

1 行スクロールを基本とした電子書籍ビューア

水口 充^{1,a)} 岡本 芙之^{1,b)}

概要: 電子書籍は、紙媒体のように紙面が固定的でなく、動的に表示を変化させられるという利点がある。我々はこの利点を活かしたスクロール表示によるビューアを実装している。読者はページによって文章を分断されることなく読み進めることができ、また、自動スクロールによりページめくり操作も不要となる。本稿では、このビューアについて、可読性を向上させるための分かち書きやズーム表示および前後のコンテキスト表示を中心に報告する。

1. はじめに

電子書籍が本格的に普及しつつあるが、紙からの移行期と見なすことのできる現状では、紙の本を模した表示と閲覧操作を採用しているのが一般的である。これは、紙と電子の両方の形態で販売したいという出版側の事情と、紙の本に慣れ親しんでいるというユーザ側の事情が背景として挙げられる。

一方、電子媒体ならではのインタラクティブ性を取り入れた電子書籍も見受けられるようになってきた。高度にマルチメディア化された電子書籍の多くは作り込まれたアプリケーションソフトウェアであり、そのもの自体には汎用性が無いが、インタラクションの方法は応用可能なものもあるだろう。また、テキストベースの書籍の閲覧方法に関する提案も増えてきている [2][6][8][14]。

電子書籍の発展を考えると現在はまだ、将来的に定番となりうる表現形態を構成する要素技術となるような新たなコンテンツ表現や操作方法を模索していく段階にあると考える。そこで我々はインタラクティブにスクロール可能な単行のテキストエリアを基本とした表現形態について提案した [9]。今回、この提案に基づいたアプリケーションを実装したので報告する。

2. 単行のスクロールにおける読みやすさ

文章というメディアは本質的に1次元であるので、単純に読解するだけであれば電光掲示板のような無限にスクロールするテキストエリアで問題がないはずであるが、実際には読みやすさに難がある。この理由としては先の提案

で考察したように、以下の要因が考えられる。

- スクロールが自動的に行われる
 - スクロール速度が固定である
 - 見えている範囲が狭く他の部分が参照できない
- 加えて、
- 単語の切れ目が把握しにくい

という問題点も考えられる。つまり、日本語においては句読点や括弧等を除いて文字が等間隔で連続的であるため、スクロール表示のように移動すると語句を一層捉えにくくなるのではないかと予想した。

前者2つの問題点についてはインタラクティブに調整可能なスクロールとすることで解決が図れる。見えている範囲に関しては可能な限り広くすればよいが、どの程度の範囲が必要か不明である。単語の切れ目を把握しやすくするには、分かち書きと呼ばれる、文節ごとに空白を挿入する表示方法が利用できる。分かち書きは平仮名のみ、あるいは少数の漢字のみを含む文章^{*1}に対して使用されることが多い表示方法である。通常の文章においては漢字や約物が文節の切れ目の目印となっていて読解を支援していると考えられるが、移動している文字表示においては分かち書きにより可読性を向上させられる可能性があると考えた。

そこでまず、分かち書きの効果とスクロール表示閲覧時の読み方について調査を行うことにした。

3. スクロール表示における分かち書きの効果と読み方の調査

3.1 実験1：分かち書きの有無

まず、分かち書きの有無による読みやすさの違いを調べた。1024 × 100 ピクセル (使用したディスプレイにおいて

^{*1} 例えば子供向けの文章や、漢字を使用できなかった初期のゲームソフトにおけるメッセージなど。

¹ 京都産業大学コンピュータ理工学部
Kamigamo-motoyama, Kita-ku, Kyoto 6038555, Japan
a) mmina@cse.kyoto-su.ac.jp
b) g1044192@cse.kyoto-su.ac.jp



図 1 実験画面。

Fig. 1 Display of the experiments.

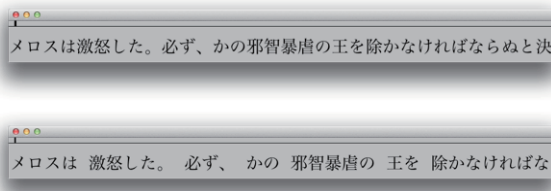


図 2 分かち書き無し (上) / 有り (下) の表示。

Fig. 2 Display without / with separations between segments.

