**	1,02%	0,95%	0,02%
Grau	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	**0,00%**	0,00%	0,00%
Weiß	0,00%	3,94%	10,32%	0,00%	2,80%	28,50%	83,69%	**98,66%**	0,00%
Hintergrund	0,00%	0,00%	0,00%	0,00%	0,88%	0,44%	0,00%	0,00%	**99,97%**

Tabelle 7.15 Algorithmus dect_m1_3

dect_m2_3	STAT	Tumor	Liquor	Oedem	Knochen	Fett	Grau	Weiß	Hintergrund
STAT	**0,00%**	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
Tumor	0,00%	**93,83%**	13,22%	0,00%	0,03%	0,03%	3,11%	0,08%	0,00%
Liquor	0,00%	1,93%	**81,30%**	0,00%	0,04%	0,00%	6,47%	0,00%	0,00%
Oedem	0,00%	0,00%	0,00%	**0,00%**	0,00%	0,00%	0,00%	0,00%	0,00%
Knochen	0,00%	0,00%	0,00%	0,00%	**95,02%**	3,16%	1,02%	0,00%	0,00%
Fett	0,00%	0,00%	0,00%	0,00%	1,15%	**88,68%**	0,32%	0,00%	0,02%
Grau	0,00%	0,00%	0,13%	0,00%	1,17%	0,28%	**9,07%**	0,19%	0,00%
Weiß	0,00%	4,25%	5,35%	0,00%	1,66%	7,66%	79,95%	**99,73%**	0,00%
Hintergrund	0,00%	0,00%	0,00%	0,00%	0,92%	0,19%	0,06%	0,00%	**99,98%**

Tabelle 7.16 Algorithmus dect_m2_3

dect_m3_3	STAT	Tumor	Liquor	Oedem	Knochen	Fett	Grau	Weiß	Hintergrund
STAT	**0,00%**	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
Tumor	0,00%	**94,88%**	14,89%	0,00%	0,04%	0,08%	4,06%	0,22%	0,00%
Liquor	0,00%	1,31%	**79,17%**	0,00%	0,14%	0,08%	7,80%	0,00%	0,00%
Oedem	0,00%	0,00%	0,00%	**0,00%**	0,00%	0,00%	0,00%	0,00%	0,00%
Knochen	0,00%	0,00%	0,00%	0,00%	**95,85%**	2,19%	2,22%	0,00%	0,00%
Fett	0,00%	0,00%	0,00%	0,00%	1,29%	**87,48%**	0,70%	0,05%	0,00%
Grau	0,00%	0,00%	0,00%	0,00%	0,03%	0,00%	**0,63%**	0,02%	0,00%
Weiß	0,00%	3,81%	5,93%	0,00%	2,03%	9,74%	84,33%	**99,71%**	0,00%
Hintergrund	0,00%	0,00%	0,00%	0,00%	0,61%	0,42%	0,25%	0,00%	**100,00%**

Tabelle 7.17 Algorithmus dect_m3_3

dect_en_3	STAT	Tumor	Liquor	Oedem	Knochen	Fett	Grau	Weiß	Hintergrund
STAT	**0,00%**	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
Tumor	0,00%	**90,06%**	4,00%	0,00%	0,00%	0,03%	0,00%	0,05%	0,00%
Liquor	0,00%	8,89%	**94,26%**	0,00%	0,00%	0,00%	0,19%	0,00%	0,00%
Oedem	0,00%	0,00%	0,00%	**0,00%**	0,00%	0,00%	0,00%	0,00%	0,00%
Knochen	0,00%	0,00%	0,00%	0,00%	**98,75%**	1,19%	1,27%	0,00%	0,00%
Fett	0,00%	0,00%	0,00%	0,00%	0,60%	**95,70%**	0,00%	0,00%	0,00%
Grau	0,00%	0,79%	1,74%	0,00%	0,57%	0,97%	**75,19%**	4,66%	0,00%
Weiß	0,00%	0,26%	0,00%	0,00%	0,03%	2,11%	23,35%	**95,29%**	0,00%
Hintergrund	0,00%	0,00%	0,00%	0,00%	0,06%	0,00%	0,00%	0,00%	**100,00%**

Tabelle 7.18 Algorithmus dect_en_3

dect_en_5	STAT	Tumor	Liquor	Oedem	Knochen	Fett	Grau	Weiß	Hintergrund
STAT	**0,00%**	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
Tumor	0,00%	**93,43%**	4,77%	0,00%	0,00%	0,00%	0,00%	0,02%	0,00%
Liquor	0,00%	6,57%	**93,04%**	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
Oedem	0,00%	0,00%	0,00%	**0,00%**	0,00%	0,00%	0,00%	0,00%	0,00%
Knochen	0,00%	0,00%	0,06%	0,00%	**98,71%**	1,11%	0,25%	0,00%	0,00%
Fett	0,00%	0,00%	0,06%	0,00%	0,54%	**96,81%**	0,00%	0,00%	0,01%
Grau	0,00%	0,00%	1,87%	0,00%	0,74%	0,94%	**83,19%**	3,26%	0,00%
Weiß	0,00%	0,00%	0,19%	0,00%	0,01%	1,14%	16,56%	**96,72%**	0,00%
Hintergrund	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	**99,99%**

Tabelle 7.19 Algorithmus dect_en_5

dect_no_3	STAT	Tumor	Liquor	Oedem	Knochen	Fett	Grau	Weiß	Hintergrund
STAT	**0,00%**	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
Tumor	0,00%	**91,24%**	3,87%	0,00%	0,00%	0,00%	0,00%	0,05%	0,00%
Liquor	0,00%	7,66%	**94,91%**	0,00%	0,00%	0,00%	0,19%	0,00%	0,00%
Oedem	0,00%	0,00%	0,00%	**0,00%**	0,00%	0,00%	0,00%	0,00%	0,00%
Knochen	0,00%	0,00%	0,00%	0,00%	**98,76%**	1,39%	0,57%	0,00%	0,00%
Fett	0,00%	0,00%	0,00%	0,00%	0,58%	**94,78%**	0,00%	0,00%	0,00%
Grau	0,00%	0,74%	0,84%	0,00%	0,63%	1,19%	**73,98%**	3,74%	0,00%
Weiß	0,00%	0,35%	0,39%	0,00%	0,00%	2,64%	25,25%	**96,21%**	0,00%
Hintergrund	0,00%	0,00%	0,00%	0,00%	0,03%	0,00%	0,00%	0,00%	**100,00%**

Tabelle 7.20 Algorithmus dect_no_3

dect_no_5	STAT	Tumor	Liquor	Oedem	Knochen	Fett	Grau	Weiß	Hintergrund
STAT	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
Tumor	0,00%	91,68%	5,22%	0,00%	0,00%	0,00%	0,06%	0,00%	0,00%
Liquor	0,00%	8,23%	92,33%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
Oedem	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
Knochen	0,00%	0,00%	0,00%	0,00%	98,56%	1,05%	0,25%	0,00%	0,00%
Fett	0,00%	0,00%	0,32%	0,00%	0,60%	97,20%	0,00%	0,00%	0,00%
Grau	0,00%	0,04%	1,74%	0,00%	0,84%	1,05%	80,27%	1,87%	0,00%
Weiß	0,00%	0,04%	0,39%	0,00%	0,00%	0,69%	19,42%	98,13%	0,00%
Hintergrund	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	100,00%

Tabelle 7.21 Algorithmus dect_no_5

res_thresh	STAT	Tumor	Liquor	Oedem	Knochen	Fett	Grau	Weiß	Hintergrund
STAT	**0,00%**	100,00%	100,00%	0,00%	0,54%	94,14%	99,05%	100,00%	99,69%
Tumor	0,00%	**0,00%**	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
Liquor	0,00%	0,00%	**0,00%**	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
Oedem	0,00%	0,00%	0,00%	**0,00%**	0,00%	0,00%	0,00%	0,00%	0,00%
Knochen	0,00%	0,00%	0,00%	0,00%	**99,46%**	5,86%	0,95%	0,00%	0,31%
Fett	0,00%	0,00%	0,00%	0,00%	0,00%	**0,00%**	0,00%	0,00%	0,00%
Grau	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	**0,00%**	0,00%	0,00%
Weiß	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	**0,00%**	0,00%
Hintergrund	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	**0,00%**

