移動速度に応じた自動ズーミングによる効率的ナビゲーション

Speed-dependent Automatic Zooming for Efficient Document Navigation

五十嵐 健夫　Ken Hinckley*

Summary. We propose a navigation technique for browsing large documents that integrates rate-based scrolling with automatic zooming. The view automatically zooms out when the user scrolls rapidly so that the perceptual scrolling speed in screen space remains constant. As a result, the user can efficiently and smoothly navigate through a large document without becoming disoriented by extremely fast visual flow. By incorporating semantic zooming techniques, the user can smoothly access a global overview of the document during rate-based scrolling. We implemented several prototype systems, including a web browser, map viewer, image browser, and dictionary viewer. An informal usability study suggests that automatic zooming can be a helpful alternative to traditional scrolling when the zoomed out view provides appropriate visual cues.

1　はじめに

情報空間のナビゲーションのための基本手法として、表示位置を動的に制御するスクロールと表示の拡大縮小を行うズーミングがあげられる。これらは、限定されたサイズのスクリーン上で膨大な情報にアクセスするために必要不可欠な基本的な技術であるが、既存のインタフェースにはいくつか問題点が認められる。

まず、スクロールには通常スクロールバーが使用されるが、わざわざカーソルを画面の端へもっていって小さいつまみをつかまなければならない他、文章が長い場合にはつまみの微小な移動で表示内容が急激に変化するといった問題がある。これに変わる手法として、マイクロソフトのホイールマウスのようにスクロールの速度を連続的に調整することによって移動を実現する方法がある[2]。この方法は、カーソルをスクロールバーまで移動させる必要が無い点が優れているが、移動速度を上げすぎると表示に目がついていかなくなってしまうという問題がある(図１)。一方、ズーミングに関しては、プルダウンメニューから選ぶ方式や Pad のように連続的に変化させるものがあるが、いずれにしても、ユーザがスクロール操作とは別に明示的に操作しなくてはならず､負担が大きく混乱を招きやすい。

* Takeo Igarashi, 東京大学 / Ken Hinckley, Microsoft Research

図 1: 速度制御によるスクロールの問題点。移動速度が速すぎると、表示に目が追いつかなくなって制御不能となる。

本稿では、このような問題点を解決する手法として、速度によるスクロール操作において、速度に応じてズームレベルを自動的に調整するインタフェースを紹介する。ユーザが移動速度を上げると、自動的にズームアウトがおこり、スクリーン上でのみかけの移動速度が一定に保たれる。逆に、ユーザが速度を落として停止すると自動的にズームインがおこり、文書の詳細が提示される。

2 関連研究

連続的ズーミングを導入し、ズーミング操作をインタフェースの主要な要素としたシステムとして、Pad[10]等がある。これらのシステムでは、広大な情報空間を効果的に提示する手法として、セマンティックズーミングやポータルといったものを提供している。しかし、ズーミングインタフェースについては一般的に「自分や目的地の場所が把握できなくなる」という問題が指摘されている[5]。本稿で提案する手法は、ズーミングとスクローリングの間に自然な制約を与えることで、この問題を緩和している。

広大な情報空間を効率的に提示する手法として、多くのインフォメーションビジュアリゼーション技術が提案されている。特に、注目されている領域を拡大表示しながら、表示空間を歪めることで周辺の情報も同時に小さく表示する Focus and Context 技法が効果的に使われている[3] [7]。我々の手法は、歪みの少ない自然な表示を保ちながら、インタラクションの内容を工夫することによって、膨大な情報への効率的なアクセスを実現している。

３次元空間でのナビゲーションに関して、ズームレベルに応じて移動速度を自動的に調整するという方法が提案されている[11]。この方法では、画面座標系でのカメラから目的となるオブジェクト表面までの距離に比例して速度を調整すると

いうもので、画面上でのみかけの距離が等しければズームレベルに関わらず対象への到達時間が常に同じに保たれる。また、Point of View　ナビゲーションといわれる手法[6]では、目的地をクリックするとそこへジャンプするといった移動の際に、移動速度が目的地への距離の対数値になるように調整される。