256mm × 25mm) のウィンドウに、32 ピクセル (8mm) の大きさの文字で文章を表示した (図 1)。分かち書き無し条件ではすべての文字を等間隔で、分かち書き有り条件では文節間に 20 ピクセル (5mm) の空白を挿入して表示した (図 2)。スクロール速度はキー操作で調整可能とした。

実験協力者にはそれぞれの条件で最も読みやすくなるようにスクロール速度を調整するよう指示した。以下、このように調整されたスクロール速度を快適速度と呼ぶことにする。また、実験中は視線計測装置 (Tobii グラスアイトラッカ) を装着してもらい、視点を計測した。順序効果を排するために分かち書きの有無の各条件の順序は実験協力者数ごとに変えた。また、記憶による影響を排するために文章中の閲覧箇所は実験ごとに変えるようにした。

実験には「走れメロス」(太宰治著)を使用した。CaboCha[5]を使用して文節ごとに切り分ける前処理を行い、目視にて文節の切れ目が適切であることを確認した。表示文字のフォントは「ヒラギノ明朝 Pro W3」を使用した。

実験協力者は裸眼または矯正された視力を持つ男子大学生 (平均年齢 22.2 歳) 6 名であった。結果は表 1 のとおりである。この結果からは分かち書きの有無に関して、どちらが読みやすいという一般的な傾向は見受けられず、個人に依存すると言える。この理由としては、分かち書きは一般的でないため好みに差が生じたこと、分かち書きの空白量が適切で無かったこと、が考えられる。

注視箇所については、すべての実験協力者は快適速度での閲覧時には右端から 8 文字目を中心としたウィンドウの

表 1 分かち書きの有無と快適速度 [文字/分]

Table 1 Result of best reading speed on the conditions of with/without separations between segments.

実験協力者	分かち書き無し条件	分かち書き有り条件
A	854.6	889.4
B	558.1	593.0
C	593.0	558.1
D	645.3	627.8
E	749.9	680.2
F	453.4	575.5

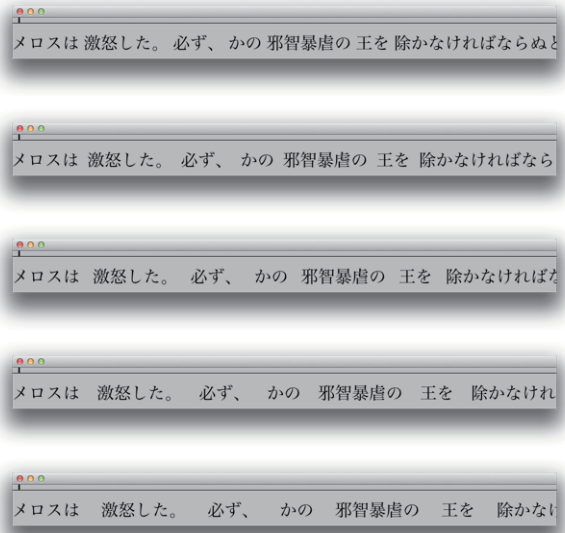


図 3 分かち書きの空白量. 上から 8, 16, 24, 32, 40 ピクセル.

Fig. 3 Separations between segments. 8, 16, 24, 32, 40 pixels respectively.

右半分を見ており、左半分はほとんど見ていなかった。快適速度時には視点を一定の位置からあまり動かさないうまま文章を読み続けることができると言え、この実験での表示条件下では十数文字程度の表示で十分読み進められることを示している。

3.2 実験 2 : 分かち書きの空白量

実験 1 では分かち書きにおける文節間の空白量が適切で無かった可能性がある。そこで、空白量を変えたときに快適速度がどのように変化するかを調査することにした。

文節間の空白量を 8, 16, 24, 32, 40 ピクセル (2, 4, 6, 8, 10mm) の 5 条件とし (図 3)、他は実験 1 と同様にして快適速度となるようにスクロール速度を調整させた。実験協力者は実験 1 と同じ 6 名であった。条件の順序は実験協力者の半数は昇順、半数は降順とした。

結果は表 2 および図 4 のとおりである。一元配置分散分析の結果は空白量の条件間に有意な差はみられなかった。一方、実験協力者ごとの結果を見ると、快適速度が極大となる空白量が存在し、その量は個人ごとに異なっているこ

表 2 分かち書きの量と快適速度 [文字/分]

Table 2 Result of best reading speed for conditions of separations amount between segments.

