

Automatische Schwimmerverfolgung und Segmentierung

Thomas Greif, Rainer Lienhart

Angaben zur Veröffentlichung / Publication details:

Greif, Thomas, and Rainer Lienhart. 2009. "Automatische Schwimmerverfolgung und Segmentierung." In Gegenstand und Anwendungsfelder der Sportinformatik: Symposium der dvs-Sektion Sportinformatik vom 22.-24. Mai 2008 in Augsburg, edited by Martin Lames, Claudia Augste, Christoph Dreckmann, Karsten Görtsdorf, and Malte Siegle, 183-88. Hamburg: Czwalina.

Nutzungsbedingungen / Terms of use:

licgercopyright

Dieses Dokument wird unter folgenden Bedingungen zur Verfügung gestellt: / This document is made available under these conditions:

Deutsches Urheberrecht

Weitere Informationen finden Sie unter: / For more information see:

<https://www.uni-augsburg.de/de/organisation/bibliothek/publizieren-zitieren-archivieren/publiz/>



Automatische Schwimmerverfolgung und Segmentierung in Videos

Einleitung

Die automatische Auswertung von Schwimmvideos, wie beispielsweise das automatische Messen der Schlagfrequenz eines Schwimmers bei verschiedenen Schwimmstilen, macht es i.d.R. erforderlich, dass sich dieser Schwimmer in jedem Videobild an nahezu der gleichen Position in einer (annähernd) genormten Größe befindet. Verwendet man z.B. mit einer tragbaren Videokamera aufgenommene Schwimmvideos, so stellt man fest, dass dieses selten zutrifft. Um also eine normalisierte Aufnahme eines Schwimmers zu extrahieren, ist es notwendig, diesen automatisch verfolgen und dessen Größe bestimmen zu können, damit die verfolgte Region später entnommen und normalisiert werden kann.

Gängige Verfahren zur Vordergrund-/Hintergrund-Segmentierung sowie zur Verfolgung von Objekten scheitern aber daran, dass der Schwimmer immer wieder unterschiedlich stark von Wasser überdeckt ist.

Wir stellen in dieser Arbeit ein Verfahren vor, das diese Probleme kompensiert und in der Lage ist, Schwimmer in herkömmlichen Schwimmvideos automatisch zu verfolgen und deren ungefähre Größe zu bestimmen. Hierzu erweitern wir den populären Mean Shift-Tracker (Comaniciu, 2000) um eine zusätzliche Komponente, die es ihm erlaubt, lediglich die hautfarbenen Bereiche der zu verfolgenden Region zu betrachten und so Schwimmer mit großer Zuverlässigkeit zu verfolgen.

Der Mean Shift-Algorithmus

Die Grundlage für die Schwimmerverfolgung bildet der weit bekannte *Mean Shift-Algorithmus* (Cheng, 1995). Dieser stellt ein Verfahren zum Vergleich von gewichteten Wahrscheinlichkeitsdichten zweier Kernelregionen im d -dimensionalen Raum R^d dar. Hierbei wird genau die Region im Raum gesucht, deren Verteilung der Verteilung einer gewissen Zielregion am ähnlichsten ist. Dazu wird iterativ ein *Mean Shift-Vektor* zwischen diesen Verteilungen berechnet, in dessen Richtung der Mittelpunkt der aktuellen Kernel-Region verschoben wird. Dies geschieht so lange, bis die maximale Ähnlichkeit bzw. die kleinstmögliche Distanz zwischen den Verteilungen der Regionen erreicht ist.

Formal kann die erwartete Kerneldichte eines Punktes x des Kernels $K(x)$ mit Größe h für eine Menge von Datenpunkten $\{x_i\}_{i=1..n}$ wie folgt definiert werden (Comaniciu, 2000)

$$\hat{f}_K(x) = \frac{1}{nh^d} \sum_{i=1}^n k \left(\left\| \frac{x - x_i}{h} \right\|^2 \right)$$

wobei der Kernel als Funktion $k : [0, \infty) \rightarrow \mathbb{R}$ mit $K(x) = k(\|x\|^2)$ definiert werden kann. Der Mean Shift-Vektor ist dann gegeben durch

$$M_{h,G}(x) = \frac{\sum_{i=1}^n x_i g \left(\left\| \frac{x - x_i}{h} \right\|^2 \right)}{\sum_{i=1}^n g \left(\left\| \frac{x - x_i}{h} \right\|^2 \right)} - x$$

wobei $g(x) = -k'(x)$ die abgeleitete Kernelfunktion beschreibt. Eine detaillierte Herleitung findet sich in (Comaniciu, 2000).

