

# Objekt-Tracking mit $n:m$ -Matching

Mark Ross

Lutz Priebe

Labor Bilderkennen  
Universität Koblenz  
56075 Koblenz, Germany  
{ross,priebe}@uni-koblenz.de  
<http://www.uni-koblenz.de/~lb/>

## Zusammenfassung

Beim  $n:m$ -Matching werden Korrespondenzen von Farbregionen zwischen aufeinander folgenden Bildern einer Sequenz ermittelt. Gegenstand dieser Arbeit ist die Steigerung der Effizienz beim Matching durch Ausnutzung von Wissen aus vorhergehenden Bildern. Durch einen neuen Ansatz konnte die Korrespondenzanalyse auf die Hälfte der Verarbeitungszeit beschleunigt werden.

## 1 Einleitung

Regionenbasierte Bewegungssegmentierungen (z. B. [Reh98], [HKR97] oder [Sch00]) sind effizient, robust und liefern dabei dichte Verschiebungsvektorfelder. Um ein Bild in Regionen zu partitionieren ist eine Segmentierung oder ein Clustern [HKR97] als Initialschritt notwendig. Aber kein Segmentierungsalgorithmus - und sei er noch so gut - kann gewährleisten, dass die zugehörigen Segmente von bewegten, natürlichen Objekten über eine Folge von Bildern in ihrer Form und Anzahl nicht variieren.

Dennoch können Bildbereiche durch das so genannte  $n:m$ -Matching [Reh98] in einer Sequenz robust verfolgt werden, da hier nicht nur versucht wird, zu einzelnen Segmenten das korrespondierende Segment im Folgebild zu finden (1:1-Match) - vielmehr werden Segmentcluster aus  $n$  Segmenten kombinatorisch gebildet, für die nun im Folgebild jeweils ein korrespondierendes Segmentcluster aus  $m$  Segmenten ermittelt wird ( $n:m$ -Match, s. Abb. 2). Aus Effizienzgründen wird zuerst versucht, den trivialen Fall des 1:1-Matches vorab zu behandeln, um so die Anzahl der Segmente, aus denen für den  $n:m$ -Fall Cluster gebildet werden, zu reduzieren.

Mit den so gefundenen Korrespondenzen kann nun die Bewegung (z. B. Translation) der einzelnen Bildbereiche (Segmente) berechnet werden. Anschließend werden zusammenhängende Bildbereiche mit ähnlicher Bewegung zu Bewegungsobjekten zusammengefasst.

Um die Effizienz der Korrespondenzanalyse zu erhöhen wird jetzt Cluster- und Matching-Information aus vorherigen Bildern ausgenutzt: Jedes im vorherigen Bild zugeordnete Segmentcluster hat bei der Verarbeitung des nächsten Bildes eine höhere Wahrscheinlichkeit, weiter zugeordnet zu werden, als irgend ein anderes, zufällig gebildetes Segmentcluster. Es wird daher zuerst versucht, Segmentcluster, die in der Vergangenheit erfolgreich zugeordnet wurden, weiter zu verfolgen (Tracking). Anschließend werden Korrespondenzen für noch nicht zugeordnete Segmente bzw. Segmentcluster ermittelt.

## Begriffe und Notationen

Einige in dieser Arbeit oft benutzte, wichtige Begriffe (Segment, Zusammenhang u. a.) sind im Anhang definiert. Es bezeichnet im Folgenden:

$I_t$  das Bild zum diskreten Zeitpunkt  $t \geq 0$

$\mathcal{S}_t$  eine Menge von disjunkten Segmenten zum diskreten Zeitpunkt  $t \geq 0$

$\mathcal{C}_t$  eine Menge von nichtdisjunkten Segmentclustern zum diskreten Zeitpunkt  $t \geq 0$

$\mathcal{T}_t \subseteq \mathcal{C}_t$  die Menge der disjunkten Segmentcluster zum diskreten Zeitpunkt  $t \geq 0$ , die einen Matchpartner zum Zeitpunkt  $t - 1$  haben.

## 2 Bildverarbeitungskette

### 2.1 Bildvorverarbeitung

Die Bildvorverarbeitung besteht aus einer nichtlinearen Farb-Filterung, der Bildsegmentierung und einer Merkmalsextraktion der Farbsegmente.

Zur kantenerhaltenden Rauschunterdrückung findet der Symmetric Nearest Neighbourhood Filter von Pietikäinen und Harwood [PH86] Verwendung; es ist aber prinzipiell auch jeder andere, nicht-lineare Filter mit rauschunterdrückenden, kantenerhaltenden Eigenschaften (z. B. Medianfilter oder Filter von Nagao und Matsuyama [NM79]) geeignet.