大規模な情報にアクセスするために、スクロールバーにおける移動量を調整可能とする手法が提案されている[1][9]。また、スクロールとセマンティックズーミングを組合せた手法なども提案されている[8]が、ズーミングとスクロール操作は独立して操作される。

3　アルゴリズム

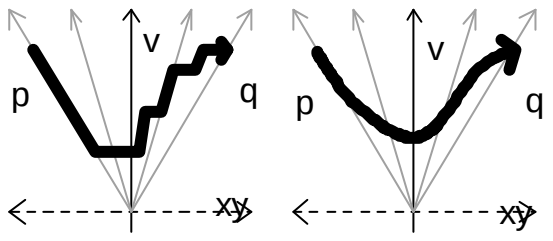
3.1 基本的なアイデア

基本的なアイデアは、スクロール速度に合わせて自動的にズームレベルを調整するというもので、高速でスクロールしているときには縮小表示、低速でスクロールしているときには拡大表示になるように調整する。これは、「早く移動しているときには目的地が遠くにあるということだから大局的な視点が必要になる」という直感的な理解に基づいている。この関係を数式で表すと、

```
scale = constant / speed                    (1)
```

となる（速度が一定値以下のときは `scale=1`）。これによって、情報空間中での実際の移動速度に関わらず、画面上での見かけの速度が一定となる。

自動的なズーミングの第一の目的は、移動速度が上がったときに表示の流れる速度が早くなりすぎて制御不能になることを防ぐことであるが、同時に、詳細な拡大表示と広範な縮小表示を行き来する際に、明示的な操作を必要としなくて済むといった利点も挙げられる。また、通常の明示的なズーミングインタフェースと比較した場合、本手法の効果は、Scale-space 図[4]におけるより滑らかな軌跡として説明できる。すなわち、通常はズーム操作とスクロール操作を交互に操作しなくてはならないために、Scale-space 図における軌跡はぎくしゃくしたものとなるが、我々の手法では、滑らかな曲線となり、よりスムーズな移動であることが示される(図 2)。



a)　Manual zooming/panning.　b) Automatic zooming.

図 2: Space-scale 図による表現。自動ズーミングによって、よりスムー

ズで効率のよい移動が実現されている。(v=scale, xy=space, p=出発点, q=目的地)

3.2 実装の詳細

上記のアイデアをそのままの形で実際のインタフェースとして利用するにはいくつか問題があり、細かい点でいくつか配慮が必要となる。ここでは２つの点について、問題点と解決方法を述べる。

最初の問題は、通常の速度制御スクロールと同様に、速度を入力デバイスの入力値（マウスの移動幅）に比例するように計算したあと、(1)の式によってズームレベルを計算することによって発生する。こうすると、ユーザが速度を上げていったときに、最初に急激なズームアウトがおこり後半変化が緩やかになるという不自然なズーム変化を経験する（図 3)。つまり、この方法ではズームレベルが入力値に反比例することとなり、同じだけマウスを動かした時に、はじめの方では大きさが半分に減るのに、後半では大きさがほとんど変わらないといったことがおきる。そこで、実装にあたっては一定割合でのズームを実現するため、ズームレベルを入力値の累乗としてまず計算し、その後式(1)にしたがって速度を計算するという手法を取った。数式で表現すると以下のようになる。

$$scale = s0^{(dy-d0)(d1-d0)} \qquad (2)$$

ここで、d0,d1,s0 は固定パラメータ、dy が入力値である。結果として入力値と速度との間の直感的な比例関係は失われるが、ズームの割合が一定になるために非常に自然な動作として感じられる。