Zur Definition des Ähnlichkeitsmaßes zweier solcher Verteilungen kann die statistische Eigenschaft ausgenutzt werden, dass zwei Wahrscheinlichkeitsdichten p und q umso ähnlicher sind, je weniger sie getrennt sind. Folglich kann die Distanzminimierung auch als Maximierung des Bayes-Fehlers der Verteilungen betrachtet werden und es zeigt sich, dass dieser durch den Bhattacharyya-Koeffizienten (Kailath, 1967) approximiert werden kann, dessen generelle Form beschrieben ist durch (Comaniciu, 2000):

$$\rho(y) \equiv \rho[p(y), q] = \int \sqrt{p_z(y)q_z} dz$$

Es gilt die Annahme, dass das Merkmal z , das beispielsweise die Farbe oder Textur beschreibt, durch die Wahrscheinlichkeitsfunktion q_z in der Zielregion repräsentiert wird und $p_z(y)$ dessen Verteilung in einer Zielkandidaten-Region, zentriert um den Punkt y , darstellt.

Das auf diesem Ausdruck basierende Distanzmaß, das es zu minimieren gilt, kann also wie folgt definiert werden:

$$d(y) = \sqrt{1 - \rho[p(y), q]}$$

Der Mean Shift-Algorithmus iteriert so lange, bis dieser Ausdruck lokal minimal ist und die ähnlichste Verteilung gefunden wurde.

Methoden

Die Schwierigkeit bei der Verfolgung eines Schwimmers besteht darin, dass die zu verfolgende Region regelmäßig zum Großteil von Wasser überdeckt wird und damit dieser ein sehr wasserähnliches Aussehen verleiht. Die Pixel des Schwimmers können daher leicht mit Wasser verwechselt werden. Populäre Objektverfolgungsverfahren wie der verwendete Mean Shift-Algorithmus scheitern an genau dieser Tatsache, da das charakteristische Merkmal einer solchen Region Wasser und nicht wie gewünscht der Schwimmer ist.

Der hier vorgestellte Ansatz erweitert den Mean Shift-Verfolgungsalgorithmus um eine vorab trainierte *Hautfarbton-Verteilung*, die die Verteilung der Hautfarbtöne von Schwimmern im Wasser repräsentiert. Der Verfolgungsalgorithmus wird dahingehend abgeändert, dass zur Berechnung der Position des Schwimmers in jedem Videobild nur diejenigen Pixel betrachtet werden, die mit einer ausreichender Wahrscheinlichkeit hautfarben sind, d.h. mit gewisser Häufigkeit in der Hautfarbton-Verteilung auftauchen.

Vorverarbeitung

Für die Erstellung der Hautfarbtonverteilung werden eine große Zahl manuell segmentierter Trainingsbilder verwendet, die lediglich hautfarbene Bereiche von Schwimmern enthalten und das große Spektrum der bei Schwimmern im Wasser vorkommenden Hautfarbtöne widerspiegeln. Aus diesen Trainingsbildern wird ein *RGB-Farbhistogramm* erstellt, welches die relativen Häufigkeiten enthält, mit der bestimmte RGB-Farbwerte auftreten. Es ist repräsentativ für die Verteilung der Hautfarbtöne von Schwimmern. Zur Erstellung des Histogramms wird jeder Farbkanal in 128 Werte diskretisiert, was zu 128^3 möglichen Farbwerten führt.

