System for Rapid Subtitling

Master of Engineering Thesis Proposal, MIT Department of Electrical Engineering & Computer Science

Sean Leonard

Spring 2004, 5/10/2004 2:02:00 PM (rev 2)

Thesis Supervisor: Prof. Harold Abelson

1. Title

The full working title of my thesis is "Subtitling system for streamlined overlay of temporal elements on audiovisual sequences, with automatic phonetic alignment of textual elements, and social and legal consequences of subtitling activities in Japanese animation fandom."

2. Overview and Historical Motivation

Subtitling is an important practice for academia and industry in the global economy. My interest in this field stems from my interest in Japanese animation, where subtitling of Japanese-language works is a regular practice. I propose to design and implement an open-source subtitling system that streamlines and automates the preparation of subtitles, to develop a user base for the software, to evaluate the software's usefulness in that community, and to discuss the system's implications for subtitling efforts and thus for broader copyright concerns. This system has both technical and "build a cool tool for the world" contributions. My program will streamline preparation by employing a well-designed user interface, so that users can time scripts much faster than current tools allow ("cool tool"). My program will automate preparation by processing audiovisual streams to extract and apply timing information to transcripts of the dialogue, extending and combining techniques in computer speech and vision (technical). Users of my software would include fan subtitlers (fansubbers), academics, and professional translators. My software would address the peculiarities of subtitling in the Japanese animation community, but would easily generalize to subtitling of various media in many languages.

There is a need for a subtitling program because no modern subtitlers exist for these users. While commercial subtitling systems exist, their users are large broadcasting houses, and their cost and complexity are beyond the reach of fans, academics, and freelance translators. Furthermore, these commercial systems do not support international features such as Unicode, language selection, collaborative translation, multilingual font selection, scrolling text, or karaoke. The last software for subtitling by the freelance community was released in 2000 and was based on workflows for generator-locking devices (genlocks) that are more than a decade old. According to [1], subtitling a 25-minute video sequence requires about 4 hours with current tools. I hope that my software, with its clean user interface, will reduce this time to 2½ hours per 25-minute video sequence. I further hope that the automated component will reduce this time to ten minutes per sequence.

3. Project Details

3.1. Design of User Interface

The user interface is the key to the system's popularity with users. Subtitling with my system should be quick and intuitive, ultimately saving time without a steep learning curve. I will not simplify the interface to the level of the novice user; rather, I will explicitly state the level of preparation required (e.g., learning certain keyboard shortcuts), and will fit user operations into a workflow that maximizes efficiency. Consider in Figure 1 the user interface for Sub Station Alpha **Error! Reference source not found.**, the most popular (and only) program for subtitling:

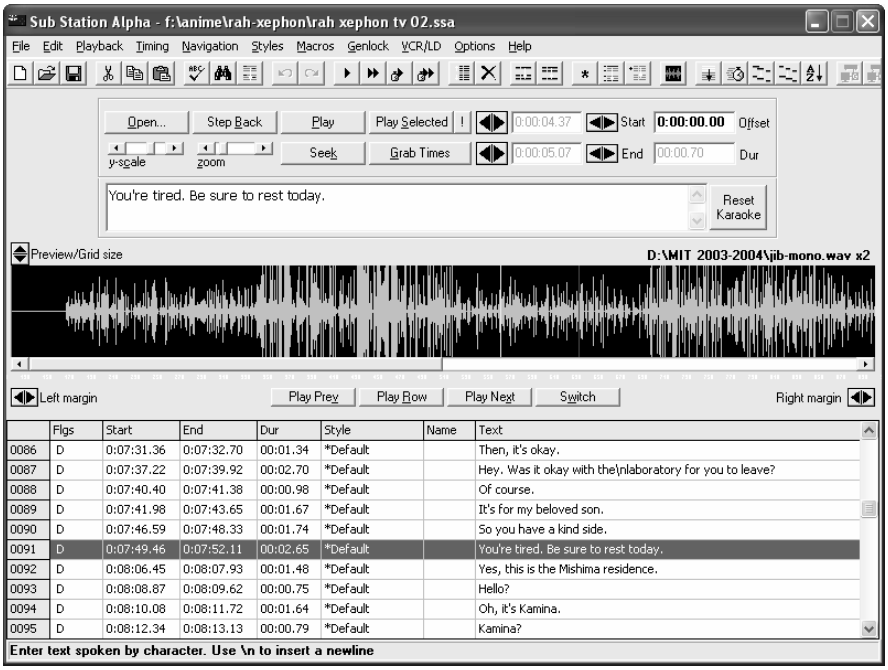

Figure 1: Sub Station Alpha 4.08. The "linear timeline view" prevents parallel workflows.

Sub Station Alpha's "linear timeline view" suffers from a number of drawbacks. First, keyboard shortcuts are awkward or do not work. Second, the waveform preview acts inconsistently: sometimes a click will update the time, other times it will not. Third, subtitles are arranged in single-file order down the table (bottom); there is no attempt to organize subtitles by author, character, or style, and there is no option to view multiple subtitle sections at once.

