

研究論文

マルチディスプレイシステムのための
可用性を保った端末位置推定法鈴木 広人^{1,a)} 小林 亜樹^{2,b)}

受付日 2021年2月26日, 採録日 2021年6月26日

概要: テーブル表面を大型のディスプレイとするテーブルトップディスプレイの代替として、テーブル上に仮想的な見えない画面を想定し、そこに置いたタブレット PC をいわば窓のようにしてその仮想的な大型画面を覗き見るシステムは、タブレットの画面領域、個別識別、向きの3要素を知る必要がある。これら位置推定は、初期登録時のほか物体追跡失敗時にも行う必要が生じる。これを外部カメラの撮影画像を用いて処理する既存方式では、タブレット画面に2次元コードを表示することで、2次元コード認識を利用してこれらを実現している。しかし、2次元コードは画面を占有する可視パターンであるため、この認識時間中は代替テーブルトップディスプレイとしての利用は中断し、可用性を失う。本論文では、これら位置推定期間中も表示画面を維持したままにできるパターンと認識方式を提案する。本方式は、可用性に影響しにくいと考えられる画面周縁領域のみを使用するため、表示コンテンツはほとんどそのまま維持される。実端末を利用した実験で、実用可能な認識時間と精度を達成できることを示した。

キーワード: マルチディスプレイシステム, 画面領域認識, 端末 ID 識別, 画面向き推定

Tablet Position and Orientation Estimation Method with Availability for Multi-display System

HIROTO SUZUKI^{1,a)} AKI KOBAYASHI^{2,b)}

Received: February 26, 2021, Accepted: June 26, 2021

Abstract: As an alternative to the tabletop display, the system that assumes an invisible screen on a table virtually and uses a tablet PC placed on it as a window to peek the virtual large screen needs to know the three factors of the tablet PC: the screen area, the device identification, and its orientation. These position estimates need to be performed not only at the time of initial registration, but also when object trackings fail. In a 2D code that displays on each tablet, recognition is used to achieve the purpose, in former studies that use images by an external camera. However, since the 2D code is a visible pattern that occupies the screen, its use as an alternative tabletop display is interrupted during this recognizing duration, resulting in a loss of availability. In this paper, we have proposed a pattern and recognition method that can keep the display screen during these position estimation periods. Since this method uses only the edge areas of the screen that are considered to be less susceptible to availability, the displayed content remains almost kept. The experiments using real devices showed that the recognition time and accuracy can be achieved practically.

Keywords: multi-display system, screen area recognition, device ID identification, screen orientation estimation

¹ 工学院大学大学院工学研究科電気・電子工学専攻
Electrical Engineering and Electronics Program, Graduate School of Engineering, Kogakuin University, Shinjuku, Tokyo 163-8677, Japan

² 工学院大学情報学部情報通信工学科
Department of Information and Communications Engineering, Faculty of Informatics, Kogakuin University, Shinjuku, Tokyo 163-8677, Japan

1. はじめに

多人数でのコラボレーションツールとしてテーブル表面に大型ディスプレイを埋め込んだ製品が存在する。これら

^{a)} cm19026@g.kogakuin.jp

^{b)} aki@cc.kogakuin.ac.jp

タブレットトップディスプレイと同様なことを手軽に実現させる方法として、タブレット PC 等を複数台テーブル上に並べることで仮想的な大画面を表現しようとする研究が進められている。このとき、仮想的な大画面（仮想大画面）はテーブル上に存在し、その一部をテーブル上に置いたタブレット画面に映し出すこととなり、いわば画面を窓のようにして仮想大画面の一部分を覗く関係性となる。このとき、仮想大画面は不可視キャンパス、タブレット群は可視ウィンドウと見なすことができ、不可視キャンパスを可視ウィンドウを用いて覗く関係性になるともいえる。タブレットは容易に移動できることをふまえ、タブレットの移動に従って移動先の仮想大画面内の領域を映し出すようにするのが一般的である。

