タイルドディスプレイのための
タブレット端末を用いたシームレスな操作手法の開発

鈴木 拓馬 [1)], 清川 清 [1),2)], 竹村 治雄 [1),2)]

1) 大阪大学 大学院情報科学研究科
2) 大阪大学 サイバーメディアセンター

kiyo@ime.cmc.osaka-u.ac.jp

A Seamless Operation Technique for Tiled Display Walls Using A Tablet Device

Takuma Suzuki[1)], Kiyoshi Kiyokawa[1),2)], Haruo Takemura[1),2)]

1) Graduate School of Information Science and Technology, Osaka University
2) Cybermedia Center, Osaka University

概要

近年，大規模で複雑なデータを可視化するために，複数の表示デバイスを複数の計算機により連動させ，全体を一枚の巨大なディスプレイとして扱うタイルドディスプレイ (Tiled Display Walls, TDW) が注目されてきている．しかし，TDW は画面の大きさや複数の計算機を用いた特殊な構成方法が原因で，操作時に頻繁な移動が必要であったり，立位では従来のマウスやキーボードによる入力操作が難しいなど様々な問題を抱えている．そこで本研究では，タブレット端末から TDW を構成する全ての計算機へ同時に VNC 接続を行うことにより，TDW の任意の範囲の映像を閲覧し操作する手法を提案する．被験者実験の結果，ワイヤレスのキーボードとマウスを用いる場合に比べて，提案手法では同等の作業時間でより快適に操作できることを確認した．

1　はじめに

近年，計算機におけるネットワーク技術やプロセッサ技術，計測技術の発展に伴い，科学技術分野で取り扱われるデータは急速に大規模化・複雑化している．このような大規模で複雑なデータを視覚的に提示し，直感的に理解・分析することを支援するための技術として可視化技術が存在する．

大規模データを欠損なく高精度に可視化するためには，高い解像度が必要であり，そのための技術としてタイルドディスプレイ (Tiled Display Walls, TDW) が注目を集めている [1]．TDW とは，図 1 に示すように複数の表示デバイスをタイル状に配置し，それらを複数の計算機により連携させることで一つの巨大な高解像度のディスプレイを構成したものである．

TDW の一般的なシステム構成例を図 2 に示す．TDW の一列分の表示デバイスを 1 台のスレーブノードと呼ばれる計算機が制御しており，それらを 1 台のマスターノードと呼ばれる計算機が制御している．全ての表示デバイスを用いて 1 枚のディスプレイとして使用するには，これらの計算機を連動させる必要があり，そのためのミドルウェアの研究開発が盛んに行われている [2], [3]．一方でテキストの編集やソフトウェアの設定ファイルの変更などコンピュータを扱う上での日常的な作業に焦点を当てた研究はあまり行われていない．

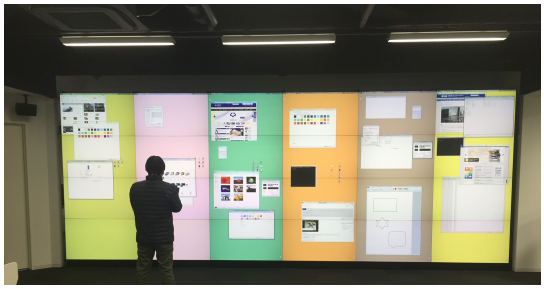

図 1　大阪大学サイバーメディアセンター大型可視化装置

一般に TDW 用ミドルウェアを用いるためには，アプリケーションのソースコードに変更を加える必要があるため，スレーブノードのネイティブアプリケーションを操作する際には，これらのミドルウェアを用いることができない．この場合，表示画面の大きさや複数台の計算機を用いた特殊なシステム構成が原

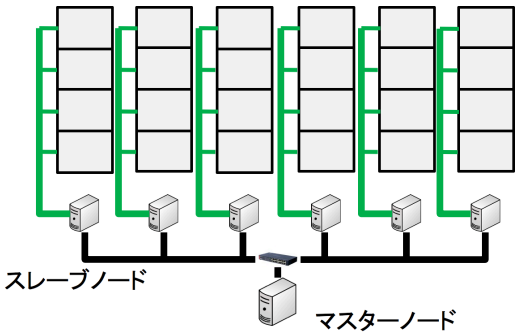

図 2　一般的な TDW のシステム構成

因で，一般的なディスプレイ環境では起こらなかった様々な問題が生じる [1]．その代表例として，操作対象の情報に合わせて頻繁に移動が必要なこと，立位での入力操作が難しいこと，入力先スレーブノードの頻繁な切り替えが必要なこと，異なるノード間のアプリケーションの移動ができないため離れたディスプレイ間での比較作業などが難しいことなどがある．

