Authoring on the Fly - Erzeugen multimediailer Dokumente on the Fly

Wolfgang Hürst, Prof. Th. Ottmann

Institut für Informatik
Albert-Ludwigs-Universität Freiburg
Basic idea:

Production of multimedia documents for teaching and learning requires a lot of resources (time & money)

Idea: make use of the educational potential at universities and organizations by exploiting the high efforts already invested in live-events in order to automatically produce instructional media content for later, offline usage

Involves questions such as:

- How can an automatic recording be done without interfering with the live-event?
- How can a high quality be guaranteed?
- How should the recording be done in order to provide a high usability for offline usage?

etc.
Authoring on the Fly – Phase I & II

Main goals:
- Automatically produce high quality documents, i.e. authoring of multimedia documents „on the fly“
- Preserve live character & keep information loss to a minimum (for quality, archiving, and retrieval)
- Provide flexibility to produce different target documents (for arbitrary distribution scenarios and replay infrastructures)

AOF achieves these goals:
- By supporting a multi-stream capturing
- By using an open intermediate format
- By building on a generic replay architecture
The **AOF approach**
- Proved its feasibility in various applications at different universities and organizations
- Has become a commercial product now, called Lecturnity (see http://www.lecturnity.de)

**Research** has focused on new tasks related to automatic lecture recording, e.g.
- Annotations by students (live and offline)
- Support of mobile devices (LARA)
- etc.

As part of **V3D2**:  
- Indexing & Information Retrieval  
- User interfaces (for both: lecturers & students)  
- Multimedia data browsing and skimming
Demo:
Lecturnity (recording & replay)
Problem with sliders:
They do not scale to random document sizes

Result:
- Jerkey visual feedback
- Loss of information even when slider is moved very slowly
Sliders for Visual Data Browsing

Three new slider designs which solve the scaling problem and enable fine as well as coarse navigation through a document:

1. **ZoomSlider**
2. NLslider
3. Elastic Panning
**ZoomSlider (I)**

**ZoomSlider:** Zoom into the scale of the slider by moving the pointer vertically

Hence: Scrolling gets slower the more the pointer moves away from the original slider.
ZoomSlider (II)

Implementation:
Mapping of pointer position to scale resolution

Original slider
Linear mapping

Sinus mapping

Linear mapping
Border of player window
**Problem** with this approach (and any other approach which relys on using a different scale for fine granular scrolling):

Scale expands across the window’s borders
Sliders for Visual Data Browsing

Three new slider designs which solve the scaling problem and enable fine as well as coarse navigation through a document:

1. ZoomSlider

2. NLslider

3. Elastic Panning
NLslider (Non-Linear Slider):
Scale with a higher resolution around the current position → Enables finer navigation around current location as well as fast skimming of larger parts of the document
Initial **usability testing** with 12 test users

Data: Video & text
Interfaces: ZoomSlider, NLslider, standard slider / scrollbar
Tasks: Search for specific information in a long file

- Both new sliders worked very well
- But both also have their disadvantages
- In addition, there are personal preferences
- ZoomSlider generally was the fastest
Three new slider designs which solve the scaling problem and enable fine as well as coarse navigation through a document:

1. ZoomSlider
2. NLslider
3. Elastic Panning
**Elastic interfaces** for visual data browsing and navigation (introduced by Masui et al., 1995 for static data)

Combine positon-based navigation with speed-based skimming

*Rubber-band metaphor*
Adapting this approach to skim continuous data is pretty straight forward, but how about **usability** in relation to time-dependent data streams?

Tests showed that it works quite well but some drawbacks exist; main problem:

- Accidental changes of scrolling direction

(Note: This holds for static data as well)
Elastic Panning (I)

Idea: Put both, interface / widget and document into the same focus of the user ⇒ Elastic Panning

1. Click anywhere on the screen
2. Clicking position ⇒ current slider thumb position
3. Window borders ⇒ mapped to slider borders
4. Horizontal mouse movements ⇒ Elastic slider functionality
5. Vertical mouse movements ⇒ Ignored
Elastic Panning (II)

Evaluation:

- Revised visualization based on users' feedback
- No accidental changes in scrolling directions
- All advantages of the elastic slider, i.e.
  - Every position is accessible
  - Smooth visual feedback
    (Independent of the actual length of a document)
- Good for full-screen mode and when using a pen as input device
Alternative Slider Designs to Solve the Scaling Problem – Conclusion

Three new slider designs:

1. ZoomSlider
2. NLslider
3. Elastic Panning

Conclusion:

All three work well and solve the scaling problem.