実験協力者	8px	16px	24px	32px	40px
A	697.6	767.4	854.6	837.1	749.9
B	697.6	784.8	837.1	784.8	749.9
C	530.6	505.8	575.5	662.7	575.5
D	749.9	819.7	749.9	819.7	889.4
E	924.3	1098.7	1063.8	994.1	872.0
F	645.3	697.6	645.3	558.1	558.1

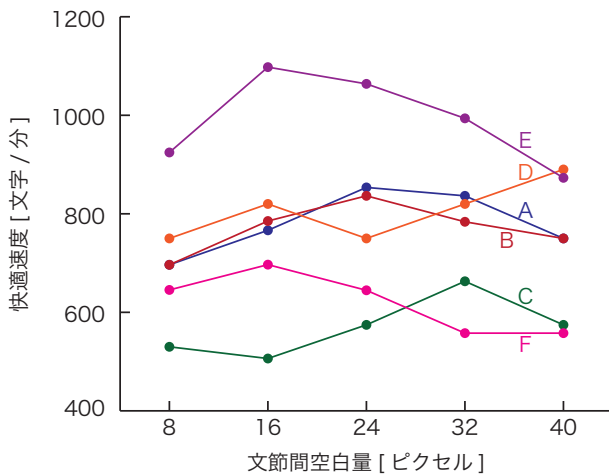


図 4 文節間空白量と快適速度 [文字/分]

Fig. 4 Result of best reading speed for conditions of separations amount between segments.

とが見受けられる。また、多くの実験協力者では実験 1 よりも実験 2 の方が快適速度が増加しており、これは慣れによる影響と思われる。

また、注視箇所はこの実験においても同様に、すべての実験協力者は快適速度での閲覧時には、ウィンドウの右端から 8 文字分を中心とした右半分の領域で文章を読んでいた。

4. スクロールビューアの実装

4.1 1 行のみのスクロールビューア

上述の調査によると、適切な量は個人差があるものの分かち書きによって可読性を向上させられることが分かった。また十数文字の表示で十分読み進められることも示唆された。もちろん、これらの結果は今回の実験条件に依存している可能性もある。例えば、表示に使用するフォントの種類、文字の大きさ、フォント自体に組み込まれている文字間の空白量、などといった要因に影響されることが予想される。

分かち書きに関しては、適切な空白量は個人に依存するので、非使用も含めて設定可能にすれば良いだろう。表示する文字数については、画面の大きさと読みやすい文字の大きさのバランス次第であるが、十数文字程度の表示が確



図 5 1 行スクロールビューア。左:横書きモード。右:縦書きモード。
Fig. 5 Single scroll viewer. Left: horizontal mode. Right: vertical mode.

保できれば良いと考えられる。

また、先の提案 [9] で考察したように、画面中央から離れた文字は小さく表示するズーム表示の利用も考えられる。上述の調査のように、表示する文字数が多くても見ている範囲は限られており、画面の片一方に寄ってしまう。ズーム表示によれば、画面が狭い場合でも読み進めるために必要な文字数を確保でき、画面が広い場合では注視領域を拡大表示部分に誘導できる。

そこでこれらの表示機能を持つ 1 行スクロールビューアを実装した (図 5)。実装は Java を使用し、CaboCha を使用して文節ごとに切り分ける前処理を行った文章データを表示対象とした。

設定に応じて横書き表示と縦書き表示を切り替えられるようにした。Java には文字を縦書き描画する標準の手段が用意されていない。縦書き用グリフを持つフォントであればグリフデータを取り出して描画することも可能であるが、処理が煩雑な上、使用できるフォントに限りが生じる。そこで、縦書き表示によく使われる手法の一つである、特定の文字を回転 (例えば括弧や長音記号など)、あるいは平行移動 (例えば句読点や拗音など) させる方法で実装した。この方法はフォントごとに移動量を決めておく必要がある、表示品質に劣るという欠点があるが、続け字などの横書きに特化したフォントでなければ汎用的に対応できる。処理時間的にも特に気になるほどの遅延は発生していない。

設定に応じて文節間および段落の区切りに空白を挿入して表示する。これらの空白量は設定可能であるが、初期設定では文節間の 20 ピクセルに対し段落の区切りは 100 ピクセルと広く設定した。

自動スクロールはキー操作で開始/停止および速度調整ができる他、マウスでドラッグするとボタンを離したときのカーソルの移動速度を保持してスクロールし続ける (タブレット端末で使用時にはフリック操作に対応する)。