このような問題を解決するため，本研究ではタブレット端末から TDW を構成する全てのスレーブノードに VNC (Virtual Network Computing) 接続を行い，TDW 全体の一括した操作を実現する．タブレット端末を用いることにより，立位でのポインティングやテキスト編集などを容易にし，さらに視点を自由に移動可能にすることで立ち位置に依らない任意の情報の閲覧および操作を可能とする．任意の範囲の映像の抽出・移動も実装し，これにより離れたディスプレイ間でのウィンドウの比較検討作業も容易にする．

以下，2 節ではこれまでに行われてきた関連研究について紹介する．3 節ではタブレット端末を用いた TDW の操作手法および試作システムについて説明する．4 節では試作システムが実際に作業時間やユーザの負荷を軽減しているかを検証するための実験を行い，結果について考察する．5 節では本研究のまとめと今後の展望について述べる．

2　関連研究

TDW を用いた可視化のために様々なミドルウェアの開発が行われている．また同様に，既存のマウスやキーボードに代わる様々な操作手法も提案されている．本節では，可視化のための TDW 用ミドルウェアや操作手法の関連研究について述べる．

2.1　TDW 用ミドルウェア

TDW 用ミドルウェアの代表例として，SAGE (Scalable Adaptive Graphics Environment) [2] や CGLX (Cross-Platform Cluster Graphics Library) [3] が挙げられる．SAGE を用いると，TDW 全体を連動させ全体を一枚のディスプレイとして扱い，SAIL という専用の API を組み込んだアプリケーションを表示できるようになる．しかし，SAGE はマスターノード上で起動するプログラムになっており，また扱うアプリケーションのソースコードの変更が必要となるため，スレーブノード上のネイティブアプリケーションの操作には利用できない．CGLX についても同様に，専用 API を組み込む必要があるのでネイティブアプリケーションでは使用することができない．

2.2　TDW の操作

TDW を操作する際は頻繁な移動が必要となるため，立位での操作をする場面が多くなる．従来のマウスやキーボードによる入力操作は立位では行いにくいため，様々な操作手法が提案されている．その中で，立位に適し，TDW の映像閲覧と操作を同時に行えるタブレット端末を用いた入力操作についての研究も行われている [4], [5]．[4] の研究はタブレット端末から TDW へ接続することで TDW の映像の任意の範囲を閲覧し，またタッチにより TDW へのクリックの入力を行えるようにする研究であるが，テキスト入力には対応していない．[5] の研究はタブレット端末によるテキスト入力も可能であるが，同じ入力信号が TDW を構成する全ての計算機へ送られてしまうため，入力先の切り替えができない．また，[4], [5] のいずれも，専用のアプリケーションに特化しており，ネイティブアプリケーションに対する操作は行えない．