Tabelle 7.22 Algorithmus res_thres

Fuzzy Resultatbilder

Pure	STAT	Tumor	Liquor	Oedem	Knochen	Fett	Grau	Weiß	Hintergrund
STAT	**0**	6	13	0	8	0	1	0	0
Tumor	0	**2134**	74	0	0	0	0	1	0
Liquor	0	144	**1447**	0	0	0	3	0	0
Oedem	0	0	0	**0**	0	0	0	0	0
Knochen	0	0	1	0	**6944**	40	4	0	0
Fett	0	0	2	0	37	**3500**	0	0	0
Grau	0	0	13	0	46	38	**1262**	110	0
Weiß	0	0	1	0	0	25	306	**5773**	0
Hintergrund	0	0	0	0	0	0	0	0	**27628**

Tabelle 7.23 Konfusionsmatrix der Klassensieger im Qualitätsmaß

Dominant	STA	Tumor	Liquor	Oedem	Knochen	Fett	Grau	Weiß	Hintergrund
STAT	**0**	0	0	0	0	0	0	0	0
Tumor	0	**2110**	73	0	0	0	1	0	0
Liquor	0	172	**1444**	0	0	0	0	0	0
Oedem	0	0	0	**0**	0	0	0	0	0
Knochen	0	0	0	0	**6934**	38	4	0	0
Fett	0	0	2	0	42	**3502**	0	0	0
Grau	0	1	26	0	59	38	**1265**	110	0
Weiß	0	1	6	0	0	25	306	**5774**	0
Hintergrund	0	0	0	0	0	0	0	0	**27628**

Tabelle 7.24 Konfusionsmatrix Verknüpfung mit einem dominanten Algorithmus

F_and	stat	Tumor	Liquor	Oedem	Knochen	Fett	Grau	Weiß	Hintergrund
STAT	**0**	214	420	0	46	167	106	200	0
Tumor	0	**2052**	56	0	0	1	0	3	0
Liquor	0	18	**1050**	0	0	0	0	0	0
Oedem	0	0	0	**0**	0	0	0	0	0
Knochen	0	0	0	0	**6918**	38	4	0	0
Fett	0	0	0	0	25	**3362**	0	0	0
Grau	0	0	24	0	46	19	**1263**	46	0
Weiß	0	0	1	0	0	16	203	**5635**	0
Hintergrund	0	0	0	0	0	0	0	0	**27628**

Tabelle 7.25 Konfusionsmatrix durch automatische Regelgenerierung

Visual	stat	Tumor	Liquor	Oedem	Knochen	Fett	Grau	Weiß	Hintergrund
STAT	**0**	214	419	0	20	5	106	200	0
Tumor	0	**2052**	56	0	0	1	0	3	0
Liquor	0	18	**1050**	0	0	0	0	0	0
Oedem	0	0	0	**0**	0	0	0	0	0
Knochen	0	0	0	0	**6997**	211	15	0	86
Fett	0	0	1	0	1	**3341**	0	0	0
Grau	0	0	24	0	17	22	**1252**	46	0
Weiß	0	0	1	0	0	23	203	**5635**	0
Hintergrund	0	0	0	0	0	0	0	0	**27542**

Tabelle 7.26 Konfusionsmatrix automatischer Regeln u.a. nach visuellen Kriterien

7.5 Beschreibung des Fuzzy Verknüpfungstools

Das Modul Fuzzy wurde als universell einsetzbare Unix-Applikation zur Verknüpfung von Daten auf Basis von Fuzzy Logik entwickelt. Es zeichnet sich für den Anwender durch einfache Handhabbarkeit und hohe Geschwindigkeit aus. Das Modul wurde im Rahmen des KAMEDIN Projekts erstellt und greift unter anderem auf dessen Programmbibliotheken zurück.

Das Modul besitzt lediglich die drei Schnittstellenfunktionen *Fuzzy_Howmany*, *Fuzzy_Input* und *Fuzzy_Output*.
Mit *Fuzzy_Howmany* läßt sich aus einer zugrundeliegenden Steuerdatei, deren Aufbau im weiteren Verlauf noch beschrieben wird, die Anzahl der Eingangs- und Ergebnisvariablen ermitteln. Mit *Fuzzy_Input* kann man dem Modul eine Liste von Eingangsvariablen übergeben, mit *Fuzzy_Output* erhält man eine Liste von scharfen Ergebnisvariablen zurück.

Ein kleines C-Programm, welches das Modul Fuzzy verwendet, könnte folgendermaßen aussehen:

```
#include <stdio.h>
#include <fuzzy.h>
#define  INPUTFILE  "myset.def"

main()
{
 double inlist[MAX], outlist[MAX]; /* Uebergabelisten definieren    */
 INT32 anzahl_in, anzahl_out;      /* Variablen für Anzahl Eigangs- */
                                   /* und Ergebnisvariablen         */
 header head;                      /* Kopf einer dynamischen        */
                                   /* Struktur erzeugen             */

 Fuzzy_Init_Header(head);          /* head initialiseren            */

 Fuzzy_Scanfile(INPUTFILE, &head); /* Aufbau eines dynamischen      */
                                   /* Speichermodells nach den      */
                                   /* Vorgaben der Steuerdatei      */
                                   /* myset.def                     */

 Fuzzy_Howmany(&head,&anzahl_in,&anzahl_out);
                                   /* Anzahl der Eingangs- und      */
                                   /* Ergebnisvariablen ermitteln   */

 ...                               /* Belegen der Uebergabeliste    */
                                   /* inlist mit Eingangsvariablen  */

 Fuzzy_Input(&head,inlist);        /* Einganvariablen verabeiten    */
 Fuzzy_Output(&head,outlist);      /* Resultatvariablen erzeugen    */

 ...                               /* Auswerten der in outlist      */
                                   /* enthaltenen Resultatvariablen */
 Fuzzy_Exit(head)                  /* Speicher freigeben            */
 return();                         /* Programmende                  */
}
```

Beispiel 7.2

Bei den Variablen *inlist* und *outlist* handelt es sich um zwei Arrays des Datentyps *double*.
In der Steuerdatei *myset.dat* sind Informationen enthalten, aus denen die Funktion *Fuzzy_Howmany* die Anzahl der Eingangs- und Ergebnisvariablen ermitteln kann.
Die beiden Arrays *inlist* und *outlist* müssen mindestens so groß gewählt werden, wie *Fuzzy_Howmany* für die beiden Variablen *anzahl_in* und *anzahl_out* ermitteln konnte.

Die Struktur *header* enthält alle zum Aufbau einer dynamischen Datenstruktur erforderlichen Variablen und Parameter. Sie muß einmalig zu Beginn des Programmes mit der Funktion *Fuzzy_Init_Header* initialisiert werden. Die genaue Funktionsweise wird bei der Besprechung der verwendeten Datenstrukturen erläutert werden.

Die Funktion *Fuzzy_Scanfile* interpretiert die Steuerdatei und baut anhand dynamischer Listen eine rechnerinterne Datenstruktur auf, die sämtliche Fuzzy Sets, Variablennamen und die komplette Regelbasis beinhaltet. Enthält die Steuerdatei den Befehl *finger*, so übernimmt *Fuzzy_Scanfile* zusätzlich die Funktion eines Kommandointerpreters, mit dessen Befehlssatz es möglich ist, sämtliche Vorgänge des Fuzzy Tools im Einzelschrittmodus zu simulieren.

Die Funktion *Fuzzy_Input* übergibt dem Fuzzy Modul ein Array von Eingangsvariablen und löst im Folgenden eine Reihe von Verarbeitungsschritten aus. Diese beinhalten u.a. das Ermitteln der Zugehörigkeitswerte der Eingangsvariablen anhand der Fuzzy Sets, den Inferenzprozeß für sämtliche Variablen und die Modifikation der Resultat Fuzzy Sets.

Die Funktion *Fuzzy_Output* ist für den Defuzzifizierungsvorgang zuständig und schreibt die scharfen Resultatvariblen in das Array *outlist*. Sie verwendet als Defuzzifizierungsstrategie die Schwerpunktmethode (COG).

Mit *Fuzzy_Exit* wird der verwendete Speicher wieder freigegben und sämtliche dynamischen Datenstrukturen entfernt.

Die Steuerdatei enthält alle weiteren, für Fuzzy Verknüpfungen notwendigen Daten wie:

- die Wahl des Inferenzoperators,
- die Definitionen der Fuzzy Sets,
- die Definition der Regelbasis,
- und weitere Kommandos.

