

VolGrab: 空中操作による三次元画面移動インターフェース

関口 峻^{1,a)} 小室 孝^{1,b)}

概要: 本稿は、既存のタッチインターフェースと空中操作によるズームやスクロールなどの三次元画面移動を組み合わせたユーザーインターフェースである VolGrab を提案する。既存のタッチインターフェースは平面的な操作しかできないが空中操作では操作の軸が増えた三次元操作が可能であるため、画面に対して垂直方向の動きをズームに平行方向の動きをスクロールに割り当てることによってズーム操作とスクロール操作を統合したスムーズな画面移動が可能となる。VolGrab による地図アプリケーションを実装し、その有効性を確認した。

VolGrab: Realizing Direct 3D Navigation by Aerial Gestures

SEKIGUCHI SHUN^{1,a)} KOMURO TAKASHI^{1,b)}

Abstract: In this paper, we propose VolGrab, a user interface that combines the existing touch interface with 3D navigation operation such as zooming and scrolling. While the touch interface enables only planar operation, aerial operation enables 3D operation with a larger number of axes, and enables smooth navigation that is integrated with zooming and scrolling by assigning the direction that is normal to the screen to zooming and the direction that is parallel to the screen to scrolling. We implemented a map application based on the proposed interface, and confirmed the effectiveness of the proposed interface.

1. はじめに

現在、PC やスマートフォンなどタッチスクリーンを用いた機器が広く一般的に使われている。さまざまなサイズのディスプレイがあるが、一つの画面領域に表示できる内容には限界があるので画面領域に表示されているドキュメントや画像などの大きさや表示される領域を状況に応じて変更したいというニーズがある。それに対し、例えばズームやスクロールなどの画面移動が用いられている。タッチスクリーンにおける画面移動の方法として両手を使ってタッチスクリーンをタッチしてズームする領域を指定する方法 [1] やズームするタッチスクリーンをこすることでズーム操作をする方法 [2] などが提案されているが、現在では Google Maps のようにタッチスクリーンでのドラッグやスワイプによるスクロールとピンチによるズームを用

いたものが一般的になっている。また、タッチスクリーンを用いない画面移動に関する研究としてマウスのボタンやマウスカーソルの位置を用いたもの [3–5] や空中操作を用いたもの [6, 7] などがある。ユーザが画面移動をしやすいように画面移動時の表示内容を変更する研究も存在しており、スクロールの速度に応じてズーム率を変更するもの [8] などがある。タッチインターフェースを拡張する研究として、タッチスクリーンの上に IR 式の外付けタッチ操作取得システムを設置することでスクリーンレイヤーと IR レイヤーからなるマルチレイヤーによる操作を実現するもの [9] や高さの異なる 3 つのレーザー平面を用いてタッチ時の指先の姿勢を取得するもの [10] などがある。

近年ではタッチインターフェースに空中操作を組み合わせたユーザーインターフェース (UI) が提案されている。既存の画面手法の拡張として既存のタッチインターフェースに手指の空中でのジェスチャなどによる操作をズームやスクロールの操作に対応付けたもの [11–13]、モーションキャプチャを用いて取得した空中での手指の操作とタッチスクリーンを組み合わせた操作の方法を提案したもの [14]

¹ 埼玉大学 埼玉県さいたま市桜区下大久保 255
Saitama University 255 Shimo-Okubo, Sakura-ku, Saitama
City, 338-8570, Japan

a) sekiguchi@is.ics.saitama-u.ac.jp

b) komuro@mail.saitama-u.ac.jp

などがあるが、ズームやスクロールの操作をそれぞれ独立したジェスチャコマンドとして割り当てているものが多く、ズームやスクロールの操作を同時に行うことが難しいという課題がある。

そこで、本研究では既存のタッチインターフェースと空中操作によるズームやスクロールなどの三次元画面移動を組み合わせた UI として VolGrab を提案する。ズーム操作とスクロール操作はそれぞれ頻繁に行われることが多いので、三次元画面移動によってこれらを同時に行うことができるようにすることで、スムーズに画面移動することが可能となる。既存のタッチインターフェースと三次元画面移動を組み合わせることで、既存のタッチインターフェースでできることをすべて含み、かつ拡張機能として三次元画面移動を利用することができるため、スムーズな画面移動によって既存のタッチインターフェースより操作しやすくなることが期待される。