2.3　本論文の位置づけ

これらの研究を踏まえ，本研究では

- 立位での容易な入力
- 立ち位置に依らない情報閲覧
- TDW の一括操作

の機能を持つ操作システムを実現し，それにより離れたディスプレイユニット間の作業における作業時間の短縮やユーザの身体的・心理的負荷の軽減を目指す．

3　提案手法および試作システム

本節では，タブレット端末を用いた TDW の操作手法を提案する．3.1 節では全体のシステム構成について，3.2 節では提案システムの機能について述べる．

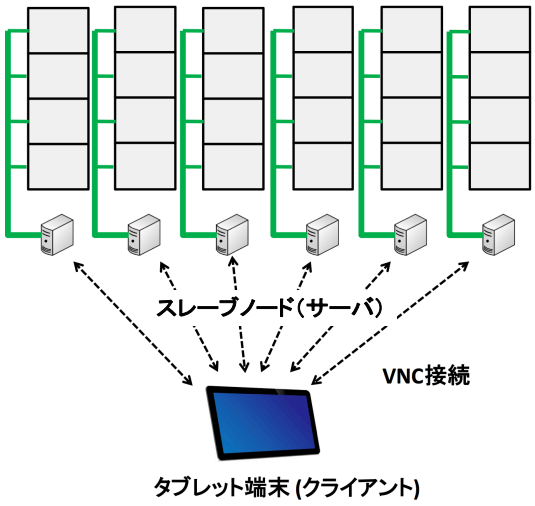

図 3　提案システムの構成

3.1　システム構成

提案システム全体の構成を図 3 に示す．提案システムは VNC サーバとクライアントから構成されており，TDW を制御する各スレーブノードがサーバ，タブレット端末がクライアントとなる．タブレット端末を用いることで，立位での容易な入力という要件を満たすことができる．クライアントから全てのサーバに同時に VNC 接続を行うことで，TDW 全体をタブレット端末上で表示することが可能となる．クライアントアプリケーションにはインターネットブラウザ (以下，ブラウザ) を使用し，それに伴いサーバアプリケーションにはブラウザから接続可能なものを使用する．

HTML のインラインフレーム (iframe) の機能により HTML ページを複数に分割することで，1 つのページ内で複数のスレーブノードへ同時に VNC 接続が可能となる．HTML ページ内から全てのスレーブノードへ同時に接続することにより，TDW の映像を一括して表示することができる．ピンチイン・ピンチアウトのジェスチャーによりページを拡大・縮小することで立ち位置に依らない情報閲覧の要件を満たすことができ，またタッチパネル機能およびソフトウェアキーボードにより各スレーブノードへ入力を行うことで，TDW の一括操作の要件を満たすことができる．また，インターネットブラウザをクライアントとして利用することで，クライアントが OS に依存せず，様々な種類のタブレット端末から利用可能になるという利点もある．試作システムにおいては，サーバには ThinVNC [6] を，クライアントには Google Chrome を使用した．

3.2　機能概要

本節では，提案システムの概要を述べる．まず，タブレット端末から全てのスレーブノードに同時に接続することで TDW 全体の映像を一括してタブレット端末上に表示する．そして，ピンチイン・ピンチアウトのジェスチャーによりその映像を拡大・縮小する．さらに，タッチパネル機能およびソフトウェアキーボードによりスレーブノードに対し入力を行う．また，映像中の任意の 2 点を指定することで，その 2 点を対角とした矩形領域を画像オブジェクトとして抽出する．

3.2.1　映像の一括表示

TDW 全体の映像をタブレット端末上に表示する．スレーブノード 1 台に接続し，その映像をブラウザ全体を用いて表示し操作を行う HTML ページ (以下，子ページ) および，複数の子ページを縮小表示して同時に呼び出す HTML ページ (以下，親ページ) を用意する．親ページ内をスレーブノードの構成に沿って iframe で分割し，各 iframe 内から子ページを表示する．図 2 のシステム構成の TDW に対応する，親ページと子ページの構成を図 3 に示す．各子ページから各スレーブノードへ VNC 接続を行うことで，1 つの HTML ページ内から全てのスレーブノードに接続でき，TDW 全体の映像を一括して閲覧可能になる．4.1 節の試作システムを用いて映像の一括表示を実行している様子を図 4 に示す．

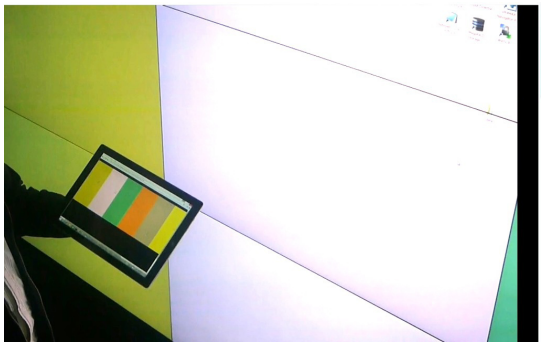

図 4　TDW の映像を一括表示する親ページ

3.2.2　入力操作

タブレット端末に搭載されている入力機能を用いて，スレーブノードに対する入力操作を行う．カーソル操作およびクリックはタッチパネル機能を用い，文字入力はタブレット端末に搭載されているソフトウェアキーボードを呼び出して行う．これらの入力信号を VNC 通信によりスレーブノードへ送り，スレーブノードへの入力を行う．図 5 にソフトウェアキーボードを

