Post-processing InkML for Random-Access Navigation of Voluminous Handwritten Ink Documents

K. A. Mohamed, L. Belenkaia, Th. Ottmann
Institut für Informatik, Albert-Ludwigs-Universität Freiburg, 79110 Germany
{khaireel, belenki, ottmann}@informatik.uni-freiburg.de

ABSTRACT

The goal of this research is the improvement of browsing voluminous InkML data in two areas: ease of rendering continuous ink-flow for replay-browsing, and ease of random access navigation in eLearning domains. The notion of real-time random access navigation in ink documents has not yet been fully exploited. Users of existing eLearning browsers are restricted to viewing static annotated slides that are inferior in quality when compared to actively replaying the same slides with sequenced ink-flow of the annotated freehand writings. We are developing a tool to investigate ways of managing massive InkML data for efficient "active visible scrolling" of recorded freehand writings in ink documents. This work will also develop and evaluate new post-processing techniques that take advantage of the relationship between ink volumes and active-rendering times for real-time random access navigation.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Query formulation; I.7.5 [**Document Capture**]: Document analysis

Keywords

InkML, Random access, Digital ink, Freehand writing

Introduction

Lecturers typically deliver their lessons through slides prepared from customary softwares like MS PowerPoint and Adobe Acrobat. Essentially, as their classes progress, they will write, highlight, underline, draw and formulate on those slides to put forth their ideas to students. And by opening "empty" slides to spontaneously write on them, they facilitate discussions amongst students, help to illustrate complex mathematical proofs, and demonstrate diagrammatic facts more clearly. Studies show that the combination of handwritten annotations, the development of thoughts through freehand writings, and the verbal explanations of lecturers, are indeed crucial in assisting students' understanding of subjects [2, 1, 4].

Random Access Navigation

Random access navigation on real-time multimedia documents provides an additional advantage on top of the simple *fast-forward* and *rewinding* of real-time information.

Müller and Ottmann [2] define *random accessibility* as the retrieval of all necessary objects, which must be made completely available in an uncompressed or dereferenced way, at any time instance, such that the entire *access time* of these objects is less than the media-specific threshold period τ. Here, this threshold period refers to the hardware-specific duration of accessing raw data (on disk, for example) and rendering them onto the screen. Putting it simply, given any instance of time on the 'active' slider-bar, we must have a complete list of all associated *visible* objects for that time instance to be made available for display on screen – so fast, that users get the impression of *immediacy*.

Representing Digital Ink

W3C's InkML standard [3] allows the description of precise digital ink information that is not only interchangeable between systems, but also allows the convenient 'archival' and 'streaming' of ink data between applications. Amongst many of its properties, it is important to note that InkML provides the means for extension – by allowing application-specific information to be added to the ink files to support issues of compatibility. In addition to the xy coordinate channels, we include a time-stamp "**T**" in the *regularChannels* to register time instances for each sampled coordinate in a *traceFormat*. We post-process the recorded elementary ink data to extract the basic instructions marked up in the file to reconstruct the exact sequence of the original handwritten traces. We then port these instructions into segmented *object modules* to coincide with the actual slider-bar time reference.

Throughout this report, we assume that all recorded ink data are properly sampled in the highest granularity so that the full dynamics of users' writing styles are preserved as per recorded.

Object Modules

Extricating the fundamental information from the InkML file returns a set $S = \{s_0, s_1, ..., s_k\}$, where tuples $s_i = (x_i, y_i, t_i)$ refer to the *regularChannels*' data corresponding to the xy coordinates and time stamps t of the i^{th} sequence. Lines are drawn between two successive tuples s_i and s_{i+1}. This set S is translated onto the slider-bar's *actual* time reference-scale and segments $S_f = \{t_{f0}, t_{f1}, ..., t_{fn}\}$ are built upon it to collectively assemble the aligned s_i tuples (see Figure 1). Each segment t_{fi} is constrained to a period of not more than a pre-defined tolerance of t_σ msec, and may result to an empty segmented content. A segment t_{fi} in S_f is considered "active" if there exist ink data within itself, and is considered "inactive" otherwise.

Our *object modules* are derived from all active segments in S_f to result in $\alpha_f = \{a_0, a_1, ..., a_m\}$. They are primitive Java graphics object that contain the underlying symbolic rendering instructions represented by the segment. Rendering an object module a_j to screen on the Java platform is carried out by executing `g.draw(aj)`.

WWW2004, May 17–22, 2004, New York, New York, USA.
ACM 1-58113-912-8/04/0005.

