

J.-W. Goethe Universität Frankfurt  
Fachbereich Informatik  
Lehrstuhl für praktische Informatik VSFT  
Dr.R.Brause

Automatische Bilderkennung

Vorlesung WS 1986/87

(C) Johann Wolfgang Goethe-Universität,  
Institut für Informatik

Dieses Skript ist nur für die internen Verwendung der Universität nach §52a Urheberrechtsgesetz bestimmt. Es enthält copyright-geschütztes Material und darf deshalb nicht ausserhalb der Hochschule benutzt werden. Eine Kopie ist untersagt.

# Automatische Bilderkennung

## Inhalt

### Maschinelle Bilderkennung

#### **1.0 Einleitung**

- 1.1 Die Bilderfassung
- 1.2 Hardware der Bilderkennung

#### **2.0 Bildverbesserung**

- 2.1 Der Mittelwertoperator
- 2.2 Die Fouriertransformation
- 2.3 Die Grauwerttransformation
- 2.4 Die Medianfilterung

#### **3.0 Kantendetektion**

- 3.1 lokale Gradientenoperatoren
  - ROBERTs Operator
  - SOBEL Operator
  - Kirsch Operator
- 3.2 Die Güte der Kantenoperatoren
- 3.3 Kantenrelaxation
- 3.4 Mexikaner-Hut Operator

#### **4.0 Regionenorientierte Segmentierung**

- 4.1 lokale Segmentierung
- 4.2 globale Segmentierung
- 4.3 Split-and-merge

#### **5.0 Textur**

- 5.1 Strukturelle Modelle
  - Form-Grammatiken
  - Baum-Grammatiken
  - Feld-Grammatiken
- 5.2 Stochastische Modelle
  - 5.2.1 Feature-Bildung
    - Features der Fouriertransformierten
    - Features mit SGLD-Matrizen
  - 5.2.2 Selbstlernen der Texturarten
    - Die Gradientenmethode
    - Der Robbins-Monroe Algorithmus

## Biologische Bilderkennung.

### **6.0 Das menschliche Sehsystem**

#### 6.1 Das menschliche Auge

Das optische System

Das Retina-System

#### 6.2 Der optische Chiasmus

#### 6.3 Neurologische Kodierung und Informationsübertragung

6.3.1 Kodierung der Information

6.3.2 Übertragung der Information

6.3.3 Das Modell von McCulloch und Pitts

#### 6.4 Die Empfindlichkeit des Sehsystems

#### 6.5 Rezeptive Felder

### **7.0 Stereo-Sehen**

#### 7.1 Gewinnung der Stereo-Bilder

#### 7.2 Der Stereo-Algorithmus

### **8.0 Eigenschaften des linearen Matrixmodells für Neuronennetze**

#### 8.1 Assoziative Speicherung

Fehlerhafte Daten

#### 8.2 Die auto-assoziative Matrix

Datenfehler

Hardwarefehler

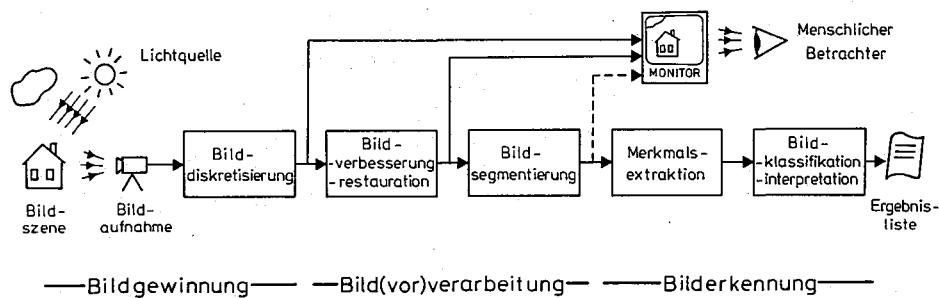
#### 8.3 Der Novelty-Filter

## 1.0 Einleitung

Eine automatische Erkennung und Verarbeitung von Bildern ist in vielen Bereichen unserer technischen Industriegesellschaft zu einer Herausforderung geworden. In Bereichen, in denen die zu erkennenden Bildinhalte gut definiert sind und die damit verbundenen Operationen einfach sind und sehr oft wiederholt werden müssen (monotone Arbeiten), ist eine automatische Bildverarbeitung schon weit verbreitet. Beispiele dafür sind das Erkennen von Schrift auf Schecks und Belegen (OCR-Leser), das Sortieren von maschinengeschriebenen Briefen bei der Bundespost, das Testen von Flüssigkeitskristallanzeigen oder das Sortieren von Werkstücken auf dem Fließband.

Auf der einen Seite gibt es Anwendungsgrenzen der Bilderkennung bezüglich der Komplexität des Problems nach unten hin. Welches sind diese? Nun, wenn das Problem einfach und standardisiert ist, daß für die Lösung besser einfache, physikalische Systeme mit robusten, simplen und preiswerten Sensoren verwendet werden können. Beispiel dafür ist das berührungslose Zählen von Flaschen auf einem Fließband. Eine komplexe Bildverarbeitungsanlage, die eine Flasche erkennt und dann einen Zähler inkrementiert ist sicher falsch angelegtes Geld; hier reicht eine simple Lichtschranke oder ein einfacher optischer Annäherungsschalter vollkommen aus. Diese sind dabei nicht nur billiger, sondern verschmutzen auch weniger leicht, brauchen keine besondere Ausleuchtung, haben eine geringere Ausfallwahrscheinlichkeit etc.

Auf der anderen Seite gibt es aber auch viele Anwendungen, die noch im Versuchsstadium sind, weil das "natürliche" Bildmaterial sehr schwierig zu verarbeiten ist. Das "Erkennen" der Bildinhalte ist dabei meist ein mehrstufiger Prozeß, der über Bilderfassung, Bildverbesserung und Bildsegmentierung bis zum Wiedererkennen von Einzelinformation (Bildteile) und zur Bildinterpretation führt.

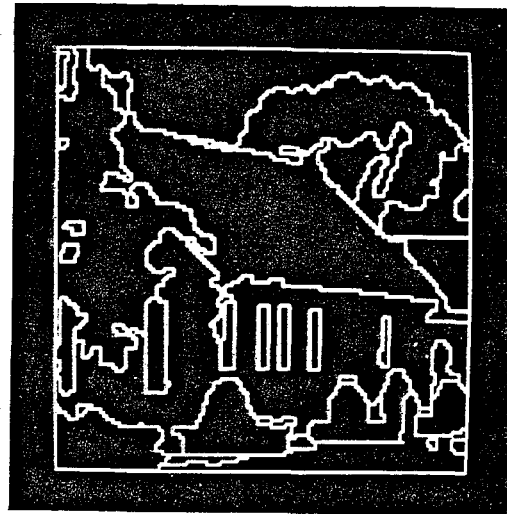


Stufen der Bildverarbeitung und Bilderkennung - Übersicht.

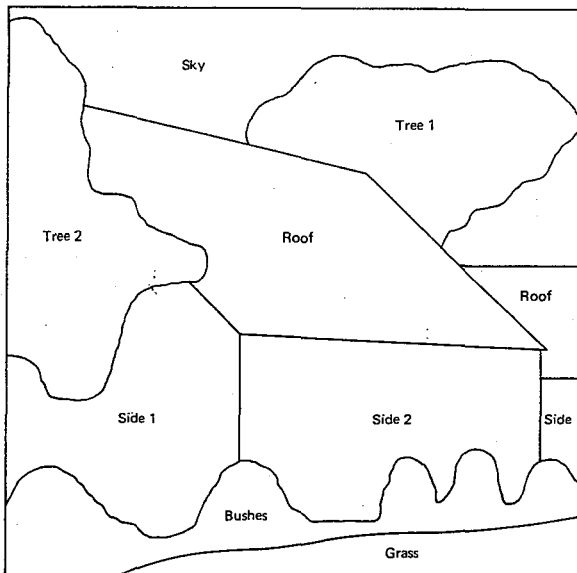
In Abb.1.0b ist die Bildreihe eines solchen Erkennungsprozesses zu sehen.



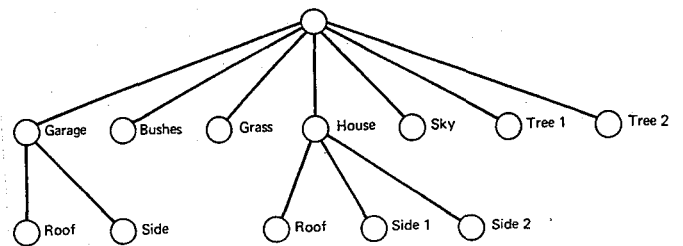
(a)



(b)



(d)



Dabei stoßen allerdings die traditionellen von-Neumann Computer als Bildverarbeitungsmaschinen bei bestimmten Problemen an die Grenzen ihrer Leistungsfähigkeit. Dies sind insbesondere Probleme, bei denen

- Bildmuster wiedererkannt werden müssen (Pattern Recognition)
- Ähnlichkeiten zwischen Bildteilen erkannt werden sollen
- verwaschene, gestörte Bilder vorliegen

Dies sind Leistungen, die für das menschliche Sehsystem kaum eine Schwierigkeit darstellen. Deshalb wird im zweiten Abschnitt kurz das menschliche Sehsystem vorgestellt und vor dem Hintergrund der zuvor erörterten, traditionellen Algorithmen der heutige Wissensstand zur menschlichen Sehmechanismus beschrieben.

Wie sehen nun die typischen Maschinen aus zur Bildverarbeitung?

## 1.1 Die Bildaufnahme

Bei der Bildaufnahme durchläuft die Bildinformation folgende Stufen:

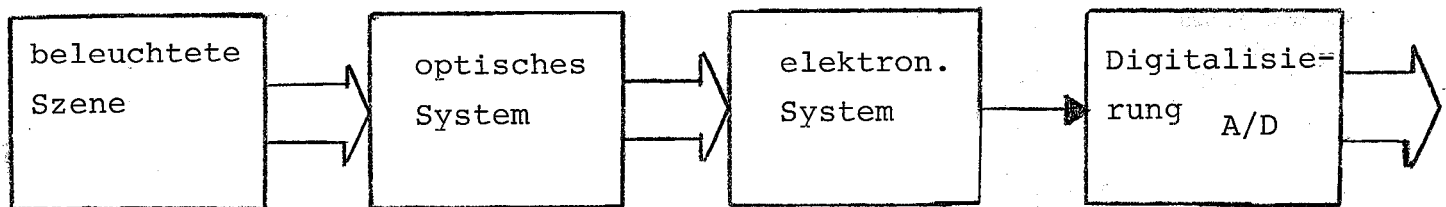


Abb.1.1.1 Stufen der Bildaufnahme

Dabei erfährt die Bildinformation verschiedene Transformationen und Verformungen, je nach Stufe. Betrachten wir dazu die folgende Abbildung.

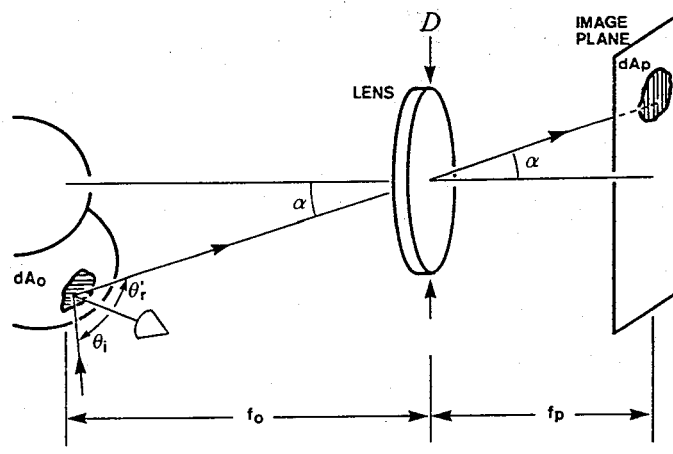


Abb. 1.1.2 Geometrie der Bildaufnahme

Die Form auf der runden Oberfläche wird durch die Optik der Kamera mit all ihren Abbildungsfehlern (Verzeichnungen, Achromat. Aberration, etc.) auf die plane Oberfläche der elektrischen Sensorelemente projiziert. Die Originalfigur auf der runden Oberfläche ist dabei natürlich in der Projektion nicht mehr erhalten. Dazu kommt die unterschiedliche Empfindlichkeit des Sensorelements für die bei der Beleuchtung verwendeten

Lichtfrequenzen.  
Charakteristika.

Abb. 1.1.3

zeigt

die

verschiedenen

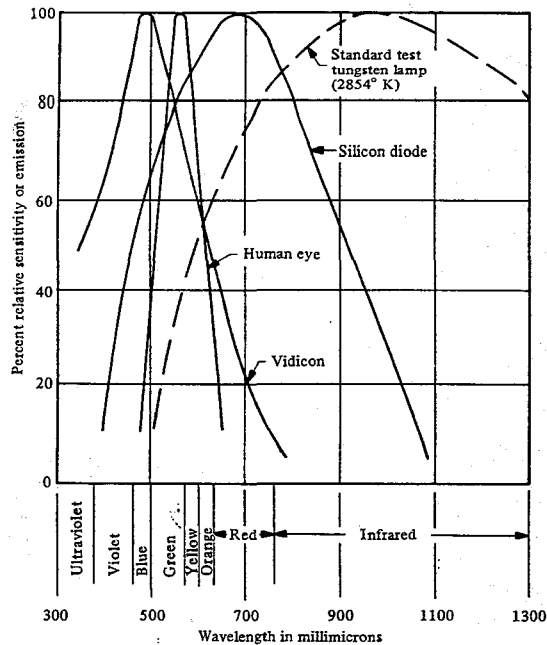


Abb. 1.1.3 spektrale Empfindlichkeiten der Bildsensoren

In den heutigen Fernsehkameras befinden sich meist Vidiconröhren, die geringere Rotempfindlichkeit besitzen als das menschliche Auge. Allerdings werden mit fortschreitender Auflösungstechnik auch als Silizium-Produkte die CCD-Elemente (Charge-Coupled-Device) eingesetzt, die sehr klein, robust und preiswert sind, aber z.Z. noch geringere Auflösung besitzen. Um die starke Rotempfindlichkeit zu kompensieren, werden die CCD Oberflächen meist so beschichtet, daß außer einer Reflexminderung auch eine Blaufilterwirkung entsteht.

Bei den eigentlichen Messungen wird die Kamera vorher mit bekanntem, genormtem Bildmaterial geeicht.

## 1.2 Hardware der Bildverarbeitung

Die Hardwarearchitektur von Bildverarbeitungsanlagen ist sehr unterschiedlich; je nach Aufgabenstellung variieren Datenpfade, Zwischenspeicher (RAM), Verarbeitungseinheiten (CPU) und die Ein- und Ausgabeperipherie. In Abb. 1.2 sind drei Beispiele gezeigt.

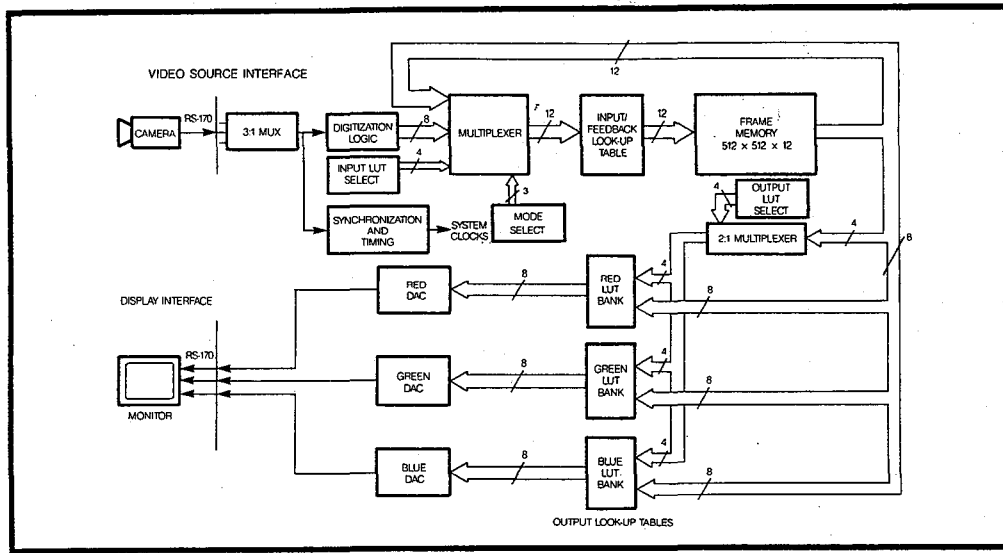


Abb. 1.2a Bildverarbeitung mit Rückkopplung, z.B. für Differenzbilder

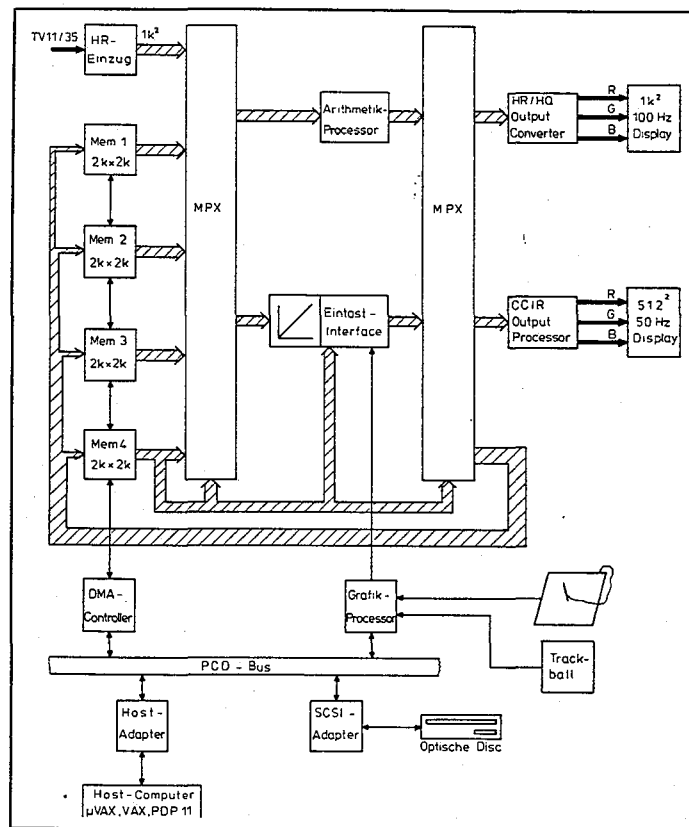
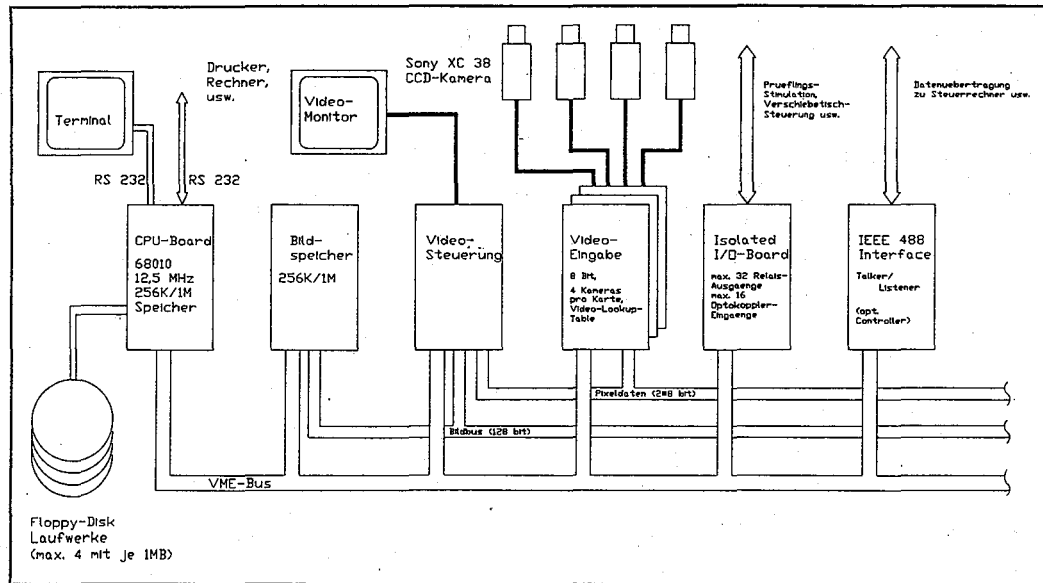


Abb. 1.2b Bildverarbeitungssystem mit großem Speicher 2048X2048X8 Bit und Umschaltmöglichkeit





*Blockschaltbild des Display-Test-Systems*

Trotz unterschiedlicher Architektur sind aber gemeinsame, wichtige Merkmale vorhanden:

Es gibt eine **Aufnahme** der Bilder mit Fernsehkameras,  
 eine **Digitalisierung**,  
 einen **Speicher** für die digitalisierten Rohbilder,  
 eine **Bearbeitungseinheit** (z.B. Floating-Point-Prozessor),  
 Möglichkeiten, das bearbeitete Bild **abzuspeichern**  
 (Massenspeicher wie Winchester und Laser-disk)  
**auszugeben** (Plotter, Monitore)  
 und per Hand **Änderungen** einzugeben (Digitalisierungstablett).

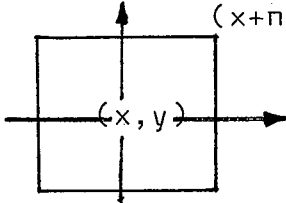
## 2.0 Bildverbesserung

Bevor ein Bild auf typische Merkmale untersucht werden kann, ist es meist sinnvoll, bildverfälschende Merkmale wie Störungen (Rauschen), Überlagerungen, Aufnahmefehler und dergleichen wegzufiltern.

Die Verfahren zur Filterung sind meist unabhängig von der betrachteten Stelle im Bild (**ortsinvariant**); sie lassen sich untergliedern in **lineare Filter** (z.B. Fouriertransformation) und in **nicht-lineare Filter** (z.B. Grauwerttransformation, Medianfilterung). Es gibt aber auch ortsvariante Verfahren (z.B. Subtraktion des Hintergrunds, signaladaptive Verfahren), die aber nicht näher betrachtet werden sollen. Ebenso sollen die Verfahren zur **Restauration** (Wienerfilterung, inverse Filterung, algebraische Verfahren), bei denen man die statistische Proportionen der Störquellen kennen muß, unberücksichtigt bleiben.

### 2.1 Der Mittelwertoperator

Ist ein Bild mit hochfrequenten Störungen überlagert (z.B. Rauschen), so lassen sich diese Intensitätsschwankungen durch Bildung eines Mittelwerts ausgleichen. Der Intensitätswert  $f(x,y)$  wird zu einem Mittelwert transformiert, indem alle Intensitäten der Bildpunkte um den betrachteten Punkt  $(x,y)$  herum im quadratischen Fenster der Seitenlänge  $2n+1$  summiert und durch die Zahl aller Punkte des Fensters geteilt werden:


$$f'(x,y) = \frac{1}{(2n+1)^2} \sum_{i=-n}^n \sum_{j=-n}^n f(x+i,y+j)$$

Dieses Verfahren muß für alle Bildpunkte  $(x,y), x=0..N-1, y=0..N-1$ , des gesamten Bildes durchgeführt werden.

Der Nachteil des Verfahrens liegt darin, daß bei der Mittelwertbildung feinere Strukturen, z.B. Kanten, neben den Störungen ebenfalls unterdrückt werden. Ein Unterscheiden der beiden Fälle setzt aber bereits ein inhaltliches Erkennen des Bildes voraus, was aber erst **nach** der Vorverarbeitung erfolgen soll.

Eine Alternative zu der Mittelwertbildung liegt in der kantenerhaltenden Medianfilterung aus Abschnitt 2.4.

### 2.2 Die Fouriertransformation

Ist ein Bild gegeben, dessen Kontraste verschärft oder bei dem hochfrequente Störungen (Rauschen) unterdrückt werden sollen, so wird gern eine frequenzabhängige, nichtlokale Filterung verwendet (Digitale Filterung).

Betrachten wir zunächst nur den eindimensionalen Fall, d.h. wir halten eine Variable (z.B. die Y-Achse) konstant und betrachten die Intensität  $f(x)$  entlang der x-Achse.