そのためには、タブレットが仮想大画面のどの領域を映し出す位置に存在しているのかを識別する必要がある。このアプリケーションの代表的な研究である HuddleLamp [1] では、別に設置した管理サーバがタブレット画面上に識別用の 2 次元コードを表示させ、外部 RGB-D カメラを用いて物体認識、タブレット識別を行い解決している。ここで問題となるのが 2 次元コードの表示である。なぜなら、コード表示により画面の大部分を専有してしまうためその間、アプリケーション本来の利用が継続できないためである。この利用阻害問題は、アプリケーション起動時の初期登録時にはほぼ問題とならない。しかし、利用中にタブレットの位置等を見失う状況^{*1}になると、再度認識する必要が生じる。このとき、できるだけアプリケーション利用を阻害しない手法が望ましく、かつ、機材も一般的なものであるほど望ましい。

そこで本論文では、RGB-D カメラよりも一般的な単眼カメラ（普及した PC Web カメラ等）での撮影画像中で、タブレット画面を同定、識別するパターン表示、検出方式を提案する。これは、本来の表示コンテンツを表示したままヒトが認知しにくいパターンを重畳表示する枠組みであり、その意味では電子透かしと似ている。しかし、電子透かしの応用 [2], [3], [4], [5], [6] は多くが著作権管理であるため、パターンの存否を認知させないことをパターンに要求するのに対して、本研究では、表示コンテンツをそれなりに認知できればパターンの存在自体は多少知覚できても構わない点が異なる [7]。また、1 枚の画像中に複数台のタブレット画面が混在し、これらを互いに識別することを目的とする点も電子透かしとは異なる応用目的である。一方、管理サーバからの指示で（多少の遅延を考慮する必要はあれど）パターン表示をタブレットごとに変化させられるのは制御のうえでは有利な点である。

*1 最も過酷な状況は、仮想大画面内で同様な画面（たとえば白紙領域）を分担表示している複数台のタブレットが、互いにすれ違う等して管理サーバがタブレットの認識を見失う場合である。このとき、一般的な物体認識手法ではそれぞれのタブレットを識別できない。

本論文では、外部カメラ撮影画像内でのタブレット画面毎固有の表示パターンを識別する手法を、表示パターンが本アプリケーションの利用をなるべく阻害しないようにする方式として提案する。具体的には、撮影画像内でのタブレット画面領域認識、端末 ID 識別、画面向き推定のための方式である。実際のタブレット端末等を用いて提案方式が本アプリケーションの可用性に及ぼす影響が小さいことと、原理的に実行時間で動作可能なことを示す。

2. マルチディスプレイシステム

本論文で想定するマルチディスプレイシステムは、図 1 に示すように、仮想大画面を机上に構成し、その表示デバイスとしてタブレット PC 端末（以下単に端末という）等を並べるといものである。仮想大画面はそのままでは見えず、端末画面に対応する領域を表示することで可視化される。各端末は仮想大画面の一部領域を覗いて表示しているイメージとなり、端末の動的な移動に表示を追従させることが目的である。システムを統括する外部サーバに各端末が接続して制御下に入るアーキテクチャであり、各端末位置等の把握にはサーバに接続された単眼カメラ（Web カメラ、スマートフォンカメラ等）の映像を用いる。

このとき、端末は特定の水平机上に存在することが仮定され、これを撮影画面内座標（スクリーン座標系）から実際の机上面の座標（ワールド座標系）へと変換することで、表示分担領域を特定できる。この変換では奥行き情報が必要であるが、カメラと机の位置関係は固定と見なせるため別途取得できているものと仮定する。そこで、本論文ではワールド座標系に変換された撮影画像内で単眼カメラ映像から端末領域（広がり）認識、端末 ID 識別（複数台の識別）、向き（画面の上下方向）推定をそれぞれ行える、可用性を保ったパターン表示、認識等方式を提案する。なお、実際の画面配信や端末上のタップ操作等 [8], [9], [10], [11] は方式としては容易であり、本論文では議論しない。

