Objektretrieval und Resultatpräsentation in der Videodatenbank CAIRO/*VE*

Stefan Geisler

geisler@informatik.tu-clausthal.de

Abstract: Mit der steigenden Anzahl digitaler Videos wächst auch der Bedarf an Verfahren und Methoden zur Speicherung und zum effizienten Wiederfinden von Videos. Die Videodatenbank CAIRO/*VE* ermöglicht die dynamische inhaltsbasierte Suche nach Objekten direkt auf dem MPEG-Videomaterial ohne Stichwortliste anhand eines Beispielbildes. Effiziente sequentielle und parallele Suchalgorithmen sorgen trotz hohen Rechenaufwands für akzeptable Antwortzeiten und hohe Ergebnisqualität. Eine benutzerfreundliche Ergebnispräsentation mit *region-of-interest*-kodierten Videos ermöglicht es, einen schnellen Überblick über die Ergebnismenge zu erhalten.

1 Einleitung

Die Anzahl digitaler Videos nimmt täglich in enormem Maße zu. Neben dem professionellen Bereich hält in den letzten Jahren auch in Privathaushalten die digitale Videotechnik Einzug. Obwohl im Forschungsbereich eine Reihe von Videodatenbanken entwickelt wurden, genannt seien hier beispielhaft VIDEOQ [CCM+97] und die VIRAGE VIDEO ENGINE [HGH+97], stellt die strukturierte Speicherung und das einfache und schnelle Wiederfinden der abgelegten Daten immer noch eine Herausforderung dar.

Die hier vorgestellte Videodatenbank CAIRO/*VE* ist eine Erweiterung der von KAO entwickelten Bilddatenbank CAIRO (C*luster* A*rchitecture for* I*mage* R*etrieval and* O*rganization*) [KS01]. CAIRO ermöglicht die Suche nach beliebigen Objekten in unterschiedlichsten Bildern mit Hilfe eines Anfragebildes. Im Gegensatz zu anderen Bilddatenbanken ist es hierfür notwendig, die zum Vergleich benötigten Merkmalsvektoren nach Übermittlung der Anfrage dynamisch zu berechnen, da *a-priori* berechnete Merkmale das Bild als Gesamtheit beschreiben. Zwar existieren effiziente Indexstrukturen, die eine Objektsuche auf statischen Merkmalen erlauben [Fa96], eine gute Bildsegmentierung wird jedoch vorausgesetzt. Anwendung finden diese Verfahren daher nur in Datenbanken mit eingeschränkten Bilddomänen, etwa medizinischen Röntgenaufnahmen bestimmter Körperteile.

Bei der dynamischen Suche wird das Anfragebild an allen möglichen Positionen über jedes Bild der Datenbank gelegt und mit dem darunter liegenden Bereich verglichen (*template matching*). Eine vorherige Segmentierung ist nicht erforderlich, das Verfahren somit flexibel einsetzbar. Die notwendige Performance wird durch einen Cluster erzielt.

Für die V*ideo* E*xtensions* von CAIRO/*VE* müssen wegen der erheblich größeren Vide-

odateien zusätzliche Optimierungsschritte durchgeführt werden. Hierzu werden spezielle Eigenschaften der MPEG-Kodierung ausgenutzt, sowie angepasste Strategien zu Parallelisierung angewandt. Außerdem ist eine neue Form der Ergebnispräsentation notwendig, um den dynamischen Inhalt von Videos darzustellen.

2 Der Aufbau von CAIRO/*VE*

Auf den Knoten eines Dualprozessor-Clusters werden disjunkte Teilmengen der Videos gespeichert. Dieser Parallelrechneraufbau hat sich durch experimentelle Untersuchungen und Simulationsergebnisse als hervorragend geeignet für die schnelle dynamische Suche in großen Videomengen erwiesen und erzielt fast optimale Beschleunigung [Ge04].