$f(x)$  läßt sich unter der hinreichenden Bedingung

$$\int_{-\infty}^{+\infty} |f(x)| dx < \infty$$

die bei physikalischen Systemen in der Regel erfüllt ist, als Überlagerung von Sinusfunktionen verschiedener Amplitude  $F$  und

Frequenz u darstellen:

$$f(x) = \int_{-\infty}^{+\infty} F(u) e^{i 2\pi u x} du \quad (2.2a)$$

Dabei wurde die Eulersche Formel verwendet

$$A e^{i\omega x} = A(\cos(\omega x) + i \sin(\omega x))$$

mit der Kreisfrequenz  $\omega = 2\pi u$

Die Amplitude  $F(u)$  errechnet sich dabei zu

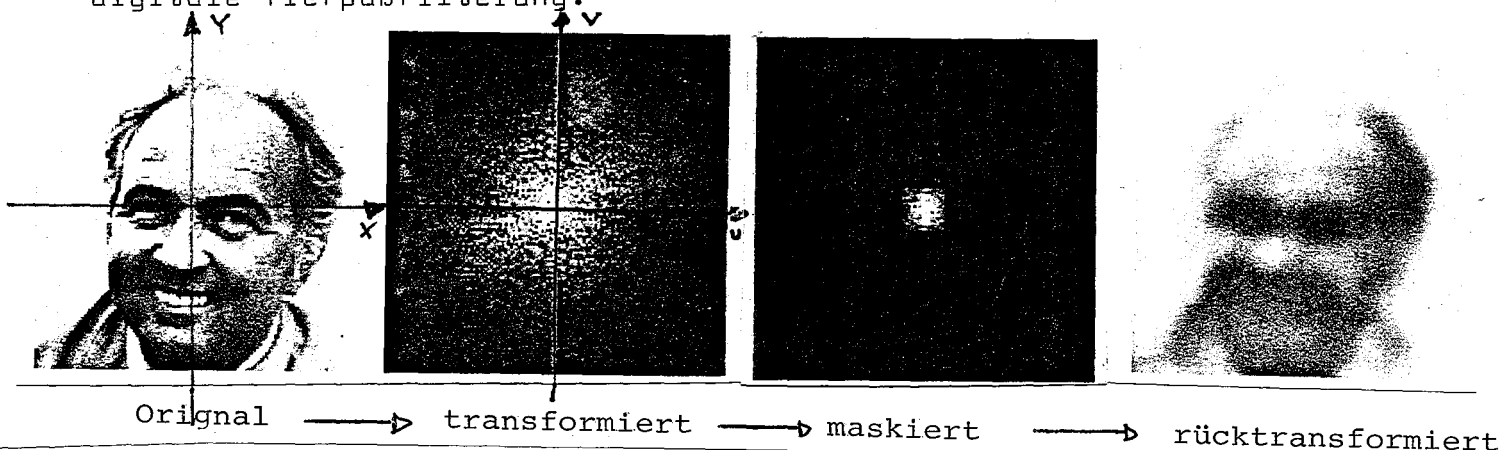
$$F(u) = \int_{-\infty}^{+\infty} f(x) e^{-i 2\pi u x} dx \quad (2.2b)$$

Die Fouriertransformation (2.2b) und ihre Inverse, die Rücktransformation (2.2.a), lassen sich auch analog zweidimensional formulieren. Für den diskreten Fall eines digitalisierten Bildes mit endlicher Auflösung von  $M \times N$  Punkten ist dies

$$F(u,v) = \frac{1}{NM} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(x,y) e^{-i 2\pi (u m/M + v n/N)}$$

An der Formel (2.2.a) bemerken wir eine wichtige Eigenschaft des fouriertransformierten Bildes :

Halten wir  $x$  konstant, so charakterisieren große Werte von  $u$  die Anteile der hohen Frequenzen in der Überlagerung. Die Amplitude dieser hohen Frequenzen sind im fouriertransformierten Bild mit  $F(u,v)$  bei großen Werten von  $u$  und  $v$  zu finden, also an den Rändern des Bildes. Hochfrequente Störungen eines Bildes lassen sich somit unterdrücken, indem das Originalbild einer Fouriertransformation unterworfen wird, die unerwünschten Frequenzanteile im transformierten Bild null gesetzt werden und das Bild rücktransformiert wird. Abbildung 2.2 zeigt eine solche digitale Tiefpaßfilterung.



Im Allgemeinen benötigt man etwa 50% der Koeffizienten  $F(u,v)$ , um ein befriedigendes Bild zu rekonstruieren, wobei mindestens 7Bit/Pixel Auflösung verwendet werden sollte. Man sieht, daß bei der Unterdrückung von Störungen auch sehr

leicht wichtige Kontraste und Bilddetails verloren gehen. Deshalb werden zur Unterdrückung von Störungen gern andere Methoden verwendet.

Die Fouriertransformation ist auf Bildverarbeitungsanlagen sehr weit verbreitet. Deshalb soll nun der Algorithmus einer schnellen Version der Fouriertransformation vorgestellt werden.

### Die Fast-Fouriertransformation (FFT)

Die zweidimensionale Fouriertransformation läßt sich durch zweimaliges Ausführen der eindimensionalen Fouriertransformation ersetzen; einmal bezüglich y und einmal bezüglich x:

$$f(x,y) \leftrightarrow \underset{y}{F}(x,v) \leftrightarrow \underset{x}{F}(u,v)$$

Damit reicht es aus, nur die eindimensionale Fouriertransformation zu betrachten. Hierbei wird bei der i.A. komplexe Funktion  $f(x)$  für alle N Werte von  $F(k)$ ,  $k=0..N-1$ , die Summe

$$F(k) = \sum_{m=0}^{N-1} f(m) e^{-i \frac{2\pi}{N} m k}$$

mit M komplexen Additionen und Multiplikationen gebildet. Dies sind insgesamt  $N \cdot M$  Operationen. Bei großem  $N \approx M$  z.B.  $N = 1024$  sind dies bei zweidimensionalen Signalen  $f(x,y)$  immerhin  $N \approx 10^4$  komplexe Operationen, was auch für schnelle Computer unakzeptabel lange dauert.

Um die Zahl der Operationen zu verringern, macht man sich bei dem Algorithmus der schnellen Fouriertransformation die Tatsache zu Nutze, daß es möglich ist, die zwei Fouriertransformationen von jeweils der Hälfte der Punkte (je  $N^2/4$  Operationen) mit einer komplexen Addition und Multiplikation wieder miteinander zu verknüpfen:

Mit der Abkürzung  $W(a,b) := e^{-i2\pi b/a}$   $N=M$

ist

$$F(k) = \sum_{m=0}^{N-1} f(m) W(N, mk)$$

aufgeteilt in Werte mit geradem und ungeradem Index

$$F(k) = \sum_{m=0}^{N/2-1} f(2m) W(N, 2mk) + \sum_{m=0}^{N/2-1} f(2m+1) W(N, (2m+1)k)$$

Mit  $W(a, 2b) = W(a/2, b)$  und  $W(a, b+c) = W(a, b) W(a, c)$

ist

$$F(k) = \sum_{m=0}^{N/2-1} f(2m) W(N/2, mk) + \sum_{m=0}^{N/2-1} f(2m+1) W(N/2, mk) W(N, k)$$

Mit der Zerlegung des Feldes  $f(m)$  in zwei Felder mit je  $N'=N/2$  Punkten ist die Fouriertransformation als Überlagerung zweier Fouriertransformationen darstellbar:

$$F(k) = \sum_{m=0}^{N'-1} f(2m)W(N', mk) + W(N', k) \sum_{m=0}^{N'-1} f(2m+1)W(N', mk)$$

$$= F_1(k) + W(N', k)F_2(k) \quad (2.2c)$$

Diese Umformung zerlegt die Transformation in zwei unabhängige Transformationen mit halbiertem Punktzahl. Wendet man die Zerlegung sofort an, bis  $F_1$  und  $F_2$  sich nur noch auf eine Summe von 2 Punkten beziehen, so reduziert sich die Fouriertransformation dieser zwei Punkte auf

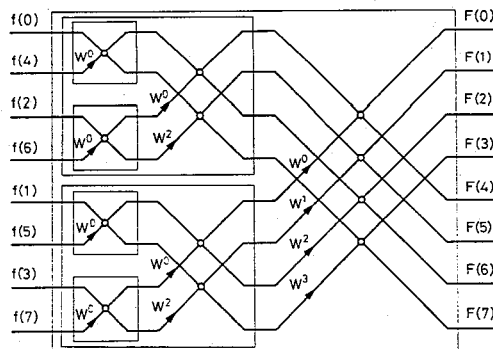
$$F'(k) = f(0) + f(1)e^{-i\pi k} = F_1 + W(2, k)F_2,$$

was der Form (2.2c) entspricht.

Zusätzlich läßt sich der Koeffizient  $W(N, k)$  als Produkt zweier komplexer Zahlen rekursiv ermitteln, ohne daß  $\cos()$  und  $\sin()$  zeitaufwendig neu ermittelt werden müssen:

$$W(N, k+Q) = W(N, k)W(N, Q)$$

In Abb. 2.2b ist das "Netzwerk" eines solchen Algorithmus für 8 Werte gezeigt.



$$W^n = W(N, k)$$

Abb. 2.2b Zur stufenweisen Zerlegung der Fouriertransformation

Man sieht, daß damit die Zahl der Operationen auf  $M \cdot \lg(M)$  heruntergedrückt werden kann. Dies bedeutet in unserem Beispiel von  $N=1024$  Punkten anstelle von  $10^4$  Operationen nur noch  $10^4$ .

Im Folgenden ist der FFT-Algorithmus in PASCAL-ähnlicher Syntax gelistet.

Aufgabe:

Man erstelle mit Hilfe der inversen Binärdarstellung der Indizes  $1..N$  eine Liste  $\text{Trans}(i)$  der neuen Indizes. Diese Liste benutze man, um beim Aufruf von FFT die Feldelemente umzuordnen.

```

PROGRAM fft(input,output);
(*test program for fast fourier procedure *)
CONST Pi = 3.14159265;

TYPE COMPLEX = RECORD
    re: REAL;
    im: REAL;
END;

    modus = (Hintransf,Ruecktransf);

FUNCTION PwrOfTwo(s:Integer):INTEGER;
VAR i2:INTEGER; I: INTEGER;
BEGIN
    IF (s>=0) AND (s<15) THEN BEGIN
        i2:=1; FOR I:=1 TO s DO i2:=i2*2;
        PwrOfTwo:=i2;
        END
    ELSE PwrOfTwo:=0;
END (*PwrOfTwo*);

PROCEDURE FFT(VAR A: ARRAY OF COMPLEX; M: INTEGER; Mode: modus);
VAR
    W,T,U : COMPLEX;
    N,N2,N1,I,j,k,l,Le,L2,Ip: INTEGER;
    Utmp : REAL;
BEGIN
    (* Berechnung der Konstanten *)
    N:= PwrOfTwo(M); (* 2M = Zahl der Punkte des Feldes *)
    N1:= N-1; (* 2M -1 *)
    N2:= N DIV 2; (* 2M-1 *)

    (* Umordnung der Feld-Elemente *)
    j:=1;
    FOR I:=1 TO N1 DO BEGIN (* top-down Methode *)
        IF I<j THEN BEGIN
            T := A[j]; (* Austausch A[j] <-> A[I] *)
            A[j]:= A[I];
            A[I]:= T;
        END;
        k:= N2;
        WHILE k<j DO BEGIN (* naechste Subteilung *)
            j:= j-k;
            k:= k DIV 2;
        END;
        j:= j+k;
    END;

    (* Multiplizieren und Addieren der Feldelemente *)
    FOR I:=1 TO M DO BEGIN (* Bottom-up alle Stufen *)
        (* Bildung der Konstanten *)
        Le:= PwrOfTwo(1); (* 2L = Zahl der Punkte pro Gruppe *)
        L2:= Le DIV 2; (* 2L-1 = Zahl der Mult. pro Gruppe *)

        U.re := 1.0; (* init von W(N',k) *)
        U.im := 0.0;

        W.re := cos(Pi/FLOAT(L2)); (* W = exp(-i Pi / L2 *)
        W.im := sin(Pi/FLOAT(L2));

        IF Mode = Hintransf THEN W.im := -W.im END;

        FOR j:=1 TO L2 DO BEGIN (* Alle Multiplikationen pro Gruppe *)
            FOR I:=j TO N STEP Le DO BEGIN
                Ip := I+Le;
                T.re := A[Ip].re*U.re - A[Ip].im*U.im; (* T := A[Ip]*U *)
                T.im := A[Ip].re*U.im + A[Ip].im*U.re;
                A[Ip].re:= A[I].re - T.re; (* A[Ip]:= A[I]-T *)
                A[Ip].im:= A[I].im - T.im;
                A[I].re := A[I].re + T.re; (* A[I]:= A[I]+T *)
                A[I].im := A[I].im + T.im;
            END;
            (* iterative Berechnung von W() *)
            Utmp:=U.re;
            U.re := U.re*W.re - U.im*W.im; (* U:= U * W *)
            U.im := Utmp*W.im + U.im*W.re;
        END;

        (* Normieren der Werte bei Ruecktransformation *)
        IF Mode = Ruecktransf THEN (* A:=A/N *)
            FOR j:=1 TO N DO BEGIN
                A[j].re := A[j].re / FLOAT(N);
                A[j].im := A[j].im / FLOAT(N);
            END;
    END;

END;

END (*FFT*);

BEGIN
END (*fft*).

```

## 2.3 Grauwerttransformationen

Für den menschlichen Betrachter sind Einzelheiten in Bildern leichter zu erkennen, wenn gute Kontrastwerte vorhanden sind. Ist ein Bild unter- oder überbelichtet, so verbessert eine Transformation, die die aktuellen Grauwerte "besser" im Grauwertbereich verteilt, das Kontrastverhalten.

### Histogrammtransformation

Eine wichtige Methode, die vorhandenen Grauwerte gleichmäßig über den Grauwertbereich zu verteilen, besteht in der Transformation des Grauerthistogramms. Dazu wird die Zahl  $h$  der Pixel im Bild mit einem bestimmten Grauwert  $i$  ermittelt. Sind beispielsweise 8 Bit pro Pixel Grauwertauflösung vorhanden, so ist  $i$  aus  $0..Q-1$  mit  $Q=256$ .

Die Funktion  $h(i)$  heißt dabei **Grauerthistogrammfunktion  $h(i)$** . Hat das Bild  $M \times N$  Pixel, so ist eine gleichmäßige Grauwertverteilung dann gegeben, wenn jede Graustufe bei der gleichen Zahl von Pixeln vorhanden ist

$$h(i) = MN/Q$$

Nun ist aber die Zahl von Pixeln pro Graustufe Graustufe im Bild vorgegeben; ohne Kenntnis des Bildinhalts, der ja erst ermittelt werden soll, ist es nicht möglich, die Pixelmenge einer Grauwertstufe auf andere Stufen zu verteilen. Stattdessen versuchen wir nur, die normierte Summe der Grauwertpixel  $H(i)$  (kumulative Verteilungsfunktion) möglichst linear wachsen zu lassen.

$$H(i) := \frac{1}{M \cdot N} \sum_{z=0}^i h(z) \quad \text{und } H(i) \sim i$$

Die transformierte  $i'$  des Grauwerts  $i$  ist mit der Normierungskonstanten  $Q-1$  (da  $H(Q-1)=1$ )

$$i' = (Q-1) H(i) = \frac{Q-1}{MN} \sum_{z=0}^i h(z)$$

wobei  $i'$  als ganzzahlige Grauwertstufe ein gerundeter Wert ist. In der Abbildung 2.3 ist dies nochmals verdeutlicht. Abb 2.3a) zeigt das Originalbild, c) das dazugehörige Grauerthistogramm  $h(i)$  und e) die nichtlineare Summenfunktion  $H(i)$ . Abb 2.3b) zeigt das grauwerttransformierte Bild mit der transformierten Histogrammfunktion  $h'(i)$  und der Summenfunktion  $H'(i)$ . Wie man sieht, haben alle Pixel mit dem selben Grauwert immer noch die gleiche, aber in ihrem absoluten Wert verschobene Graustufe. Da die jeweilige Zahl der Pixel eines Grauwerts erhalten blieb, ist die Funktion  $H'(i)$  in Abb. 2.3f) nicht exakt linear, sondern wächst in Stufen an.

### Aufgabe:

Wie verändert eine erneute Transformation  $i''$  die Funktionen  $h'(i)$  und  $H'(i)$ ?

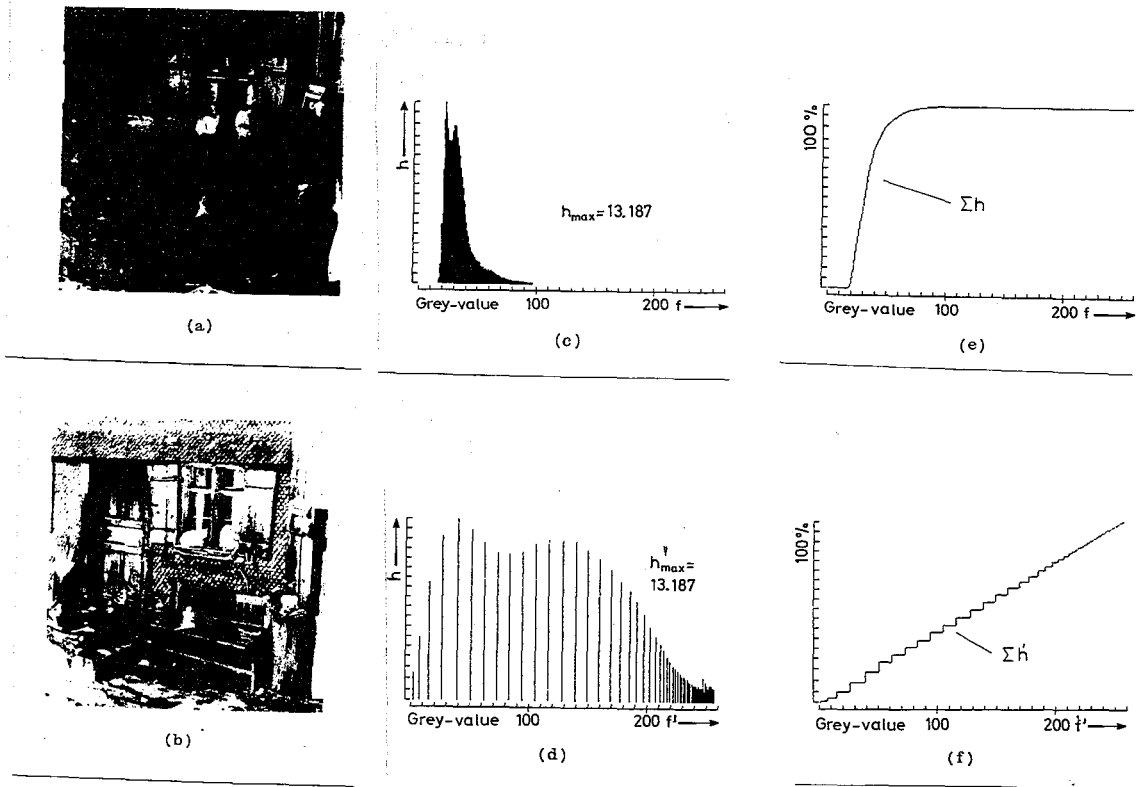


Abb. 2.3. Beispiel einer Histogrammeinebnung: (a) Originalbild, (b) verarbeitetes Bild. (c), (d) zu (a), (b) korrespondierende Histogramme, (e), (f) zu (a), (b) korrespondierende Intensitätsverteilungsfunktionen.

### 2.4 Medianfilterung

Tiefpaß- und Hochpaßoperationen haben den entscheidenden Nachteil, daß entweder Kanten unterdrückt (Tiefpaß) oder Bildstörungen (Rauschen) gefördert werden.

Ein nichtlinearer Operator, der störendes Rauschen unterdrückt, aber Kanten beibehält, ist der Medianoperator

$$f(x, y) := f_{m/2} \quad \text{mit} \quad \begin{cases} f_k := f(x+i, y+j) \\ f_1 \leq f_2 \leq \dots \leq f_{m-1} \leq f_m \end{cases}$$

Operatorfenster  
 $i, j \in [-n, \dots, +n]$   
 $k \in [1, \dots, M]$   
 $M := (2n+1)^2$

geordnete Pixelmenge

Der Nachteil der Medianfilterung liegt darin, daß neben dem Rauschen auch alle Strukturen beseitigt werden, die kleiner als das Operatorfenster  $-n, +n$  sind (z.B. Linien, Punkte, Ecken).



### 3.0 Kantendetektion

Für sehr viele Bildverarbeitungsaufgaben, beispielsweise in der Kartografierung, in der Medizin oder bei industrieller, automatischer Sortierung ist es wichtig, die Umrise oder Kanten von Objekten zu ermitteln, um anschließend das Objekt zu indentifizieren und seine Lage zu bestimmen.

#### 3.1 Lokale Gradientenoperatoren

Kanten sind in Bildern meist durch Helligkeitsunterschiede gekennzeichnet; andere Kantenarten (z.B. Texturkanten) sind nach einer Vorverarbeitungsoperation genauso zu behandeln.

Betrachten wir nun einen Schnitt durch eine Kante in Abb.3.1a.

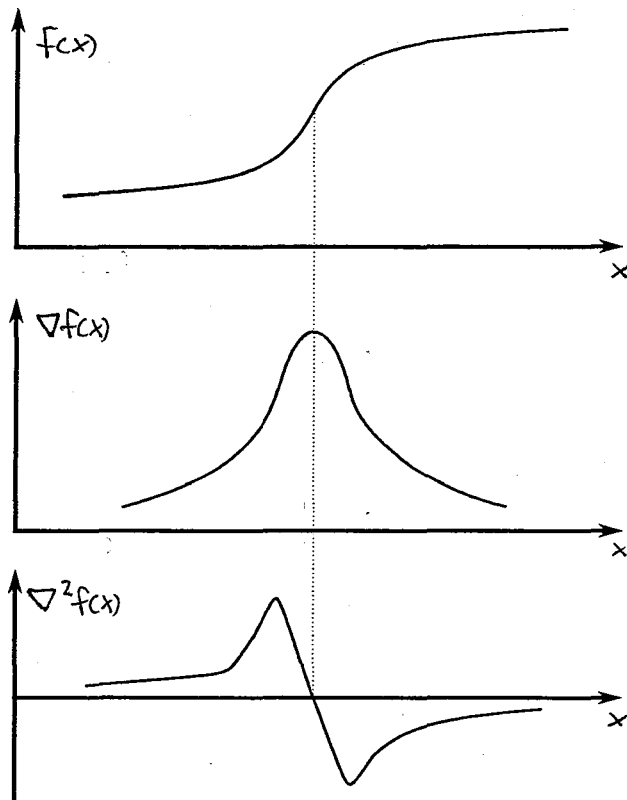


Abb 3.1 Bei einem Funktionsverlauf entsprechend der oberen Kurve erreicht die Steigung oder erste Ableitung ein Maximum, wie in der zweiten Kurve dargestellt. Daraus folgt für die zweite Ableitung (untere Abb.) ein Vorzeichenwechsel oder „Nulldurchgang“ an dieser Stelle

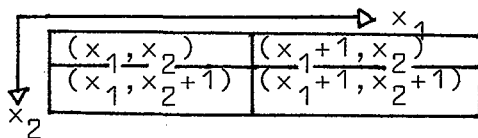
Abb. 3.1a Intensitätsverlauf und Gradienten einer Kante

Die Kante ist durch eine rasche Helligkeitsänderung im Ortsraum gekennzeichnet. Der Betrag des Gradienten  $S(\underline{x}) = |\nabla f(\underline{x})|$  ist somit bei Kanten besonders groß. Setzen wir eine Schwelle fest für die Grauwertänderung, bei der auf "Kante" erkannt wird, so müssen nur noch alle benachbarten Pixel mit  $S(\underline{x}) > \theta$  verbunden werden, um ein Kante zu finden. Ersetzen wir die Ableitung durch den Differenzenquotient, so ist

$$\nabla f(\underline{x}) = \partial f(\underline{x}) / \partial \underline{x} = (f(x_1+1) - f(x_1), f(x_2+1) - f(x_2)) =: (\Delta_1, \Delta_2)$$

Damit ist  $|\nabla f(\underline{x})| = (\Delta_1^2 + \Delta_2^2)^{1/2}$

Die Operationen  $\Delta_1, \Delta_2$  lassen sich auch durch ein Schema darstellen, in dem die Gewichte der Nachbarpixel eingetragen sind, die dann addiert werden.