2. VolGrab

本研究では空中操作による三次元画面移動を可能とする UI である VolGrab を提案する。既存のタッチインターフェースは平面的な操作しかできないが空中操作では操作の軸が増えた三次元操作が可能であるため、画面に対して垂直方向の動きをズームに平行方向の動きをスクロールに割り当てることによってズーム操作とスクロール操作を統合したスムーズな画面移動が可能となる。VolGrab は既存のタッチインターフェースに三次元画面移動を可能とする空中操作を加えたインターフェースである。VolGrab では手がつかむ姿勢をしていれば空中操作に応じて画面移動を行い、そうでなければ画面移動を行わない。これによってマウスにおけるドラッグやタッチスクリーンにおけるスワイプと同様の操作が空中で可能になる。すなわちつかんで離すという動作を繰り返すことで複数回に渡るスクロールをすることができる。また、つかみ姿勢以外では画面移動が行われないことによって、既存のタッチインターフェースに干渉せずに空中操作による画面移動をすることができる。すなわち空中操作によって画面移動操作をした後のビューを維持したままそのビューにおいて従来通りのタッチによる選択操作をすることが可能となる。

2.1 空中操作による三次元画面移動

VolGrab はズームやスクロールといった画面移動を空中操作によって行う UI である。この UI では図 1 に示すような角錐台の体積体を手で掴んで動かすことで画面移動を可能にする。この体積体はスケールの異なる画像を積み重ねた仮想的なものでありインフォメーションボリュームと呼ぶ。インフォメーションボリュームは手で掴んだ状態で動かすことで手の位置に応じて平行移動させることができる。インフォメーションボリュームの中の赤い枠で切り取

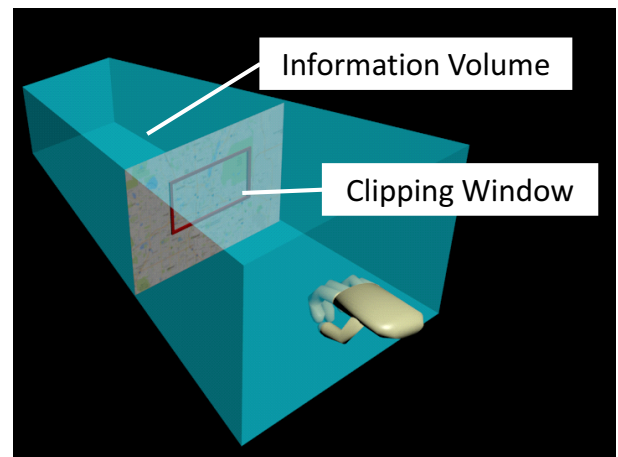


図 1 VolGrab の概念図

Fig. 1 Concept of VolGrab

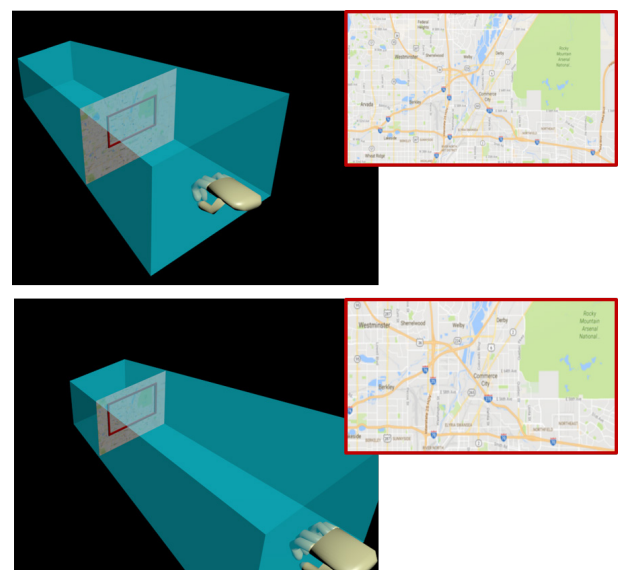


図 2 VolGrab の操作

Fig. 2 Operation of VolGrab

られた部分が画面に表示される。この赤い枠をクリッピングウィンドウと呼ぶ。インフォメーションボリュームは画像を積み重ねられて作られていて、例えば様々な拡大率で拡大された画像によって作ることによってクリッピングウィンドウに対してインフォメーションボリュームの垂直方向の動きがズーム、平行方向の動きがスクロールに対応付けられる。インフォメーションボリュームを空間内で動かすことでズームとスクロールを統合して行うことができる。

