

BTU Cottbus
Lehrstuhl Grafische Systeme
(Prof. Dr. W. Kurth)

Seminar Mustererkennung in Bildern und 3D-Daten

**Objekterkennung mit farbannotierten
Adjazenzgraphen**

Oliver Huras

ohuras@informatik.tu-cottbus.de

25.07.2004

Inhalt:

- 1 Umriss
- 2 Extrahieren von Objektregionen aus Bildern
 - 2.1 Erkennen der ungefähren Regionengrenzen
 - 2.2 Triangulation unter Nebenbedingungen
 - 2.3 Regionenzusammenschluss
- 3 Gewinnung des Graphen eines Objekts
- 4 Die 3 stufige Anpassungsmethode
 - 4.1 Lokaler Vergleich
 - 4.2 Nachbarschaftsvergleich
 - 4.3 Graphenvergleich
 - 4.4 Lösungskriterien
- 5 Ergebnisse
- 6 Quellen

1. Umriss

- Algorithmus zum Finden von Objekten in Bildern
- typische Situation:
 - gegeben ist das Bild eines Objekts („template image“ – Schablone, Vorlage)
 - gesucht ist das Objekt in einem neuen Bild (Suchbild)
- Suchbild wird in Regionen ungefähr gleicher Farben zerlegt
- um geometrische Beziehungen darzustellen, wird ein attributierter Graph erstellt
 - jede Region wird als Knoten dargestellt
 - mit nahen Regionen gibt es eine Kante im Graphen
 - an die Knoten wird Größe, Form und Farbe gehängt

Wie kann man nun Objekte wiederfinden?

- durch Übereinstimmung zweier Graphen (Graph der Vorlage wird versucht mit einem Teilgraphen des neuen Bildes zu matchen)
- nur eine Annäherung, da unterschiedliche Lichtverhältnisse und Perspektiven auf das Objekt

2. Extrahieren von Objektregionen aus Bildern

Ziel: Erkennen/Extrahieren von Regionen („faces“)
annähernd gleicher Farbe von einem Bild mittels
folgender Schritte

2.1 Erkennen der ungefähren Regionengrenzen

- Erkennen der Grenzen, die das Gebiet einschließen (mit Hilfe eines CANNY-Kanten-Detektors)
- Canny, J., 1986. “A computational approach to edge detection”.
IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-8, No.6, Nov. 1986.
- schwierig, die exakte Geometrie der eingeschlossenen Gebiete zu identifizieren

- Lösung: Zuhilfenahme einiger Heuristiken
 - Vereinigung von Linien, die annähernd kollinear verlaufen
 - Erstellung von T-Knotenpunkten bei Linienpaaren, wenn eine Linie im Inneren der anderen Linie endet
 - Schneiden von Linien, bei denen die Endpunkte nahe beieinander liegen mit stumpfem Winkel ($90^\circ - 180^\circ$)
- Erfahrungen haben gezeigt, dass diese Heuristiken entscheidend die Gebietsgrenzen wieder richten können

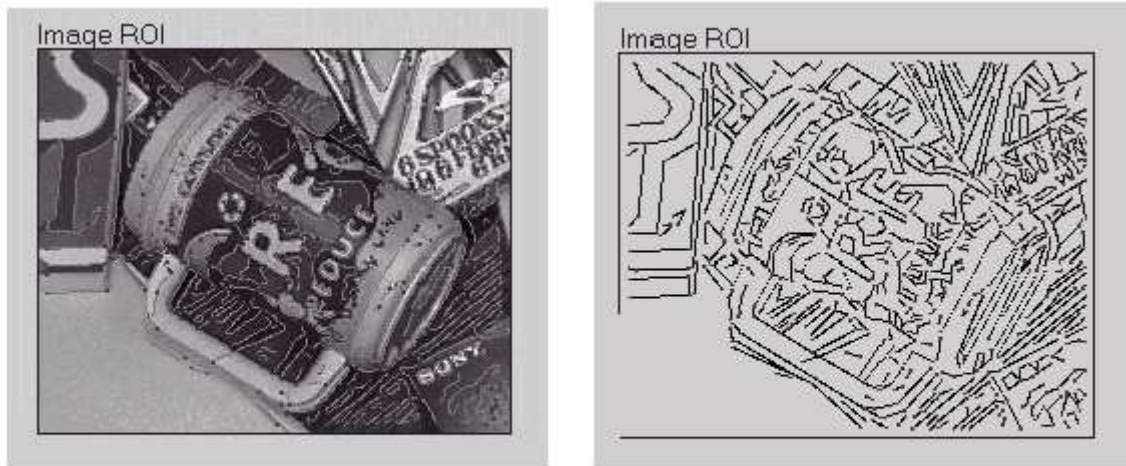


Fig. 1. Left : *The result of edge detection in a template image containing a cup.* Right : *The result of adjusting lines fitted to the cup edge segmentation.*