Damit ist  $\Delta_1 = \begin{bmatrix} -1 & 1 \\ 0 & 0 \end{bmatrix}$

$\Delta_2 = \begin{bmatrix} -1 & 0 \\ 1 & 0 \end{bmatrix}$

Natürlich sind die Ergebnisse der Gradientenmethode rotationsinvariant, so daß auch

$\Delta_1 := \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$

$\Delta_2 := \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$  (Roberts-Operator)

das gleiche Ergebnis liefert. Der obige Operator wurde zuerst von Roberts (1965) benutzt. Leider ist der Operator ziemlich empfindlich gegenüber Störungen wie Rauschen (zufällige Intensitätsfluktuationen), so daß man lieber eine Variante davon benutzt, die mit einem größeren Fenster mehr Bildpunkte benutzt, dem **SOBEL-Operator**:

$\Delta_1 = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$

$\Delta_2 = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$

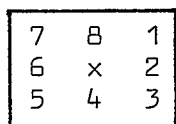
Um festzustellen, ob überhaupt ein großer Gradient (eine Kante) vorhanden ist, ist es unerheblich, ob man den tatsächlichen Betrag des Gradienten wählt oder beispielsweise

$S(x) := \max(|\Delta_1|, |\Delta_2|)$

Das Maximum in verschiedenen Richtungen läßt sich auch allgemeiner suchen:

$S(x) := \max(1, \max_k \sum_{k-1}^{k+1} |f(x_k) - f(x)|)$  **KIRSCH-Operator**

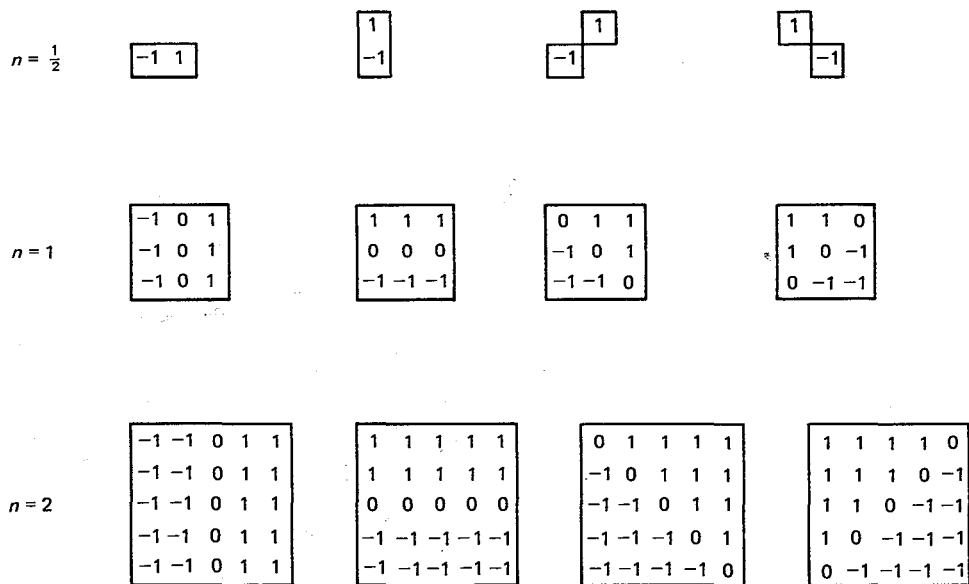
wobei k ein Index Modulo 8 aus den 8 Nachbarn von x ist:



Dabei wird also immer die mittlere Differenz in Richtung des Pixels k gebildet.

Anstelle der 8 Richtungen reicht es auch, den Gradienten in 4 Richtungen zu bilden und das Maximum zu suchen.

Varianten des Kirsch-Operators benutzen diskrete Muster und mitteln über verschieden große Umgebungsbereiche des Pixels x. In Abbildung 3.1.b sind Beispiele für verschiedene Operatoren-Varianten gezeigt.

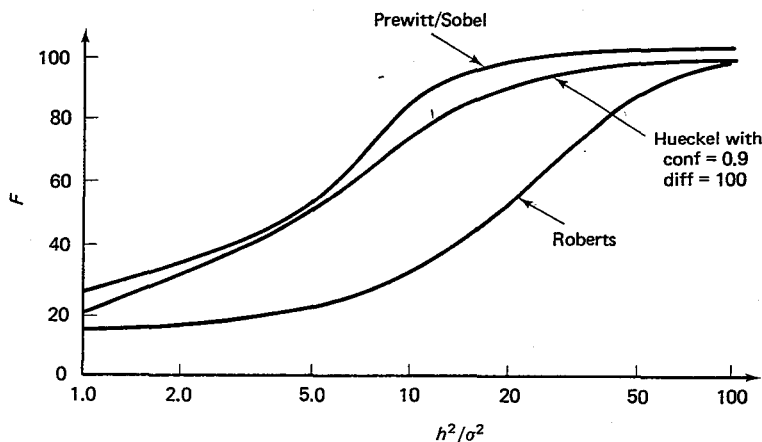


### 3.2 Die Güte der Kantenoperatoren

Es gibt eine große Zahl von verschiedenen Kantenoperatoren. Pratt (1978) stellte ein Gütemaß auf, um die Leistung dieser Operatoren miteinander vergleichen zu können. Als Kriterium wurde der Fehler genommen, der bei der Kantenoperation (Abweichung  $d_i$  im Punkt  $i$  von dem Punkt  $i$  der idealen Kante) entsteht. Betrachten wir den quadratischen Fehler  $d_i^2$ , gewichtet mit dem Faktor  $a$ . Der Kehrwert davon ist die "Güte" des Operatorresultats, so daß bei  $N_A$  aktuellen Kantenpunkten und  $N_I$  Punkten der idealen Kante die mittlere Güte pro Punkt

$$F = \frac{1}{\max(N_A, N_I)} \sum_{i=1}^{N_A} \frac{1}{1 + (ad_i^2)}$$

Das Verhalten der Kantenoperatoren bei einer idealen Kante in Gegenwart einer Gaußschen Verteilung von Störungen (weißes Rauschen) ist in der folgenden Abbildung gezeigt, bei der die Güte in Abhängigkeit vom Signal-Rauschverhältnis aufgetragen ist:



Dabei wurde die Schwelle, oberhalb derer auf einen Kantenpunkt erkannt wurde, für jeden Operator einzeln nach  $F$  optimiert.

### 3.3 Kantenrelaxation

Die bisher vorgestellten Verfahren haben verschiedene Nachteile, z.B. die Empfindlichkeit gegenüber Rauschen und auftretende Widersprüche bei der Kantendetektion, da nur lokale Information

verwendet wird. Eine Möglichkeit, die Detektion von "sinnvollen" Kanten zu verbessern, ist die Vertrauenswürdigkeit einer Kante in Abhängigkeit von seinen Nachbarn iterativ zu verbessern. Dabei wird entweder die Gewißheit, daß ein Element einer Kante vorliegt, größer, oder aber es wird kleiner, bis es unterhalb einer Schwelle wegfällt.

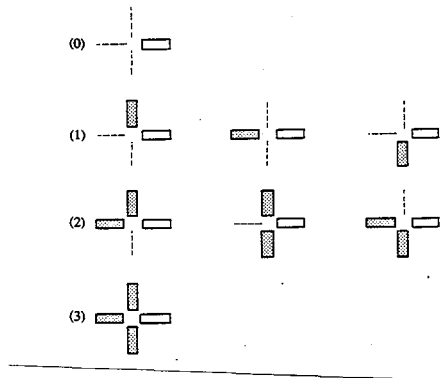
Dieser Algorithmus kann prinzipiell parallel durchgeführt werden und besteht aus folgenden 3 Schritten:

Initial gehen wir von einem Bild  $f(x)$  aus, bei dem an jedem Punkt  $x$  der Gradient errechnet wird. Dieses Gradientenbild  $CF(x)$  wird mit dem maximalen Gradienten normalisiert. Sei initial  $k:=1$ .

1) Berechne den "Kantentyp" jedes Kantenstücks

mittels der Vertrauenswerte  $CF()$  der Nachbarn:

Sei ein Kantenelement  $e$  gegeben, d.h. der Gradient ist an dieser Stelle größer als ein Schwellwert, z.B. 0.25. Zuerst wird der Vertextyp  $i$ ,  $i=0..3$ , als Zahl der existierenden Nachbarkanten bestimmt.



Die Existenz der Nachbarkanten wird dabei nicht einfach über den Schwellwert bestimmt, sondern der Vertextyp ist derjenige Index, der eine Funktion  $C(i)$  maximiert.

Seien die ab steigend geordneten  $CF()$ -Werte der Nachbarkanten  $a, b, c$ , so ist mit  $m := \max(a, b, c, q)$ ,  $q = \text{z.B. } 0.1$ ,

$$C(0) = (m-a) \cdot (m-b) \cdot (m-c)$$

$$C(1) = a \cdot (m-b) \cdot (m-c)$$

$$C(2) = a \cdot b \cdot (m-c)$$

$$C(3) = a \cdot b \cdot c$$

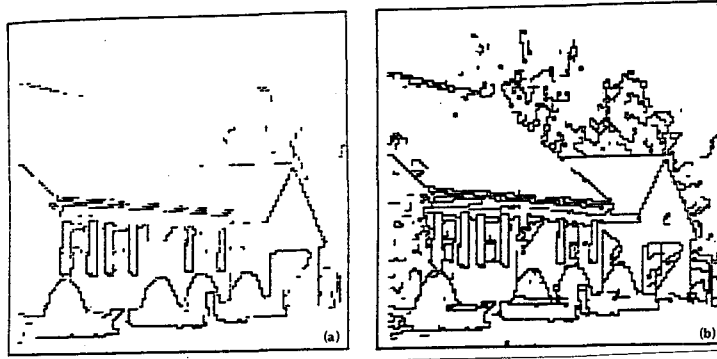
Die Vertextypen  $i$  und  $j$  beider Seiten ergeben zusammen den Kantentyp  $(ij)$ .

2) Berechne den Vertrauensfaktor  $CF()$  eines Kantenstücks neu unter Berücksichtigung des Kantentyps.

Kantentyp	Neuberechnung	z.B. $d=0, 1, \dots, 0, 3$	
0,0 0,2 0,3	}	$CF(e) := \max(0, CF(e)-d)$	
1,1 1,2 1,3		}	$CF(e) := \min(1, CF(e)+d)$
0,1 2,2 2,3 3,3			}

3) Sind alle Vertrauensfaktoren CF() 0 oder 1, dann STOP.  
 Sonst  $k:=k+1$ , weiter mit 1)

In der folgenden Abbildung ist das Ergebnis eines solchen Verfahrens gezeigt. Im linken Bild sind nur die Punkte mit einem Gradienten größer als 0.25 zu sehen; im rechten Bild das Ergebnis der iterativen Neubewertung der Kanten durch die Kantenrelaxation.



Die Probleme dieses Verfahrens liegen in einer möglichen globalen Inkonsistenz der ermittelten Kanten. Beispielsweise können nicht-existente Kanten (zufäll. Intensitätsschwankung) gefunden werden oder aber eine unterbrochene Kante wird nicht als eine einzige erkannt. Diese muß vielmehr manuell mit einem passenden Polynom ergänzt werden. Eine Alternative bietet hier die regionenorientierte Segmentierung aus Abschnitt 4.

### 3.4 Der Mexikanerhut-Operator

Außer den Gradientenverfahren, die die 1. Ableitung verwenden, lassen sich auch Verfahren angeben, um mittels der 2. Ableitung Helligkeitsschwankungen festzustellen.

Betrachten wir in Abb. 3.4 die ideale rechteckige Kantenfunktion  $f(x)$  in einer Dimension.

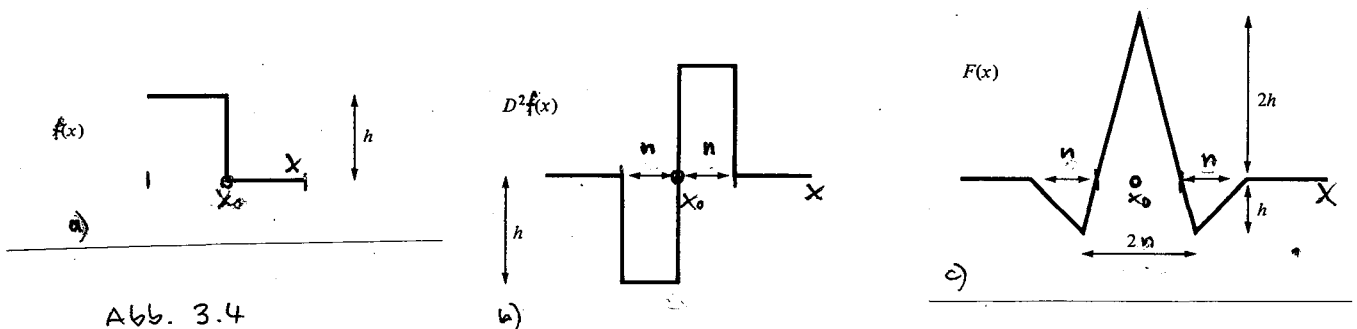


Abb. 3.4

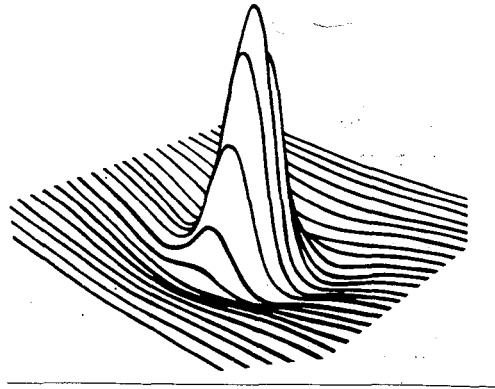
Die 2. Ableitung dieser Funktion erfährt die größte Änderung an dem Kantenpunkt  $x_0$ . Addieren wir nun die Werte der 2. Ableitung an den Stellen  $x_0 + i$ ,  $i \leq n$  zu den Wertbeträgen an den Stellen  $x_0 - i$ , so ist der resultierende Operator  $F(x)$

$$F(x) := \frac{1}{n} \left( \sum_{i=1}^n \partial^2 f(x+i) - \sum_{i=1}^n \partial^2 f(x-i) \right)$$

mit  $n$  Pixeln als Operatorfenster.

Wie in Abb. 3.4 am Beispiel einer Stufenfunktion (a) zu sehen ist,

an der Kante. Bemerkenswert ist die Tatsache, daß durch die Summation zusätzlich eine Rauschunterdrückung stattfindet. Die kontinuierliche, zweidimensionale Form dieses Operators ist bei einer Kantendetektion in der folgenden Abbildung zu sehen.



Es wird vermutet, daß im visuellen System des Menschen ähnliche Mechanismen auf neuronaler Basis eine Rolle spielen. Dies wird ausführlicher in späteren Kapiteln besprochen werden.

## 4.0 Regionen-orientierte Segmentierung

Obwohl bei der Segmentierung das Auffinden von Regionen und das Bestimmen der Kanten (Regionengrenzen) äquivalente Operationen sind, haben beide Verfahrenstypen unterschiedliche Probleme und bilden damit komplementäre Operationen, um ein vorgegebenes reales Bild mit Störungen und Bildfehlern zu segmentieren.

Im Prinzip handelt es sich immer um Operationen, das Bild bzw. Regionen des Bildes in Unterregionen zu zerteilen oder kleinere Unterregionen zu größeren Regionen zu verschmelzen. Kriterium dabei ist eine vorgegebene boolsche Homogenitätsfunktion  $H()$ , die Anhand z.B. von Kontrasten oder Formen in einer Region die Aussage über eine Region  $R$  macht

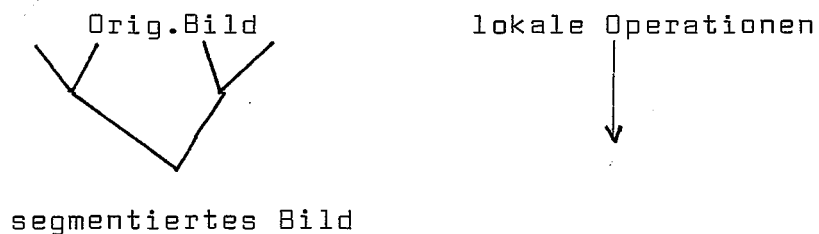
$H(R) = \text{TRUE}$  , wenn die Region homogen ist

$H(R) = \text{FALSE}$  , wenn Inhomogenitäten (z.B. andere Subregionen) existieren

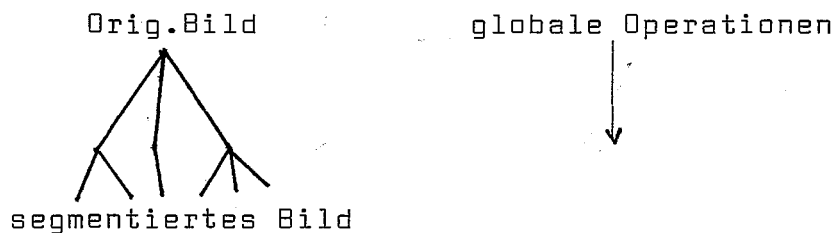
Ist  $R$  die Vereinigung zweier Regionen  $R_i$  und  $R_j$ , so bedeutet  $H(R) = \text{TRUE}$ , daß man die Regionen zu einer einzigen zusammenfassen kann, ansonsten nicht.

Es gibt drei verschiedene Methoden, Verschmelzungs- und Zerteilungsoperationen zu konzipieren.

I) Bottom-up : iteratives Verschmelzen von Regionen



II) Top-Down : iteratives Aufspalten der Regionen



III) Split-and-Merge : Kombination von I) und II)

Für jede der drei Operationen soll im Folgenden ein Beispielalgorithmus angegeben werden.

#### 4.1. Lokale Segmentierung

Beispiel: Verschmelzen von Regionen durch Fleck-Färbung

Dabei wird ein 3-Pixel-Fenster zeilenweise über ein binäres Gesamtbild geschoben und dabei von jedem Pixel  $x$  ein "Farbwert"  $F(x)$  bestimmt. Punkte mit gleichem Farbwert gehören dem gleichen Gebiet an.

Seien die Nachbarn von  $x$  mit  $L$  und  $U$  bezeichnet:

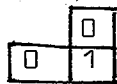


Region und Homogenitätskriterium lauten dann

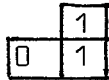
$R := x$  vereinigt mit dem Nachbarpixel

$$H(R) := \begin{cases} \text{TRUE} & f(x) = f(\text{Nachbarpixel}) \\ \text{FALSE} & \text{sonstig} \end{cases}$$

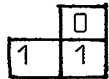
In der nachfolgenden Abbildung sind die Regeln zu sehen :



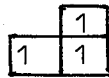
neue Farbe:  $F(x) = k, k:=k+1$



$F(x) := F(U)$



$F(x) := F(L)$



$F(x) := F(U) := F(L)$



nichts tun

Aufgaben: Warum ändert sich nicht an der Segmentierung nach einem erneuten Durchlauf des Operators?

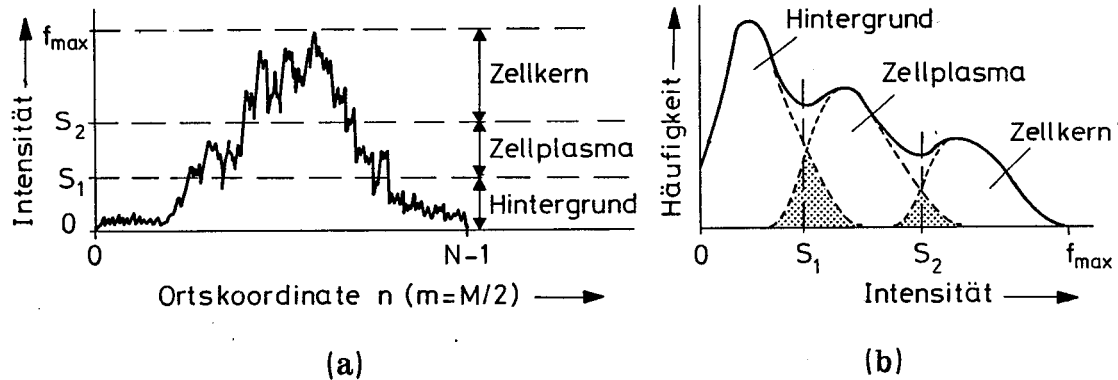
Wie lautet der Algorithmus für ein nicht-binäres Bild mit Grauwerten, wenn z.B.  $x=1, L=1$  übersetzt wird mit "Grauwert( $x$ ) ist ungefähr Grauwert( $L$ )"?



## 4.2. Globale Segmentierung

Beispiel: Aufspalten von Regionen mittels Histogramm-Teilung

Summiert man alle Pixel mit gleichem Grauwert, so erhält man ein Grauerthistogramm (S. Kapitel 2.3.). In der folgenden Abbildung ist das Intensitätsprofil sowie das zugehörige Grauerthistogramm des Bildes einer Zelle gezeigt.



(a) Intensitätsprofil und (b) Histogramm eines typischen Zellenbildes.

Geht man von der einfachen Annahme aus, daß Punkte mit gleicher Helligkeit auch zur gleichen Region gehören, so kann man eine globale Segmentierung vornehmen, indem man die zu einem Maximum des Grauerthistogramms gehörende Punkte als Region segmentiert. Die Intensitätsschwelle  $S_1$  und  $S_2$  in obiger Abbildung lassen sich mit der Annahme herleiten, das Histogramm sei eine Überlagerung verschiedener Gaußverteilungen. Deren Parameter werden dann so bestimmt, daß der Fehler minimal wird.

$$H(R) = \begin{cases} \text{TRUE} & \text{alle Parameter aus } R \text{ gehören zu einem Peak} \\ \text{FALSE} & \text{sonst} \end{cases}$$

Damit läßt sich folgender globaler Segmentierungsalgorithmus zum Aufteilen des Bildes in Regionen herleiten :

- 0) Berechne das Histogramm für das Bild
- 1) Suche das Maximum im Histogramm und teste, ob es ein Peak ist. Existiert Keiner, STOP. Bestimme die Grenzen auf allen Seiten des Peaks und damit mindestens eine Region des Bildes.
- 2) Glätte die Ränder der Region und streiche den entsprechenden Teil aus dem Histogramm.
- 3) Gehe zu 1)

Eine Möglichkeit zur Verbesserung des Verfahrens liegt in der Einführung eines multi-dimensionalen Intensitätsvektors  $f(x)$ , der an Stelle der einfachen Intensität  $f(x)$  aus den Farbwerten, der Sättigung etc. gebildet wird. In Schritt 1 wird dann nach dem größten Maximum aller Komponenten-Histogramme gesucht.

### 4.3 Teilen und Verschmelzen

Leider hat die Methode I) und II) einige Nachteile. Beispielsweise ist die Hintergrundintensität nicht konstant oder Teile, die wir als Region betrachten, enthalten Intensitätsschwankungen, die zur Segmentierung führen.

