

マルチディスプレイ環境における 並行作業を支援する操作性向上手法

本本亮司[†] 市村哲[†]

本論文では、マルチディスプレイ環境下で通常のマウス操作と顔の動きを併用した操作により並行作業を支援するシステムを提案する。近年、マルチディスプレイを使用したコンピュータの使用者が増加しており、効率的に情報を取得し、並行作業を行う手段としての役割は非常に大きい。しかし、作業スペースが広がった分だけ、マウスカーソルの移動する距離・時間が増加してしまうなど、ユーザへの負担も少なくない。そこで本研究では、Webカメラを用いてユーザを撮影した映像から、顔の向きを自動的に検出する機能、対応するディスプレイにマウスカーソルの移動やダイアログウィンドウの表示を行う機能を持つシステムを構築した。評価実験の結果、マウスカーソルを移動させる時間・距離を共に短縮する事ができ、本システムの有効性が確かめられた。

Improvement of Operation in Multi-Display Environment for Supporting Multitasking

Ryoji Kimoto[†] Satoshi Ichimura[†]

This paper presents improvement of operation in multi-display environment for supporting multitasking. Recently, the use of multi display is becoming popular. Multiple monitor systems allow people to be more productive. However, it is difficult to use multiple monitors as easy as single monitor. We propose an operational support system with facial direction recognition function based on camera image analysis. We also implemented a function to move the mouse pointer between monitors automatically and use the mouse to move within each monitor. The experiment showed the proposed system required less mouse movement and reduced task completion time.

1. はじめに

近年、マルチディスプレイを使用したコンピュータの使用者が増加している。価格の低下により、高解像度のディスプレイが手に入れやすくなったことや、一般的にコンピュータに比べてディスプレイのライフサイクルは長いために、ディスプレイを再利用できる事が、利用者増加の理由として挙げられる。特に、映像系のクリエイター、株取引のトレーダー、ソフトウェア開発者などの間では広く利用されており、効率的に情報を取得し処理する手段としての役割は非常に大きい。また同様の理由から、一般利用者に対しても利用の幅は広く、今後も益々利用者の増加が期待される。

そこで本稿では、マルチディスプレイ環境における並行作業を支援するシステムの提案・実装を行った。

事前に行った予備実験によって、マルチディスプレイの使用者がディスプレイを見る際の顔の向きによって、注目しているディスプレイを判別できたということから、本研究では、Webカメラを用いてユーザを撮影し、その映像から顔の向きを自動的に検出する機能、対応するディスプレイにマウスカーソルの移動やダイアログウィンドウの表示を行う機能を持つシステムを構築した。

本システムを使用した評価実験によって、マウスカーソルを移動させる時間と距離共に短縮する事ができ、本システムの有効性が確かめられた。

2. 背景・問題点

マルチディスプレイ環境では、従来よりも広い作業スペースを得ることができるようになる。そのためウィンドウを重ならず複数同時に表示することや、ディスプレイを跨いで1つのウィンドウを表示するなど、ユーザに1度に多くの情報が提供することができる。さらに、複数のドキュメントやアプリケーションの間を切り替えて使用するという必要性が低減または排除される。その為ビジネスユースにおいては、手軽に生産性の向上が図れるという点で利用される機会が多く、マルチディスプレイ環境によって約20~40%の生産性向上が図れると言った報告もされている⁵⁾⁶⁾。また、個人ユースの使用においても、テレビを見ながらインターネットをしたり、一方のディスプレイでローカルマシンの操作しながら、他方のディスプレイでは遠隔地のマシンを全画面で操作したりと、様々な理由から使用されている。

しかし、マルチディスプレイを使用すると作業スペースが広がった分だけマウスカーソルの移動する距離・時間が増加してしまう。また、ディスプレイ間を跨いでマウスカーソルが移動する状況が頻繁に起こり、マウスカーソルを見失うなどの問題がある。さらに視角の関係から、ユーザが一度に目視して内容を理解できる範囲は、高々

[†] 東京工科大学 コンピュータサイエンス学部
School of Computer Science, Tokyo University of Technology