7.5.1 Aufbau der Steuerdatei

Die Steuerdatei gliedert sich im Wesentlichen in zwei große Teile und zwar in die Definitionen der Fuzzy Sets, genauer gesagt in die Definitionen der verschiedenen linguistischen Variablen und der in ihnen enthaltenen Fuzzy Sets und in die Definition der Regelbasis. Jedes Fuzzy Set erhält einen Namen, anschließend können die Sets durch eine beliebig lange Folge von Koordinatenpaaren definiert werden. Die Verbindungen von einem Punkt zum nächsten erfolgen intern durch lineare Interpolation. Ein sinnvoller Erweiterungsvorschlag wäre, als weitere Möglichkeit der Berechnung von Zwischenwerten, eine Bezier-Approximation anzubieten.

Die Definition einer linguistischen Variablen und ihre Zugehörigkeitsfunktionen könnten in der Steuerdatei wie in Beispiel 7.3 aussehen. Abbildung 7.7 zeigt die graphische Darstellung einer textuell beschriebenen linguistischen Variablen.

```
FUZZY_VAR TEMPERATUR INPUT
FUZZY_SET kalt
   FUZZY_VAL  0    0.0
              10   1.0
              20   0.0
FUZZY_SET kühl
   FUZZY_VAL  10   0.0
              20   1.0
              30   1.0
              40   0.0
FUZZY_SET warm
   FUZZY_VAL  30   0.0
              40   1.0
              50   0.0
```

Beispiel 7.3

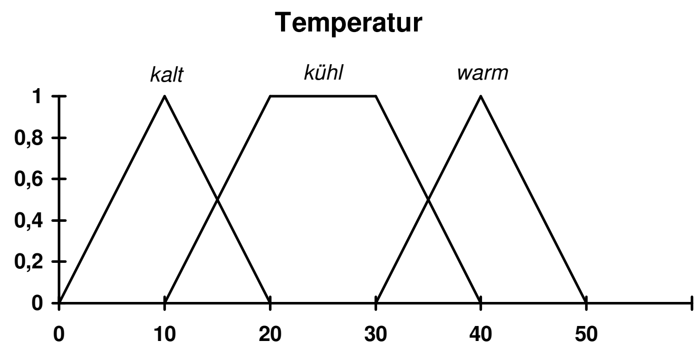


Abbildung 7.7 Graphische Repräsentation einer lingustischen Variablen und ihrer Fuzzy Sets

Zur Definition einer linguistischen Variablen sind die drei Befehle *FUZZY_VAR*, *FUZZY_SET* und *FUZZY_VAL* notwendig, dabei spielen Groß- und Kleinschreibung keine Rolle.
Eine vollständige Beschreibung aller momentan verfügbaren Befehle findet der geneigte Leser auf Seite 86 *ff*.

Der in Beispiel 7.3 verwendete Befehl *FUZZY_VAR* und der Parameter TEMPERATUR erzeugen eine linguistische Variable *TEMPERATUR*. Falls es sich bei der linguistischen Variable um eine Eingangsvariable handelt, so kann *FUZZY_VAR* mit einem zweiten optionalen Parameter *INPUT* versehen werden.

Nach der Variablendeklaration wird mit *FUZZY_SET* <TERM> wird der Name des ersten Fuzzy Sets festgelegt. Anschließend können nach dem Befehl *FUZZY_VAL* beliebig viele, nach Abszissenwerten aufsteigend geordnete Koordinatenpaare angegeben werden, durch die die Zugehörigkeitsfunktion definiert wird. Alle weiteren erforderlichen Terme können nach dem eben beschriebenen Muster erzeugt werden. Anschließend kann die nächste linguistische Variable nach dem selben Schema wie in Beispiel 7.3 definiert werden.

Im nächsten Schritt muß noch eine Regelbasis erzeugt werden, in der die Beziehungen der verschiedenen linguistischen Objekte untereinander geklärt werden. Ausschnitte einer Regelbasis wurden in den vorherigen Kapiteln schon des öfteren vorgestellt. Das Augenmerk soll hier jedoch auf den prinzipiellen Aufbau einer Regel und deren Einfluß auf andere Objekte gelenkt werden.

Linguistische Regeln ermitteln Zugehörigkeitswerte aus den Fuzzy Sets linguistischer Variablen, verknüpfen diese mit den entsprechenden Fuzzy Operatoren und modifizieren damit die entsprechenden Terme weiterer linguistischer Variablen. Im Vorliegenden Fuzzy Modul werden die Fuzzy Regeln in umgekehrt polnischer Notation (UPN) formuliert.

If ***TEMPERATUR kühl TEMPERATUR warm AND*** **Then** ***RESULT warm RESULT kühl***

Abbildung 7.8

In Abbildung 7.8 handelt es sich bei den in Großbuchstaben geschriebenen Wörtern um linguistische Variablen. Direkt dahinter befinden sich die in Kleinbuchstaben geschriebenen Fuzzy Terms. Aus der Kombination von Linguistischer Variable und linguistischem Term kann ein Zugehörigkeitswert μ ermittelt werden, sofern der entsprechende Term ein Attribut der linguistischen Bezugsvariablen darstellt. Die verschiedenen Zugehörigkeitsfunktionen können durch einen Operator verknüpft werden und modifizieren im betrachteten Beispiel die beiden Terme *kühl* und *warm* der linguistischen Variable *RESULT*. Die Regeln können dabei beliebig kompliziert werden, und dementsprechend viele Operatoren enthalten. Eine Verknüpfung von vier linguistischen Variablen kann in UPN folgendermaßen geschrieben werden:

If *T kalt* *T kühl* *AND* *T warm* *T heiß* *AND* *AND* Then *Result warm* *Result kühl*

Abbildung 7.9

Für die linguistische Variable TEMPERATUR wurde die Abkürzung T verwendet.
Die Pfeile geben die Verknüpfungshierarchie an. UPN Notation läßt sich immer als Binärbaum darstellen. Für ein tiefergehendes Verständnis von UPN sei allerdings an dieser Stelle auf die gängige Fachliteratur verwiesen.

Da für Fuzzy Operatoren die T-Norm Bedingungen gelten, gilt auch insbesondere das Kommutativgesetz, d.h. sofern in einer Regel immer der selbe Operator verwendet wird, lassen sich die verschiedenen Ausdrücke (die unterstrichenen, bzw. umrahmten Elemente) beliebig untereinander vertauschen.

Sollten mehrere Regeln existieren, die ein und denselben Term einer linguistischen Variablen modifizieren, so wird mittels des Inferenzoperators aus den verschiedenen Zugehörigkeitswerten ein endgültiger Zugehörigkeitswert gebildet. Die Default Einstellung verwendet hier das arithmetische Mittel (ARITH), es kann allerdings jeder der zur Verfügung stehenden Operatoren als Inferenzoperator eingesetzt werden. In der Steuerdatei kann der Inferenzoperator durch Hinzufügen einer Zeile „OPERATOR <Inferenzoperator>" vor die Regelbasis individuell ausgewählt werden.

Tabelle 7.27 enthält alle momentan implementierten Verknüpfungsfunktionen:

Funktionsname	*Bedeutung*	*Funktionsname*	*Bedeutung*
AND	Minimum	AP	Arithmetic product
OR	Maximum	AS	Arithmetic sum
DP	Drastic product	HP	Hamacher product
DS	Drastic sum	HS	Hamacher sum
BD	Bounded difference	GAMMA	Gamma operator
BS	Bounded sum	ARITH	Arithmethic middle
EP	Einstein product		
ES	Einstein sum		

Tabelle 7.27

Es sei an dieser Stelle ausdrücklich betont, daß das arithmethische Mittel nicht den T-Norm Bedingungen genügt und damit keinen Fuzzy Operator darstellt. Es kann als Inferenzoperator verwendet werden, sollte aber - wenn es denn doch als Operator eingesetzt wird - pro Fuzzy Regel nur ein einziges Mal verwendet werden.