Manche dieser Probleme lassen sich durch eine vorher durchgeführte Hochpaßfilterung lösen. Es gibt aber noch eine andere Methode, lokale Variationen der Hintergrundhelligkeit zu behandeln. Zerteilen wir das Bild in genügend kleine Unterbilder, so ist die Helligkeit einer Region wieder konstant und es läßt sich für jedes Unterbild separat Methode II) anwenden. Durch rekursives Aufteilen und Vereinigen der betrachteten Unterbilder läßt sich eine Segmentierung erreichen, die nur homogene Regionen enthält :

R = (alle Bildpunkte)

```

REPEAT
  IF H(R)= FALSE THEN split(R)
  IF Nachbar(R1,R2,R3,R4) AND H(R1UR2UR3UR4)=TRUE
    THEN merge(R1,R2,R3,R4)

```

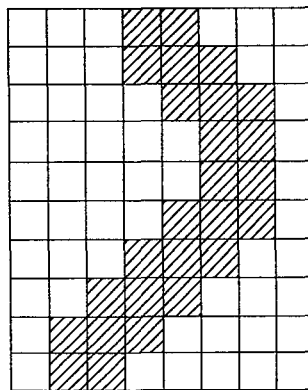
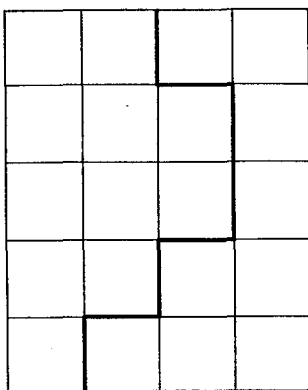
UNTIL (keine der obigen Operationen wurde ausgeführt)

```

IF Nachbar(Ri,Rj) AND H(RiURj)=TRUE THEN merge(Ri,Rj)

```

Dieser Algorithmus benutzt die Voraussetzung, daß ein Bildteil jeweils durch 4 Unterteile gebildet wird. Damit werden nur die problematischen Zonen, beispielsweise die Ränder der Regionen, feiner aufgelöst und damit rechenintensiv bearbeitet.



 Pixels to be classified

## 5.0 Textur

Betrachten wir Bilder, die in nicht-künstlicher Umgebung aufgenommen worden sind, so fällt auf, daß vielfach die Regionen und Flächen, die wir auf dem Bild unterscheiden können, nicht durch diskrete Kanten getrennt sind, sondern durch verschiedene Oberflächenbeschaffenheiten.

In Abb. 5.0a ist ein Beispiel dafür zu sehen.

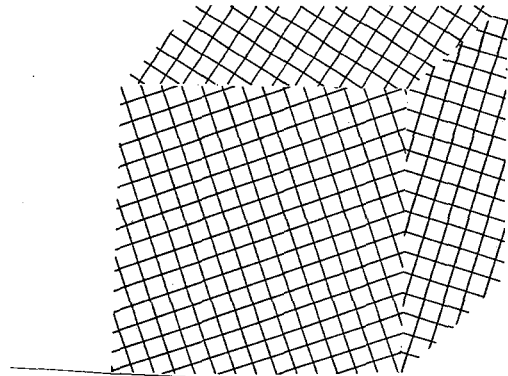


Abb.5.0a Objekt aus Texturflächen

Es ist sehr schwierig, Texturen allgemein zu erkennen und danach ein Bild zu segmentieren. Für verschiedene Arten von Texturen gibt es unterschiedliche Ansätze.

Sind die Texturmuster sehr regelmäßig, so kann man dies in strukturellen Modellen zur Beschreibung ausnutzen. Ist das Muster unregelmäßig, so sind eher statistische Modelle geeignet.

Gemeinsam ist beiden Ansätzen, durch geeignet ermittelte Regelmäßigkeiten die Textur zu charakterisieren und damit eine Wiedererkennung zu ermöglichen. Bei den strukturellen Modellen sind dies grafische Elemente des Bildes, bei den stochastischen Modellen indirekte Merkmale wie lokale Energie und Korrelationen.

### 5.1 Strukturelle Modelle

Besteht die Textur aus regelmäßigen Mustern, wie sie beispielsweise bei den von Menschen hergestellten Gegenständen auftritt, so läßt sich die Textur vielfach als Anreihung von Grundmusterelementen ("Texel") beschreiben. Wie die Abbildung 5.1a zeigt, ist diese Beschreibung nicht unbedingt eindeutig.

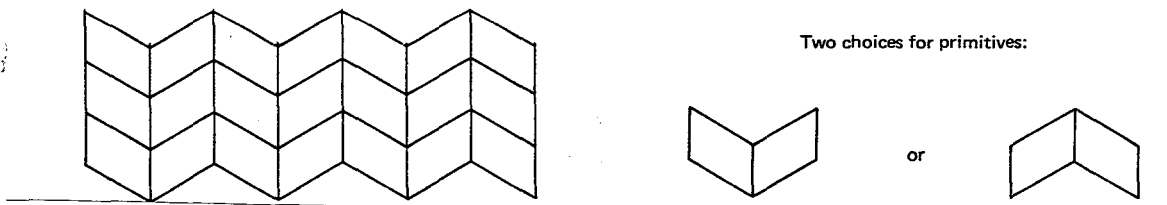
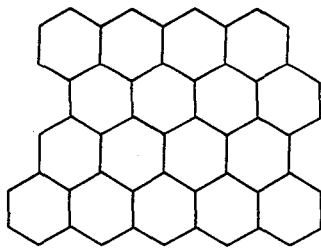


Abb.5.1a nicht eindeutige Texel einer Textur

Es gibt zwei verschiedene Ansätze der formalen Beschreibung: Die Beschreibung durch geeignet gewählte Form-Elemente (Form-Grammatiken) oder der Aufbau der Textur durch regelmäßige Pixel-Anordnung (Baum- und Feldgrammatiken).

## Form-Grammatiken

Betrachten wir als Beispiel die Textur in Abb. 5.1b.



$$\begin{aligned}
 V_t &= \{ \text{hexagon} \} \\
 V_m &= \{ \bullet \} \\
 R &: \text{hexagon} \rightarrow \text{hexagon} \bullet; \text{hexagon} \bullet; \text{etc.} \\
 S &= \{ \text{hexagon} \bullet \}
 \end{aligned}$$

Abb.5.1b hexagonale Textur

Gehen wir von einer Grundform  $S$  aus, so läßt sich die gesamte Textur durch iteratives Hinzufügen weiterer Elemente einer Menge  $V_t$  aufbauen. Die Stelle, an der ein Element angereicht werden kann, sei mit einer Markierung aus  $V_m$  gekennzeichnet. Die Aneinanderreihung ist dabei nicht beliebig, sondern folgt den in einer Abbildung  $R$  festgelegten Regeln. Damit läßt sich eine Textur mit dem Datensatz  $(V_t, V_m, R, S)$  charakterisieren, wobei folgender Platzierungsalgorithmus zu Grunde gelegt wird:

- 1) Suche eine markierte Form in der Textur die "ähnlich" einer Form auf der linken Seite einer Regel von  $R$  ist. Dies bedeutet, daß beide Formen durch eine Transformation  $T$  ineinander übergeführt werden können, die nur aus den elementaren Transformationsoperationen Translation, Rotation, Spiegelung und Skalierung besteht.
- 2) Wende die selbe Transformation  $T$  auch auf die Formen der rechten Seite der Regel aus  $R$  an.
- 3) Ersetze die Form in der Textur durch die transformierte rechte Seite der Regel.

Damit ist die formale Definition einer Formgrammatik verständlich:

Eine Form-Grammatik ist ein Tupel  $(V_t, V_m, R, S)$ ,

- 1)  $V_t$  sei eine endliche Menge von Formen,  $V_t^+$  die Menge aller Anordnungen aller Formen; wobei zusätzliche Formvarianten durch die Transformationsoperationen generiert werden können.  
 $V_t := V_t^+ \cup \{ \}$
- 2)  $V_m$  ist eine endliche Menge von Markierungen mit  $V_t \cap V_m = \emptyset$   
 $V_m^+, V_m^*$  sind analog zu 1) definiert.

- 3) Die endliche Menge  $R$  der Regel besteht aus geordneten Paaren  $(u,v)$ , wobei  $u$  Elemente von  $V_m^+$  kombiniert mit Elementen von  $V_t^+$  enthält.  
 $v$  ist eine Form, die aus einem Element von  $V_t^*$  kombiniert mit einem Element aus  $V_m$  besteht.
- 4) Eine Grundform  $S$ , bestehend aus einem Element aus  $V_t^*$  kombiniert mit einem Element aus  $V_m$ .

### Baum-Grammatiken

Der andere Ansatz, regelmäßige Strukturen zu beschreiben, geht nicht von komplexen Formen aus, die für einen sequentiellen Rechner nur mit erheblichem Rechenaufwand für die Suche nach Ähnlichkeiten verbunden ist.

Stattdessen wird ein Verfahren gesucht, daß einheitlich, einfach und schematisch nur einfache Bildteile (z.B. die einzelnen Bildpunkte oder Pixel) verwendet.

Das zweidimensionale Pixelfeld eines Bildes wird dabei in eine Baumstruktur eingebettet. Dazu ordnen wir jedem Pixel einen Knoten zu und verbinden die Knoten in geeigneter Weise mit Kanten, so daß von einem Startpunkt aus jedes Pixel auf genau einem Weg erreichbar ist. Im Allgemeinen gibt es mehrere mögliche Bäume zu einem Pixelfeld.

In Abbildung 5.1c ist ein Beispiel zu sehen.

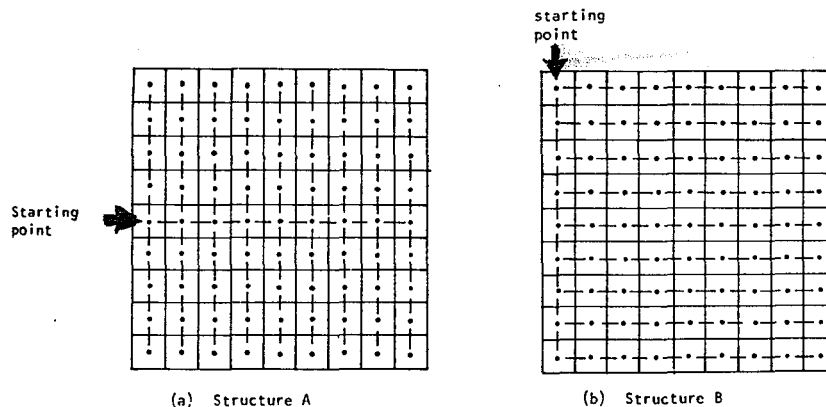


Abb.5.1c Zwei Möglichkeiten, eine Baumstruktur einzubetten

Dabei kann man auch mehrere Hierarchieebenen einführen, wobei das Pixelfeld in mehrere Unterfelder unterteilt wird.

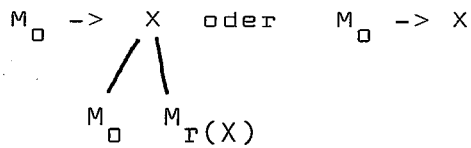
Jedes Unterfeld entspricht einem Knoten des Baumes der obersten Ebene. Jedes Unterfeld kann wieder unterteilt sein mit einem zugehörigen Baum und so fort.

Die Baumstruktur gibt somit eine Anreihungsordnung der Pixel wieder. Formal läßt sich dies analog zu der Form-Grammatik schreiben:

Eine Baum-Grammatik besteht aus einem Tupel  $(V_t, V_m, r, R, S)$

- 1)  $V_t$  Menge der terminalen Elemente (z. B. Pixel)
- 2)  $V_m$  Menge von Markierungselementen, wobei  $V_t \cap V_m = \emptyset$
- 3)  $r: V_t \rightarrow N^+$  Rang der Terminalelemente (Index der Verzweigung)

4) R Menge der Regeln der Form



$$X \in V_t$$

$$M_0 \dots M_{r(X)} \in V_m$$

Abbildung 5.1d zeigt als Beispiel die Zerlegung der hexagonalen Textur aus Abb.5.1a. Das ganze Pixelfeld von 36 x 36 Pixeln sei in 16 Unterfelder von je 9 x 9 Pixeln eingeteilt.

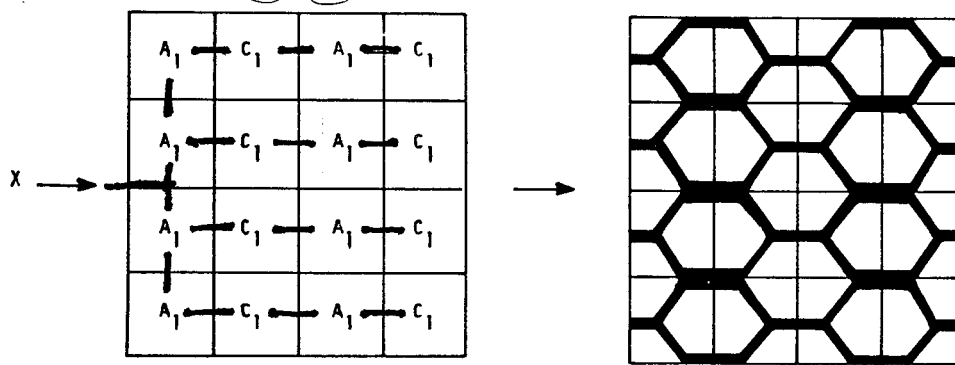


Abb.5.1d Baumstruktur der hexagonalen Textur

Die Grammatik ist dabei:

$$V_t = \{A_1, C_1\}$$

$$V_m = \{M_0, M_1, M_2\}$$

$$r = \{0, 1, 2\}$$

$$R = M_0 \rightarrow \begin{array}{c} A_1 \\ / \quad \backslash \\ M_0 \quad M_1 \end{array} ; \quad \begin{array}{c} A_1 \\ | \\ M_1 \end{array}$$

$$M_1 \rightarrow \begin{array}{c} C_1 \\ | \\ M_2 \end{array} ; \quad C_1$$

$$M_2 \rightarrow \begin{array}{c} A_1 \\ | \\ M_1 \end{array} ; \quad A_1$$

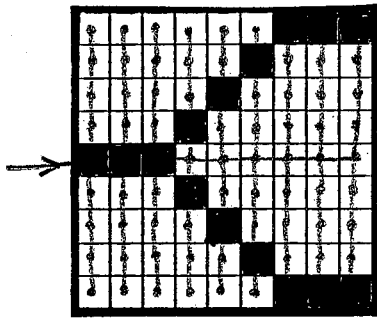
Die Gesamttextur ergibt sich also als Kombination aller Einzelfelder. Jedes Einzelfeld besitzt wieder seine eigene Baumgrammatik, siehe Abb. 5.1e.

$$V_t = \{1, 0\}$$

$$V_m = \{A_1, A_2, A_3, A_4, A_5, A_6, A_7, C_1, C_2, C_3, C_4, C_5, C_6, C_7, N_0, N_1, N_2, N_3, N_4\}$$

$$r = \{0, 1, 2\}$$

$$S = \{A_1, C_1\}$$



R:

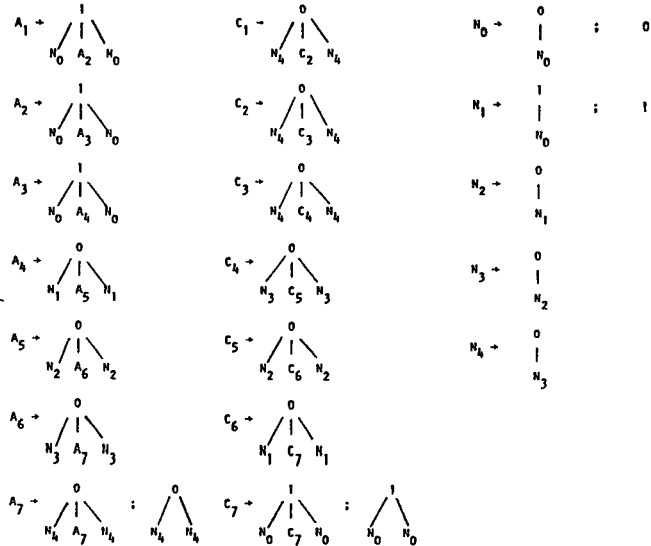


Abb.5.1e Baumgrammatik der Unterfelder

Eine weitere Variante ist die Feldgrammatik, wo die Baumstrukturen nicht vorgegeben wird, sondern durch eine Notation bei den grammatischen Regeln indiziert wird, in welcher Richtung (oben, unten, rechts, links) die Anreihung durchgeführt wird.

Die grammatischen Modelle dienen übrigens nicht nur zur Analyse einer Textur, sondern werden auch zu ihrer Synthese eingesetzt.

## 5.2. Stochastische Modelle

elbsbM edceitesdcofB .S.2

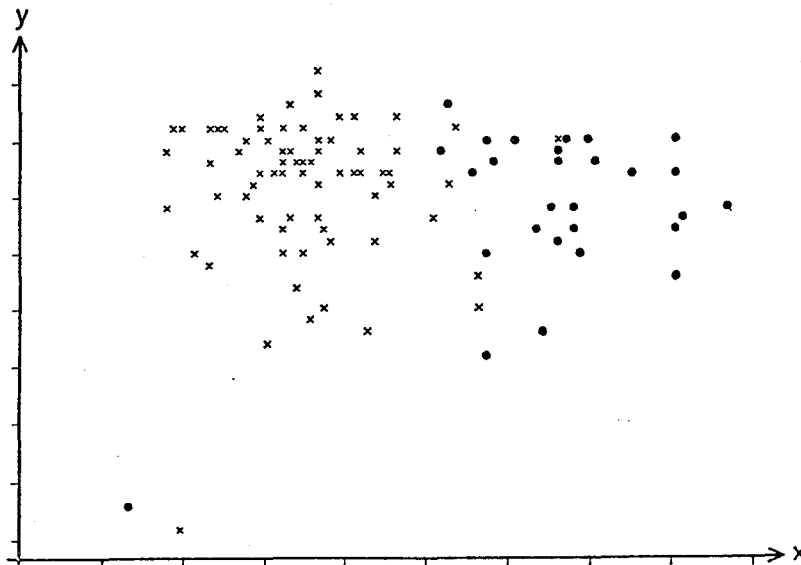
Ist die zu erkennende Textur zu unregelmäßig, so werden zweckmäßigerweise nur noch die Auftretswahrscheinlichkeiten von Grundmustern betrachtet. Existieren regelmäßige Grundformen, die regelmäßig angeordnet sind, so kommt für die Beschreibung eine Grammatik in Frage, deren Regeln mit Wahrscheinlichkeiten gewichtet sind, sog. stochastische Grammatiken. Das Problem besteht hierbei wieder, von einer vorliegenden Textur die Regeln abzuleiten. Ein häufiger verwendeter Ansatz besteht darin, das gesamte Problem der Texturerkennung mit den Hilfsmitteln der stochastischen Mustererkennung zu behandeln.

Worum geht es dabei?

Die **stochastische Mustererkennung** geht von folgendem Grundproblem aus : Gegeben sei ein Muster, das durch eine Anzahl von typischen Merkmalen, zusammengefaßt in einem Vektor  $x$ , beschrieben wird. Anhand dieser Beschreibung kann die Menge aller  $x$  in mehrere Klassen eingeteilt werden.

Abb. 5.2a zeigt die Charakterisierung einer chron. Bauchspeichel Drüsenentzündung anhand der Konzentrat. bestimmter Stoffe X,Y im

Blut der Patienten.



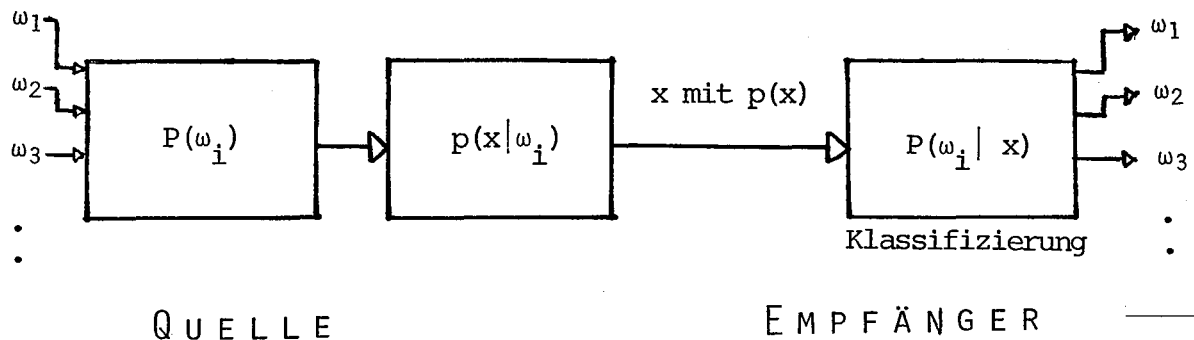
Chronical pancreatitis (x) and normal patients (•).

Abb.5.2a Charakterisierung von chron.Bauchspeicheldrüsenentzündung

Wie man sieht, besteht die Problematik des Arztes darin, einen gegebenen Datensatz  $x$  eines Patienten einer der beiden Klassen zuzuordnen und damit zu entscheiden, ob der Patient eine gesunde oder kranke Bauchspeicheldrüse hat. Bei einer bestehenden Stoffwechselabweichung entscheidet das die Frage: liegt ein Tumor vor oder eine chronische Entzündung? Muß sofort operiert werden oder bringt eine medikamentöse Langzeitbehandlung die Besserung?

Die Entscheidungsproblematik läßt sich folgendermaßen formalisieren :

Eine der Klassen  $\omega_1, \dots, \omega_n$  liege mit der Wahrscheinlichkeit  $P(\omega_i)$  vor und produziere ein Muster  $x$ . Dieses Muster wird von der Quelle mit  $P(x)$  empfangen und im Empfänger in Klassen  $\Omega_i$  klassifiziert.



Direkt oder indirekt wird also vom Empfänger der Wert  $P(\omega_i)$  erschlossen. Dabei sind zwei Situationen zu unterscheiden:



### a) Quelle bekannt

Wenn  $p(x/w_i)$  und  $P(w_i)$  bekannt sind, läßt sich mit

$$p(x) = \sum_i p(x/w_i) \cdot P(w_i) \quad \sum_i P(w_i) = 1$$

und der Bayesregel  $p(y/x)p(x) = p(x/y)p(y)$

$P(w_i/x)$  bestimmen:

$$P(w_i/x) = \frac{p(x/w_i) P(w_i)}{p(x)}$$

Wählt man bei der Klassifikation der  $x$  die Klasse  $O_i$ , für die  $P(w_i/x)$  am größten ist, so läßt sich zeigen, daß dann der mittlere Fehler am kleinsten wird (**Bayes-Entscheidungskriterium**).

In der Praxis wird dieser Gedanke benutzt, um Maschinen für Klassifikationen einzurichten. Bekannte Trainingsmuster werden eingegeben und nach jeder Klassifikation wird signalisiert, ob von der Maschine richtig oder falsch eingeordnet wurde.

Durch die 'richtig'-'falsch' Meldungen der Menschen lernt der Automat  $P(w_i)$ , durch die Trainingsmuster  $p(x/w_i)$ .

Die Klassengrenze ist dabei operational durch die Klassifikationsvorschrift definiert.

Falls  $\bigcap_i O_i = \emptyset$  und  $\bigcup_i O_i = \Omega$ , so läßt sich die Klassengrenze zwischen zwei Klassen auch als Hyperfläche definieren, die beide Teilmengen  $O_i$  und  $O_j$  voneinander trennt.

### b) Quelle unbekannt

Wenn über die Quelle nichts bekannt ist, muß das lernende System  $P(w_i/x)$  direkt lernen.

Da in diesem Fall kein Lehrer dabei ist wie bei a), möchte ich den Vorgang als **Selbstlernen** bezeichnen im Unterschied zu **lernen** in a).

Das Klassifizieren und iterative Verschieben der Klassengrenzen  $O_i$  geschieht hier durch Bilden eines internen Gütekriteriums. Ziel der Klassifizierung und Einordnung der  $x$  ist es, diese Güte so groß wie möglich, oder, invers gesehen, eine Strafe bzw. Kosten so klein wie möglich zu machen.

In der Auswahl und im Einsatz von Straffunktionen steckt natürlich auch hier wieder die Funktion eines Lehrers; nur muß er hier nicht bei jedem einzelnen  $x$  entscheiden, dies tut der "Schüler" selbst. Im Folgenden soll nur noch dieser Fall behandelt werden.

Ein einfaches Beispiel möge dies veranschaulichen.

Betrachten wir die beiden Klassen aus Abb.5.1a. Die Muster sind in zwei Klassen eingeteilt:  $O_1$  und  $O_2$ . Die Klassengrenze wird durch einen Parameter  $c$  festgelegt.