一つのウィンドウまたは一つのディスプレイである．そのため、ディスプレイごとに配置されているアプリケーションの状況の把握が困難になり、エラーダイアログに気付かないなどの問題がある．これらはマルチディスプレイを使用する上でユーザの負担になる．これらの問題点に対して、視線を利用してウィンドウを操作する手法として Ohno²⁾³⁾の研究がある．また、Mark ら¹⁾はヘッドトラッキングによってマウスの移動を支援する手法を提案している．美原ら⁴⁾は、タッチ操作とスライド操作を併用可能な入力デバイスを作成して、1つのデバイスから、マウスカーソルの複数の動かし方を可能にしている．

3. 予備実験

上記の問題点を解決するためにマルチディスプレイ環境の PC を操作する際にマウス操作とユーザの顔の動きを併用し使用するインターフェースを考えた．そこで、マルチディスプレイ使用時のユーザの顔の方向と注視しているディスプレイとの相関性について予備実験を行った．17 インチディスプレイを横に 3 台並べた環境を使用し、ディスプレイ毎に番号を設定する．被験者は普段コンピュータを使用するのと同様に、アプリケーションを操作する・マウスを移動するなどの操作を行う．観測者は正面から被験者の動作の様子を観測する．観測者は顔の方向から被験者の注視しているディスプレイを判別し、対応する番号を被験者に随時報告する．被験者が注視ディスプレイを 10 回変更した時点で実験を終了し、被験者が注視していたディスプレイと対応する番号と、観測者が報告した番号が一致した回数を集計した．実験は大学生 5 人に対して被験者を依頼し、観測者は著者が行った．表 1 に実験結果を示す．

表 1 実験結果

被験者\結果	一致 (回)	不一致 (回)	合計 (回)
被験者 A	10	0	10
被験者 B	10	0	10
被験者 C	10	0	10
被験者 D	10	0	10
被験者 E	10	0	10
合計 (回)	50	0	50

表 1 の様に、全ての被験者に対して顔の方向から注視しているディスプレイを判別することが 100% 可能であった．また、顔の方向にあるディスプレイとは別のディスプレイを、視線の動きによって見るという行為は観測できなかった．実際に被験者に対して、上記の行為を行ってもらったが、「目が疲れる」、「不自然である」などの意見

が得られた．さらに、注視しているディスプレイ内で操作を行っている場合にも多少の顔の動きがあるが、見ているディスプレイを切り替える時の顔の動きと比較した場合、非常に小さい動きであることも分かった．これは上記に加えて、1 枚のディスプレイ内であれば、表示してある物を目の動きによって捉えることができるためだと考えられる．

4. 提案

Web カメラの映像を使用したマルチディスプレイ環境のための操作支援システムを提案する．本稿で提案するシステムは、以下の機能を有する．

1. 注視ディスプレイにマウスカーソルを移動
2. 注視ディスプレイにウィンドウを移動
3. 注視ディスプレイにメッセージウィンドウを表示

1 に関し、本システムを利用するユーザは、通常のマウス操作と顔の動きを併用して PC を操作する．ユーザが別のディスプレイに表示されている物を操作する場合に、ディスプレイを見る為に顔の向きを変えた段階でそのディスプレイの特定の位置にマウスカーソルが移動し、その後の操作を通常のマウス操作によって行うことができる．図 1 に動作のイメージを示す．

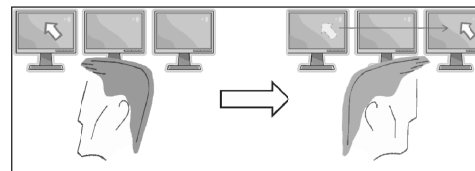


図 1 顔の向きによってマウスカーソルが移動する様子

また 2 に関し、ウィンドウをクリックまたはドラッグ中に顔の向きを変更することで、マウスで指定しているウィンドウを移動させることができる．本システムではディスプレイをまたいでマウスカーソルが移動する際には、顔の動きによってマウスカーソルが自動的に移動するため、手によってマウスを動かす必要がなく、移動時間が軽減する．また、手によるマウスカーソルの動かす範囲は、1 枚のディスプレイ内に限定されるため、従来のマウスカーソルの精度を維持することができる．その為、1