So, which one’s the best?

The answer depends on the task, data and input device.

So, which one’s the best for our task (lect. recs.)?

The answer will be given later.
**Problem:** Visual browsing works quite well in many situations but fails if the visual signal does not contain much changes over a longer period of time.

**Solution:** Approaches for audio / speech browsing

- Transformation to static signal, e.g. transcript of the speech signal
- Segmentation, summarization, etc. (for static representation or to support better interaction & navigation)
- Time-scaling (faster & slower replay)
Audio Browsing – Related Work

Example: *SpeechSkimmer* (B. Arons, 1997)

![Diagram of Audio Browsing Features](image)
Interactive, dynamic audio skimming: **Time-Scaling** (compression & expansion), e.g. with the SOLA algorithm to preserve pitch

Different studies proved its usefulness:

- Speech is still understandable at up to 1.6 till 1.8 times the normal replay rate
- One is able to identify the overall topic at up to 2.5 till 3.0 times the normal replay rate

In relation to learning:

- 2 times double speed might be more effective than 1 time normal speed
- Students concentrate more (& learn better) with higher replay speed
- Slower replay (i.e. 0.84) might be better for non native speakers
Audio Browsing – Interfaces

Time-scaling can be very powerful for search, i.e. to skim audio files for relevant information.

However, the **interface** has to provide flexibility and interactivity; most interfaces just offer a simple **speed controller** (even the SpeechSkimmer!)

![Image of a speed controller interface for audio browsing]

Wolfgang Hürst, Institut für Informatik, Albert-Ludwigs-Universität Freiburg, Germany
Audio Browsing – Interfaces

This has several disadvantages, e.g.:

- Users often have problems picking the right replay speed, e.g. studies indicate that user’s perception might not be linear

- Low flexibility, frequent speed changes are hard to do, no possibility to go back, etc.

Why no `audio scrollbar´? Thumb movements are generally too fast, too slow, and too jerky
Audio Brows. – Elastic Audio Slider

Elastic slider with audio feedback

Distance-to-speed mapping (visual)

Distance-to-speed mapping (audio)

Wolfgang Hürst, Institut für Informatik, Albert-Ludwigs-Universität Freiburg, Germany
Audio Brows. – Elastic Audio Slider

Implementation:

ORIGINAL SIGNAL

SOUNDBROWSER

SOLA

SPEED-UP s(d)

DISTANCE-TO-SPEED MAPPING

DISTANCE d(t, p)

TIME-SCALED SIGNAL

USER INPUT
(Pointer position p)

AUDIO PROGRESS
(Thumb position t)
**Evaluation**: Different tests with news show recordings

**Result**: Works very well for quick speed-ups & slow-downs, but not for continuous listening with different but fixed speed

Therefore: speed controller & elastic audio skimming are both useful and important ⇒ Combine speed controller with elastic audio slider

Main advantages of the elastic audio slider:
- Offers more flexibility for temporary speed-ups
- Integrates seamlessly into common UI design
Conclusion

Visual browsing: 3 sliders which solve the scaling problem
- ZoomSlider
- NLslider
- FineSlider / Elastic Panning

Audio browsing:
- Elastic Audio Slider

Which of the visual sliders is best for our scenario (i.e. browsing of lecture recordings)?

The elastic slider, because we can combine it with audio feedback!
Conclusion

The elastic slider approach enables `real´ multimedia data browsing by combining elastic audio skimming with the elastic visual panning approach.

Distance-to-speed mapping of the aofJSync implementation:

Demo: aofJSync
Summary – Multimedia Browsing

Visual browsing:
  Static: Thumbnails (Keyframes, Slides, …)
  Dynamic: 3 sliders which solve the scaling problem
  • ZoomSlider, NLslider, FineSlider / Elastic Panning

Audio browsing:
  • Elastic Audio Slider

Multimedia / Audio-visual browsing:
  • Elastic Slider with visual & acoustic feedback for `real´ multimedia data browsing

Applications:
  Very good for browsing of lecture recordings, but might be useful for other data & scenarios as well, e.g. video