7.5.2 Kurzübersicht aller Schlüsselbefehle einer Steuerdatei

7.5.2.1 Befehle zum Erzeugen von linguistischen Objekten

FUZZY_VAR	Erzeugen einer neuen linguistischen Variablen
FUZZY_SET	Erzeugen eines neuen Fuzzy Terms
FUZZY_VAL	Definieren einer Zugehörigkeitsfunktion

7.5.2.2 Befehle, die für die Regelerzeugung und Bearbeitung zuständig sind.

IF	Wählt die zu betrachteten Terme aus und verknüpft diese ggf.
THEN	modifiziert Ergebnis Fuzzy Sets mit dem aus der If Anweisugng ermittelten Zugehörigkeitswert.
OPERATOR	Wählt den für die Regelabarbeitung notwendigen Operator aus

7.5.2.3 Befehle für den Einzelschrittmodus

FINGER	Schaltet in den Kommandointerpretermodus um. Standardeingabe ist die Tastatur, Standardausgabe der Bildschirm
QUIT	Beendet den Kommandointerpretermodus
SHOW	Schreibt Informationen zu den Fuzzy Sets auf die Standardausgabe
SWITCH	Schaltet auf die aktuelle linguistische Variable um
MGET	Ermittelt die Zugehörigkeitswerte für alle Terme der aktuellen linguistischen Variablen
NEXT	Sorgt für Inferenz und Defuzzifizierung der Ergebnisvariablen

7.5.2.4 Weitere Befehle

ECHO	Gibt einen Text auf der Standardausgabe aus
#	Kommentar , nach dem Kommentarzeichen muß ein Leerzeichen folgen. Das Kommentarzeichen bezieht sich auf die aktuelle Zeile
LOAD	Lädt eine weitere Steuerdatei
SAVE	Speichert alle Linguistischen Variablen im Format der Steuerdatei
END	Beendet das Fuzzy Tool

7.5.3 Beschreibung der Befehle einer Steuerdatei

Befehle die mit der Klammerung (Beta) versehen sind, sind funktionstüchtig und können als zusätzliche Features betrachtet werden. Sie wurden allerdings nicht eingehend gestestet, da sie im Rahmen der hier vorliegenden Arbeit lediglich als zusätzliche Debugmöglichkeit bei der Programmimplementierung verwendet wurden.

7.5.3.1 FUZZY_VAR

SYNTAX:

FUZZY_VAR *name* [INPUT|OUTPUT]

BESCHREIBUNG:

Der Befehl FUZZY_VAR dient der Erzeugung einer neuen Linguistischen Variablen. Sollte der Variablenname *name* schon vergeben sein, so wird keine neue linguistische Variable erzeugt. Die aktuelle Bezugsvariable ist die Variable mit dem Namen *name*.
Mit den optionalen Parametern INPUT oder OUTPUT kann man festlegen, ob es sich um eine Eingangs- oder Ergebnisvariable handelt.

VERWEIS:

FUZZY_SET, FUZZY_VAL

7.5.3.2 FUZZY_SET

SYNTAX:

FUZZY_SET *name*

BESCHREIBUNG:

Der Befehl FUZZY_SET dient der Erzeugung eines neuen Terms einer linguistischen Variablen. Sollte der Variablename des Terms in der aktuellen linguistischen Variablen schon existieren, so wird kein neuer Term erzeugt.

VERWEIS:

FUZZY_VAR, FUZZY_VAL

7.5.3.3 FUZZY_VAL

SYNTAX:

FUZZY_VAL *zahl zahl* [*zahl zahl*] ...

BESCHREIBUNG:

Der Befehl FUZZY_VAL dient der Erzeugung von Punktkoordinaten, die eine Zugehörigkeitsfunktion definieren. Er bezieht sich immer auf das aktuelle Fuzzy Set einer linguistsichen Variablen und erweitert die betreffende Liste mit Koordinatenpaaren. Eine Koordinate besteht immer aus einem Punktepaar (x,f(x)). *Zahl* kann eine beliebige vorzeichenbehaftete reelle Zahl im Gleitpunktformat sein. Die x - Koordinaten müssen in aufsteigender Reihenfolge angegeben werden. Da die Anzahl der verwendeten Koordinatenpaare beliebig groß werden kann, kann die Zugehörigkeitsfunktion der Vorgabefunktion ebenfalls beliebig genau angenähert werden. Da Fuzzy Operatoren nur im Intervall [0,1] definiert sind, sollten auch alle Zugehörigkeitswerte f(x) Elemente dieses Intervalls sein. Die Anzahl der hinter FUZZY_VAL angegebenen Zahlenwerte muß gerade sein, ansonsten wird das Einlesen der Steuerdatei mit einer Fehlermeldung beendet.

VERWEIS:

FUZZY_VAR, FUZZY_SET

7.5.3.4 IF ... THEN ...

SYNTAX:

IF *extended_expression* THEN *expression* [*expression*] ...

BESCHREIBUNG:

Expression bezeichnet immer die Kombination von linguistischer Variablen und einem zu dieser Variablen gehördenen Fuzzy Term. *Extended_expression* ist ein aus mehreren *expressions* und Operatoren bestehender, in umgekehrt polnischer Notation geschriebener Ausdruck. Aus verschieden IF ... THEN ... Regeln wird die Regelbasis erzeugt.

VERWEIS:
OPERATOR

7.5.3.5 OPERATOR

SYNTAX:

OPERATOR *operator*

BESCHREIBUNG:

Der Befehl OPERATOR erlaubt eine freie Wahl des für Fuzzy Verknüpfungen notwendigen Inferenzoperators. Der Befehl muß in der Steuerdatei vor der Regelbasis stehen. Wird er nicht angegeben, so wird als Defaulteinstellung die Funktion ARITH verwendet. Als *operator* kommen derzeit folgende Operatoren in Frage: AND, OR, DP, DS, BD, BS, EP, ES, AP, AS, HP,HS, GAMMA, ARITH. Tabelle 7.27 enthält eine Kurzbeschreibung der hier aufgeführten Operatoren. Der Gamma Operator ist eigentlich ein variabler Operator. Je nach Wahl des Parameters Gamma nimmt der Operator mehr die Eigenschaften einer UND-Verknüpfung bzw. einer ODER-Verknüpfung an. Beim derzeitigen Implementierungsstand ist eine freie Wahl des Parameters Gamma noch nicht möglich. Die Voreinstellung ist Gamma = 0.8.

VERWEIS:
IF ... THEN ...

7.5.3.6 FINGER (BETA)

SYNTAX:

FINGER

BESCHREIBUNG:

Nach Auftreten des Befehls FINGER, schaltet das Fuzzy Modul in den Kommandointerpretermodus um. Standardeingabe ist in diesem Falle die Tastatur, Standardausgabe der Bildschirm. Es können anschließend alle Kommandos ohne Einschränkung weiterverwendet werden. Der Kommandointerpreter ließt zeilenweise ein, die Bestätigung erfolgt mit der <RETURN> Taste.

VERWEIS:
EXIT

7.5.3.7 QUIT (BETA)

SYNTAX:

EXIT

BESCHREIBUNG:

Mit dem Befehl EXIT wird der Kommandointerpretermodus wieder verlassen. Die Abarbeitung der alten Steuerdatei wird direkt nach dem letzten FINGER Befehl fortgesetzt.

VERWEIS: FINGER

7.5.3.8 SHOW (BETA)

SYNTAX:

SHOW [ALLNAME|ALL|NAME|VAR|SET|VAL|NUM]

BESCHREIBUNG:

Die Funktion SHOW zeigt in Verbindung mit einem der oben aufgeführten Parameter bestimmte Informationen zum aktuellen Stand der Fuzzy Sets an. Dies sind im einzelnen:

ALLNAME: Gibt eine Liste aller momentan existierenden Namen linguistischer Variablen auf der Standardausgabe aus.

ALL: Gibt zu allen existierenden linguistischen Variablen deren Namen aus, die einzelnen Termnamen und zu allen Terms sämtliche Koordinaten der Zugehörigkeitsfunktionen.

NAME: Gibt den Namen der aktuellen linguistischen Variablen aus.

VAR: Wie SET, gibt jedoch noch zusätzlich den Namen der aktuellen linguistischen Variablen aus.

SET: Gibt die Namen der verschiedenen Fuzzy Terme der aktuellen linguistischen Variablen und deren Koordinaten aus.

VAL: Gibt alle Koordinaten der aktuellen Fuzzy Variablen aus.

NUM: Gibt einen internen Zählindex für die aktuelle linguistische Variable aus.

VERWEIS:

SWITCH, MGET, NEXT

7.5.3.9 SWITCH (Beta)

SYNTAX:

SWITCH *name*

BESCHREIBUNG:

Wechselt von der alten auf eine neue aktuelle linguistische Variable *name*.
Wurde für *name* der Name einer linguistischen Variablen angegeben, der nicht existiert, so erscheint auf der Standardausgabe eine Fehlermeldung und die alte linguistische Variable bleibt weiterhin aktuelle Bezugsvariable.