図 5　ソフトウェアキーボードを用いた文字入力

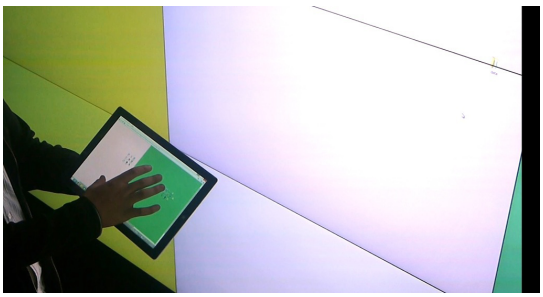

図 6　ピンチアウトによる拡大操作

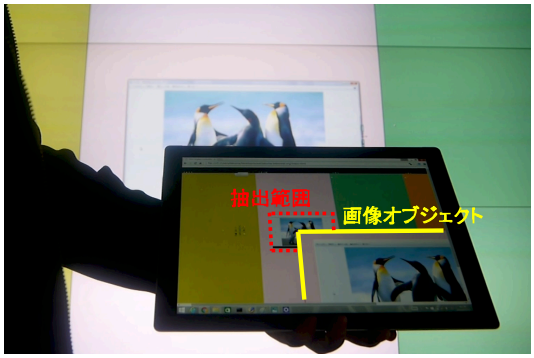

図 7　映像の部分抽出

用いて文字入力している様子を示す．

3.2.3　拡大・縮小

タッチパネル機能を持つ端末向けにブラウザに実装されている機能を用いて，タブレット端末上に表示される映像の拡大・縮小を行う．タッチパネル上で，2 本の指でピンチアウト・ピンチインのジェスチャーを行うことにより画面を拡大・縮小する．得たい情報の詳細度に応じて画面を拡大・縮小することで，立ち位置を変えることなく任意の情報を得ることができる．図 6 に拡大操作を実行している様子を示す．

3.2.4　映像の部分抽出機能

GUI の一部を切り取り，任意の場所に再配置することによりアプリケーションの利便性を向上させる研究として Façades [?] や Prefab [8] がある．GUI の一部を切り取り移動させることができれば，離れた場所にある GUI 同士の比較が容易になると考えられる．しかし，これらの研究は単一の計算機での使用を想定しているため，TDW には対応していない．そこで，これらの研究を参考に以下のような機能を実装した．

タブレット端末に表示される映像中の任意の 2 点を指定することで，図 7 のように 2 点を対角とした矩形領域の映像をブラウザ上で画像オブジェクトとして抽出する．抽出された画像オブジェクトは映像よりも手前に配置され，指でドラッグすることで任意に移動させることが可能である．TDW 上で離れた座標にある 2 つの画像などを比較するとき，片方の画像をブラウザ上で抽出し移動させることで，2 つの画像を隣接させ容易に比較することが可能になる．

3.3　負荷軽減機能

TDW へ接続すると，TDW の巨大な画面全てをタブレット端末上で同時に表示するため，映像表示の負荷が高くなり動作が遅くなると考えられる．そのため，スクロール時に iframe の列ごとに現在タブレット端末上に表示されているかを判定し，表示されていない列は VNC 接続の切断を行う．スクロール時の映像の読み込みには時間がかかるが，画面内に表示されていないスレーブノードからの通信負荷をなくせるため，総合的な操作速度は向上すると考えられる．

4　評価実験

提案手法 (以下，TB) の有用性を評価するため，従来のマウス・キーボードによる操作手法 (以下，MK) との比較実験を行った．

4.1　実験環境

実験は，図 1 に示す大阪大学吹田キャンパスに設置されている縦 4× 横 6 の 24 面 TDW (吹田 TDW，横 6.5m, 縦 2.4m) を用いて行った．手法 MK においては，キーボードやマウスを一時的に置くために，被験者にはキャスターおよび補助台の付いている椅子を 1 台運搬しながら移動することを許可した．また，手法 MK では，Input Director [9] を使用することで，画面端を越えると隣接するディスプレイへカーソルが移動するようにし，1 台の無線マウスおよび無線キーボードで全てのスレーブノードを操作可能とした．手法 TB では，タブレット端末として Microsoft Surface Pro3 を使用した．

