

# Ein neues Interface zur flexiblen Navigation in Videos auf PDAs

Wolfgang Hürst, Georg Götz, Martina Welte

Institut für Informatik, Albert-Ludwigs-Universität Freiburg

## **Zusammenfassung**

In dieser Design-Präsentation stellen wir eine Schnittstelle zur interaktiven Navigation in Videos mit Stifteingabe auf PDAs (Personal Digital Assistant) vor. Unsere Lösung erlaubt es Benutzern, auf intuitive Art in unterschiedlichen Granularitätsstufen durch ein Video zu navigieren. Des Weiteren besteht die Möglichkeit einer stufenlos verstellbaren, geschwindigkeitsbasierten Navigation durch die betreffende Datei. Beide Interaktionsmechanismen sind aufeinander abgestimmt und auf eine Weise in das Abspielprogramm integriert, die den nur beschränkt zur Verfügung stehenden Platz optimal ausnutzt.

## 1 Einleitung

SmartPhones, PDAs (Personal Digital Assistant) und ähnliche mobile Handgeräte der neuesten Generation sind mittlerweile derart leistungsfähig, dass sich damit problemlos Videos in unterschiedlichsten Ausprägungen betrachten lassen – angefangen bei selbst aufgenommenen Clips mit niedriger Auflösung bis hin zu hochqualitativen, längeren Videos, wie zum Beispiel ganzen Fernsehsendungen oder Kinofilmen. Das große Spektrum der Einsatzmöglichkeiten wird nicht nur in den vielfachen Ausprägungen des entsprechenden Datenmaterials deutlich, sondern auch in einem sehr breiten Nutzungsspektrum: Während analoge Videokassetten meist zum passiven, kontinuierlichen Betrachten kompletter Sendungen über einen vergleichsweise langen Zeitraum hinweg genutzt werden, gibt es beim Abspielen von Videos auf mobilen Endgeräten äußerst vielfältige Motivationen und damit einhergehend unterschiedliche Interaktionsarten: das gezielte Ansteuern einzelner Ereignisse in einem Ferienvideo, die Betrachtung bestimmter Meldungen im Video-Podcast einer Nachrichtensendung, die gezielte Auswahl einzelner Szenen in einem langen Film usw. Die Vielfalt des Datenmaterials und der unterschiedlichen Nutzungsabsichten der Endanwender resultieren in einem gesteigerten Bedürfnis nach komfortablen, intuitiven und einfach zu handhabenden Navigationsmechanismen. Benutzer sollten in der Lage sein, schnell und einfach einen groben Überblick über den Inhalt eines Videos zu bekommen, einzelne, für sie in der jeweiligen

Situation interessante Stellen schnell zu finden und abzuspielen, auf interessante Stellen einfach zurückzusetzen, besonders wichtige Dinge langsam und detailliert zu betrachten usw.

