Übersicht der Vorlesung

1. Einführung
2. Bildverarbeitung
3. Morphologische Operationen
4. Bildsegmentierung
5. Merkmale von Objekten
6. Klassifikation
7. Dreidimensionale Bildinterpretation
8. Bewegungsanalyse aus Bildfolgen
9. PCA (Hauptkomponentenanalyse)
10. ICA (Independent Component Analysis – Unabhängigkeitsanalyse)

4 Bildsegmentierung

4. Bildsegmentierung

4.1 Einführung

4.2 Punktorientierte Segmentierung

4.3 Mathematische Grundlagen

4.4 Bestimmung von Komponenten

4.5 Regionenorientierte Segmentierung

4.6 Kantenorientierte Segmentierung

4.7 Kantenverfolgung

4.8 Gebietsnachbarschaftsgraph

4.9 Modellabhängige Verfahren zur Segmentierung

4.1 Einführung

Einführung

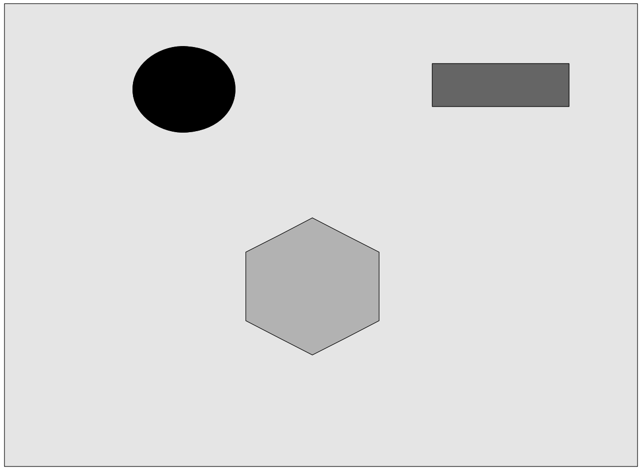
- Bildsegmentierung bezeichnet den Vorgang, das Bild in sinnvolle Bildteile (Segmente) aufzuteilen.
- Die Segmente bilden dann die Grundlage für die Erkennung und Klassifikation von Objekten in Bildern.

Einführung

Ziele der Segmentierung eines Bildes:

- Auffinden einer Menge von Segmentierungsobjekten, von denen sich jedes Objekt durch bestimmte homogene Eigenschaften oder Attribute auszeichnet, bzw. eine bestimmte Relation erfüllt (gleicher Grauwert).

- Auffinden von Kanten (Objektgrenzen), die durch Änderungen in ansonsten gleichmäßigen Mustern entstehen. Verschieden helle Grauwertbereiche lassen sich beispielsweise durch Kanten voneinander trennen.



4 Segmente

Einführung

Verfahren:

- punktorientierte Verfahren
- kantenorientierte Verfahren
- regionenorientierte Verfahren
- regelbasierte Verfahren

Unter einer Segmentierung von G_E verstehen wir zunächst eine Zerlegung von G_E, d.h.:

$$G_E = \bigcup_{n=1}^{N} X_n$$

$$X_{n_1} \cap X_{n_2} = 0 \quad \textit{für} \;\; n_1 \neq n_2$$

zusammenhängend

in sich homogen

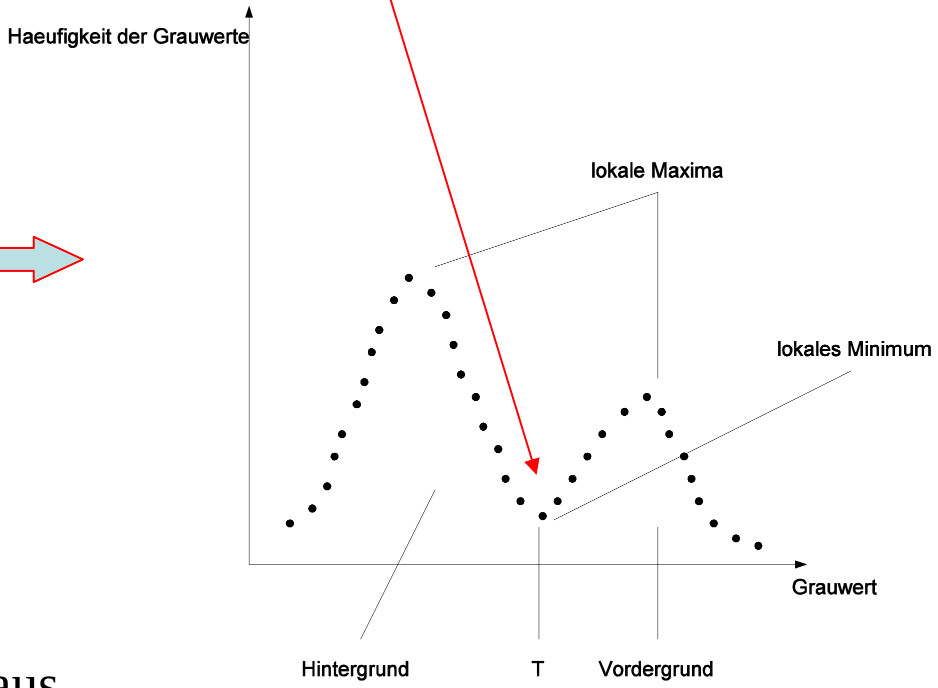
benachbarte unterscheiden sich

4.2 Punktorientierte Segmentierung

Schwellwertverfahren

$$g_A(i,j)=\begin{cases}1 & falls\ g_E(i,j)\geq T\\ 0 & falls\ g_E(i,j)<T\end{cases}$$

Histogramm



Bei mehreren Objekten mit verschiedenen Grauwerten ergibt sich ein multimodales Histogramm und ein einziger Schwellwert reicht nicht mehr aus.