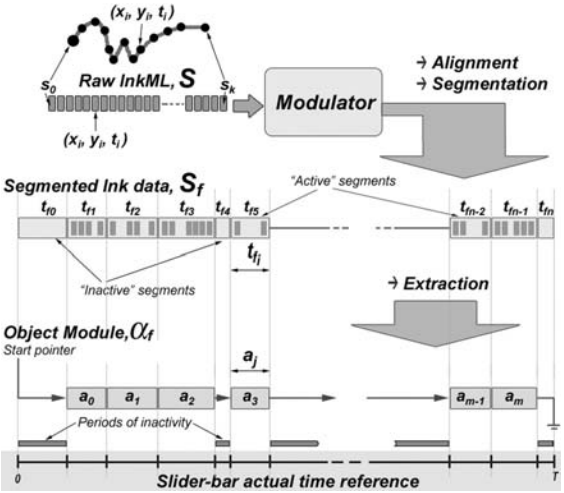

Figure 1: Modulated alignment of the resultant *object modules* α_f to the segmented time scale of the slider-bar.

Applied Object Modules

At query point t_q, we identify the last *complete* object module a_{j-1} and start rendering from module a_0 to a_{j-1} (Figure 2). Afterwhich, the remaining set of data contained in ε_q, is rendered as for normal; reading the InkML instructions contiguously from the file.

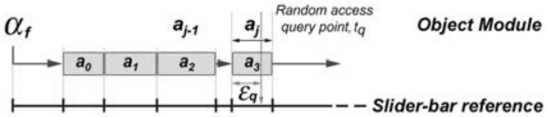

Figure 2: Applied data modules.

Cumulative Object Modules

We define a cumulative property for each module a'_j, such that it collectively stores all previous modules running up to its position. This cumulative property of α'_f is reflected pictorially in Figure 3.

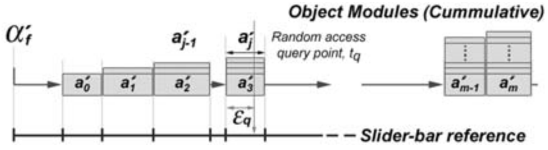

Figure 3: Cumulative data modules.

At query point t_q, we identify the last *complete* module a'_{j-1} and render it straight away (as all associated *visible* objects prior to t_q have already been pre-calculated and contained inside a'_{j-1}). The remaining set of data in ε_q is rendered as for normal.

Discussion of Methods

Data volumes are measured from the total number of coordinates constructed on screen at any stipulated time instance referenced by the slider-bar. When discussing *reaction* times, we refer to the duration the program takes to relay complete rendering instructions to the screen.

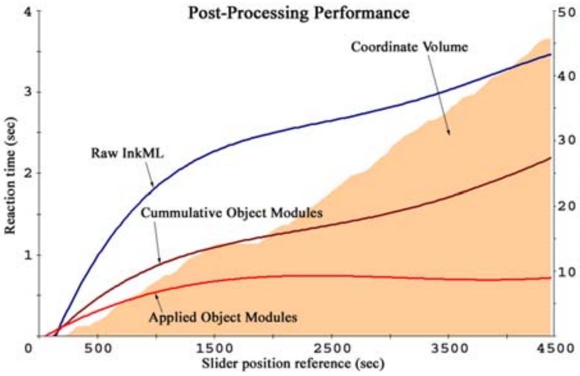

Figure 4: Average reaction time vs data volume.

We anticipated the cumulative data model to supersede the random accessibility performances of the other two models in terms of the speed of access, as this cumulative method of storing multimedia objects is proven to be speed-efficient and sound for retrievals of multi-stream entities used in domains that engage the "random-access replay" facility [2]. However, the results in Figure 4 tells otherwise. For randomly navigating through huge data volumes containing digital ink, the method of *applied object modules* maintained a reaction time of less than 1,000 msec on average.

Conclusion

Of the three models experimented, we find that the *applied object module* is the most efficient method for handling random-access navigation. Its rate of convergence is 3.792 times faster than without post-processing and is 2.029 times faster than the *cumulative module*. Its ability to keep the reaction times below the 1,000 msec mark is a tremendous advantage for the "active visible scrolling" technique in relation to τ for issues of *immediacy*. Through it, we can sustain data volumes of up to 80,000 coordinate-capacity.

Acknowledgments

This research is partially funded by the DFG-Projekt Ot64/8-3 and the BMBF-Projekt FKZ 08NM210.

1. REFERENCES

[1] S. Mukhopadhyay and B. Smith. Passive capture and structuring of lectures. In *Proceedings of the seventh ACM international conference on Multimedia (Part 1)*, pages 477–487. ACM Press, 1999.

[2] R. Müller and T. Ottmann. The "authoring on the fly" system for automated recording and replay of (tele)presentations, Special issue on "multimedia authoring and presentation techniques". *Multimedia Systems*, 8(3):158–176, Oct 2000.

[3] G. Russell, Y.-M. Chee, G. Seni, L. Yaeger, C. Tremblay, K. Franke, S. Madhvanath, and M. Froumentin. Ink markup language [online], Available: http://www.w3.org/tr/inkml/ [Accessed 14 Oct 2003]. *W3C Working Draft*, Aug 2003.

[4] B. Zupancic and H. Horz. Lecture recording and its use in a traditional university course. In *Proceedings of the 7th annual conference on Innovation and technology in computer science education*, pages 24–28. ACM Press, 2002.