2.2 Triangulation unter Nebenbedingungen

- mittels der Liniensegmente folgt eine erste Unterteilung des Bildes in Dreiecke gleicher Intensität und Farbe
- Triangulation der Grenzlinien unter Nebenbedingungen, d.h. nahe liegende Punkte werden zu Dreiecken verbunden
- jedes Liniensegment stellt eine Kante eines Dreiecks dar
- weiterhin gilt: jedes Dreieck liegt innerhalb einer Region („face“) bzw. eine Region ist die Vereinigung einer bestimmten Anzahl von Dreiecken

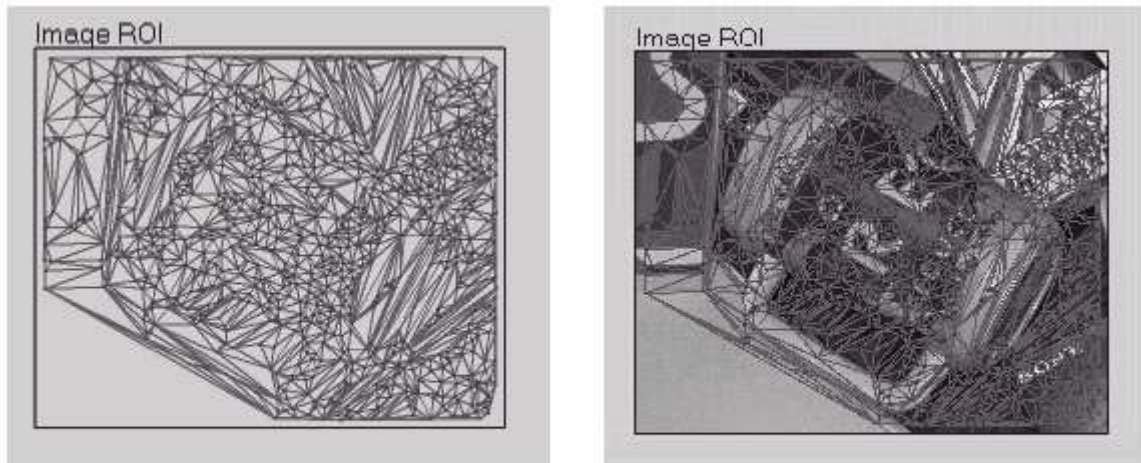


Fig. 2. *Constrained triangulation on the adjusted cup lines. On the left, the triangles only; on the right the triangles superimposed on the image.*

2.3 Regionenzusammenschluss

- inkrementell werden die sichtbaren Objektregionen erzeugt

Wie geschieht das?

- ausgehend vom triangulierten Bild werden benachbarte Regionen vereinigt, wenn mindestens eine der folgenden Eigenschaften erfüllt ist:

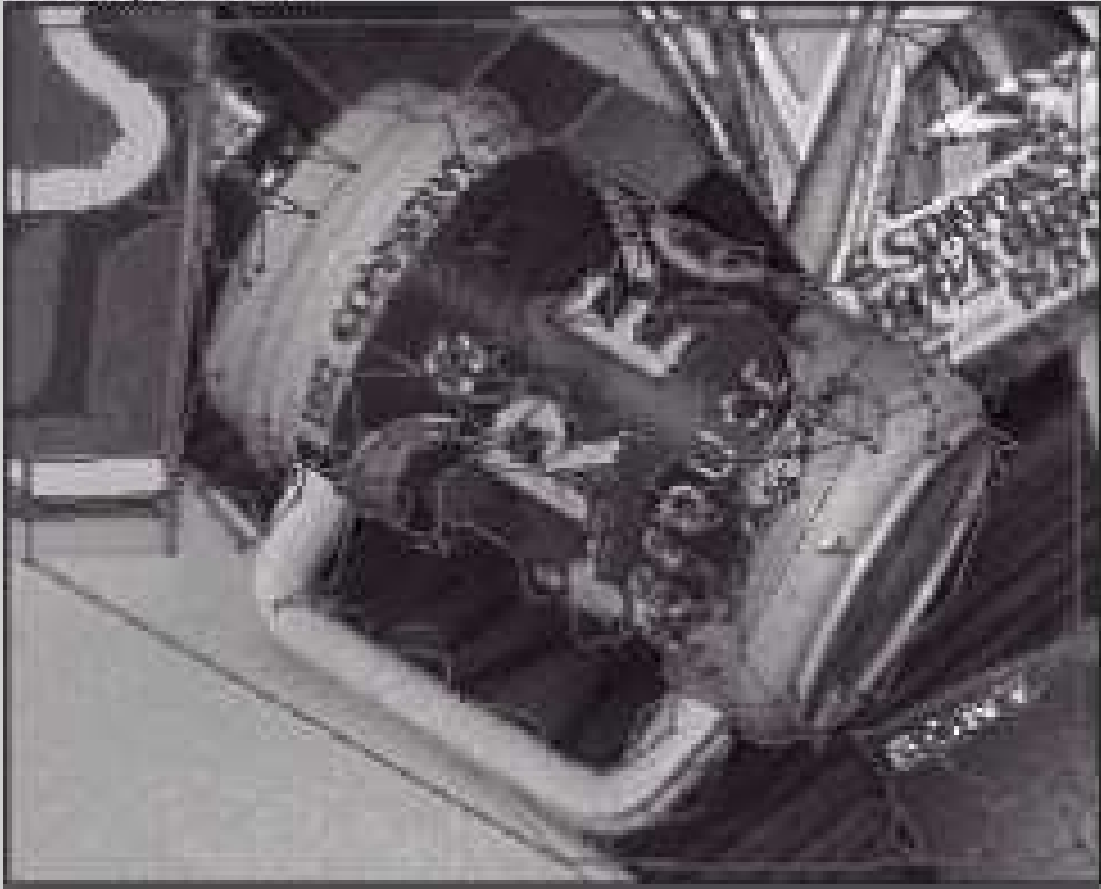
1. Ähnliche Farbintensitäten:

- zwei adjazente Regionen werden vereinigt, wenn der Unterschied zwischen ihren Durchschnitts-Farbintensitäts-Vektoren unterhalb eines Grenzwertes liegen

2. Nicht-unterstützte Brücken („unsupported bridge“):

- zwei adjazente Regionen werden vereinigt, wenn der Prozentsatz der gemeinsamen Kanten zwischen ihnen, die „unsupported“ sind, größer ist als ein Grenzwert
 - *Wann ist eine Kante „supported“?*
 - wenn ein bestimmter Prozentsatz der Pixel dieser Kante zu einem „edgel“ gehören, der durch einen Kanten-Detektor erkannt wurde
- nach jedem Zusammenschließen werden die Eigenschaften der neuen größeren Bilder aus den Eigenschaften der Einzel-Gebiete berechnet (Größe und Farbe)
- wird solange fortgesetzt, bis jedes Paar von benachbarten Regionen unterschiedliche Farb-Intensitäten hat und die meisten Kanten „supported“ sind
- man erhält eine Menge von Regionen mit assoziierten Farbwerten (RGB)
- es kann vorkommen, dass an den Regionengrenzen kleine enge Gebiete liegen, die durch Farbübergänge quer über den Grenzen hervorgerufen werden

Image ROI



3. Gewinnung des Graphen eines Objekts

- Darstellung der Objektregionen als Graph (Adjazenzgraph)
 - jeder Knoten im Graphen ist eine Region (mit Information über Form, Position und Farbeigenschaften)
 - Repräsentation als Ellipse (vollkommen ausreichend)
 - Farbe der Region wird durch einen RGB-Vektor dargestellt
 - aber auch andere Repräsentationen möglich

- da Regionen auch fragmentiert sein können oder auch unpassend zusammengefasst, wäre es wenig vorteilhaft, Kanten zu benutzen, die die adjazenten Regionen im segmentierten gemergten Bild darstellen (Graph wäre gegenüber Veränderungen sehr beschränkt und daher sehr empfindlich)
- anstelle dessen „verbindet“ man jeden Knoten mit den N nächsten Regionen im segmentierten Bild ($N = 8$)

4. Die 3 stufige Anpassungsmethode

- die Suche nach einer Übereinstimmung der beiden Graphen gliedert sich in 3 Phasen:

1. Lokaler Vergleich

- ein 1-zu-1-Vergleich von jedem Knotenpaar wird ausgeführt
- diesem wird ein Punktwert zugeordnet abhängig von der Gleichheit von Form, Größe und Farbe

2. Nachbarschaftsvergleich

- die lokale Nachbarschaft (bestehend aus einem Knoten und seinen Nachbarn) in der Vorlage wird mit der lokalen Nachbarschaft im Suchbild verglichen
- auch hier gibt es wieder Punkte

3. Graphenvergleich

- gut bewertete Vergleiche (aus Punkt 2) werden hier zusammengesetzt und ein (Teil-)Graph-Matching erfolgt

Warum geschieht das Matching 3 stufig?