Beispielsweise kann  $c$  die Parameter  $a$  und  $b$  der Geraden enthalten, die die beiden Klassen trennt.

$$c = (a, b) \quad y = ax + b$$

Äquivalent dazu kann man die Klassengrenze auch als Gerade beschreiben, die senkrecht auf der Verbindungslinie zwischen zwei besonders ausgezeichneten Punkten  $c_1$  und  $c_2$  steht. Diese beiden Punkte sind die "typischen" Vertreter ihrer jeweiligen Klasse, die

sog. Klassenprototypen. Damit ist

$$c = (c_1, c_2)$$

Die Straffunktion  $R(c)$  kann das Quadrat des Abstands von  $x$  zu dem Klassenprototypen  $c$  sein:

$$R(c) = R_1(c_1) + R_2(c_2) = \sum_{x \in O_1} (x - c_1)^2 + \sum_{x \in O_2} (x - c_2)^2$$

Um das Gesamtrisiko zu erhalten, muß man das Risiko aller  $x$  aller Klassen bei einer vorgegebenen Klasseneinteilung  $O$  aufaddieren. Ziel des Selbstlernens ist es, eine solche Klasseneinteilung zu finden, daß  $R(c)$  minimal wird. Im Idealfall sind dann alle  $x$  der Klasse  $w_i$  auch in die Klasse  $O_i$  eingeordnet worden.

### 5.2.1 Featurebildung

Wie bereits erwähnt, ist die Auswahl und Berechnung der typischen Merkmale ("Features") entscheidend für die Leistung des mustererkennenden Systems. Welche Features lassen sich bei der Texturerkennung verwenden?

#### Features der Fouriertransformierten

Eine Möglichkeit besteht darin, die Textur durch ihre "Grobheit" zu charakterisieren, d.h. die Zahl der Schwarz-Weiß-Wechsel pro Längeneinheit. Dies entspricht einem bestimmten Bereich im fouriertransformierten Bild. In Abb. 5.2.c sind zwei mögliche Unterteilungen der Fouriertransformierten gezeigt.

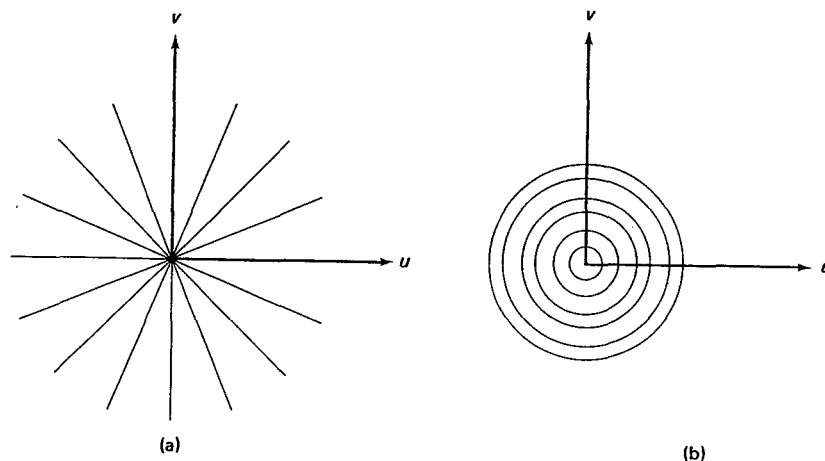
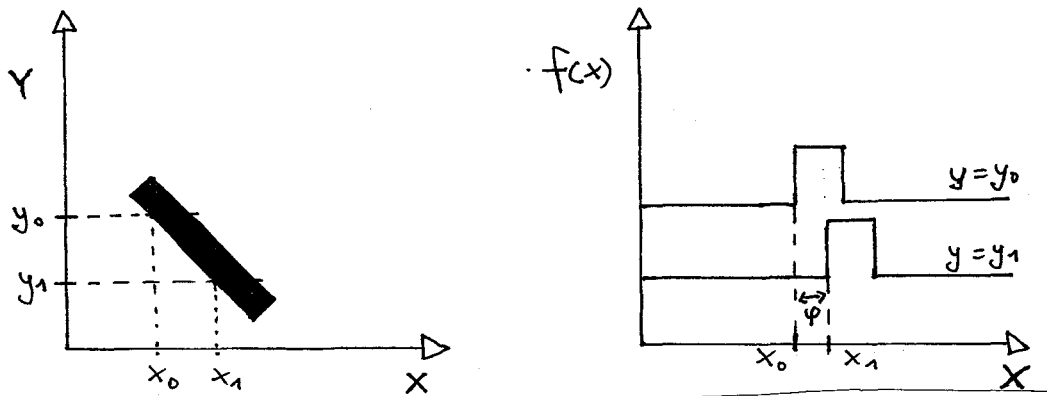


Abb.5.2c Features der Fouriertransformierten

Die Features sind die Gesamtintensität (Integral des Amplitudenquadrats) in den einzelnen Bereichen; die Gesamtheit aller Features bilden den Mustervektor  $x$ . Die Teilung a) berücksichtigt die Phasenwinkelverhältnisse und ist

damit empfindlich für Richtungen der Texturlinien.



Originalbild

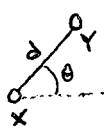
Intensitätsverlauf zweier Zeilen

Abb.5.2d Phasenverschiebung bei schrägen Konturen

Die Teilung b) diskriminiert dagegen stärker die Frequenzen und ist damit empfindlich für die unterschiedliche Dicke von Linien und Mustern in der Textur.

### Features mit SGLD \_ Matrizen

Eine der beliebtesten Featuregruppe der Texturerkennung benutzt sog. "Spatial-gray-level-dependance" Matrizen. Dies sind Matrizen, die über das gleichzeitige Auftreten zweier verschiedener Grauwerte  $i$  und  $j$  im Abstand  $d$  mit dem Winkel  $\theta$  Auskunft geben :



$$\left. \begin{aligned} f(x) &= i \\ f(y) &= j \end{aligned} \right\}, \text{ wobei } y = x + d \quad d = (d \cos\theta, d \sin\theta)$$

$S(i, j)_{d, \theta}$   $\left\{ \begin{array}{l} \text{Zahl der Ereignisse} \left\{ \begin{array}{l} \text{Grauwert } i \text{ am Punkt } x, \\ \text{Grauwert } j \text{ am Punkt } y, \end{array} \right. \text{ Abstand } \overline{xy} = d, \\ \text{Winkel} = \theta \end{array} \right\}$   
 muß durch systematische Computeranalyse gewonnen werden.

SGLD-Matrix-Element

Da  $S(i, j)_{d, \theta} = S(j, i)_{d, \theta + \pi}$  (Sinus, Cosinus sind periodisch bzw. der Abstand kann symmetrisch betrachtet werden), reicht es,  $S(i, j)_{d, \theta}$  mit  $\theta < \pi/2$  zu betrachten.

Damit lassen sich folgende Features bilden

1. Energy

$$E(d, \theta) = \sum_{i=0}^K \sum_{j=0}^K [S(i, j|d, \theta)]^2$$

2. Entropy

$$H(d, \theta) = \sum_{i=0}^K \sum_{j=0}^K S(i, j|d, \theta) \log S(i, j|d, \theta)$$

### 3. Correlation

$$C(d, \theta) = \frac{\sum_{i=0}^K \sum_{j=0}^K (i - \mu_x)(j - \mu_y) S(i, j|d, \theta)}{\sigma_x \sigma_y}$$


---

$$\mu_x = \sum_{i=0}^K i \sum_{j=0}^K S(i, j|d, \theta)$$

$$\sigma_x^2 = \sum_{i=0}^K (i - \mu_x)^2 \sum_{j=0}^K S(i, j|d, \theta)$$

$$\mu_y = \sum_{i=0}^K j \sum_{j=0}^K S(i, j|d, \theta)$$

$$\sigma_y^2 = \sum_{i=0}^K (j - \mu_y)^2 \sum_{j=0}^K S(i, j|d, \theta)$$


---

### 4. Inertia

$$I(d, \theta) = \sum_{i=0}^K \sum_{j=0}^K (i - j)^2 S(i, j|d, \theta)$$


---

### 5. Local Homogeneity

$$L(d, \theta) = \sum_{i=0}^K \sum_{j=0}^K \frac{1}{1 + (i - j)^2} S(i, j|d, \theta)$$


---

Man muß dabei beachten, daß keines dieser Features dem entspricht, was menschliche Beobachter als "typisch" für Texturen empfinden.

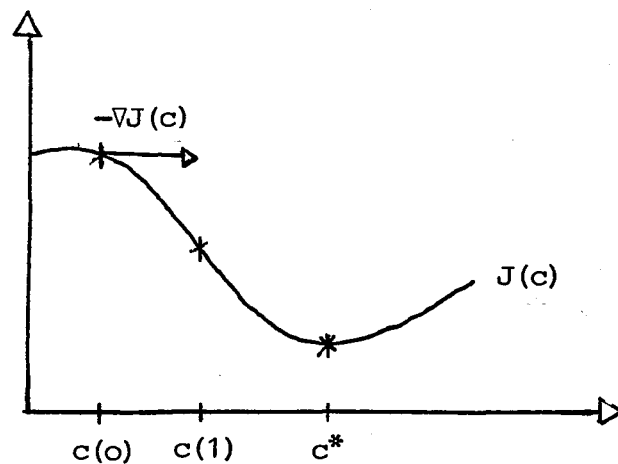
## 5.2.2 Selbstlernen der Texturarten

### Die Gradientenmethode

Im Abschnitt 5.2 wurde als Kriterium der besten Klassifizierung das Ziel genannt, ein Risiko der Falschklassifizierung oder den gewichteten Fehler  $R(\mathbf{c})$  mit der Variation der Klassengrenzen, die von einem Parametervektors  $\mathbf{c}$  festgelegt sind, minimal zu machen.

Betrachten wir allgemein das Problem, das Minimum  $J(\mathbf{c}^*)$  einer vorgegebenen Funktion  $J(\mathbf{c})$  zu finden.

Dies löst sich am besten mit Hilfe folgender Iteration:



Sei ein Wert  $c(0)$  gegeben.

Dann gibt der Gradient  $\nabla_c J(c)$  in  $c(0)$  die Richtung der stärksten Steigung und  $-\nabla_c J(c(0)) := g$  zeigt in die stärkste Gefällerrichtung. Dieser Gefälle-Vektor zeigt daher auch in Richtung des Minimums von  $J(c)$  und ermöglicht ein verbessertes  $c(1) := c(0) + g$ , das dichter an  $c^*$  liegt.

Der allgemeine Algorithmus enthält noch einen Proportionalitätsfaktor  $\alpha$ , um je nach Funktion die Schrittweite zu korrigieren:

$$c(n+1) = c(n) - \alpha \nabla_c J(c(n))$$

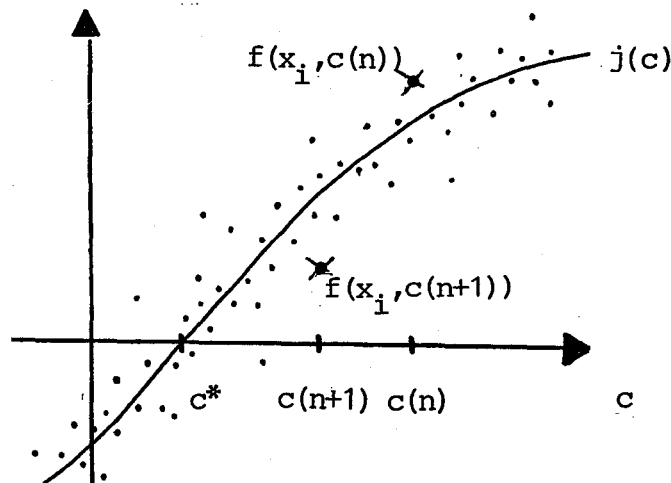
Wenn  $c$  ein Vektor ist, so ist im allgemeinen eine Matrix; häufig reicht es, die Einheitsmatrix mit dem skalaren Faktor  $\alpha$  zu nehmen. Für schnellere Konvergenz empfiehlt es sich manchmal auch, eine Matrix mit nichtdiagonalen Termen  $\neq 0$  zu verwenden.

Für unser Problem, das Minimum von  $R(c)$  zu finden, ist aber der Erwartungswert  $J(c) = \int F(x,c) dx$  als Funktion explizit gar nicht gegeben. Mit gegebenen  $x$  und  $c$  läßt sich nur  $F(x,c)$  als "verrauschte Form von  $J(c)$ " bilden.

Wie läßt sich nun trotzdem das Minimum finden?

### Der Robbins-Monro Algorithmus

Robbins und Monro (1951) gaben als erste eine Methode an, wie man die Nullstelle einer "verrauschten" Funktion findet.



Sei  $x$  eine stochastische Variable, so ist  $f(x,c)$  ebenso eine. Würde man sehr viele  $x$  abwarten, so könnte man den Erwartungswert

$$j(c) := E(f(x,c))$$

angenähert bilden und so z.B. mit der Gradientenmethode das Minimum von  $j(c)$ , also  $c^*$  mit  $j(c^*)=0$  finden.

Die Methode der **stochastischen Approximation** von Robbins und Monro kommt aber ohne das lange Abwarten aus.

Mit ähnlichen Überlegungen wie bei der Gradientenmethode erhielten sie ein dem  $c^*$  näheres  $c(n+1)$ :

$$c(n+1) = c(n) - \gamma_n f(x(n), c(n))$$

Für die Konvergenz dieses Algorithmus läßt sich

u. a. zeigen, daß  $\lim_{n \rightarrow \infty} P(c^* = c(n)) = 1$

der Abstand  $|c^* - c(n)|$  in endlicher Zeit also sehr klein wird,  
wenn folgende Voraussetzungen erfüllt sind:

$$\left. \begin{array}{l} f(x, c) \\ j(c) \end{array} \right\} \text{ beschränkt im betrachteten Intervall}$$

$$E(f(x, c) - j(c)) = 0$$

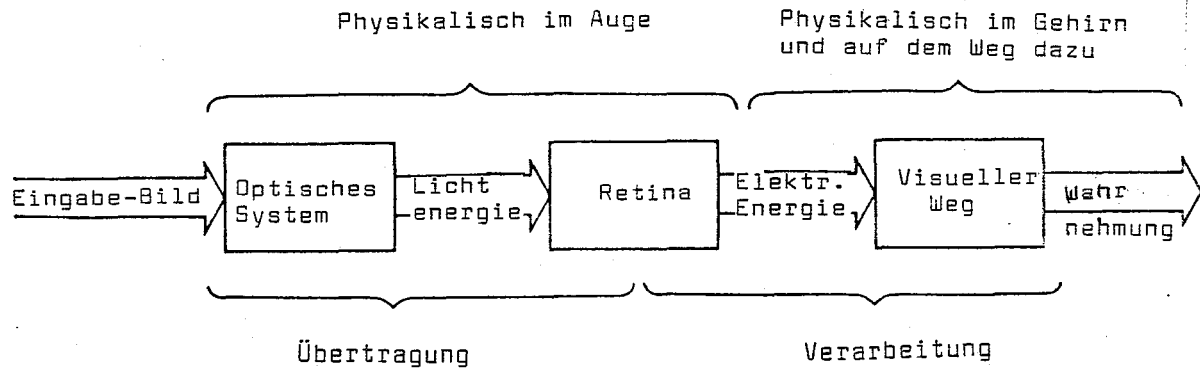
$$\lim_{n \rightarrow \infty} \gamma_n = 0$$

$$\sum_{n=1}^{\infty} \gamma_n > \infty$$

$$\sum_{n=1}^{\infty} \gamma_n^2 < \infty$$

## 6. Das menschliche Sehsystem

Das menschliche Sehsystem ist in seiner Funktion ziemlich unvollkommen bekannt. Im Unterschied zu primitiven Tieren wie Kaltblütlern, Insekten usw. erfolgt die hauptsächliche Auswertung der Bilder nicht direkt beim Auge, sondern in einem extra Teil des Gehirns, dem visuellen Cortex oder Sehzentrum. Bild 6.0 zeigt das Blockschema des menschlichen Sehsystems, so wie man es sich in etwa heutzutage vorstellt.



Betrachten wir nun die einzelnen Teile etwas näher.

### 6.1 Das menschliche Auge

Als Photorezeptor hat das menschliche Auge bestimmte Charakteristika, die kurz dargestellt werden. In Bild 6.1a ist der Querschnitt durch ein menschliches Auge gezeigt.

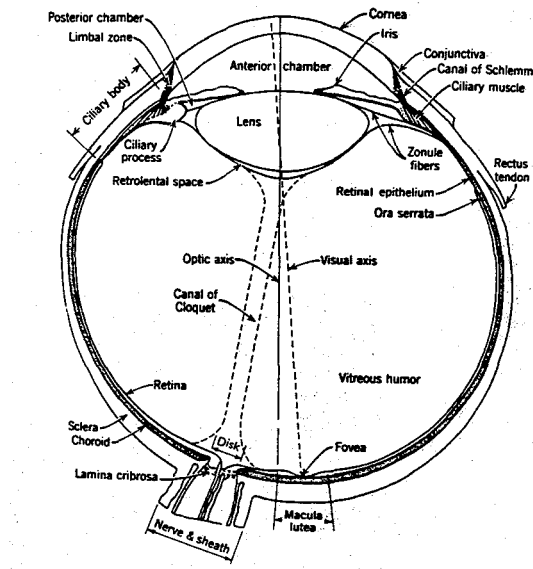


Abb.6.1a Das menschliche Auge

Dabei durchläuft die optische Information verschiedene Stufen.

Im **optischen System** wird das Bild nach dem Durchlaufen verschiedener lichtdurchlässiger Häute durch das Linsensystem und den Glaskörper seitenverkehrt auf die Retina projiziert. Dieser Vorgang ist zwar prinzipiell mit einem einfachen Fotoapparat zu vergleichen, ist aber bekanntlich fehlerhafter. Zum einen treten beim Durchgang durch die verschiedenen, nicht perfekten biologischen Materialien Streulicht und Absorption auf,

so daß nur 50% des ins Auge fallenden Lichts auch wirklich auf der Retina erscheint.  
 Zum andern bedingt diese Imperfektion auch Verzerrungen des einfallenden Bildes. Beispielsweise gibt es keine von den in Fotoobjektiven üblichen Korrekturen der chromatischen Aberation, die zwangsläufig auftritt. Zwar ist mit der Pupille eine gewisse Korrektur der Tiefenschärfe möglich, aber sie dient eher der Intensitätsanpassung des einfallenden Lichts an die Empfindlichkeit der fotosensitiven Zellen auf der Retina.

Im **Retina-System** wird nun dieses imperfekte Bild in Nervenimpulse umgewandelt. Auch hier treten wieder (bedingt durch den Bau der Retina) zahlreiche Zusatzeinflüsse auf. Betrachten wir dazu einen Schnitt durch die Retina in Abb.6.1b.

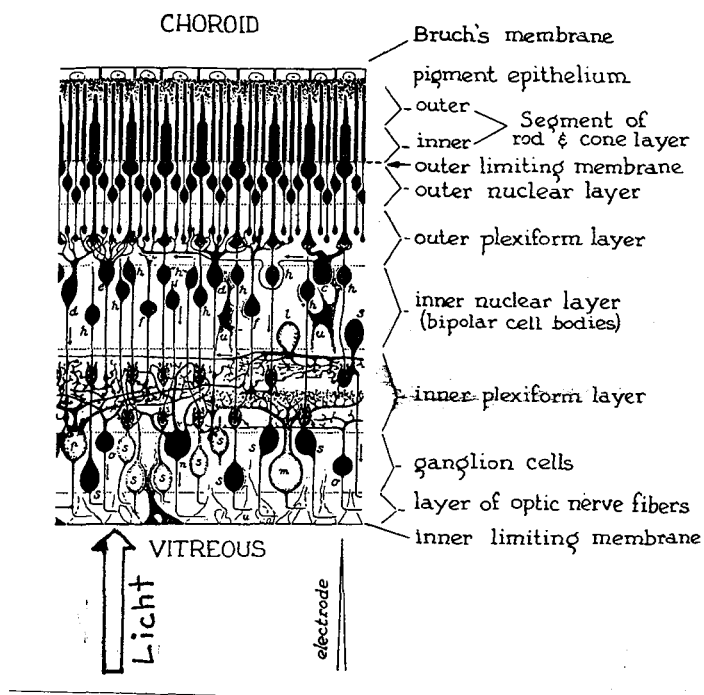


Abb. 6.1b Schnitt durch die Retina

Das Licht durchläuft zwei Schichten von Nervenzellen: Zuerst die Ganglion-Zellen, an die die Sehnerven angeschlossen sind, und dann eine Schicht von bipolaren Zellen, bevor es die eigentlichen Zellen, die Stäbchen und Zäpfchen, erreicht. Stäbchen und Zäpfchen haben verschiedene Funktionen. Es gibt drei verschiedenartige Zäpfchenzelltypen, die verschiedene fotosensitive Verbindungen enthalten. Diese chemische Verbindungen reagieren verschieden auf verschiedene elektromagnetische Frequenzen (Lichtfarben). Abb.6.1c zeigt die Absorptionsbereiche sowie die relativen Empfindlichkeiten der drei Substanzen.



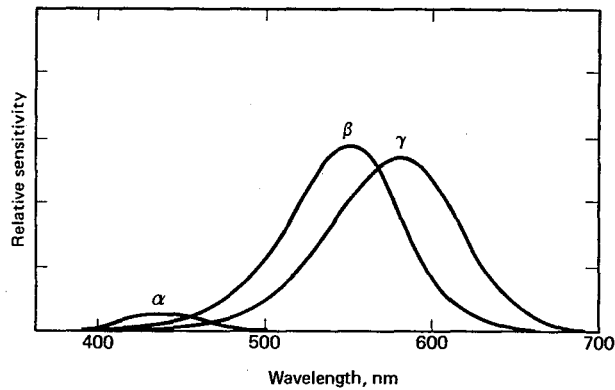
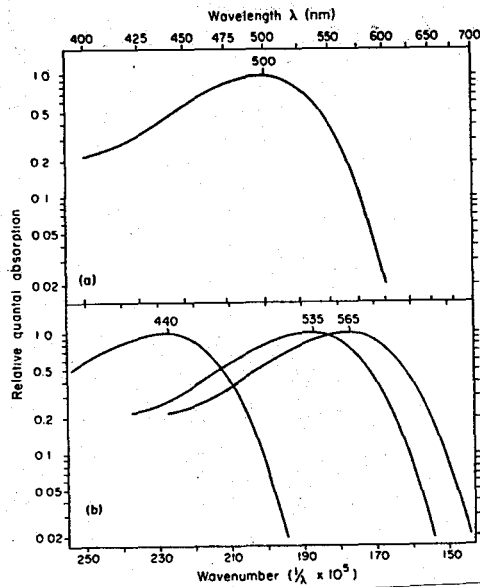


Abb.6.1.c Absorptionsbereiche und rel.Empfindlichkeiten der drei farbsensitiven Substanzen

Die 20 mal häufigeren Stäbchenzellen enthalten im wesentlichen nur eine Substanz, sind dafür aber empfindlicher. Entsprechend diesen Charakteristiken spielt sich das Sehen zwischen zwei verschiedenen Seharten ab: in der Dämmerung (wenig Licht) sind die Zäpfchen kaum gereizt; dafür lassen die Stäbchenzellen das Bild gut erkennen. Bei hellem Sonnenlicht sind die Stäbchenzellen im "Sättigungsbereich", d.h. Helligkeitskontraste führen kaum zu Änderungen der abgegebenen Nervenimpulse. Die Zäpfchen dagegen nehmen nun ausreichend Lichtenergie auf, um ein Farbsehen zu ermöglichen.