VERWEIS:

SHOW, MGET

7.5.3.10 MGET (Beta)

SYNTAX:

MGET *zahl*

BESCHREIBUNG:

Der Befehl MGET berechnet anhand der Zugehörigkeitsfunktionen der aktuellen, linguistischen Variablen und dem scharfen Wert *zahl* die Zugehörigkeitswerte μ für die verschiedenen Fuzzy Sets und gibt diese auf der Standardausgabe aus. *Zahl* kann eine beliebige Zahl aus dem reellen Zahlenbereich sein.

VERWEIS:

SHOW

7.5.3.11 NEXT (Beta)

SYNTAX:

NEXT, MGET

BESCHREIBUNG:

Der Befehl NEXT führt anhand der Fuzzy Regelbasis den Inferenzprozeß durch. Anschließend wird nach der COG Methode defuzzifiziert und die Ergebnisse in die Standardausgabe geschrieben.

VERWEIS:

SHOW

7.5.3.12 ECHO (Beta)

SYNTAX:

ECHO *string*

BESCHREIBUNG:

Der Befehl ECHO schreibt eine beliebige Zeichenfolge auf die Standardausgabe. *String* repräsentiert diese Zeichenfolge. Die Zeichenfolge darf nicht länger als eine Zeile sein.

VERWEIS:

#

7.5.3.13

SYNTAX:

#

BESCHREIBUNG:

Die Steuerdatei läßt auch Kommentare zu, die mit dem Kommando # eingeleitet werden. Beim derzeitigen Stand der Implementierung muß dem Kommentarzeichen ein Leerzeichen folgen, ansonsten wird das Kommentarzeichen nicht als solches interpretiert.

VERWEIS: ECHO

7.5.3.14 LOAD (Beta)

SYNTAX:

LOAD *commandname*

BESCHREIBUNG:

Der Befehl LOAD unterbricht die Bearbeitung der aktuellen Steuerdatei und bearbeitet eine weitere Steuerdatei mit dem Namen *commandname.* Nachdem die Bearbeitung von *commandname* beendet wurde, wird die Bearbeitung der alten Steuerdatei nach dem LOAD Befehl fortgesetzt. Mit Hilfe des LOAD Befehls ist beispielsweise eine Untergliederung der Steuerdatei in einen Teil für die Variablendeklarationen und einen weiteren Teil für die Regelbasis möglich.

VERWEIS:

SAVE

7.5.3.15 SAVE (Beta)

SYNTAX:

SAVE *commandname*

BESCHREIBUNG:

Der Befehl SAVE erzeugt nach den Vorgaben der internen dynamischen Datenstrukturen eine Steuerdatei und speichert diese unter dem Dateinamen *commandname* im aktuellen Arbeitsverzeichnis ab. Die derzeitige Implementierung sieht nur die Speicherung der Variableninformationen vor, die Regelbasis kann momentan noch nicht mit abgespeichert werden.

VERWEIS: LOAD

7.5.3.16 END (Beta)

SYNTAX:

END

BESCHREIBUNG:

Mit dem Befehl END wird die Bearbeitung der Steuerdatei beendet und das Fuzzy Tool terminiert.

VERWEIS:

7.5.4 Beschreibung der Schnittstellenfunktionen des Moduls FUZZY

7.5.4.1 Fuzzy_Init_Header

NAME:

Fuzzy_Init_Header - Headerstruktur initialisieren

SYNOPSIS:

INT32 Fuzzy_Init_Header(
header *head,
MEMORY *mem)

BESCHREIBUNG:

Die Funktion Fuzzy_Init_Header initialisiert eine Header Struktur und bereitet diese für die Funktion Fuzzy_Scanfile vor. Die Initialisierung erfolgt derart, daß sämtliche Arrayzeiger auf NULL gesetzt werden und die für die Regelbasis notwendigen Zählvariablen definierte Werte erhalten.

RÜCKGABEWERT:

VERWEIS:

Fuzzy_Scanfile

7.5.4.2 Fuzzy_Scanfile

NAME:

Fuzzy_Scanfile - Einlesen und Bearbeiten der Steuerdatei.

SYNOPSIS:

```
INT32 Fuzzy_Scanfile(
char *filename,
header *head)
```

BESCHREIBUNG:

Die Funktion Fuzzy_Scanfile baut mit den Vorgaben der Steuerdatei im Speicher ein dynamisches Speichermodell auf, und initialisiert dieses mit Werten. Dazu gehören die für Fuzzyverknüpfungen notwendigen Speicherstrukturen zur Aufnahme von Zugehörigkeitswerten- und funktionen und verschiedene Stacks zur Speicherung und Abarbeitung der Regelbasis. Darüberhinaus beinhaltet Fuzzy_Scanfile einen vollständigen Kommandointerpreter dessen Funktionalität schon beschrieben wurde. Die Verwendung von Fuzzy_Scanfile ist zwingend notwendig, ansonsten sind logische Verknüpfungen auf Fuzzy Basis nicht möglich.

RÜCKGABEWERT:

Im Fehlerfall wird ein Wert ungleich Null zurückgeliefert.

VERWEIS:

Fuzzy_Init_Header

7.5.4.3 Fuzzy_Howmany

NAME:

Fuzzy_Howmany - Anzahl Eingangs- und Ergebnisvariablen ermitteln

SYNOPSIS:

```
void Fuzzy_Howmany(header *head,
INT32 *in,
INT32 *out)
```

BESCHREIBUNG:

Die Funktion Fuzzy_Howmany ermittelt die Anzahl der Eingangs- und Ergebnisvariablen aus den Befehlsparametern INPUT und OUTPUT der zugrundeliegenden Steuerdatei.

RÜCKGABEWERT:
keiner

VERWEIS:
Fuzzy_Input, Fuzzy_Output

7.5.4.4 Fuzzy_Input

NAME:
Fuzzy_Input - Schnittstelle zur Übergabe von scharfen Eingangswerte an das Fuzzy Tool

SYNOPSIS:

```
INT32 Fuzzy_Input(
header *head,
double *in)
```

BESCHREIBUNG:

Die Funktion Fuzzy_Input stellt eine Schnittstelle der aufrufenden Applikation zum Modul Fuzzy dar. Mit ihr ist es möglich, ein Array vom Typ *double* mit scharfen Eingangswerten dem Fuzzy Modul zur Weiterverarbeitung zu übergeben.
Das Array sollte mindestens so groß gewählt werden, daß es die Anzahl der mit der Funktion Fuzzy_Howmany ermittelten Eingangsvariablen fassen kann. Sollte das Array größer sein, so werden nur die ersten n Eingangvariablen gelesen. Sollte

das Array kleiner sein als erforderlich, so wird für die restlichen Variablen automatisch der Zahlenwert Null angenommen. Die Übergabe an die entsprechenden Fuzzy Sets geschieht in der Reihenfolge, wie sie in der Regelbasis definiert sind.

RÜCKGABEWERT:

Liefert die Anzahl der Eingangsvariablen.

VERWEIS: Fuzzy_Output

7.5.4.5 Fuzzy_Output

NAME:

Fuzzy_Output - Schnittstelle, zur Übergabe der defuzzifizierten Ergebnisvariablen an die aufrufende Applikation.

SYNOPSIS:

```
INT32 Fuzzy_Output(
header *head,
double *in)
```

BESCHREIBUNG:

Die Funktion Fuzzy_Output stellt eine Schnittstelle vom Fuzzy Modul zur aufrufenden Applikation dar.
Beim Aufruf der Funktion belegt das Fuzzy Modul das Übergabearray vom Typ *double* mit scharfen, defuzzifizierten Ergebnisvariablen. Der Funktion Fuzzy_Output sollte ein Aufruf von Fuzzy_Input vorausgehen, ansonsten kann das Array unvorhersehbare Werte zurückliefern.
Das Array sollte mindestens so groß gewählt werden, daß es die Anzahl der mit der Funktion Fuzzy_Howmany ermittelten Resultatvariablen fassen kann. Sollte das Array größer sein, so werden nur die ersten *n* Resultatvariablen beschrieben. Sollte das Array kleiner sein als erforderlich, so können die restlichen Variablen nicht zurückgeschrieben werden und der Vorgang wird abgebrochen. Die Übergabe an das Array geschieht in der Reihenfolge der in Regelbasis definierten Resultatvariablen.

RÜCKGABEWERT:

Liefert die Anzahl der Ergebnisvariablen zurück.