HuddleLamp [1] では、奥行き情報を取得できる 3D RGB-

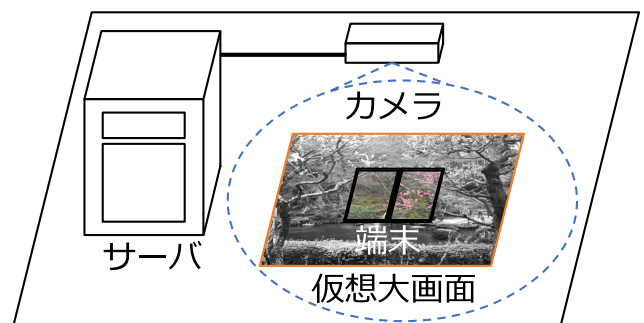


図 1 マルチディスプレイシステム（仮想大画面は端末画面範囲のみ実際には見ることができる）

Fig. 1 Multi-display system (Users can see the image on the virtual screen only within the screen of the tablet devices).

D カメラの画像を手がかりとしたハイブリッドセンシング手法を用いている。端末検出時は端末画面全面に2次元コードパターンを表示し、認識時間は1, 2秒程度である。利用中に端末位置を見失った場合等は認識に1, 2秒程度の短時間、継続利用が停止することとなる。Dipponら [12]も同様に端末検出時にはマーカを表示し設定完了まで利用者は操作ができない。

一方、外部カメラ画像を用いずに端末位置等を推定するLiら [13]の手法は、スマートフォン端末等の前面カメラによる撮影画像中で天井の基準マーカ検出を利用している。これは、天井に準備が必要な方式である。本研究とは逆に、携帯電話カメラで大型ディスプレイ上の位置を推定しようとするBallagasらのPoint & Shoot [14]もあり、点滅するマーカを導入しており、本研究のパターンの一部はこの方式と類似している。

仮想大画面ではなく、複数端末や人の位置関係推定等を行う研究としては、赤外線放射カメラを用いたモーションキャプチャシステムを利用し、赤外線反射マーカを貼付した操作者の位置や向きを推定するBallendatらのインタラクティブメディアプレーヤー [15]や、参加する各端末背面カメラで撮影した利用者の足や床の映像により相対的位置関係を推定して、端末間インタラクションに活用しようとするOrienteer [16]や、大型ディスプレイとユーザの携帯電話とのインタラクションのために、携帯電話の背面カメラで撮影された大型ディスプレイ画面による携帯端末位置推定を行うHerbertら [17]のシステム等があげられるが、本研究目的には適合しない。また、ノートPCの相対的位置を超音波測距によって定めようとするシステム [18], [19]も提案されているが、このような追加機器を要しないほうが望ましい。

なお、端末内蔵センサを利用して相対位置を計算する研究も存在する [20], [21], [22], [23]。GrønbaekらのSlam-to-Share [20]は、ユーザの机を叩くジェスチャをトリガとして、各端末は内蔵センサで振動を検知し、相対位置を計算する。KorzetzらのTape to Unite [22]は、隣接配置した端末の2画面を同時に2本の指で平行にスワイプさせることで、相対的な位置関係を認識する。これらは既存センサのみで位置関係を認識できる反面、ユーザアクションを要する手間が問題となる。

3. 提案手法

3.1 概要

提案手法は、端末画面上に識別用のパターンを表示するが、利用時のコンテンツ画面(端末画面)の端領域のみを利用することで、システムの可用性を位置推定時にも保ったままとする。多くの端末には縁(ベゼル)が存在するため、この画面非表示領域が若干太くなった程度の利用感となることになる。パターン表示は管理サーバから各端末毎への

通知によって行われる。パターンの表示は各端末で行い、時間順次で通知することで端末ID識別が可能であるほか、パターンの違いを利用した同時識別も一部に導入する。

位置推定は、撮影画像内での推定を意味し、画面領域認識と画面向き推定の2段階で行う。画面領域認識は、各端末の表示画面領域を特定することで、動画像(複数フレーム)から推定する。このとき、時間軸方向のパターンを異なるものとするので、それぞれがいずれの端末の画面領域であるかを識別する端末ID識別にも利用する。認識用のパターンは、一般に撮影環境で認識を実行できるだけの空間領域を必要とするが、本方式では時間変化パターンとすることで原理的に空間変化のみの2次元コード等より狭い空間領域(面積)への表示パターンとすることができる。一方、画面向き推定は各端末の画面の上下左右を特定するための処理で、単一フレームから推定する。これは端末の識別後で、向きの情報だけをパターンに符号化すればよいので、最小限の空間パターンで済むため、面積的に問題にならないと考えたためである。画面の上下端に異なるパターンを表示することで向きを推定する。