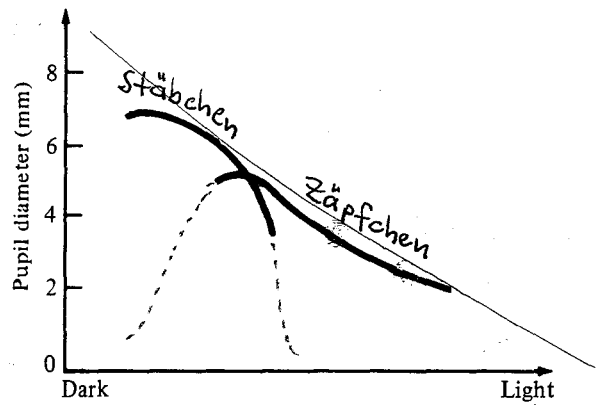


Abb 6.1d Zusammenhang zwischen den beiden Sehmodi, dem einfallenden Licht und der Pupillenerweiterung.

Auch die relative und die absolute Verteilung der Stäbchen und Zäpfchenzellen ist verschieden innerhalb des Auges. Betrachten wir nochmals Abb.6.1a. Dicht bei dem Schnittpunkt der optischen Achse mit der Retina befindet sich ein Bereich auf der Retina, bezeichnet mit Fovea. Diese Fovea ist zwar nur ein ca. 1,5 mm großer Fleck, aber er konzentriert massiv fast ausschließlich Zäpfchenzellen in hohem Maße. Die restlichen Zäpfchenzellen verteilen sich fast gleichmäßig über die restliche Retina. Anders dagegen die Stäbchenzellen, die außerhalb der Fovea überwiegen und deren absolute Konzentration bei steigender Entfernung von der

optischen Achse abnimmt (s. Abb.6.1e).

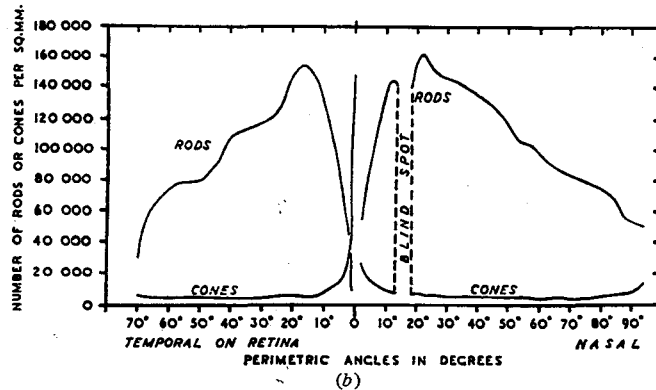
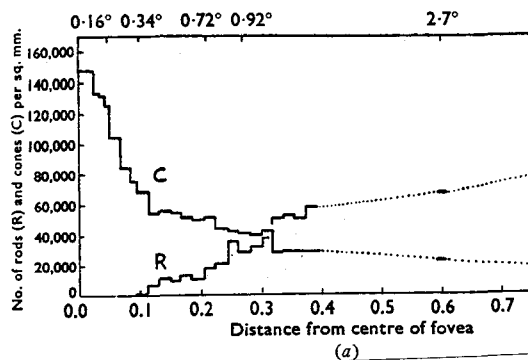


Abb.6.1e Verteilung der Stäbchen und Zäpfchenzellen

Diese Inhomogenität und Nicht-Linearität in der Verteilung der Fotorezeptoren läßt sich mit den Sehgewohnheiten und Seherfordernissen erklären. Zweifelsohne stellt ein Sehmechanismus, der bei einer begrenzten Zahl von Fotorezeptoren eine Möglichkeit hoher Sehschärfe (hohe Auflösung in der Fovea) enthält, verbunden mit einem Mechanismus zum Ausrichten (Augenbewegung) auf den Brennpunkt des Geschehens, einen Evolutionsvorteil dar.

Eine weitere Bearbeitung erfährt das empfangene Bildsignal durch die **Verarbeitungsschichten** der Retina. Sind die Anschlüsse der Zäpfchenzellen in der Fovea noch 1:1 den bipolaren Zellen zugeordnet, die wiederum genau einer Ganglionzelle zugeordnet ist, so sind es 20° von der optischen Achse bereits hunderte von Zäpfchen, die zu einer bipolaren Zelle die Signale senden. Bedenkt man, daß bei ca. 120 Millionen Fotorezeptoren auf der Retina der aus dem Auge herausführende Sehnerv nur eine Million Fasern hat, so findet augenscheinlich bereits in der Retina eine Informationsverarbeitung statt. Bei größerem Abstand der Fovea und damit kleinerer Auflösung scheint dies, vorgegeben durch die Anatomie, eine räumliche Integration der Bildsignale zu bedeuten, was auch Experimente mit Katzenfischen nahe legen. Eine andere Möglichkeit der Interaktion ist die Inhibition der Zellen untereinander, was eine Kontrastanhebung bewirkt.

## 6.2 Der optische Chiasmus (Überkreuzung)

Die optischen Impulse auf beiden Nervenbündeln, die die Augen verlassen, beschreiben erst zusammen die reale Welt. Um zusammengehörige Strukturen der Umwelt, von denen jeweils ein Teil in jedem Auge verarbeitet wird, auch zusammen auswerten zu können, werden "zusammengehörige" Nervenfasern von jedem Sehnerv in einem gemeinsamen Strang regroupiert. In Abb.6.2a ist dies skizziert. Zum Vergleich ist daneben das visuelle System des Frosches abgebildet, bei dem durch Überkreuzung der Sehnerven die Nachbarschaftsbeziehungen der Bildpunkte gewahrt bleiben.

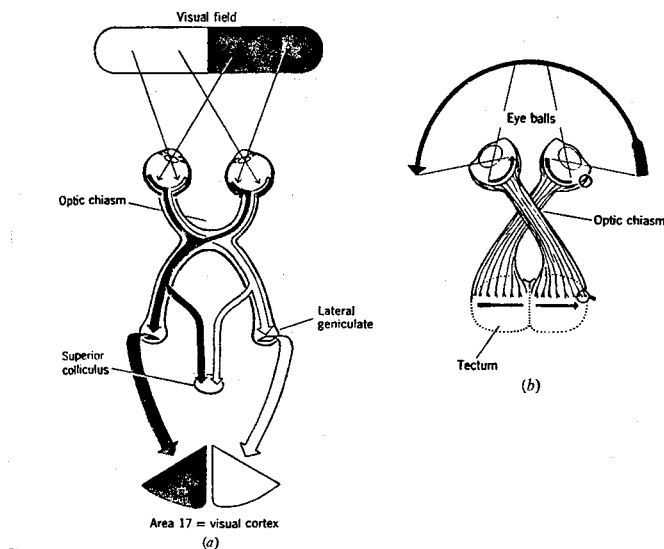


Abb.6.1.e Visueller Cortex des Menschen und des Frosches

### 6.3 Neurologische Kodierung und Informationsübertragung

Die Informationsübertragung im Nervensystem wird durch die Funktion der grundlegenden Einheiten, der Neuronen, bestimmt. Deshalb sollen im Folgenden kurz die Grundlagen dazu referiert werden.

Im Gehirn gibt es verschiedene Arten von Nervenzellen, die den folgenden prinzipiellen Aufbau gemeinsam haben. Betrachten wir dazu die Abbildung 6.3a.

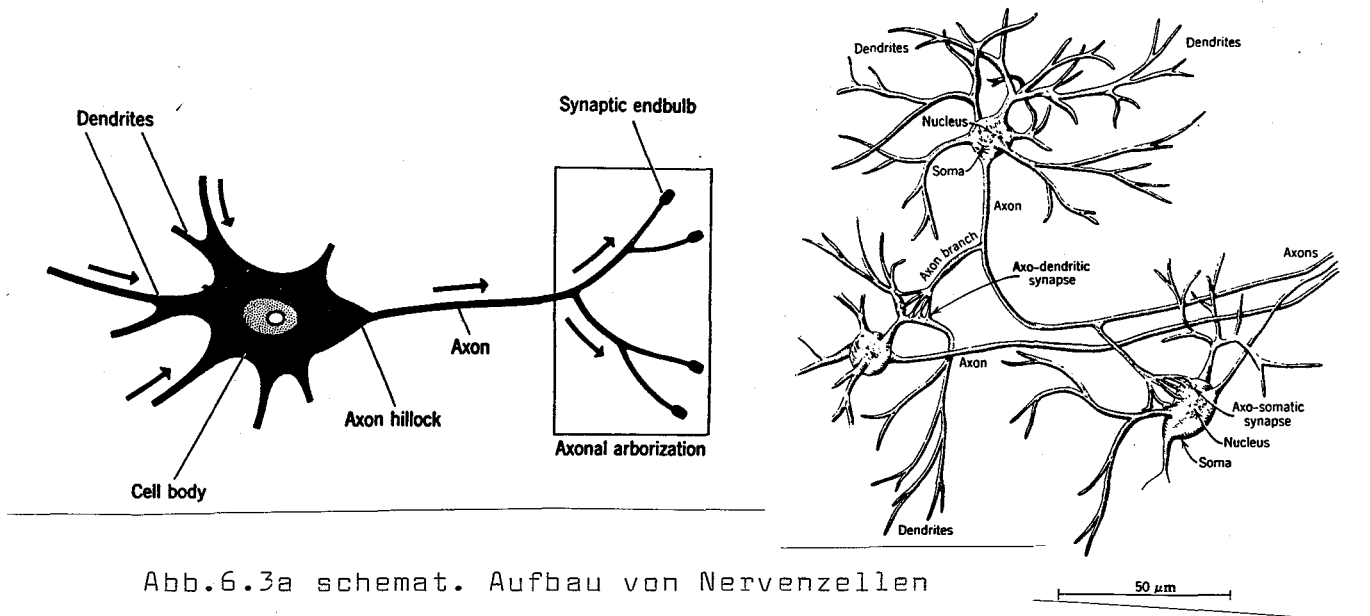


Abb.6.3a schemat. Aufbau von Nervenzellen

Im linken Teil der Abbildung ist der Informationsfluß im schematischen Aufbau einer Nervenzelle gezeigt. Die Zelle hat, abgesehen von dem obligatorischen Zellkern, verschiedene Verzweigungen. Diese Ausläufer haben verschiedene Aufgaben: Die Informationsaufnahme (Sensoren) wird durch ein feines Geflecht von "Dendriten" gewährleistet: die Informationen werden durch einen einzigen, meist dicker ausgeführten Strang, das "Axon", abgegeben. Die Leistung des Gehirns beruht auf im Wesentlichen auf einer Verbindung der Neuronen untereinander von Axon bzw. Axonauswüchsen zu den Dendriten, wie im rechten Teil der Abb. gezeigt ist.

### 6.3.1 Kodierung der Information

Die Information im Gehirn ist in verschiedenen physikalischen Energie- und Impulsformen kodiert. Betrachten wir dazu das Potential am Axon eines Neurons, das elektrisch durch eine eingeführte Mikroelektrode gereizt wird.

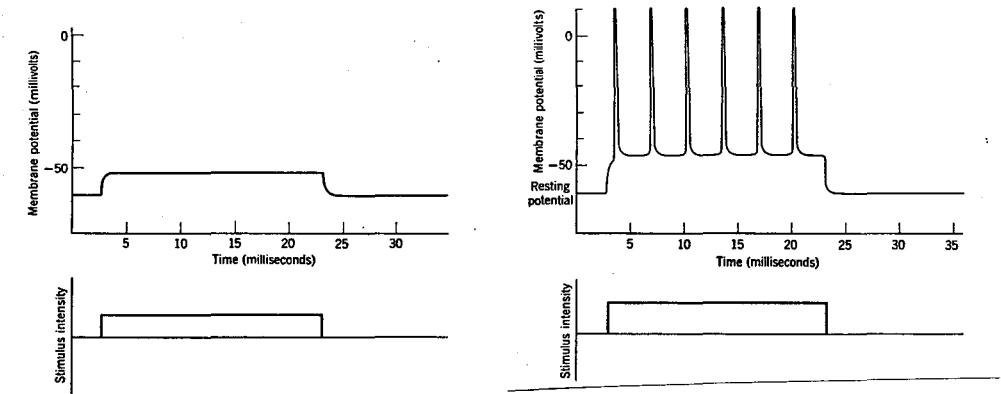


Abb.6.3b Schwellwertverhalten des Neurons

In Abbildung 6.3b ist das Antwortverhalten bei einer Reizstärke unterhalb eines Schwellwertes, in der Abbildung daneben bei Überschreiten des Schwellwertes dargestellt. Wie man sieht, erfolgt nach Überschreiten einer Schwelle  $\theta$  eine Impulsfolge, bei der die Frequenz und nicht die Amplitude von der Größe des Reizes abhängt. Die Abhängigkeit der Antwortfrequenz von der Reizstärke ist weitgehend linear (Abb.6.3c).

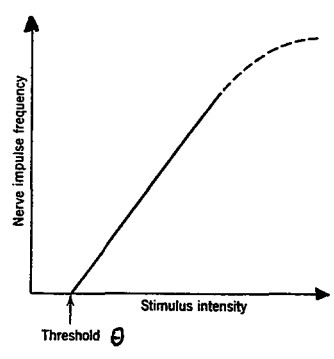


Abb.6.3c Antwortfrequenz als Funktion der Reizstärke

Dabei ist zu beachten, daß die bio-chemischen Regenerationsprozesse der Neuronen und Axonen durch das umgebende Gewebe eine gewisse Zeit brauchen, um wieder genügend Ladungsträger zu erzeugen und das ursprüngliche negative Potential aufzubauen. Dies ist ca. 1 ms, so daß ein Neuron mit maximal 1000 Hz "feuern" kann.

### 6.3.2 Die Übertragung der Information

Betrachten wir nun die Weiterleitung der Information. Zu jedem Axon gibt es eine Vielzahl von Dendriten (2000-16000), die eine Verbindung dazu haben. Diese Verbindungen zwischen den Außenhüllen zweier Nervenzellen heißen "Synapsen" und sind teilweise direkt elektrisch; in der Mehrzahl der Fälle allerdings wird die elektr. Energie in eine vermehrte Freisetzung von chemischen Transmittersubstanzen (z.B. Acetylcholin) umgesetzt,

die wiederum zu einem Potentialanstieg ("postsynaptic potential PSP") im Dendriten führt. Da durchschnittlich  $10^3$ - $10^5$  Synapsen an einem Neuron existieren, erreicht die Überlagerung der PSP den Neuronenkern gemittelt. Abbildung 6.3.d zeigt den Verlauf einer solchen Informationsübertragung.

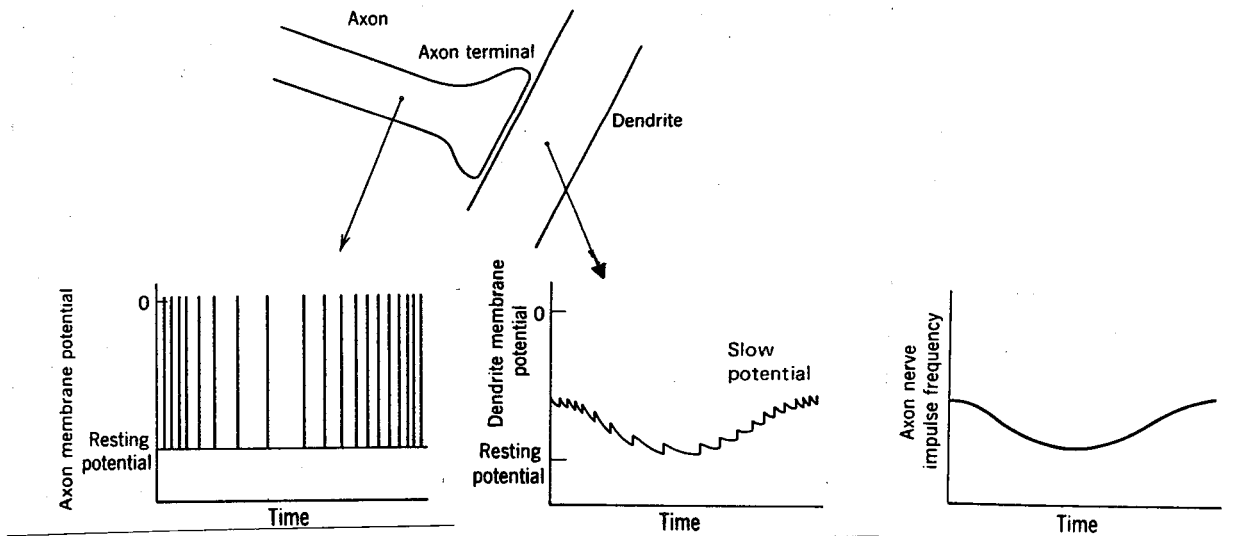


Abb.6.3d Übertragung der Information

Dabei ist noch nachzutragen, daß die Erzeugung der PSP nicht nur positiv sein kann ("exitatorische Synapse"), sondern auch negativ ("inhibitorische Synapse") oder in ihrer Größe multiplikativ von einer langsamen sich ändernden Axon-Amplitude ("präsynaptische Inhibition") abhängen kann.

### 6.3.3 Das Modell von McCulloch und Pitts

Wie wir im vorigen Abschnitt gesehen haben, ist die Frequenzmodulation nur eine intermediäre Zwischenform. Wollen wir das neuronale Verhalten modellieren, so reicht es sicher, für die Axonaktivität und die interneuronale Aktivität dimensionslose Konstante zu verwenden. Die Frequenzmodulation hat somit nur die Aufgabe einer Störungsunterdrückung, ähnlich der 0/5V Pegel der logischen Schaltungen in Computern.

Seien nun die Eingabe an den Dendriten mit  $x_1, \dots, x_n$  bezeichnet, die synaptischen Gewichte (Größe, Zahl etc der Synapsen) mit  $m_1 \dots m_n$  und die Axon-Aktivität mit  $y$ , so ist der Zusammenhang zwischen der Eingabe und der Ausgabe mit

$$y = \text{sign}\left(\sum_i m_i x_i - \theta\right)$$

gegeben. Das Funktionsdiagramm ist in Abb.6.3e wiedergegeben.

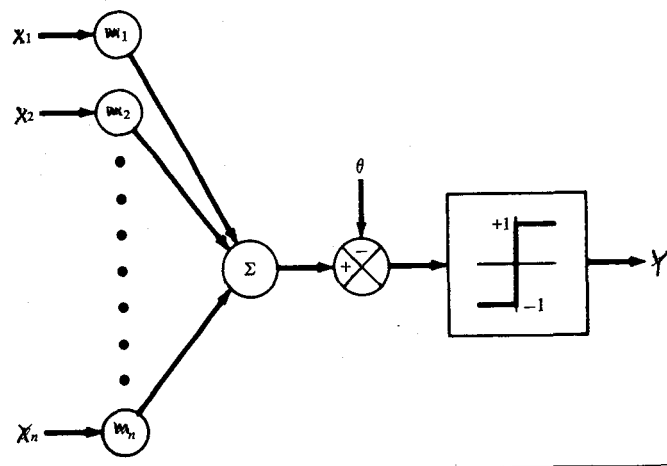


Abb.6.3e Das Modell von McCulloch und Pitts

Dieses Modell ist sehr einfach und simplifiziert die neurologischen Vorgänge erheblich. Beispielsweise wird der lineare Zusammenhang aus Abbildung 6.3c auf eine Stufenfunktion mit zwei Werten  $-1$  und  $+1$  reduziert. Trotzdem ist das makroskopische Verhalten damit modellierter Neuronennetze in wichtigen Teilen äquivalent zu dem naturgetreuerer Modelle.

#### 6.4 Die Empfindlichkeit des Sehsystems

Es ist bekannt, daß das menschliche Sehsystem einen sehr großen Bereich von Lichtintensitäten abdecken kann; von einem Photon pro Stäbchen bis zu einer relativen Intensität von  $10^{15}$  zu  $1$ . Dieses Verhältnis läßt sich nicht allein durch die Variation der Pupille (1:16) oder der direkten Umsetzung in Nervenimpulse (max 1:100) erklären.

Obwohl der genaue Mechanismus noch unbekannt ist, lassen sich doch einige Aussagen machen.

Zum einen scheinen Nichtlinearitäten der Rezeptorzellen das Signal zu komprimieren. Abb.6.4a zeigt das Antwortverhalten von 10 Zäpfchenzellen von Schildkröten bei Reizung mit der Lichtintensität  $I$ .

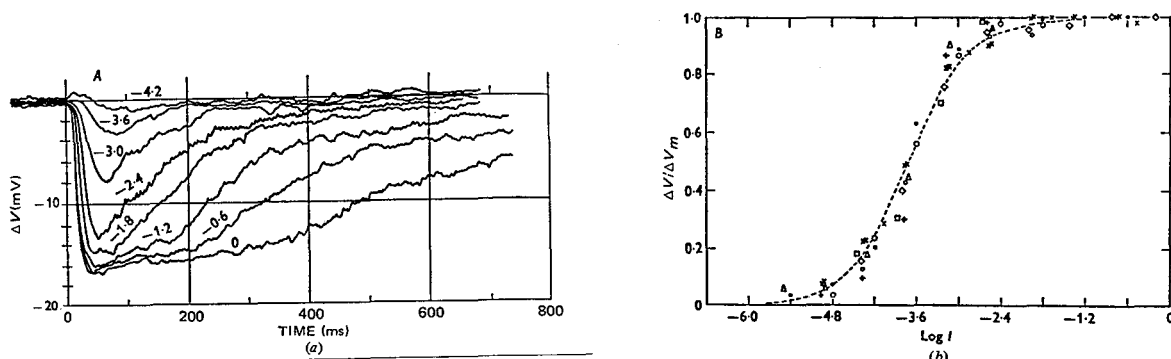


Abb.6.4a Antwortverhalten von Zäpfchenzellen

Die Änderung  $\Delta V$  im Membranpotential relativ zur stärksten, möglichen Änderung  $\Delta V_m$  ist dabei in weiten Bereichen (bis auf die Schwell- und Sättigungseffekte) proportional zu  $\log(I)$ . Dies entspricht auch dem psycho-physiologisch gefundenen Weber-Fechnerschem Gesetz, nach dem die subjektive Stärke der

Sinneseindrücke proportional zu  $\log(I)$  ist. Allerdings gibt es einen Unterschied: Ist die neurologisch erklärable Kompression der Intensitätsdynamik für menschliche Stäbchenzellen bei  $4.2 \log$  Einheiten, so ist in psycho-physiologischen Experimenten der weit höhere Wert von  $6.2 \log$  Einheiten gefunden worden. Dies läßt sich durch die für psycho-physiologische Experimente wichtige, zusätzliche neuronale Informationsverarbeitung erklären. In Abb.6.4b ist ein Modell für die Retina-Zellen abgebildet, das die Beeinflussung der Übertragungskennlinien durch Nachbarzellen beschreibt. Dabei resultiert links eine Nervenaktivität entsprechend der statischen Helligkeit, rechts durch die dynamische Kontraste.

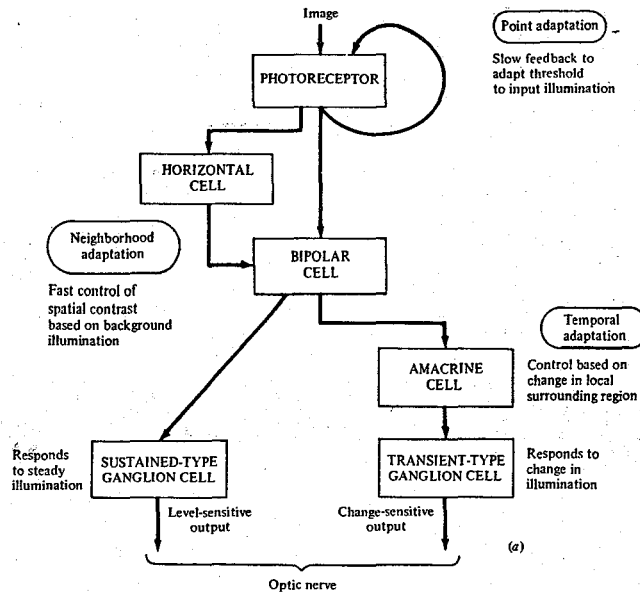


Abb.6.4b Regelmodell der Retina-Zellen

## Rezeptive Felder

In der Modellierung und Interpretation biologischer Sehvorgänge hat sich der Begriff der "Rezeptiven Felder" (receptive fields) gut bewährt. Seitdem es möglich ist, aus dem Bündel der optischen Nervenfasern eine Faser elektrisch anzuzapfen (Hartline 1940), zeigten verschiedene Experimente, daß die Ganglion-Zellen nicht nur empfindlich für Lichtreize an einer Stelle, sondern für ein kleines Gebiet im  $1^\circ - 2^\circ$  Sehwinkel, genannt "Rezeptives Feld", sind. Dies entspricht etwa der Fläche, die das Abbild einer Stecknadelkuppe in ein Meter Entfernung auf der Retina einnimmt. Nach der Art der Reaktion lassen sich die Ganglion-Zellen dabei in drei Kategorien einteilen:

Die ON-Zellen (20%) zeigen erhöhte Aktivität bei Licht im Zentrum des Rezeptiven Feldes,  
 die OFF-Zellen (30%) bei Dunkelheit und  
 die ON-OFF-Zellen (50%) bei Änderungen der Helligkeit.