Zur Segmentierung hat sich der Color Structure Code (CSC [RP98]) als robustes und effizientes Verfahren bewährt<sup>1</sup>. Bei den hier untersuchten Straßenverkehrssequenzen wird als Homogenitätskriterium eine Farbähnlichkeitstabelle im HSV-Farbraum verwendet, die eine Untersegmentierung der relevanten Bildobjekte vermeidet. Eine Übersegmentierung hingegen ist aufgrund des späteren Zusammenfassens von mehreren Segmenten zu Clustern – zumindest in gewissen Grenzen – unkritisch.

Für jedes Farbsegment  $s \in \mathcal{S}_t$  werden nun die charakteristischen Merkmale

- Durchschnittlicher Farbwert,
- Position (Lage des Schwerpunktes),
- Größe (Anzahl der zugehörigen Pixel),
- Form (Länge von Haupt- und Nebenachse der Äquivalenten Ellipse) und
- Orientierung (Richtung der Hauptachse der Äquivalenten Ellipse)

berechnet. Aus diesen Merkmalen werden die Ähnlichkeitswerte für Segmentpaare beim Matching bestimmt.

Zur Gewährleistung des räumlichen Zusammenhangs von Segmentenclustern werden außerdem noch alle Nachbarschaftsrelationen von je zwei Farbsegmenten  $s_i, s_j \in \mathcal{S}_t$  ermittelt (vgl. [Ros03]) und gespeichert.

### 2.2 Klassisches $n:m$ -Matching

Ziel des  $n:m$ -Matching ist es, Korrespondenzen von Bildregionen zwischen zwei Bildern zu bestimmen [Reh98]. Aufgrund der Variationen der Segmentierung von einem Bild zum nächsten wird hier nicht nur versucht, zu einzelnen Segmenten  $s_i \in \mathcal{S}_{t-1}$  ein korrespondierendes Segment  $s_j \in \mathcal{S}_t$  im Folgebild zu finden (1:1-Match); vielmehr werden Segmentcluster kombinatorisch gebildet, für die nun im Folgebild ein korrespondierendes Segmentcluster ermittelt wird. D. h. es werden Korrespondenzen von  $n$  Segmenten des einen Bildes  $I_{t-1}$  mit  $m$  Segmenten des Folgebilds  $I_t$  ermittelt [Reh98]. Da es wegen des hohen Aufwandes (Rechenzeit) nicht möglich ist, für ein Bild alle Kombinationen

---

<sup>1</sup>Der CSC-Algorithmus ist als Download verfügbar: <http://www.uni-koblenz.de/~lb/>

von Segmentclustern (bei 100 Segmenten  $2^{100}$  mögliche Cluster) zu bilden, werden nur Cluster aus Segmenten gebildet, die

- zueinander *farblich ähnlich* und
- untereinander *zusammenhängend* sind und
- eine *ähnliche Bewegungsvergangenheit* haben.

Außerdem gibt es in Bildsequenzen oft einige Bildbereiche, deren Segmentierung stabil ist. Es läßt sich dadurch die Anzahl der Segmente, aus denen für den  $n:m$ -Fall Cluster gebildet werden, reduzieren, indem zunächst versucht wird, den trivialen Fall des 1:1-Matches zu behandeln. Segmente, die dann mit sehr hoher Güte bereits einen Partner beim 1:1-Match gefunden haben, werden dann nicht mehr zu Segmentclustern kombiniert und bleiben beim weiteren  $n:m$ -Matching unberücksichtigt.

Die explizite Vorwegverarbeitung von 1:1-Matches kann dazu führen, dass anstelle der optimalen Matchpartner nur suboptimale Partner gefunden werden. Dies ist allerdings durch den damit erheblich reduzierten Aufwand gerechtfertigt.

Abbildung 1 zeigt im mittleren Pfad ( $t=1$ ) die Verarbeitungsschritte des klassischen  $n:m$ -Matchings, die im Folgenden detailliert beschrieben werden. Die Erweiterung des Systems ist im rechten Pfad ( $t \geq 2$ ) dargestellt<sup>2</sup> und wird im nächsten Abschnitt erläutert.

