

Grundlagen der Künstlichen Intelligenz

20 Segmentierung und Objekterkennung

Region Merging, Edge Linking,
Deformable Matching, Alignment

Volker Steinhage

- Segmentierung
 - nach Homogenitätskriterien
 - Region Merging
 - nach Diskontinuitätskriterien
 - Edge Linking
- Objekterkennung
 - Deformable Matching
- Objektlokalisierung
 - Alignment

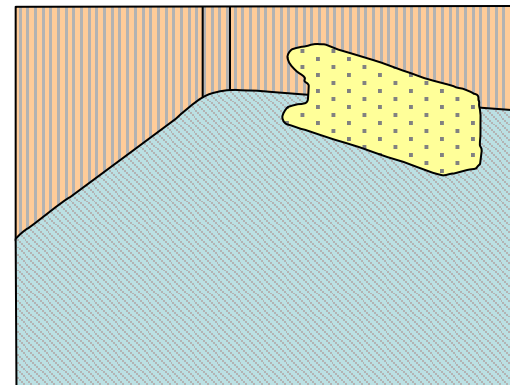
Segmentierung (1)

- Menschen ordnen wahrgenommene visuelle Perzepte an:
 - Statt einer **Sammlung von einzelnen Helligkeitswerten** nehmen wir **visuelle Gruppen** wahr.
 - Diese visuellen Gruppen können wir meist Objekten oder Teilen von Objekten zuordnen.



320×256
Grauwerte

→
Visuelle
Gruppierung



Fünf visuelle
Gruppen

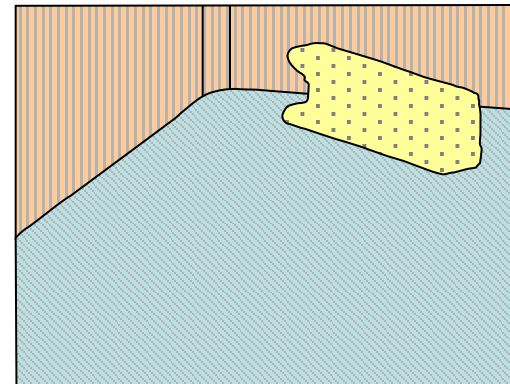
Segmentierung (2)

- Im Computersehen wird der Prozess der Zerlegung eines Bildes in Pixelgruppen basierend auf Pixelähnlichkeiten als **Segmentierung** bezeichnet.
 - Grundlegende Idee: innerhalb einer visuellen Gruppe sind deren Pixel ähnlich, während Pixel verschiedener visueller Gruppen stärker variieren.
 - Die Messung von Pixelähnlichkeit erfolgt über Eigenschaften wie Intensität, Farbe oder Textur.



320×256
Grauwerte

→
Segmentierung



Fünf visuelle
Gruppen

Segmentierung (3)

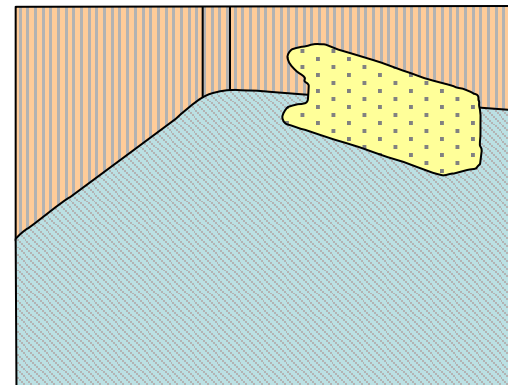
- Im dreistufigen Informationsfluss der visuellen Verarbeitung setzt die **Segmentierung** somit die 2. Stufe der Mid-Level-Vision um:

die Transformation von pixelbasierten Bildrepräsentationen in segmentbasierte Bildrepräsentationen.



320×256
Grauwerte

→
Segmentierung



Fünf visuelle
Gruppen

Segmentierung nach Homogenitätskriterien

- Eine Klasse von Segmentierungsverfahren basiert auf allgemeinen Homogenitätskriterien.
- Ein **Homogenitätskriterium** definiert die Zusammengehörigkeit von Segmenten mit Bezug zu den Pixeln der Segmente.
- Beispiele für Homogenitätskriterien:
 - dieselben oder ähnliche Intensitäts- oder Farbwerte,
 - begrenzte Intensitätsvarianz innerhalb eines Segments,
 - maximaler Abstand zwischen maximalem und minimalem Intensitätswert eines Segments.

Region Merging (1)

Region Merging stellt die einfachste Methode einer Segmentierung nach Homogenitätskriterium dar:

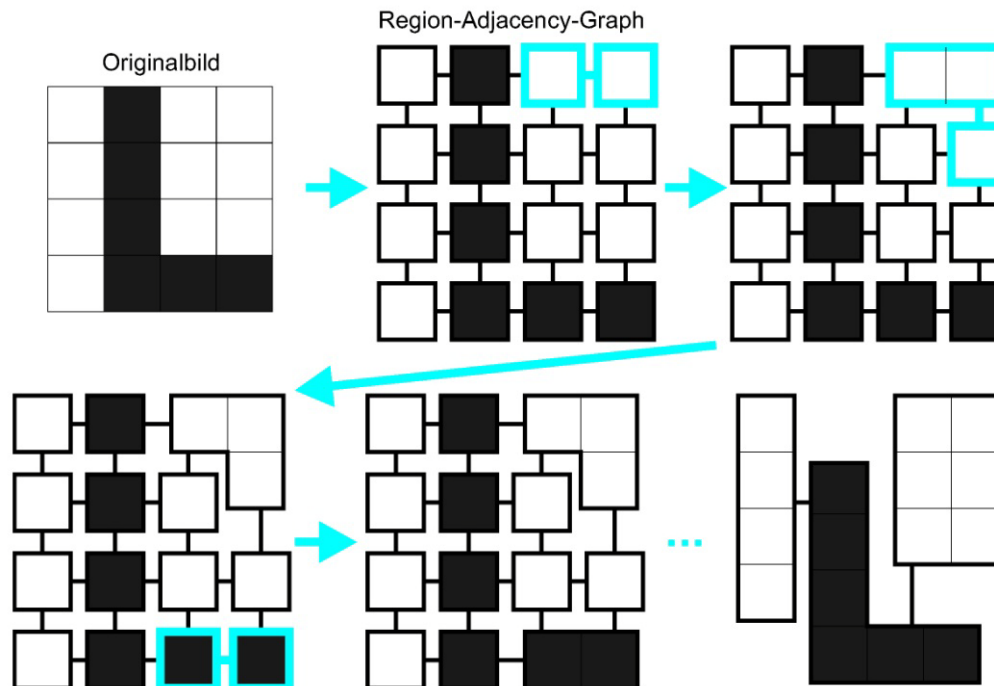
- 1) Start: jedes Pixel bildet ein eigenes Segment
- 2) Merging: zwei benachbarte Regionen werden zusammengefasst, wenn sie auch gemeinsam das Homogenitätskriterium erfüllen.
Zur Reihenfolge: i.A. werden immer die benachbarten Regionen zusammengefasst, die das Homogenitätskriterium am stärksten erfüllen.
- 3) Terminierung: die Segmentierung terminiert, wenn keine Regionen mehr zusammengefasst werden können.

Da Region Merging mit Pixeln als Segmenten startet, muss das Homogenitätskriterium sowohl für einzelne Pixel als auch für Regionen berechenbar sein.

Region Merging (2)

Einfaches Beispiel: Anwendung von Region Merging auf ein Binärbild mit gleichem Intensitätswert als Homogenitätskriterium für alle Segmente.

Der Prozess wird durch einen sog. Regionenadjazenzgraphen (RAG) visualisiert. In einem RAG sind alle Segmente (bzw. Regionen) als Knoten und alle direkten Nachbarschaften als Kanten dargestellt.



Segmentierung nach Diskontinuitätskriterien (1)

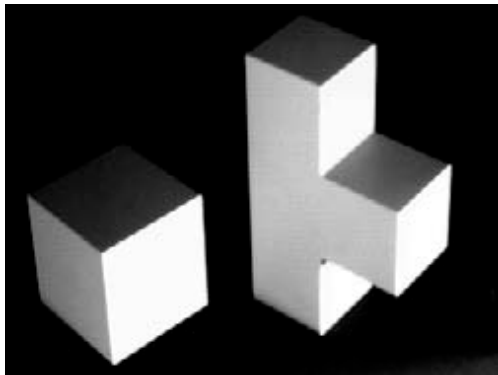
- Bei der Segmentierung nach Homogenitätskriterien muss das Homogenitätskriterium für *alle* Pixel des gesamten Segments gelten! Bei stärkeren Variationen der Segmentcharakteristik können Segmente dann jedoch fälschlicherweise in mehrere Segmente zerlegt werden. Es kommt zu einer **Übersegmentierung**.
- Daher bietet sich ein komplementärer Ansatz der Segmentierung an, nämlich die **Segmentierung nach Diskontinuitätskriterien**.

Solche Ansätze zerlegen ein Bild nach Diskontinuitätskriterien an den Segmenträndern und sind weniger anfällig bzgl. Variationen der Segmentcharakteristik.