- damit kann verhindert werden, dass mögliche Übereinstimmungen bereits zeitig ausgeschlossen werden
- Prozess ist sehr robust gegenüber Veränderungen der Segmentierung und der Kameraperspektive

- vergleichbar ist das mit dem Punktesystem beim Tennis
 - mit Punkten macht man Spiel, Satz und Sieg (Match)
 - wer 55 % der Punkte macht, gewinnt 62 % der Spiele, 82 % der Sätze und 97,5 % des Matches

4.1 Lokaler Vergleich

- individuelle Knotenpaare werden ausgewertet und mit Punkten versehen (basierend auf Form und Farbe)
- Form wurde durch Ellipse repräsentiert (bis zu einem Faktor von 2 gibt es bei der Größe keine „Punktabzüge“; das ermöglicht verschiedene Skalierungen)
- der wichtigste Farbunterschied besteht in der Helligkeit; Farbvektoren, die sich nur durch einen konstanten Faktor unterscheiden, werden als gleich gewertet
- Kosten des Vergleichs:

$$C_{local}$$

4.2 Nachbarschaftsvergleich

- jeder Knoten (hier „core vertex“) hat 8 Nachbarn im Graphen
- wenn eine lokale Nachbarschaft (Knoten und Nachbarn) eines „core vertex“ v_0 mit der lokalen Nachbarschaft eines potentiellen Partners v'_0 verglichen wird, versucht man, die Nachbarknoten zu paaren
- v_1, \dots, v_n seien die Nachbarn von v_0 (zyklisch geordnet)
- v'_1, \dots, v'_m seien die Nachbarn von v'_0 (zyklisch geordnet)
- gesucht sind die Untermengen $S \subseteq \{1, \dots, n\}$ und $S' \subseteq \{1, \dots, m\}$ und ein 1-zu-1-Vergleich $\sigma: S \rightarrow S'$, so dass das Matching $v_i \leftrightarrow v'_{\sigma(i)}$ die zyklische Ordnung einhält
- Kosten eines Nachbarschaftsvergleichs:

$$C_{nbhd} = w_0 C_{local}(v_0, v'_0) + \sum_{i \in S} w_i C_{local}(v_i, v'_{\sigma(i)})$$

- w_i ist ein Gewicht zwischen 0 und 1, das vom Verhältnis der Abstände zwischen den „core vertices“ und den Nachbarn v_i und $v'_{\sigma(i)}$ abhängt

4.3 Graphenvergleich

- Ziel: Erstellen einer konsistenten Menge von Knoten-
übereinstimmungen zwischen dem Suchbild und der
Vorlage
- ein neuer Graph wird eingeführt (Assoziationsgraph G ; nicht
zu verwechseln mit dem Adjazenzgraphen der Regionen)
 $G = \{V, E\}$ mit $E \subseteq V \times V$
- ein Knoten $v \in V$ repräsentiert ein mögliches Übereinstimmen
einer Region aus der Vorlage und einer Region aus dem
Suchbild
- bei N Regionen aus der Vorlage und M Regionen im Such-
bild ergeben sich NM Knoten (wird aber meist aufgrund von
Komplexitätsgründen reduziert)
- Knoten v_{ij} stellt die j -te mögliche Zuweisung für die i -te
Region aus der Vorlage dar
- zusätzlich wird ein „slack node“ (loser Knoten) v_{i0} für jede
Region im Template eingeführt, d.h. keine Region aus dem
Suchbild matched
- gewichtete Kanten im Assoziationsgraphen repräsentieren
Kompatibilitäten zwischen den Regionen $e = (v_{ij}, v_{kl})$

- die Gewichte erhält man aus der Kompatibilitätsmatrix C

$$C_{(ij)(kl)} = \begin{cases} 0 & j=0 \vee l=0 \\ 0 & i=k \wedge j \neq l \\ 0 \text{ bis } 1 & (i,j)=(k,l) \\ 0 \text{ bis } 1 & v_{ij} \text{ kompatibel zu } v_{kl} \\ -N & v_{ij} \text{ nicht kompatibel zu } v_{kl} \end{cases}$$

