

電子テキストのシームレスな表示に関する研究 — システム概要と入力インタフェース —[¶]

杉山正治[§], 生田敦司^{*}, 柴田みゆき^{*}
立命館大学[§], 大谷大学^{*}

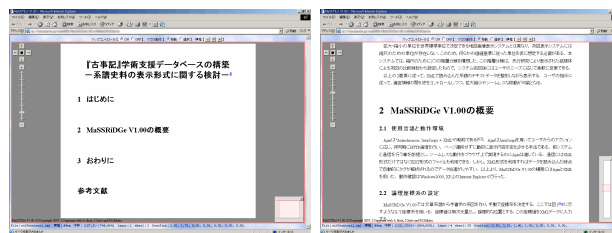
1. はじめに

我々はすでに、Webブラウザによる電子テキストのシームレスな表示方法としてMaSSTeXt (Magnifying And Simplifying System for Text EXtension) を提案した[1],[2]。MaSSTeXtでは文章の階層毎に表示・非表示を伴いながら拡大縮小移動が可能であり、一画面に全てのテキストを格納できる。このため、大量の電子テキストをページ遷移なく閲覧でき、全体を俯瞰しながら細部のテキストを読むことが出来る(図1(a),(b))。さらに、離れた位置にある文章を隣接させる手法としてWaRP (Wheel Action for Reeling Paragraphs) も提案した[3]。これにより単一ウィンドウのみで文章の比較も容易である。一方、これまでのMaSSTeXtはPCでの利用を前提に開発されたため、スマートフォンやタブレットのタッチ入力には対応できていない。そこで、本研究ではMaSSTeXtをタッチ入力に対応させるための問題点を整理するとともに、現在主流となっているHTML5との整合性について再検討を行う。

2. 電子テキストの問題点

電子テキストの利点と、それに対する問題点は次のようにまとめられる。

- (A) 紙面サイズによる制約がないため、自由な字数で記述できる。字数固定のページ概念の利用は紙媒体の模倣であり、電子テキストの利点を狭める。
- (B) 文字列検索・タグ検索が容易である。ただし、該当する文字列が分からない場合、およびタグ付けが行われていない場合は横断検索が困難である。
- (C) 紙媒体に比べ、コンパクトに保管できる。ただし、通信の制約などから一冊の内容が多数のファイルに細かく分割される場合には管理が煩雑である。
- (D) リンク機能により、関連した内容に直ちに遷移できる。ただし、紙媒体と比べて物理的な移動感覚が無いため、読書位置の把握が困難である。
- (E) 全体を俯瞰的に把握しながら詳細な部分を読むのが難しい。さらに、上述(A)および(D)の影響により、テキスト全体の分量の把握は困難である。
- (F) テキストの拡大を画像のように縦横比を維持したまま行くと行頭・行末が見えなくなり、列方向スクロールが必須となる。速読は極めて困難である。
- (G) テキスト俯瞰のために縮小すると字が潰れてしまい、画像を除いて読めるテキストは限定的になる。



(a) 目次の表示 (b) 本文の表示
図1: MaSSTeXtによるシームレスなテキスト表示

3. テキストの階層構造とレイヤの概念

電子テキストを読みやすくするために、MaSSTeXtでは以下の機能を実装することにより解決した。

- (1) 全体を俯瞰する時に必要でない下位の文章構造を非表示にし、上位の文章構造を読める文字サイズで表示して「目次」のような画面を作る(図1(a))。これにより、別途目次のページを作らずに済む。
- (2) (1)で非表示にしていた下位の文章構造を拡大操作によって徐々に表示できるように切り替える。この時、文章の前後方向は文字サイズに応じて大きくなるが、左右方向には行頭と行末が画面からはみ出さないように制限を設けておく(図1(b))。これにより、読者の好みの文字サイズが実現する。
- (3) テキスト全体の俯瞰と部分の詳細な表示をシームレスに行うため、複数のレイヤで表示・非表示を切り替える。このレイヤ構造と文章構造を連携させ、単一の入力インタフェースのみで切り替える。これにより、画面上に表示非表示メニューボタンなどを作らずに済むので配置がシンプルになる。
- (4) 字数固定のページ概念を採用せず、自由な字数で書きたいように記述できるようにする。ファイル上では別ファイルになることがあっても、画面上では一冊の電子テキストを一体化させて表示する。これにより、文章全体の分量が直ちに分かる。
- (5) 読書位置は上位のレイヤに上がれば直ちに把握できる。その結果、紙媒体と同様な感覚で文章全体を行き来できるのでリンクボタンが不要となる。さらに、文章取得に非同期通信を用いればページ遷移が無くなるので、読書位置の混乱が無くなる。

ここで示したレイヤの概念は、文章構造の表示非表示切り替え階層のことであり、地図画像表示システムにおける全体と部分の切り替えと同種のものを、テキスト表示のみで実現するものである。一般に「見出し」と「本文」は別レイヤとしなければならないため、これを実現できる適切なマークアップが必要である。

[¶]Displaying Electronic Text with Seamless Operations

[§]Seiji Sugiyama: Ritsumeikan University

^{*}Atsushi Ikuta, and Miyuki Shibata: Otani University

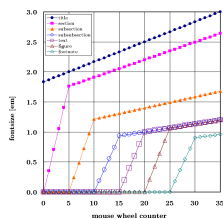
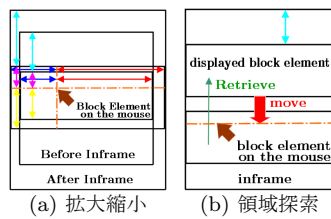


図 2: フォントサイズ曲線



(a) 拡大縮小 (b) 領域探索

4. MaSSTExt

これまでの MaSSTExt の実装方法を示す [1],[2]. PC 上で前章の手法を実装する重要なポイントは2つある.