図 2 と図 3 は空中操作を行う様子を示している。図 2 ではインフォメーションボリュームをつかんで画面移動操作を行う様子を表している。一方、図 3 は画面に対して空中操作を行う様子を示している。図 3 の画面上の丸はポイントであり手から画面に直交する位置に描画される。図 3 左は手がつかみ姿勢ではないのでポイントのみが動くが、図 3 右ではつかみ姿勢なのでポイントとともに画面がスクロールしポイントの位置を拡大中心としてズームしている。

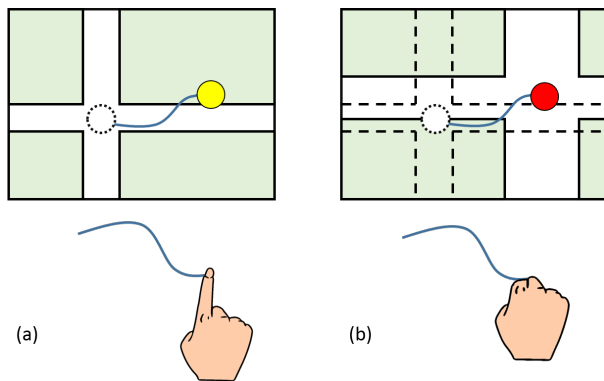


図 3 三次元画面移動の方法: (a) 非つかみ姿勢, (b) つかみ姿勢
Fig. 3 3D navigation method
(a)Non-grabbing posture, (b)Grabbing posture

図 1 においてインフォメーションボリュームの形は角錐台であるので拡大率の変化は手の動きに比例して変化するが、インフォメーションボリュームを指数的にスケールの異なる画像によって構成することで指数的なズームも可能となる。また、スケールだけでなく詳細度の異なる画像からインフォメーションボリュームを構成することで、地図などのアプリケーションにおいてスケールに応じて表示する内容を変えろといったことを実現できる。

2.2 想定使用場面

VolGrab の想定される使用場面として画面全体を操作対象とする場合と画面の一部を操作対象とする場合の 2 パターンが考えられる。画面全体を操作対象とするものとして典型的なのは地図閲覧である。Google Maps のような地図を表示するアプリケーションは特定の場所を探しその場所までの行き方を確認する用途に用いられることが多い。図 4 は動物園から空港への道を調べたいという場面を想定している。動物園から空港への道を調べたい時には動物園 (図 4(a)) や空港 (図 4(c)) の周辺をズームインして見たり、道のりの概観 (図 4(b)) をズームアウトして見たい場合がある。そのような場合にはズーム操作とスクロール操作を頻繁に行うことになる。そのときに提案する UI を使うことでズームとスクロールを統合した操作によってそれらを一動作でできるので、ズームとスクロールを別々に行う従来のマウスなどよりも必要とする手順が減り、そのような操作を快適に実行することができると期待される。

一方、画面の一部を操作対象とするものとして典型的なのはボタンやスクロールバーなどの UI コンポーネントの操作である。細かいボタンなどの UI コンポーネントはタッチ操作では押し間違いなどの意図しない操作をしやすいが、一時的に拡大できれば操作ミスを減らすことができる。例えば、図 5 のようにスクロールバーやボリュームスライダーのような UI コンポーネントに対してズーム操作をすることによってバーやスライダーの表示される幅を変