- ein Teilgraph G ist eine Lösung des Matching-Problems

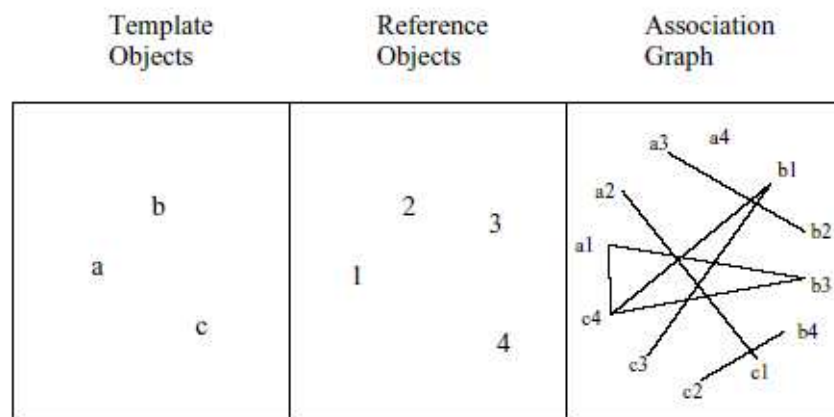


Fig. 4. The template and search images are reduced to a set of regions. Each possible pair of assignments are assigned to a node in the association graph. Edges in the graph connect compatible assignments.

Wie legt man Kompatibilitäten und deren Gewichte fest?

- betrachtet wird eine mögliche Übereinstimmung $R_i \leftrightarrow R'_j$
- in der Stufe „Nachbarschaftsvergleich“ wurde die lokale Nachbarschaft von R_i mit der von R'_j überprüft
- angenommen, eine Menge Nachbarn von R_i wurde mit einer Menge von R'_j gematched

- für die Übereinstimmungen wird eine projektive Transformation H berechnet, die die Mittelpunkte von R_i und R'_j (aber auch die der Nachbarn) mapped
- nun soll ein anderes Kandidatenpaar $R_k \leftrightarrow R'_l$ überprüft werden
- um Kompatibilitäten zwischen v_{ij} und v_{kl} zu berechnen, wird die Übereinstimmung von R'_l mit $H(R_k)$ getestet

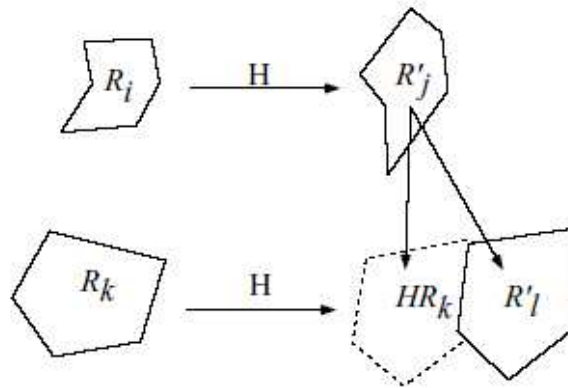


Fig. 5. Compatibility of two matches is determined by applying the transformation H defined by the neighbors of the first pair (R_i, R'_j) to the region R_k belonging to the second pair. The positions of HR_k and R'_l relative to R'_j are compared.

- ein Kompatibilitätswert wird, abhängig vom Winkel und Längenunterschied der beiden Vektoren, der Kante zugewiesen (nicht kompatibel, wenn der Winkel zwischen 2 Vektoren größer 45° oder das Längenverhältnis 2 übersteigt)
- auch ein Farb-Kompatibilitätswert wird berechnet (hier dient eine affine Transformation der Farbräume)

- der Gesamt-Kompatibilitätswert wird berechnet durch

$$C_{(ij)(kl)} = C_{nbhd}(i,j) \times C_{nbhd}(k,l) \times angle_compatibility_score \\ \times length_ration_compatibility_score \\ \times color_compatibility_score$$