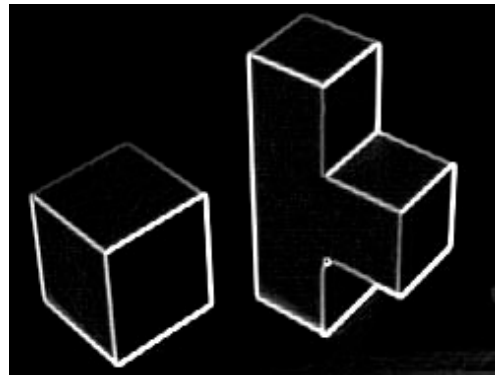
Typische Diskontinuitätskriterien beziehen sich auf Intensitätsgradienten, die **Konturkanten** wiedergeben.

Eigenschaften von Kantenzügen → **Kantenmodell**

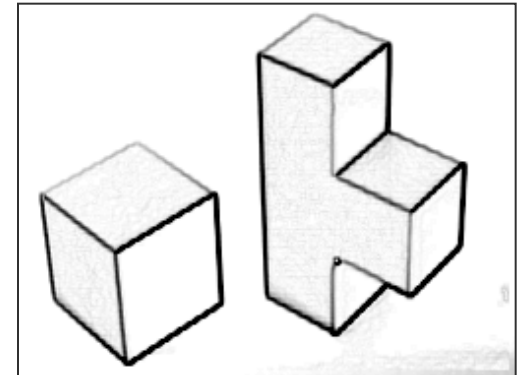
- (1) Der **Betrag des Gradienten** gibt die Stärke der Kanteneigenschaft an. Ist der Gradientenbetrag gering, so kann die zugrunde liegende Intensitätsänderung durch Rauschen bedingt sein.
- (2) Aus **der Richtung des Gradienten** kann auf die (lokale) Richtung des Kantenzuges geschlossen werden. Diese sollte orthogonal zur Gradientenrichtung des betrachteten Kantenpixels sein.
- (3) Kantenzüge sind i.A. **kontinuierlich**: in der lokalen Umgebung eines Kantenpixels sollten weitere Kantenpixel mit ähnlichen Eigenschaften (Betrag und Richtung des Gradienten) zu finden sein.



Grauwertbild



Antworten des
Sobel-Operators



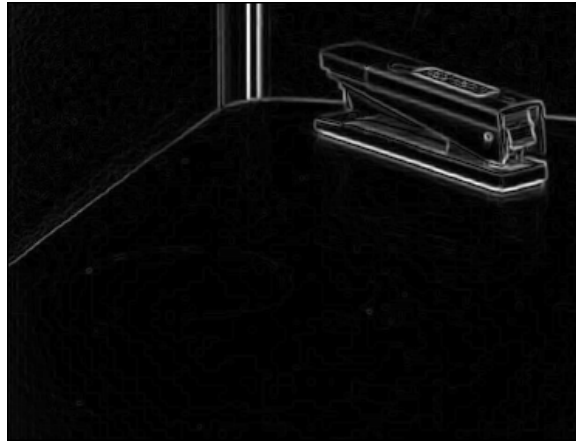
Invertierte Antworten des
Sobel-Operators

Binarisierung zur Selektion von Kantenpixeln (3)

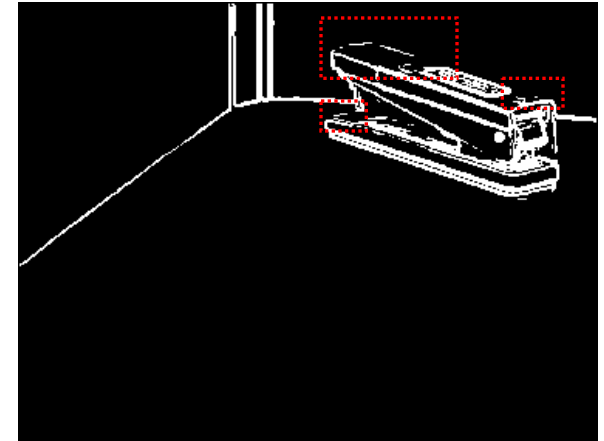
I. A. wird die schwellwertbasierte Binarisierung jedoch dazu führen, dass nicht mehr alle Kantenpixel über ihre Nachbarschaft miteinander verbunden sind, weil es häufig Bildbereiche gibt, in denen der Kontrast zwischen benachbarten Regionen, die aber verschiedene Szenenflächen abbilden, zu gering ist (s. Bspl.).



Tacker-Grauwertbild*



Antworten Sobel-Operator



Binarisierungsergebnis für $t_B=40$

Edge Linking ist ein Ansatz, der Kantenpixel auch über **Lücken** in den Pixelnachbarschaften miteinander zu Kantenzügen verbinden kann.

Bemerkung: im Bspl. hätte ein niedrigerer Schwellwert zwar zu weniger Lücken geführt, aber auch zu eher flächenhaften statt linienhaften Anordnungen von Kantenpixeln, was wiederum zu falschen Linienhypothesen führen kann. Prinzipiell kann aber der Kontrast zwischen zwei Regionen beliebig gering sein. Das Problem besteht also generell.

* Bildquelle: S. Russel, P. Norvig: Künstliche Intelligenz (2. Aufl.), Pearson Studium, 2004.

Algorithmus *Edge Linking*

(1) Markiere alle Pixel mit **hinreichend großem Gradientenbetrag** (Schwellwert) als *Kantenpixel*.

1. Eigenschaft von Kantenpixeln

(2) Markiere alle *Kantenpixel* als *unbearbeitet*.

(3) Wähle nächstes *Kantenpixel*, das noch *unbearbeitet* ist und erkläre es als *aktives Pixel* p_a eines Kantenzuges k .

2. Eigenschaft von Kantenpixeln

(4) Wenn in der Umgebung $U(p_a)$ des *aktiven Pixels* p_a **in Kantenrichtung** (orthogonal zur Gradientenrichtung) *unbearbeitete* Kantenpixel p_i gefunden werden, die **ähnliche Gradientenrichtungen und -beträge** aufweisen, dann

a) markiere die Kantenpixel p_i als zum **Kantenzug** k gehörend,

3. Eigenschaft von Kantenpixeln

b) erkläre die Kantenpixel p_i zu neuen *aktiven Pixeln*,

c) markiere das aktive Kantenpixel p_a als *bearbeitet*.

d) weiter mit Schritt (3), bis alle Kantenpixel als *bearbeitet* markiert sind.

(5) Wenn in $U(p_a)$ Kantenpixel gefunden wurden, die bereits einem Kantenzug zugeordnet sind, dann wurde eine **Verzweigung** von Kanten gefunden.

Zum Algorithmus *Edge Linking* (1)

- Edge Linking terminiert, wenn alle Kantenpixel als bearbeitet markiert sind. Kantenpixel, die als einzige Pixel einem Kantenzug zugeordnet wurden (s. Schritt (3), werden als *Störungen* eingeordnet und entfernt.
- Edge Linking zeigt drei heuristische Parameter in Schritt 4:
 - die *Größe* der Umgebung $U(p_a)$ des *aktiven Pixels* p_a , in der Kantenpixel mit *ähnlichen* Gradientenrichtungen und –beträgen gesucht werden, sowie
 - die *Ähnlichkeit von Gradientenrichtungen*,
 - die *Ähnlichkeit von Gradientenbeträgen*.

Diese drei Größen müssen der Qualität der zu untersuchenden Bilder angepasst werden.

Zum Algorithmus *Edge Linking* (2)

- Durch die *Größe* der Umgebung $U(p_a)$ können **Lücken** in Kantenzügen überbrückt werden.
- Beispiel:

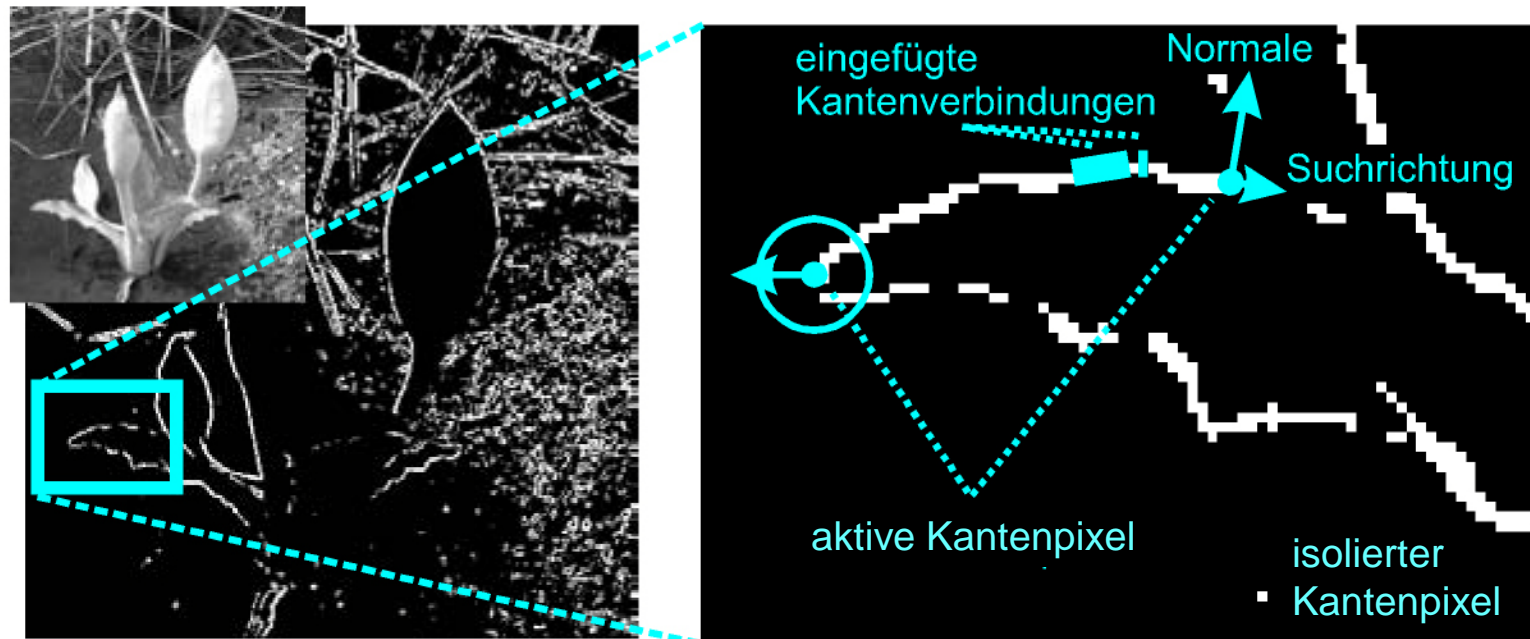
wenn die Umgebung $U(p_a)$ als ein in p_a zentriertes 5×5 -Pixelfeld definiert ist, dann lässt dies Lücken in den Kantenzügen von einem Pixel zu.

Diese Lücken werden überbrückt – im einfachsten Fall durch lineare Interpolation oder aber auch durch Splines o.ä.

Beispiel zu *Edge Linking*

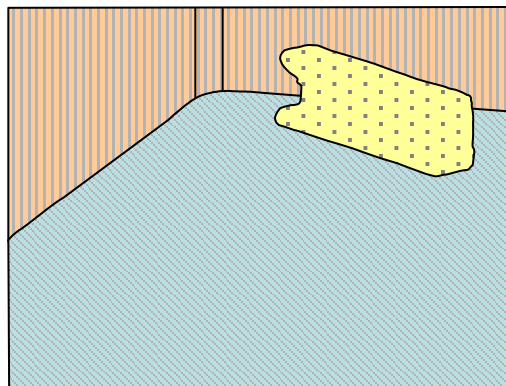
Der ausgewählte Bildausschnitt zeigt u.a.

- zwei aktive Kantenpixel mit aktuellen Suchrichtungen orthog. zur Gradientenrichtung,
- zwei bereits überbrückte Lücken,
- einen Kantenpixel, der nach Terminierung als isolierter Kantenpixel entfernt wird.

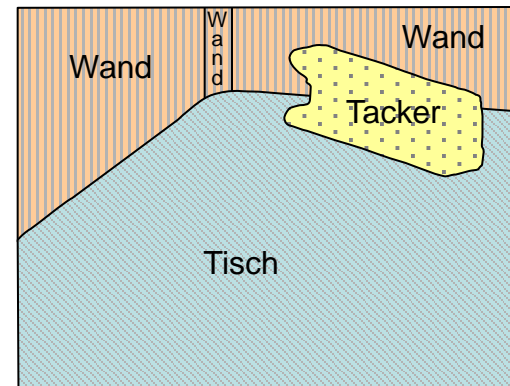
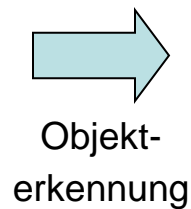


Objekterkennung (1)

- Menschen können zuverlässig Menschen, Tiere u. a. Objekte erkennen.
- In der KI und im Computersehen wird hierfür der Begriff der **Objekterkennung** verwendet.
- Der Begriff Objekterkennung umfasst dabei
 - die Erkennung von Objektklassen (also die Klassifikation)
 - die Erkennung von bestimmten Objekten (also die Wiedererkennung).



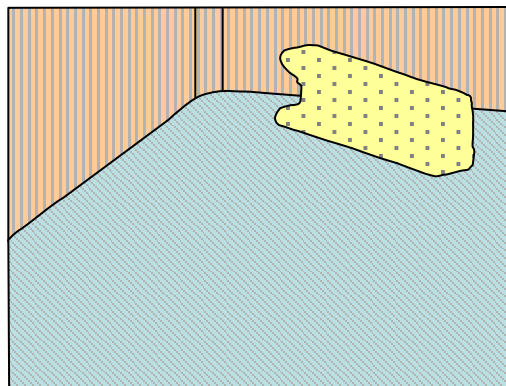
Fünf visuelle
Gruppen



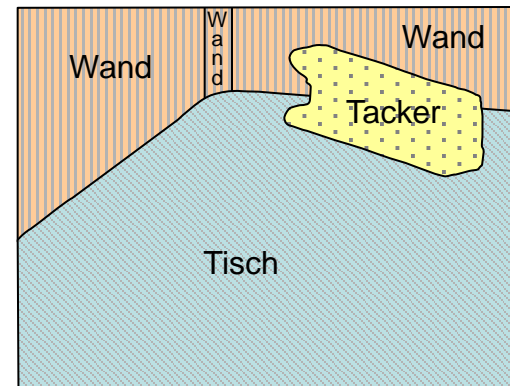
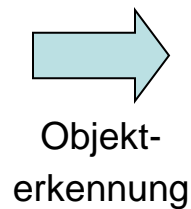
Fünf Objekt-
klassifikationen

Objekterkennung (3)

- Bei der **Segmentierung** wurden lediglich **allgemeine Annahmen** über die Ähnlichkeiten von Pixeln oder die Verläufe von Konturen eingesetzt.
- Bei der **Objekterkennung** werden **domänenspezifische Modelle** der zu erkennenden Objekte oder Objektklassen verwendet.
- Damit gehört die Objekterkennung zur 3. Phase der visuellen Verarbeitung, der High-Level-Vision.