4.2　実験手順

TDW の各列を，左から順に列 A〜F と呼ぶ．実験タスクは，列 A に一般的なソフトウェアのシステム設定を模した GUI を表示し，その設定を列 B〜列 F のいずれかに表示される GUI に反映させるものである．

4.2.1　実験用 GUI

列 A のデスクトップ上には Read.jar が置かれ，列 B〜F のデスクトップ上には Write1.jar および Write2.jar が置かれている．Read ファイルをクリックすると図 8 のような GUI (以下，Read GUI) が表示され，Write ファイルをクリックすると図 9 のような GUI (以下，Write GUI) が表示される．Read GUI 上部には Write GUI に設置されているものと同一のコンポーネントに対して入力を行った画像 (以下，入力情報) が表示されている．下部は，Next ボタン，入力先，進捗度の 3 つの要素で構成されている．

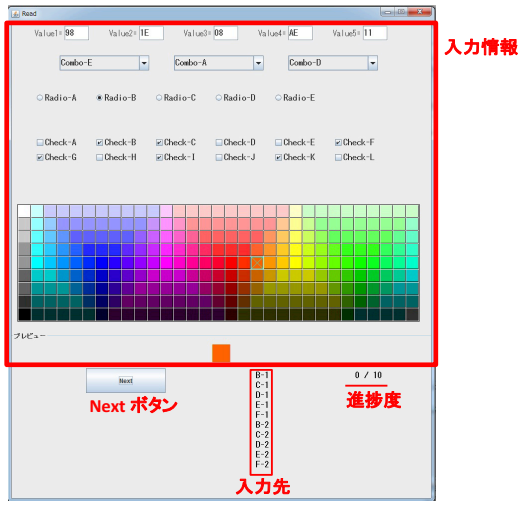

図 8　Read GUI の例

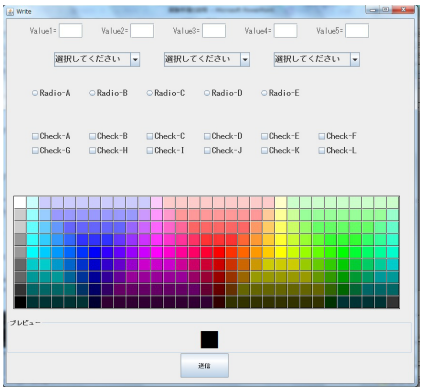

図 9　Write GUI の例

4.3　実験タスク

被験者はまず，列 A に存在する Read.jar をクリックする．すると，図 8 のように Read GUI が表示されるので，次は入力先に指示された Write1.jar または Write2.jar をクリックする．そして，Read GUI の入力情報と一致するように，Write GUI に入力を行う．全て入力し終えたら送信ボタンを押し，一致しているなら「OK」というダイアログが表示された後 Write GUI が閉じられる．一致していないなら「設定が異なります」というダイアログが表示され，入力を続行する．「OK」が表示され Write GUI が閉じられたら，Read GUI の Next ボタンを押す．すると，入力情報，入力先および進捗度が更新され，被験者は新しい入力先に対して再び同じ作業を行う．これを全てのタスクが完了するまで繰り返す．Read GUI が開かれた時点から実験が開始し，タスクが全て終了し Next ボタンを押した時点で実験が終了する．ここで，「Read GUI に入力情報が表示されてから，Write GUI への入力を完了し，Next ボタンを押すまでの時間」を 1 タスクとする．各列 2 回ずつの全 10 回のタスクを 1 セットと呼ぶ．

4.3.1　従属変数

上記の実験を通して，タスク達成時間，歩数，主観評価の 3 項目について計測を行う．

4.3.2　仮説

実験結果について，各手法におけるデバイスの特徴やユーザインターフェースの特徴により以下の仮説を立てた．

H1 タスク達成時間は手法 TB のほうが短くなる

H2 身体的・心理的負荷は手法 TB のほうが低くなる

仮説 **H1** については，手法 MK は Read ファイルの確認および広範囲のカーソル操作のために頻繁に移動する必要が生じるのに対し，手法 TB は移動の必要性がないため作業時間を短縮できるという予想による．仮説 **H2** については，手法 MK は Read GUI の確認および広範囲のカーソル操作のために頻繁に移動する必要が生じるのに対し，手法 TB は移動の必要性や情報を覚える必要性がないため，身体的，心理的負担が低いという予想による．

