Trennung von Dozenten und Tafel in einem E-Kreide Video

Kristian Jantz, Gerald Friedland, Rául Rojas
{jantz|fland|rojas}@inf.fu-berlin.de

3. Juni 2004

Zusammenfassung

Die automatische Trennung eines Videos in Vorder- und Hintergrund ist eine anspruchsvolle Aufgabe, die in verschiedenen Bereichen Anwendung findet. Dabei wird ein vorliegendes Video analysiert und es wird versucht Vordergrundobjekte, wie zum Beispiel Personen, oder sich bewegende Fahrzeuge zu extrahieren. Die bestehenden Verfahren können dabei in zwei Klassen unterteilt werden. Es gibt einerseits Methoden, die sich auf eine statische Kamera und einen unveränderlichen Hintergrund verlassen und es gibt Verfahren, bei denen dynamische Hintergründe auftreten können. Die meisten Verfahren verfolgen einen sehr allgemeinen Ansatz und treffen nur wenige Annahmen über den Inhalt des zu segmentierenden Videos. Anwendungen finden sich in der automatischen Personen oder Gebietsüberwachung. In dieser Arbeit wird ein selbstentwickelter Ansatz zur Segmentierung eines Videos vorgeschlagen und es werden Resultate mit Ergebnissen bereits bekannter Verfahren verglichen. Es wird außerdem der Nutzen der Segmentierung für die Verbesserung des E-Kreide Systems erklärt.

1 Videos in E-Kreide

E-Kreide ist ein System, das entwickelt wurde um den Präsenzunterricht in den Schulen und Universitäten zu verbessern. Es vereinigt die Vorteile einer klassischen Kreidetafel, wie zum Beispiel eine gleichförmige, dem Mitschreiben angepasste Vortragsweise, mit denen eines modernen Teleteaching Systems WWW-EKREIDE (2004). So ist es zum Beispiel möglich in einen Vortrag, der das System als klassische Tafel verwendet, verschiedene multimediale Inhalte wie zum Beispiel Bilder oder interaktive Internet Dienste einzufügen. Das System erlaubt das einbinden lokaler als auch entfernter Ressourcen. Weiterhin ist eine handschriftliche Formelerkennung und eine Anbindung an verschiedene mathematische Systeme zum Zeichnen von Funktionen enthalten. Das System speichert automatisch alle produzierten Inhalte wie Tafelbild, Audio und auch Video und stellt diese für die, zum Vortrag zeitgleiche, oder versetzte Betrachtung im Internet bereit. Außerdem wird eine automatische PDF-Mitschrift vom Tafelinhalt erzeugt, so dass Zuhörer sich besser auf den Vortrag konzentrieren können.

Bisher spielten Videos in E-Kreide eine eher untergeordnete Rolle. Den Hauptteil der Information liefert der Tafelstrom, der Audiostrom unterstützt die Vorlesung akustisch. Der Videostrom dient nicht dazu Inhalte zu transportieren, sondern vielmehr die Stimmung und Atmosphäre der Vorlesung einzufangen. Eine Vorlesung die ohne E-Kreide Video angeschaut wird, ist inhaltlich zwar vollkommen verständlich, wirkt aber etwas unpersönlich. Bisher wurden in E-Kreide Videos der Raum oder der Dozent gefilmt. Die Auflösung und Bildrate des Videos waren dabei allerdings sehr gering, so daß Details des Dozenten verloren gingen und die Videos im allgemeinen schlechter Qualität waren. Weiterhin werden E-Kreide Videos bisher bei der Wiedergabe der Vorlesung in einem getrennten Fenster dargestellt, daher muß sich der Zuschauer auf zwei Bereiche konzentrieren, zum einen auf das Tafelbild und zum anderen auf das Videobild. In Baar (1988) schreibt Baar, daß jeder Mensch aber nur ein Zentrum der Aufmerksamkeit hat. Es besteht also die Gefahr den Zuhörer zu überfordern.

Um das vorhandene Video für E-Kreide besser zu nutzen, soll ein hochauflösendes Video des Dozenten während des Vortrags gefilmt werden. Dieses Video ist als Nahaufnahme des Dozenten