4.4 Lösungskriterien

- die größte Knotenmenge in V , die gegenüber einer projektiven Transformation konsistent ist, stellt eine mögliche Lösung dar
- Knoten können trotz dieser Konsistenzen zueinander unpassend sein (vgl. Fig.4: Knoten c4, b1, c3; c kann nicht gleichzeitig mit 3 und 4 gematched werden)
- mit Hilfe einer einfachen Tiefensuche kann der größte zusammenhängende (Teil-)Graph von G gefunden werden (in einem zusammenhängenden Graphen existiert zwischen jedem Knotenpaar ein Kanten-Pfad)
- ein zusammenhängender Graph kann trotzdem nicht als endgültige Lösung angesehen werden, da Anfangs- und Endknoten eines Pfades zueinander inkonsistent sein können
- deshalb Suche nach einer Clique
- eine Untermenge $R \subseteq V$ ist eine Clique auf G , wenn

$$\forall v_{ij}, v_{kl} \in R: (v_{ij}, v_{kl}) \in E$$
- die Suche nach einer maximalen Clique ist NP-vollständig

- es wurde herausgefunden, dass die Bestimmung der maximalen Clique analog zur Suche des globalen Maximums einer binären quadratischen Funktion ist
- F. Lin und M. Pelillo nutzen „Relaxation“ und „Neuronale Netze“, um sich dem globalen Maximum einer quadratischen Funktion anzunähern (das Maximum entspricht der größten Clique in einem Assoziationsgraphen)
- obwohl die größte Clique ein hohes Maß an gegenseitiger Konsistenz aufweist, gehen Feinheiten in der Kompatibilitätsmatrix C verloren
- zur Lösung dieses Problems ist es notwendig, eine quadratische Formel zu erstellen, wo das globale Maximum der Clique in G entspricht, die die maximale Summe der Kantengewichte hat
- eine Annäherung basiert auf „Gold and Rangarjans’s gradual assignment algorithm“ (GAA), ein iterativer Optimierungsalgorithmus

5. Ergebnisse

- der Algorithmus wurde auf einige Farbbilder angewendet
- Beispiel eines Computer-Handbuchs (Fig.7, Fig.8), welches in verschiedenen Bildern wiedergefunden wurde, selbst, wenn das Handbuch teilweise verdeckt war

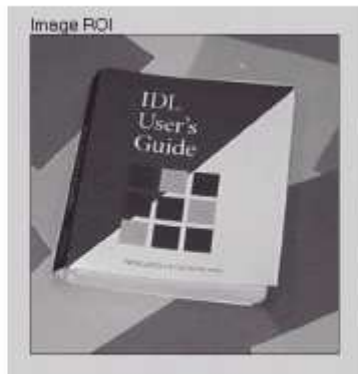


Fig. 7. *The computer manual used as a template*

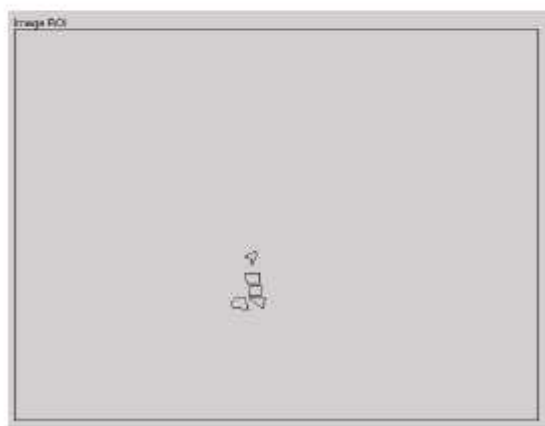
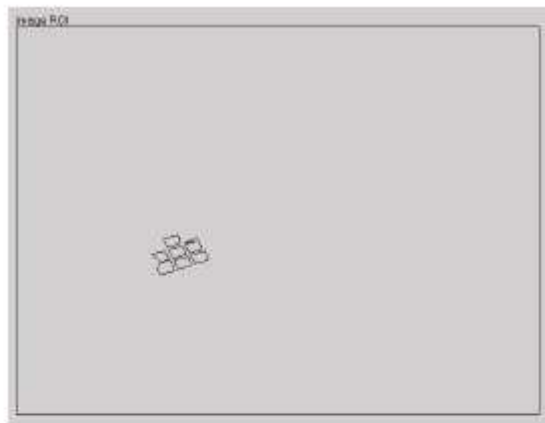


Fig. 8. *Two examples of recognition. On the left the search image, and on the right the outlines of the regions matched against the template.*

6. Quellen

Recognizing Objects Using Color-Annotated Adjacency Graphs

By Peter Tu, Tushar Saxena, Richard Hartley

<http://www.cs.albany.edu/~saxena/>

Tushar Saxena's Webseite (University at Albany [NY])

<http://rsise.anu.edu.au/~hartley/>

Richard Hartley's Webseite (Australian National University)