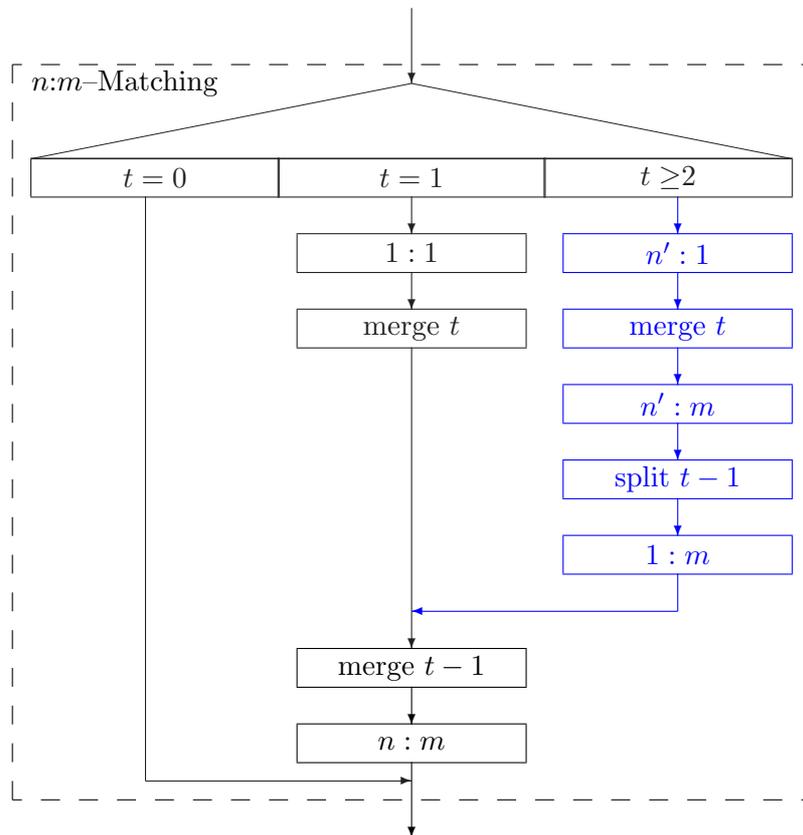


Abbildung 1: Verarbeitungsschritte des  $n:m$ -Matchings  
Mittlerer Pfad ( $t=1$ ): klassisches  $n:m$ -Matching (Kap. 2.2)  
Rechter Pfad ( $t \geq 2$ ):  $n:m$ -Matching mit Tracking (Kap. 2.3)

Im Modul 1:1 werden Korrespondenzen  $K_{1:1}$  mit sehr hoher Güte<sup>3</sup> zwischen einzelnen Segmenten der Bilder  $I_{t-1}$  und  $I_t$  ermittelt (1:1-Matching) und die beiden Segmentmengen um

<sup>2</sup>Ein Tracking erfolgt erst ab dem dritten Bild der Sequenz ( $t \geq 2$ )

<sup>3</sup>Hoher Ähnlichkeitswert bzgl. der charakteristischen Merkmale (vgl. 2.1).

die zugeordneten Segmente reduziert.

$$\begin{aligned} K_{1:1} &= \{(c, d) \in \mathcal{S}_{t-1} \times \mathcal{S}_t \mid \forall (c', d') \in \mathcal{S}_{t-1} \times \mathcal{S}_t : \text{Güte}(c, d) \geq \text{Güte}(c', d')\} \\ \mathcal{S}'_t &= \mathcal{S}_t - \{s \in \mathcal{S}_t \mid s \text{ hat Korrespondenz in } I_{t-1}\} \\ \mathcal{S}'_{t-1} &= \mathcal{S}_{t-1} - \{s \in \mathcal{S}_{t-1} \mid s \text{ hat Korrespondenz in } I_t\} \end{aligned}$$

Im Schritt merge  $t$  wird aus der reduzierten Segmentmenge  $\mathcal{S}'_t$  eine Menge  $\mathcal{C}_t^*$  ( $= \mathcal{C}_t^5$ ) nicht-disjunkter Segmentcluster<sup>4</sup> iterativ gebildet:

$$\begin{aligned} \mathcal{C}_t^1 &= \{\{s\} \mid s \in \mathcal{S}'_t\} \\ \mathcal{C}_t^{k+1} &= \mathcal{C}_t^k \cup \left\{ c \cup \{s\} \mid c \in \mathcal{C}_t^k \wedge s \in \mathcal{S}'_t \wedge c \text{ farbähnlich zu } s \wedge c \text{ benachbart mit } s \right\} \end{aligned}$$

Es werden dabei nur Cluster gebildet, die zusammenhängend sind und deren beteiligte Segmente untereinander farbähnlich sind. Die Merkmale der Segmentcluster können unmittelbar aus den zugehörigen Segmenten berechnet werden. Analog ist merge  $t-1$ .