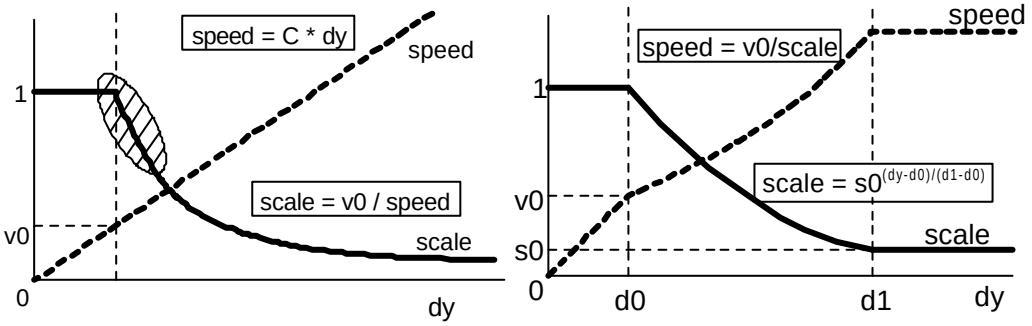


図 3: 入力と速度・スケールの関係図。左側が改良前、右側が改良後。改良前では、斜線部において急激なズームが起こり不自然であった。

もう一つの問題は、マウスボタンを離したときに移動速度が突然０になるため、画面表示の急激な拡大が起こることである。また、移動の途中で逆向きに方向転換をすると、必然的に一瞬速度が０になるためやはり画面表示が一時的に拡大されてしまう。いずれにしても、意図しない急激な変化であり、ユーザの混乱を招く。この問題に対しては、ズームイン時にズーム変化速度に上限を設けることで

対処できる。すなわち、マウスを放したり、移動速度が急激に下がったときには、ズームレベルが徐々に目標ズームレベルに近づいていくような緩衝効果をいれる(図 4)。これによって、(1)の関係は一時的に満たされなくなるが、ズームイン時の緩衝効果では本来のよりも縮小された表示がなされるだけであり、画面上のみかけの移動速度が本来のものよりも大きくなることはない(逆にズームアウト時に緩衝効果をいれるとみかけの移動速度が大きくなりすぎるので導入するべきでない)。

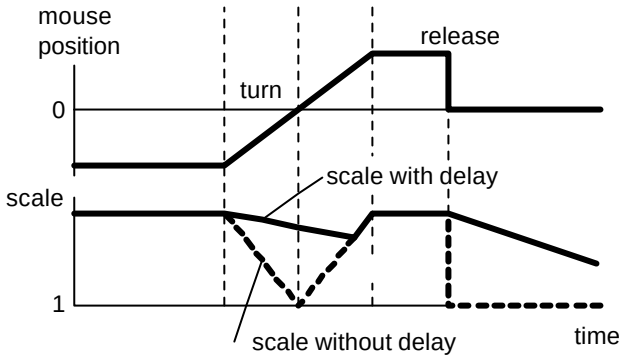


図 4: ズームイン時における、干渉効果。移動方向を変えたり、マウスを放したりしたときに､ゆっくりとズームインが起こるようにしている。

4 アプリケーション例

ここでは、本手法の有効性を確認するために実装した、具体的アプリケーションを想定したプロトタイプについて紹介する。

4.1 ウェブブラウザ

Html で記述されたウェブページはその構造上、だらだらと長いページができやすく、ブラウジング時にはスクロール作業が必須である。しかし、長いページの閲覧にスクロールバーは使いにくく、より適切なインタフェースが求められている。我々は、簡単なウェブブラウザ上に自動ズーミング機能を実装し、有効性を調べている。マイクロソフトのホイールマウスを利用したものと同様に、マウスのボタンを押して上下に動かすことでスクロール速度が制御される｡具体的には、マウスボタンを押すと、図５のようなスライダが表れる。スライダの中心には、ズームを起こさない緩衝帯があるが、それを過ぎるとズームアウトが始まる。ズーミングに際しては、タイトルやイメージ部がズームアウトするにしたがって目立ってくるような､セマンティックズーミングを実現している｡マウスを放すと、徐々にもとの大きさに戻ってくる。セマンティックズーミングによって、速度を上げたときに、複数の章からなる文章の全体構造が見えるようになり、目的の位置を発見することが容易になっている。簡単なユーザテストでは、スムーズに全体構造が把握できること、余分な操作が必要無いことなどに関して、特にポジティブな反応を得ている。

