

携帯端末を用いたマルチディスプレイ間での コンテンツ操作手法

松元崇裕[†] 瀬古俊一[†] 青木良輔[†] 井原雅行[†] 小林透[†]

テーブル型や壁面型、タブレットやスマートフォンなど大小様々なディスプレイが偏在するマルチディスプレイ環境においては、コンテンツを複数ディスプレイ間でシームレスに移動できる事がユーザビリティの向上に繋がる。本研究では、カメラと加速度・地磁気センサを持つ携帯端末を用いて、ユーザがディスプレイへ端末を向けるだけでコンテンツの操作やディスプレイ間の移動が可能になる手法を提案する。本手法では、環境中にカメラや位置計測のセンサの設置やディスプレイの位置・向きの事前登録をせずに、複数ディスプレイ間でのシームレスなコンテンツ操作を実現する。検証実験の結果、ユーザビリティに大きく影響を与える操作中の応答時間は71.3msec、ポインティング時の精度は正確な方位角取得時に2.15度となり、マルチディスプレイ環境における本手法の有効性を示した。

Digital Object Movement in Multi Display Environments Using the Mobile Devices

TAKAHIRO MATSUMOTO[†] SHUNICHI SEKO[†]
RYOUSUKE AOKI[†] MASAYUKI IHARA[†] TORU KOBAYASHI[†]

Currently, many types of display were proposed like table type, wall type, tablet pc and smartphone and Multi-display environments are becoming common. In that environment, it is required for user be able to move digital objects from one display to another. In this paper we describes cross-display pointing techniques by mobile devices that have camera and direction sensor. Our approach does not set any kind of sensors in environment like camera or position tracking sensor and does not need of pre-registration of display position or orientation. We implemented a prototype system and examined the pointer position update time and the accuracy of pointing direction. The results were 71.3 msec update time on average and 2.15 degree at a maximum error. That showed effectiveness of our approach.

1. はじめに

現在、タブレット・スマートフォン・ノートPC・テレビなど大小様々なディスプレイを有する端末が普及している。その中で、一人が複数の端末を連続的に利用したり、同時に利用したりする場面が増えている[2]。また、最近では2K・4Kの大型テレビの販売が始まっており、今以上に画面の大型化や高解像度化が進む事が予想される。さらにLife Wall[4]のような壁面型ディスプレイやMicrosoft Surface[5]のようなテーブル型ディスプレイが安価に設置可能となれば、今後はオフィスや家庭などの一般環境でも、マルチディスプレイ環境化がより進んでいくと考えられる。

図1に示すようなマルチディスプレイ環境においては、各端末を各端末内に閉じてそれぞれ単独で利用するだけではなく、各端末のディスプレイ領域を1つの共有領域として利用することが考えられる。そうすることで、ユーザは自由にコンテンツをディスプレイ間で移動させて、状況に応じてディスプレイリソースを活用することが可能となる。

例えば、会議において各自のノートPCと壁面型ディスプレイある環境では、各個人が自身のノートPC内の資料を、話題に応じて壁面ディスプレイへ移動させ皆で共有することが出来る。また、会議の中で比較する資料を並べたり、内容に応じて資料の表示先を変えグループ分けを行っ

たりなど、会議の参加者間のより円滑な意識共有に利用できる。

上記のような、マルチディスプレイ間での自由なコンテンツ操作を実現するには、各デバイスのハードウェア特性やOSの違いを吸収する事と、ディスプレイ間でコンテンツ操作を自由に行えるポインティング手法が求められる。

前者においては、現在HTML5がブラウザを通じた仮想的でプラットフォーム非依存のアプリケーション実行環境として注目されて始めている[29]。そこで本論文では、後者のマルチディスプレイ環境における、シームレスなコンテンツ移動の為のポインティングに着目する。

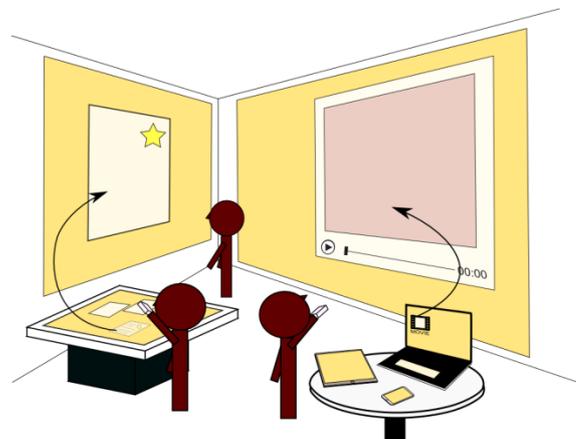


図1 マルチディスプレイ環境

[†] 日本電信電話株式会社 NTT サービスエボリューション研究所
NTT Service Evolution Laboratories, NTT Corporation

マルチディスプレイ間のポインティング実現には、ユーザがポインティングしているディスプレイ自体とその座標位置をシステムへ認識させる必要がある。そのため、従来手法の多くは、ディスプレイ設置位置の事前登録を必要としたり、操作端末や操作者をセンシングする為のセンサを環境に設置する必要があった。しかしながら、登録やセンサ設置は、家庭やオフィス等の一般環境へ適用する際の大きな負荷となってしまう。

一方で、環境にセンサを設置せずに、携帯端末のカメラでポインタ操作を行う手法も提案されている。しかし、カメラを使う手法は画像処理による計算量が多く、ポインタ位置のフィードバックにおけるリアルタイム性に課題が残っている。