Im Modul  $n:m$  werden Korrespondenzen  $K_{n:m}$  zwischen Segmentclustern der Mengen  $\mathcal{C}_{t-1}^*$  und  $\mathcal{C}_t^*$  ermittelt. Im möglichen Konfliktfall, dass es für mehrere Segmentcluster einen möglichen Matchpartner gibt, aber diese Segmentcluster ein oder mehrere gemeinsame Segmente beinhalten, setzt sich die Korrespondenz mit der besten Güte durch.

$$\begin{aligned} K_{n:m} &= \{(c, d) \in \mathcal{C}_{t-1}^* \times \mathcal{C}_t^* \mid \forall c' \in \mathcal{C}_{t-1}^*, \forall d' \in \mathcal{C}_t^* : \\ &\quad (c' \cap c \neq \emptyset \vee d' \cap d \neq \emptyset) \wedge (c \neq c' \vee d \neq d') \implies \text{Güte}(c, d) > \text{Güte}(c', d')\} \end{aligned}$$

Für das im folgenden Abschnitt beschriebene Tracking werden nun alle zugordneten Segmente und Segmentcluster des Bildes  $I_t$  in die Menge  $\mathcal{T}_t$  von disjunkten Tracking-Objekten genommen.

$$\mathcal{T}_t = \{\{s\} \mid s \in \mathcal{S}_t - \mathcal{S}'_t\} \cup \{c \in \mathcal{C}_t^* \mid \exists c' \in \mathcal{C}_{t-1}^* : (c', c) \in K_{n:m}\}$$

Genau die nicht zugeordneten (einzelnen) Segmente – die also weder selbst, noch als Bestandteil eines Clusters zugeordnet wurden – verbleiben in der Segmentliste  $\mathcal{S}_t$ .

## 2.3 $n:m$ -Matching mit Tracking

### 2.3.1 Motivation

Das klassische  $n:m$ -Matching ermittelt jeweils zwischen zwei aufeinanderfolgenden Bildern einer Bildsequenz Korrespondenzen, ohne dabei die Matching- oder Cluster-Information aus der Vergangenheit zu berücksichtigen. Um die Verarbeitungsgeschwindigkeit zu erhöhen wird diese Information nun ausgenutzt und das System dadurch zu einem Tracking-System erweitert.

Ein Segmentcluster  $c_i \in \mathcal{T}_t$ , das sich bereits im vorherigen Bild  $I_{t-1}$  zugordnet ließ, läßt sich bei der Verarbeitung des nächsten Bildes  $I_{t+1}$  mit höherer Wahrscheinlichkeit weiter zuordnen, als irgend ein anderes, zufällig gebildetes Segmentcluster  $c_j \in (\mathcal{C}_t - \mathcal{T}_t)$ . Es wird jetzt also zuerst versucht, Segmentcluster, die in der Vergangenheit erfolgreich zugeordnet wurden, weiter zu verfolgen (Tracking). Anschließend werden Korrespondenzen für noch nicht zugeordnete Segmente ermittelt.

Auch die Matching-Güte läßt sich durch die Nutzung der Information aus vergangenen Zuordnungen verbessern. Wenn sich z. B. ein Objekt von der Kamera entfernt, werden die zugehörigen Segmente von Bild zu Bild stetig kleiner. Damit läßt sich eine Objektprädiktion bezüglich seiner Größe machen, die beim Matching genutzt werden kann. Die Verbesserung der Matching-Güte ist Gegenstand einer zukünftigen Arbeit.

<sup>4</sup>Zur Aufwandsbegrenzung werden nur Cluster mit einer max. Segmentanzahl gebildet (hier:  $\forall c \in \mathcal{C}_t^* : \|c\| \leq 5$ ).

### 2.3.2 Beschreibung

Sowohl das Tracking wie auch das Zuordnen noch nicht verfolgter Segmentcluster erfolgt wie das klassische  $n:m$ -Matching aus Effizienzgründen in jeweils zwei Schritten, nämlich Matching mit Segmenten und Matching mit Clustern. Abbildung 1 zeigt im Pfad  $t \geq 2$  die einzelnen Verarbeitungsphasen des  $n:m$ -Matchings mit Tracking. Dabei symbolisiert  $n'$  ein Cluster, das aus  $n' \geq 1$  Segmenten besteht und eine Bewegungsvergangenheit aufweist.