VERWEIS:

Fuzzy_Input

7.5.5 Erläuterung der für das Modul Fuzzy wichtigen Datenstrukturen und Programmteile

Dieser Abschnitt soll dem Leser einen Einblick in die Struktur und Arbeitsweise des Moduls Fuzzy geben. Bedingt durch die Größe des Programms und vieler programmtechnischer Details, soll hier lediglich auf die wesentliche Programmteile und deren prinzipiellen Aufbau eingegangen werden.

Wie schon des öfteren erwähnt, verwendet das Modul Fuzzy eine sogenannte Steuerdatei als Grundlage. Diese Steurdatei enthält den gesamten *Bauplan* der verschiedenen Fuzzy Sets. Dieser Bauplan ist notwendig und kann für jedes gestellte Regelungs- oder Steuerproblem anders aussehen. Das heißt, die Zahl der Eingangs- und Ergebnisvariablen ist variabel, genauso wie die Anzahl linguistischer Variablen und deren Zugehörigkeitsfunktionen. Eine programmtechnische Lösung der Datengenerierung stellt daher die Verwendung dynamischer Listen dar, die zur Programmlaufzeit erstellt, verändert, und eliminiert werden können. Mittels dynamischer Listen wird folgende Speicherstruktur erzeugt:

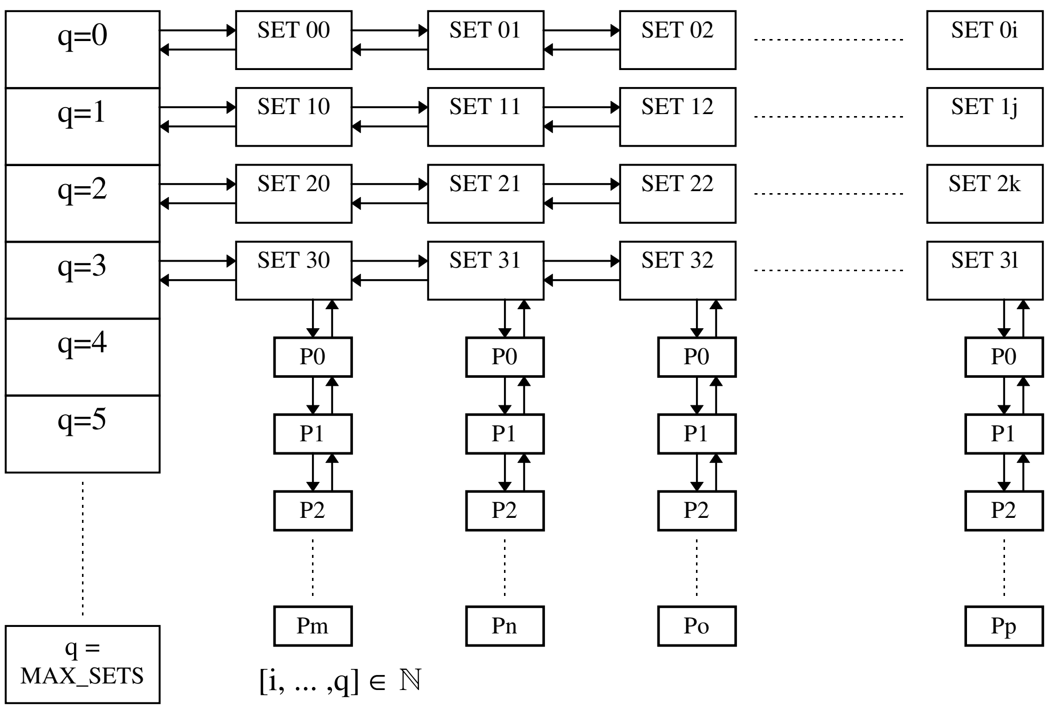


Abbildung 7.10 Speicherstruktur and Aufbau der Fuzzy Sets

Die maximale Anzahl linguistischer Variablen q ist begrenzt und beläuft sich derzeit auf 50. Durch ändern der Variablen MAX_SETS in fuzzy.h und anschließendem Neukompilieren kann sie allerdings erhöht oder vermindert werden.

Abbildung 7.10 stellt die interne Speicherstruktur dar, die beim Aufruf der Funktion *Fuzzy_Scanfile* erzeugt wird. Dabei ist die Länge der jeweils betrachteten doppelt verketteten Listen abhängig von der in der Steuerdatei definierten Anzahl und Größe der

Zugehörigkeitsfunktionen. In Abbildung 7.10 symbolisieren die verschiedenen Rechtecke jeweils unterschiedliche Datenstrukturen. Bei Rechtecken mit der Bezeichnung *q* handelt es sich um ein Pointerarray vom Typ *fsglob*, welches in der alles umfassenden Datenstruktur *header* erklärt ist.

```
typedef struct
       { char       **commandlist; /* Befehlsliste fuer Interpreter     */
         char          *fuzzyname; /* Obermenge alle Fuzzy-sets         */
         fsglob             **set; /* Array für linguistische Variablen  */
         pstack    *rule[MAXRULE]; /* Regelbasis                        */
         int              newrule; /* naechste Regel                    */
         double          (**op)(); /* Referenzen auf Operatorfunktionen  */
         int                  now; /* Index fuer aktuelles Fuzzy Set    */
         int           inf_op_num; /* Inferenzoperator festlegen        */
         ELEMENT    *elpt0,*elpt1; /* Zum erzeugen neuer Sets           */
         MEMORY              *mem; /* Memory Handle für Kamedinanbindung */
       } header;
```

Die Datenstruktur header umfaßt weiterhin die Datenstruktur *pstack*, welche die speicherinterne Repräsentation der Regelbasis, sowie ein Array mit Referenzen auf Fuzzy Operatoren und weitere Variablen enthält. Der Memory Handle MEMORY ist aus Konsistenzgründen zur Einbindung des Fuzzy Moduls in das System KAMEDIN notwendig.

Je nach Anzahl der in der Steuerdatei verwendeten linguistischen Variablen enthält das betreffende Arrayelement einen Pointer auf die Datenstruktur *fsglob* oder den Nullpointer. In der Datenstruktur *fsglob* wird der Name der linguistischen Variablen festgelegt, außerdem werden hier, die gesamte linguistische Variable betreffenden Zwischenwerte - wie die scharfe Eingangsvariable und der Gesamtschwerpunkt - gespeichert. *Fsglob* enthält weiterhin den Datentyp *Element*, mit dem sich eine allgemeine, doppelt verkettete Liste aufbauen läßt.

```
typedef struct
       { char           *set_name;
         int                input; /* Inputvariable ?                    */
         double           gewicht; /* zur Zeit nicht benutzt             */
         double             sharp; /* scharfe Eingangszahl               */
         ELEMENT              *fg, /* Liste der Fuzzy Sets               */
                    *el0,      /* aktueller Pointer einer Fuzzy Variablen */
                    *el1,      /* aktueller Pointer einer Fuzzy Wertemenge*/
                    *result;   /* Hier steht die Ergebnisfunktion drin    */
         position cog; /* Reserviert für Schwerpukt bei Defuzzyfizierung */
       } fsglob;        /* globale  Parameter für ein FuzzySet (n terms) */
```

Bevor nun der Inhalt der verketteten Liste in der Struktur *fsglob* geklärt wird, soll zunächst kurz auf den Datentyp *ELEMENT* eingegangen werden. Der Datentyp *ELEMENT* präsentiert sich mit den für doppelt verkettete Listen üblichen Referenzen auf das nächste, bzw. vorherige zu verkettende Datenelement.

Die verkettete Liste enthält zur für den Datenteil einen typenlosen Zeiger **data*, der zur Aufnahme verschiedenartiger Datenstrukturen verwendet werden soll. Momentan werden die

Listenelemente im Fuzzy Modul zur Verkettung der Fuzzy Terme, der die Zugehörigkeitsfunktion definierenden Koordinatenliste und zur Realisierung eines Ringbuffers zum Einlesen und Auswerten der Steuerdatei verwendet. Die Datenstruktur der doppelt verketteten Liste sieht folgendermaßen aus:

```
struct
      verkettete_liste
{
   void                       *data;
   struct verkettete_liste    *next;
   struct verkettete_liste    *prev;
};

typedef struct
      verkettete_liste ELEMENT;
```

Der Vorteil der Verwendung eines typenlosen Zeigers auf einen unabhängigen Datenteil ist, daß man trotz des vielseitigen Einsatzes der Listenelemente immer auf die selben bekannten Listenfunktionen zum Erzeugen, Eliminieren und An- bzw. Umhängen von Elementen zurückgreifen kann. Der Nachteil ist, daß beim Einsatz typenlosen Zeiger vom Compiler keine Kompatibilitätsprüfungen bzgl. des verwendeten Datentyps durchgeführt werden. Die Listenfunktionen sind in der Datei *liste.c*, bzw. *liste.h* definiert bzw. deklariert. Da die Funktionalität den aus der Fachliteratur bekannten Listenfunktionen entspricht, wird an dieser Stelle auf die zahlreich vorhanden Grundlagenbücher zur C Programmierung verwiesen.