Schwimmerverfolgung

Die automatische Schwimmerverfolgung basiert auf dem in Abschnitt 2 vorgestellten Mean Shift-Algorithmus sowie auf dem globalen Hautfarbton-Histogramm aus Abschnitt 3.1. Zum Initialisieren des Algorithmus wird zunächst eine zu verfolgende Region, d.h. ein Schwimmer ausgewählt, die als *Zielregion* bezeichnet wird. Für jedes Videobild wird dann eine binäre *Maske* erstellt. Diese gibt für jedes Pixel an, ob es mit ausreichender Wahrscheinlichkeit einen Hautfarbton beschreibt und später von dem Algorithmus betrachtet werden muss. Dies kann festgestellt werden, indem der dem Pixel entsprechende Farbeintrag in der Hautfarbton-Verteilung ermittelt und geprüft wird, ob die relative Häufigkeit für diese Farbton über einem festgelegten Schwellwert liegt. Trifft dies zu, enthält das Pixel an dieser Position in der Maske die Farbe weiß, andernfalls schwarz. Abbildung 1 zeigt das Eingabevideobild mit der umrandeten und zu verfolgenden Region sowie die für dieses Bild erstellte Maske anhand des Hautfarbton-Histogramms.

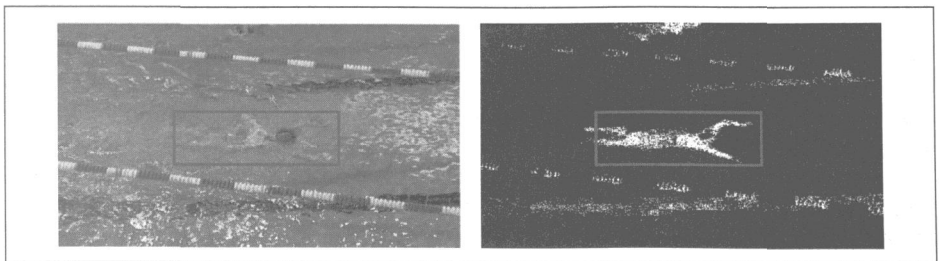


Abb. 1. Das Eingabe-Videobild mit selektierter Region (links) sowie die durch das Hautfarbton-Histogramm erstellte Maske (rechts).

Bevor der Mean Shift-Tracker angewendet wird, wird jedes RGB-Videobild in den HSV-Farbraum konvertiert. Dieser ist dafür bekannt, dass er den Farbton (*Hue*) von der Sättigung (*Saturation*) und der Helligkeit (*Value*) trennt, was von Vorteil ist, da für die Verfolgung lediglich der Farbton relevant ist.

Die zu Beginn initialisierte Region stellt die Verteilung q über alle Pixel $\{x_i^*\}_{i=1..n}$ der um O zentrierten Zielregion dar. Es ist zu beachten, dass nur diejenigen Pixel x_i^* betrachtet werden, die mit einem weißen Farbwert in der Maske auftauchen, also mit ausreichender Wahrscheinlichkeit hautfarben sind.

Eine Funktion $b : R^2 \rightarrow \{1..m\}$ gibt die Abbildung der Pixelposition auf einen Farbwert $b(x_i^*)$ des H-Histogramms mit m Quantisierungsintervallen an, durch welches die Verteilung repräsentiert wird. Diese mit der Kernelfunktion gewichtete Verteilung kann nun wie folgt definiert werden (Comanicu, 2000)

$$q_u = C \sum_{i=1}^n k\left(\|x_i^*\|^2\right) \delta(b(x_i^*) - u)$$

wobei δ die Delta-Funktion beschreibt und $u = 1..m$ den quantisierten Farbton (hue). Der Normalisierungsfaktor C wird so gewählt, dass die Summe über alle Bins der Verteilung 1 ergibt: $\sum_{u=1}^m q_u = 1$.