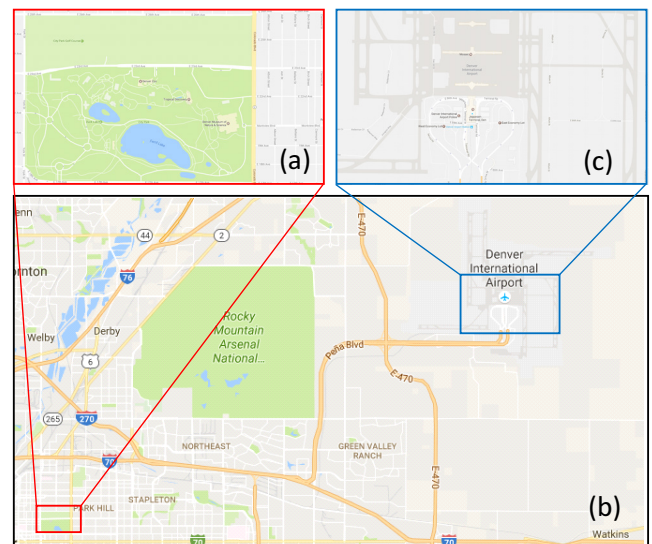


図 4 想定される地図での使用: (a) 動物園, (b) 概観図, (c) 空港
Fig. 4 Use case on map application:
(a)Zoo, (b)Overview, (c)Airport

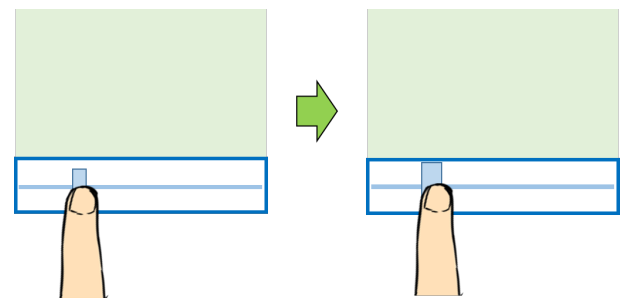


図 5 UI コンポーネントのズーム操作
Fig. 5 Zooming operation for UI components

えることで細かい値を調整しやすくなる。

3. 実装

提案する VolGrab を実装した。以下でシステム構成と空中操作の検出手法、使用例を示す。

3.1 システム構成

今回実装したシステムの構成は図 6 に示すとおりである。近年、デスクトップ PC においてもタッチ操作が可能なものが増えていることから今回はデスクトップ PC の UI を想定した。タッチ操作の取得のために 21.5 インチのフル HD マルチタッチモニタを用いた。また、空中操作の取得のために三脚でディスプレイの近くに固定したデプスカメラ DS325 (SoftKinetic 社) を用いた。取得するデプス画像の解像度は 320×240 画素、デプス値取得可能距離は $0.15 - 1.0\text{m}$ 、デプス画像取得フレームレートは 60fps である。デプスカメラで距離情報を取ることで、カラー画像のみの場合よりも手や画面などの位置を判別しやすくなり、空中操作が取得しやすくなる。つかみ姿勢を検出するためにデプスカメラから得られるデプス画像から手を検出し、

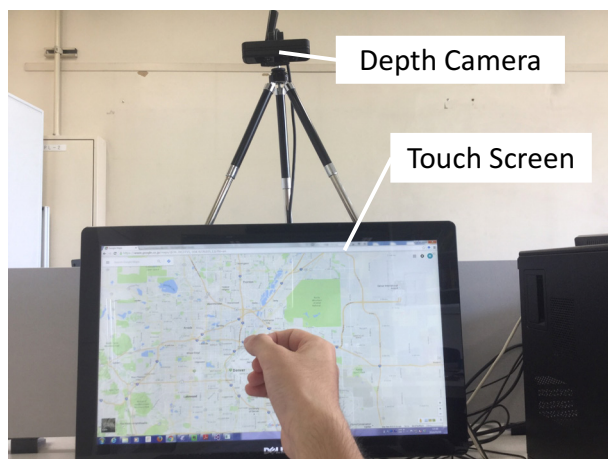


図 6 装置構成

Fig. 6 System configuration

その手の形状からつかみ姿勢をしているかどうかを判定する．また検出された手の位置と画面の位置を利用して画面に対する手の移動量を求め、それに応じてズームやスクロールの変化量が計算される．