4.4　結果および考察

4.4.1　タスク達成時間

まず，手法ごとの全タスクの平均タスク達成時間の平均および標準誤差を図 10 に示す．

全タスクの達成時間の平均は手法 MK では 1303.0

秒に対し手法 TB では 1200.0 秒であった．しかし，t 検定を行った結果 $p = 0.47$ となり，2 手法の平均時間に統計的有意差は認められなかった．この理由として，被験者はマウスおよびキーボードは日常的に使用し慣れているため手法 MK におけるタスク達成時間のばらつきは少なかったのに対し，手法 TB におけるタブレット端末の操作法は不慣れのため人により得手，不得手の差が明確になったことが考えられる．

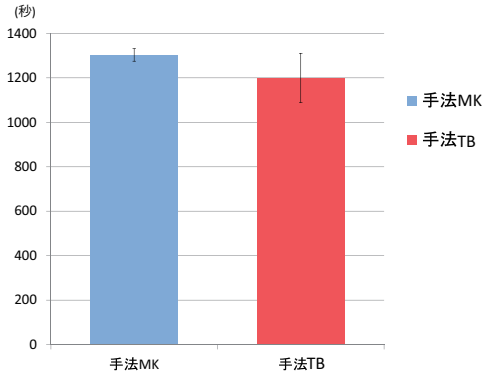

図 10　セットあたりの平均タスク達成時間

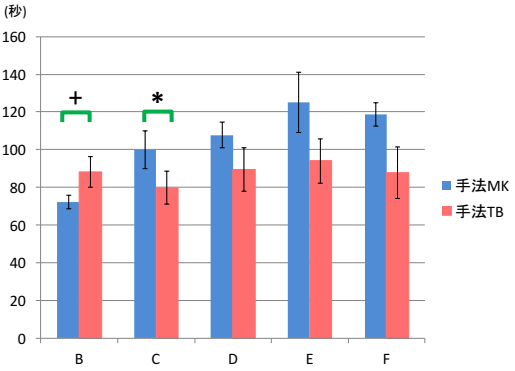

図 11　入力先ごとのタスクあたりの編集所要時間

より詳細な分析を行うため，「Write GUI を開いてから入力が完了するまでの時間」(以下，編集所要時間) について分析する．入力先の列ごとのタスクあたりの編集所要時間の平均と標準誤差を図 11 に示す．

編集所要時間について，列 B では有意傾向 ($p < 0.1$) が，列 C では 5% 水準で有意差 ($p < 0.05$) が見られた．従って，**H1** を明確に採択することはできない．これにより，隣接する列 B では Write GUI を編集しながら移動なしで Read GUI を確認できるため手法 MK のほうが編集所要時間は短かったが，列 C では移動が必要となるため手法 TB のほうが編集所要時間が短くなったと考えられる．また，列 D〜F ではばらつきが大きいため有意差および有意傾向は見られなかったが，平均時間では手法 TB のほうが短かった．このため，被験者数を増やすと有意差が見られる可能性がある．

4.4.2　歩数

手法 MK および手法 TB における，1 セットあたりの被験者の歩数を図 12 に示す．このグラフより，手法 TB はディスプレイ間の作業において，被験者の歩数を大幅に減少させていることが分かる ($p < 0.01$)．よって，手法 TB により被験者の身体的負荷を軽減することが確認できた．