そこで本論文では、端末をディスプレイへ向けるだけでポインタ操作をシームレスに実現可能な手法を提案する。本手法は環境中にセンサを設置する必要はなく、ポインタ位置の視覚フィードバックも高速に実現できる。本提案ではカメラと加速度・地磁気センサを持つ携帯端末をポインタ操作に用い、カメラによる端末-ディスプレイの位置関係のキャリブレーションと、キャリブレーション結果に基づいた加速度・地磁気センサによるポインタ操作を行う。

本稿において、2章では本分野の関連研究とその課題点に関して述べる。3章では本提案の機能要件とシステム構成に関して述べる。4章では本提案システムの精度・速度評価の実験・結果に関して述べ、5章では実験に対する考察、6章では本稿のまとめを述べる。

2. 関連研究

現在まで、各端末のディスプレイを連携・動作させるマルチディスプレイ環境構築の研究・開発が行われている[3][9][28]。また、マルチディスプレイ環境においてユーザがウィンドウやファイルなどのコンテンツを操作する手法に対しても現在まで様々な研究が行われてきた[20]。

[20]では、マルチディスプレイ環境でのコンテンツ操作手法として、ディスプレイの位置関係を考慮しない・する手法の大きく2つに分類している。

2.1 ディスプレイ位置関係を考慮しない手法

位置関係を考慮せずにオブジェクト操作を行う手法としては[10][12][16]などが存在する。位置関係を考慮しない手法では、コンテンツを移動させるディスプレイ候補をディスプレイ名やアイコンによるリストで表示する。そしてユーザはリストから移動先を選択することで、コンテンツを選択されたディスプレイに移動させることができる。

上記のような位置関係を考慮しない方法は、例えば別の部屋のディスプレイとのやり取り等、離れた空間同士でのコンテンツのやり取りには効果的に利用できる。

しかしながら、今回想定するマルチディスプレイ環境のように、1つの空間に多数のディスプレイが存在する環境

では効果的ではない。例えば目の前の壁面ディスプレイのコンテンツを手元のテーブル型ディスプレイへ移動させる際に、位置関係を考慮しない手法では、目の前のコンテンツがリストのディスプレイ群内のどのディスプレイに属し、ディスプレイ中のどのコンテンツが目の前のコンテンツか、というマッチングを行わなければならない。また移動先のディスプレイ内における、コンテンツの位置まで調整したい場合には、別の手法によりコンテンツ位置の操作を行う必要が生じてしまう。

2.2 ディスプレイ位置関係を考慮する手法

ディスプレイ間の位置関係を考慮する手法は、直接コンテンツに触れて操作を行う手法と、離れた距離から間接的に操作を行う手法に分類できる。

2.2.1 直接コンテンツに触れて操作を行う手法

直接コンテンツに触れて操作を行う手法では、移動するコンテンツとコンテンツの移動先をペン型の端末で触れることで選択する Pick&Drop[28]やタッチパネル間においてタッチ操作でコンテンツをやり取りする手法[23]がある。[23][28]の手法では、ユーザは手の届く範囲において、極めて簡単に操作ができる利点がある。しかしながら、離れた場所のコンテンツを移動するためには、ユーザはコンテンツ位置まで移動する事が必要であるという問題がある。

2.2.2 間接的にコンテンツ操作を行う手法

間接的に操作する手法では、更に相対ポインティングと、絶対ポインティングの2種類が存在する。

2.2.2.1. 相対ポインティング手法

相対ポインティングとは、手元のデバイス操作量に応じてマルチディスプレイ側のカーソルの移動量が相対的に変化する手法を指す。間接的に相対ポインティングを行う手法で代表的な手法として、マウスを用いる手法が挙げられる[9][21]。

しかしながら、マルチディスプレイ環境における相対ポインティングは、カーソルの位置が現在どのディスプレイ上のどの位置に存在するのか見失いやすい問題がある。

2.2.2.2. 絶対ポインティング手法

絶対ポインティングは、ポインタ操作の携帯端末や手によるジェスチャで指し示す先に、ポインタが表示される手法である。そのため、絶対ポインティング手法はディスプレイの数が多くなったり、ディスプレイのサイズが大きい場合でもポインタ位置が認識し易い利点がある。

絶対ポインティングによるマルチディスプレイ環境におけるポインタ操作手法は[7][8][17][22][24][25][28]など、現在まで多くの研究が行われてきた。

[8][28]では、レーザポインタによりポインタ操作を行う手法が提案されている。[28]では環境に設置されたカメラにより、レーザポインタの照射されたディスプレイ面を撮影することで、照射された場所のディスプレイ座標の取得を行う。一方、[8]の手法ではディスプレイ側に設置された

カメラから、ディスプレイへ向けられたレーザーポインタの方向を計測し照射されたディスプレイ座標の取得を行う。

[22]では三次元位置計測機器によるポインティング手法が行われている。操作する端末とユーザ頭部にマークを取り付け、端末と頭部の位置・方向を環境中に設置した三次元位置計測機器により計測する。端末と頭部の位置・方向計測結果と登録されたディスプレイ位置・向き情報からポインティングを実現している。

更に、加速度・地磁気センサ・LED ライトを持つ携帯端末によりポインティングを操作する手法としては[7]が提案されている。[7]の手法では加速度・地磁気センサより端末の向いている方位角を取得する。そして LED を環境中に設置した複数のカメラから撮影することで、操作端末の空間座標を計測し、予め登録されたディスプレイ位置・向きからポインティング先を計算する。