Leider verhindert die geringe Bildschirmgröße mobiler Handgeräte in der Regel die sinnvolle Umsetzung entsprechender, für derartige Interaktionen geeigneter grafischer Benutzerschnittstellen, wie sie sich für vergleichbare Video-Abspielprogramme auf Desktop PCs und Laptops bewährt haben. Standardverfahren, wie zum Beispiel die automatische Erkennung einzelner Szenen und ihre Repräsentation durch so genannte Storyboards, d.h. verkleinerter Standbilder („Thumbnails“, vgl. Girgensohn et al., 2001), die ein Überfliegen des Inhalts und direktes Ansteuern einzelner Szenen erlauben, sind für die kleinen Bildschirme mobiler Handgeräte beispielsweise wenig geeignet. Verfahren zur so genannten „Auto Summarization“ (siehe z.B. van Houten 2002), also der automatischen Erstellung von kleinen, einem Abstract ähnlichen Zusammenfassungen, sind zwar nützlich, um einen groben Überblick über einen bestimmten Inhalt zu bekommen, eignen sich jedoch offensichtlich nicht für den Zugriff auf eine bestimmte Detailinformation. Vielversprechend erscheinen jedoch im Englischen oft als „Video Scrubbing“ bezeichnete Ansätze, bei denen der Benutzer durch Manipulation der Abspielgeschwindigkeit (Stichwort „Schneller Vor-/Rücklauf“) oder der positionsbasierten Navigation entlang der Zeitachse das visuelle Feedback so manipuliert, wie es für die gegebene Situation geeignet erscheint – zum Beispiel eine beschleunigte Wiedergabe bzw. schnelle Bewegung entlang der Zeitachse, um einen groben Überblick über den Inhalt zu bekommen, oder eine verlangsamte, zeitlupenartige Wiedergabe bzw. langsame Bewegung entlang der Zeitachse, um eine bestimmte Szene detailliert zu betrachten oder nach einem konkreten Ereignis zu suchen. Auch hier erweist sich die geringe Bildschirmgröße jedoch als sehr problematisch. Großflächige GUI-Komponenten wie Einstellknöpfe und Schieberegler, die die erforderliche, vielfältige Funktionalität bieten würden, nehmen sehr viel Platz ein und verringern damit den für die Darstellung des eigentlichen Inhalts zur Verfügung stehenden Bereich. Bei einer verkleinerten Darstellung sind sie mit einem Stift, der bei PDAs das dominierende Eingabegerät darstellt, oft schwer zu treffen, was zu häufigen Fehleingaben führen kann. In dieser Design-Präsentation stellen wir deshalb den MobileZoomSlider vor, ein Interface zur flexiblen, positions- und geschwindigkeitsbasierten Navigation in Videos auf mobilen Handgeräten mit Stifteingabe, wie z.B. herkömmlichen PDAs. Er basiert auf einem von uns ursprünglich für die Navigation in Videos auf Desktop PCs entwickelten Ansatz (Hürst 2006).

## 2 Der MobileZoomSlider

Die Grundidee des MobileZoomSliders, die ihn insbesondere für die kleinen Bildschirme mobiler Handgeräte interessant macht, ist es, ohne direkte Bedienelemente auszukommen und stattdessen die komplette Bildschirmfläche sowohl zur Wiedergabe, als auch zur Interaktion zu benutzen. Das Abspielen eines Videos geschieht daher im Vollbildmodus. Etwaige notwendig werdende Steuerelemente wie z.B. Start-, Stopp- und Pausetaste, werden bei Berühren des Bildschirms am unteren Rand eingeblendet (vgl. Abb. 1a). Zur positionsbasierten Navigation entlang der Zeitachse besitzt der von uns implementierte Video-Player einen,

von vergleichbaren Medien-Playern bekannten Slider, mit dem man sich entlang der Zeitachse durch das Video bewegen kann (vgl. Abb. 1b).



Abbildung 1: (a) Einblendung der Standard-Bedienelemente bei Berühren des Bildschirms, (b) Slider zur positionsbasierten Navigation (oben), Abspieldauer, Gesamtlänge und Abspielgeschwindigkeit (links unten), Abspielbuttons (Mitte/rechts) und Icon zum Verlassen des Vollbildmodus, Laden neuer Files, etc. (rechts unten).



Abbildung 2: Positions-basierte Navigation mit dem ZoomSlider durch Klicken auf den Bildschirm mit anschließender Links-/Rechts-bewegung. Ein Piktogramm verdeutlicht die Funktionalität.