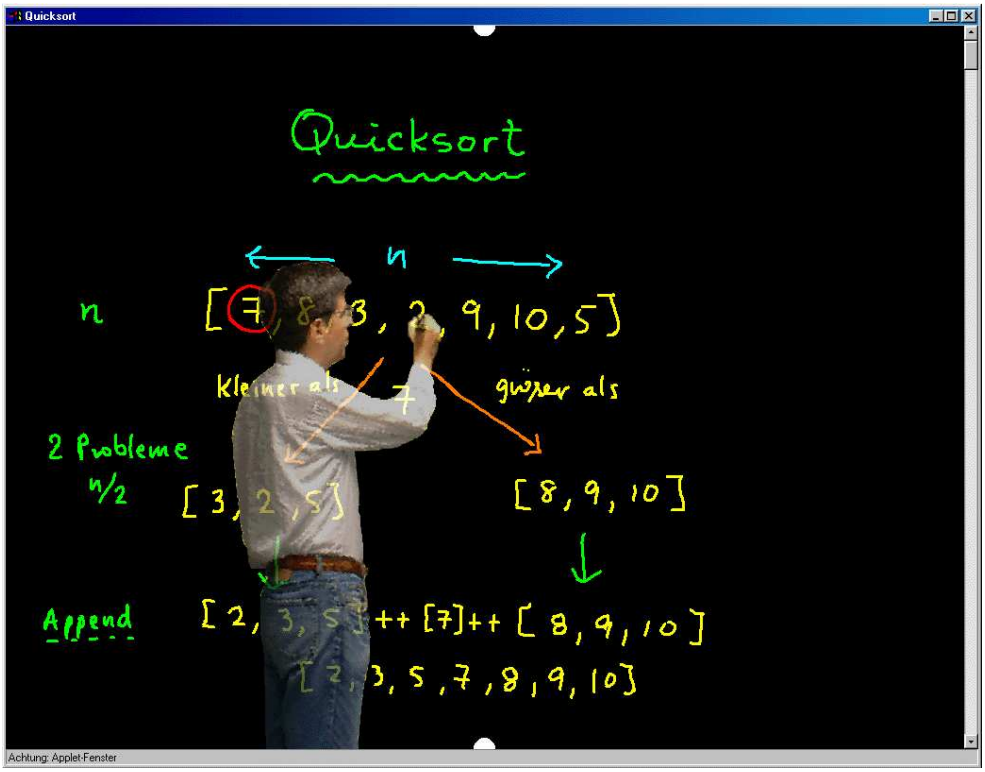


Abbildung 1: Fotomontage des angestrebten Resultats

gedacht, so daß Gesten und Mimik des Dozenten gut erkennbar sind. Auf einem solchen Video ist zwar in erster Linie der Dozent zu sehen, das Video enthält zusätzlich aber auch einen dynamischen Hintergrund, das Tafelbild. Die Aufgabe besteht nun darin den Dozenten von diesem Hintergrund zu trennen und ein neues Video zu erzeugen, in dem nur noch der Dozent zu sehen ist und der gesamte Hintergrund durch eine vereinbarte Farbe, etwa schwarz, ersetzt wird. Die Information die in dem Hintergrund des Videos gespeichert wird, ist redundant weil für ihre Übertragung bereits der Tafelstrom verwendet wird, welcher wesentlich effizienter als ein Video von Dozent und Tafel übertragen werden kann. Die Segmentierung von Dozent und Hintergrund soll zunächst offline erfolgen, das heißt, es liegt ein aufgezeichnetes Video vor, welches in beliebiger Zeit in Vorder- und Hintergrund getrennt werden kann. Es ist aber angestrebt, daß das gesamte System später in Echtzeit arbeitet. Die für die Realisierung notwendigen technischen Voraussetzungen sollen möglichst einfach gehalten sein. Für den hier gezeigten Ansatz ist nichts weiter als eine fest montierte Kamera nötig. Das gefilterte Video, welches nur noch den Dozenten enthält soll bei der Wiedergabe dann transparent oder zur besseren Übersicht halb-transparent auf das Tafelbild gelegt werden. Abbildung 1 zeigt eine Fotomontage des angestrebten Resultats.

Um das dargestellte Ergebnis zu realisieren sind außer der Segmentierung noch weitere Schritte, wie zum Beispiel das Tafelbildsynchrone Aufzeichnen und das Entzerren des Videos für die Wiedergabe notwendig. In dieser Arbeit wird zunächst nur die Segmentierung des Videos behandelt, in Jantz (2004) sind weitere Details zum Gesamtsystem nachzulesen.

2 Vergleichbare Arbeiten

Um einen Algorithmus für die Segmentierung in E-Kreide zu verwenden, muss der Algorithmus gewisse Voraussetzungen erfüllen. Der Algorithmus muss mit nur einer Kamera funktionieren, um die Hardwareanforderungen für E-Kreide Nutzer möglichst gering zu halten. Ein Ansatz der auf

einer Stereosicht der Tafel basiert ist zunächst also nicht akzeptabel. Weiterhin soll es möglich sein eine ausreichend gute Segmentierung vor einem dynamischen Hintergrund zu erzielen.

In der Literatur exisiteren verschiedene Verfahren zur automatischen Segmentierung allgemeiner Videos. In Li et al. (2003)schreiben Li et al., daß es eine Reihe von Algorithmen zum Erkennen von Vordergrundobjekten bei statischen Hintergründen gibt. Dabei wird versucht den Hintergrund durch ein geeignetes Modell zu beschreiben und dann eine Hintergrundsubtraktion durchzuführen. Ein System zum Tracken und Auffinden von Personen, welches auf einem statischen Hintergrundmodell basiert ist *W4* Haritaoglu et al. (2000); WWW-w4 (2004). Das System arbeitet nur mit Graustufenvideos. Jeder Pixel des Hintergrunds wird dabei beschrieben durch einen minimalen, einen maximalen Intensitätswert und eine maximale zeitliche Änderung dieses Wertes zwischen aufeinander folgenden Bildern. Das Hintergrundmodell wird dabei periodisch aktualisiert. Der Vordergrund lässt sich dann bestimmen, indem pixelweise mit dem Hintergrundmodell verglichen wird. Bei überschreiten eines empirisch bestimmten Abstandes des zu betrachtenden Pixels zum Hintergrundmodell, wird der Pixel als Vordergrund klassifiziert. Um Artefakte zu vermeiden finden zusätzlich noch eine Entfernung des Kamerarauschens und eine Analyse der zusammenhängenden Komponenten statt. *W4* ist ein sehr komplexes System, das für die automatische Personenüberwachung entwickelt wurde. Es ermöglicht neben der Segmentierung das Identifizieren von Körperteile wie zum Beispiel die Hände und darin gehaltene Objekte, wie etwa eine Waffe.

Ein weiterer Algorithmus für die Subtraktion eines festen Hintergrunds, der ähnlich arbeitet, ist der in Li and Leung (2002) beschriebene RBS (Robust Background Subtraction) Algorithmus. Ein entscheidender Vorteil beider Algorithmen ist, daß sie auf herkömmlichen Rechnern echtzeitfähig sind. In dem System, in dem der Algorithmus verwendet werden soll, gibt es zwar eine feste Kamera, so daß eine Hintergrundsubtraktion funktionieren könnte, der Dozent produziert aber vor der Tafel immer wieder neue Inhalte, so daß sich über die Zeit hinweg der Hintergrund ändert. Plötzliche Änderungen des Hintergrunds sind allerdings nur in wenigen Fällen zu erwarten, zum Beispiel wenn der Dozent ein Bild einfügt oder den Tafelinhalt verschiebt. In diesen Fällen ist mit herkömmlichen Segmentierungsalgorithmen nur schwer feststellbar, wo sich der Dozent befindet. In dem vorliegenden System besteht aber die Möglichkeit auf Informationen des Tafelstroms zurückzugreifen um diese Art von Problem besser zu lösen. Es existieren auch Algorithmen die versuchen das Problem des dynamischen Hintergrunds direkt zu lösen. Solche Systeme werden meist für die Freilandüberwachung verwendet.