Modifikationen

$$g_A(i,j) = \begin{cases} 1 & \text{falls} \quad g_E(i,j) \in D \\ 0 & \text{sonst} \end{cases} \qquad D \subseteq R$$

$$g_A(i,j) = \begin{cases} k & \text{falls} \quad g_E(i,j) \in D_k \\ 0 & \text{sonst} \end{cases} \qquad D_k \subseteq R \qquad k = 1, \ldots, N$$

Bemerkungen

- bei gutem Kontrast zwischen Vorder- und Hintergrund
  - Stanzteile (hell, reflektierend) auf einem dunklen Förderband

  liefert es gute Ergebnisse
- reicht aber oft nicht aus

4.3 Mathematische Grundlagen

Mathematische Grundlagen

Für die diskreten Bilder benötigen wir Definitionen von:

- Nachbarschaft von Bildpunkten
- Wege von Bildpunkten
- zusammenhängendes Segment
- benachbarte Bildsegmente
- Kanten

4.3.1 Nachbarschaft

Nachbarschaftsstruktur

Tupel: $\boxed{[P, N]}$

nichtleere Menge

$$N \subset P \times P$$

irreflexive, symmetrische Relation:

$$(p, p) \notin N \quad \forall p \in P$$

$$(p,q) \in N \;\Rightarrow\; (q,p) \in N \quad \forall p,q \in P$$

Das Tupel *[P,N]* ist ein ungerichteter Graph.

Nachbarschaft – isolierter Punkt

Nachbarschaft des Punktes *p*:

$$\boxed{N(p) = \{q : q \in P, (p,q) \in N\}}$$

$q \in N(p)$ → Nachbar von p

Isolierter Punkt: $\boxed{N(p) = \emptyset}$

Beispiel – Bildraster

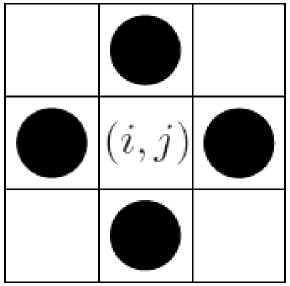
$$P = \{(i, j): \ i = 0, \ldots, I - 1, \ \ j = 0, \ldots, J - 1\}$$

Anzahl der Zeilen   Anzahl der Spalten

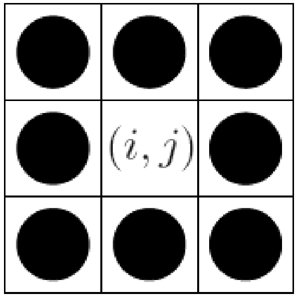
Für N gibt es mehrere Möglichkeiten

4 und 8 Nachbarschaft im Bildraster

$$N = N_4 = \left\{ \left[(i_1, j_1),(i_2, j_2)\right] : |i_1 - i_2| + |j_1 - j_2| = 1 \right\}$$

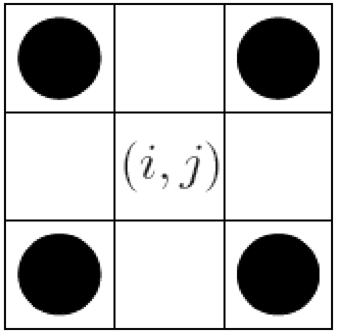


$$N = N_8 = \left\{ \left[(i_1, j_1),(i_2, j_2)\right] : \max\left\{ |i_1 - i_2|, |j_1 - j_2| \right\} = 1 \right\}$$



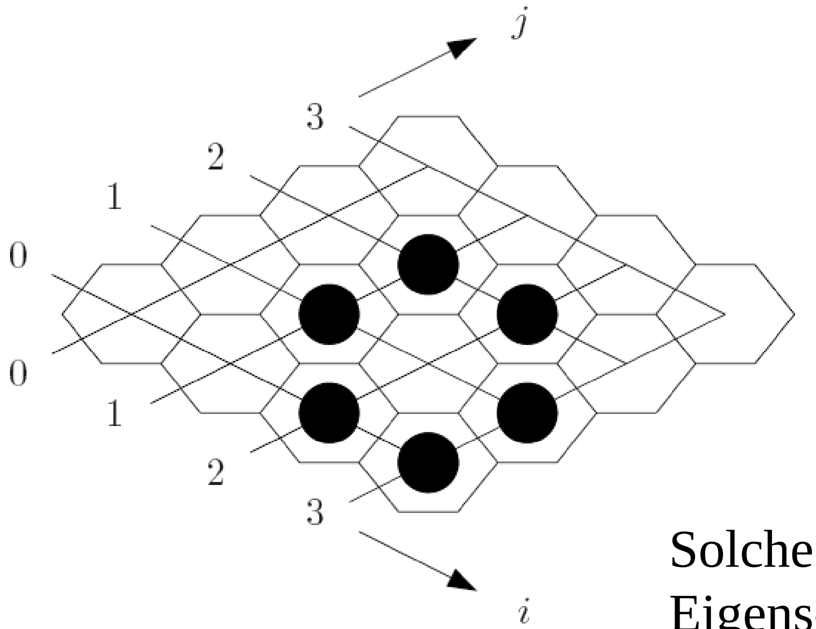
Diagonal – Nachbarschaft

$$N = N_D = \left\{ \left[(i_1, j_1), (i_2, j_2) \right] : |i_1 - i_2| = 1 \text{ und } |j_1 - j_2| = 1 \right\}$$



6 – Nachbarschaft

$$N = N_6 = \left\{ \left[(i_1, j_1),(i_2, j_2)\right] : |i_1 - i_2| + |j_1 - j_2| = 1 \text{ oder } \left(|i_1 - i_2| + |j_1 - j_2| = 2 \text{ und } i_1 + j_1 = i_2 + j_2 \right) \right\}$$



Interpretation des Bildrasters als regelmäßige Sechsecke.