3.2 デプス画像を用いた空中操作の検出

空中操作の検出にはデプスカメラで取得されたデプス画像を用いる．空中操作による三次元画面移動のためには、つかんでいる手の位置とつかみ姿勢の検出が必要である．今回の実装ではつかんでいる手の位置として指先の位置を用いる．したがって、指先の位置とつかみ姿勢の検出を行う．

3.2.1 指先位置の検出

指先位置の検出にはデプス画像から抽出した指先領域を用いる．指先領域抽出の処理の様子を図 7 に示す．まず最初に指先領域を得るために用いる手領域を得る．今回はデプス画像に写るもののうち、手のみが動体であると仮定する．手領域抽出のためにデプス画像に写るディスプレイなどの静的な物体を背景差分によって除去する(図 7(a))．次に背景差分によって得られた画像を二値化する．二値化した画像には細かいノイズが含まれているため、それを除去するためにオープニングを行う．オープニングによって得られた画像では手領域が分断されている可能性があるのでクロージングを行い、それによって得られた画像を手領域画像とし、その中の白い部分を手領域とする．指先を画面に向けて操作することとユーザの位置から画面のある方向を指先が向くべき方向であると仮定する．手領域画像を仮定した画面方向の先の端(図 7 の下端)から手前方向に走査し、最初に見つかった手領域のピクセルから手領域を数ピクセル走査して得られた手領域のピクセルを指先領域とする(図 7(b))．得られた指先領域の画像上の座標とデプス値からそれぞれの三次元座標を求め、それらの重心の三次元位置を指先位置として用いる．



図 7 指先領域抽出: (a) 背景差分画像, (b) 指先領域画像

Fig. 7 Fingertip area extraction:

(a)Background difference image, (b)Fingertip area image

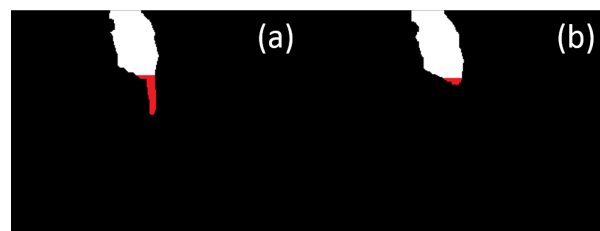


図 8 つかみ姿勢検出: (a) 非つかみ姿勢, (b) つかみ姿勢

Fig. 8 Grabbing posture detection:

(a)Non grabbing posture, (b)Grabbing posture

3.2.2 つかみ姿勢の検出

今回は簡易的につかみ姿勢を検出するために指が伸びている状態を非つかみ姿勢、そうでない状態をつかみ姿勢とする．つかみ姿勢の検出の様子を図 8 に示す．つかみ姿勢の検出ではつかみ姿勢時に検出される指の長さが非つかみ姿勢時に検出される指の長さよりも短くなることを利用して検出する．指先領域の検出と同様に手領域画像を走査し、指先が検出されてから指として想定される太さの範囲外となる太さが検出されるまで走査された領域を指領域とする．これは図 8 の赤で示した領域である．指領域の検出の時に走査されたピクセル数を指の長さとして用いる．得られた指領域の長さが閾値を超えた場合は非つかみ姿勢、超えなかった場合はつかみ姿勢であるとする．

3.2.3 画面と指との距離とポインタ表示座標の取得

VolGrab では画面と指との距離を用いてズーム率の変更を行う．また、インフォメーションボリュームをつかんでいる位置がユーザにわかりやすいようにポインタを表示する．そのために画面と指との距離とポインタ表示座標を取得する必要がある．今回、ポインタは指先位置から平面に対して垂直に下ろした直線と平面との交点に表示するので、それらを取得するために画面の三次元空間上の位置を求める必要がある．

ポインタを表示する画面上の座標を求めるために、指の座標と画面上の座標を対応付けるキャリブレーションを行う．指先位置の検出を利用して指先を画面にくっつけた状態でその三次元座標を検出することで画面平面上の三次元座標を得ることができる．その方法を用いて画面平面上の