Ein Ansatz, der auf einer Bayes Entscheidungsregeln basiert wird in Li et al. (2003) beschrieben. Dort wurde versucht, eine Entscheidungsregel für die zwei Klassen Vorder- und Hintergrund eines Bildes zu formulieren. Für die Klassifizierung wird dann jeweils der Helligkeitswert eines Pixels betrachtet. Dieser stellt die Messung dar, über die mit Hilfe der formulierten Bayes Regel eine Aussage gemacht werden soll. Da ein Pixel entweder Vordergrund oder Hintergrund ist, stellt dieser Ansatz eine geschickte Modellierung des Problems dar, die Entscheidungsregel ist hier allerdings recht kompliziert. Damit das Verfahren funktioniert, müssen die apriori Wahrscheinlichkeiten dafür, daß ein Pixel Vorder- oder Hintergrund ist gelernt und gepflegt werden. Das so abgeleitete Verfahren ist algorithmisch nicht sehr komplex, stützt sich aber elegant auf Methoden der Mathematik und Mustererkennung und ist zudem effektiv und für den hier beschriebenen Zweck sicher ebenfalls gut zu verwenden.

Ein weiterer Ansatz für die Segmentierung bei einem dynamischen Hintergrund ist Mixture of Gaussian (MoG) Stauffer and Grimson (2000). Dabei wird die Helligkeit eines Hintergrundpixels durch mehrere Gaußverteilungen beschrieben. Die Verteilungen beschreiben die für einen Hintergrundpixel erlaubten Farbänderungen über die Zeit. Auch hier wird dann für jeden Pixel mit Hilfe der Gaussverteilungen die Wahrscheinlichkeit für Voder- und Hintergrund bestimmt. MoG wurde ebenfalls für die Überwachung entwickelt, daher ist damit möglich sehr kleine sich bewegende Objekte sicher zu finden und zu verfolgen. In Lipton et al. (1998) kann man nachlesen, daß die extrahierten Konturen für die hier angestrebte Anwendung allerdings nicht scharf genug für die hier angestrebte Anwendung sind.

Für die hier angestrebte Verwendung der Segmentierung können verschiedene Annahmen über das Video gemacht werden. Das System wird später so ausgerichtet, dass das Video den Dozenten zu einem Großteil enthalten wird und keine störenden Bewegungen abseits des Dozenten auftreten.

Zusätzlich können für die Segmentierung auch Ereignisse der Tafel zur Hilfe genommen werden, siehe 4.

3 Der Ansatz

3.1 Die Idee

Der Algorithmus verlässt sich bei der Segmentierung von Vorder- und Hintergrund auf die Tatsache, das Vordergrundobjekte sich in der Regel bewegen und sich ständig leicht verändern, während Hintergrundobjekte, wie zum Beispiel eine Tafel keinen großen Änderungen unterworfen sind. Um vergleichen zu können, welche Teile des Bildes sich über einen bestimmten Zeitraum geändert haben und welche nicht, bedient sich der Algorithmus dem Prinzip der Vektorquantisierung. Bevor diese Quantisierung jedoch durchgeführt werden kann, findet eine Vorverarbeitung statt. Während der Vorverarbeitung, in der Bildelemente grob in Vorder und Hintergrund unterteilt werden, wird außerdem eine Historie der Vorder und Hintergrundelemente angelegt. Anschließend wird versucht das Resultat durch eine Komponentenanalyse, ein Nachrechnen der Kanten und einen Alterungsmechanismus für lange gleichbleibende Bildteile zu verbessern.

3.2 Vorverarbeitung

Während der Vorverarbeitung wird die Auflösung des Videostroms auf die Hälfte (160 * 120 Pixel) reduziert um den Aufwand für die Quantisierung möglichst gering zu halten, weiterhin wird das Eingabebild in 40*40 Blöcke der Größe 4*3 Pixel zerlegt. Diese Parameter beeinflußen Laufzeit und Qualität des Ergebnisses, und wurden experimentell gefunden. Für den Algorithmus stellen die definierten Blöcke im folgenden die kleinsten Einheiten dar, die zu Vorder- oder Hintergrund klassifiziert werden müssen. Es werden also nur noch Unterschiede zwischen Blöcken und nicht mehr zwischen einzelnen Pixeln betrachet. Dadurch reduziert sich die Komplexität des Problems entscheidend, allerdings wird auch die Genauigkeit bei den Konturen verringert. Da die Umrisse eines Dozenten im Verhältnis zu seiner gesamten Fläche aber wesentlich kleiner sind, können die Kanten effizient in einer höheren Genauigkeit nachgerechnet werden und es ist nicht nötig, das gesamte Bild in hoher Qualität zu berechnen.

Im nächsten Schritt wird eine Historie aufgebaut mit deren Hilfe in weiteren Durchläufen entschieden werden kann, ob ein Bildblock zum Vordergrund gehört oder nicht. Dazu wurde eine eigene Datenstruktur definiert: das Blockwörterbuch

3.3 Das Blockwörterbuch

Das Blockwörterbuch ist ein Container der Bildblöcke enthält. Es hat eine fest definierte Größe und kann so viele Blockelemente speichern wie in 10 Frames, also in einer Sekunde Video vorkommen. Die konkrete Größe hängt also von der Auflösung des Videobildes und der einzelnen Blöcke ab. Das Blockwörterbuch stellt dann folgende zwei Operationen zur Verfügung.