上記の例の様に、絶対的なポインティング手法の多くは環境中にカメラや3次元位置計測をするための機器設置や、ディスプレイの位置・向き情報の登録が必要となる。

一方で、最近では環境中に機器を設置せずにカメラ画像とディスプレイ画像を用いて、画像処理によるポインタ操作手法が提案されている[17][24][25]。しかしながら、画像処理による手法は計算量が多いため、ポインタの視覚フィードバックにおけるリアルタイム性に問題を生じてしまう。更に、画像処理計算量は、画像枚数や画面解像度、画像中の特徴点数に大きく影響を受けてしまう。そのため[24]の従来研究でも、320x240 の解像度で中程度の特徴点数である画像を表示するディスプレイ一台でも、325ms ものレスポンス時間が必要となる。

[26]の研究では、ポインタ操作における視覚フィードバックの遅延は 100~200msec までの間であれば、操作に与える影響は僅かであるが、250msec を超えると急激に操作精度へ悪影響を与える事が示されている。その為、画像処理によりポインタ操作を行う手法ではポインタ操作時のフィードバック遅延の大きさが問題となる。

上記の例の様に、絶対的なポインティング手法は環境中に多くの機器を設置するかディスプレイの位置・向き情報を事前に登録するか、機器設置や事前登録は必要ないがリアルタイム性を犠牲にするかのいずれかデメリットを選ばなければいけないという問題がある。

3. 提案システム

3.1 機能要件

本研究では1つの部屋に多数のディスプレイが存在する環境において、ディスプレイ間におけるコンテンツのシームレスな移動操作の実現を目的としている。また家庭や会議室といった一般環境でも容易に操作環境を実現でき、ユーザに対して容易・快適なコンテンツ操作を実現する。

上記目的の達成にむけて従来研究の課題をまとめると、本

研究で同時に達成すべき機能要件は次の4つとなる。

- ① カメラや位置測定のセンサなど、環境中に特別な機器を設置せずともポインタ操作が実現できること
- ② ディスプレイの空間位置をユーザが予めシステムへ登録せずともポインタ操作が実現できること
- ③ ユーザがコンテンツから離れた位置からでも、絶対ポインティング操作を実現できること
- ④ ポインタ操作時にユーザへリアルタイムな操作の視覚フィードバックを実現する事

本研究では上記で述べた4つの機能要件のうち、①②③の要件を満たすが④は満たさないカメラによる画像処理の手法と、③④の要件を満たすが①②に課題の残る加速度・地磁気センサの手法に着目した。本提案では両手法を組み合わせる事で全要件を実現する新しい手法を提案する。

3.2 提案手法

本提案手法は大きくキャリブレーションとポインタ位置決定の2つから構成される。キャリブレーションでは、端末カメラ画像・撮影時の端末方位角・操作対象の全ディスプレイ画像の3つを入力として、端末の方位角から直接ディスプレイ座標への変換を行う変換行列を算出する。一方、ポインタ位置決定では端末の方位角のみを入力としてキャリブレーション結果の変換行列を参照し、ポインティングされたディスプレイとその座標を求める。

しかしながら、キャリブレーション結果の変換行列は、携帯端末位置が移動すればするほど、端末を向けた方向と実際のディスプレイ座標との間で齟齬が生じてしまう。そこで、キャリブレーションはポインタ位置決定と別スレッドで駆動させ、ポインタ位置決定よりも低い優先度で繰り返し実行される。

本手法により、ユーザは離れた場所であっても携帯端末のカメラを向ける事で、向けた先のポインタ操作が可能となる。また、本手法は変換行列をカメラ・ディスプレイ画像のみから算出する。そのため、本手法では環境中にセンサを設置せずに端末のポインティング先の座標を決定する事が出来る。またキャリブレーションとポインタ位置決定を別スレッドとすることで、システムが時間の必要なキャリブレーション中であっても、ポインタの位置決定は方位センサデータと前回のキャリブレーション結果の変換行列により近似的に求め続ける事ができる。従って、ポインタ位置の更新は、ディスプレイ枚数の増加やサイズの増加による画像処理の処理量に関わらず、高速に実行することが可能となる。

3.3アーキテクチャ

図2に本提案手法におけるディスプレイ上におけるポインタ操作部分のシステム構成図を示す。

本システムは入力として操作対象となる全ディスプレイ画像、端末カメラ画像、端末センサ方向情報、端末操作情報を受け取る。端末操作情報は携帯端末に対するタップ

1. 操作により実現し、マウスのクリック操作に対応する。端末操作情報は携帯端末に対するタップ操作により実現し、出力としては、端末が向けられたディスプレイに対して、ディスプレイ座標におけるポインタ位置とクリック・ドラックなどのポインタ状態がポインタ位置・操作命令として送信される。

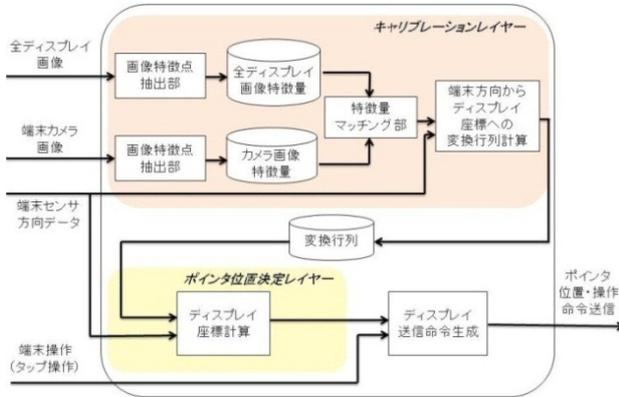


図 2 提案システム構成図

3.3.1 特徴点抽出部

全ディスプレイ画像と端末カメラ画像の入力に対し、特徴点抽出部ではカメラ画像内にどのディスプレイがどの位置に表示されているかのマッチングを行うための局所特徴点検出と特徴量抽出を行う。