Solche Hexagonalraster haben viele interessante Eigenschaften. Wir werden aber im Weiteren nur noch quadratische Bildraster betrachten.

4.3.2 Weg und Zusammenhang

Weg

Nachbarschaftsstruktur: $[P, N]$

$M \subseteq P \qquad n \geq 1$

Weg in M: $\boxed{(p_1, \ldots, p_n)}$ $\qquad p_i \in M \qquad i = 1, \ldots, n$

$\boxed{(p_i, p_{i+1}) \in N}$ $\qquad i = 1, \ldots, n-1$

$n = 1 \qquad (p_1) \qquad$ Weg ist nur ein Punkt

Verbundenheitsrelation

Nachbarschaftsstruktur: $[P,N]$

$p,q \in P$ verbunden bezüglich $M \subseteq P$

$$\exists \text{ Weg } (p,\ldots,q) \text{ in } M \text{ oder } \exists \text{ Weg } (p,\ldots,q) \text{ in } \overline{M} = P \setminus M$$

$$V_M = \{(p,q) \in P \times P : p \text{ und } q \text{ sind bezüglich } M \text{ verbunden}\}$$

Zusammenhang

Nachbarschaftsstruktur: $[P, N]$

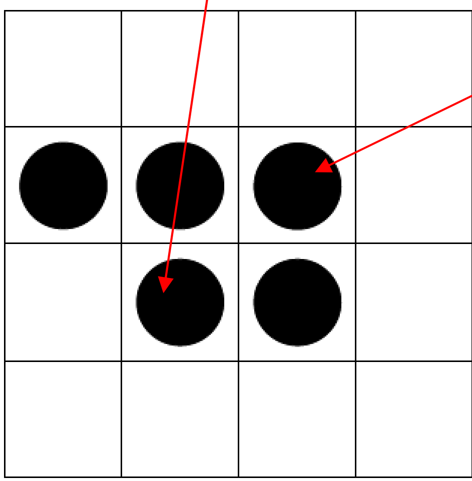
$M \subseteq P$ heißt zusammenhängend (Gebiet)

$$\boxed{\forall p,q \in M \Rightarrow (p,q) \in V_M}$$

Beispiel

$$P = \{(i, j) : \ i = 0, \ldots, 3, \ \ j = 0, \ldots, 3\} \qquad N = N_4$$

$$M = \{(1,1), (1,2), (2,0), (2,1), (2,2)\}$$



M ist zusammenhängend

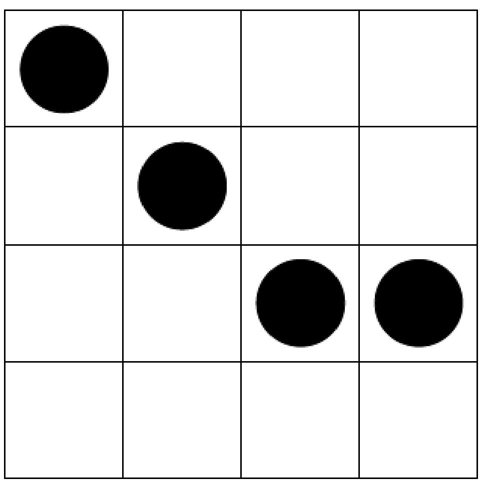
Bemerkung

- solche Nachbarschaftstrukturen kann man sich auch als binäre Bilder vorstellen
- die Teilmenge M interpretieren wir als die Menge der Bildpunkte mit $g(i,j)=1$

Beispiel

$$P = \{(i, j) : \; i = 0,\ldots,3, \;\; j = 0,\ldots,3\}$$

$$M = \{(1,2), (1,3), (2,1), (3,0)\}$$



$N = N_4$ *M* ist nicht zusammenhängend

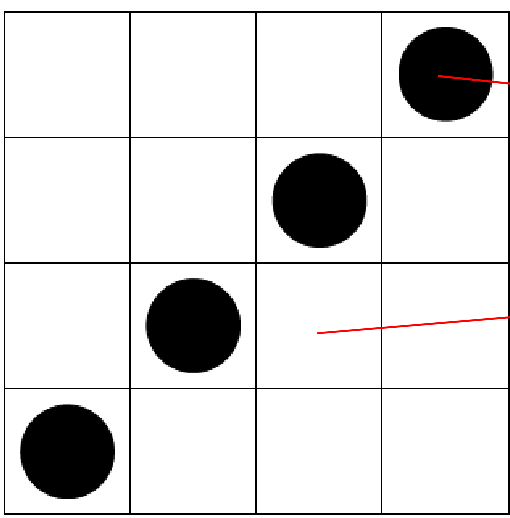
$N = N_8$ *M* ist zusammenhängend

Dies zeigt, dass der Begriff zusammenhängende Menge von der Definition der Nachbarschaft *N* abhängt.