1つはフォントサイズの拡大縮小倍率を表示非表示のタイミングで切り替えることである. 図 2 にフォントサイズ曲線を示す. 横軸はマウスホイールの回転回数を, 縦軸はフォントサイズを表す. 文章構造の上位下位の違いにより, 拡大縮小時の倍率を別々に設定し, 表示非表示のタイミングで極端に大きな傾きの直線とすることで拡大縮小のシームレスな表現を可能にする.

もう1つは拡大縮小の手順が, 単なる画像の拡大縮小と異なり, 複雑なことである. 一般にマウス位置を中心に拡大縮小が行われるが, 画像ならばマウス位置と画像の領域さえ分かれば1手順で計算が終了する. 一方, 文章の前後方向と左右方向で異なる拡大縮小倍率を採用する場合, 図 3(a) に示すようなマウス位置のブロック要素を特定し, この要素に対して拡大縮小を縦横異なる倍率で計算し, その後に前後の要素を近づける計算が必要である. さらに, 図 3(b) に示すようにマウス位置のブロック要素が非表示になった場合, 前後の非表示でない要素を特定して, マウス位置に近づける計算が必要である. この (a),(b) の手法をフォントサイズ曲線と連動させて初めて前章のレイヤ構造が実現し, シームレスな拡大縮小・表示非表示が可能となる.

5. タッチイベントへの対応

MaSSTExt はスマートフォンやタブレットのタッチ操作が普及する以前に実装された. MaSSTExt をタッチ操作に対応させるには, 以下の問題を全て解決しなければならない.

- iOS ではタッチ・ジェスチャー両イベントがあるのに対し, Android ではタッチイベントしか存在しない. このため, ジェスチャーイベントのひとつであるピンチイン・ピンチアウトをタッチイベントのみを用いて独自に実装しなければならない.
- ピンチイン・ピンチアウトは2本指の各点の座標値の変化からマウスホイール相当のコントロールを実現するものである. 単一の画像を拡大縮小するだけならば2本指の中点を中心に行うだけでよい. 一方, 複数ブロックで異なる操作を必要とするテキストの拡大縮小では, 2本指の各点が中点位置のブロック要素以外に乗る場合が殆どである. このため, ディスプレイ上の指位置の座標から直接イベントを取得することは不可能である. すなわち, テキスト全体から2本指の位置ではなく, 中点位置のブロック要素を特定してからマウスホイール相当の動作を実現しなければならない.

(c) ピンチイン・ピンチアウトによる拡大縮小では一般に scale という倍率が用いられる. しかし, scale が有効になるのは画像の時だけであり, 縦横の拡大縮小倍率の異なるテキスト表示では scale 以外の方法を用いて独自に実装しなければならない.

(d) PC であればマウス操作に Shift キーなどを併用できるため, 2つの要素を隣接させる WaRP 機能などに適用している [3]. しかし, タッチでの Shift キー操作は不可能なため, 3本指操作などの利用し得る別の方法を検討しなければならない.

6. データフォーマット

MaSSTExt は <section> タグの無かった HTML4.01 時代に XHTML 1.0 準拠で開発された. このため, 便宜上, 見出しタグ <h1>~<h6> や段落タグ <p> などにレイヤ番号を設定する方法を採用していた. 最新の HTML5 を見据えたマークアップを考えるならば <section> タグに移行するのが望ましいと考えられる.

ところが, MaSSTExt の表示非表示レイヤ構造では見出しや段落は別扱いとなるため, <section> タグだけでは実装できず, その内部に書かれた見出しタグや段落タグの有無によってレイヤの設定をせざるを得ない. すなわち, <section> タグが無くても現状ではレイヤの設定に支障がない. しかし, <section> タグで文章の階層構造が作られると通常のタグが文章構造の上位下位のどこにあっても良いことになるため, MaSSTExt でのレイヤ設定に支障を来す. この件に関しては今後の HTML の動向を注視しつつ決定しなければならない.

7. おわりに

本研究では, すでに提案してきた MaSSTExt をタッチ入力に対応させるための問題点を整理するとともに, 現在主流となっている HTML5 との整合性について再検討を行った. 今後は, 問題点克服と検証のため, Javascript を用いてタッチイベント対応のプロトタイプを実装する予定である. さらに, マークアップの書式についても検討し, より現実的かつ分かりやすい形式になるように改良していく予定である. また, 一般公開出来るようにシステムを改良していく予定である.

参考文献

- 杉山, 柴田, 生田, 齋藤, 宮下, “文章のシームレスな表示に関する研究—電子テキストの拡大・縮小表示の構想と概要—”, 情報処理学会・第 63 回デジタルドキュメント研究会, 2007-DD-63(6), pp. 37-44, 2007.
- S. Sugiyama, M. Shibata, A. Ikuta, S. Saito and S. Miyashita: “A Study of Displaying Electronic Text with Seamless Pages –Magnifying And Simplifying System for Text EXTension (MaSSTExt)–”, International Journal of Computer Information Systems and Industrial Management Applications (IJCISIM), ISSN: 2150-7988, Vol. 2, pp. 137-145, 2010.
- S. Sugiyama, A. Ikuta and M. Shibata, “A Basic Concept of Displaying Electronic Text with Seamless Pages: Wheel Action for Reeling Paragraphs (WaRP) in the MaSSTExt”, International Conference on Computer Information Systems and Industrial Management Applications (CISIM2010), pp. 222-227, 2010.