In der Struktur *fsglob* wird eine verkettete Liste vom Typ Element aufgebaut. Ihr Datenteil enthält einen Zeiger auf die Struktur *linguistic_set*. Die Struktur *linguistic_set* enthält im Wesentlichen eine Struktur *zbv* mit Variablen zur Speicherung von Zwischenergebnissen und den eigentlichen Termnamen. Darüber hinaus kann jedes Strukturelement des Typs *linguistisc_set* eine weitere Liste von Koordinaten enthalten, durch die eine Zugehörigkeitsfunktion repräsentiert wird. Die Liste wird ebenfalls mit Elementen des Typs ELEMENT erzeugt, deren Zeiger auf die Struktur *position* verweisen. *Position* enthält lediglich zwei Variablen des Datentyps *double*, die eine x- und y-Koordinate aufnehmen können.

7.5.6 Struktur und Aufbau der Regelbasis

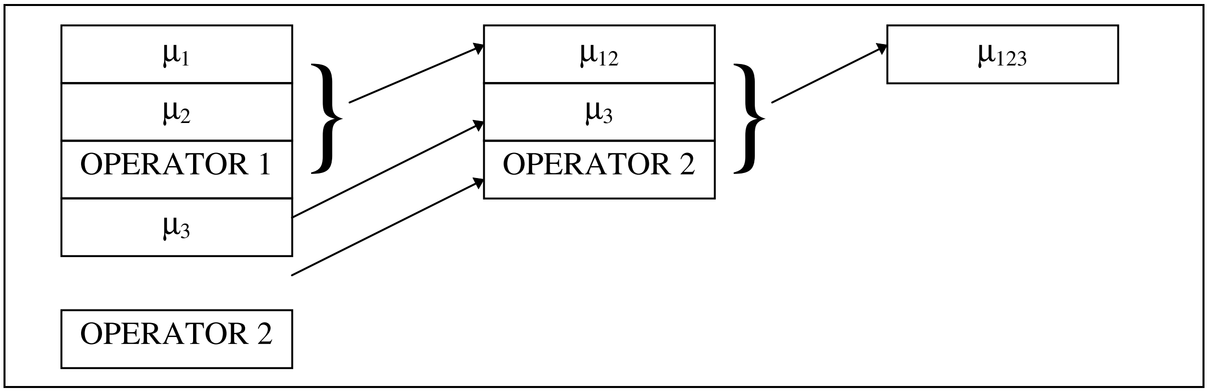


Abbildung 7.11 Speicherung und Abarbeitung von Fuzzy Regeln in einem Stack

Die Regelbasis wird in einer Art Stack abgelegt. Der Prinzipielle Aufbau geht aus Abbildung 7.11 hervor. Ein Stackelement besteht aus einer Referenz auf eine Struktur, die einen Zeiger auf eine weitere Struktur *zbv* in *liguistic_set* enthält, bzw. auf den entsprechenden Operator *head* weist.

Die Stackinhalte können mit den Funktionen *ppush* und *ppop* auf den Stack gelegt, bzw. abgeholt und mit den Operatoren verknüpft werden. Wurde eine Regel vollständig abgearbeitet, so muß der aktuelle Stackzeiger auf das oberste Stackelement zeigen, ansonsten wurde die Regel nicht nach UPN Konventionen eingegeben und das Programm gibt die Fehlermeldung „*error in rulebase*“ aus. Jede Regel wird in einem gesonderten Stack abgelegt, die Abarbeitung der Regeln erfolgt mit Hilfe eines gesonderten Berechnungsstacks. Da ein Stack lediglich aus Referenzen auf Zugehörigkeitswerte und Operatoren besteht, benötigt die interne Darstellung der Regeln sehr wenig Speicher und die Abarbeitung ist sehr schnell, da sich der Berechnungsaufwand bei der Verarbeitung linear verhält. Die Berechnung erfolgt durch die Funktion *fuzzy_inference*.

7.5.7 Funktionsweise des Kommandointerpreters

Das Einlesen der Steuerdatei erfolgt zeilenweise durch die Funktion *fuzzy_rb_push*. Diese trennt die einzelnen Strings (das Leerzeichen ist u.a. Trennzeichen) und schreibt die einzelnen Zeichenketten in einen Ringpuffer. Nach der Übergabe können durch die Funktion *fuzzy_rb_pop* die einzelnen Zeichenketten auf Schlüsselworte untersucht werden. In einer Case Struktur werden dann die auf ein Schlüsselwort zu aktivierenden Folgeaktionen ausgeführt, wie z.B. das Einlesen von weiteren Parametern, das Aufbauen von Speicherstrukturen usw.

Da das Einlesen der Steuerdatei zeilenweise erfolgt, ist es sehr einfach einen Kommandointerpretermodus zu realisieren. Dazu muß lediglich der Eingabekanal des Steuerdateifiles auf die Standardeingabe gelegt werden, dann werden sodaß alle weiteren Kommandos zeilenweise, nach Drücken der Return Taste von der Tastatur gelesen werden.

Der zum Einlesen der Steuerdatei notwendige Ringpuffer besteht wieder aus Listenelementen des Typs *ELEMENT*, die mittels der Funktion *fuzzy_init_ringbuf* zu einem Ring verkettet werden. Zugriff auf den Ringpuffer hat man über die Struktur *ringbuf*, die zusätzlich noch die beiden zur Synchronisation erforderlichen Zeiger *low* und *high* enthält.

7.5.8 Der Defuzzifizierungsvorgang

Durch Aufrufen der Funktion *fuzzy_cog* wird der Schwerpunkt eines Fuzzy Terms berechnet. Die Funktion *fuzzy_cog* arbeitet nach der Schwerpunktmethode und ruft die Funktionen *fuzzy_calc_cog* und *fuzzy_plane* auf, die den Schwerpunkt der zugrundeliegenden Zugehörigkeitsfunktionen unter Berücksichtigung einer flächenbezogenen Gewichtung berechnen. Alternativ steht eine weitere Funktion fuzzy_calc_mom (Mean of Maximum) zum Defuzzifizieren zur Verfügung. Die Änderungen müssen allerdings im Programm selbst vorgenommen werden. Es empfiehlt sich jedoch sehr, die voreingestellte COG Methode zu verwenden, da sie sowohl qualitativ als auch quantitativ bessere Ergebnisse als die Mean of maximum Methode liefert.

Die Berechnung eines Schwerpunktes aus beliebig vielen Punkten P_n erfolgt sehr einfach durch Vektoraddition und anschließender Normalisierung. Die Flächenberechnung erfolgt durch Determinantenbestimmung unter Beachtung der Kurvenorientierung.
Die Funktion *Fuzzy_Output* enthält den vollständigen Fuzzifizierungsvorgang für sämtliche linguistische Ergebnisvariablen und stellt gleichzeitig die Schnittstelle zur Einbindung in andere Fuzzy Module dar.