3.2 本方式の目的

端末は矩形画面を持ち、(一般に動)画像 I を表示している。本手法はこの端末画面の縁領域を用いる。解像度 $W \times H$ (幅×高さ)の短辺を上下、長辺を左右と呼ぶこととし、 $W < H$ とする。画面の縁領域にパターンを表示し、これを外部カメラで撮影する。本方式の目的は、この撮影した画像内での端末画面の位置推定である。

撮影(動)画像中より特定のパターンを検出し、画面領域認識、画面向き推定を行う。撮影画像は一般に幾何変換、その他撮影光学系の収差等を含むが、カメラと机平面との関係は固定的であるため事前に幾何変換できているものとし、本論文では撮影画像内の処理のみについて記述する。収差等は実用上無視できるため考慮しない。簡単のため、机平面座標が撮影画像座標と同一と見なせるものとして説明することとし、以後これを撮影(動)画像と呼ぶ。

3.3 画面領域認識

画面領域認識では、端末画面領域を明確にする4辺の縁領域にパターンを表示することとし、画面端にあたる幅 N 画素の領域をパターン領域と呼ぶ。このとき、 N 画素が撮影画像上での1画素を十分に上回る大きさであることが必要である。一般の物体の輝度は撮影画像内で短時間では変動しないため、時系列で繰り返し輝度変動するパターンを表示することでパターンとして検出させる。繰り返し系列長(フレーム長)を $2m$ [frame], $m \geq 2$ とする。図2に、画面領域認識撮影画面モデルを示す。

一般的な2次元コードは、パターン領域を検出確定するために符号埋め込み領域の周囲に余白を要求するため、符



図 2 画面領域認識撮影画面モデル

Fig. 2 Model of the area estimation for the screen.

号量以上の面積を要する。提案方式では、短時間で輝度変化する物体は通常ないことを利用してパターン領域を検出しているため、このような余白を要しない点が優位である。また、パターン面積が同一の場合に提案方式のほうが撮影解像度が低くても動作できることは設置に気を遣うことが少なく済むと考えている。

表示パターンは輝度 y_1, y_2 ($y_1 < y_2$) の 2 種類を用い、それぞれ連続する m [frame] に順に周期的に表示するようにする。すなわち、輝度 y_1 パターン m [frame]、輝度 y_2 パターン m [frame] の繰り返しである。一般に撮影カメラ動画像におけるフレーム（以下、撮影フレーム）と、表示端末上のフレーム（表示フレーム）とは同期せず、フレームレートも異なる。ここで、撮影と表示のフレームレートをそれぞれ r_c, r_p [frame/s] と表すとき、 $\frac{m}{r_p} \geq \frac{1}{r_c}$ を満たすと、表示フレーム周期 $2m$ [frame] 中で、撮影フレーム中に輝度 y_1, y_2 のいずれかだけの状態を撮影したフレームがそれぞれ 1 [frame] は存在することが保証される。これは実質的に表示フレームレートを撮影フレームレート以下にしたのと同様であるためである。また、輝度 y_1, y_2 のいずれかだけの状態を撮影した撮影フレームを用いて認識処理を行うことで、表示と撮影のフレームは同期していると見なして処理できる。

検出は次のように行う。ここで、撮影動画像において、任意の系列長 $2m$ [frame] について考える。フレームを前半と後半とに分け、両者の直積に基づくタプルを用意する。たとえば、 $m = 3$ のとき、前半 3 フレーム $\{f_1, f_2, f_3\}$ 、後半 3 フレーム $\{f_4, f_5, f_6\}$ とすると、直積集合 $\{(f_1, f_4), (f_1, f_5), \dots, (f_3, f_6)\}$ の 9 個のタプルが得られる。