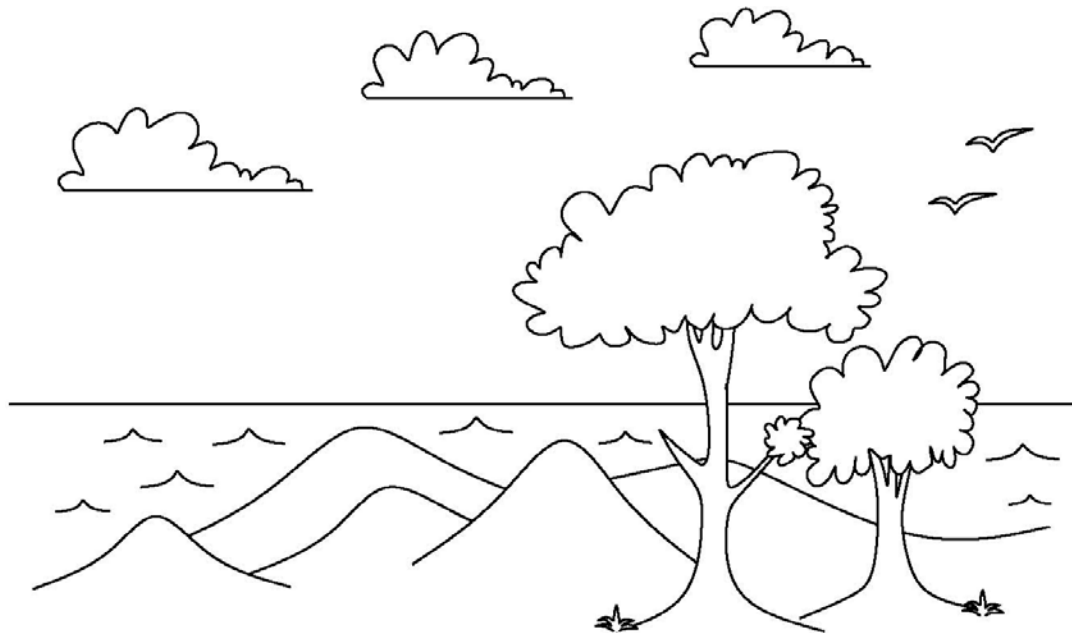
Fünf visuelle
Gruppen



Fünf Objekt-
klassifikationen

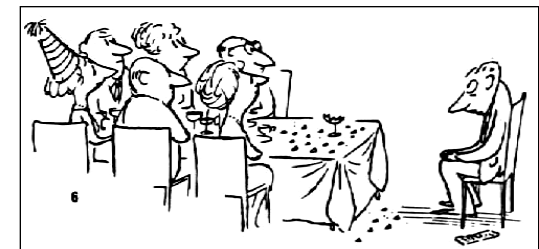
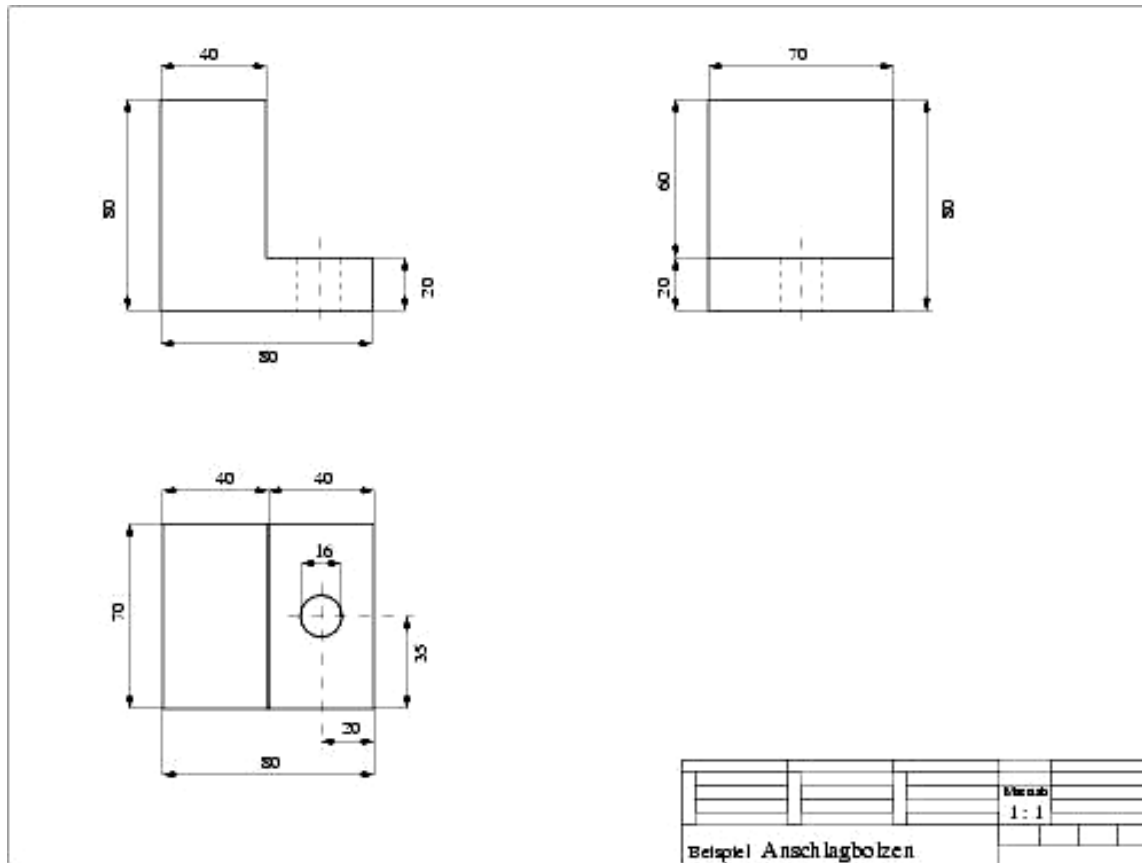
Konturbasierte Objekterkennung (1)

- Die Form der Kontur ist für viele Objekte charakteristisch.
 - So zeigen Strichzeichnungen eine starke Reduktion bzgl. der Bilddarstellung: es werden lediglich die Konturen der abgebildeten Objekte dargestellt – es wird abstrahiert von Farben, Texturen sowie Licht- und Schatteneffekten.
 - Dennoch sind so komplexe Szenarien darstellbar:



Konturbasierte Objekterkennung (2)

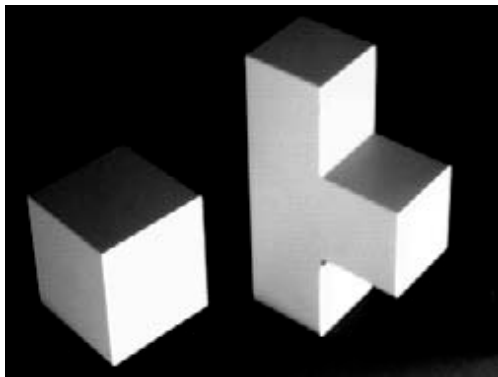
- Auch **technische Zeichnungen** sowie **Cartoons** und **Comics** zeigen eine konturbasierte Repräsentation von Objekten und Handlungsabläufen:



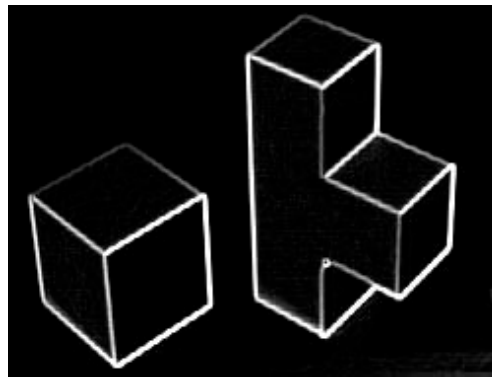
Konturbasierte Objekterkennung (3)

... und konturbasierte Objekterkennungsverfahren passen gut zu den bisher vorgestellten Methoden der Low- und Mid-Level-Vision,

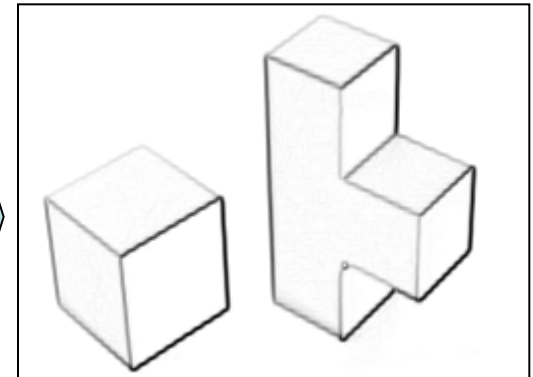
- also Glättungsfiltern und Kantenhervorhebung sowie konturbasierter Segmentierung.
- So sieht das unten abgebildete invertierte Ergebnis des Sobel-Operators eines einfachen Grauwertbildes doch schon wie eine Strichzeichnung aus!



Grauwertbild



„Sobel-Bild“



Invertiertes „Sobel-Bild“

Deformierbare Übereinstimmung

- Für einen konturbasierten Ansatz der Objekterkennung bietet die Idee der deformierbaren Übereinstimmung (Deformable Matching) einen Lösungsweg.
 - Bereits 1917 hat der brit. Mathematiker und Biologe D'Arcy Thompson beobachtet, dass ähnliche, aber nicht identische Umrisse häufig unter Verwendung einfacher Koordinatentransformationen in die gleiche Ausrichtung deformiert werden können.
 - Dieses Konzept wird in der Computergrafik als *Morphing* bezeichnet.
- Die Idee der deformierbaren Übereinstimmung wird im Rahmen eines konturbasierten Ansatzes der Objekterkennung für die Bestimmung von Umrissähnlichkeiten umgesetzt.

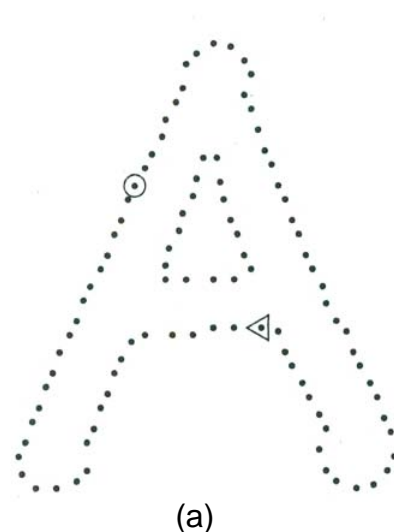
Umrissähnlichkeit

- Die Ermittlung einer Umrissähnlichkeit zwischen zwei Umrissen wird in drei Stufen umgesetzt:
 - 1) Lösung des **Zuordnungsproblems** zwischen zwei Umrissen
 - i. A. zwischen einem Segmentumriss und einem Modellumriss
 - 2) Schätzung der **Ausrichtungstransformation** unter Verwendung der ermittelten Zuordnungen.
 - 3) Berechnung der **Distanz** zwischen beiden Umrissen
 - als Summe der Vergleichsfehler zwischen zugeordneten Punkten
 - zusammen mit einem Term, der die Größe der Ausrichtungstransformation misst.

Repräsentation von Umrissen (1)

- Der Umriss eines Objektes wird durch eine diskrete Menge von Umrisspunkten repräsentiert:
 - Die **Umrisspunkte** werden als Stichproben aus den internen oder externen Konturen auf dem Umriss ermittelt.
 - Die Ermittlung dieser Umrisspunkte kann durch einen Kantenoperator (z.B. den Sobel-Operator) umgesetzt werden.
 - Ergebnis ist eine Menge $\{p_1, \dots, p_N\}$ von N Kantenpixeln bzw. Umrisspunkten.