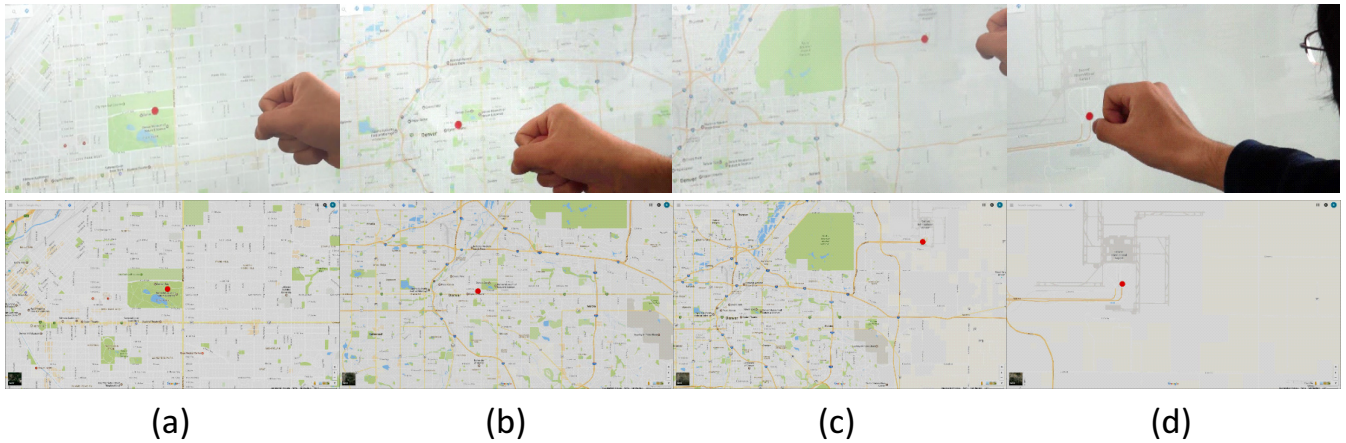


図 9 地図アプリケーションの操作例: (上段) 操作の様子 (下段) 画面上のビュー

Fig. 9 Operation example of map application:

(Top row) Operation, (Bottom row) View on a screen

座標を 4 点以上取得しその座標から式 (1) のように平面の方程式の連立方程式を作り，最小二乗法で解くことで平面の方程式を推定することができる．

$$ax^{(i)} + by^{(i)} + cz^{(i)} = 1 \quad (i = 1, \dots, n) \quad (1)$$

推定した平面の方程式を用いて画面平面が三次元座標上のデブスカメラに対して正対するようにすべての三次元座標を回転し，その時の x, y 座標を画面平面上の同次座標 $p = (x \ y \ 1)^T$ として， z 座標を画面と指との距離として用いる．既知の内部処理上の画面の同次座標 $p' = (x' \ y' \ 1)^T$ と求めた画面平面上の同次座標 p の組み合わせを 4 組以上取得しその組み合わせからホモグラフィ行列 H を求める．そして式 (2) によるホモグラフィ変換を用いて対応付ける．対応付けて求めた p' から得られる座標 (x', y') をポインタを表示する画面上の座標として用いる．

$$p' = \frac{1}{w} Hp \quad (2)$$

3.3 使用例

VolGrab を利用したアプリケーションとして図 9 のようにブラウザで動作する Google Maps を空中操作によって操作するアプリケーションを作成した．このアプリケーションではつかみ姿勢のまま画面に対して平行方向に動かすことでスクロール，垂直方向に動かすことでズームが可能となっている．今回のアプリケーションでは指数的にズーム率の変わるインフォメーションボリュームを用いている．そして空間的に手を動かすことでスクロールとズームを同時に行うことができる．図 9 では実際に操作を行う様子を示している．この図では動物園から空港へ行く場面を想定した操作をする様子を示している．まず動物園の詳細なビュー (図 9(a)) を得てその周辺を確認し，ズームアウトして動物園と空港を含む概観 (図 9(b)(c)) を得て，空港