このすべてのタプル (f_i, f_j) , $i < j$ について、対応する全画素の輝度値差分を求めた差分フレーム d_{ij} を計算する。この差分フレーム中の全画素の画素値総和を s_{ij} とする。すると、異なるパターンを表示したフレーム組の画素値総

和が他の組よりも大きくなることが期待される。そこで、

$$(i, j) = \arg \max_{i, j} s_{ij} \tag{1}$$

と組 (i, j) を求め、 d_{ij} について、領域の輪郭を求める。

具体的には、輝度値を 2 値化*2し、この 2 値化差分フレーム画像でオブジェクト輪郭抽出*3を行う。検出されたもののうち最大サイズの領域が端末画面領域を構成するものと判定する。その後、この輪郭を囲う最小矩形を求め、この矩形を求める端末画面領域 A とする。撮影画像上での矩形による推定縦横画素数を (\tilde{H}, \tilde{W}) とする。

3.4 端末 ID 識別

同時に複数の端末に対して異なる半系列長 m を与えると、一定時間（一定フレーム長）における端末画面領域認識頻度が異なることになる。このことを利用して、系列長を識別することで画面領域認識に引き続いて端末 ID 識別も可能となる。撮影画像上で画面領域認識が可能であることは必要条件であり、検出処理との同期問題や、ノイズ等の影響を軽減するため一定の時間（フレーム数）表示することとなる。そこで、この表示方法を流用しつつ、他のパターン等の表示で可用性を低下させることなく端末 ID 識別も行えるように拡張したものである。

システムに接続されている端末数を n とする。すると、画面認識において認識された領域の大きい順に領域 A_c を列挙できる。これを $\{A_c\}$ とする。この認識された領域は、撮影フレーム上でそれぞれの領域を認識させる表示の明暗の繰り返し周期時間が異なる。この周期の逆数を端末画面領域認識頻度と呼ぶ。各 A_c は異なる認識頻度が得られるため、それぞれ別の端末 ID を割り当てることで端末 ID 識別を実現する。このとき、位置認識に誤差が見込まれるため、別に定める誤差 L ($L \ll W$) 画素以内のものは同一端末であると見なす。

たとえば、接続端末数 $n = 3$ 、それぞれの端末の半系列長 $m = 2, m = 3, m = 4$ のときの、端末画面領域認識頻度は図 3 のようになる。簡単のため、表示と撮影のフレームが同期した 20 [fps] の 1 秒間での例で示す。ここで一定時間 (= 1 [s]) での端末画面領域認識頻度は、 m の値が小さい順にそれぞれ、5, 3, 2 [回/s] になる。

端末 ID 識別は、一定時間内の各端末の画面領域認識フレーム数が異なればよいため、表示と撮影のフレームレートが高いほど、短時間で端末の識別を行うことができる。一方で、システムに接続されている端末が増加するほど、端末 ID 識別時間も増加する。端末数とフレームレートを変化させたときの、端末 ID 識別時間を表 1 に示す。接続端末数が 4 [台]、撮影フレームレートが 20 [fps] の場合でも、1 [s] 未満の短時間での識別が可能である。現在、市場に出

*2 大津の 2 値化手法を採用。

*3 実装は、OpenCV の findContours() メソッドによる。

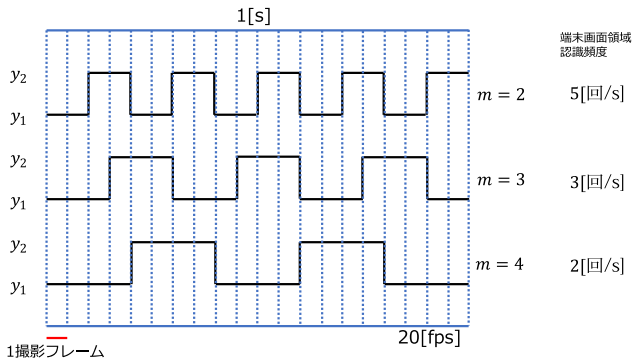


図 3 端末画面領域認識頻度

Fig. 3 Frequency of device screen area recognition.

表 1 端末 ID 識別時間

Table 1 Device ID identification time.