Zunächst wird in  $\boxed{n':1}$  versucht, Korrespondenzen zwischen Tracking-Objekten  $c \in \mathcal{T}_{t-1}$  im Bild  $I_{t-1}$  und einzelnen Segmenten  $s \in \mathcal{S}_t$  im neuen Bild  $I_t$  zu finden. Gefundene Korrespondenzen werden aus der Menge der Tracking-Objekte  $\mathcal{T}_{t-1}$  bzw. der Segmentliste  $\mathcal{S}_t$  entfernt.

$$\begin{aligned}\mathcal{S}'_t &= \mathcal{S}_t - \{s \in \mathcal{S}_t \mid s \text{ hat Korrespondenz in } I_{t-1}\} \\ \mathcal{T}'_{t-1} &= \mathcal{T}_{t-1} - \{s \in \mathcal{T}_{t-1} \mid s \text{ hat Korrespondenz in } I_t\}\end{aligned}$$

Beim  $\boxed{\text{merge } t}$  wird aus der Segmentliste  $\mathcal{S}'_t$  die Clusterliste  $\mathcal{C}_t^* (= \mathcal{C}_t^5)$  iterativ erzeugt.

$$\begin{aligned}\mathcal{C}_t^1 &= \{\{s\} \mid s \in \mathcal{S}'_t\} \\ \mathcal{C}_t^{k+1} &= \mathcal{C}_t^k \cup \left\{c \cup \{s\} \mid c \in \mathcal{C}_t^k \wedge s \in \mathcal{S}'_t \wedge c \text{ farbähnlich zu } s \wedge c \text{ benachbart mit } s\right\}\end{aligned}$$

Die Menge nicht-disjunkter Cluster  $\mathcal{C}_t^*$  liefert die potentiellen Matchpartner für die Tracking-Objekte der Trackingliste  $\mathcal{T}'_{t-1}$  in der Phase  $\boxed{n':m}$ . Die Mengen  $\mathcal{T}'_{t-1}$  und  $\mathcal{C}_t^*$  werden dann entsprechend der ermittelten Korrespondenzen reduziert.

Die Phase  $\boxed{\text{split } t-1}$  bedarf genauerer Erklärung: Für das Bild  $I_{t-1}$  existiert nun eine Trackingliste  $\mathcal{T}'_{t-1}$ , die genau dann leer ist, wenn alle Tracking-Objekte im Folgebild  $I_t$  zugeordnet wurden. Außerdem gibt es eine Segmentliste  $\mathcal{S}_{t-1}$ , die genau dann leer ist, wenn beim Matching des vorherigen Bildes  $I_{t-2}$  allen Segmenten ein Matchpartner zugeordnet wurde. Beide Listen sind jedoch i. A. nicht leer. Aus den verbleibenden Clustern der Trackingliste  $\mathcal{T}_{t-1}$  werden die beteiligten Segmente extrahiert und der Segmentliste  $\mathcal{S}_{t-1}$  zugefügt.

$$\mathcal{S}'_{t-1} = \mathcal{S}_{t-1} \cup \{s \mid \exists c \in \mathcal{T}_{t-1} : s \in c\}$$

Die Trackingliste  $\mathcal{T}_{t-1}$  behält die noch nicht zugeordneten Cluster (s.u.).

Das  $\boxed{1:m}$  arbeitet auf der Segmentliste  $\mathcal{S}'_{t-1}$  und der Clusterliste  $\mathcal{C}_t$ . Werden hier Korrespondenzen gefunden, dann werden die beiden Listen entsprechend reduziert. Außerdem wird die Trackingliste  $\mathcal{T}_{t-1}$  um die Cluster reduziert, von denen ein oder mehrere zugehörige Segmente zugeordnet wurden.

$$\begin{aligned}\mathcal{S}''_{t-1} &= \mathcal{S}'_{t-1} - \{s \in \mathcal{S}'_{t-1} \mid s \text{ hat Korrespondenz in } I_t\} \\ \mathcal{C}'_t &= \mathcal{C}_t - \{c \in \mathcal{C}'_t \mid c \text{ hat Korrespondenz in } I_{t-1}\} \\ \mathcal{T}''_{t-1} &= \mathcal{T}'_{t-1} - \{c \in \mathcal{T}'_{t-1} \mid \exists s \in c : s \text{ hat Korrespondenz in } I_t\}\end{aligned}$$

Beim  $\boxed{\text{merge } t-1}$  werden aus der Segmentliste  $\mathcal{S}_{t-1}$  eine Menge  $\mathcal{C}_{t-1}^*$  von Segmentclustern, die noch nicht in der Trackingliste existieren, gebildet. Es wird hier verhindert, dass ein Cluster, das bereits eine Bewegungsvergangenheit besitzt und beim Tracking nicht zugeordnet werden konnte, hier wieder neu gebildet wird und damit die Vergangenheitsinformation verliert<sup>5</sup>. Da beim Tracking eine sehr hohe Güte gefordert wird, ist es möglich, dass ein solches Cluster erst in der letzten Matching-Phase zugeordnet werden kann.