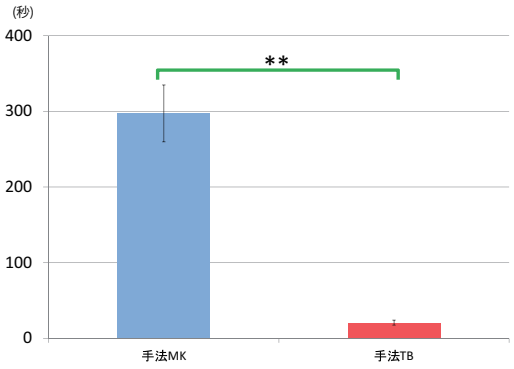

図 12　セットあたりの歩数

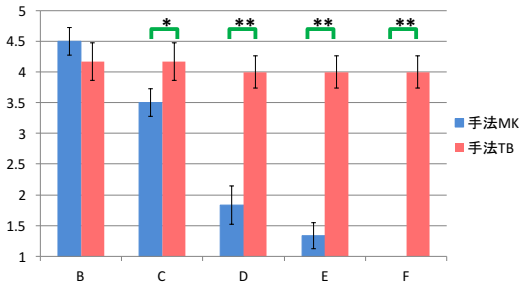

図 13　各列における編集作業のしやすさ

4.4.3　主観評価

実験後，各列における編集作業のしやすさについて被験者に 5 段階のアンケートを実施した．5 が「非常に同意できる」，3 が「どちらでもない」，1 が「全く同意できない」とした．図 13 にその結果を示す．

このグラフより，手法 MK ではディスプレイ間の距離が離れるにつれ作業の負荷が増大しているのに対し，手法 TB では距離が離れても負荷はほぼ変わらないことが分かる．そして，隣接していない列 C〜列 F において手法 TB は手法 MK よりも作業者の心理的負荷を軽減することが確認できる．歩数の結果と合わ

せて，**H2** は採択されたといえる．

5 おわりに

本研究では，タブレット端末から TDW を構成する全てのスレーブノードへ同時に VNC 接続を行うことにより，TDW の映像の一括した閲覧・操作や拡大縮小による任意の詳細度の情報の取得ができるシステムを提案し試作した．そして，被験者実験により，離れたディスプレイユニット間で映像を比較する作業において，一部で作業時間を短縮し，ユーザの身体的・心理的負荷を軽減することを確認した．今後の課題としては，タブレット端末上の映像表示のさらなる通信負荷軽減による応答性の向上や抽出した画像オブジェクトのリアルタイム更新などが挙げられる．

参考文献

[1] T. Ni, G. S. Schmidt, O. G. Staadt, M. A. Livingston, R. Ball, and R. May, “A survey of large high-resolution display technologies, techniques, and applications,” in Proceedings of the IEEE Virtual Reality Conference, pp. 223236, 2006.

[2] L. Renambot, A. Rao, R. Singh, J. Byungil, N. Krishnaprasad, V. Vishwanath, V. Chandrasekhar, N. Schwarz, A. Spale, C. Zhang, G. Goldman, J. Leigh, and A. Johnson, “SAGE: the Scalable Adaptive Graphics Environment,” in Proceedings of WACE, Vol. 9, No. 23, pp. 200409, 2004.

[3] K.-U. Doerr and F. Kuester, “Cglx: a scalable, high- performance visualization framework for networked display environments,” IEEE Transactions on, Visualization and Computer Graphics, vol. 17, no. 3, pp. 320-332, 2011.

[4] K. Cheng, J. Li, and C. Mller-Tomfelde, “Supporting interaction and collaboration on large displays using tablet devices,” in Proceedings of the ACM International Working Conference on Advanced Visual Interfaces, pp. 774775, 2012.

[5] S. K. Badam, E. Fisher, and N. Elmqvist, “Munin: A peer- to-peer middleware for ubiquitous analytics and visualization spaces,” IEEE Transactions on Visualization and Computer Graphics, vol. 21, no. 2, pp. 215228, 2015.

[6] Thinvnc, http://sourceforge.net/projects/thinvnc/ (Last access: Oct. 7, 2016)

[7] W. Stuerzlinger, O. Chapuis, D. Phillips, and N. Roussel, “User interface façades: towards fully adaptable user interfaces,” in Proceedings of the 19th annual ACM Symposium on User Interface and Software and Technology, pp. 309318, 2006.

[8] M. Dixon and J. Fogarty, “Prefab: implementing advanced behaviors using pixel-based reverse engineering of interface structure,” in Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems, pp. 15251534, 2010.

[9] Input Director, http://www.inputdirector.com/ (Last access: Oct. 7, 2016)