Rather than emulate existing interfaces, my subtitling system calls for a complete re-evaluation of the needs of subtitlers. I plan to rethink the interface from the ground up, starting with interviews and recommendations from current practitioners. Video display is a must, but video preview is not an end in-and-of itself. Floating windows, multiple views, interactive previews, multiple clips in a loop,

visualization of spectral analysis, and intuitive karaoke input are some of the many possibilities. With multiple monitors, it may be possible to display enough information that accurate timing can occur in realtime, *e.g.*, a display with live video plus video several frames in advance, waveform views that show anticipated dialogue, and bookmarks based on the subtitles to which reviewers can rapidly traverse.

3.2. Research and Algorithm for Automation

While the problems raised by automation in my subtitling system are domains with substantial research, there are opportunities to create new knowledge in these areas. Taking an English/Japanese transcript and a Japanese animation sequence and turning them into a subtitled work evaluates to two problem domains: phonetic audio alignment, and video scene boundary and character recognition. My solution is to solve both of these cases, and to combine them for optimal extraction of timing data.

A sizeable corpus of research has been conducted on speech recognition and synthesis. Phonetic alignment falls under this broad category, and multiple systems exist in fully-specified paper form. For example, Huang et. al. implemented a SPHINX-II system [3] that is available for implementation. My interest lies in the case when phonetics are already specified (*i.e.*, when a transcript is available) and in order, but must be aligned to an audio stream by recognizing the phonetics in the speech. Recent work by Ezzat, et. al. [4] suggests that my subtitling system is possible to implement for cases when the repertoire of the recognition system is limited.

Japanese language betrays many complications, however, which are worth noting. Most systems for phonetic alignment have been tested in limited English, not in large-volume Japanese or other languages. Further research is necessary. It is possible that the repertoire of syllables be fewer than English (Japanese has fewer *mora*, or syllable-units, than English), but Japanese tends to be spoken faster than English, and the phonetic alignment routine must treat a complex and noisy waveform. In literature on the topic, researchers almost always provide a single, unobstructed speaker as input data to their systems. Using an audio stream that includes music, sound effects, and other speakers presents additional algorithmic challenges that require additional study.

Likewise, Japanese animation tends to cast a great variety of characters with a few voice-types. These variations may confuse the alignment routine, and may prevent detection of speaker change when two similar voices are talking in succession (or worse yet, at the same time). To compensate for this, I propose employing "video scene boundary and character recognition" to identify the speaker,

and thus to help decide when to time a dialogue event. Recent work by Wang and Chua [5] discuss video scene boundary detection based on the principle of continuity, but their results have only 82.7% precision after an intensive study of footage. While impressive for videos, 82.7% precision is far too low for a subtitle application. I am still searching for research on character recognition, but have good reason to believe that such research exists as an extension to the general face recognition problem. Fortunately, Japanese animation frequently consists of well-defined, high-contrast lines and solid swashes of color under light or shadow. Furthermore, in most scenes only one character talks at a time. Characters' musings are represented by small, repeated, unsynchronized lip movements in the well-defined area inside a character's outline, since to reduce costs Japanese studios tend to record voices after the production of the original animation (*afureko*). These characteristics of Japanese animation make it much more tractable than the general case—at least in principle—so I should not have too much difficulty specifying a video processor to aid the timing process.

3.3. Choice of Platform

I would prefer to create a platform-agnostic tool so that many others can implement my system. My system, however, employs several different technologies that have traditionally resisted platform-independence. The user interface must include an audio waveform view and a live video preview with dynamic subtitle overlay. The project may demand additional video views such as multiple frames side-by-side, multiple video loops side-by-side, zoom, pan, color manipulation, or detection of mouse clicks on specific pixels. The automation component requires a transcript of the original language as well as the target language in order to perform phonetic audio alignment: for Japanese, this requirement means that advanced text services [6] must convert *kanji* (Chinese characters) to *furigana* (ruby text [7] to aid in phonetization of ideograms), and must recognize distinct words and phrases (Japanese has no spaces). These functions, which are found on modern versions of Windows, are not central to my technical contribution but are central to the system's practical implementation.

Furthermore, most subtitle preparers use Windows machines because existing subtitling software is Windows-based, and because Windows has a very mature multimedia API through DirectShow. Unless I find cross-platform toolkits that can address these issues, I will probably implement the system in C++ on Windows.

3.4. Evaluation and Testing

Throughout the design and development of this project, I will ensure that subtitle preparers know about the project via word-of-mouth (through various communication mechanisms, including the Internet). I will ask all users of this software. Because this community has been looking for a new subtitling application for many years, many users (500 individuals and groups) are expected to jump on the bandwagon. While evaluating the system's use in the community, I plan on extracting more primary source material for my study of fansubbing from 1993 (when my "Progress Against the Law" stopped discussing) through the present.