の詳細なビュー (図 9) を確認するという操作を行っている．手を画面に対して左手前に動かすことでズームアウトとスクロールを同時に行うことができる．タッチスクリーンでのズーム操作はピンチ操作によって行われる．この操作は細かい倍率の変更はしやすいが，例えば地図上のある都市の拡大図からその都市と別の都市を同時に表示する概観を得るような大きな倍率の変更は複数回のピンチ操作が必要となるためにしづらい．空中操作ではピンチ操作よりも一回の操作で手を動かせる幅が大きいので，より大きなズーム倍率の変更がしやすい．VolGrab ではタッチスクリーンにおける操作と空中操作が共存しているために，タッチスクリーンでは細かい調整をし空中操作で大きな変更を行うことで細かいズーム倍率の変更と大きなズーム倍率の変更を独立した操作ですることができるため，よりナビゲーション操作がしやすくなると考えられる．

4. まとめ

本稿では，空中操作による三次元画面移動を可能とする UI として VolGrab を提案した．また，VolGrab を実装しアプリケーションを作成して有効性を確認した．今後の課題としては UI コンポーネントをズームやスクロールするアプリケーションの実装や被験者実験によるユーザビリティの評価が挙げられる．

参考文献

- [1] Hrvoje Benko, Andrew D Wilson, and Patrick Baudisch. Precise selection techniques for multi-touch screens. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 1263–1272, 2006.
- [2] Alex Olwal, Steven Feiner, and Susanna Heyman. Rubbing and tapping for precise and rapid selection on touch-screen displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 295–304, 2008.

- [3] Benjamin B Bederson and James D Hollan. Pad++: a zooming graphical interface for exploring alternate interface physics. In *Proceedings of the 7th annual ACM symposium on User interface software and technology*, pages 17–26, 1994.
- [4] Benjamin B Bederson. Fisheye menus. In *Proceedings of the 13th annual ACM symposium on User interface software and technology*, pages 217–225, 2000.
- [5] Fangzhou Wang, Yang Li, Daisuke Sakamoto, and Takeo Igarashi. Hierarchical route maps for efficient navigation. In *Proceedings of the 19th international conference on Intelligent User Interfaces*, pages 169–178, 2014.
- [6] Xing-Dong Yang, Tovi Grossman, Pourang Irani, and George Fitzmaurice. Touchcuts and touchzoom: enhanced target selection for touch displays using finger proximity sensing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2585–2594, 2011.
- [7] Amartya Banerjee, Jesse Burstyn, Audrey Girouard, and Roel Vertegaal. Waveform: remote video blending for vjs using in-air multitouch gestures. In *CHI’11 Extended Abstracts on Human Factors in Computing Systems*, pages 1807–1812, 2011.
- [8] Takeo Igarashi and Ken Hinckley. Speed-dependent automatic zooming for browsing large documents. In *Proceedings of the 13th annual ACM symposium on User interface software and technology*, pages 139–148, 2000.
- [9] Yujin Tsukada and Takeshi Hoshino. Layered touch panel: the input device with two touch panel layers. In *CHI’02 Extended Abstracts on Human Factors in Computing Systems*, pages 584–585, 2002.
- [10] Yoshiki Takeoka, Takashi Miyaki, and Jun Rekimoto. Z-touch: an infrastructure for 3d gesture interaction in the proximity of tabletop surfaces. In *ACM International Conference on Interactive Tabletops and Surfaces*, pages 91–94, 2010.
- [11] Sven Kratz and Michael Rohs. Hoverflow: expanding the design space of around-device interaction. In *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services*, page 4, 2009.
- [12] Xiang Anthony Chen, Julia Schwarz, Chris Harrison, Jennifer Mankoff, and Scott E Hudson. Air+ touch: interweaving touch & in-air gestures. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*, pages 519–525, 2014.
- [13] Jie Song, Gábor Sörös, Fabrizio Pece, Sean Ryan Fanello, Shahram Izadi, Cem Keskin, and Otmar Hilliges. In-air gestures around unmodified mobile devices. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*, pages 319–329, 2014.
- [14] Nicolai Marquardt, Ricardo Jota, Saul Greenberg, and Joaquim A Jorge. The continuous interaction space: interaction techniques unifying touch and gesture on and above a digital surface. In *Human-Computer Interaction–INTERACT 2011*, pages 461–476. Springer, 2011.