枚のディスプレイを使用している時と同等の操作性を維持したまま、マルチディスプレイを使用できると考えられる。

3に関し、前述したように、ユーザが一度に目視して内容を理解できる範囲は、高々一つのウィンドウまたは一つのディスプレイである。複数のアプリケーションを同時に起動し、マルチタスクに作業をする際には、マルチディスプレイ環境は作業スペースが広がる一方で、アプリケーションの状況の把握が困難になる。そこで、アプリケーションが出すエラーメッセージや確認ダイアログ等の優先度の高いメッセージウィンドウの表示位置を強制的に注視ディスプレイに変更する。

本システムではマルチディスプレイ使用時に、ユーザを正面から Web カメラを用いて撮影し、リアルタイムにユーザの顔の向きを検出することで、注視しているディスプレイを検出する。動画像からの顔認識には、飯村葉子ら⁷⁾、武岡さおりら⁸⁾⁹⁾、opencv¹⁰⁾などの研究がある。本方式では、Web カメラで正面から自身の顔を撮影するだけでよく、特別な装置の使用や特殊なデバイスに体につける必要は無く、ユーザにとって負担が少ない。また、ビデオチャットなどの利用によって、Web カメラは幅広く利用されており、比較的に安価に入手することができるなどの利点がある。顔の向きの検出は顔の位置と口の位置から判定する。顔領域の中心の位置と口領域の重心との距離の差を所定の範囲内によって場合分けすることによって、顔の向きを判別する。

5. 実装

システムは、Windows XP 上で実装を行った。画像処理には DirectShow を利用した。処理は、顔の向き検出とマウス・ダイアログ表示制御から構成される。システムの流れを図 2 に示す。

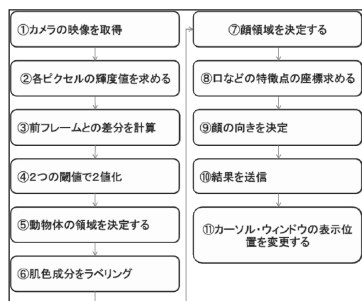


図 2 動作の流れ

5.1 顔の向き検出

- ① Web カメラを使用して、人物・背景の映っている映像を取得する。本方式では、352×288 ピクセルの映像を使用し、各ピクセルの RGB を取得している。RGB はそれぞれ 0~255 の値をとり、R は赤、G は緑、B は青の度合いを示す。
- ② すべてのピクセルから輝度を計算する。
- ③ 前フレームの輝度との差分を求める。非動物体の範囲はこの値が非常に小さくなるが、この際に、背景に映っている物体の中で実際には動いていないにも拘らず、細かく輝度が変化してしまうために動物体として検出されてしまうという現象が起こった。また、背景に蛍光灯（フリッカーの影響）、ディスプレイ、カーテンなどが映っていると輝度が高速に変化してしまうため、ノイズとして誤検出されてしまうという現象も見られた。この場合の輝度差の変化量は、人間が顔を動かす行為による変化よりも非常に大きくなる。
- ④ そこで、輝度差の絶対値を求め、大小の二つの閾値で 2 値化することでノイズの影響を抑えるようにした。変化のある領域の輝度差を 1 とする。
- ⑤ X 軸 (0~351) のそれぞれの値に対して、Y の値が 1 である個数の合計を求める。その合計が所定の値を超えた中で、X の最大最少の値を動物体の X の最大最少の値とする。同様にしては、Y の最大 (頭の頂点の位置) を求める。求めた動物体の領域は、図 3 の[1]の線で囲まれた範囲である。このとき、動物体の領域の幅・高さが所定の値よりも小さい場合は、以降の処理を行わない。