- addBlock(Block)
- getReferences()

Der Algorithmus 1 beschreibt das Einfügen in das Blockwörterbuch.

Ein neuer Block wird nur dann in das Blockwörterbuch eingefügt, falls kein ähnlicher Block bereits existiert (2) und noch genügend Platz vorhanden ist (7). Dabei wird die Ähnlichkeit zweier Blöcke über deren Farbwerte definiert.

Die Ähnlichkeit wird berechnet indem ein Block als 12 dimensionaler Vektor aufgefasst wird, deren Komponenten die einzelnen Pixel des Blocks sind. Jeder Pixel des einzufügenden Blocks, welcher zunächst im RGB Farbraum kodiert ist, wird in den YUV Farbraum konvertiert. Wobei die Komponenten nach dem Verhältnis 4:1:1 gewichtet werden. Dieses Vorgehen ist vernünftig,

Algorithm 1 Einfügen des Blocks bn in das Blockwörterbuch

```
1. für alle gespeicherten Blöcke b{
2.   wenn |b-bn| < Abstand dann {
3.      b.zähler++;
4.      return;
5.   }
6. }
7. wenn Dictionary voll dann ersetze Block mit wenigsten Referenzen durch bn
8. sonst füge bn ein
```

weil für die Ähnlichkeit zweier Blöcke die Y Komponente - die Luminanz, wesentlich wichtiger ist als die beiden Chrominanz Komponenten, da die Farbwerte vom menschlichen Auge erst bei ausreichender Helligkeit differenziert werden können.

Seien a, b Vektoren mit 12 Komponenten. Jede Komponente stellt in einer Integerzahl den Farbwert des dazugehörigen Pixels dar.

Der Abstand in Algorithmus 1 (2) zwischen a und b berechnet sich im Allgemeinen wie folgt:

$$d_1(a,b) = \sqrt{\sum_{i=1}^{n}(a_i - b_i)^2}$$

Für die Implementation des Algorithmus kann die Abstandsformel vereinfacht werden und die äußere Quadratwurzel weggelassen werden, da sie an der Ordnung der Elemente nichts ändert und aufwendig in der Berechnung ist. Es wurden auch Experimente mit folgender Abstandsformel durchgeführt:

$$d_2(a,b) = |\sqrt{\frac{\sum_{i=1}^{n} a_i^2}{n}} - \sqrt{\frac{\sum_{i=1}^{n} b_i^2}{n}}|$$

Die zweite Methode birgt einen entscheidenden Vorteil, man kann für alle Objekte im Blockwörterbuch den Wurzelterm einmal beim Einfügen berechnen und somit beim Vergleichen mit einem neuen Block auf bereits fertige Teilergebnisse zurückgreifen. Besonders auffällig ist dies, weil in dem Vektor b, ja bereits die Farbkonvertierung enthalten ist, diese kann man sich somit ebenfalls sparen. Im Algorithmus wird jedoch die erste Methode zur Abstandsbestimmung verwendet. In Abschnitt 3.4 werden die durch beide Methoden erzielten Ergebnisse miteinander verglichen.

Um die Operationen auf dem Blockwörterbuch zu beschleunigen, wird das Blockwörterbuch linear durchsucht um einen bereits vorhandenen ähnlichen Block zu finden, wobei direkt an der Stelle angefangen wird, an welcher der Block zuletzt eingefügt wurde. So ist es möglich bereits bekannte Blöcke sehr effizient zu finden. Da die Blöcke innerhalb des Wörterbuchs ihre Position nicht mehr verändern, ist ein Durchlauf durch das gesamte Wörterbuch nur noch notwendig, wenn es keinen ähnlichen Block in dem Wörterbuch gibt.

3.4 Der Algorithmus

Mit Hilfe des Blockwörterbuchs und der Abstandsfunktionen wird die Klassifikation in Vorder- und Hintergrund vorgenommen. Um zu klassifizieren, welche Blöcke Vorder- und Hintergrund sind, betrachten wir die Anzahl der Referenzen auf einen Block im Blockwörterbuch. Wurde ein Block zum Beispiel mehr als zwei mal referenziert, so gilt dieser Block als Hintergrund. Wurde er nur einmal referenziert oder kam er erst einmal vor, so wird er als Vordergrund betrachtet. Die Anzahl der Referenzen entspricht somit der Häufigkeit mit der ein Block in den letzten Frames beobachtet wurde. Bisher ist der Referenzwert fest kodiert, es ist aber auch denkbar, daß der Wert durch den Algorithmus gelernt wird. Es wäre zum Beispiel möglich die durchschnittliche Größe der als Vordergrund gefundenen Fläche zu betrachten, um zu entscheiden ob der Referenzwert zu

(a) Bild 11 (b) Bild 12 (c) Bild 13 (d) Bild 14

Abbildung 2: Bildsequenz gefiltert mit dem Abstandsmaß aus der ersten Formel

(a) Bild 11 (b) Bild 12 (c) Bild 13 (d) Bild 14

Abbildung 3: Bildsequenz gefiltert mit dem Abstandsmaß aus Formel 1

groß oder zu klein ist. Eine weitere grundlegende Annahme bei dieser Herangehensweise ist, daß sich Vordergrundblöcke häufig verändern, weil der Dozent sich leicht bewegt und dadurch zum Beispiel die Lichtverhältnisse auf seiner Kleidung anders sind, oder große Bewegungen vollführt und sich dadurch ganz andere Ansichten des Dozenten bieten. Hintergrundblöcke sind bildweise sehr homogen, weil die Tafel eine einheitliche Farbe hat und somit eine Struktur aufweist. Außerdem ändern sich Hintergrundblöcke in einer Sequenz von Bildern ebenfalls nicht so stark und es besteht die Möglichkeit Hintergrundblöcke in aufeinanderfolgenden Bildern wiederzuentdecken. Damit ist es also möglich im gegenwärtigen Bild einen Block korrekt als Hintergrund zu erkennen, der in dem unmittelbar vorhergehenden Bild vom Dozenten überdeckt wurde, vor einigen Frames aber korrekt als Hintergrund erkannt wurde sofern dieser noch im Blockwörterbuch gespeichert ist. Dies ist auch der Grund, warum das Blockwörterbuch eine größere Anzahl von Blöcken speichern kann als in einem Bild maximal vorkommen können.