**Positions-basierte Navigation.** Zusätzlich zu den oben erwähnten Standardfunktionalitäten gibt es nun die durch den ZoomSlider realisierten Navigationsvarianten, die, wie bereits erwähnt, unabhängig von konkreten Interaktionselementen realisiert werden und durch direkte Interaktion auf der Video-Wiedergabefläche des Players erfolgen. Berührt ein Benutzer mit dem Stift diesen Wiedergabebereich, kann er durch Links- und Rechtsbewegungen entlang der Zeitachse durch das Video navigieren, vergleichbar einer Interaktion mit dem am unteren Rand dargestellten Slider-Interface (vgl. Abb. 2). Die zwei wesentlichen Vorteile sind jedoch zum einen, dass der Benutzer die entsprechende Funktionalität unabhängig von einer relativ kleinen, mit dem Stift schwer exakt zu treffenden Schnittstellenkomponente abrufen kann. In informellen Tests haben wir festgestellt, dass das sich ggfs. sogar bewegende Slider-Interface selbst bei vergleichsweise großer Realisierung oft schwer genau anzusteuern ist. Beim ZoomSlider dagegen positioniert der Benutzer den Stift „irgendwo“ auf dem Bildschirm und bekommt im Anschluss durch die exakt identische Eingabe (Links-/Rechtsbewegungen des Stifts) das gleiche Feedback. Der zweite wesentliche Vorteil dieser Umsetzung ist es, dass wir durch die zusätzlich zur Verfügung stehende Fläche (der gesamte Bildschirm wird ja für die Interaktion ausgenutzt), die Möglichkeit haben, relativ einfach unterschiedliche Granularitätsstufen der zugrunde liegenden Zeitachse zu realisieren. Der Standard-Slider am unteren Bildrand ist in natürlicher Weise durch die Bildschirmgröße beschränkt. Beginn und Ende des Videos werden auf das linke bzw. rechte Ende des Sliders abgebildet. Dazwischen findet eine lineare Interpolation zwischen Slider-Positionen und der durch die Länge des Videos definierten Zeitachse statt. Bei kleinen Bildschirmen führt dies dazu, dass bei der Bewegung

des Sliders entlang der Zeitachse einzelne Frames des Videos übersprungen und bestimmte Szenen schlimmstenfalls gar nicht mehr direkt angesteuert werden können.

Bei Standard-Video-Playern wird dieses Problem z.B. durch das Einblenden zusätzlicher Zeitskalen unterschiedlicher Granularität gelöst (siehe z.B. Casares et al. 2002), oder es werden weitere Bedienelemente zur manuellen Modifikation der Skalenauflösung zur Verfügung gestellt. Derartige Lösungen haben nicht nur den Nachteil, dass sich ein Benutzer oft mehr mit der Bedienung unterschiedlicher Interaktionselemente beschäftigt statt mit dem eigentlichen Inhalt, sondern verbieten sich bei mobilen Endgeräten natürlich wieder aufgrund des beschränkten Platzangebots. Beim ZoomSlider wird diese Problematik vergleichsweise einfach gelöst, indem die Auflösung der Zeitskala, entlang derer ein Benutzer durch Links-/Rechtsbewegungen im Video navigiert, proportional zum Abstand vom unteren Bildschirmrand gesetzt wird: Bei einer Links-/Rechtsnavigation am unteren Rand des Bildschirms (also in der Nähe des „normalen“ Slider-Interfaces) hat man eine relativ grobe Auflösung der Zeitachse, die der durch die Bildschirmgröße implizit vorgegebenen Auflösung des Standard-Sliders entspricht. Je weiter man sich vom unteren Rand entfernt, desto detaillierter wird die Auflösung der Zeitachse, d.h. die Navigation wird entsprechend feingranularer. In der Nähe des oberen Bildschirmrands bekommt man schließlich die kleinstmögliche Auflösung. Hier entspricht ein Pixel auf der virtuellen, durch Links-/Rechtsbewegung an dieser Position definierten Zeitachse genau einem Frame im Video. Visualisiert wird diese Funktionalität durch eine entsprechende Anpassung der Skala im Original-Slider am unteren Bildschirmrand: Bei größerer Auflösung im unteren Bildschirmbereich sind die entsprechenden Striche auf der Zeitachse enger beieinander, bei einer feingranularen Navigation im oberen Teil des Bildschirms liegen sie dementsprechend weiter auseinander (vgl. Abb. 3). Dieses „Hineinzoomen“ in die Skala war auch der Grund für die Namensgebung des ZoomSliders.

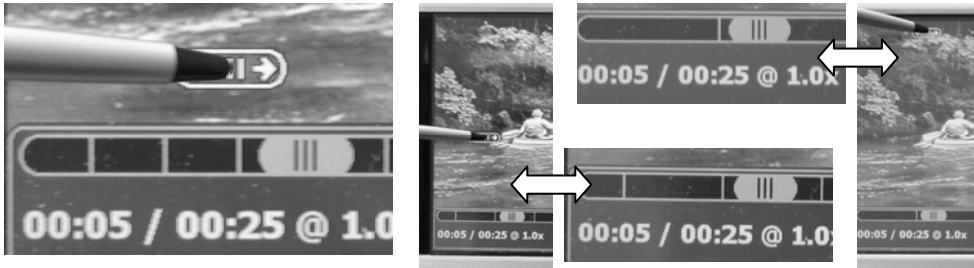


Abbildung 3: Anpassung der Skala in Abhängigkeit von der horizontalen Position des Stifts.