図 6 はズーム表示の例である。このようにウィンドウ中央部を最大サイズとして、両端に近づくほど文字サイズが

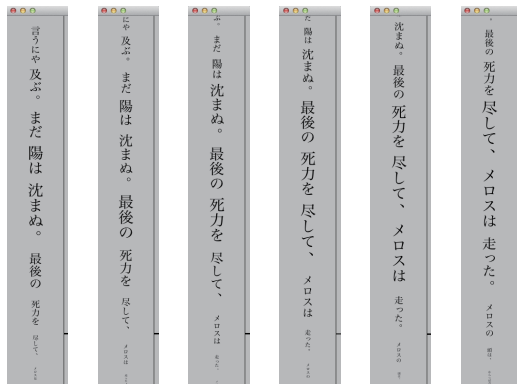


図 6 ズーム表示の例.

Fig. 6 Example of zooming.

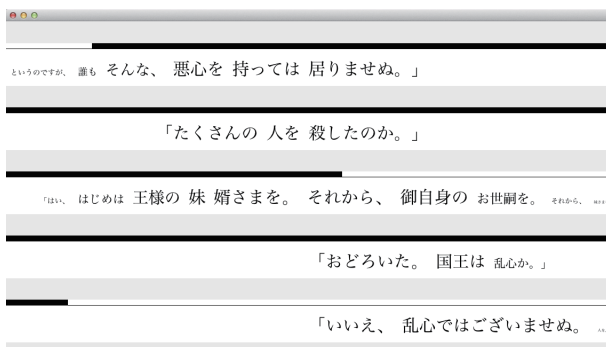


図 7 複数行スクロールビューア (横書きモード).

Fig. 7 Multiple lines scroll viewer, horizontal mode.

小さく表示される。文字の表示サイズは文節単位で同じとなるようにした。

また、文章全体の中の閲覧範囲を視覚化し、閲覧箇所を任意に移動できるように、スクロールバーを設けた。

4.2 複数行のスクロールビューア

1行のみのスクロール表示では文章全体が直列的に表示されることになる。順次読み進めている状態であればこれで問題無いはずであるが、読みの際には前後を参照している可能性もある。

そこで、1つのスクロール領域は1つの段落に対応し、複数段落を同時に扱えるビューアを実装した(図7, 8)。1行のみのスクロールビューアと同様、起動時の設定で横書きと縦書きを切り替えられる。また、同時に表示する段落数も起動時に設定可能とした。

ユーザは上記の1行のみのスクロールビューアと同様の操作でそれぞれの段落の表示位置を変更できるが、自動的なスクロールについては混乱を防ぐために中央のスクロール領域のみ行うようにした。自動スクロール時に表示が段落の最後に達すると、次の段落が中央のスクロール領域となるようにそれぞれのスクロール領域が移動し、その後自動スクロールが再開するようにした。また、ユーザはスクロール領域間の部分を文章に対して垂直方向にドラッグ

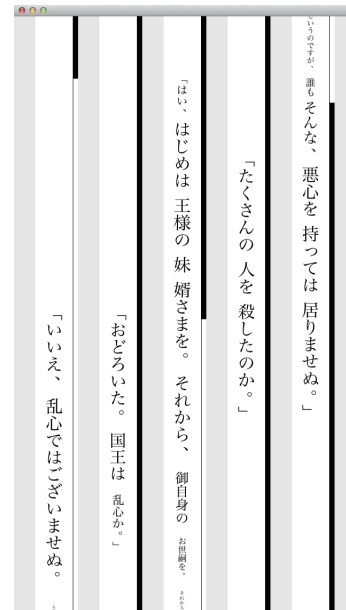


図 8 複数行スクロールビューア (縦書きモード).

Fig. 8 Multiple lines scroll viewer, vertical mode.

することで手動で表示対象となる段落を切り替えられるようにした。

5. 議論

実装したスクロールビューアに関して、操作性や読みやすさといった有効性の実験的評価はまだ行っていない。開発段階でのテストにて分かったことについて以下にまとめる。

5.1 行数

1行のみと複数行を比較すると、集中して読み続けている際には1行でも十分読み進めることができる。一方、複数行の場合は文章の前後を俯瞰でき、また、段落の切れ目が明確に意識付けされるので文章の流れを把握しやすい可能性がある。これは会話が続くような場面などに有効そうであるが、表示の遷移に時間を要するため閲覧のテンポが崩れる恐れもある。