Es wurde mit beiden in Abschnitt 3.3 vorgestellten Methoden zur Abstandsbestimmung experimentiert, leider war das Resultat unter Verwendung der zweiten Methode nicht zufriedenstellend. Abbildung 2 zeigt, vier Bilder aus einem Video das mit dem ersten Abstandsmaß gefiltert wurde. Die Konturen wurden relativ sauber ausgeschnitten und es wurden auch nicht zu viele Blöcke als identisch erkannt.

In Abbildung 3 sind die selben vier resultierenden Bilder gefiltert mit dem zweiten Abstandsmaß dargestellt. Alle anderen wesentlichen Verarbeitungsschritte blieben bei beiden Tests gleich.

Im zweiten Fall ist das Resultat deutlich „ausgefranster“, es wurden viele Blöcke als identisch erkannt die sich eigentlich unterscheiden sollten. Auch eine Veränderung des Schwellwertes für den Abstand oder eine Vergrößerung des Blockwörterbuchs brachte keine Verbesserung, das Resultat enthielt immer deutlich zu wenig oder deutlich zu viele Blöcke als Vordergrund. Formel 2 erscheint somit schlechter geeignet.

Ein Grund warum die erste Methode besser ist könnte der sein, daß in den Abstand die einzelnen Pixelunterschiede zwischen zwei Blöcke fließen. In Formel 2 wird erst so etwas wie ein

(a) Bild 11 (b) Bild 12

Abbildung 4: Bildsequenz gefiltert mit Abstandsmaß aus Formel 2 angewendet auf den RGB Farbraum

Abbildung 5: Identifizierter Vordergrund ohne Zusammenhangsanalyse

Durchschnittsfarbwert für alle in einem Block enthaltenen Pixel gebildet und dann der Abstand zu dem nächsten Block berechnet. Dadurch verliert man Informationen über die Anordnung der einzelnen 12 Pixel innerhalb eines Blocks. Es wurde noch ein Experiment mit Formel 2 durchgeführt, in dem als zugrundelegender Farbraum der RGB Raum erhalten blieb, Abbildung 4 zeigt zwei Bilder aus diesem Experiment, das ursprüngliche Video war das gleiche wie in den anderen Abbildungen. In diesem Farbraum wurde das Resultat noch schlechter und der Dozent verschwand in den nachfolgenden Bildern vollkommen.

Um die in den Abbildungen dargestellten Resultate zu erzielen ist noch eine Nachbearbeitung der gefundenen Blöcke notwendig.

3.5 Komponentenanalyse

Nachdem das Bild in Blöcke zerlegt worden ist, wird das gesamte Bild durchlaufen um eine Komponentenanalyse durchzuführen. Während dieser Analyse werden zusammenhängende Blöcke als „Blobs“ betrachtet. Es werden alle Blobs traversiert und gesucht ist der größte Blob, von dem angenommen wird, daß er den Dozenten zu großen Teilen enthält. Würde diese Analyse nicht gemacht, so wird der als Vordergrund identifizierte Bereich zu groß, da durch Kamerarauschen und auch durch kleine Bewegungen abseits des Dozenten Veränderungen im Bild auftreten. Abbildung 5 zeigt ein Bild in dem die Komponentenanalyse nicht durchgeführt wurde, man erkennt deutlich kleine Cluster von Blöcken, die ebenfalls als Vordergrund erkannt wurden. Diese werden aber durch die Komponentenanalyse in der weiteren Verarbeitung entfernt, sofern sie nicht zu größten Komponente gehören. Die Komponentenanalyse ist wie in Cormen et al. (2001) implementiert und nicht laufzeitkritisch.

3.6 Zusammenschluss

Nachdem die größte Komponente gefunden wurde, wird diese Komponente nachbearbeitet. Da in dem vorherigen Schritt auch diagonal verbundene Blöcke zu einer Komponente zusammengefasst werden können, ist es möglich, daß diese Komponente „Löcher“ enthält. Diese Lücken werden

Abbildung 6: Erkennung ohne Zusammenschluss

(a) Horizontaler Zusammenschluss (b) Vertikaler Zusammenschluss

Abbildung 7: Erkennung mit Zusammenschluss

geschlossen in dem das Bild horizontal oder vertikal durchlaufen wird. Experimente haben gezeigt, das ein horizontales Schließen der Lücken ein optisch besseres Resultat liefert als der vertikale Zusammenschluss. Verwendet man beide, so erhält man in der Regel zu viele Blöcke und unsaubere Konturen. Abbildung 6 zeigt einen extrahierten Dozenten ohne Zusammenschluss, der gefundene Vordergrund enthält große Lücken. Abbildung 7 zeigt dagegen zwei Bilder einmal horizontal einmal vertikal geschlossen.