**Geschwindigkeitsbasierte Navigation.** Zusätzlich zur oben diskutierten positionsbasierten Bewegung bietet sich auf mobilen Geräten auch die geschwindigkeitsbasierte Navigation durch ein Video an. Dabei werden einzelne Positionen nicht wie zuvor mit einem Slider direkt angesteuert, sondern die Wiedergabegeschwindigkeit des Videos wird entsprechend der jeweiligen Bedürfnisse angepasst: Langsam bis hin zur Zeitlupe für eine detaillierte Suche in der Datei, schnell in einer Art „Fast Forward“, um zum Beispiel einen groben Überblick über den Inhalt zu bekommen. Die Nützlichkeit einer derartigen Funktionalität scheint zwar offensichtlich, offen ist jedoch ihre konkrete Umsetzung. Die entsprechende Standardrealisierung mittels diverser zusätzlicher Tasten oder einem Schieberegler zum flexiblen

Einstellen der Abspielgeschwindigkeit verbietet sich bei mobilen Geräten wieder aufgrund des bereits diskutierten geringen Platzangebots. Beim ZoomSlider wird das Dilemma, einerseits mehr Funktionalität anbieten zu wollen, andererseits jedoch das Interface weder visuell noch bezüglich der Bedienung zu überfrachten, wieder dadurch gelöst, dass die entsprechende Funktionalität durch direkte Interaktion auf der Wiedergabefläche abrufbar ist. Bewegt ein Benutzer den Stift an den rechten Bildschirmrand (vgl. Abb. 4), resultiert dies in einer kontinuierlichen Wiedergabe des Videos in einer bestimmten Geschwindigkeit: Am oberen Ende des Bildschirms erfolgt ein langsames, zeitlupenartiges Abspielen, am unteren Bildschirmrand erzielt man die schnellstmögliche Geschwindigkeit, die von der Player-Software unterstützt wird (vergleichebar dem von Videorecordern bekannten „Fast Forward“). Dazwischen findet eine lineare Interpolation der Abspielgeschwindigkeiten statt. Visualisiert wird dies durch Piktogramme auf dem Bildschirm (vgl. Abb. 4) sowie die Angabe der Abspielgeschwindigkeit bei den normalen Interaktionselementen des Players (vgl. Abb. 1b).

Der Benutzer hat somit die Möglichkeit, die Abspielgeschwindigkeit kontinuierlich zu verändern. Im Gegensatz zu normalen Schiebereglern zur Einstellung der Abspielgeschwindigkeit, bei denen das vergleichsweise kleine Interaktionselement direkt getroffen werden muss, lässt sich hier die entsprechende Funktionalität wieder recht einfach abrufen, indem man den Stift einfach „irgendwo“ am rechten Rand positioniert und auf- bzw. abbewegt. Tests mit verschiedenen Benutzern ergaben, dass diese Art der Geschwindigkeitsmanipulation als einfach zu handhaben empfunden wird, was insbesondere an dem leicht hervorstehenden Rahmen des Geräts um den Bildschirm herum liegt – der Stift lässt sich hier sehr leicht entlang dieses physikalischen Widerstands bewegen.



Abbildung 4. Geschwindigkeitsbasierte Navigation am Rand des Bildschirms: Oben = langsam (linkes Bild), unten = schnell (rechtes Bild).