Der Benutzer sendet über das Internet ein Anfragebild und weitere Suchparameter an die Datenbank. Diese werden vom Master-Knoten an die einzelnen Slave-Knoten geschickt, auf denen solange nacheinander pro CPU ein Suchprozess für ein bisher nicht betrachtetes Video gestartet wird, bis alle lokalen Dateien bearbeitet wurden. Sobald am Ende der Suche durch die unterschiedliche Länge der Videos nur noch ein Prozessor ausgelastet ist, wird für jede CPU ein Thread erzeugt, der jeweils die Hälfte der verbliebenen Frames durchsucht. Hat ein gesamter Knoten seine Arbeit beendet, fordert er von einem überlasteten Knoten ein Video an und bearbeitet dieses. Durch die verwendete *largest-task-first*-Strategie werden nur die kleineren Videodateien über das Netz gesendet.

Auf dem Master-Knoten wird abschließend aus den Einzelergebnissen eine Gesamtliste mit ähnlichen Videoszenen erzeugt. Die Ergebnissequenzen werden in einem nächsten Schritt aus den Gesamtvideos extrahiert und zur Darstellung nachbearbeitet. Danach werden die Ergebnisse vom Master-Knoten an den Anfrageclient übermittelt. Durch die Auswahl eines der Ergebnisvideos kann der Benutzer dann das Originalvideo anfordern.

3 Effiziente dynamische Suche in MPEG-Videos

Der Vergleich des Anfragebildes an allen Positionen in jedem Frame ist nicht ohne weitere Optimierungsschritte sinnvoll durchführbar. Daher wurden zwei Techniken zur Beschleunigung der Suche entwickelt, die die Eigenschaften von MPEG-1/2 ausnutzen.

Reduzierung der Frameanzahl: Die Ähnlichkeit benachbarter Bilder einer Szene erlaubt die Beschränkung der Suche auf zwei Frames pro Sekunde, ohne kleine Änderungen bewegter Objekte zu vernachlässigen. Dies entspricht gleichzeitig der I-Frame-Rate üblicher Encoder und ermöglicht eine effiziente Dekodierung, da diese im Gegensatz zu den anderen Bildtypen ohne Kenntnis benachbarter Frames dargestellt werden können.

Suche auf komprimierten Videodaten: Jedes I-Frame wird bei der Kodierung in Blöcke der Größe 8×8 Pixel unterteilt. Anstelle der Pixelwerte werden die Koeffizienten der diskreten Kosinustransformtion (DCT) nach weiterer Komprimierung gespeichert. Bei der Dekodierung ist die inverse DCT der aufwändigste Schritt, kann aber durch einen Ver-

gleich im Frequenzraum umgangen werden (siehe [CKT00] für JPEG-Bilder). Hierfür werden die DCT-Koeffizienten des Anfragebildes benötigt, die einmalig für die gesamte Suche berechnet werden müssen. Eine weitere Vereinfachung besteht darin, nur die ersten Koeffizienten, die den Durchschnittswert des Blockes angeben, zu vergleichen.

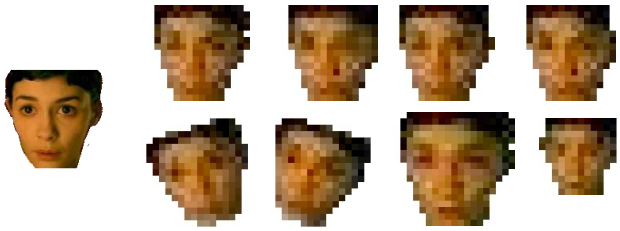

Abbildung 1: Templates zu gegebenem Originalbild: Oben: DC-Bilder mit erstem Block beginnend an Positionen (0, 0), (4, 0), (0, 4), (4, 4). Unten: DC-Bilder des gedrehten bzw. skalierten Originals.

Um eine bessere Invarianz gegen Skalierung und Rotation zu erzielen, wird nicht nur mit dem Originaltemplate gesucht, sondern auch mit folgenden Modifikationen des Anfragebildes: Skalierung mit dem Faktor 0,8, und 1,2, Rotation um $\pm 0,3$, sowie Verwendung von Subpixel-DC-Bildern, die erzeugt werden, indem die Blockgrenze in X- und/oder Y-Richtung um vier Pixel verschoben wird (Abbildung 1).