Gleichermaßen ist die Verteilung p jeder *Zielkandidaten-Region* definiert, die alle Pixel $\{x_i\}_{i=1..n}$ enthält und zentriert um einen bestimmten Punkt y im Bild ist

$$p_u(y) = C_h \sum_{i=1}^{n_h} k\left(\left\|\frac{y - x_i}{h}\right\|^2\right) \delta(b(x_i) - u)$$

mit Normalisierungsfaktor C_h . Wieder gilt, dass stets nur von der Maske selektierte Pixel in die Berechnung der Verteilung eingehen, so dass der Algorithmus nur auf Pixeln arbeitet, für die angenommen wird, dass sie hautfarben sind. Der verwendete Kernel $k(x)$ gewichtet die jeweilige Pixelposition durch ihren Abstand vom Mittelpunkt der Region. Dies ist deshalb sinnvoll, da erwartungsgemäß Positionen mit erhöhtem Abstand vom Ursprung des zu verfolgenden Objektes oft lediglich Hintergrund oder überdeckte Regionen beschreiben und deshalb weniger stark gewichtet werden dürfen.

Der verwendete Verfolgungs-Algorithmus vergleicht in jedem Videobild iterativ die Verteilung p der aktuellen Zielkandidaten-Region mit der Zielverteilung q , die die Verteilung der ursprünglich den Schwimmer enthaltenden Region darstellt. Hierbei wird die Position der Region im aktuellen Videobild mit der zuletzt geschätzten Position des Vorgänger-Videobildes initialisiert, deren Mittelpunkt in Folgendem mit \hat{y}_0 bezeichnet ist. Ausgehend davon, wird die Region in Richtung des Mean Shift-Vektors solange verschoben, bis das Abbruchkriterium erfüllt ist und die Region, deren Verteilung die kleinste Distanz zur Zielverteilung hat, gefunden wurde. Wir gehen davon aus, dass diese Region wieder den Schwimmer enthält. Der Ablauf des Verfahrens ist in Tabelle 1 beschrieben:

Tab. 1. Anwendung des Mean Shift-Trackers.

1.	<p>Initialisiere die Position der Zielkandidaten-Region des aktuellen Videobildes mit \hat{y}_0 und berechne die Verteilung $\{p(\hat{y}_0)\}_{u=1,\dots,m}$ aller von der Maske selektierten Pixel der Region, sowie deren Bhattacharyya-Koeffizient</p> $\rho[p(\hat{y}_0), q] = \sum_{u=1}^m \sqrt{p_u(\hat{y}_0)q_u}$
2.	<p>Berechne die neue Position \hat{y}_1 anhand des Mean Shift-Vektors</p> $\hat{y}_1 = \frac{\sum_{i=1}^{n_a} x_i w_i g\left(\left\ \frac{\hat{y}_0 - x_i}{h}\right\ ^2\right)}{\sum_{i=1}^{n_a} w_i g\left(\left\ \frac{\hat{y}_0 - x_i}{h}\right\ ^2\right)} \quad \text{mit } w_i = \sum_{u=1}^m \delta(b(x_i) - u) \sqrt{\frac{q_u}{p_u(\hat{y}_0)}}$
3.	<p>Berechne die neue Verteilung $\{p_u(\hat{y}_1)\}_{u=1,\dots,m}$ sowie den daraus resultierenden Bhattacharyya-Koeffizient $\rho[p(\hat{y}_1), q]$</p>
4.	<p>Solange $\rho[p(\hat{y}_1), q] < \rho[p(\hat{y}_0), q]$ setze $\hat{y}_1 \leftarrow \frac{1}{2}(\hat{y}_0 + \hat{y}_1)$</p>
5.	<p>Falls $\ \hat{y}_1 - \hat{y}_0\ < \varepsilon$ beende, andernfalls setze $\hat{y}_0 = \hat{y}_1$ und gehe zu Schritt 1.</p>

Die im Mean Shift-Vektor vorkommenden Gewichte w_i gewichten den Anteil, den der aktuelle Farbwert (hue) des Histogramms im Bezug auf die Zielregion hat. Veranschaulicht bedeutet das, dass Farben, die häufig in der Zielverteilung q auftauchen bei der Berechnung des Mean Shift-Vektors höher gewichtet sind als solche, die weniger oft vorhanden sind. Will man beispielsweise ein rotes Objekt verfolgen, ist der Rotanteil in der Zielregion sehr hoch. Enthält der Zielkandidat jedoch viele Blautöne, so müssen diese weniger stark gewichtet werden, da eine Verschiebung in diese Richtung die Distanz der Verteilungen offensichtlich nicht verringern würde. Das Verfahren entspricht dem des ursprünglichen Mean Shift-Trackers (Comaniciu, 2000), mit dem Zusatz, dass nur über jene Pixel eine Wahrscheinlichkeitsverteilung berechnet wird, die Hautfarbe beschreiben.