a) Static view　b) Scrolling slowly　c) Scrolling fast d) Scrolling very fast

図 5: 自動ズーミングによるウェブブラウザ

同じように、長いテキスト上での移動が必要となる応用例として、ワープロやプログラムコードのエディタなどが考えられる。

4.2 地図ビューワ

２次元空間上での移動の例として、地図ビューワを実装した(図 6)。ウェブブラウザと異なり、ユーザは上下左右に移動することができる。移動速度が上がるにつれて、ズームアウトが起こり、マウスを放すともとの大きさに自動的に戻る。２次元での移動は特にマウスでの操作が困難と考えられるため、ジョイスティックの利用についてもテストを行った[12]。ジョイスティックの利用に関しては、レバーの角度によって速度を微妙に制御するといったことが通常あまり行われないために、ユーザがいきなり一番端までレバーを倒し、そのために急激なズームアウトがおきてしまう、という現象が観察された。しかし、レバーを微妙に調整すればよいと理解した後はスムーズに操作できていた。

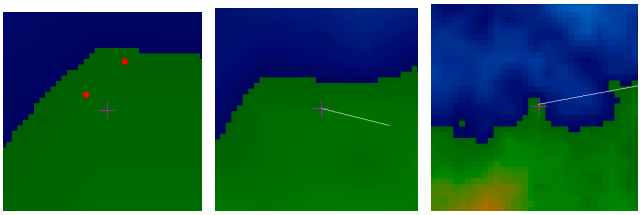


図 6: 自動ズーミングによる地図ビューワ。右がズームアウト時。

地図と同様に、広大な２次元空間を移動する必要のある応用例としては、ＣＡＤシステムやフォトショップのようなイメージエディタ、あるいは表計算プログラムなどが挙げられる。

4.3 イメージブラウザ

デジタルカメラで取りためた写真など、多くの画像ファイルを閲覧するためのブラウザを実装した(図 7)。自動ズーミングによって、早く移動したときにでも流れを把握することが可能となるが、上の２例のようなズームアウト時に使用できる

アブストラクションが存在せず個別の画像を認識しなければいけないため、極端なズームアウトは不可能である。テストした感じでは、多くの画像の中から写真を探し出すときには小さなサムネイルをグリッドの静止画として並べた方が目への負担がすくないと思われるが､特定の画像の間をいったりきたりする場合には、連続的な移動が可能な本手法が有効であると思われる。

a) Static view　b) Scrolling slowly　c) Scrolling fast　d) Scrolling very fast

図 7: 自動ズーミングによるイメージブラウザ

4.4 その他

文字情報を検索する例として英単語辞書なども実装した。ズームアウトするにしたがって、表示する単語を間引きすることによって見かけのスクロール速度を一定に保つことができる。しかし、ズームアウト時に目的とする単語がどこにあるのかを判断するためには、常にアルファベットの順番を頭の中で処理して判断しなくてはならず、負担が大きく使いくいことが判明した。他に、ズーミングとスクロール操作を頻繁に行うアプリケーションの例として波形表示を伴う音声ファイルプレイヤーを実装したが、これもズームアウト時の波形から視覚的に目的位置を判断するのが難しく、有効性が低いと思われる。

5　ユーザテスト

5.1 実験内容

本手法の有効性を確認するため、簡単なユーザスタディを行った。被験者は研究所外部で募集された 7 人で、年齢は 10 代から 50 代程度、マウスやキーボードによる基本的な操作には慣れている。アプリケーションとして、ウェブブラウザと地図ビューワを使用した。ウェブブラウザでは、通常のスクロールバーによる方法と、我々の提案手法を比較した。地図ビューワにおいては、ジョイスティックによる操作とし、ボタンによる明示的ズーミングと提案手法とを比較した。ウェブブラウザにおいては、図 8 左のように、主画面の横に提示されている章中の絵を見つけ出してクリックするというタスクが与えられ､地図ビューワにおいては、図 8 右のように画面右下のレーダで示された目的地へ移動してクリックするというタスクが与えられた。双方において、一連のタスク終了までの時間を記録した他、インタフェースとしてどちらを好むかという主観的評価を調べた。