3.5. Implications

This work has several implications that I plan to discuss, based on my evaluation data. A revised subtitling tool may change relationships in fansubbing groups, as the role of the timer may be significantly diminished in a group if my automation algorithm proves successful. I hypothesize that the timer position will never be fully eliminated, however, because fansubbers routinely embellish their subtitles with puns, cultural explanations, and translations of written Japanese in the video—my algorithm does not attempt to replicate these features. The tool has strong implications for the ways that academics communicate and teach media, because they will be able to subtitle obscure works with ease. These activities, in turn, imply that distribution of these translated and timed works may change, and that cross-cultural distribution may explode once this system is coupled with an accurate translation process. Based on my work with fansubbing and copyright [8], I would like to suggest different forms that copyright may take in support of these creative endeavors.

This tool suggests a solution to tagging media in many different forms. One could use the index points output from the timed algorithm as a kind of digital watermark, uniquely identifying some media based on its response to a script comparator. While this system may have some draconian implications, consider the following advantageous outcome. One may use the index points to define a *temporal spline*, which would later be use for dynamic realignment and reconstruction of subtitles. The underlying media would not have to be entirely the same second-by-second: if commercials were added (or removed!), or if the media was recorded (legally!) off of television for personal use, one could apply the same subtitle script with unique index points to warp the spline around the new audiovisual cues. These cues could be pre-computed concurrent with recording of the media, so that no preprocess time would be required when applying the translation script. The result, of course, would

be near-perfect alignment of subtitles, even reconstruction of subtitles from multiple sources, for every viewing of a media clip.

4. Collaboration

Although I intend to do the majority of the design and implementation and all of the thesis writing, my subtitling application would not be possible without the support of other dedicated individuals and groups. James Roewe '04, Ariel Rodriguez '05, and Diane Christoforo '05 intend to help implement the system and release it to the community. I am actively seeking the help and advice of other groups in CSAIL, such as the Computer Graphics Group (Prof. Frédo Durand) and the Spoken Language Systems Group (Tony Ezzat). Several translators (Neil Nadelman) and fansubbers have already agreed to help define and evaluate—with copious feedback—the features of the system as relates to their daily practice.

5. Timeline

7/2004: Research existing art and solutions in the field.
Query fansubbers, academics, and professionals—get an idea of what users want, and evaluate their workflow.
8/15/2004: Present mock user interfaces to test subjects. Learn three APIs (DirectShow, Windows display, file format manipulations) relevant to the implementation process.
9/15/2004: Design thorough system model with classes, data structures, and relevant decompositions.
10/15/2004: Implement main UI and design.
11/15/2004: Finalize UI and system; debug 90% of errors.
12/15/2004: Complete research into fansubbing 1993-2004. Implement phonetic alignment; test on simple video clips.
1/30/2005: Test complex video clips and refine phonetic alignment. Implement scene boundary and character detection, compare with audio-only results. Release to the community, requiring responses to survey as condition for use.
3/15/2005: Receive survey results. Aggregate data and start writing evaluation.
4/20/2005: Complete evaluation, refine subtitling system. Discuss implications for fansubbing, academic work, and professional subtitling based on user feedback.

6. References

[1] N. Nadelman, Personal Interview. 2004.

[2] N. Pappas, Fansub Information Network, "Application and Script Vault," [online document], 2004, [cited 2004 May 7], Available HTTP: http://www.fansubs.info/vault.php

[3] X. Huang, F. Alleva, H.-W. Hon, M.-Y. Hwang, K.-F. Lee, R. Rosenfeld, "The Sphinx-II speech recognition system: an overview (http://sourceofrge.net/projects/cmusphinx/)" *in Computer Speech and Language*, [online document], 1992 Jan. 15, [cited 2004 May 7], Available HTTP: http://citeseer.ist.psu.edu/huang92sphinxii.html

[4] T. Ezzat, G. Geiger, and T. Poggio, "Trainable Videorealistic Speech Animation," *in Proceedings of SIGGRAPH 2002*, [online document], 2002, [cited 2004 May 7], Available HTTP: http://www.ai.mit.edu/projects/cbcl/publications/ ps/siggraph02.pdf

[5] J. Wang, T.-S. Chua, "A Framework for Video Scene Boundary Detection," *in Proceedings of SIGGRAPH 2002*, [online document], 2002, [cited 2004 May 7], Available HTTP: http://portal.acm.org/citation.cfm?id=641055&dl=ACM&coll=GUIDE

[6] Microsoft, "Text Services Framework," *in MSDN Library*, [online document], 2004, [cited 2004 May 7], Available HTTP: http://msdn.microsoft.com/library/default.asp?url=/library/en-us/tsf/tsf/text_services_framework.asp

[7] M. Sawicki, M. Suignard, M. Ishikawa, M. Dürst, T. Texin, "Ruby Annotation," *in W3C Technical Reports and Publications*, 2001 May 31, [cited 2004 May 7], Available HTTP: http://www.w3.org/TR/2001/REC-ruby-20010531/

[8] S. Leonard, "Progress Against the Law: Fan Distribution, Copyright, and the Explosive Growth of Japanese Animation," 1.10, 2004 Apr. 29, [cited 2004 May 10], Available HTTP: http://web.mit.edu/seantek/www/papers/