Beispiel

$$P = \{(i, j) : i = 0,\ldots,3, \;\; j = 0,\ldots,3\}$$

$$M = \{(0,0), (1,1), (2,2), (3,3)\} \qquad\qquad N = N_8$$



M ist zusammenhängend

$P \backslash M$ ist zusammenhängend

(widerspricht unserer Anschauung)

bei der 6 – Nachbarschaft treten
diese Probleme nicht auf

4.3.3 Komponentenzerlegung

Zerlegung

Nachbarschaftsstruktur: $[P, N]$

$$Z = \{G_1, G_2, \ldots, G_n\}$$

$$\bigcup_{i=1}^{n} G_i = P$$

$G_i \subseteq P$

$G_i \cap G_j = \emptyset,\ i \neq j$

zusammenhängend

Eigenschaften der Verbundenheitsrelation

$$V_M = \{ (p,q) \in P \times P : p \text{ und } q \text{ sind bezüglich } M \text{ verbunden} \}$$

$\exists$ Weg $(p,\dots,q)$ in M oder $\exists$ Weg $(p,\dots,q)$ in $\overline{M} = P \setminus M$

$$V_M \subseteq (M \times M) \cup (\overline{M} \times \overline{M}) \qquad \overline{M} = P \setminus M$$

$$(p,p) \in V_M$$

$$(p,q) \in V_M \Rightarrow (q,p) \in V_M$$

$$((p,q) \in V_M \wedge (q,r) \in V_M) \Rightarrow (p,r) \in V_M$$

Äquivalenzrelation

Äquivalenzklassen

$$\boxed{K_M^i \quad i = 1,\ldots n(V_M)}$$

Äquivalenzklassen von P bezüglich der Relation V_M

$$(p,q) \in V_M \Leftrightarrow \exists i : p \in K_M^i , q \in K_M^i$$

$$K_M^i \subseteq M \vee K_M^i \subseteq \overline{M}$$

$$\boxed{Z_M = \left\{K_M^1 ,\ldots, K_M^{n(V_M)}\right\}}$$ $\Rightarrow$ Zerlegung von P

Komponenten

Komponenten von *M*:

$$\boxed{K_M^i \subseteq M}$$

Komplementärkomponenten von *M*:

$$\boxed{K_M^i \subseteq \overline{M}}$$

Komponentenzerlegung von *P* bezüglich *M*:

$$\boxed{Z_M = \left\{K_M^1, \ldots, K_M^{n(V_M)}\right\}}$$

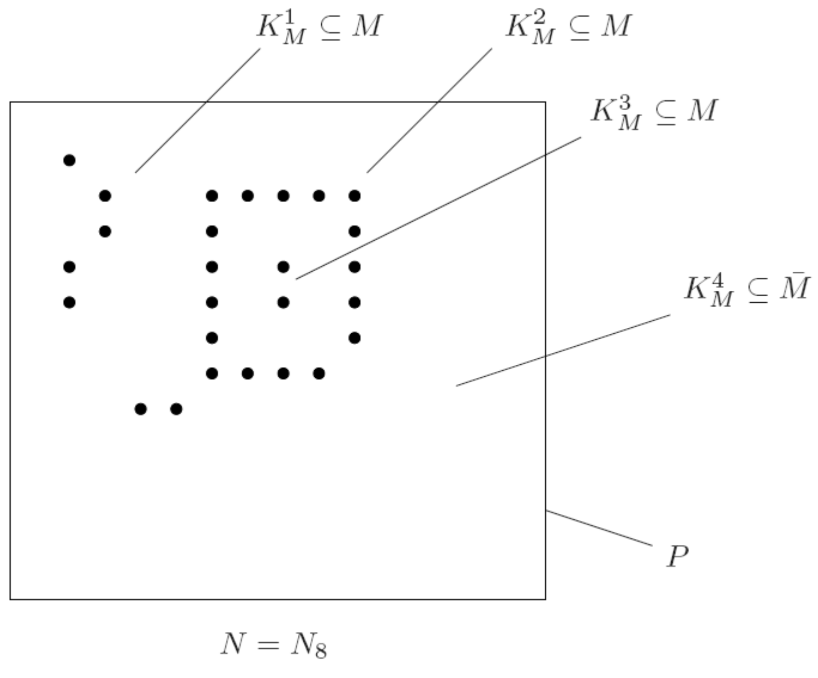
Komponenten

$$\forall p \in P \;\; \exists i : K_M^i = K_M(p) = \{q \in P : (p,q) \in V_M\}$$

maximal zusammenhängende Teilmengen

Komponenten – Beispiel

Die Komponentenzerlegung von P bezüglich der Menge M (schwarze Punkte).
Die ersten drei Komponenten sind Teilmengen von M
und die vierte Komponente ist gleich $P \backslash M$.



Bemerkung

- Die Komponentenzerlegung bildet eine einfache Möglichkeit zur Segmentierung eines Binärbildes.
- Wir benötigen aber noch Algorithmen zur Konstruktion der Komponenten (Segmente).