Ergebnisse

Das Verfahren wurde mit herkömmlichen Schwimmvideos evaluiert, wobei sich die Kamera mit den Schwimmern bewegte, um sie im Bildmittelpunkt zu halten. Obwohl die Ergebnisse rein qualitativer Natur sind, können Aussagen darüber gemacht werden, wie gut der Algorithmus in der Lage ist, Schwimmer zu verfolgen. Wird der Schwimmer zu Beginn korrekt selektiert, so ist es mit dem hier vorgestellten Verfahren möglich, diesen für eine komplette Bahn zu verfolgen. Weiterhin zeigt sich, dass der Schwimmstil hierfür nicht entscheidend ist, da die Ergebnisse stets zufriedenstellend sind. Abbildung 2 zeigt Beispiele für verfolgte Schwimmer.

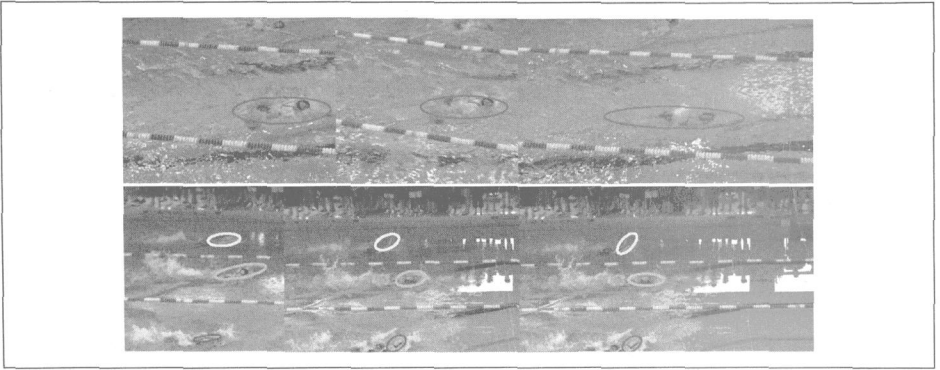


Abb. 2. Beispielsequenzen der automatischen Schwimmerverfolgung verschiedener Schwimmstile.

Problematisch dagegen sind Wenden, da dort der Schwimmer kaum sichtbar ist und der Hintergrund meist viel Hautfarbe durch herumstehende Personen enthält. Das Verfahren sollte daher um eine spezielle Wendenerkennung mit Tauchphase ergänzt werden, da der Schwimmer bisher mit der Wende verloren geht

Diskussion

Wir haben eine Erweiterung des ursprünglichen MeanShift-Trackers vorgestellt, die durch das Erstellen einer für Hautfarbe repräsentativen Farbverteilung den ansonsten überaus hohen Blauanteil der vom Algorithmus verwendeten Hue-Verteilung herausrechnet und so ermöglicht, Schwimmer, die einmal ausgewählt wurden, automatisch zu verfolgen und deren ungefähre Größe zu schätzen. Erweiterungen des Verfahrens sind denkbar, die den Hintergrund wie Beckenrand und hereinragende Gegenstände eliminieren, um eine noch robustere und zuverlässigere Schwimmerverfolgung zu erreichen.

Literatur

- Cheng, Y. (1995). Mean Shift, Mode Seeking, and Clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17.
- Comaniciu, D (2002). Mean Shift: A Robust Approach Toward Feature Space Analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24.
- Comaniciu, D. & Ramesh, V. & Meer, P. (2000). Real-Time Tracking of Non-Rigid Objects using Mean Shift. *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, 142-151.
- Kailath, T. (1967). The Divergence and Bhattacharyya Distance Measures in Signal Selection. *IEEE Trans. Commun. Techn.*, 52-60.