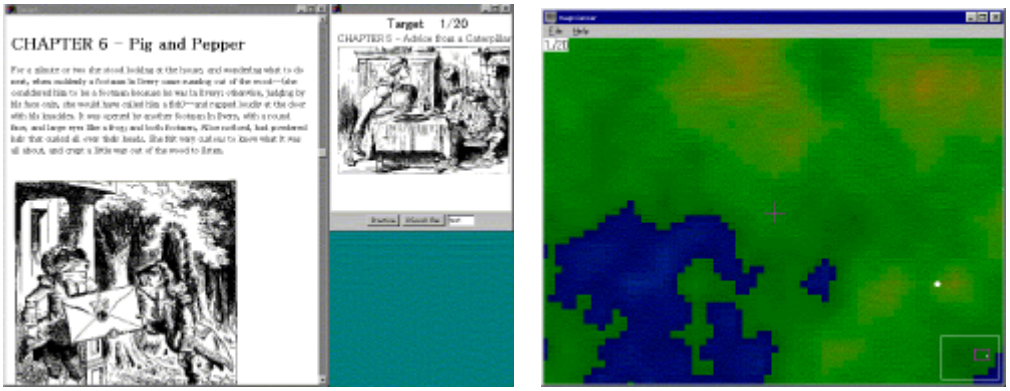


図 8: 実験で使用したシステムの画面例

5.2 結果

図 9 にウェブブラウザにおいてタスク終了までにかかった時間の総計をしめす。被験者間のばらつきが大きいが、平均するとほとんど変わらない結果となっている。一方、主観的評価を尋ねたところ、ほとんどの被験者が自動ズーミングの方がよいと答えている。具体的には、ズームアウトによって文章の全体像が把握できる点がよいという回答が多かった。

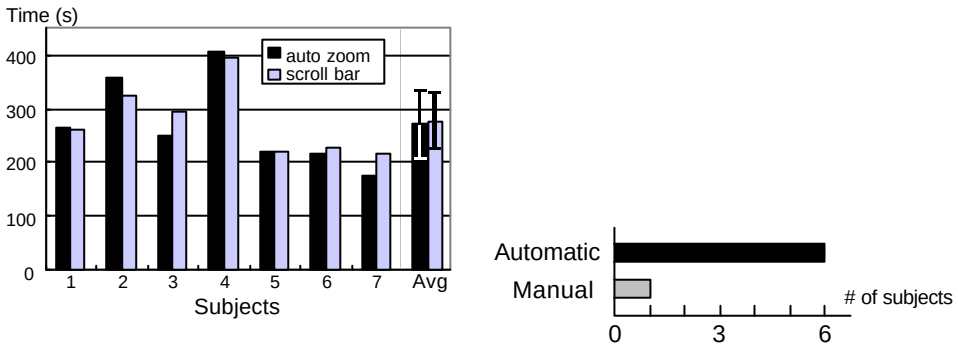


図 9: ウェブブラウザの実験結果。
左がタスク遂行時間で、右がニ者択一での主観的評価の結果。

図 10 に地図ビューワにおける結果を示す。ウェブブラウザの時とくらべよりばらつきが大きい。平均では、明示的なズーミングの方が微妙に早い結果となっている。主観的評価は完全に二つに割れている。すなわち、ゲームなどになれている被験者は提案手法が非常に使いやすいと答えていたが、高齢の被験者などは目がちらちらして操作しにくいと答えていた。また、操作を観察していたところ、速度の微調整がうまくできずに目的地の近くでぐるぐる回ってしまう現象が見られた。