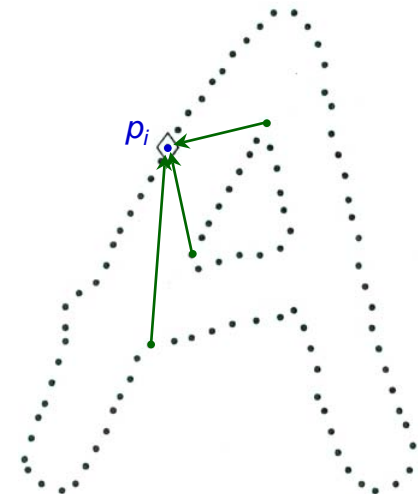
Zwei stichprobenartig
ermittelte Konturen:



Repräsentation von Umrissen (2)

Jedem **Umrisspunkt** p_i wird ein **Umrisskontext** (*shape context*) zugeordnet:

- Zunächst werden die **relativen Koordinaten** $(p_k - p_i)$ *aller* restlichen $N-1$ Umrisspunkte p_k **relativ zu Umrisspunkt** p_i ermittelt.
 - Diese Vektoren repräsentieren den gesamten Umriss **relativ zum Umrisspunkt** p_i .
- Im Bspl. sind die relativen Koordinaten $(p_k - p_i)$ für drei Punkte p_k angedeutet.



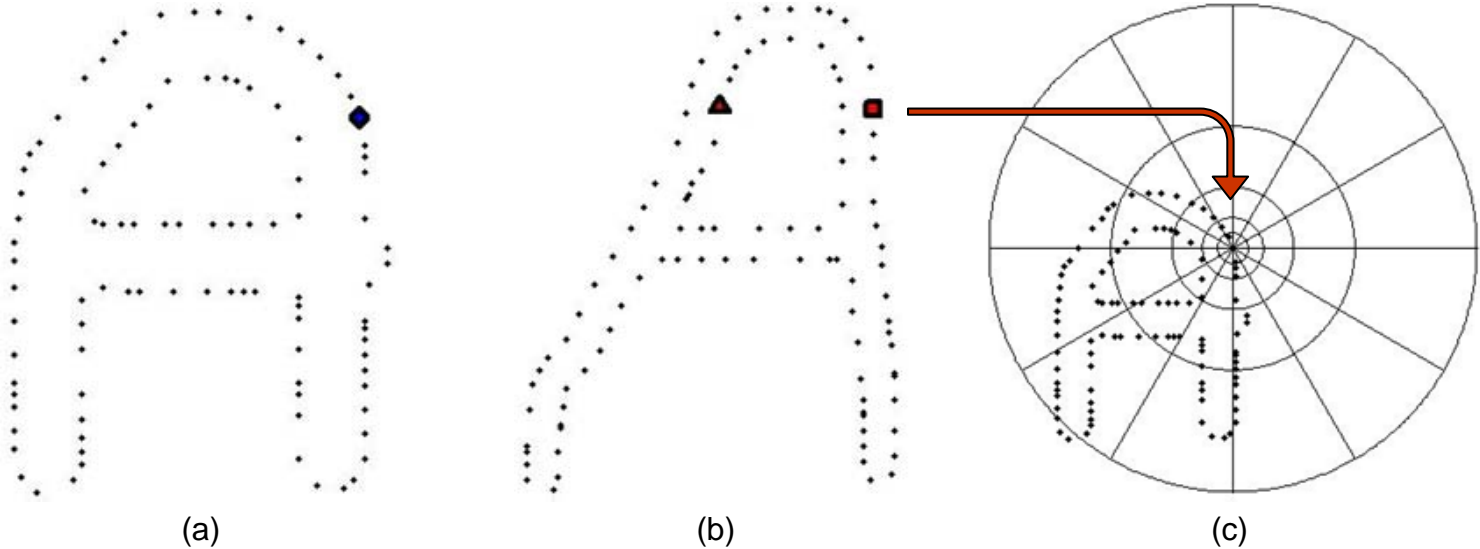
Repräsentation von Umrissen (3)

Jedem Umrisspunkt p_i wird ein Umrisskontext zugeordnet:

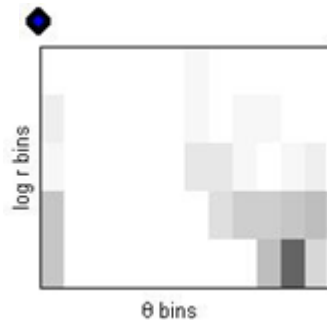
- Der Umrisskontext von Umrisspunkt p_i ist ein räumliches Histogramm h_i der relativen Koordinaten $(p_k - p_i)$, das wie folgt ermittelt wird.
- Das Histogramm basiert auf einem logarithmischen Polarkoordinatensystem, um sensibler bzgl. Unterschieden bei naheliegenden Punkten zu sein .
- Der Umrisskontext ist translationsinvariant, da alle Angaben relative Maße zu den Objekturnrisspunkten sind.
- Skalierungsinvarianz wird erzielt, indem alle radialen Distanzen durch die mittlere Distanz zwischen den Punktepaaaren normiert werden.

Repräsentation von Umrissen (4)

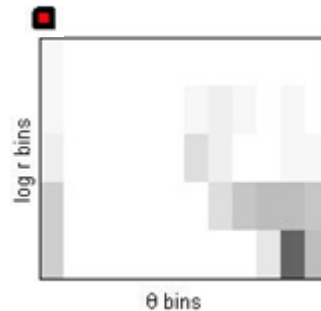
- Beispiel:



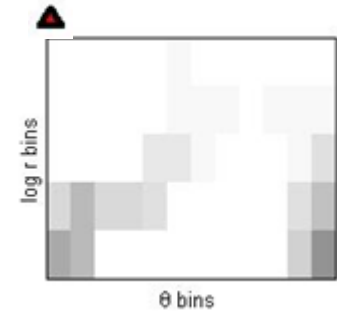
Zwei Konturen von zwei Objekten in (a) und (b). Das logarithm. Polarhistogramm mit 5 Werten für $\log r$ und 12 Werten für θ in (c). In (d) - (f) die Umrisskontexte der drei markierten Umrisspunkte in (a) und (b): dunkle Felder stellen mehr Punkte im Wert dar.



(d)



(e)



(f)

Zuordnung zwischen Umrissen (1)

Umrisskontexte erlauben die Bewertung der Zuordnung von zwei ähnlichen, aber nicht identischen Objekturnissen P und Q .

- Für jede Zuordnungshypothese zwischen einem Umrisspunkt p_i des einen Umrisses P und einem Umrisspunkt q_j des anderen Umrisses Q wird die Distanz $c_{ij} = c(p_i, q_j)$ über eine χ^2 -Distanz ermittelt:

$$c_{ij} = \frac{1}{2} \sum_{k=1}^K \frac{[h_i(k) - h_j(k)]^2}{h_i(k) + h_j(k)}.$$

In unserem Bspl.:
 $K = 5 \cdot 12 = 60$.

Für $h_i(k) + h_j(k) = 0$
ist der Summand
gleich Null.

Dabei bezeichnen $h_i(k)$ und $h_j(k)$ den k -ten Wert der normalisierten 60-elementigen Polarhistogramme der Punkte p_i und q_j .

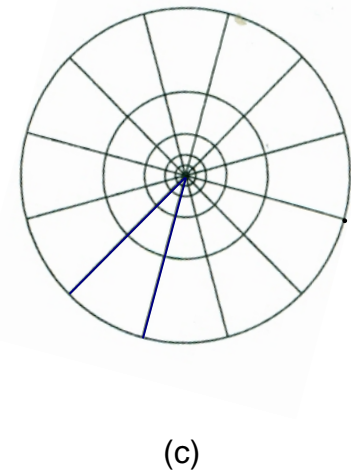
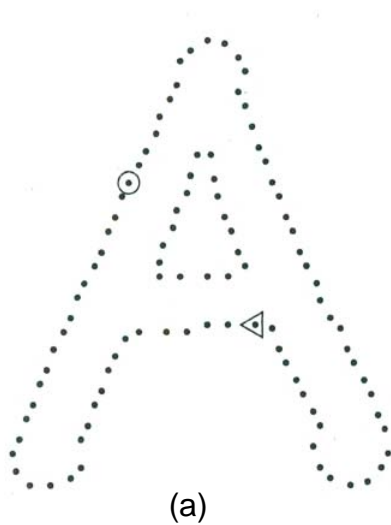
Zuordnung zwischen Umrissen (2)

- Gesucht ist nun die eindeutige Zuordnung zwischen den Punkten $\{p_1, \dots, p_N\}$ des einen Umrisses und den Punkten $\{q_1, \dots, q_N\}$ des anderen Umrisses.
 - Die zu wählende Zuordnung folgt aus der Minimierung der Gesamtdistanz über alle $c_{ij} = c(p_i, q_j)$.
 - Dies ist ein **gewichtetes Zuordnungsproblem auf bipartiten Graphen** und mit der **Ungarischen Methode** bzw. dem **Kuhn-Munkres-Algm.** in $O(n^3)$ zu lösen*.
 - Aus der eindeutigen Zuordnung aller Umrisspunkte ist eine Ausrichtungstransformation schätzbar, die einen Umriss in den anderen abbildet.