どちらの形態がよいかは好み次第であるが、複数行の表示は1行のみの拡張版と言えるので表示領域が十分であれば複数行が良いと言えるだろう。一方、ウェアラブル端末では視界をあまり遮らない1行のみの方が適しているだろう。

また、表示可能な行数は端末の画面の大きさと文字の大きさによることになる。図7は横1136×縦640ピクセルの例であり、これはiPhone5の画面解像度に相当する。対角10cmの実機上で見ることを想定すると文字の大きさから妥当な行数と言える。が、実験では十数文字程度の表示で読み進められることが示唆されており、横長の画面に縦書きでも必要な文字数を確保できる(図9)。この例では

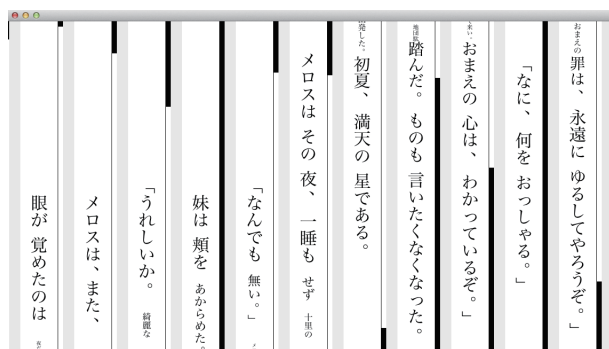


図 9 横長画面に縦書き表示の例.

Fig. 9 Vertical scroll areas on a landscape display.

より広い前後関係を表示できるが、煩雑な印象となる懸念もある。

5.2 読解時間

実験で得られた快適速度は 600~1000 文字/分であった。約 10,000 文字の短編小説であれば 10~17 分程度で読めることになり、注視し続けて最後まで読み切ることが十分可能であろう。試しに、より高速にスクロールさせて読むと、内容の詳細まで捉えるのは難しくなるが、読み飛ばす感覚で閲覧することは可能であった。

中長編となると一度に読み切ることが難しくなるので、中断と再開の手段が必要になる。この場合、単に読んでいた箇所を記録する契機能だけで十分か、あるいは前後の文脈を確認する手段が必要かは調査する必要があるだろう。

5.3 操作感

PC 上でマウスを使用したフリック操作によるスクロール速度調整は、慣れればほぼ希望通りの速度に調整可能であった。

タブレット端末では連続的にフリック操作を行うとスクロール速度が加速するように設定されている [10]。文章を読み進める操作としては加速する必要は無いと考えられるが見たい場所を探し出すためには有効であるので、導入を検討したい。

また、端末の大きさと操作感については、様々な実機上で動作させて検証する必要がある。

5.4 ズームの効果

図 7 や図 8 のように、1 行が長い表示形態の場合はズームの中心に目が行きやすく、視点の誘導効果はあることが予想される。一方、図 9 のように 1 行が短い場合は誘導の必要性は薄い。

また、現在の実装では表示の文字サイズは文節ごとに決定しているため、文節の長さによっては変化が急になり、スクロール時に動きがごちこちなくなることがある。例えば図 9 の右端の行の「ゆるしてやろうぞ」や右から 4 行目の

「言いたくなくなった」は長い行端まで大きな文字サイズのまま表示されている。文節単位での読みを想定してこのような表示としたが、読みやすさに対する影響は今後調査したい。

6. 関連研究

電子書籍の読みやすさ、読みの効率に関する調査は多くなされているが [1], [3], [11], [12], [13], いずれも従来の書籍の形態で表示させたものを対象としている。

スクロールする文字の読みやすさについては、窪田らは横スクロール表示におけるピクセル単位と文字単位のスクロール方法による読みの違いを調査し、ピクセル単位のスクロールの方が文字単位よりも読みやすいことを示している [4]。八木らは横スクロール表示における表示文字数と快適速度の関係を調査し、表示文字数の増加に伴い快適速度も増加すること、特に表示文字数が 5 文字以下の場合に快適速度の増加が顕著であることを明らかにしている [15]。また、快適速度でスクロールされる文章に対しては表示文字数にかかわらず、通常の静止する文章の読みと同程度の意味的理解が可能であることを示している。

分かち書きの効果については、松田は分かち書きによる文節間空白が平仮名文と漢字仮名交り文の読みに及ぼす影響を調査している [7]。これによると文節間空白は統語解析を容易にし眼球運動誘導の手がかりを与えることが示され、特に平仮名文において有効であることが示された。