画像中の特徴点検出において、として SIFT[11]や SURF[14]・Fast[13]・Star[18]特徴量が広く知られている。また、特徴量抽出では、回転や拡大・縮小に頑強な特徴量として SIFT・SURF・Freak[6]特徴量などが知られている。本システムでは、特徴量抽出においては、精度・速度の面でリアルタイムの特徴量抽出に向いている FREAK を利用した。特徴量抽出器におけるパラメータにおいては OpenCV におけるデフォルトのパラメータを利用している。また、特徴点検出手法に関しては、SIFT・SURF・Fast・Star の中で、予備実験において最も精度の高かった SURF を用いた。特徴点検出器に関するパラメータもデフォルトの値を用いた。

3.3.2 特徴量マッチング部

特徴量マッチング部では、カメラ画像・各ディスプレイ画像から求められた特徴量に対して、マッチングを行う。

マッチングでは、まずカメラ画像内にどのディスプレイ画像が撮影されているか、又は全く撮影されていないかの判定を行う。そして、撮影されている場合にはカメラ画像とディスプレイ画像間で対応する特徴点を抽出する。

本手法においては、扱う画像がカメラとディスプレイ画像であり、ディスプレイの表示画像は常に変化する可能性がある。そのため、カメラによる通常の物体認識の様に事前に学習器を作成しておく事ができない。そこで本手法ではマッチングを非常にシンプルな手法で実現した。

まず、カメラ側の特徴点と各ディスプレイの特徴点に対

し局所特徴量同士のマッチング手法である NDDR 法の適応を行った[15]。NDDR 法では、カメラ画像側の特徴点に 1 つに対し各ディスプレイ画像の特徴点で、最も特徴量ベクトルのユークリッド距離が近い特徴点と 2 番目に近い特徴点を選択する。そして、最も近い特徴点間のユークリッド距離を D_A 、2 番目に近い特徴点間のユークリッド距離を D_B としたとき、特徴点間の不確かさを表すスレッシュホールド R に対して $D_A/D_B < R$ の条件を満たす特徴点ペアを正しい対応点として選択する。NDDR を用いる事で単純にユークリッド距離が近い対応点のみを選択するよりも高い精度で対応点を抽出できる。

特徴点对は、本システムにおいて次の 2 つの目的で用いられる。①特徴点对の数から、カメラ画像に対してどのディスプレイ画像が撮影されているか又は撮影されていないかの判定を行う。②特徴点对よりディスプレイ画像からカメラ画像への射影変換行列を求める。そこで、2 つの目的にスレッシュホールド $R_d \cdot R_t$ をそれぞれ設定した。

カメラ画像とディスプレイ画像間における特徴点对において、 $D_A/D_B < R_t$ を満たす特徴点对の合計数を全ディスプレイ画像に対して求める。そして、合計数が最も多いディスプレイ i の合計数を S_i と定義する。この時、カメラ画像内にディスプレイ i が、本当に存在するか否かを判定するスレッシュホールド S_t に対して、 $S_i > S_t$ を満たすならば、ディスプレイ i はカメラ画像に存在すると判定する。一方で、 $S_i > S_t$ を満たさない場合には、カメラ画像内にディスプレイは存在しないと判定され、再度新しいカメラ・ディスプレイ画像により計算される。

最後に、 $S_i > S_t$ を満たすカメラ画像とディスプレイ i 画像間の特徴点对のうち $D_A/D_B < R_t$ を満たす特徴点对を、変換行列計算部へ送る。

3.3.3 変換行列計算

本節では、特徴量マッチング部によって求められたカメラ・ディスプレイ画像間の特徴点对と、撮影時の端末方向を元に、端末方向からディスプレイ座標への変換行列を求める手法を述べる。

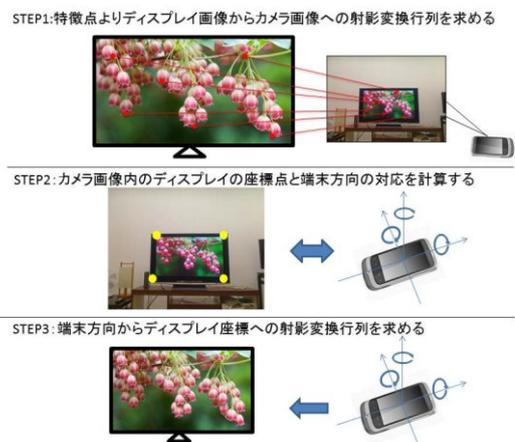


図 3 端末方向からディスプレイ座標への変換行列計算

まず、図3に変換行列計算のプロセスを示す。図3に示す様に、変換行列計算のプロセスは3STEPで実行される。

STEP1では、計算された特徴点のペアより、ディスプレイ画像座標からカメラ座標への射影変換行列を求める。この時、特徴点のペアにはノイズとなる誤った対応関係の点対が含まれるため、ノイズを考慮した射影変換行列の計算が必要となる。従来研究の[17]では2つの画像間の射影変換行列を求める際に、ノイズの含まれた対応関係に対してRANSACアルゴリズムを用いる事で誤った特徴点対の影響を減じている。そこで、本手法においても同手法によりディスプレイ画像座標からカメラ画像座標への射影変換行列を求めた。