フレームレート [fps]	20	30	60
端末数 n [台]			
2	$\frac{2}{5}$ [s]	$\frac{4}{15}$ [s]	$\frac{2}{15}$ [s]
3	$\frac{3}{5}$ [s]	$\frac{2}{5}$ [s]	$\frac{1}{5}$ [s]
4	$\frac{9}{10}$ [s]	$\frac{3}{5}$ [s]	$\frac{3}{10}$ [s]

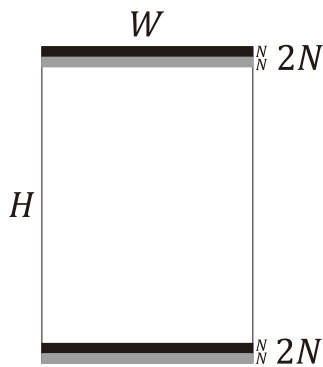


図 4 画面向き推定撮影画面モデル

Fig. 4 Model of displaying image for the orientation estimation.

回っている多くのカメラやスマートフォンは、60 [fps] での撮影が可能でこれは本手法にとって都合が良い一方、10 台を超えるような多数台では相対的に長時間必要であり、この点は今後の課題である。

3.5 画面向き推定

画面領域認識完了後、引き続き画面向き推定を行う。画面向き推定では、端末画面の上下辺（短辺側）それぞれに沿った N 画素幅の線状領域 2 本 2 組にパターンを表示する。端末では上下が設定（存在）していることから、各組の上側領域と下側領域がそれぞれ異なる輝度 y_3, y_4 （説明上 $y_3 < y_4$ とする）の画素となるように表示する。図 4 に、画面向き推定撮影画面モデルを示す。

これを撮影した画像（フレーム）で検出を行う。端末画面領域を構成する長辺方向で、 N 画素相当分の輝度値の画

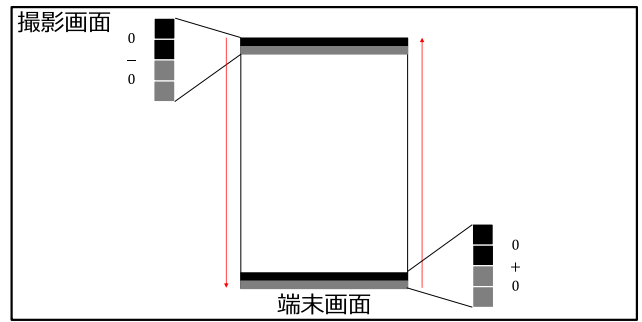


図 5 画素間差分モデル

Fig. 5 Model of differentials between pixels.

素間差分を計算する。 N 画素幅は撮影画像中の端末画面領域の大きさから推定できるため、輝度値の異なる画像を表示している付近の画素を拾い出して推定する。

簡単のため、撮影画像上で端末画面領域は縦横方向に正立しているものとする*4。また、処理対象は端末画面領域内のみとなり、その撮影画像上の解像度は $\tilde{W} \times \tilde{H}$ である。すると、端末上パターン領域 $2N$ 画素幅に相当する撮影画像上の幅 $2\tilde{N}$ は、

$$\tilde{N} = \left\lceil \frac{\tilde{H}}{\tilde{H}} N \right\rceil \quad (2)$$

のように求められる。

次に、画面の長辺方向の一行について考える。この端末の上下端からそれぞれ $2\tilde{N}$ 画素範囲内の画素について、図 5 に示すように、上下方向の一定方向（任意）で隣接画素間での差分を計算する。すると、この上側または下側の差分 1 列分は、項数 $2\tilde{N} - 1$ 個の数列としてそれぞれ、

$$\begin{aligned} s_{px} &= \{e_1, e_2, \dots, e_{2\tilde{N}-1}\}, \\ p &\in \{1, 2\}, \\ x &\in \{1, 2, \dots, \tilde{W}\} \end{aligned} \quad (3)$$