Im Modul  $\boxed{n:m}$  werden Korrespondenzen zwischen noch nicht zugeordneten Segmentclustern  $c \in \mathcal{C}_{t-1} \cup \mathcal{T}_{t-1}$  und  $c' \in \mathcal{C}_t$  ermittelt.

Alle zugeordneten Cluster des Bildes  $I_t$  bilden die Menge  $\mathcal{T}_t$  von disjunkten Tracking-Objekten für die Verarbeitung des nächsten Bildes. Nur die nicht zugeordneten Segmente verbleiben in der Segmentliste  $\mathcal{S}_t$ .

<sup>5</sup>Es wäre allerdings effizienter, diese Cluster neu zu bilden, als nach deren Existenz in der Trackingliste zu suchen.

## 2.4 Bewegungsschätzung

Die Bewegungsschätzung umfasst zwei Verarbeitungsstufen: die Bestimmung der Bewegung der jeweiligen Segmente und die Bewegungssegmentierung, bei der Segmente mit ähnlicher Bewegung zu Bewegungsobjekten zusammengefasst werden. Für nähere Information sei auf [Reh98] und [Ros03] verwiesen.

## 3 Evaluation

Das Effizienzverhalten des  $n:m$ -Matchings wurde an 4 Bildsequenzen mit insgesamt über 300 Bildern der Größe  $340 \times 275$  untersucht. Dabei benötigt die gesamte Bildvorverarbeitung im Durchschnitt etwa 80 ms (Filterung 15 ms, CSC-Segmentierung 55 ms, Merkmalsextraktion 5-10 ms)<sup>6</sup>.

In Tabelle 1 sind die durchschnittliche Verarbeitungszeit, deren Standardabweichung und deren Maximalwert für das  $n:m$ -Matching mit und ohne Tracking angegeben<sup>7</sup>. Die Einführung des Trackings zeigt eine deutliche Effizienzsteigerung. Die mittlere Rechenzeit konnte hier für alle 4 Sequenzen durch das Tracking auf etwa die Hälfte (oder sogar noch weniger) im Vergleich zum Matching ohne Tracking reduziert werden. Auch die Varianz und der Maximalwert der Rechenzeit wurde für alle Sequenzen signifikant reduziert.

	$\mu$	$\sigma$	max.
Sequenz 0002, 36 Bilder			
Klassisches $n:m$ -Matching	18,6 ms	3,2 ms	23,4 ms
$n:m$ -Matching mit Tracking	9,4 ms	1,6 ms	11,1 ms
Sequenz 0015, 125 Bilder			
Klassisches $n:m$ -Matching	20,3 ms	3,5 ms	44,7 ms
$n:m$ -Matching mit Tracking	10,4 ms	2,6 ms	30,9 ms
Sequenz 0030, 100 Bilder			
Klassisches $n:m$ -Matching	19,0 ms	2,2 ms	24,7 ms
$n:m$ -Matching mit Tracking	9,8 ms	1,1 ms	12,7 ms
Sequenz sz3, 50 Bilder			
Klassisches $n:m$ -Matching	18,7 ms	2,7 ms	20,6 ms
$n:m$ -Matching mit Tracking	9,7 ms	1,4 ms	10,8 ms

Tabelle 1: Vergleich der Verarbeitungszeiten des  $n:m$ -Matchings mit und ohne Tracking. Es sind jeweils Durchschnitt  $\mu$ , Standardabweichung  $\sigma$  und Maximum der Rechenzeit<sup>6</sup> angegeben.

## 4 Ausblick

### 4.1 Weitere Nutzung von Vergangenheitsinformation

Durch die Objektverfolgung bestehen vielseitige Möglichkeiten, Information beim  $n:m$ -Matching aus der Vergangenheit auszunutzen. Dadurch konnte bereits eine deutliche Effizienzsteigerung erreicht werden.

Außerdem wurden bereits Überlegungen unternommen, auch die Matching-Güte weiter zu verbessern. Dies wird in Zukunft untersucht. Hier sollen beispielhaft zwei Möglichkeiten aufgezeigt werden.

<sup>6</sup>Normaler Arbeitsplatzrechner: Intel Pentium 4 mit 2,4 GHz und 512 MB, SuSE Linux 8.2

<sup>7</sup>Die Verarbeitungszeiten variieren mit der Bildkomplexität der jeweiligen Szenen. Je größer die Anzahl der Segmente in einem Bild ist und je mehr benachbarte Segmente farbähnlich sind, umso größer ist die Verarbeitungszeit.