Die rote Linie in Abbildung 7, die um den Dozenten gezeichnet wurde, ist der gefundene äußere Kantenzug innerhalb dessen Lücken geschlossen werden. Durch den Zusammenschluss der Komponenten ergibt sich ein wesentlich besseres Resultat. Der Dozent enthält keine Lücken mehr, allerdings kann die Fläche die der Dozent einnimmt etwas größer als nötig werden. Die zugrundelegende Annahme bei diesem Vorgehen ist, daß der Dozent keine „Lücken“ enthält. Dies bringt allerdings ein Problem mit sich, sobald der Dozent eine Hand von sich streckt und sie zum Beispiel auf Kopfhöhe hält. In diesem Fall wird fälschlicherweise der Hintergrund zwischen Dozenten und Hand als zum Dozenten dazugehörig klassifiziert. Dieses Problem nennen wir das *„Arm-Kopf-Arm Problem“*. In Abbildung 9 ist dieses Problem dargestellt. Im Kapitel 4 wird auf dieses und andere auftretende Probleme weiter eingegangen.

3.7 Nachrechnen der Kanten

Während der Zusammenschlußphase, werden zusätzlich alle Blöcke bestimmt, die zum äußeren Rand des Blobs gehören. Die so gefundenen Blöcke können im nächsten Verarbeitungsschritt noch einmal in höherer Qualität wie beschrieben nachgerechnet werden. Dazu wird zunächst die Skalierung rückgängig gemacht. Die in dem runterskalierten Bild gefundenen Vordergrundblöcke werden proportional auf das originale Bild mit höherer Auflösung zurückgerechnet. Für die zuvor bestimmten Randblöcke wird nun erneut ein Blockwörterbuch angelegt in das nur der Rand eingefügt wird. Da die Blöcke nun mehr Bildinformation enthalten als zuvor werden andere eventuell identische Blöcke nachträglich aus der Menge der Vordergrundelemente entfernt oder es werden neue Blöcke hinzugenommen. Durch diesen Schritt sollen die Konturen des Dozenten verbessert werden. Abbildung 8 zeigt einen Vergleich zwischen einem Bild, das nicht noch einmal an den

(a) Kanten wurden nicht nachgerechnet

(b) Kanten wurden nachgerechnet

Abbildung 8: Einfluss der Kantenverbesserung

Kanten nachgerechnet wurde und einem bei dem dieser Schritt durchgeführt wurde.

Man erkennt in der linken Teilabbildung deutlich die großen Randblöcke an den Konturen des Dozenten. Diese entstehen durch das proportionale Umrechnen auf das originale Bild. Rechts dagegen sind die Konturen des Dozenten deutlich genauer zu erkennen. Natürlich ist diese Verbesserung mit erhöhtem Rechenaufwand verbunden, je größer der Kantenzug eines Dozenten ist, desto langsamer läuft der Algorithmus in dem dargestellten Bild betrug die Rechenzeit für den linken Frame 13742 ms, die für den rechten Frame dagegen 18321 ms[1].

3.8 Altern von Blöcken

Im letzten Schritt, werden nun alle Blöcke die endgültig als Hintergrund klassifiziert wurden, in der definierten Hintergrundfarbe z.B. weiß gezeichnet. Da für die Klassifizierung im wesentlichen die Bewegung entscheidend ist, kann es passieren, dass ein Dozent,der nach einer Bewegung innehält in der Ausgabe plötzlich verschwindet. In Wirklichkeit befindet sich der Dozent allerdings noch an genau der selben Stelle wie vorher und sollte immer noch als Vordergrundobjekt erkannt werden. Um diesem Problem entgegenzuwirken, wird eine zusätzliche Alterungsmatrix eingeführt. Die Matrix enthält genauso viele Elemente wie es Bildblöcke gibt und die Einträge in dieser Matrix beschreiben wie lange der dazugehörige Block schon als Vordergrund klassifiziert wurde. Bei der Berechnung eines jeden Bildes wird also für jeden Block der Eintrag in der Alterungsmatrix um 1 erhöht, falls der Block zum Vordergrund gehört. In jedem Schritt wo der Block nicht zum Vordergrund gehört, wird der Eintrag um 1 verringert. Beim Zeichnen eines Blockes, der in dem aktuellen Bild nicht als Vordergrund erkannt wurde, geht nun sein jeweiliges Alter mit ein. Je dichter das Alter eines Blocks an 0 liegt umso transparenter wird dieser Block gezeichnet. Ein Block mit Alter 0 ist vollständig transparent. Blöcke, die explizit als Vordergrund klassifiziert wurden, d.h. die sich in diesem Schritt signifikant geändert haben, werden von dieser Alterung ausgenommen. Diese Blöcke werden immer undurchsichtig gezeichnet, da sie auf jeden Fall zum Vordergrund gehören.

Ein Problem, das bei diesem Verfahren auftritt, ist, dass wenn sich ein Dozent sehr lange an einer Stelle aufhält und sich immer leicht bewegt, der Bereich in dem er sich befindet sehr stark „aufgeladen“ wird. Bewegt sich der Dozent aus diesem Bereich heraus, so kann es passieren, dass in dem Resultat ein „Abdruck“ des Dozenten sichtbar bleibt, der erst langsam verschwindet. Um das zu verhindern, kann das Alter eines Blocks maximal so groß werden, dass es $\frac{1}{4}$Sekunde dauert bis das Alter des Blocks auf 0 fällt nachdem keine Bewegung mehr in dem Block vorhanden ist. Bei 10 Bildern pro Sekunde entspräche das einem maximalen Alter von 2. Dieser Parameter wurde in verschiedenen Experimenten bestätigt.

Abbildung 9 stellt das beschriebene Problem anhand einer Bildsequenz dar. Zur besseren Sichtbarkeit des Effektes wurde der Hintergrund diesmal auf die Farbe schwarz gesetzt.

[1] Die beiden Bilder wurden auf einem 2.6 GHz Pentium 4 unter Windows XP berechnet.