と s_{px} で示せる。ここで、 p は上下端いずれかを示す 2 値、 x は \tilde{W} 幅内での横位置を示す座標である。また、 s_{px} の各要素は、撮影画像上の $\tilde{W} \times \tilde{H}$ 矩形内の座標 (x, y) の画素値を $p(x, y)$ とすると、

$$e_i = p(x, i) - p(x, i - 1) \quad (4)$$

のように表すことができる（下端相当分では y 座標が異なる）。

ノイズを回避しつつパターンによる上下情報を抽出するため、1つの列 (x) に対応する数列を $s_p (= \{e_i\})$, $p \in \{1, 2\}$ とするとき、

$$d_p = \text{sgn}(\arg \max |e_i|) \quad (5)$$

のように、数列の向き情報 d_p を計算する。ここで、関数

*4 事前に回転処理をしておくことと同義である。

sgn は符号判定を行う

$$\text{sgn}(r) = \begin{cases} -1 & \text{if } r < 0 \\ 0 & \text{if } r = 0 \\ +1 & \text{if } r > 0 \end{cases} \quad (6)$$

である.

さらに, この列の上下向き判定 o_k を

$$o_k = \begin{cases} +1 & \text{if } d_1 > 0 \wedge d_2 > 0 \\ -1 & \text{if } d_1 < 0 \wedge d_2 < 0 \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

として得る. この上下向き判定 $\{o_k\}$ は \tilde{W} 個得られるから, この総和

$$O = \sum_{k=1}^{\tilde{W}} o_k \quad (8)$$

を計算する. さらに,

$$\tilde{O} = \text{sgn}(O) \quad (9)$$

として, 当該端末画面領域の撮影画像上での上下判定値 \tilde{O} とする. すると, 処理画像上で端末画面は,

$$\begin{cases} \text{正立位置} & \text{if } \tilde{O} = \text{sgn}(y_3 - y_4) \\ \text{倒立位置} & \text{otherwise} \end{cases} \quad (10)$$

として判定される.

4. 実験

提案手法が利用シーンを想定した状況下で原理的に機能することを機材を用いた実験で確かめる. 実利用での堅牢性を検証するためにはあらゆる環境を想定しなければならず, これは現実的ではない. また, 方式内パラメータごとの適用範囲を明らかにすることも環境要因変数が多く, 実際には, 端末画面で大きく反射が映り込んでいる場合等, そもそも適用できない場合もありうる. そこで本論文では, 低価格帯の機材を用いて, 本方式が原理的に機能することを示し, このときの認識性能や時間が実用領域にあることを示す.

4.1 条件等

実験に用いたタブレット端末の画面解像度は $1,200 \times 1,920$ (機種: ASUS MeMO Pad 7), 撮影には, 撮影動画画像解像度 800×600 画素の Web カメラ (機種: Logicool HD Pro Webcam C910), 露出は auto とした. 端末を水平の机上 (木製) に置き, そのほぼ真上約 45 [cm] から, カメラスタンドを用いて固定したカメラで撮影した. このため, 事前の幾何変換処理は行っていない. この撮影距離 45 [cm] は,

表 2 実験環境

Table 2 Experiment environment.

撮影条件	
撮影照度	334 [Lux]
撮影角度	垂直
撮影距離	45 [cm]
サーバ	
プロセッサ	Intel Core i5-6500 3.2 GHz
メモリ	8 [GB]
OS	Ubuntu 20.04.1 LTS
カメラ	
製品名	Logicool HD Pro Webcam C910
解像度	800×600
フレームレート	20 [fps]
露出	Auto
端末	
製品名	ASUS MeMO Pad 7 (ME572C)
解像度	$1,200 \times 1,920$

接続端末数 $4 \sim 6$ [台] が撮影画面内に収まる距離であり, 本方式の原理的動作を検証するために選択した. 使い勝手を考えると机面~カメラは $1 \sim 2$ [m] 程度の距離が望ましいと考えるが, その場合はカメラ画角や解像度によって利用可否が影響を受ける可能性がある. 多数台を撮影画面に収めて本方式を動作させるためにはそれに応じた撮影画像解像度が必要であると考えられるが, 具体的な検証は今後の課題である. なお, 使用カメラの最大フレームレートは 30 [fps] であるが, 使用機材は最大フレームレートでは撮影解像度が 640×480 に制限されてしまう. 本実験では撮影解像度を 800×600 とするため, 対応するフレームレート 20 [fps] にて撮影を行った. 撮影環境は一般家庭の居室に相当する環境といえる LED 照明下で机上照度は 334 [Lux] である. 条件の一覧を表 2 に示す.