4.4 Bestimmung von Komponenten

2 Algorithmen

- Region Growing
  - ein allgemeiner Algorithmus
- Zeilenkoinzidenzverfahren
  - speziell für Bildraster

Region Growing

Nachbarschaftsstruktur: $[P, N]$

$$M \subseteq P$$

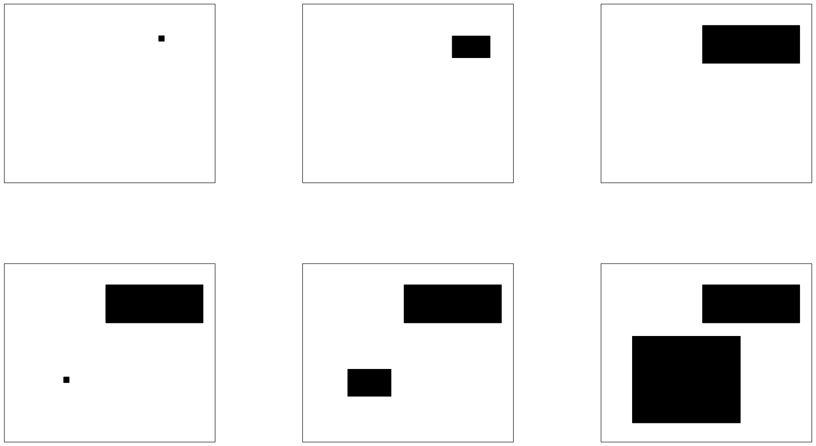
```
 Eingabe: P,N,M
while M <> ∅ do begin
        wähle p∈M;
        M := M \ {p};
        K := {p};
        L := {p};
        while L <> ∅ do begin
                wähle q ∈ L;
                L := L \ {q};
                bestimme N(q);
                N_M(q) := N(q) ∩M;
                K := K ∪ N_M(q);
                L := L ∪ N_M(q);
                M := M \ N_M(q);
        end;
        Ausgabe: K ist eine der Komponenten von M
end;
```

Der Algorithmus wählt einen beliebigen, noch nicht betrachteten Punkt aus M und fügt dann solange Nachbarn hinzu, bis dies nicht mehr möglich ist. Dann haben wir eine Komponente K gefunden.

Die Komponenten werden nacheinander gefunden.

Dieses Verfahren erinnert an die Tiefensuche.

Beispiel



Hier werden 2 Komponenten nacheinander gefunden

Zeilenkoinzidenzverfahren

Eingabebild (binär) segmentiertes Ausgabebild

$$G_E = (g_E(i,j)) \longrightarrow G_A = (g_A(i,j))$$

$$g_E(i,j) \in \{0,1\}$$

$$g_E(0,j) = g_E(i,0) = 0$$

$$g_A(i,j) = \begin{cases} 0 & \text{falls } (i,j) \in \text{Hintergrund} \\ k & \text{falls } (i,j) \in \text{k-te Komponente von M} \end{cases}$$

$$P = \{(i,j):\ i = 0,\ldots,I-1,\ \ j = 0,\ldots,J-1\}$$

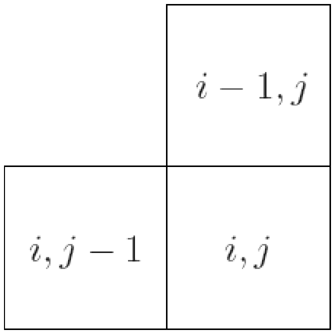
$$M \subseteq P, \quad M = \{(i,j) : g_E(i,j) = 1\}$$

$$N = N_4$$

Algorithmus

$g_A(i,j) = 0 \quad \forall i,j;$
$m := 0;$
for $i := 1$ to $I-1$ do
  for $j := 1$ to $J-1$ do
    if $g_E(i,j) = 1$ then begin
      if $(g_E(i-1,j) = 1)$ and $(g_E(i,j-1) = 0)$
          then $g_A(i,j) := g_A(i-1,j);$
      if $(g_E(i-1,j) = 0)$ and $(g_E(i,j-1) = 1)$
          then $g_A(i,j) := g_A(i,j-1);$
      if $(g_E(i-1,j) = 1)$ and $(g_E(i,j-1) = 1)$ then begin
        $g_A(i,j) := g_A(i,j-1);$ oder $g_A(i,j) := g_A(i-1,j);$
        if $g_A(i,j-1) <> g_A(i-1,j)$ then
          Identifizieren der Komponenten mit den Marken
          $g_A(i,j-1)$ und $g_A(i-1,j);$
      end;
      if $(g_E(i-1,j) = 0)$ and $(g_E(i,j-1) = 0)$ then begin
        $m := m + 1;$
        $g_A(i,j) := m;$
      end;
end;
Neunummerierung der Marken unter Berücksichtigung äquivalenter Marken.

Anschaulich verschiebt man ein Fenster über das Bild. Der Punkt *(i, j)* liegt auf einer 1 des binären Eingabebildes G_E.



Beispiel – Eingabebild

Die 0 – te Zeile ist hier oben

0	0	0	0	0	0	0	0
0	1	1	0	0	1	1	0
0	1	1	0	0	1	1	0
0	1	1	1	1	1	1	0
0	1	1	1	1	0	0	0
0	0	0	0	0	0	1	1
0	1	1	1	1	1	0	0

Beispiel – Zwischenergebnis

0	0	0	0	0	0	0	0
0	1	1	0	0	1	1	0
0	1	1	0	0	1	1	0
0	1	1	1	1	1	1	0
0	1	1	1	1	0	0	0
0	0	0	0	0	0	1	1
0	1	1	1	1	1	0	0

$\Rightarrow$

0	0	0	0	0	0	0	0
0	1	1	0	0	2	2	0
0	1	1	0	0	2	2	0
0	1	1	1	1	1	1	0
0	1	1	1	1	0	0	0
0	0	0	0	0	0	3	3
0	4	4	4	4	4	0	0

Hier könnte auch 2 stehen.
Deshalb sind die Marken 1 und 2 äquivalent.