(a) Bild 277 (b) Bild 278 (c) Bild 279 (d) Bild 280

Abbildung 9: Alterungsmechanismus

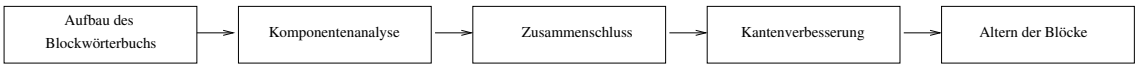


Abbildung 10: Blockdiagram der Verarbeitungskette

In Teilbild 277 sieht man eine ausgeschnittene Person. Im Teilbild 278 wurde der Arm nach unten bewegt, da der Bereich zuvor aber als Vordergrund erkannt wurde, verschwindet er nicht sofort im Teilbild 279 sondern erst allmählich. Anderseits verschwinden die Beine der Person nicht mehr, obwohl sie in der Bilderfolge nicht bewegt wurden. In einem vollständigen Video wird dieser Effekt eher positiv als negativ wahrgenommen. Die Transparenz erscheint dort als kurze Unschärfe, geht aber in der Geschwindigkeit der Bewegung unter. In Abbildung 10 kann der Zusammenhang aller Verarbeitungsschritte in einem Blockdiagramm nachgelesen werden.

4 Ausblick

Dieses Kapitel gibt einen Überblick auf die bisherigen Schwachstellen des Algorithmus und und es werden Möglichkeiten aufzeigt wie diese zu beseitigen sein könnten. Es wird auch ein Ausblick auf die weitere Entwicklung des Projekts gegeben.

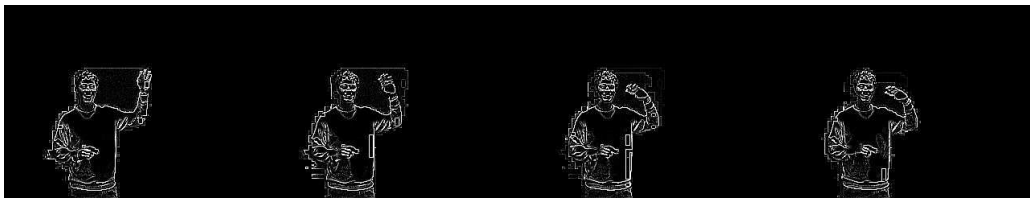
4.1 Das „Arm-Kopf-Arm“ Problem

Ein Problem das bereits in Abschnitt 3.6 erwähnt wurde ist das so genannte *„Arm-Kopf-Arm Problem“*. Dieses Problem, das durch Zusammenschluss der Blöcke entsteht, könnte gelöst werden indem der Algorithmus lernt wie die Tafel aussieht. Kennt der Algorithmus die dominierende Farbe der Tafel, in Abbildung 9 wäre dies das helle Grau, so könnte man Blöcke dieser Farbe aus dem erkannten Vordergrundobjekt entfernen. Analog zur Funktionsweise eines Tarnanzugs, könnte es dann allerdings vorkommen, dass Teile des Dozenten mit verschwinden, falls die Kleidung des Dozenten in einer ähnlichen Farbe wie Tafel gehalten ist. Alternativ könnte man versuchen Kantenübergänge in jedem Bild zu erkennen um die Konturen des Dozenten besser zu erfassen. Der Kantenerkennungsfilter wurde bereits implementiert, dazu wird das Bild zunächst in Graustufen umgewandelt und anschließend ein einfacher Kantenerkennungskernel K darauf angewendet.

$$K = \begin{matrix} -0.5 & -1 & -0.5 \\ -1 & 6 & -1 \\ -0.5 & -1 & -0.5 \end{matrix}$$

Es wurde experimentell bereits ein Video mit diesem Kernel gefiltert. Abbildung 11 zeigt die erkannten Kantenzüge.

Von den erkannten Kantenzügen könnte man den jeweils stärksten, äußeren Kantenzug verwenden und diesen als Grenze zum Dozenten verwenden.