次にSTEP2における、カメラ画像内のディスプレイ座標点と端末方向の対応計算について述べる。STEP2での目的は、端末カメラの光学中心軸がディスプレイ平面と交わる点におけるディスプレイ座標と、その際の携帯端末の方向センサ値との対応関係を4点求める事である。

STEP2では、まず射影変換行列よりディスプレイ座標における任意の点 (x',y') に対応するカメラ座標点 (x,y) を計算する。次にカメラ座標点 (x,y) に対して、カメラの光学中心軸座標 (θ_x, θ_y) を (x,y) へ合わせるように端末を回転させた場合のパン・チルト回転角 $(\theta_\alpha, \theta_\beta)$ を求める。このとき、カメラの横方向・縦方向の画角を $\theta_x \cdot \theta_y$ 、カメラの横方向・縦方向のピクセル数を其々 $N_x \cdot N_y$ とすると、あるカメラ座標 (x,y) に対して携帯端末カメラの光学中心軸座標を向ける時のパン・チルト回転角 $(\theta_\alpha, \theta_\beta)$ は式(1)・(2)のように計算することができる。図4に上記のカメラ・ディスプレイ・携帯端末向きの変数の幾何関係を示す。

$$\theta_\alpha = \tan^{-1} \left(\frac{2 \tan \frac{\theta_x}{2} |x - \theta_x|}{N_x} \right) \quad (1)$$

$$\theta_\beta = \tan^{-1} \left(\frac{2 \tan \frac{\theta_y}{2} |y - \theta_y|}{N_y} \right) \quad (2)$$

さらに式(1)・(2)により求めたパン・チルト回転角 $(\theta_\alpha, \theta_\beta)$ に向けて端末を回転させた際に、携帯端末の加速度・地磁気センサから取得される方位角を求める。この時の方位角のパン・チルト値を $(\theta'_\alpha, \theta'_\beta)$ と置く。すると、方位角のロール方向がカメラ座標平面と平行に回転する場合、 $(\theta'_\alpha, \theta'_\beta)$ はキャリブレーション時のパン・チルト・ロール角 $(\theta_\alpha, \theta_\beta, \theta_\gamma)$ から、式(3)(4)のように計算できる。

$$\theta'_\alpha = \theta_\beta + \theta_\alpha \cos(-\theta_r) - \theta_\beta \sin(-\theta_r) \quad (3)$$

$$\theta'_\beta = \theta_t + \theta_\alpha \sin(-\theta_r) + \theta_\beta \cos(-\theta_r) \quad (4)$$

上記の処理により、携帯端末の光学中心軸をディスプレイ座標 (x',y') へ向けた際に得られる端末のセンサ値 $(\theta'_\alpha, \theta'_\beta)$ を計算することが出来る。

(x',y') をディスプレイの任意の4点とすることで、ディ

スプレイ座標と携帯端末の方向センサ値との対応関係を4点得る事が出来る。本研究では、ディスプレイの4隅の座標点を任意の4点として用いた。

最後にSTEP3ではディスプレイ座標と携帯端末の向きの4つの対応点から、携帯端末方向からディスプレイ座標への射影変換行列 H を計算する。 H はセンサ値により求まるパン・チルト角 $(\theta'_\alpha, \theta'_\beta)$ に対するディスプレイ座標 (x',y') において(5)式の条件を満たす 3×3 の行列である。

$$[x', y', 1]^T = H[\theta'_\alpha, \theta'_\beta, 1]^T \quad (5)$$

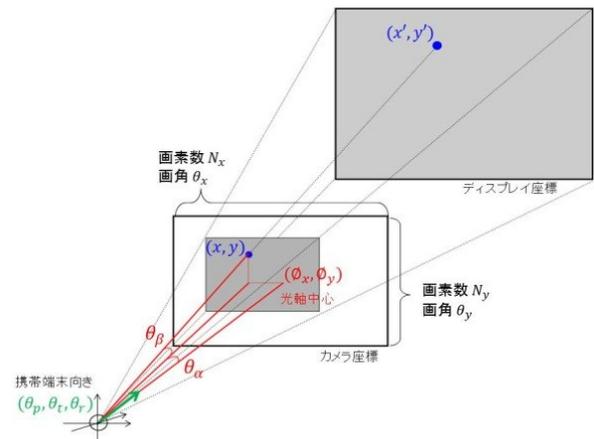


図4 ディスプレイ・カメラ・携帯端末方向の幾何関係

3.3.4 ディスプレイ座標計算・操作命令送信

ディスプレイ座標計算部では、携帯端末の加速度センサ・地磁気センサより取得された方位情報をリアルタイムで受け取り、変換行列を元にディスプレイ座標へ変換する。

また操作命令送信では、変換されたディスプレイ座標へマウスカーソルを移動させる命令と、マウスボタンの押し・リリースといったユーザ操作情報の命令をディスプレイ i に対し送信する。