Beispiel – segmentiertes Ausgabebild

0	0	0	0	0	0	0	0
0	1	1	0	0	1	1	0
0	1	1	0	0	1	1	0
0	1	1	1	1	1	1	0
0	1	1	1	1	0	0	0
0	0	0	0	0	0	2	2
0	3	3	3	3	3	0	0

Durch eine Neunummerierung erhalten wir 3 Komponenten
(1 bleibt, 2 wird 1, 3 wird 2, 4 wird 3).

Zeilenkoinzidenzverfahren

Bei der 8 – Nachbarschaft $N = N_8$ verwendet man ein Fenster der Form:

$i-1, j-1$	$i-1, j$	$i-1, j+1$
$i, j-1$	i, j	

Die Abarbeitung im Fenster geschieht in folgender Reihenfolge:

1. *i-1,j-1*
2. *i-1,j*
3. *i-1,j+1*
4. *i, j-1*

Beispiel

Eingabebild:

0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	0	0	1	1	0	0	0
0	0	1	1	1	1	1	0	0	1	0	0	1	0
0	0	0	0	1	0	1	0	0	0	0	0	1	0
0	1	1	1	1	1	1	1	1	1	1	1	1	0
0	0	0	1	1	1	1	1	1	1	1	1	1	0
0	1	0	0	0	0	1	0	1	0	0	1	1	0
0	0	0	1	1	0	0	0	0	0	0	0	0	0

Zwischenergebnis:

0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	0	0	2	2	0	0	0
0	0	3	3	1	1	1	0	0	2	0	0	4	0
0	0	0	0	3	0	1	0	0	0	0	0	4	0
0	5	5	3	3	3	1	1	1	1	1	4	4	0
0	0	0	5	3	3	3	1	1	1	1	1	4	0
0	6	0	0	0	0	3	0	1	0	0	1	1	0
0	0	0	7	7	0	0	0	0	0	0	0	0	0

Für $\boxed{x}$ könnte auch ein anderer Wert stehen.
Deshalb sind die Marken 1 und 3, 3 und 5 sowie 1 und 4 jeweils äquivalent.

Beispiel

Durch eine Neunummerierung erhalten wir 3 Komponenten
(1 bleibt, 2 bleibt, 3 wird 1, 4 wird 1, 5 wird 1, 6 und 7 bleibt).

0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	0	0	2	2	0	0	0
0	0	1	1	1	1	1	0	0	2	0	0	1	0
0	0	0	0	1	0	1	0	0	0	0	0	1	0
0	1	1	1	1	1	1	1	1	1	1	1	1	0
0	0	0	1	1	1	1	1	1	1	1	1	1	0
0	6	0	0	0	0	1	0	1	0	0	1	1	0
0	0	0	7	7	0	0	0	0	0	0	0	0	0

Bemerkung

- Das Zeilenkoizidenzverfahren liefert noch weitere Informationen, z.B.:
  - Lage der Komponenten im Bild
  - Flächeninhalt der Komponenten

Verallgemeinerung

$$G_E = (g_E(i,j)) \qquad g_E(i,j) \in \{0,1,\ldots,n\}$$

$$M_k = \left\{(i,j) \in P : g_E(i,j) = k\right\}$$

$$P = \bigcup_{k=0}^{n} M_k \qquad M_k \cap M_{k'} = \varnothing \quad k \neq k'$$

Bestimmung der Komponenten von M_k *(k=0,…,n)*

$$M_k = \bigcup_{q=1}^{m_k} K_{kq}$$

4.5 Regionenorientierte Segmentierung

4.5.1 Allgemeine Definition der Segmentierung

benachbarte Mengen

Nachbarschaftsstruktur: $[P, N]$

$$M_1 \subset P \qquad M_2 \subset P$$

benachbart:

$$\boxed{\begin{array}{c} M_1 \cap M_2 = \varnothing \\ \exists (p,q) \in N : p \in M_1 \text{ und } q \in M_2 \end{array}}$$

2 disjunkte Mengen heißen getrennt, wenn sie nicht benachbart sind.

Homogenitätsfunktion

Nachbarschaftsstruktur: $[P, N]$

$$\boxed{h : 2^P \rightarrow \{ true, false \}}$$

Teilmengen von *P*

Teilmenge ist homogen

Teilmenge ist nicht homogen

Segmentierung

Nachbarschaftsstruktur: $[P, N]$

Homogenitätsfunktion: $h: 2^P \to \{ true, false \}$

Unter einer Segmentierung einer Nachbarschaftsstruktur *[P,N]* bezüglich *h* verstehen wir eine Zerlegung von *P*:

$$\boxed{Z_S = \{X_1, \ldots, X_n\}}$$

$$\boxed{h(X_i) = true}$$

X_i, X_j - benachbart $(i \neq j)$ ⇒

$$\boxed{h(X_i \cup X_j) = false}$$

disjunkt, zusammenhängend

Zusammenhang zur Komponentenzerlegung

$$P = \{(i, j) : \; i = 0, \ldots, I-1, \;\; j = 0, \ldots, J-1\}$$

$N = N_4$ oder $N = N_8$ $\qquad M_1 \subseteq P$

$$\boxed{h(M) = \begin{cases} \mathit{true} & \text{falls} \quad M \subseteq M_1 \vee M \subseteq (P \setminus M_1) \\ \mathit{false} & \text{sonst} \end{cases}}$$

Die Komponenten von M_1 und $P \backslash M_1$ bilden eine Segmentierung bezüglich h.