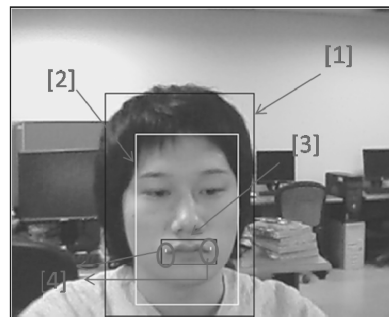


図 3 領域検出の様子

- ⑥ ⑤で求めた領域の各ピクセルの RGB 値から HSV 値を求める。HSV 色空間において、H は色相であり、0~359 の値で色の種類を表す。S は彩度であり、0~1.0 の値で

色の鮮やかさを表す。0に近い程くすんだ色となる。Vは明度であり、0~1.0の値で色の明るさを表す。求めたHSV値から肌色部分を特定する。求めたHSV値から、以下の数式(1)、(2)の条件を用いて肌色部分を特定する。

$$0 < H < 35 \quad (1) \quad 0.3 < S \quad (2)$$

⑦ ⑥で求めた領域に対してラベリング処理を行う。ラベリング処理は連結している画素の個数を求め、番号をつける処理である。ラベリングされたオブジェクトごとに面積を求め、最も面積の大きいラベルを顔の領域とし、X軸Y軸共に最大最少を求める。得られた領域が、図3の[2]の線で囲まれた領域である。

⑧ ⑦でも求めた領域の下半分の領域に対して、②で求めたグレースケールを用いて2値化を行う。⑥と同様にラベリング処理し面積、重心を求める。最も面積の大きいラベルを口の領域(図3の[3])とし、口の両端の座標を求める。得られた座標が図3の[4]で示した位置である。

⑨ 顔の方向の決定には、口の重心のXの値(Xm)、顔の中心のXの値(Xf)、閾値(t)を使用する。ディスプレイを3台横に並べて使用した場合を例にすると、Xmの値は、以下の数式(3)の様に、 α 、 β 、 γ の3つの範囲に収まる。

$$\alpha < Xf - t \leq \beta \leq Xf + t < \gamma \quad (3)$$

Xmがとる値の範囲によって、顔の方向を認識する。

⑩ マウス・ダイアログ表示制御に顔の方向を認識した結果を送信する。

5.2 マウス・ダイアログ表示制御

注視ディスプレイの変更の通知を受け、マウス・ダイアログの表示を制御するプログラムである。

5.2.1 マウスカーソル

マウスカーソルの移動パターンとして、4パターンの中から選択して使用できるようにした。注視ディスプレイが変更されたときのみ、強制的に表示位置を変更する。以下に示す図4、図5、図6、図7は3枚のディスプレイをスクリーンキャプチャしたものを1枚に繋げた物であり、ディスプレイ①からディスプレイ③にマウスカーソルを移動する様子を表している。

- 画面中央(図4)・・・各ディスプレイの中央にマウスカーソルを移動する。

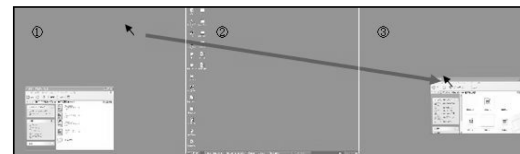


図4 画面中央に移動している様子

- Y座標保持(図5)・・・画面の縦方向をY座標とする。Y座標は現在のマウスカーソルのY座標を使用し、X座標は各ディスプレイの中央を使用する。



図5 Y座標を保持して移動する様子

- XY座標保持(図6)・・・画面の横方向をX座標、縦方向をY座標とする。X、Yの座標は移動前のディスプレイの座標を注視ディスプレイの変更後も使用する。



図6 X,Y座標を保持して移動する場合

- 履歴使用(図7)・・・ディスプレイごとにマウスカーソルの座標を保存しておき、変更通知があった場合に、過去にマウスカーソルがあった座標に移動する。

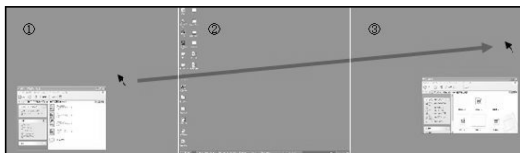


図 7 履歴を使用して移動する様子

5.2.2 ダイアログ

メニューからダイアログを選択することで機能が開始される。ダイアログ表示位置を注視ディスプレイ画面中央に変更する。

6. 評価・考察

本システムの利用によって、マルチディスプレイ環境における適切な操作支援が行えていれば、マウスカーソルの操作性向上が図れるはずである。まず、被験者 5 人に対して、見ているディスプレイを任意で変更する動作を 10 回行ってもらい、システムが正しく方向認識しているかどうかを集計した。