**Beispiel:** Beim 1:1-Matching werden Korrespondenzen mit hoher Güte detektiert. Sei die Verteilung der Matchgüten  $q_{i,t}$  eines Clusters  $c_i$  über mehrere Bilder durch die mittlere Güte  $\mu_{i,t}$  und die Varianz  $\sigma_{i,t}$  charakterisiert, kann man mit hoher Wahrscheinlichkeit annehmen, dass der nächste Match mit einer Güte  $q_{i,t+1}$  von mehr als  $\mu_{i,t} - k \cdot \sigma_{i,t}$  mit z. B.  $k = 3$  erfolgt. Dementsprechend kann man den Schwellwert 'hohe Güte' für das Cluster  $c_i$  auf den Wert  $\mu_{i,t} - k \cdot \sigma_{i,t}$  setzen. ■

**Beispiel:** Die Matchgüte  $q$  bestimmt sich aus einer gewichteten Summe einzelner Güten  $q_i$  für Fläche, Farbe, Position, Proportion und Orientierung [Reh98].

$$q = \sum_i \omega_i \cdot q_i \quad \text{mit} \quad \sum_i \omega_i = 1$$

Wurde ein Cluster  $c$  bereits über mehrere Bilder verfolgt, so lassen sich die Gewichtungsfaktoren derart anpassen, dass stabile Merkmale höher gewichtet werden als instabile. Die Stabilität der Merkmale resultiert dabei aus den jeweiligen Güten  $q_i$ . ■

## A Online-Dienste

Der CSC-Segmentierungsalgorithmus, der SNN-Filter [PH86] und der Filter von Nagao [NM79] können unter [http://www.uni-koblenz.de/~lb/lb\\_downloads/](http://www.uni-koblenz.de/~lb/lb_downloads/) geladen und unter <http://bit.uni-koblenz.de/vibi/> online getestet werden.

Beispielsequenzen des klassischen  $n:m$ -Matching [Reh98] finden sich unter: [http://www.uni-koblenz.de/~lb/lb\\_motion/motion.html](http://www.uni-koblenz.de/~lb/lb_motion/motion.html)

## B Definitionen

**Position:**  $\mathcal{L} = [0, N[ \times [0, M[ \cap \mathbb{N}^2$

$$p \in \mathcal{L} \Rightarrow \exists p_1 \in [0, N[ \exists p_2 \in [0, M[ : p = (p_1, p_2)$$

**Farbe:**  $\mathcal{V}_{RGB} = [0, 255]^3$  (RGB-Farbraum) oder  $\mathcal{V}_{HSV} = [0, 359] \times [0, 255] \times [0, 255]$  (HSV-Farbraum)

**Bild:**  $I : \mathcal{L} \rightarrow \mathcal{V}$

**Pixel:**  $\mathcal{P} = (p, I(p)) \in \mathcal{L} \times \mathcal{V}$

**Nachbarschaft:** eine reflexive und symmetrische Relation  $\nu \subseteq \mathcal{L} \times \mathcal{L}$

$$\text{Nachbarschaft von } p \text{ ist } \nu(p) = \{q \in \mathcal{L} | p\nu q\}$$

**Ähnlichkeit:** eine reflexive und symmetrische Relation  $\sigma \subseteq \mathcal{V} \times \mathcal{V}$

**Direkte Nachbarn:**  $p, q \in \mathcal{L} \wedge |p_1 - q_1| \leq 1 \wedge |p_2 - q_2| \leq 1$

**Zusammenhang:** Eine Menge  $M \in \mathcal{L}$  ist zusammenhängend, gdw.

$$\forall p, q \in M : \exists n \in \mathbb{N} : \exists p_0, \dots, p_n \in M : p = p_0 \wedge q = p_n \wedge \forall i : 0 \leq i < n \Rightarrow (p_i, p_{i+1}) \in \nu$$

**Segment:**  $\sigma$ -ähnliche, zusammenhängende Teilmenge eines Bildes

**Segmentcluster:** Segment oder Teilmenge eines Bildes, die sich aus der Vereinigung von benachbarten Segmenten mit einer Ähnlichkeit  $\sigma'$  ergibt