Beispiel

Eingabebild:

1	1	1	2	2
1	2	2	2	3
3	2	2	3	1
3	3	2	2	2
1	3	1	1	2
1	1	1	1	2

$$G_E = (g_E(i,j)) \quad g_E(i,j) \in \{1,2,3\}$$

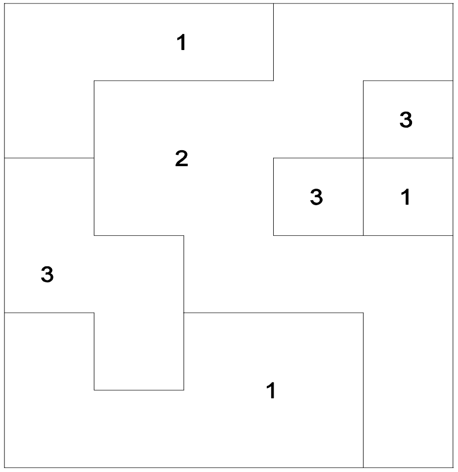
$$P = \{(i,j) : i = 0,\ldots,5 \quad j = 0,\ldots,4\}$$

$$N = N_4$$

$$M_k = \{(i,j) : g_E(i,j) = k\} \quad k = 1,2,3$$

$$h(M) = \begin{cases} true & \text{falls} \quad M \subseteq M_1 \vee M \subseteq M_2 \vee M \subseteq M_3 \\ false & \text{sonst} \end{cases}$$

Die Komponenten von M_1, M_2 und M_3 bilden eine Segmentierung bez. h.



Beispiel – andere Homogenitätsfunktion

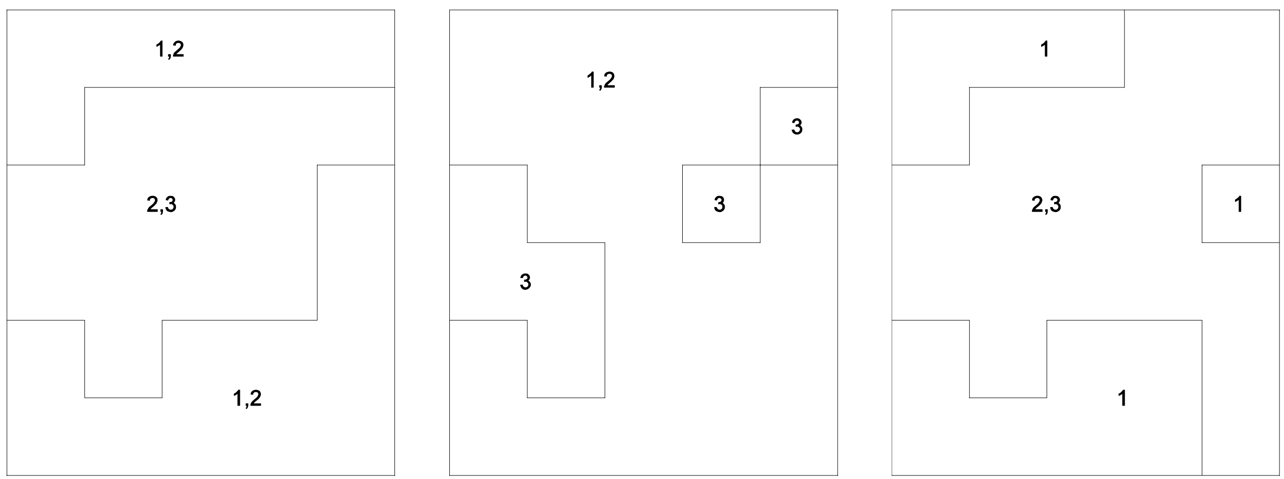
Eingabebild:

1	1	1	2	2
1	2	2	2	3
3	2	2	3	1
3	3	2	2	2
1	3	1	1	2
1	1	1	1	2

$$h(M) = \begin{cases} true & \text{falls} \quad (M \subseteq M_1 \cup M_2) \vee (M \subseteq M_2 \cup M_3) \\ false & \text{sonst} \end{cases}$$

Die Grauwerte einer homogenen Teilmenge unterscheiden sich höchstens um 1.

Segmentierung nicht eindeutig



4.5.2 Algorithmen zur Segmentierung

Algorithmen zur Segmentierung

2 mögliche Verfahren :

- **Region Growing**
  - arbeitet analog zur Komponentenbestimmung

- **Split – and – Merge – Verfahren**
  - lässt sich nur für Bildraster anwenden

Region Growing

Eingabe: P, N, h

while $P <> \emptyset$ do begin

    wähle $p \in P$;

    $P := P \setminus \{p\}$;

    $X := \{p\}; \quad T := \{p\}$;

    while $T <> \emptyset$ do begin

        wähle $q \in T$;

        $T := T \setminus \{q\}$;

        bestimme $N(q)$;

        wähle $S \subseteq N(q)$ mit $h(X \cup S) = true$;

        $X := X \cup S$;

        $T := T \cup S$;

        $P := P \setminus S$;

    end;

    Ausgabe: X als ein Objekt der Segmentierung

end;

Bemerkung

- Man kann auch mehrere Punkte p am Anfang betrachten und parallel arbeiten.
- Sobald sich dabei zwei Gebiete einander berühren, wird überprüft, ob das Homogenitätskriterium für die Vereinigung der beiden Gebiete erfüllt ist.
- Ist dies der Fall, so werden die beiden Gebiete zu einem einzigen verschmolzen.