(a) Bild 270 (b) Bild 271 (c) Bild 272 (d) Bild 273

Abbildung 11: Erkannte Kantenzüge

(a) Bild 322 (b) Bild 323 (c) Bild 324 (d) Bild 325

Abbildung 12: Verschwinden von Personenteilen

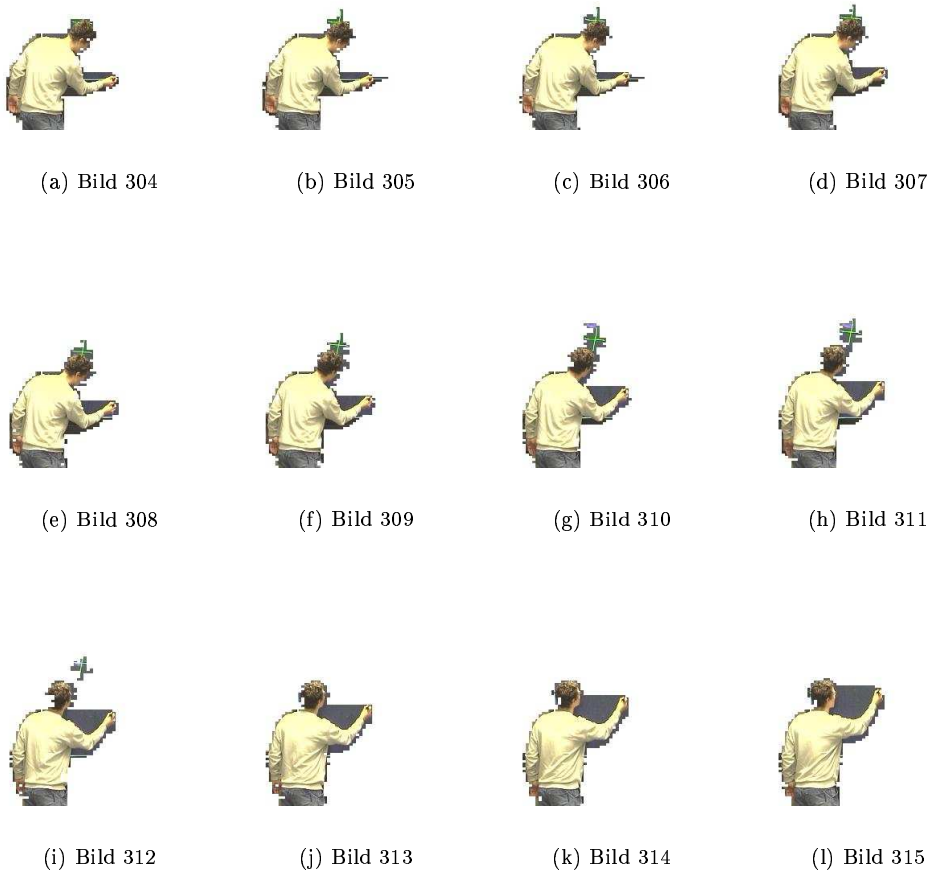
4.2 Verschwinden des Dozenten

Ein weiteres Problem das der Algorithmus hat, welches auch schon durch den Alterungsmechanismus verbessert wird, ist das Verschwinden des Dozenten falls keine signifikante Bewegung mehr vorhanden ist. Um dieses Problem zu lösen wurde auch mit der Bildrate experimentiert. Geringere Bildraten bringen natürlich größere Veränderungen von Bild zu Bild mit sich, allerdings wirkt das resultierende Video dann abgehackt. Experimente haben gezeigt, dass eine Bildrate von 10 Bildern pro Sekunde einen guten Kompromiss darstellt. Abbildung 12 zeigt eine Situation, in der Teile der ausgeschnittenen Person verschwinden. Dieses Verschwinden wird im Video allerdings nur als kurzes Flackern wahrgenommen.

4.3 Verschieben des Tafelinhalts

Ein Problem, welches durch den Algorithmus schon recht gut gelöst wird ist das Verschieben des Tafelinhalts. Diese Verschiebung stellt eine Bewegung des Hintergrunds dar. Die Verschiebung des Tafelinhalts stellt für die Teile der Tafel kein Problem dar, die sich eine Sekunde vor der Verschiebung nicht mehr verändert haben. Diese Teile befinden sich ja bereits in dem Blockwörterbuch und werden somit korrekt als bereits bekannt entdeckt. Teile die weit vom Dozenten entfernt sind, stellen ebenfalls kein Problem dar, da sie durch die Komponentenanalyse wegfallen. Abbildung 13 zeigt eine Bilderfolge in der der Tafelinhalt nach oben verschoben wird.

In der Abbildung ist zu erkennen, dass ein Teil des Tafelinhalts als Dozent erkannt wurde. Allerdings wurde dieser Teil vorher von der ausgeschnittenen Person längere Zeit überdeckt. Man sieht auch, dass die Erkennung wieder besser funktioniert, nachdem der Inhalt sich weiter vom Dozenten entfernt hat. Um dieses Problem noch besser zu lösen könnte man auf Informationen des Tafelstroms zugreifen. Denkbar wäre zum Beispiel solche Verschiebungsereignisse zu erkennen und geeignet zu behandeln. Beispielsweise könnte man ein Wachsen oder ein Verschieben der Fläche des Dozenten in dieser Phase verhindern, da davon auszugehen ist, dass der Dozent sich während



(a) Bild 304 (b) Bild 305 (c) Bild 306 (d) Bild 307

(e) Bild 308 (f) Bild 309 (g) Bild 310 (h) Bild 311

(i) Bild 312 (j) Bild 313 (k) Bild 314 (l) Bild 315

Abbildung 13: Verschiebung des Tafelinhalts

des Verschiebens nicht stark bewegt. Die vorhandenen Tafelinformationen könnten auch verwendet werden um größere farbige Flächen die der Dozent eventuell durch das Einfügen eines Bildes erzeugt auszublenden. Dafür ist es notwendig einem Punkt auf der Tafel genau einem Punkt in dem Video zuzuordnen, diese Aufgabe stellt sich allerdings auch für die skalierte Wiedergabe auf Clientseite, so dass dadurch kein zusätzlicher Aufwand entsteht. Für den gesamten Algorithmus ist auch notwendig die Laufzeit zu verbessern. Die Berechnungszeit für ein Bild variiert zwischen Bruchteilen einer Sekunde bis in zu mehreren Sekunden, je nach Größe des gefundenen Vordergrundblockes.

Das Projekt wurde als Studienarbeit begonnen, wird aber im Rahmen der Forschung weiterentwickelt. Es soll mit weiteren Segmentierungsalgorithmen, wie zum Beispiel in Kapitel **??** beschrieben experimentiert werden um das Resultat weiter zu verbessern.

Literatur

Baar, B. J. (1988). *A Cognitive Theory of Consciousness*. Cambridge University Press, Cambridge, U.K.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2001). *Introduction to Algorithms*, volume 1. The MIT Press, Cambridge, Massachusetts London, England, 2nd edition.

Haritaoglu, I., Harwood, D., and Davis, L. (2000). W4, real-timee surveillance of people and their activities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):809–830.

Jantz, K. (2004). Studienarbeit: Trennung von dozenten und tafel in einem e-kreide video.

Li, L., Huang, W., Gu, I. Y. H., and Tian, Q. (2003). Foreground object detection from videos containing complex background. *ACM Multimedia*, pages 2–10.

Li, L. and Leung, M. K. H. (2002). Integrating intensity and texture differences for robust change detection. *IEEE Transactions on Image Processing*, 11(2):105–112.

Lipton, A., Fujiyoshi, H., and Patil, R. (1998). Moving target classification and tracking from real-time video. In *IEEE Workshop on Application of Computer Vision*, pages 8–14.

Stauffer, C. and Grimson, W. (2000). Learning patterns of activity using real-time tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):747–757.

WWW-EKREIDE (2004). E-Kreide. `http://www.e-kreide.de/`.

WWW-w4 (2004). W4. `http://www.umiacs.umd.edu/users/hismail/ W4_outline.htm`.