Die Ähnlichkeit des Templates T an Position x_0, y_0 im Frame F wird durch die Summe des gewichteten euklidischen Abstand der einzelnen Farbkanalwerte bestimmt:

$$d(T, F, x_0, y_0) = \sqrt{\sum_{x,y,c} a(x,y) w_c \left(T(x,y,c) - F(x_0+x, y_0+y, c)\right)^2}$$

mit x, y, der Pixelposition im Template; $c \in \{Y, Cb, Cr\}$, dem MPEG-Farbkanal; $a(x,y) \in [0,1]$, dem Alpha-Kanal im Template zur Suche nach nicht rechteckigen Formen und $w_c \in [0,1]$, der Gewichtung für jeden Farbkanal. Mit $w_Y = 0$ ist somit eine helligkeitsinvariante Suche möglich.

Die Ähnlichkeit eines Frames wird durch das Minimum über alle Differenzen aller modifizierten Templates an den unterschiedlichen Positionen definiert. Für jede Szene wiederum wird der Frame mit der größten Ähnlichkeit als Repräsentant gewählt.

Geschwindigkeit und Qualität der Suche

Für die so optimierte Suche nach einem 10.000 Pixel großen Template in einem 15 Minuten langen MPEG-1-Video werden auf einem 2,2 GHz Xeon 74 Sekunden benötigt.

Die Ergebnisgenauigkeit wurde mit einer Testvideomenge von 15 Stunden verschiedener Spielfilmsequenzen und Fernsehmitschnitte, u.a. von Nachrichten- und Sportsendungen, bestimmt. Trotz der verlustreichen Optimierung wurde bei der Objektsuche eine Genauigkeit von 60% in einer Ergebnismenge der Größe 20 gemessen. Dabei wurden alle Farbkanäle gleich stark gewichtet ($w_c = 1 \; \forall c$). Ein Beispiel zeigt Abbildung 2.

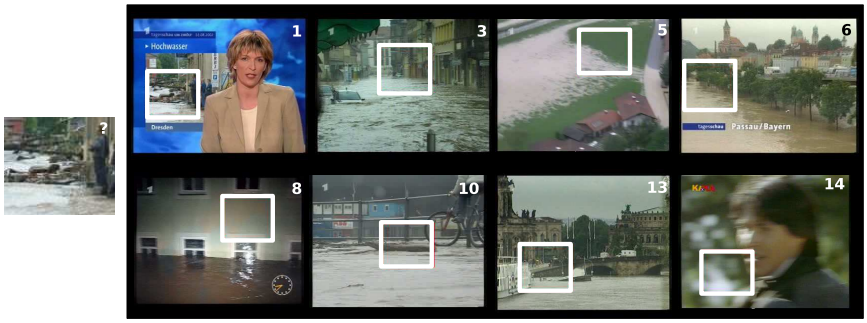

Abbildung 2: Anfragebild und Beispiele aus der Ergebnismenge. Hier nicht gezeigte Treffer sind zu anderen abgebildeten Frames sehr ähnlich. Position 14 ist ein Fehltreffer.

4 Ergebnisvideos mit *region of interest* Videokodierung

Das gleichzeitige Abspielen aller Ergebnisvideos würde den Benutzer überfordern, da sich dieser nicht auf eine Vielzahl Videos gleichzeitig konzentrieren kann. Daher werden häufig nur ein oder mehrere Schlüsselbilder angezeigt. Eine repräsentative Bildauswahl ist jedoch schwierig und lässt insbesondere bei längeren Szenen oder bei viel Kamera- oder Objektbewegung nur selten einen kompletten Überblick über die Szene zu. Beispiele alternativer Darstellungsformen sind die hierarchische Anordnung (z.B. [ZLSW97]) oder Collagen von Schlüsselbildern (z.B. [UFGB99]), sowie aus einzelnen Frames zusammengefügte Panoramen mit zusätzlichen Bewegungsvektoren an den Objekten (z.B. [TAT97]).

Im Folgenden soll ein neuer Ansatz vorgestellt werden, der keine statische Ansicht erzeugt, sondern eine speziell aufbereitete Version der Videos darstellt. Das Ziel ist es, die Aufmerksamkeit des Benutzers schnell auf die interessanten Stellen zu lenken, ohne ihm die Möglichkeit zu nehmen, die räumliche und zeitliche Umgebung wahrzunehmen.