4. 実験

4.1 実装アーキテクチャ

本研究において、4章で提案したシステム部分の実装はC++のOpenCV 2.4により行った。また、今回システムの実装はMacBook Pro OSX 10.8.2で行われ、PCのCPUはQuad core 2.7GHz Intel Core i7、メモリは16GB 1600Mhzの物を用いた。また、携帯端末にはNTTドコモ GALAXY S III SC-06Dを用いている。また今回は携帯端末側のカメラとして、携帯付属のカメラではなく一般的なwebカメラであるLogicool WebCam Pro 9000 QCAM-200SXを用いた。カメラの解像度は 1600×1200 である。

4.2 精度検証実験

システムの精度検証を行うため、本研究ではディスプレイ・カメラ間における、テストセットを用意した。まずディスプレイ上に画像を表示し、スクリーンの正面から $1m \cdot 2m \cdot 3m$ の距離よりカメラによる撮影を行った。図5にス

クリーン画像と各距離からの撮影画像の例を示す。表示用のディスプレイには Panasonic VIERA TH-42PZ800(42 インチ)を用い、解像度は 1980×1080 で行った。

図 5 に示すデータセットをディスプレイに表示する画像を変え、15 セットで撮影を行った。表示する画像には Mac OSX に予め用意されているデスクトップピクチャの中からランダムで 15 枚を選択した。また、テレビからテレビへ移動する際に生じる状況のテストセットとして、ディスプレイが全く移っていないカメラ画像を 10 枚撮影した。各カメラ画像・ディスプレイ画像における特徴点の検出数の平均はそれぞれ 2427 と 4380、標準偏差が 2152・5148 である。



図 5 テストセット画像例

(左上：スクリーン画像・左下：カメラ画像(1m)・右上：カメラ画像(2m)・右下：カメラ画像(3m))

精度検証の為、スレッシュホールド値 $R_d \cdot R_t \cdot S_t$ の決定をテストセットより行う。 $R_d \cdot S_t$ はカメラによる撮影画面と対応するディスプレイを決定する値である。また、 R_t はカメラ撮影画像と対応するディスプレイ画像決定後、両画像間の変換行列計算の精度に影響するスレッシュホールドである。

$R_d \cdot S_t$ の値を定めるため、テストセットにおいて各 $R_d \cdot S_t$ 値に対する、カメラ画像とディスプレイ画像の対応の false-negative 値と false-positive 値を求めた。 false-negative 値はカメラ画像に対してディスプレイ画像の選択を誤る確率と選択は正しかったがスレッシュホールド S_t により適合する画像ではないと判断されてしまう確率の和となる。一方で、 false-positive はカメラ画像に対して誤ったディスプレイ画像を選択し、かつスレッシュホールド S_t に適合する画像であると判断されてしまう確率となる。上記の false-negative と false-positive に対して F 値が最も小さくなる $R_d \cdot S_t$ を選択する。両パラメータにおいては、 R_d は 0.0 から 1.0 まで 0.01 の刻みで探索し、 S_t は各 R_d において F 値が最も小さくなる整数値を選択した。探索においては各ディスプレイ画像に対し、1m・2m・3m の各距離から撮影した 3 枚のカメラ画像を正解画像、その他の 42 枚+10 枚のカメラ画像を失敗画像とし、全ディスプレイ画像において探索を行った。

探索の結果、 $R_d = 0.66 \cdot S_t = 7$ の際に false-negative と false-positive 最も F = 0.102 と F 値の値が最小となった。表 1 に操作ディスプレイの数がそれぞれ 1 枚、3 枚、5 枚の場

合における false-negative, false-positive, F 値を示す。

表 1 ディスプレイ探索時の精度

画面数	false-negative	false-positive	F 値
1	0.178	0.047	0.112
3	0.220	0.138	0.179
5	0.260	0.220	0.240

次に、本システムのポインティング向きの精度は STEP2 で示した変換行列計算時におけるディスプレイとカメラの対応点の精度と、携帯端末の方位計測の精度によって定まる。そこで、ポインティング向きの検証では、まずディスプレイ - カメラ間における対応点の取得精度に対して検証を行う。

R_t 決定において、カメラ画像とディスプレイ画像の正しいペアが与えられた際に、カメラとディスプレイの間における対応点の取得精度において、最も高い精度となるように R_t 値を決定する。検証のため 15 セット・計 45 枚のカメラ画像に対し、ディスプレイの 4 隅にあたる座標点位置の解答を人手により作成した。検証では、解答の座標点とシステムにより計算されたディスプレイ 4 隅の座標点の距離の大きさによって検証を行う。検証時に R_t 値を 0.0 から 1.0 までの間で 0.01 の刻みで変化させた。

検証の結果、 $R_t = 0.82$ の際に最も対応点の対応精度が高く、対応点の変換行列が正しく生成された確率は 0.96、また成功したものの中でカメラ座標における正しい座標点と推定座標点の平均座標距離は 42.3 となった。42.3 ピクセルの差異はポインティング方向の最大誤差において 2.15 度に相当する。

4.3 速度検証実験

実際の操作時におけるポインタのフィードバック速度の計測実験を行った。検証環境では図 6 に示すような 3 画面環境で行い、操作対象のディスプレイとして Panasonic VIERA TH-42PZ800(42 インチ)、Panasonic TH-L32X2(32 インチ)、MacBookPro Retina Display (15 インチ)を用いた。各解像度は 1980×1080, 1980×1080, 1440×900 である。各ディスプレイは 1 つの PC へ HDMI により接続されており、一つの PC のマルチスクリーン環境として実現される。また、本提案システムの処理部は同一の PC 内に実装されている。操作側の携帯端末と提案システムは無線 LAN により接続され、方位情報は無線 LAN を経由して提案システムへ送られる。携帯端末における方位情報取得の遅延時間は平均 60(msec)により実行される。また、携帯端末の web カメラは USB2.0 で PC と接続され、本提案システムへ画像情報が送信される。