7.6 Vorstellung der Regelbasen zur Segementierung aus Resultatsequenzen

7.6.1 Regelbasis der Verknüpfungsstrategie Pure

```
if ALG7 tumor   THEN tumor big
if ALG8 liquor  THEN liquor big
if ALG7 knochen THEN knochen big
if ALG9 fett    THEN fett big
if ALG9 grau    THEN grau big
if ALG9 weiss   THEN weiss big
if ALG9 back    THEN back big
```

7.6.2 Regelbasis der Verknüpfungsstrategie Dominant

```
# Statistische Regelerzeugung

if ALG9 tumor   THEN tumor mid
if ALG9 liquor  THEN liquor mid
if ALG9 knochen THEN knochen mid
if ALG9 fett    THEN fett mid
if ALG9 grau    THEN grau mid
if ALG9 weiss   THEN weiss mid
if ALG9 back    THEN back big # trifft zu 100 % zu !

# und verknuepfte Regeln werden besser bewertet

if ALG7 tumor   ALG8 tumor   AND THEN tumor   big
if ALG8 liquor  ALG7 liquor  AND THEN liquor  big
if ALG7 knochen ALG9 knochen AND THEN knochen big
if ALG9 fett    ALG7 fett    AND THEN fett    big
if ALG9 grau    ALG7 grau    AND THEN grau    big
if ALG9 weiss   ALG7 weiss   AND THEN weiss   big
```

7.6.3 Regelbasis der Verknüpfungsstrategie F_and

```
# Regeln mit 1 % Abweichung

if ALG9 back THEN back big
if ALG10 knochen ALG9 knochen AND THEN knochen big
if ALG9 knochen ALG9 back AND THEN back big
if ALG4 fett ALG8 fett AND THEN fett big
if ALG4 grau ALG8 fett AND THEN fett big
if ALG4 weiss ALG6 tumor AND THEN tumor big
if ALG1 tumor ALG6 grau AND ALG8 grau ALG7 grau AND AND THEN grau big
if ALG1 grau ALG6 grau AND ALG8 grau ALG7 grau AND AND THEN grau big
if ALG2 weiss ALG6 grau AND ALG8 weiss ALG7 weiss AND AND THEN weiss big
if ALG1 grau ALG6 weiss AND ALG8 grau ALG7 grau AND AND ALG9 grau AND THEN grau big
if ALG2 weiss ALG6 weiss AND ALG8 weiss ALG7 weiss AND AND ALG9 weiss AND THEN weiss big

# 2. Iteration (2.5 % Abweichung )

if ALG2 tumor ALG6 tumor AND THEN tumor mid
if ALG1 weiss ALG9 liquor AND THEN liquor mid
if ALG1 weiss ALG8 fett AND THEN fett mid
if ALG3 liquor ALG4 liquor AND ALG9 liquor AND THEN liquor mid
if ALG1 liquor ALG9 weiss AND ALG6 liquor AND THEN liquor mid
if ALG2 fett ALG6 tumor AND ALG9 tumor AND THEN tumor mid
if ALG1 fett ALG8 weiss AND ALG6 fett AND THEN fett mid
if ALG1 weiss ALG8 weiss AND ALG6 fett AND THEN fett mid
if ALG8 liquor ALG2 weiss AND ALG6 grau ALG7 grau AND AND THEN grau mid
if ALG1 fett ALG8 fett AND ALG6 fett ALG7 fett AND AND THEN fett mid
if ALG1 weiss ALG9 weiss AND ALG6 liquor ALG7 liquor AND AND THEN liquor mid
if ALG8 weiss ALG2 grau AND ALG6 weiss ALG7 grau AND AND ALG9 grau AND THEN grau mid


# von Hand erzeugte Regeln
if ALG3 grau ALG4 grau AND ALG5 grau AND THEN grau min
```

```
if ALG4 tumor ALG5 tumor AND ALG6 tumor AND THEN tumor min
if ALG7 grau ALG9 grau AND THEN grau mid
if ALG7 weiss ALG9 weiss AND THEN weiss mid
```

7.6.4 Regelbasis der Verknüpfungsstrategie Visual

```
if ALG9 back THEN back big
if ALG10 knochen then knochen big
if ALG9 fett then fett big

# Regeln mit 1 % Abweichung


if ALG4 weiss ALG6 tumor AND THEN tumor big
if ALG1 tumor ALG6 grau AND ALG8 grau ALG7 grau AND AND THEN grau big
if ALG1 grau ALG6 grau AND ALG8 grau ALG7 grau AND AND THEN grau big
if ALG2 weiss ALG6 grau AND ALG8 weiss ALG7 weiss AND AND THEN weiss big
if ALG1 grau ALG6 weiss AND ALG8 grau ALG7 grau AND AND ALG9 grau AND THEN grau big
if ALG2 weiss ALG6 weiss AND ALG8 weiss ALG7 weiss AND AND ALG9 weiss AND THEN weiss big

# 2. Iteration (2.5 % Abweichung )

if ALG2 tumor ALG6 tumor AND THEN tumor mid
if ALG1 weiss ALG9 liquor AND THEN liquor mid
if ALG1 weiss ALG8 fett AND THEN fett mid
if ALG3 liquor ALG4 liquor AND ALG9 liquor AND THEN liquor mid
if ALG1 liquor ALG9 weiss AND ALG6 liquor AND THEN liquor mid
if ALG2 fett ALG6 tumor AND ALG9 tumor AND THEN tumor mid
if ALG1 fett ALG8 weiss AND ALG6 fett AND THEN fett mid
if ALG1 weiss ALG8 weiss AND ALG6 fett AND THEN fett mid
if ALG8 liquor ALG2 weiss AND ALG6 grau ALG7 grau AND AND THEN grau mid
if ALG1 fett ALG8 fett AND ALG6 fett ALG7 fett AND AND THEN fett mid
if ALG1 weiss ALG9 weiss AND ALG6 liquor ALG7 liquor AND AND THEN liquor mid
if ALG8 weiss ALG2 grau AND ALG6 weiss ALG7 grau AND AND ALG9 grau AND THEN grau mid



if ALG3 grau ALG4 grau AND ALG5 grau AND THEN grau min
if ALG4 tumor ALG5 tumor AND ALG6 tumor AND THEN tumor min


if ALG7 grau ALG9 grau AND THEN grau mid
if ALG7 weiss ALG9 weiss AND THEN weiss mid
```

Weitere Regelsätze und Fuzzy Set Definitionen sind aus Gründen der Übersichtlichkeit auf der beiliegenden Diskette zu finden..

7.7 Literaturverzeichnis

[TRAE95] Traeger, Dirk H., 1993: „**Einführung in die Fuzzy Logik**“, 2. Aufl., Teubner Verlag

[WEI94] Lin, Wei 1994: „**EinBeitrag zur kartographischen Mustererkennung mittels Methoden der künstlichen Intelligenz**“. ISBN 3 7696 9463 5

[KRU93] Nauck, Klawonn, Kruse: „**Fuzzy Sets, Fuzzy Controllers, and Neuronal Networks**“, Abteilung Informatik, Technische Universität Braunschweig.

[HABER95] Haberäcker, Peter: „**Praxis der Digitalen Bildverarbeitung und Mustererkennung“,** Hanser 1995

[KRU95] Kruse, Nauck, Klawonn: „**Neuronale Fuzzy Systeme**“, Spektrum der Wissenschaft, Juni 1995

[GOTT93] Siegfried Gottwald: „**Fuzzy Sets and Fuzzy Logic**“, Vieweg 1993

[GOTT89] Gottwald, Kreiser, Stelzner: „**Mehrwertige Logik**“, Akademie Verlag 1989

[MAY93] Mayer, Mechler, Schlindwein, Wolke: **„Fuzzy Logic**“, Addison-Weseley 1993

[SHANN] Shannon, „**The Entropy Impurity Function**“

[QUIN95] Quinlan, J.R.:“**C4.5: Programs for machine Learning**“ http://yoda.cis.temple.edu:8080/UGAIWWW/lectures/C45

[QUIN88] Quinlain, J.R.: „**Decision Trees and Multi-Valued Attributes**“, Clarendon Press S. 305-318

[BÖHME93] Böhme: „**Fuzzy-Logik**“, Springer Verlag 1993

[BAND93] Bandemer/Gottwald, „**Einführung in Fuzzy-Methoden**“, Akademie Verlag 1993

[FOLEY91] Foley, van Dam, Feiner, Hughes, „**Computer Graphics: Principles and Practice**“, Addison-Weseley 1991

[WEBB95] „**The Physics of medical imaging**“ Medical Science Series, Institute of Physics, Publishing 1995

[ZIM91] Zimmermann, H.-J.: „**Fuzzy Set Theory - and Ist Applications**“. Kluwer, Boston 1991

[LEHN92] J. Lehn / H. Wegmann: „**Einführung in die Statistik“**, Teubner Stuttgart 1992

[ZADEH65] Zadeh, L.A.: „**Fuzzy sets. Information and Control 1965“**, 8, 338-353

[KOHON84] T. Kohonen: „**Self-Organization and Associative Memory**“. Berlin - Heidelberg - New York: Springer 1984

Danksagung

Zum Schluß danke ich allen, die zum Gelingen meiner Diplomarbeit beigetragen haben. Mein besonderer Dank gilt Herrn Dipl.-Ing. Christoph Busch, der mir immer mit einer Tasse Tee und guten Ratschlägen zur Seite stand.