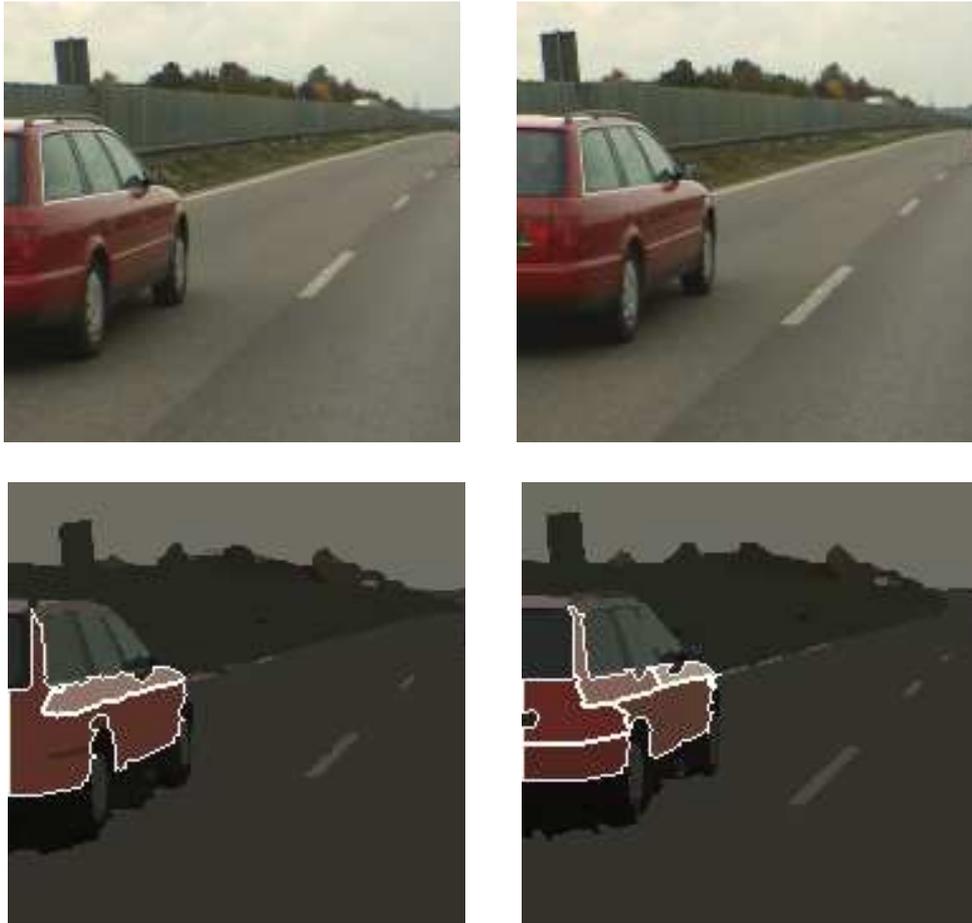


Abbildung 2: Beispiel für einen  $n:m$ -Match (hier 2:5)  
 Ein Teil der Fahrzeugkarosserie zerfällt in zwei aufeinanderfolgenden Bildern zum Zeitpunkt  $t-1$  (links) in 2 Segmente und zum Zeitpunkt  $t$  (rechts) in 5 Segmente. Die Bilder sind oben im Original und unten als Regionenbild dargestellt, wobei die Segmente der korrespondierenden Cluster durch einen weißen Rand hervorgehoben sind und der Bildhintergrund abgedunkelt ist.

## Literatur

- [HKR97] Bernd Heisele, U. Kreßel, and Werner Ritter. Tracking non-rigid, moving objects based on color cluster flow. *IEEE Conference on Computer Vision and Pattern Recognition, San Juan*, pages 253–257, 1997.
- [NM79] M. Nagao and T. Matsuyama. Edge preserving smoothing. In *Computer Graphics and Image Processing*, volume 9, pages 394–407, 1979.
- [PH86] M. Pietikäinen and D. Harwood. Segmentation of color images using edge-preserving filters. In *Advances in Image Processing and Pattern Recognition*, pages 94–99, 1986.
- [Reh98] Volker Rehrmann. Object oriented motion estimation in color image sequences. In *ECCV 1998, Vol. I, LNCS 1406*, pages 704–719, 1998.
- [Ros03] Mark Ross. Evaluation and improvement of region-based motion segmentation. Eingereicht zum 8th Workshop on Vision, Modeling, and Visualization 2003, 2003.
- [RP98] Volker Rehrmann and Lutz Priese. Fast and robust segmentation of natural color scenes. In *Proc. of 3rd Asian Conf. on Computer Vision, Special Session on Advances in Color Vision*, volume I, pages 704–719. Springer Verlag, 1998.
- [Sch00] Bernt Schiele. Model-free tracking of cars and people based on color regions. In *Proceedings 1st IEEE Int. Workshop on PETS, Grenoble, France, March 2000*.