Split – and – Merge

Prinzip:

- **Split – Ansatz:** Ausgehend vom vollständigen Bild wird das Bild rekursiv in Teilgebiete zerlegt (z.B. jedes Teilgebiet in 2 × 2 *Teile), bis jedes der Teilgebiete* das Homogenitätskriterium erfüllt.
- **Merge – Ansatz:** Ausgehend von einer feinen Unterteilung des Bildes in Teilgebiete (alle diese Gebiete erfüllen das Homogenitätskriterium) werden benachbarte Gebiete, deren Vereinigung das Homogenitätskriterium erfüllt, zusammengefasst.
- Kombinationen dieser beiden Ansätze führen zu **Split – and – Merge** – Algorithmen.

Algorithmusidee (1)

Eingabe:

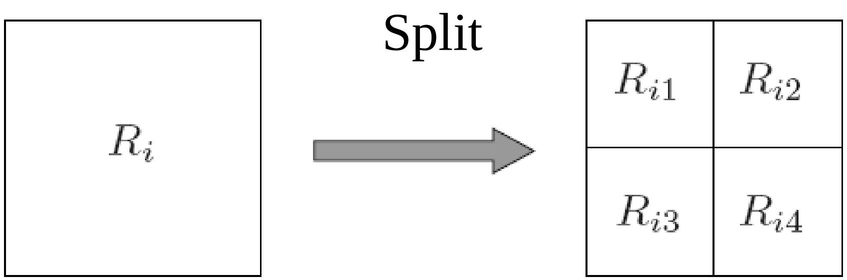
- Ausgangsunterteilung des Bildes in Gebiete:

$$P = \bigcup_{i=1}^{n} R_i$$

- Homogenitätskriterium *h*

Algorithmusidee (2)

$$P = \bigcup_{i=1}^{n} R_i \qquad \exists i : h(R_i) = \mathit{false}$$



$$R_i = \bigcup_{j=1}^{4} R_{ij}$$

Wenn die Vereinigung von benachbarten Gebieten aus der Menge:

$\{R_{i1}, R_{i2}, R_{i3}, R_{i4}\}$ für ein festes *i* das Homogenitätskriterium erfüllt, so fasse man diese Gebiete zusammen (Merge)

fortsetzen, bis kein Split oder Merge mehr möglich ist $\Rightarrow$ $P = \bigcup_{i=1}^{m} R_i'$

Algorithmusidee (3)

$$P = \bigcup_{i=1}^{m} R_i'$$

wenn zwei benachbarte Gebiete R'_i, R'_j
das Homogenitätskriterium h erfüllen,
so fasse man diese zusammen (Merge)

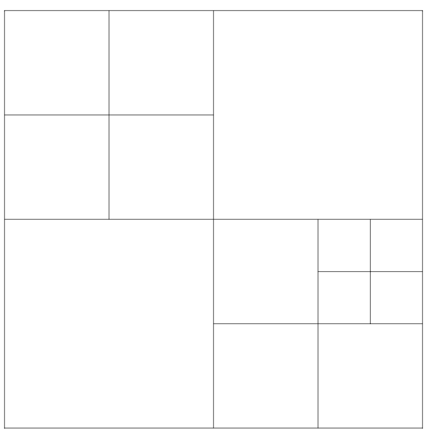
R'_i, R'_j entstanden nicht durch den gleichen Split

Bemerkung

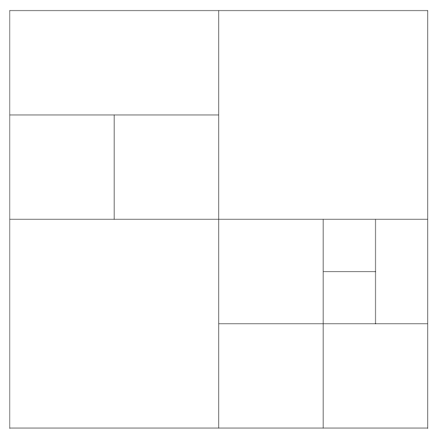
- Die rekursive Unterteilung eines Bildes in 4 Quadrate nennt man auch Quadtree.
- Sie ist besonders günstig, wenn $I = J = 2^n$.

Beispiel

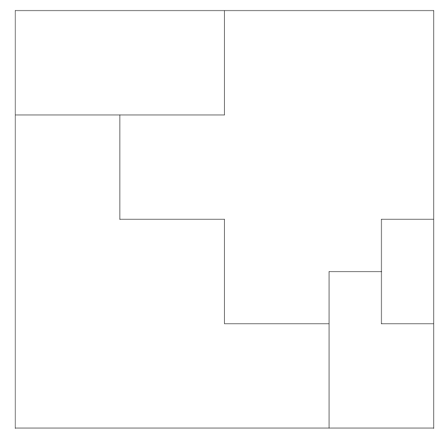
1	1	1	1	2	2	2	2
1	1	1	1	2	2	2	2
3	3	2	2	2	2	2	2
3	3	2	2	2	2	2	2
3	3	3	3	2	2	2	7
3	3	3	3	2	2	8	7
3	3	3	3	3	3	8	8
3	3	3	3	3	3	8	8



Split



Merge



Merge