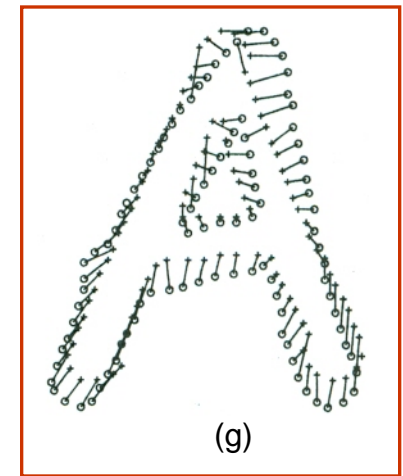
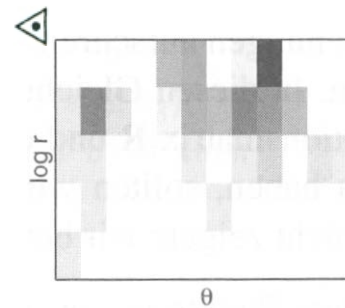
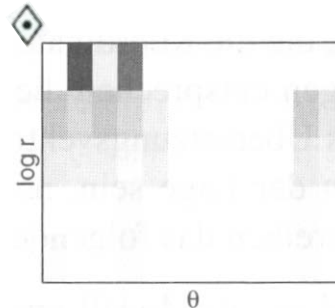
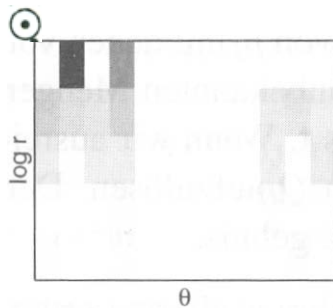
* Die Ungarische Methode wurde 1955 von Harold W. Kuhn auf der Basis von Ideen der ungarischen Mathematiker Dénes König und Jenő Egerváry entwickelt und von James Munkres 1957 optimiert.

Zuordnung zwischen Umrissen (3)

- Beispiel:



Zwei stichprobenartig ermittelte Konturen von zwei Objekten in (a) und (b). Das logarithm. Polarhistogramm mit 5 Werten für $\log r$ und 12 Werten für θ in (c). In (d)-(f) die Umrisskontexte der drei markierten Umrisspunkte in (a) und (b). Die eindeutige Zuordnung zw. den Umrisspunkten (a) und (b) über die bipartite Zuordnung über die χ^2 -Differenzen zwischen den Polarhistogrammen in (g).



Ungarische Methode (1)

- Die ungarische Methode basiert auf der Repräsentation der eindeutigen Zuordnungsaufgabe durch eine

quadratische $N \times N$ -Koeffizientenmatrix $\mathbf{C} = (c_{ij})$, $i, j \in \{1, 2, \dots, N\}$.

- \mathbf{C} entspricht dabei der mit den Zuordnungskosten gewichteten Adjazenzmatrix des bipartiten Graphen.
- Die Matrix wird schrittweise durch elementare Umformungen so modifiziert, dass sich die optimale Zuordnung nicht ändert, aber ableitbar wird.

Ungarische Methode (2)

- Der Trick der ungarischen Methode besteht genauer darin, die Koeffizientenmatrix durch elementare Umformungen so zu modifizieren bzw. reduzieren, dass sich zwar die Koeffizientenwerte der Matrix – und damit die absoluten Distanzkosten – ändern, nicht aber die optimale Zuordnung.*
- Konkret werden in den Umformungen die Distanzminima von Zeilen und Spalten subtrahiert oder addiert. Die Matrix wird so auf möglichst viele Nulleinträge reduziert, die die Erkennung von optimalen Zuordnungen ermöglicht.
- Als Beispiel wird die Methode für die eindeutige Zuordnung zwischen vier Umrisspunkten $\{p_1, \dots, p_4\}$ des einen Umrisses P und vier Umrisspunkten $\{q_1, \dots, q_4\}$ des anderen Umrisses Q dargestellt.

* Die Vorgehensweise ähnelt dem Lösen von homogenen linearen Gleichungssystemen durch elementare Zeilenumformungen auf deren Koeffizientenmatrizen.

Ungarische Methode (3)

- Zunächst der einfache Fall für die eindeutige Zuordnung zwischen vier Umrisspunkten $\{p_1, \dots, p_4\}$ des einen Umrisses und vier Umrisspunkten $\{q_1, \dots, q_4\}$ des anderen Umrisses.
- Umformung in eine quadratische 4×4 -Matrix:
 - jeder Zeile entspricht dabei ein p_i , jeder Spalte ein q_j ($i, j = 1, \dots, 4$),
 - jede Matrixkomponente enthält die Bewertung c_{ij} der Zuordnung:

Ausgangsmatrix:

	q_1	q_2	q_3	q_4
p_1	2	1	3	4
p_2	3	3	2	3
p_3	3	1	2	3
p_4	4	2	2	1

Ungarische Methode (4)

- Schritt 1: Zeilenumformung

- für jede Zeile z_i wird der minimale Eintrag \min_i gesucht,
- der minimale Eintrag \min_i wird von allen Komponenten der Zeile z_i subtrahiert,
- die reduzierte Matrix hat in jeder Zeile mindestens einen Null-Eintrag.

	q_1	q_2	q_3	q_4
p_1	2	1	3	4
p_2	3	3	2	3
p_3	3	1	2	3
p_4	4	2	2	1

Ausgangsmatrix M^0



	q_1	q_2	q_3	q_4	\min_i
p_1	2	1	3	4	1
p_2	3	3	2	3	2
p_3	3	1	2	3	1
p_4	4	2	2	1	1

Min. Zeileneinträge von M^0



	q_1	q_2	q_3	q_4
p_1	1	0	2	3
p_2	1	1	0	1
p_3	2	0	1	2
p_4	3	1	1	0

Reduzierte Matrix M^1

Ungarische Methode (5)

- Schritt 2: Spaltenumformung

- für jede Spalte s_j wird der minimale Eintrag \min_j gesucht,
- der minimale Eintrag \min_j wird von allen Komponenten der Spalte s_j subtrahiert,
- die reduzierte Matrix hat auch in jeder Spalte mindestens einen Null-Eintrag.

	q_1	q_2	q_3	q_4
p_1	1	0	2	3
p_2	1	1	0	1
p_3	2	0	1	2
p_4	3	1	1	0

Matrix M^1



	q_1	q_2	q_3	q_4
p_1	1	0	2	3
p_2	1	1	0	1
p_3	2	0	1	2
p_4	3	1	1	0
\min_j	1	0	0	0

Min. Spalteneinträge von M^1



	q_1	q_2	q_3	q_4
p_1	0	0	2	3
p_2	0	1	0	1
p_3	1	0	1	2
p_4	2	1	1	0

Reduzierte Matrix M^2

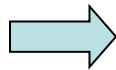
Ungarische Methode (6)

Schritt 3: Test auf optimale Zuordnungen

- Die minimale Zahl n von horizontalen *und* vertikalen Linien wird so in \mathbf{C} eingezeichnet, dass *alle Nullen* überdeckt sind.
- Ist $n = N = \dim(\text{Matrix})$, so ist die optimale Zuordnung durch Markierung von Nullen mit einem Stern \star so zu finden, dass in jeder Zeile und jeder Spalte genau ein Stern steht \leadsto erfolgreicher Abschluss.

	q_1	q_2	q_3	q_4
p_1	0	0	2	3
p_2	0	1	0	1
p_3	1	0	1	2
p_4	2	1	1	0

Reduzierte Matrix M^2



	q_1	q_2	q_3	q_4
p_1	0	0	2	3
p_2	0	1	0	1
p_3	1	0	1	2
p_4	2	1	1	0

$n = N = 4$ in M^2



	q_1	q_2	q_3	q_4
p_1	\star	0	2	3
p_2	0	1	\star	1
p_3	1	\star	1	2
p_4	2	1	1	\star

Zuordnungen in M^2

- Die gefundene Zuordnung ist hier also $\{ (p_1, q_1), (p_2, q_3), (p_3, q_2), (p_4, q_4) \}$ mit den Verschiebungskosten $2 + 2 + 1 + 1 = 6$ aus der Ursprungsmatrix \mathbf{C} .

Ungarische Methode (7)

Jetzt der schwierigere Fall für die Zuordnung zw. $\{p_1, \dots, p_4\}$ und $\{q_1, \dots, q_4\}$.

- Wenn $n < N$, dann Schritt 4: kleinsten unmarkierten Eintrag nutzen:
 - Finde den minimalen Eintrag e_{min} von allen Feldern in \mathbf{C} , die nicht von horizontalen oder vertikalen Linien überdeckt sind,
 - subtrahiere e_{min} von allen Feldern in \mathbf{C} , die nicht von horizontalen oder vertikalen Linien überdeckt sind,
 - addiere e_{min} zu allen Feldern in \mathbf{C} , die von einer horizontalen *und* einer vertikalen Linie überdeckt sind,
 - weiter mit Schritt 3.