表 2 顔の方向認識

被験者\認識	成功	失敗
被験者 A	10	0
被験者 B	9	1
被験者 C	8	2
被験者 D	9	1
被験者 E	10	0
合計	46	4

表 2 の様に正しく認識された回数は 46 回であり、認識率は 92% となった。誤認識した理由として、正しく口の位置が取得されない事、顔の方向が、方向を判別する閾値周辺に留まった事などが挙げられる。次に、本システムを利用した場合と利用していない場合のマウスカーソルの移動時間・移動距離の測定を行った。評価に使用したプログラムは、図 7.2、図 7.3 の様にディスプレイ①とディスプレイ③のランダムな場所に交互にウィンドウを生成する。表示されているウィンドウは常に 1 つとし、OK ボタンをクリックするたびに新たに作られる。10 回ウィンドウが閉じられると終了する。プログラム開始と同時に時間とマウスカーソル移動距離を計測し始め、終了時

にその値を記録する。移動距離は、手によって動かしたマウスカーソルの移動距離とし、マウスカーソルが本システムの利用によって自動的に動いた移動距離は含まれない。図 8、図 9 は評価プログラムが終了するまでの 5 回の平均所要時間・平均移動距離を、システムを利用した場合と利用していない場合とで比較した結果を示したものである。

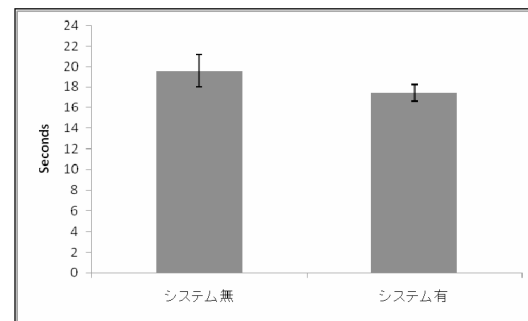


図 8 平均所要時間の比較 エラーバーは標準偏差を表す

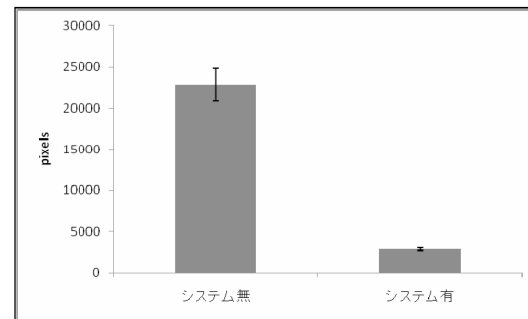


図 9 平均移動距離の比較 エラーバーは標準偏差を表す

図 8 の様に、本システムを利用した場合の、評価を完了するまでの所要時間は僅かではあるが短縮され、マウスカーソルを移動させる時間は短縮されたと言える。また、

図9の様に, 手によるマウスカーソルの移動距離を約12%に減少させることができた. このことから, ディスプレイを跨いでマウスカーソルを大きく動かすような場合において, 特に本システムの利用が有効であると考えられる.

さらに, マウスカーソルの移動パターンについて最も違和感なくマウスカーソルが移動するものを選択してもらったところ, 使用者によってばらつきがあることがわかった. これは, 移動パターンのそれぞれに適した場面が存在するためと考えられる. 画面中央はマウスカーソルが移動した後に, 手による移動によってディスプレイ上のどの場所にも移動させやすい事. Y 座標を維持, XY 座標を維持するとマウスカーソルがジャンプした後, 視線がマウスを捕らえ易く, 顔の動きによってウィンドウを移動させた際に, ウィンドウがディスプレイ内からはみ出さない事. 履歴を使用すると, 移動前に使用していたアプリケーションを素早く操作できることから, 例えば, 文書ファイルの作成時にウェブの文章をコピー&ペーストする際, マウスカーソルの移動をスムーズに行うことができる. 以上のような理由から, マウスカーソルの移動パターンは利用する場面と個人の嗜好によって様々である. このカーソルの移動パターンをシステムによってユーザに適切に場合分けすることができれば, より使いやすいインターフェースになると考えられる. また, 自由意見として, マウスの移動する位置がもっと細かく設定できたほうが良いという需要があった. 現在の実装では, 更に細かく方向を検出することは難しいため, 事前にキャリブレーション等を行い, 検出精度を高める方法によって可能であると考えられる.