電子書籍に対する新たなコンテンツ表現の試みは幾つかなされてきている。美馬らのマルチモーダルテキストレイアウトシステムは文字の大きさに揺らぎを与える表現手法である [8]。丸谷らの Yu bi Yomu はなぞった部分を順次表示していくインタラクティブな閲覧形式である [6]。川崎らは文節単位で表示することで視線の移動を排した電子書籍リーダを提案している [2]。武田らは Level of Detail の概念を導入して文章の詳細度を制御できるシステムとコンテンツを提案している [14]。

7. まとめ

スクロール表示による電子書籍ビューアにおいて、可読性を向上させる要素として分かち書きの効果を調査し、有効である可能性を確認した。また、快適速度でスクロールする文字を読んでいる際には視点は固定的で、十数文字程度の表示があれば十分であることを確認した。これらの結果を利用し、分かち書きとズーム表示で可読性の向上を図ったビューアを実装した。今後、実装したビューアについて多様なデバイスやコンテンツでの閲覧を試し、可読性や操作性を評価する予定である。

謝辞 本研究は JSPS 科研費 24603028 の助成を受けたものです。

参考文献

- [1] 今井順子, 面谷信. 文章理解度のディスプレイ上における低下要因の抽出-読みやすい電子ペーパーを目指して. 日本画像学会誌, Vol. 46, No. 2, pp. 90-94, 2007.
- [2] 川崎祐一郎, 中田豊久. ページ送りと改行が無い電子ブックリーダー・インタフェースの開発. インタラクション 2013, 2EXB-16, 2013.
- [3] 小林亮太, 池内淳. 表示媒体が文章理解と記憶に及ぼす影響-電子書籍端末と紙媒体の比較-. 情報処理学会研究報告, 2012-HCI-147(29), pp. 1-7, 2012.
- [4] 窪田悟, 伊藤瑞穂, 岡田想, 小田泰久. 横スクロール文字の可読性. 映像メディア学会誌: 映像情報メディア, Vol. 57, No. 11, pp. 1595-1597, 2003.
- [5] 工藤拓, 松本裕治. チャンキングの段階適用による日本語係り受け解析. 情報処理学会論文誌, Vol. 43, No. 6, pp. 1834-1842, 2002.
- [6] 丸谷和史, 植月美希, 安藤英由樹, 渡邊淳司. ユーザのなぞり動作に基づく動的文章表示方式. 情報処理学会論文誌, Vol. 54, No. 4, pp. 1507-1517, 2013.
- [7] 松田真幸. 日本語文の読みに及ぼす文節間空白の影響. 基礎心理学研究, Vol. 19, No. 2, pp. 83-92, 2001.
- [8] 美馬義亮, 柳英克, 木村健一. マルチモーダルテキストレイアウトシステム. エンタテインメントコンピューティング 2011, 03B-05, 2011.
- [9] 水口充. 電子書籍時代のタイポグラフィ技法に関する考察. 情報処理研究報告, 2012-EC-26(1), pp. 1-4, 2012.
- [10] Philip Quinn, Sylvain Malacria, Andy Cockburn. Touch scrolling transfer functions. in Proceedings of the 26th annual ACM symposium on User interface and technology (UIST '13), pp. 61-70, 2013.
- [11] 柴田博仁, 大村賢悟. 文書の移動・配置における紙の効果: 複数文書を用いた相互参照の読みにおける紙と電子メディアの比較. ヒューマンインタフェース学会論文誌, Vol. 21, No. 3, pp.103-113, 2010.
- [12] 柴田博仁, 大村賢悟. ページ間の行き来を伴う読みにおける紙と電子メディアの比較. ヒューマンインタフェース学会論文誌, Vol. 13, No. 4, pp. 63-74, 2011.
- [13] 高野健太郎, 柴田博仁, 大村賢悟. ページめくりの操作性に着目した電子書籍端末の評価. ヒューマンインタフェース学会論文誌, Vol. 14, No. 1, pp. 89-100, 2012.
- [14] 武田港, 望月茂徳, 川村健一郎. Text Level Of Detail: ズーム動作と階層構造による新しい読書体験のための文章システム及びコンテンツの提案. インタラクション 2013, 3EXB-32, 2013.
- [15] 八木善彦, 菊池正. スクロール表示された文章の読み特性. 心理学研究, Vol. 81, No. 4, pp. 388-396, 2010.