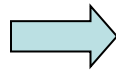
Ungarische Methode (8)

Neues Beispiel zur Demonstration von **Schritt 4**:

- Zunächst Schritte 1 und 2 der Zeilen- und Spaltenumformung:

	q_1	q_2	q_3	q_4
p_1	2	1	4	4
p_2	4	3	3	2
p_3	1	2	3	3
p_4	4	1	3	2

Ausgangsmatrix M^0



	q_1	q_2	q_3	q_4	\min_j
p_1	2	1	4	4	1
p_2	4	3	3	2	2
p_3	1	2	3	3	1
p_4	4	1	3	2	1

Min. Zeileneinträge von M^0



	q_1	q_2	q_3	q_4
p_1	1	0	3	3
p_2	2	1	1	0
p_3	0	1	2	2
p_4	3	0	2	1

Reduzierte Matrix M^1

	q_1	q_2	q_3	q_4
p_1	1	0	3	3
p_2	2	1	1	0
p_3	0	1	2	2
p_4	3	0	2	1

Reduzierte Matrix M^1



	q_1	q_2	q_3	q_4
p_1	1	0	3	3
p_2	2	1	1	0
p_3	0	1	2	2
p_4	3	0	2	1
\min_j	0	0	1	0

Min. Spalteneinträge von M^1



	q_1	q_2	q_3	q_4
p_1	1	0	2	3
p_2	2	1	0	0
p_3	0	1	1	2
p_4	3	0	1	1

Reduzierte Matrix M^2

Ungarische Methode (9)

Weiter neues Beispiel

- Schritt 3: Test auf optimale Zuordnungen liefert zunächst $n = 3 < N = 4$.
- Schritt 4: kleinsten unmarkierten Eintrag nutzen:
 - Finde minimalen Eintrag e_{min} von allen nicht überdeckten Feldern in C ,
 - subtrahiere e_{min} von allen nicht mit Linien überdeckten Feldern,
 - addiere e_{min} zu allen Feldern, die von einer horizontalen *und* einer vertikalen Linie überdeckt sind.

	q_1	q_2	q_3	q_4
p_1	1	0	2	3
p_2	2	1	0	0
p_3	0	1	1	2
p_4	3	0	1	1

Reduzierte Matrix M^2



	q_1	q_2	q_3	q_4
p_1	1	0	2	3
p_2	2	1	0	0
p_3	0	1	1	2
p_4	3	0	1	1

$n = 3 < N = 4$ in M^2

$e_{min} = 1$

	q_1	q_2	q_3	q_4
p_1	1	0	1	2
p_2	3	2	0	0
p_3	0	1	0	1
p_4	3	0	0	0

Reduzierte Matrix M^3

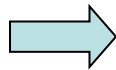
- Weiter mit Schritt 3.

Ungarische Methode (10)

- Weiter neues Beispiel mit **Schritt 3** (nach Schr. 4): **Test auf optimale Zuordnung**
 - Es ist $n = N = 4$. Somit folgt die optimale Zuordnung durch Markierung von Nullen mit einem Stern ★ derart, dass in jeder Zeile und jeder Spalte genau ein Stern steht \leadsto erfolgreicher Abschluss.

	q_1	q_2	q_3	q_4
p_1	1	0	1	2
p_2	3	2	0	0
p_3	0	1	0	1
p_4	3	0	0	0

Reduzierte Matrix M^3



	q_1	q_2	q_3	q_4
p_1	1	0	1	2
p_2	3	2	0	0
p_3	0	1	0	1
p_4	3	0	0	0

$n = N = 4$ in M^3



	q_1	q_2	q_3	q_4
p_1	1	★	1	2
p_2	3	2	★	0
p_3	★	1	0	1
p_4	3	0	0	★

Zuordnungen in M^3

- Die gefundene Zuordnung ist hier also: $\{ (p_1, q_2), (p_2, q_3), (p_3, q_1), (p_4, q_4) \}$.

Ungarische Methode (11)

Zusammenfassung der Schritte:

- 1) **Zeilenumformung**: für jede Zeile z_i wird der **minimale Eintrag \min_i** gesucht und von allen Komponenten der Zeile z_i subtrahiert.
- 2) **Spaltenumformung**: für jede Spalte s_j wird der **minimale Eintrag \min_j** gesucht und von allen Komponenten der Spalte s_j subtrahiert.
- 3) **Test auf optimale Zuordnungen**: die minimale Zahl n von horizontalen und vertikalen Linien wird so in **C** eingezeichnet, dass alle Nullen überdeckt sind
Wenn $n = N$, dann ist die optimale Zuordnung durch Markierung von Nullen mit einem Stern \star so zu finden, dass in jeder Zeile und jeder Spalte genau ein Stern steht \leadsto **erfolgreicher Abschluss**, **sonst weiter mit Schritt 4**.
- 4) **kleinsten unmarkierten Eintrag nutzen**: finde minimalen Eintrag e_{\min} von allen nicht überdeckten Feldern in **C**, **subtrahiere e_{\min} von allen nicht mit Linien überdeckten Feldern**, **addiere e_{\min} zu allen Feldern, die von einer horizontalen und einer vertikalen Linie überdeckt sind**. **Weiter mit Schritt 3**.

Ungarische Methode (12)

- Es gibt auch Fälle, in denen das Ausgangsproblem keine quadratische Form besitzt:

Von Objekt A werden N_A Umrisspunkten ermittelt und von Objekt B werden $N_B \neq N_A$ Umrisspunkten ermittelt.

- In diesen Fällen löst man das Problem
 - durch eine graphentheoretische Version der Ungarischen Methode oder
 - indem die Matrix künstlich quadratisch gemacht wird, indem sog. Dummy-Positionen („keine Position“) eingeführt werden. Dummy-Positionen werden üblicherweise mit solchen Werten belegt, die nicht in der jeweiligen Anwendung auftreten können.

Schätzung der Pose

- Neben der Erkennung, **was ein Objekt ist**,
- wird in vielen Anwendungsbereichen wie der Robotik und der Fahrerassistenz auch gefragt, **wo ein Objekt ist**.

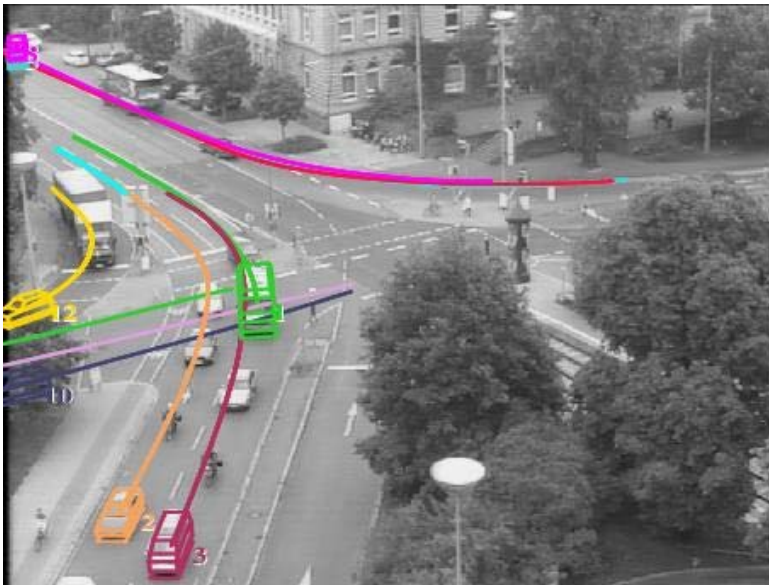


Die Angabe, wo ein Objekt ist, umfasst häufig genauer: **Position und Orientierung**, die unter dem Begriff **Pose** zusammengefasst werden.

Merkmalsbasiertes Objektmodell

Ein Objekt werde durch M Merkmale bzw. unterschiedliche Punkte m_1, m_2, \dots, m_M im dreidimensionalen Raum dargestellt.

Beispiel: für polyederförmige Objekte bzw. polyederförmige Objektapproximationen seien dies alle **Eckpunkte** der Objekte (s. Abb.).



Bildquelle: Univ. Karlsruhe, Inst. für Algorithmen und Kognitive Systeme, FG Prof. Dr. H.-H. Nagel



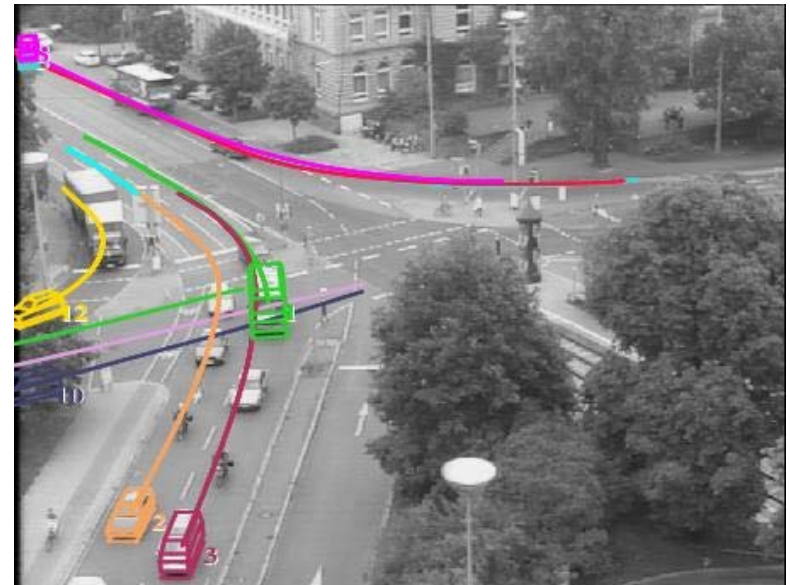
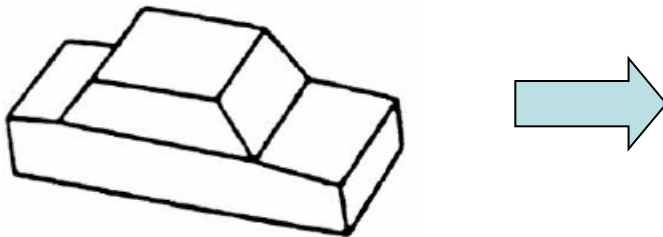
Bildquelle: J. Behley, V. Steinhage: *Generation of 3D City Models Using Domain-Specific Information Fusion*. Proc. 7th Intern. Conf. on Computer Vision Systems, ICVS 2009, Liège, Oct 13-15, 164-173, 2009.