7. おわりに

本研究では, マルチディスプレイ環境における並行作業を支援するシステムとして, 顔の方向とマウスカーソルを併用した入力インターフェースを提案した. 予備実験によって顔の方向と注視ディスプレイとの相関関係を調査し, 手軽に利用できる Web カメラを使用して実装を行った. 本研究中に行った評価実験において, マウスカーソルの移動時間・移動距離共に短縮され, 本システムの利用による有効性が確かめられた. しかしながら, 本システムのようにマウスカーソルがディスプレイ間をジャンプして移動するため, 顔の方向認識の誤りによってマウスカーソルを見失うという問題は依然として残っている. さらに, マウスカーソルの移動位置を細かく設定できる方が良いという意見も多く, 顔の方向の認識率をさらに高くしていかなければならない. そのため, 特定の人物を撮影できるという特徴を生かし, 事前にキャリブレーションを行う手法や, カメラを2台した三角測量による位置測量等の手法について検討していく予定である.

また, マウスカーソルの移動場所に関しても適材適所であり, ユーザの利用シーンに応じて変更していく必要がある. よって, マルチディスプレイ使用時のマウスカー

ソルの操作, ウィンドウの配置の仕方等, 細かく分析・調査していく必要がある. このような, ユーザの嗜好に合わせてマウスカーソルが移動するシステムが, 結果的に操作性を快適なものし, 生産性を向上させるか, またはマウスを見失う事に繋がるのかを今後検討していきたいと考えている.

参考文献

- 1) Mark Ashdown, Kenji Oka, and Yoichi Sato, "Combining Head Tracking and Mouse Input for a GUI on Multiple Monitors", Extended Abstract of ACM CHI 2005, pp. 1188-1191, April 2005.
- 2) Ohno Takehiko, "Eye Mark Window Management Environment", Technical report of IEICE. HIP 99(291), 17-24, 19990913
- 3) Ohno Takehiko, "Quick Menu Selection Task with Eye Mark", Transactions of Information Processing Society of Japan 40(2), 602-612, 19990215
- 4) 美原義行, 田中利治, 柴山悦哉, 佐藤周平, "マルチディスプレイ環境のためのハイブリッド型カーソル移動方式", インタラクション 2006 論文集 65~66
- 5) JPR Special Report: Multiple Display Market and Consumer Attitudes
<http://www.jonpeddie.com/special/MultDisp.shtml>
- 6) The Virtues of a Second Screen - New York Times
<http://www.nytimes.com/2006/04/20/technology/20basics.html?ei=5090&en=6fc17b9bf54ea2ef&ex=1303185600&adxnml=1&partner=rssuserland&emc=rss&adxnml=1145537733/-Kdyvqpu0/eVBVNBNUcsqg>
- 7) 飯村葉子, 金子賢一, 甲藤二郎, 小林哲則 "監視カメラ映像からの実時間顔検出手法の検討", 第3回情報科学技術フォーラム講演論文集, 2004
- 8) 武岡さおり, 尾崎正弘, 川田博美, 足達義則 "学習者認識のための顔画像検出と顔の向き認識の基礎的実験", 名古屋女子大学紀要 49 (人・社) 129~136 2003
- 9) 武岡さおり, 尾崎正弘, 足達義則 "個人認証のための顔画像抽出と顔方向の自動認識", 名古屋女子大学紀要 50 (人・社) 145~151 2004
- 10) opencv.jp - トップページ -
<http://opencv.jp>