Die Abbildung der unterschiedlichen Geschwindigkeitsstufen (oben = langsam bis unten = schnell) entspricht den Skalenauflösungen bei der zuvor beschriebenen positionsbasierten Navigation (oben = feingranular bis unten = grobe Auflösung). Diese Abstimmung der beiden Interaktionskonzepte aufeinander zeigt die eigentliche Stärke des ZoomSlider-Konzepts. Während eine positionsbasierte Interaktion in der Regel besser für eine kurzzeitige Navigation sowie zur Detailsuche geeignet ist, findet ein geschwindigkeitsbasiertes Navigieren eher bei einer länger dauernden Navigation Anwendung, oder wenn man sich einen größeren Überblick über den Inhalt verschaffen will. Mit der Realisierung des ZoomSliders werden beide Interaktionskonzepte nicht nur unterstützt, sondern auch insofern aufeinander abgestimmt, dass man einfach und intuitiv zwischen beiden wechseln kann. Die resultierenden Scrollgeschwindigkeiten sind dabei aufeinander abgestimmt. Das heißt, bewegt man sich beispielsweise sehr schnell entlang einer gröber aufgelösten Skala, geht man automatisch auch in eine schnelle, geschwindigkeitsbasierte Navigation über. Bei einer feinauflösenden

Zeitachse am oberen Bildschirmrand ist die entsprechende Geschwindigkeit am rechten Rand dagegen ebenfalls langsamer. Somit werden typische Interaktionsmuster, wie zum Beispiel das schnelle, geschwindigkeitsbasierte Scrollen durch ein Dokument bei der Suche nach einer bestimmten Information sowie das anschließende, positionsbasierte Zurücksetzen auf den Anfang einer gefundenen Szene optimal unterstützt.

### 3 Implementierung und Benutzerfeedback

Der zuvor beschriebene und in den Abbildungen illustrierte MobileZoomSlider wurde auf einem Dell Axim™ X51v-624 Handheld PDA implementiert, der einen Intel® XScale-TM PXA270-Prozessor mit 624 MHz sowie ein 3,7-Zoll-TFT-VGA-Display mit einer Auflösung von 640x480 Pixeln besitzt. Als Grundlage für die Video-Wiedergabe diente der als Open-Source erhältliche TCPMP (The Core Pocket Media Player, siehe „TCPMP“). Nach zahlreichen informellen Tests, die vor allem der Parameteroptimierung dienten, wurde mit der abschließenden Implementierung eine erste qualitative Benutzerstudie mit zwölf Personen durchgeführt. Insgesamt konnten damit unsere zuvor gemachten Aussagen bezüglich Intuitivität und Usability bestätigt werden. Beispielsweise waren sämtliche Teilnehmer ohne lange Eingewöhnungsphase in der Lage, den MobileZoomSlider nach kurzem Ausprobieren problemlos zu bedienen und die ihnen für diese Studie gestellten Aufgaben zu lösen. Die Visualisierung wurde als intuitiv und – obwohl über dem eigentlichen Videoinhalt liegend – nicht als störend empfunden. Einige Benutzer merkten jedoch an, dass eine umgekehrte Abbildung der Geschwindigkeit und Skalenauflösung (also z.B. oben schnell und unten langsam statt umgekehrt) intuitiver wäre. Da keine rationalen Gründe für eine der beiden Realisierungsvarianten gefunden werden konnten, scheint dies ein individueller Parameter zu sein, der daher in zukünftigen Implementierungen vom Benutzer konfigurierbar sein sollte. Weitere Ansatzpunkte für zukünftige Arbeiten bilden eine genauere Untersuchung und Optimierung der Granularitätsstufen bei der Skalenauflösung, da es Hinweise in der Evaluation gab, dass eine lineare Interpolation hier nicht optimal zu sein scheint.

#### Literaturverzeichnis

- Casares, J., Myers, B.A., Long, A.C., Bhatnagar, R., Stevens, S.M., Dabbish, L., Yocum, D., Corbett, A. (2002): Simplifying Video Editing Using Metadata. Proceedings of Designing Interactive Systems (DIS 2002), London, UK.
- Hürst, W. (2006): Interactive audio-visual video browsing. Proceedings of the 14th Annual ACM International Conference on Multimedia, Santa Barbara, CA, USA.
- Girgensohn, A., Boreczky, J., Wilcox, L. (2001): Keyframe-Based User Interfaces for Digital Video. IEEE Computer, 34(9), pp. 61-67.
- TCPMP (2007): The Core Pocket Media Player, <http://tcpmp.corecodec.org/>
- Van Houten, Y. (2002): A framework for video content browsing. Enschede, Telematica Institute, June 2002. Siehe <http://doc.telin.nl/dscgi/ds.py/ViewProps/File-12409>