Abbildungsmodell (1)

Die Eckpunkte $\mathbf{m}_1, \mathbf{m}_2, \dots, \mathbf{m}_M$ eines 3D-Objektmodells mit $\mathbf{m}_i = (x_i, y_i, z_i)^T$, $i = 1, \dots, M$,

- werden in 3D-Koordinaten dargestellt,
- werden einer Rotation \mathbf{R} , einer Translation \mathbf{t} sowie einer Projektion Π unterworfen, um Bildmerkmalspunkte $\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_N$ auf der Bildebene entstehen zu lassen*:

$$\mathbf{p}_i = \Pi(\mathbf{R} \mathbf{m}_i + \mathbf{t}) \quad \forall i \in \{1, \dots, M\}.$$



* I.A. gilt: $N \neq M$, da einige Modellpunkte im Bild verdeckt sind oder der Merkmalsdetektor nicht alle Bildmerkmalspunkte findet.

Abbildungsmodell (2)

Das Abbildungsmodell definiert also eine Transformation

$$\mathbf{p}_i = \Pi_P (\mathbf{R} \mathbf{m}_i + \mathbf{t}) = Q(\mathbf{m}_i) \quad \forall i \in \{1, \dots, M\}.$$

Dabei ist \mathbf{R} eine Rotationsmatrix, \mathbf{t} ein Translationsvektor und Π_P eine perspektivische Projektion oder eine ihrer Approximationen wie z.B. die skalierte Orthogonalprojektion Π_{SO} .

Die skalierte Orthogonalprojektion (auch skalierte orthogonale Projektion oder skalierte Parallelprojektion) basiert auf der einfachsten Projektion, nämlich der Orthogonalprojektion und einer Skalierung zur Approximation der perspektivischen Verzeichnung.

Für die Orthogonalprojektion spielt die Translation in die Tiefe keine Rolle, daher zeigt sie 3 Freiheitsgrade in der Rotation und 2 Freiheitsgrade in der Translation. Durch die Skalierung kommt ein 6. Freiheitsgrad hinzu.

Abbildungsmodell (3)

Unter Annahme der skalierten Orthogonalprojektion ist die Transformationsfunktion $Q(m_i)$ mit

$$\mathbf{p}_i = \Pi_{\text{SO}}(\mathbf{R} \mathbf{m}_i + \mathbf{t}) = Q(\mathbf{m}_i) \quad \forall i \in \{1, \dots, M\}$$

a priori unbekannt mit 6 Freiheitsgraden!

Aber: sind für mind. 3 nicht kollineare Modellmerkmalspunkte $\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3$ deren korrespondierende Bildmerkmalspunkte $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3$ bekannt, so ist die Transformation $Q(m_i)$ durch Auflösen eines entsprechenden Gleichungssystems approximierbar.

Also sind zunächst Korrespondenzen zwischen mind. 3 Modellmerkmalspunkten und Bildmerkmalspunkten zu finden.

Ausrichtungsmethode (1)

Für die Bestimmung der Korrespondenzen arbeitet die *Ausrichtungsmethode* nach dem Prinzip des Generierens und Validierens von Hypothesen:

- Zunächst wird *eine initiale Korrespondenz* zwischen einem Tripel (m_1, m_2, m_3) von Modellmerkmalspunkten und einem Tripel (p_1, p_2, p_3) von Bildmerkmalspunkten *geraten* und aus dieser eine Hypothese für die Transformation Q über die Funktion FIND-TRANSFORM geschätzt.
- Ist die Korrespondenz korrekt, muss die geschätzte Hypothese von Q auch die Bildpunkte der restlichen $M-3$ Modellmerkmalspunkte im Bild prädictieren.
- Bei falscher Korrespondenz wird Q falsch geschätzt und die Prädiktion für die restlichen $M-3$ Modellmerkmalspunkte im Bild wird entsprechend falsch sein.

Dies ist die Grundlage für die *Ausrichtungsmethode* (*Alignment Method*).

Ausrichtungsmethode (2)

Die *Ausrichtungsmethode*:

```
function ALIGN (image,model) returns a solution or failure
  inputs: image, a list of image feature points
           model, a list of model feature points
  for each  $p_1, p_2, p_3$  in TRIPLETS (image) do
    for each  $m_1, m_2, m_3$  in TRIPLETS (model) do
       $Q \leftarrow \text{FIND-TRANSFORM} (p_1, p_2, p_3, m_1, m_2, m_3)$ 
      if projection according to  $Q$  explains image then
        return  $Q$ 
  return failure
```

Dies ist die Grundlage für die *Ausrichtungsmethode* (*Alignment Method*).

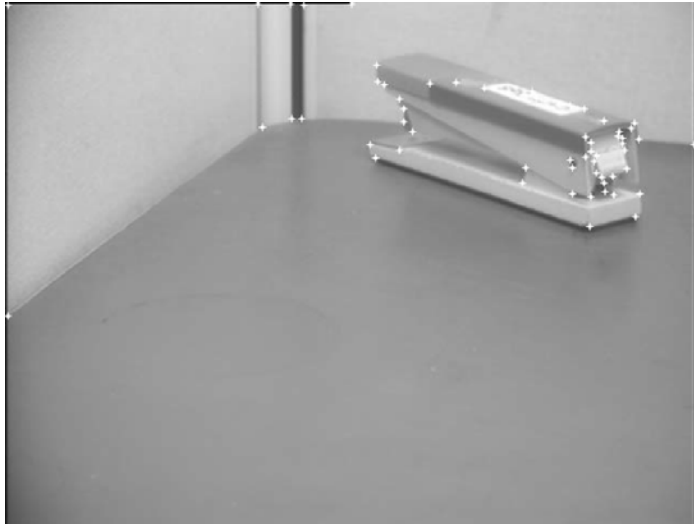
Ausrichtungsmethode (3)

Die Zeitkomplexität der Ausrichtungsmethode für den Worst-Case:

- Proportional zur Zahl der möglichen Modell- und Bildpunkttripel, also $\binom{N}{3}\binom{M}{3}$.
- Die Validierung von $M-3$ modellbasierten Bildpunktprädiktionen durch Vergleich mit den N Bildmerkmalspunkten im Bild ist bei geeigneter Datenstruktur für die Bildmerkmalspunkte möglich in $O(M \log N)$.
- Zusammen also: $O(M^4 N^3 \log N)$.
- Mit Techniken des Posen-Clusterings und Randomisierung ist die Zeitkomplexität reduzierbar auf $O(MN^3)$.

Ausrichtungsmethode (4)

Ergebnisse der Ausrichtungsmethode auf das Tacker-Bild:



Die im Tacker-Bild gefundenen Eckpunkte.



Die hypothetisierte Rekonstruktion.

Unbekannt ist nun noch, wie Eckpunkte in Bildern gefunden werden können.

Ein Ansatz dazu wird in der nächsten Vorlesung vorgestellt.

Zusammenfassung 1

- Im Computersehen wird der Prozess der Zerlegung eines Bildes in Gruppen basierend auf Pixelähnlichkeiten als **Segmentierung** bezeichnet.
 - Verfahren für die Segmentierung können Pixel aufgrund von **Homogenitätskriterien** zu Segmenten gruppieren oder Segmentgrenzen durch **Diskontinuitätskriterien** detektieren.
 - **Region Merging** stellt eine sehr einfache Methode einer Segmentierung nach Homogenitätskriterium dar.
 - **Edge Linking** ist ein Ansatz einer Segmentierung nach Diskontinuitätskriterien, der Kantenpixel auch über **Lücken** in den Pixelnachbarschaften miteinander zu Kantenzügen verbinden kann.

Zusammenfassung 2

- Bei der **Objekterkennung** werden i.A. **domänenspezifische Modelle** der zu erkennenden Objekte oder Objektklassen verwendet.
 - Für einen **konturbasierten Ansatz der Objekterkennung** bietet die Idee der **deformierbaren Übereinstimmung** (*Deformable Matching*) einen Lösungsweg, in dem Umrissähnlichkeiten zwischen den Projektionen von Objektmodellen einerseits und segmentierten Objektabbildungen in Bildvorlagen andererseits bewertet werden.
- Neben der Erkennung von Objekten ist auch häufig deren Pose zu schätzen.
 - Die **Ausrichtungsmethode** (*Alignment Method*) ist ein Ansatz zur Posenabschätzung.