上記の環境において、本提案システムのキャリブレーション部分のキャリブレーション時間と、ポインタの位置・操作が更新時間の計測を行った。

キャリブレーション時間では、特徴点抽出時間・特徴点

マッチング時間・変換行列計算時間の3つに関して計測を行った。また、ポインタ位置・操作の更新時間では直前のポインタ位置が決定されてから、次のポインタ位置が決定するまでの間隔を測定した。

各計測時間に関する、平均時間と標準偏差を表2・表3にまとめる。本計測においてキャリブレーション時に検出されたカメラの特徴点数の平均は1552.6、標準偏差は565.1である。また、ディスプレイの背景画像は実験中ディスプレイ毎に共通のものを利用し、ディスプレイ3枚の特徴点数合計の平均は8435.9であった。



図 6 速度検証における実験環境

表 2 キャリブレーション時間

	特徴点 抽出時間	特徴点 マッチング時間	変換行列 計算時間
平均	492.0 (msec)	1635.8 (msec)	5.2 (msec)
標準偏差	197.6 (msec)	35.9 (msec)	4.9 (msec)

表 3 ポインタ位置・操作の更新時間

	ポインタ位置・操作更新時間
平均	71.3 (msec)
標準偏差	7.0 (msec)

5. 考察

5.1 精度検証実験に関する考察

ディスプレイ探索時の精度に関し false-negative は、携帯端末がディスプレイへ向けられているにも関わらず認識されない誤りである。その為、false-negative が生じた場合は再度キャリブレーションを行う必要がある。一方で、false-positive はポインタが端末方向と関係無いディスプレイに表示されてしまう誤りとなる。そのため、本提案手法において false-positive はユーザにとってクリティカルな誤りとなる。従って、今回の精度結果における false-positive の精度は未だ実用上の問題が残る。精度を落とす大きな要因として、3m 離れた位置からの画面認識の精度が挙げられる。今回のテストセットにおいて、3m 距離からの認識を結果は $R_d = 0.66 \cdot S_t = 17$ の際に F 値が最少になり表 2 の

ようになった。

表 4 1・2m でのディスプレイ探索時の精度

画面数	false-negative	false-positive	F 値
1	0.167	0.0027	0.084
3	0.212	0.0077	0.110
5	0.255	0.0128	0.134

3m 距離からの認識が失敗する要因として、画面以外の特徴点がノイズとして働く事が挙げられる。その為、精度向上の為には[27]の様に、カメラ写真内のテレビ画面領域を高速に写真から抽出する手法が有効だと考えられる。

また、変換行列の精度として最大誤差の 2.15 度は十分な精度であると考えられる。操作時の精度は変換行列の精度とセンサによる方位角取得の精度の和となるが、近年センサ技術の発達により方位角分解能が 0.1 度、精度は 5 度未満の物も存在しており[1]、今後は更に精度が向上していく事が予想される。

5.2 速度検証に関する考察

ポインタの視覚フィードバック時間は実験結果において 71.3(msec)と目標とする 200(msec)を大きく下回り、十分なレスポンス時間を達成できた。また、本レスポンス時間は画面数に影響されないため今後マルチディスプレイ化が進んだ場合においても本手法が有効であると考えられる。キャリブレーション時間においては、合計の時間が平均 2133(msec)と多くの時間が必要となっている。本手法において、キャリブレーションの時間は最初に操作を開始まで時間と操作するディスプレイを変更した際に、ユーザが待つ時間となる。その為、現在の値よりも更なる高速化を行う事が求められる。

実験結果より、キャリブレーション時間の多くが特徴点抽出時間と特徴点マッチングによるものであることが分かる。その理由として、本実験のカメラ、ディスプレイ解像度は従来研究よりも高く、それに応じて特徴点数も多い事が挙げられる。処理時の特徴点を減らす方法として、計算時にカメラ画像・ディスプレイ画像に対してリサイズを行う事で、特徴点量を減らす方法が考えられる。しかしながら、カメラ・ディスプレイの解像度を下げるとポインティングの精度に影響を与えてしまう。そのため、今後の改善方法として操作開始時やディスプレイ変更時においては速度を優先したキャリブレーションを行いユーザが操作不可能な時間を減らし、その後に精度を優先したキャリブレーションを実行する方法が考えられる。

6. まとめ

本研究では、マルチディスプレイ環境においてコンテンツをシームレスに操作する為のポインティング手法の提案を行った。本手法では、環境中にカメラ・センサを設置しない状況においても、携帯端末側の方位センサ・カメラを用いる事で端末を向けた先にポインタを表示する絶対ポイ

ンティングを可能にした。また、ポインタの視覚フィードバックレートに関して従来手法を大きく上回る 71.3(msec)を達成し、ポインタ操作時に人へ悪影響を与える 250(msec)を大きく下回った。