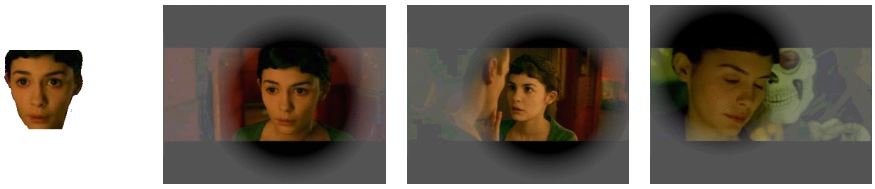

Abbildung 3: Bild von drei aufbereiteten Ergebnisvideos zu einem zufälligen Zeitpunkt.

Zunächst wird die genaue Position des Suchbildes in jedem Frame des Ergebnisvideos bestimmt, ausgehend vom Ergebnis der schnellen Suche. Ein umschließender Kreis wird als *region of interest* (ROI) definiert. Deren optische Hervorhebung wird dadurch erreicht, dass der Farbwert aller Pixel außerhalb der ROI der Bildschirmhintergrundfarbe angenähert wird. Zusätzlich wird zur Speicherersparnis eine Auflösungsreduktion und stärkere Quantisierung des Farbraums vorgenommen. Einen Eindruck gibt Abbildung 3.

Zur zeitlichen Hervorhebung wird die Abspielgeschwindigkeit so variiert, dass Frames mit hoher Übereinstimmung länger angezeigt werden, als Frames mit geringer Ähnlichkeit.

5 Zusammenfassung und Ausblick

Mit CAIRO/*VE* wurde eine Videodatenbank mit dynamischer inhaltsbasierter Suche nach Objekten vorgestellt. Der notwendige Rechenaufwand wird durch eine effiziente Suche auf dem MPEG-kodierten Videomaterial verringert. Eine parallele Suche auf einem Cluster ermöglicht die Bearbeitung großer Datenbestände. Trotz der verlustbehafteten Optimierung wird eine hohe Genauigkeit bei der Suche erzielt. Für die Ergebnisdarstellung wird durch ROI-Videokodierung die Aufmerksamkeit des Benutzers schnell auf die interessanten Stellen des Videos gelenkt, ohne Bewegungsinformationen zu unterdrücken.

In zukünftigen Arbeiten soll die Objektbewegung als weiteres Anfragekriterium integriert, sowie die ROI für nicht rechteckige Objekte exakter definiert werden.

Literatur

[CCM+97] Chang, S.-F., Chen, W., Meng, H. J., Sundaram, H., und Zhong, D.: VideoQ: An automated content based video search system using visual cues. In: *Proc. ACM Multimedia '97*. S. 313–324. 1997.

[CKT00] Chang, R.-F., Kuo, W.-J., und Tsai, H.-C.: Image retrieval on uncompressed and compressed domains. In: *Proc. of intern. Conf. on Image Processing*. S. II–546–549. 2000.

[Fa96] Faloutsos, C.: *Searching Multimedia Databases by Content*. Kluwer Academic Press. 1996.

[Ge04] Geisler, S.: Efficient parallel search in video databases with dynamic feature extraction. In: *Proc. Parallel Computing (ParCo 2003)*. To be published 2004.

[HGH+97] Hampapur, A., Gupta, A., Horowitz, B., Shu, C.-F., Fuller, C., Bach, J., Gorkani, M., und Jain, R.: Virage video engine. In: *Proc. SPIE vol. 3022, Storage and Retrieval for Image and Video Databases*. S. 188–198. 1997.

[KS01] Kao, O. und Stapel, S.: Case study: Cairo, a distributed image retrieval system for cluster architectures. *Distributed Multimedia Databases: Techniques and Applications*. S. 291–303. 2001.

[TAT97] Taniguchi, Y., Akutsu, A., und Tonomura, Y.: Panoramaexcerpts: extracting and packing panoramas for video browsing. In: *Proc. ACM Multimedia '97*. S. 427–436. ACM Press. 1997.

[UFGB99] Uchihashi, S., Foote, J., Girgensohn, A., und Boreczky, J.: Video manga: generating semantically meaningful video summaries. In: *Proc. ACM Multimedia '99 (Part 1)*. S. 383–392. ACM Press. 1999.

[ZLSW97] Zhang, H., Low, C. Y., Smolier, S. W., und Wu, J.: *Video parsing, retrieval and browsing: an integrated and content-based solution*. MIT Press. 1997.