Dabei verhält sich die zentrale Region gerade umgekehrt wie der Rest der Rezeptiven Felder. In Abb.6.4a ist das Aktivitätsverhalten von OFF und ON-units je nach Lage innerhalb eines Rezeptiven Feldes verdeutlicht.

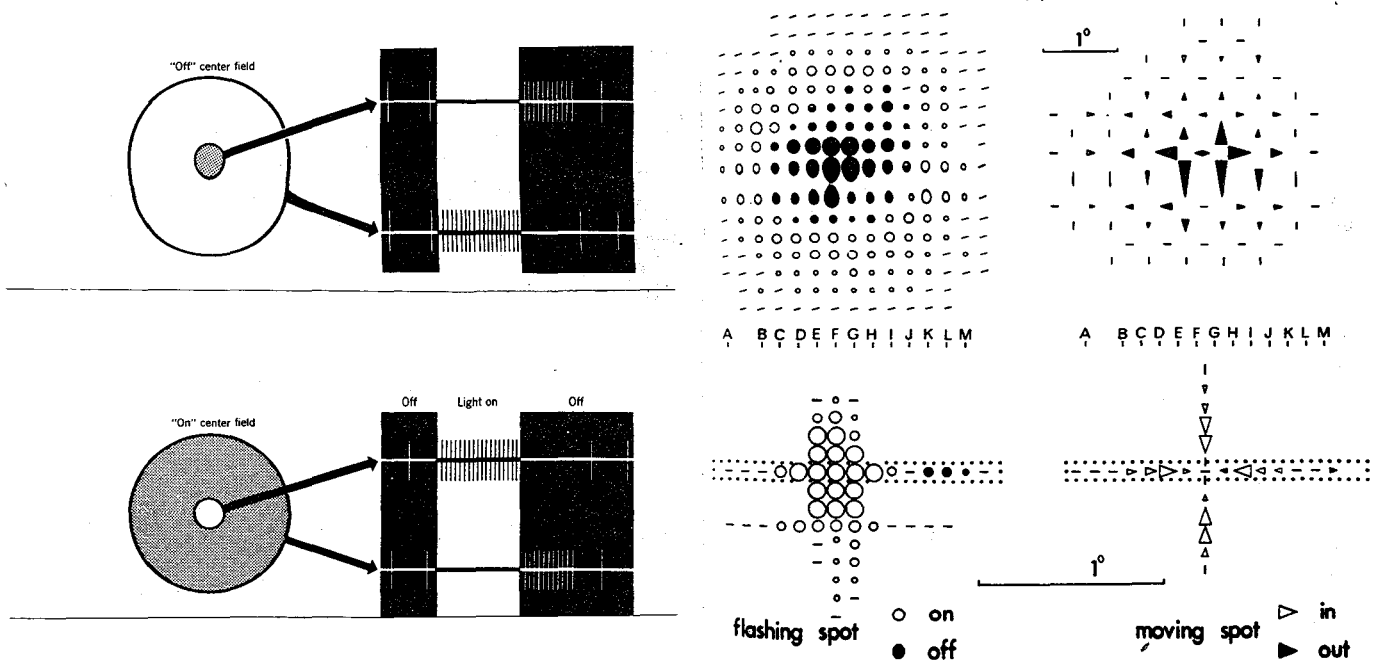


Abb.6.4a ON und OFF- units und Reaktion auf Reize

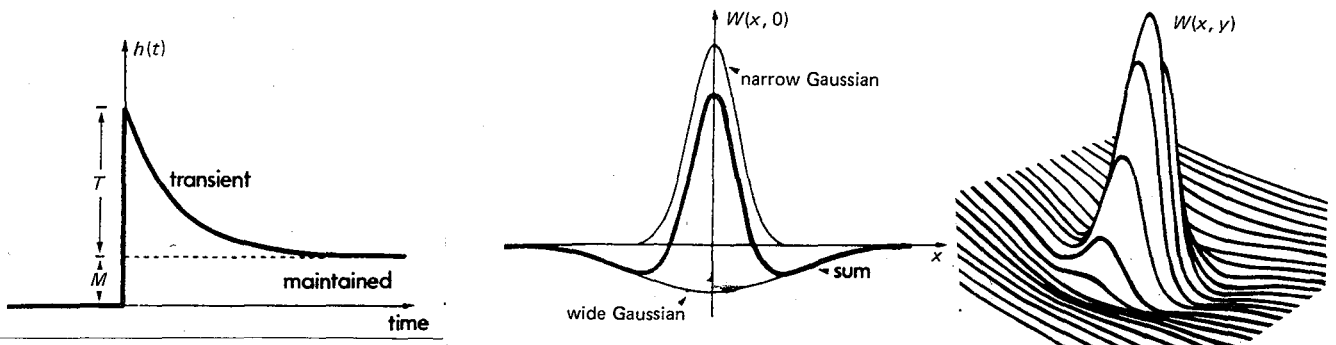
Wie man sieht, ist die Spikefrequenz bei Dunkelheit (im Unterschied zu Licht) nicht konstant. In Abb. b) und c) sind Reaktionen der Zelle bei einem Reiz blinkender und bewegter Lichtpunkt an verschiedenen Stellen des Rezeptiven Feldes angegeben.

Die Größe der Symbole (Kreis, Punkt, Dreieck) ist analog zu der ausgelösten Aktivität (klein, mittel, groß) eingezeichnet.

Nehmen wir an, daß die aktivitätsbeschreibende Funktion  $f(x,y,t)$  als Produkt einer Zeit und einer Ortsfunktion formuliert werden kann

$$f(x,y,t) = w(x,y)h(t),$$

so läßt sich die Funktion  $h(t)$  als Einschaltfunktion (s. Abb.6.4d) und  $w(x,y)$  als Überlagerung zweier Gauß-Funktionen beschreiben.



Wegen der Form (s. Abb.6.4d rechts) ist  $w(x,y)$  auch bekannt als Mexikanische Hut-Funktion. Interessanterweise ist die experimentell ermittelte "Reichweite" der Gauß'schen Funktionen gerade in der Größenordnung der Dendriten-Bäume der Ganglion-Zellen (!).

Bisher wurden neurophysiologisch drei Arten von Ganglion-Zellen



gefunden:

**X-Ganglion-Zellen:** kl. rezept. Feld, statische Helligkeit Konzentration im zentralen Retina-Teil. Sind verbunden mit dem lateralen geniculate Nucleus (LGN). Langsame Nervenleitgeschwindigkeit.

**Y-Ganglion-Zellen:** großes Rezeptives Feld. Reagiert auf Helligkeitsänderungen und ist verbunden zum LGN und Superior Colliculus.

**W-Ganglion-Zellen:** Unspezifisches Rezeptives Feld reagiert auf statische und dynamische Helligkeit. Ist verbunden mit dem Superior Colliculus und leitet Nervensignale sehr schnell.

Im Gegensatz dazu wurden von Hubert und Wiesel sog. einfache kontrast-sensitive Zellen gefunden. Abb.6.4e zeigt die Reaktion einer Kette von kontrast-sensitiven Zellen auf eine Kante verschiedener Orientierung. Man bemerke, daß eine Verlängerung der Kante keine Steigerung der Spikerate bewirkt, dagegen ein Verkürzen eine Abnahme. Eine kleine Unterbrechung der Linie bewirkt dabei praktisch nichts.

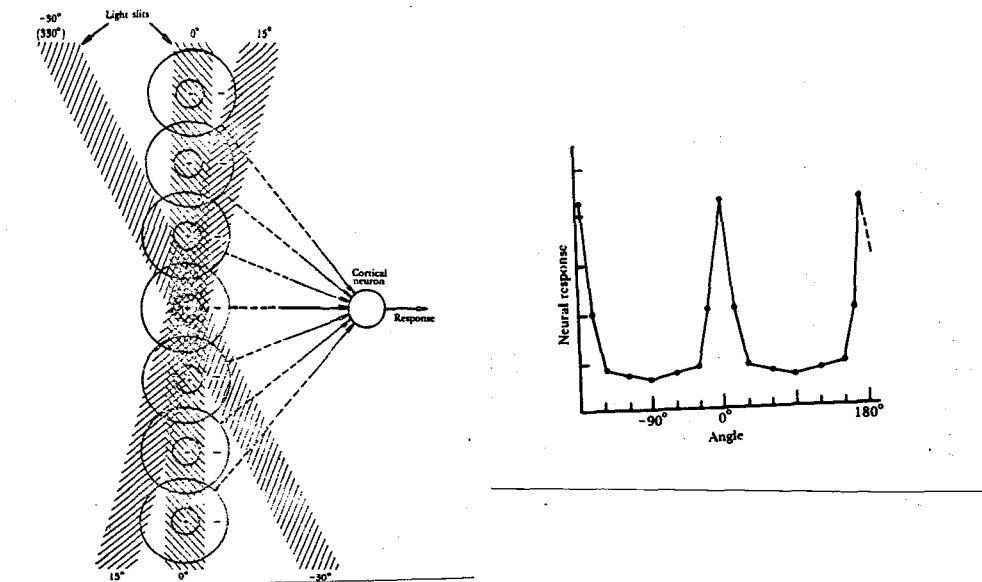


Abb 6.4e Reaktion einer Kette von Rezeptiven Feldern

In Abb.6.4f a, b, c, sind einige Eigenschaften eines Rezeptiven Feldes von Hyperkomplex Zellen, gemessen im visuellen Cortex der Katze gezeigt. Abb.6.4f-a) zeigt, daß es gleich ist, wo im Rezeptor-Feld der Reiz erfolgt, in b), daß die Reizauslösung von der Form des Reizes abhängt und in c), wie spezifisch ein 90°-Winkel von einer Hyperkomplexzelle detektiert wird.

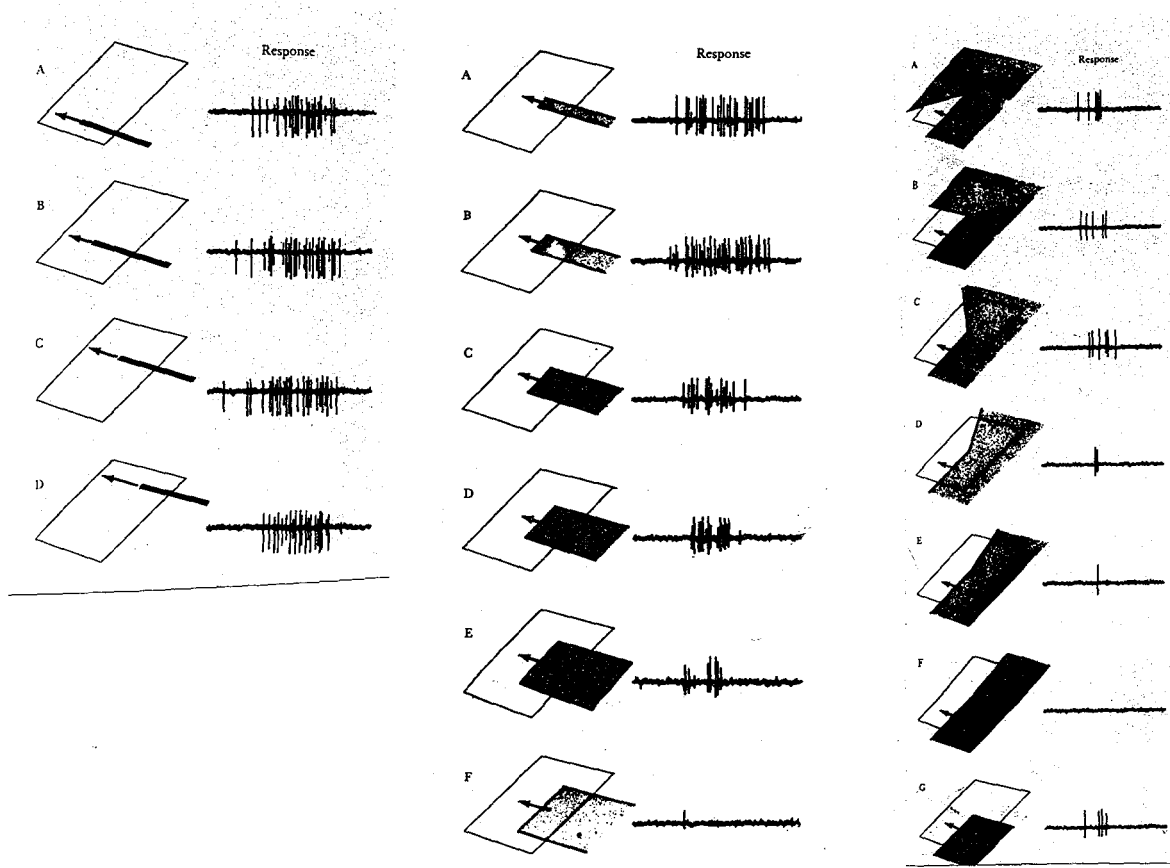


Abb 6.4f Eigenschaften von spez. Rezeptiven Feldern

Dabei scheint nur festzustehen, daß als Bildprimitive Ecken und Kanten abstrahiert werden, die verschiedene Orientierung und Bewegung haben. Wie diese Bildprimitive codiert und weiterverarbeitet werden ist unbekannt. Aus Experimenten mit einzuprägenden Objekten weiß man nur, daß unabhängig von der Komplexität man sich gerade 4-7 Objekte merken kann. Also existiert höchstwahrscheinlich eine interne Codierungsmöglichkeit für komplexe Muster, die bei den Bildprimitiven beginnt.

## 7.0 Gewinnung von Tiefeninformation

Das Erkennen von räumlicher Tiefeninformation nur mit Hilfe unseres Sehsystems ermöglicht uns, nicht nur, Objekte anhand der räumlichen Proportionen zu identifizieren, sondern auch eigenes Verhalten darauf einzustellen. Beispiel dafür ist der Kraftaufwand, einen Ball über eine Entfernung in ein Netz zu werfen.

Wie wird nun die Entfernungsinformation gewonnen?

Technische Systeme (Echolot, Kamera-Autofocus, Laser-entfernungsmesser) benutzen dazu meist einen aktiven Sender, der Schall-oder Lichtimpulse aussucht und aus der Verzögerungszeit zwischen ausgesandtem und reflektiertem empfangene Signal auf die Entfernung des Objekts schließen.

Eine unserem Sehsystem schon ähnlichere Methode benutzen zwei Bilder, die aus verschiedenen Aufnahmewinkeln heraus gemacht wurden, und schließt aus der Differenz der Bilder auf die Entfernung des Objekts. Dabei muß allerdings das System vorher mit bekannten Entfernungen und Objekten geeicht werden, da die verwendeten Linsensysteme, Kameratypen und Objektformen in die rein geometrischen Kalkulationen einbezogen werden müssen.

## 7.1 Gewinnung der Stereo-Bilder

Bei diesem Verfahren, das ohne aktiven Sender auskommt, gibt es nur eine Schwierigkeit zu bewältigen : Um die Abweichungen zwischen beiden Bildern feststellen zu können, müssen die beiden Bilder deckungsgleich übereinanderliegen. Will man dies bei maschinellen Bildern automatisch erreichen, so kann man beispielsweise die beiden Bilder solange verschieben, bis die Kreuzkorrelation beider Bilder besonders hoch ist. Dies ist aber eine sehr rechenzeitintensive Methode.

Betrachten wir das menschliche Sehsystem.

Wie in 6.2 ausgeführt wurde, werden im menschlichen Sehsystem die Nervenfasern, die Information aus korrespondierenden Teilen der Retina beider Augen transportieren, beim optischen Chiasmus zu gemeinsamen Bündeln zusammengefaßt und erreichen so den visuellen Cortex.

Diese Zuordnung der optischen Nerven ist aber, bedingt durch genetische und Wachstumsfaktoren, nicht besonders spezifisch; die exakte Codierung durch genetische Information wäre deshalb auch nicht sehr sinnvoll. Stattdessen werden die neurologischen Verbindungen in einer "Prägephase" in Abhängigkeit der Signale, die sie tatsächlich transportieren, verstärkt bzw. geschwächt. Wie aus Experimenten mit Katzen hervorgeht, gibt es dabei folgende Regeln:

Sei  $N_i$  ein Neuron mit der Aktivität  $x_i$  und  $Z_j$  eine nachgeschaltete Zelle mit der Aktivität  $y_j$ , die mit dem Kopplungsfaktor  $m_{ij}$  angekoppelt ist;  $Z_j$  kann auch durch andere Neuronen erregt werden.

$N_i$  aktiv UND  $Z_j$  aktiv  $\rightarrow$  höheres  $m_{ij}$   
 $N_i$  passiv UND  $Z_j$  aktiv  $\rightarrow$  geringeres  $m_{ij}$   
 $N_i$  aktiv UND  $Z_j$  passiv  $\rightarrow$  gleiches  $m_{ij}$   
 $N_i$  passiv UND  $Z_j$  passiv  $\rightarrow$  gleiches  $m_{ij}$

Kodieren wir "aktiv" mit "1" und "passiv" mit "0", so gilt für unser Modell aus Abschnitt 6.3.3 für die Änderung  $\Delta$  der Koppelungsfaktoren

$$\Delta m_{ij} = ax_i y_j - by_j \quad (7.1a)$$

und bei  $a=2$ ,  $b=1$

$$= (2x_i - 1)y_j$$

Die bei Katzen auf wenige Wochen beschränkte Prägephase ist dabei der große zeitliche Rahmen; Die Modifizierung der Kopplungen (Synapsenverbindungen) erfolgt außerdem nur in dem Zeitpunkt, wenn genügend visuelle Aufmerksamkeit (Motivation!) und die Stellung der Augäpfel "richtig" ist. Dann aber werden erhebliche Modifikationen ausgeführt, die erst das verstehende Sehen ermöglichen und nach Abschluß der Prägephase nicht mehr nachholbar sind. Dies ist auch der Grund, warum schielende Kinder nach Eintritt ins Schulalter stereoblind bleiben, egal ob sie operiert werden oder nicht.

Die korrekte Zuordnung der optischen Nerven des rechten und linken Auges wird also gelernt.

Wie wird nun das Stereobild gewonnen? Betrachten wir die rezeptiven Felder von 6.4, die die Funktionscharakteristik des Mexikanerhut-Operators haben. Wie wir in Abschnitt 3.4 sahen, erzeugt ein Operator mit der Mexikanerhut-Charakteristik ein Kanten-Bild, das im wesentlichen aus Kontrasten besteht. Wir können also davon ausgehen, daß im visuellen Cortex unter anderem auch solch ein Kantenbild vorliegt.

## 7.2 Der Stereo-Algorithmus

Wie läßt sich nun aus einem solchen Bild, das normal aus Schwarz-Weiß Punkten besteht, eine Tiefeninformation gewinnen?

Der Anlaß zu der Konstruktion eines parallelen, verteilten Stereoalgorithmus gaben die überraschenden Versuchsergebnisse von Julesz (Bell-Laboratorien), der fand, daß ein Stereo-Sehen nicht nur bei realen Grauwert-Bildern möglich ist, sondern auch bei Mustern aus zufällig verteilten Punkten. Dazu wird ein quadratischer Bereich aus einem Zufallspunktmuster herausgeschnitten und für die Stereobilder jeweils etwas nach links und etwas nach rechts verschoben wieder eingesetzt. Die entstandenen Lücken wurden, wie in der Abbildung gezeigt, mit Zufallspunkten gefüllt.

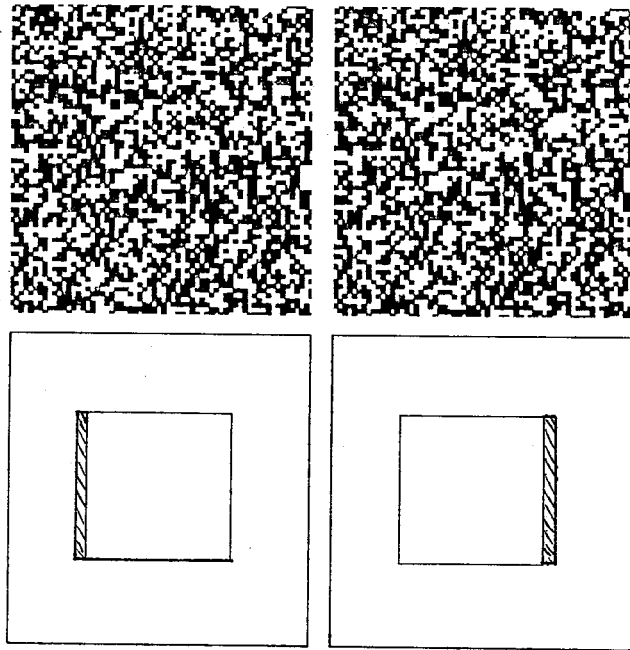
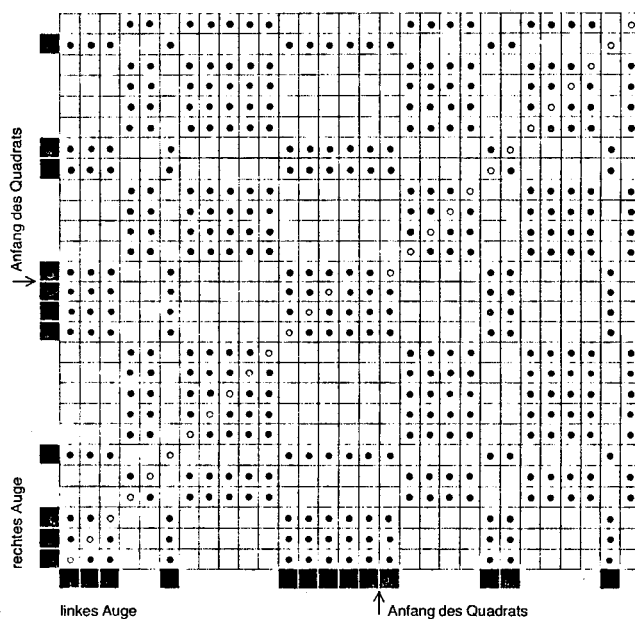


Abb.7.2a Zufallspunkt-Stereogramm von Julesz

Betrachtet man die beiden resultierenden Bilder mittels einer Stereobrille, so daß jedes Auge nur ein Bild sieht, so scheint das ausgeschnittene Quadrat über und unter der Bildfläche zu schweben.

Diese Entdeckung von Julesz veranlaßten D. Marr und T. Poggio zu folgenden Überlegungen, die 1976 zu ihrem bekannten Stereo-Algorithmus führten:

Angenommen, wir betrachten jeweils nur eine Zeile der Zufallsmuster und tragen diese als Koordinaten an einem Quadrat an. Bei jeder Zelle dieser Matrix wird ein Punkt "1" eingezeichnet, wenn die Randwerte übereinstimmen, nichts ("0") wenn sie nicht übereinstimmen und ein Kreis, wenn die Punkte sich entsprechen. Letzteres ist natürlich nur durch unser Wissen der Bildkonstruktion her möglich.



Wie wir sehen, bildet die Folge der korrespondierenden Punkte eine Diagonale, die im Bild an der Stelle versetzt weiterläuft, an der der Ausschnitt versetzt wurde. Die Kurvenschar aller parallelen Diagonalen gibt also die verschiedenen Abstände von der Bildebene an. Sind die beiden Bilder zu unterschiedlich, beispielsweise weit vor und hinter der Betrachtungsebene, so werden die beiden Bilder getrennt gesehen und nicht mehr zu einem Bild mit Tiefeninformationen fusioniert. In unserem Modell entspricht dies einer Begrenzung der Schar der Diagonalen auf einem Bereich um die Hauptdiagonale. In der folgenden Abbildung ist dies gezeigt.