参考文献

- 1) Freescale. FXOS8700CQ: 6-Axis Xtrinsic Sensor. http://cache.freescale.com/files/sensors/doc/data_sheet/FXOS8700CQ.pdf
- 2) Google. The New Multi-screen World : Understanding Cross-Platform Consumer Behavior. http://services.google.com/fh/files/misc/multiscreenworld_final.pdf, 2012.
- 3) Oblong G-speak. <http://www.oblong.com>
- 4) Panasonic. Life Wall – gesture-controlled interactive TV. International Consumer Electronics Show (CES). <http://www.panasonic.com/cesshow>, 2008.
- 5) Samsung SUR40 with Microsoft PixelSense. <http://www.samsunglfd.com/solution/sur40.do>
- 6) A. Alexandre, O. Raphael, V. Pierre. FREAK: Fast Retina Keypoint. IEEE Conference on Computer Vision and Pattern Recognition(CVPR'12). pp,510-517, 2012.
- 7) A. Wilson and S. Shafer. XWand: UI for Intelligent Space. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'03). pp,545-552, 2003.
- 8) B. A. Myers, R. Bhatnagar, J. Nichols, C. H. Peck, D. Kong, R. Miller and A. C. Long. Interacting at a distance: measuring the performance of laser pointers and other devices. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'02). pp,33-40, 2002.
- 9) B. Johanson, G. Hutchins, T. Winograd and M. Stone. PointRight: Experience with Flexible Input Redirection in Interactive Workspaces. Proceedings of the 15th annual ACM symposium on User interface software and technology(UIST'02), pp, 227-234, 2002.
- 10) B. Johanson, S. Ponnekanti, C. Sengupta and A. Fox. Multibrowsing: Moving Web Content across Multiple Displays. Proceedings of the 3rd international conference on Ubiquitous Computing(UbiComp'01). pp,346-353, 2001.
- 11) D.G.Lowe. Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision, pp,91-110.2004.
- 12) D. S. Tan, B. Meyers and M. Czerwinski. WinCuts: Manipulating Arbitrary Window Regions for More Effective Use of Screen Space. CHI '04 Extended Abstracts on Human Factors in Computing Systems (CHI EA'04). pp,1525-1528, 2004.
- 13) E. Rosten and T. Drummond, Machine learning for high-speed corner detection, Proceedings of the 9th European Conference on Computer Vision, vol.1, pp,430-443, 2006.
- 14) H. Bay, A. Ess, T. Tuytelaars and L. V. Gool. Speed-Up Robust Features (SURF). Computer Vision and Image Understanding, pp.346-359. 2008.
- 15) K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. Pattern Analysis and Machine Intelligence (PAMI), pp,1615-1630. 2005.
- 16) K. S. Booth, B. D. Fisher, C. J. R. Lin and R. Argue. The "Mighty Mouse" Multi-Screen Collaboration Tool. Proceedings of the 15th annual ACM symposium on User interface software and technology(UIST'02). pp,209-212, 2002.
- 17) L. Hebert, N. E. Pears, D. Jackson and P. Oliver. Mobile Device and Intelligent Display Interaction via Scale-Invariant Image Feature Matching. Conf. on Pervasive and embedded Computing and Communication Systems (PECCS'11). pp.207-214, 2011
- 18) M. Agrawal, K. Konolige and M. R. Blas. CenSurE: Center Surround Extremas for Realtime Feature Detection and Matching. In Proceedings of the European Conference on Computer Vision (ECCV), pp,91-110, 2008.
- 19) M. Ashdown, K. Oka and Y. Sato. Combining head tracking and mouse input for a GUI on multiple monitors. CHI '05 Extended Abstracts on Human Factors in Computing Systems (CHI EA'05). pp,1188-1191, 2005.
- 20) M. A. Nacenta, C. Gutwin, D. Aliakseyeu and S. Subramanian. There and back again: cross-display object movement in multi-display environments. HCI Journal 24, vol.1, pp,170-229, 2009.
- 21) M. A. Nacenta, S. Sallam, B. Champoux, S. Subramanian and C. Gutwin. Perspective Cursor: Perspective-Based Interaction for Multi-Display Environments. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'06). pp,289-298, 2006..
- 22) R. Fukazawa, K. Takashima, G. Shoemaker, Y. Kitamura. Y. Itoh and F. Kishino. Comparison of Multimodal Interactions in Perspective-corrected Multi-display Environment. IEEE Symposium on 3D User Interfaces (3DUI). pp.103-110, 2010.
- 23) S. Bachl, M. Tomitsch, K. Kappel and Thomas Grechenig. The Effects of Personal Displays and Transfer Techniques on Collaboration Strategies in Multi-touch Based Multi-Display Environments. Proceedings of the 13th IFIP TC 13 International Conference on Human-Computer Interaction (INTERACT'11),vol.3,pp,373-390,2011.
- 24) S. Boring and D. Baur, Making Public Displays Interactive Everywhere. IEEE Computer Graphics and Applications, 2012.
- 25) S. Boring, D. Baur, A. Butz, S. Gustafson and P. Baudisch. Touch projector: mobile interaction through video. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'10). pp,2287-2296, 2010.
- 26) S. Bryson. Effects of lag and frame rate on various tracking tasks. Proceedings of the Stereoscopic Displays and Applications (SPIE). 1993.
- 27) 北原亮, 中村高雄, 片山淳, 安野貴之, 携帯端末上における幾何補正のためのリアルタイム矩形追跡手法. 電子情報通信学会技術研究報告.. OIS, オフィスインフォメーションシステム, pp,1-6, 2006.
- 28) 暦本純一, InfoRoom: 実世界に拡張された直接操作環境. インタラクシオン 2000, pp, 9-16, 2000.
- 29) 小林 透, 瀬古 俊一, 川添 雄彦: HTML5 によるマルチスクリーン型次世代 Web サービス開発, 翔泳社, 2013.