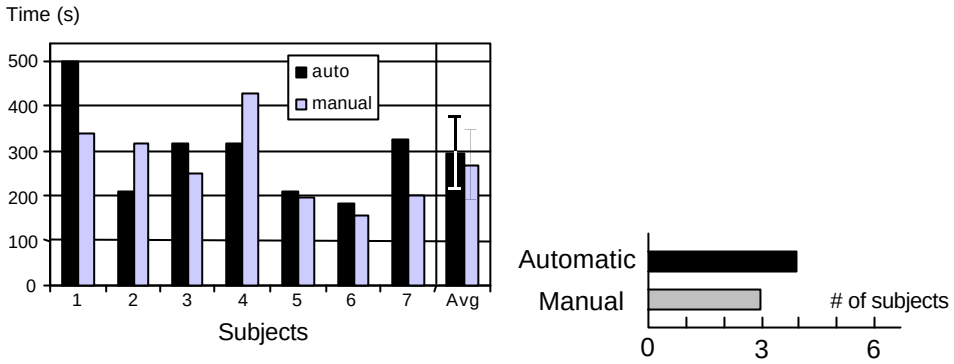


図 10: 地図ビューワの実験結果。
左がタスク遂行時間で、右がニ者択一での主観的評価の結果。

実験については、具体的なパフォーマンスの比較よりも、ユーザの操作の様子を観察することによって、本手法が実用に耐えうるかという点を確認し、問題点等を明らかにすることを第一目的とした。操作性に関しては、地図ビューワに関して課題が見られたものの、ウェブブラウザにおいてはほとんどすべての被験者が十分スムーズに操作している様子が確認された。特に、ゲームのようなダイナミックな操作に慣れているユーザに非常に好まれることが確認された。また、主観的評価に現れているように、アプリケーションによっては、本手法が既存手法に変わる選択肢の一つとして有効だといえよう。

6 まとめ

ユーザの制御するスクロール速度に応じて、自動的にズーミングを行うインタフェースについて紹介した。本手法によって、速度制御によるスクロールの問題点であった極端に早い表示の流れが解消され、大規模な情報空間中において拡大表示と縮小表示を効率良く行き来するようなナビゲーションが実現される。本手法は、ズーミングとスクロールを伴う広範なアプリケーションに適用可能であり、具体例としてウェブブラウザや地図ビューワなどを実装した。さらに簡単なユーザテストを行った結果、ゲームなどに慣れている被験者が効率的に操作できる反面、高齢者などには負担が大きいことが観察された。

参考文献

[1] Ahlberg, C. Shneiderman, B., The alphaslider: a compact and rapid selector, CHI'94 conference proceedings, pp.365-371,1994.

[2] Barrett, R.C., Sleker, E.J., Rutledge, J.D., Olyha, R.S. The Negative Inertia: A dynamic pointing function, CHI'95 conference companion, pp. 316-317, 1995.

[3] Furnas, G.W. Generalized Fisheye Views, Proceedings of CHI'86, pp. 16-23, 1986.

[4] Furnas, G.W., Bederson, B.B. Space-Scale Diagrams: Understanding Multiscale Interfaces. Proceedings of CHI'95, pp. 234-241, 1995.

[5] Jul, S., Furnas, G., Critical Zones in Desert Fog: Aids to Multiscale Navigation, Proceedings of UIST'98, pp. 97-106, 1998.

[6] Mackinlay, J.D., Card, C.K., Robertson, G.G., Rapid Controlled Movement Through a Virtual 3D Workspace, SIGGRAPH 90, pp. 171-176, 1990.

[7] Mackinlay, J.D., Robertson, G.G., Card, C.K. The Perspective Wall: Detail and Context Smoothly Integrated. Proceedings of CHI'91, pp. 173-179, 1991.

[8] Masui,T. LensBar - Visualization for Browsing and Filtering Large Lists of Data. In Proceedings of InfoVis'98, pp.113-120, 1998.

[9] Masui,T., Kashiwagi,K., Borden,G.R., Elastic graphical interfaces for precise data manipulation. CHI'95 Conference Companion, pp. 143-144, 1995

[10]Perlin, K., Fox,D. Pad: An Alternative Approach to the Computer Interface, SIGGRAPH 93, pp. 57-64, 1993.

[11]Ware, C., Fleet, D., Context Sensitive Flying Interface, 1997 Symposium on Interactive 3D Graphics, pp. 127-130, 1997.

[12]Zhai, S., Smith, B.A., Selker, T., Improving Browsing Performance: A Study of Four Input Devices for Scrolling and Pointing Tasks, INTERACT'97, pp. 286-292, 1997.