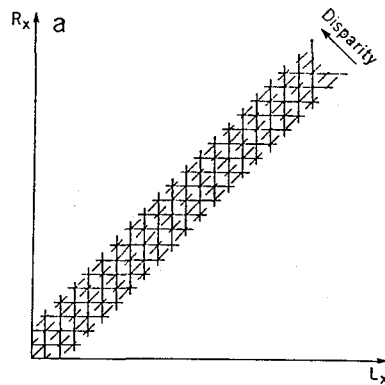


Abb. 7.2c Tiefeninformation und Diagonalen

Das Ziel des Algorithmus besteht nun darin, den Verlauf der Diagonalen mit allen Versetzungen (Bildtiefen) zu finden. Dies wird im Wesentlichen durch einen Relaxationsalgorithmus erreicht. Nehmen wir an, an jeder Zelle der Matrix in Abb. 7.2a sitzt eine neuronale Zelle, die mit ihren Nachbarn verschaltet ist und einen initialen Wert besitzt, der aus den entsprechenden Randwerten  $R_x$  und  $L_x$  mit  $C_{xy}^0 = R_x \oplus L_x$  bestimmt ist.

Die Grundidee des Algorithmus besteht darin, daß alle Zellen, die 1 sind, ihre diagonalen Nachbarn bestärken, und ihre nicht-diagonalen Nachbarn schwächen. Erweitern wir unsere Betrachtung auf die x-y-Ebene, so wird die Diagonale zur Fläche. Da die Schar der Diagonalen nun zur Ebenen parallel zur Betrachtungsebene geschichtet ist, ist ein Punkt zusätzlich zu x und y durch einen dritten Parameter festgelegt, z.B. dem Abstand d zur Betrachtungsebene.

Sei die Menge aller diagonalen Nachbarn  $N_s(x,y,d)$  eines Punktes  $(x,y,d)$  im Umkreis M dadurch festgelegt, daß sie angrenzen und verschiedenes x,y bei gleichem d haben. Die Menge der nicht-diagonalen Nachbarn, die entweder x oder y mit  $(x,y,d)$  gemeinsam haben, sei  $N_r(x,y,d)$ . Dann ist der Wert der Zelle bei der n+1-ten Iteration durch eine Schwellwertfunktion dreier Komponenten gegeben:

$$\text{Mit } S(z) = \begin{cases} 0 & z < T \\ 1 & z \geq T \end{cases}$$

$$\text{ist } C_{xyd}^{(n+1)} = S \left( \sum_{x',y',d' \in N_s(x,y,d)} C_{x'y'd'}^{(n)} - b \sum_{x',y',d' \in N_r(x,y,d)} C_{x'y'd'}^{(n)} + C_{xyd}^0 \right)$$

---

Der erste Term repräsentiert die Verstärkung, der zweite die Inhibition und der dritte den Startwert.

Beispielswerte, für die der Algorithmus schnell konvergiert, sind  $\theta = 3$ ,  $b = 2$ ,  $M = 5$ , wobei die "Füllung" der Stereogramme mit Punkten 50%-10% sein kann. Bei geringerer Füllung wird die Konvergenz langsamer, was aber dadurch behoben werden kann, daß als Funktion an die mittleren Zellenaktivität gekoppelt wird.

## 8.0 Das Lineare Kreuzkorrelations-Matrixmodell

In dem Modell von McCulloch und Pitts in Abschnitt 6.3.3 wird die Ausgabe zusätzlich zur gewichteten Summe der Eingabewerte auch von einem Schwellwert  $\theta$  abhängig.

Viele interessante Eigenschaften des Modells lassen sich mathematisch einfacher untersuchen, wenn das Modell bei einer Schwelle mit dem Wert Null betrachtet wird. Im Folgenden betrachten wir somit das sog. lineare Matrix-Modell der Neuronenaktivität.

Das lineare, hologische Matrix-Modell, besser auch Korrelations-Matrix Modell genannt, beruht auf dem linearen Zusammenhang zwischen der Aktivität aller Input-Einheiten  $(x_1 \dots x_n) = x$  und dem Output  $(y_1 \dots y_m) = y$

$$y = M x \quad (8.0)$$

mit der Matrix der Kopplungswerte  $M = (M_{ij})$ .

Abb.8.0a zeigt den schematischen Aufbau mit Modell-Neuronen.

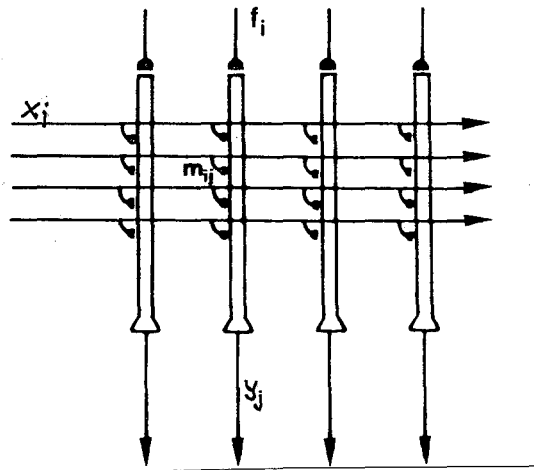


Abb.8.0a Aufbau des ass.-korr. Matrix-Speichers

## 8.1 Assoziative Speicherung

Wie werden nun die Werte von  $M$  ermittelt?

Folgen wir analog der Hebb'schen Regel, bei gleichzeitigem Auftreten von post- und präsynaptischer Erregung die Synapsenkopplung zu verstärken, so ist

$$\Delta M_{ij} = a y_i x_j \quad (8.1)$$

Dies ist eine vereinfachte Version von (7.1a).

Angenommen, es existieren nun  $p$  Reize  $x^1, \dots, x^p$  mit den Antworten  $y^1, \dots, y^p$ , so ist nach  $p$ -maliger Änderung die Matrix  $M$

$$M_{ij} = \sum_{k=1}^p a_k y_i^k x_j^k \quad x^T = x \text{ transponiert}$$



Bei Eingabe eines der gespeicherten Muster  $x^r$  ist der Output

$$y = M x^r = \sum_k y^k x^r x^k T$$

bzw. 
$$y_i = \sum_j M_{ij} x_j^r = \sum_k \sum_j a_k y_i^k x_j^k x_j^r \quad (8.1a)$$

$$= a_r y_i^r x^r x^r T + \sum_{\substack{k \\ k \neq r}} a_k y_i^k x^k x^r T \quad (8.1b)$$

Antwort + Übersprechen (Störung)

Bei orthogonalen Inputvektoren (d.h. sehr verschiedenen Reizen) fällt mit  $x^k x^r T = 0$ ,  $k \neq r$  das Übersprechen durch andere Reize weg, so daß bei der Normierung

$$x^k x^k T = 1/a_k$$

gilt

$$y = y^r.$$

Das lineare Matrix-Modell verhält sich also wie ein **assoziativer Speicher**; zu jedem Schlüsselwort  $x^k$  existiert ein Inhalt  $y^k$ . Der Vektor  $x$  ist dabei nicht eine numerische Adresse unter der  $y$  zu finden ist; der Adressraum ist z.B. nicht vollständig. Vielmehr wird die Korrelation zwischen  $x$  und  $y$  gespeichert und bei Eingabe von  $x$  das am besten dazu korrelierte  $y$  ausgegeben.

### Fehlerhafte Daten

Wie reagiert das System nun auf fehlerhafte, verfälschte Daten? Sei  $x$  ein unbekanntes Signal. Dann läßt sich  $x$  aufspalten in

$$x = x' + \tilde{x} \quad (8.1c)$$

wobei  $x'$  eine beliebige Linearkombination der  $x^1 \dots x^p$  und der Rest  $\tilde{x}$  das "Residuum" ist. Da jeder Vektor  $x^k$  entsprechend auf  $y^k$  abgebildet ist, wird auch der resultierende Output  $y$  eine Linearkombination der eingespeicherten  $y^1, \dots, y^p$  sowie des "Störterms"  $\tilde{y}$ :

$$y = M (x' + \tilde{x}) = M x' + M \tilde{x} = y' + \tilde{y} \quad (8.1d)$$

Ist die Abweichung  $\tilde{x}$  des Vektors  $x'$  von  $x$  nicht allzu groß, so läßt sich dies zur **Mustererkennung** verwenden. Ist die Zahl  $m$  der Klassen gleich  $\dim(y)$  und seien  $y^1 = (1, 0, \dots, 0)$ ,  $y^2 = (0, 1, 0, \dots, 0)$  usw. die Klassenprototypen, so ist bei dem Ausgabevektor  $y$  nach der Eingabe eines unbekanntes Musters  $x$  der Index der stärksten Komponente die Nummer der Klasse, der  $x$  zugerechnet werden muß.

Beispiele dazu zeigt die folgende Abbildung.

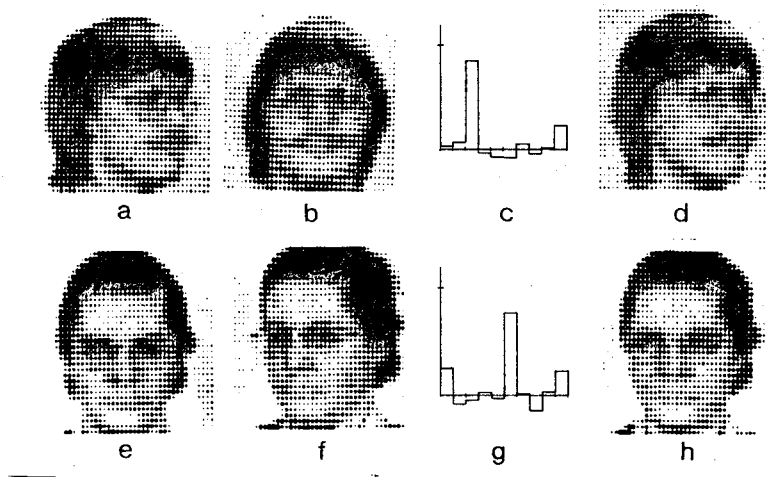


Abb.8.1a Mustererkennung mit der ass.Matrix

Dabei waren 10 Musterklassen gespeichert, die jeweils Fotos über eine Person aus 5 verschiedenen Blickwinkeln ( $-45^\circ$  zu  $+45^\circ$ ) enthielt.

Die unbekannt, nicht gespeicherten Fotos (d) und (h) wurden in digitalisierter Form eingegeben; das System erkannte richtig auf die Klassen 3 (Bild c) und 6 (Bild g).

## 8.2 Die Autoassoziative Matrix

Eine neue Situation ergibt sich, wenn das System rückgekoppelt wird, also,  $y = x$ . Dann ist

$$x = M x$$

Eine triviale Lösung dieser Gleichung wäre die Einheitsmatrix I

$$M = I$$

Eine interessantere Lösung ist bei nicht orthogonalen  $x^1, \dots, x^p$

$$\hat{M} = X X^+$$

wobei

$$X_{ij} = x_i^j$$

und  $\hat{M}$  ist der beste, lineare, unverschobene Schätzer und  $X^+$  die Pseudoinverse von X ist. Die Pseudoinverse einer Matrix X existiert auch dann, wenn Zeilen- oder Spaltenvektoren linear abhängig sind und deshalb  $X^{-1}$  nicht existieren kann. Es gilt

$$X X^+ = I \text{ bei lin.unabh.x} \quad (8.2a)$$

$$X^+ \text{ existiert, } \left\{ \begin{array}{l} X X^+ X = X \\ X^+ X X^+ = X^+ \\ X^+ X \text{ und } X X^+ \text{ sind hermitesch} \end{array} \right.$$

## Datenfehler

Damit ergibt sich für unsere gestörten Daten folgende interessante Proportion

$$\hat{M} x = X X^+ x = \hat{x} \quad (8.2b)$$

$\hat{x}$  ist dabei die eindeutig bestimmte, beste Linearkombination im Sinne des geringsten quadratischen Fehlers, die  $x$  beschreibt.

Für Formel 8.1d ergeben sich dabei neue Aspekte.

Sind die  $p$  Inputvektoren  $x_k$  linear unabhängig, so ist  $M$  gleich der Näherungslösung

$$M = \hat{M} = YX^+$$

und mit 8.2a ist

$$Y' = Mx' = YX^+x' = YX^+ \underbrace{XX^+}_{X} x' = M\hat{x} := \hat{y}$$

Da  $\hat{x} = \sum_k a_k x^k$

ist  $\hat{y} = M \sum_k a_k x^k = \sum_k a_k Mx^k = \sum_k a_k y^k$

Bei Eingabe eines beliebigen Inputvektors  $x$  wird also das Ausgangssignal auf die Linearkombination der  $y^1, \dots, y^p$  mit den Koeffizienten  $a_k$  der besten Linearkombination  $\hat{x}$  abgebildet.

Mit Hilfe der fehlerkorrigierenden Abbildung (8.2b) ist es möglich, aus einer fragmentarischen oder gestörten Version  $x$  eines  $x^T$  das vollständige, ungestörte  $x^T$  wieder zu rekonstruieren. Beispiele für Korrekturen von defekten Bildmustern sind in der folgenden Abbildung zu sehen.

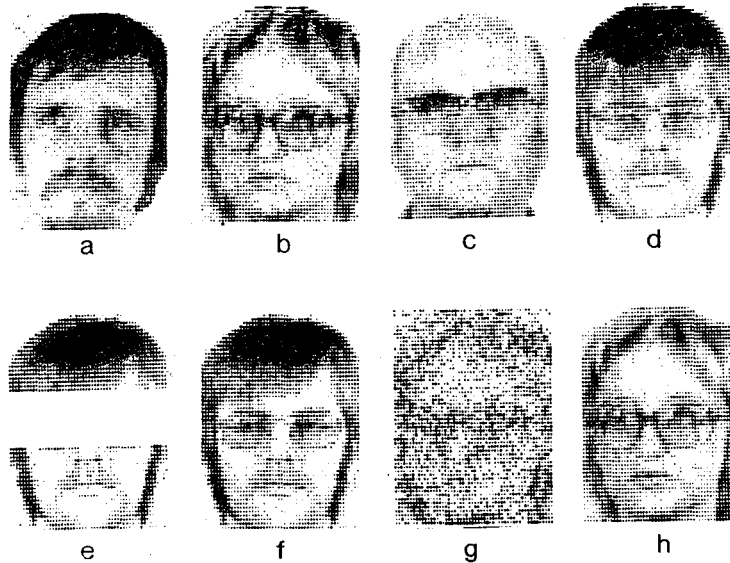


Abb.8.2a Ergänzung und Rekonstruktion von gestörten Bildern

Hierbei sind in (a)-(d) 4 von 100 digitalisierten, assoziativ gespeicherten Fotos gezeigt. Auf Präsentation von unvollständigen (e) oder gestörten (g) Fotos gibt die auto-assoziative Matrix das vollständige (f) bzw. ungestörte (h) Muster aus.

Die Ergänzungsoperation ist auch als schneller Zugriff auf eine **relationale Datenbank** vergleichbar. Beschreibe das Tupel (Relation, Objekt1, Objekt2) eine Relation zwischen Objekt1 und Objekt2, so wird bei der Eingabe des unvollständigen Tupels z.B. (Relation, Objekt1, -) dieses assoziativ ergänzt zum vollständigen Tupel. Damit ist eine Repräsentation von semantischen Netzwerken in paralleler, schneller Hardware möglich /HIN2/. Allerdings besteht dabei ein grundsätzlicher Unterschied zwischen der Abfrage einer relationalen Datenbank, wie sie mit konventionellen assoziativen Speicherchips realisiert ist und der assoziativen Ergänzung des Kreuzkorrelations-Matrix-Speichers: Sind mehrere Tupel abgespeichert, beispielsweise (REL, OBJ1, OBJ2) und (REL, OBJ1, OBJ3) und es wird nach allen Tupeln mit gemeinsamen Eigenschaften gefragt, beispielsweise (REL, OBJ1, -), so liefert die relationale Datenbank sequentiell alle Werte OBJ2, OBJ3, die diese Relation erfüllen; der lineare Matrixspeicher aus Abschnitt 8.2 dagegen gibt eine Überlagerung der Kodierungen für OBJ2 und OBJ3. Deshalb ist der Matrix-Speicher als Inferenz-Maschine für Sprachen wie PROLOG in dieser Form nicht unmittelbar geeignet.

### Hardware-Fehler

Nicht nur bei defekten Daten weist der autoassoziative Matrix-Speicher interessante Proportionen auf; diese Eigenschaften gelten auch - analog zum holographischen Modell- auch für die zu Grund liegende Hardware. Der Ansatz in Abschnitt 8.0 ging von einer vollständigen Matrix  $M_{ij}$  aus, d.h. einer Kopplung von allen Einheiten zu allen Einheiten. Werden Inputleitungen gestört, so ist dies gleichbedeutend mit einem Kurzschluß oder einer Leitungstrennung von Einheiten innerhalb der Matrix. Somit wird auch bei dem Ausfall von Einheiten ein korrektes Ergebnis erreicht. Dies untersuchte Kohonen genauer und fand, daß ein Faktor von ca 40 ausreicht zwischen der Zahl der Signalleitungen ( $=\dim(x)$ ) und der Zahl der Kreuzungspunkte ( $=\text{Zahl der } M_{ij} \neq 0$ ). Dies wurde auch bei anderen Computersimulationen bestätigt.

### 8.3 Der Novelty-Filter

Sind unsere Ausgangsvektoren  $x^1 \dots x^p$  orthogonal, so stellt der Residuumsvektor  $\tilde{x}$  von Gleichung (8.1c) den Anteil der Eingabe  $x$  dar, der nicht in  $M$  gespeichert ist und auch nicht als Linearkombination der  $x^1 \dots x^p$  darstellbar ist, also das "Neue" an der Eingabe.

Wie läßt sich  $\tilde{x}$  errechnen ?

Es ist mit (8.1c)

$$\tilde{x} = x - \hat{x} = x - Mx = (I-M)x = M'x \quad (8.3a)$$

Die Differenz entspricht im neurologischen Bereich der Wirkung von

hemmenden (inhibitorischen) Synapsen und läßt sich in das mit ursprünglich excitatorischen Synapsen ausgestattete Matrixmodell hineinnehmen. Mit (8.1a) ist

$$M_{ij}' = (I-M)_{ij} = d_{ij} + \sum_k -x_i^k x_j^k \quad d_{ij} = \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases}$$

Die Änderung der Kopplung bei der Speicherung des k-ten Musters ist dann

$$\Delta M_{ij}'(k) = -x_i^k x_j^k$$

entsprechend einer inhibitorischen Synapse. Die Matrix  $M'$  ist als "Novelty-Filter" von Kohonen bezeichnet worden. Beispiele für die Wirkung des Novelty-Filters sind in der folgenden Abbildung zu sehen.

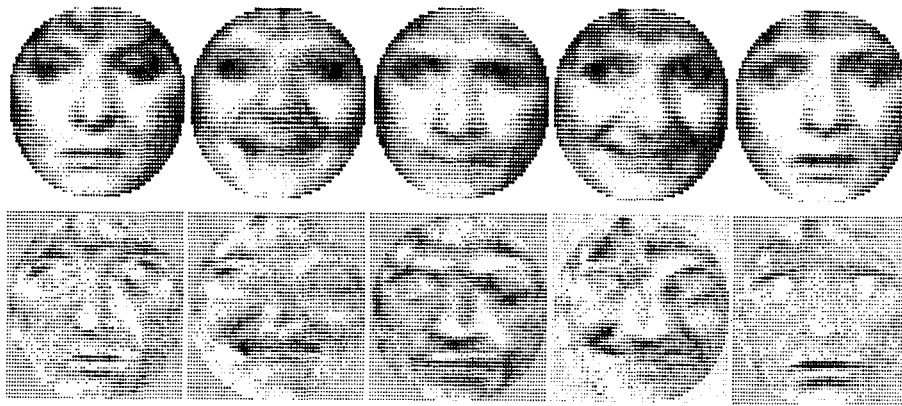


Abb.8.3a "Neue" Aspekte von eingegebene Gesichtern

Auf einen wesentlichen Aspekt des Novelty-Filters muß noch gesondert hingewiesen werden. Wie aus Abschnitt 8.2 hervorgeht, ist eine "optimale autoassoziative Abbildung" (Ergänzung) möglich, wenn die  $x^1 \dots x^p$  orthogonal zueinander sind. Um dies für die Beispiele in Abb.8.2a zu erreichen, mußten sämtliche Muster  $x^1 \dots x^p$  vorher einem sog. Gram-Schmidt Orthogonalisierungsprozeß unterworfen werden.

Betrachten wir diesen Prozeß etwas näher.

Angenommen, ein Vektor  $x^1$  sei gegeben und ein weiterer Vektor  $x^2$  wird dazugenommen. Der zweite Vektor, der orthogonal auf  $x^1$  steht, läßt sich mit

$$x^2 = x - \frac{xx^1}{|x^1|^2} x^1$$

bestimmen.

Der Beweis dafür ist einfach: Multiplizieren von rechts mit  $x^1$  führt zu

$$x^1 x^2 = xx^1 - xx^1 = 0$$

Dies läßt sich auf eine Sequenz  $x^1 \dots x^k$  ausdehnen:

Ein Vektor  $x$  mit  $\dim(x) > k$  wird zum Errechnen eines weiteren

Basisvektors eingesetzt mit

$$x^{k+1} = x - \sum_{j=1}^k \frac{1}{|x^j|^2} x x^j x^j \quad (8.3b)$$

Der Beweis dafür ist analog zum vorherigen: Für alle  $x^i$ ,  $i=1..k$  ergibt ein Multiplizieren an 8.3b von rechts jeweils Null.

Mit den Abkürzungen

$$a_j := 1/|x^j|^2 \quad \text{und} \quad \tilde{x} := x^{k+1}$$

wird aus Gleichung 8.3b

$$\tilde{x} = x - \sum_{k=1}^p a_k (x^k x) x^k = x - M x = x - \hat{x}$$

die bekannte Gleichung 8.3a.

Damit ist gezeigt, daß der Residuumsvektor orthogonal zu  $x^1 \dots x^p$  ist und mit ihm eine Basis bildet.

Der Novelty-Filter ist somit ein Operator zur Vorverarbeitung der Signale, der mit dem Mechanismus der autoassoziativen Matrix die Orthogonalisierung von beliebigen Eingangsvektoren vornimmt und damit eine optimale assoziative Speicherung ermöglicht. Die Orthogonalisierungsoperation ist dabei eine derartige Umcodierung der Eingangsvektoren, daß die resultierenden Vektoren die für den Eingangsvektor gegenüber den anderen, bereits gespeicherten Vektoren "typischen" Merkmale kodieren; also eine Extraktion von "Features". Der n-dimensionale Zustandsraum ( $\dim(x)=n$ ) wird damit auf den nur tatsächlich nötigen Unterraum der Features mit den Basisvektoren  $x^1 \dots x^p$  reduziert. Hier zeigt das autoassoziative Matrixmodell einen eleganten Ansatz für das Problem in der Mustererkennung der automatischen Sprach- und Bilderkennung, aus den Daten problemspezifische Features zu generieren.

## Kurze Liste der Literatur zur Bildverarbeitung

### Lehrbücher:

F.W.Wahl  
Digitale Bildverarbeitung  
Springer Verlag 1984

M.D. Levine  
Vision in man and machine  
McGraw Hill 1985

D.H.Ballard,Ch.Brown  
Computer Vision  
Prentice-Hall 1982

### außerdem:

T.Kohonen  
Self-Organisation and Associative Memory  
Springer Verlag 1984

Hinton, Anderson  
Parallel Models of Associative Memory  
Lawrence Erlbaum 1981

H.Bunke  
Modellgesteuerte Bildanalyse  
Teubner Verlag 1985 Stuttgart

Spektrum der Wissenschaft  
6/84 S.114      Stereo-Algorithmus  
2/85 S.110      Rotation von Körpern in der Vorstellung  
5/86 S.78        Vorstellen und Wahrnehmen  
8/86 S.104      Scheinbewegungen