表示パターン輝度は $y_1 = y_3 = 0, y_2 = y_4 = 128$ (輝度の値域は $[0, 255]$) とした. 本方式はこの輝度値の差異 (時間変化) をとらえる手法であるため, 周辺物体の色合い等は撮影画像の露出の影響は受けるもののあまり影響しない. 撮影画像上での輝度は, 黒いベゼルは 0 に近く, 机は 110 前後であった. パターンは黒色側 (y_1, y_3) で 50 前後, 灰色側 (y_2, y_4) が 120 前後であった.

4.2 画面領域認識

表示パターン幅を経験的に $N = 5$ 画素, 表示パターンの半系列長 $m = 3$ とした. 端末の撮影画面内座標における角度を変化させ異なる条件について実験を行った. パターン画像を図 6, 図 7 に示す. 検出は連続する $2m = 6$ [frame] で行われるため, その時間は $\frac{3}{10}$ 秒である. 撮影画面例は, 図 8, 図 9 であり, 撮影画面に占める端末画面面積比は, 約 0.11 である. 端末の画面内角度は表 3 に示すように一



図 6 パターン画像 ($y_1 = 0$)
Fig. 6 Pattern image ($y_1 = 0$).



図 7 パターン画像 ($y_2 = 128$)
Fig. 7 Pattern image ($y_2 = 128$).



図 8 端末角度 (45°)
Fig. 8 Device angle (45°).



図 9 端末角度 (90°)
Fig. 9 Device angle (90°).

部に不規則な角度も交えた多様な角度で実験した。

推定結果例として、推定矩形を緑枠で描画したものを図 10 に示す。また、推定結果の正解率は表 3 に示すと

表 3 画面領域認識の正解率

Table 3 Accuracy of estimated device screen area.

端末角度	正解率
0°	1.00
22°	1.00
45°	1.00
90°	1.00
113°	1.00
135°	0.80
180°	1.00
225°	1.00
265°	0.90
270°	0.90
299°	1.00
315°	0.95

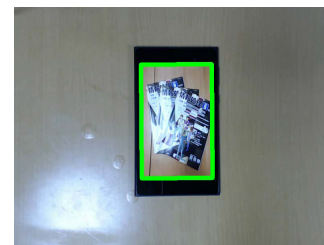


図 10 画面領域認識結果例
Fig. 10 Example of screen area recognition.

りで、角度は撮影画面水平方向に対して反時計回りでの端末短辺方向の正解方向を示している。正解か否かは撮影画面内で正解位置を目視で定め、これと最大誤差 2 画素以内であるか否かで判定しているが、不正解の場合はまったく異なる矩形位置を推定している場合がほとんどで、正誤の判定は容易であった。

結果からは、一部の角度で少し低い正解率となっており、この要因は不明であるが、画面に映り込んだ照明や撮影者の動き等が影響している可能性があげられる。おおむね高い正解率であり、数分の 1 秒単位で検出できるため、この複数回の結果による投票等で 1, 2 秒程度の実用的な推定を行えると推定できる。表示パターン幅は経験的に設定したが、実験条件の場合、 $N = 3$ 程度まで認識可能であった。実験の $N = 5$ のとき、端末画面に占める表示パターンの面積比は、約 1.3% である。

4.3 同時画面領域推定による端末 ID 推定

半系列長にあたる $m = 2$ (端末 1)、 $m = 3$ (端末 2) の 2 通りを 2 台の端末に同時に与えることで、端末画面領域と端末 ID の同時推定を行う。図 11 は撮影画像に左側端末の領域推定矩形を緑枠で描画した例である。全 50 [frame] の撮影動画上で推定したところ、端末 1, 2 それぞれで、26, 16 [frame] ずつ領域認識に成功した。各領域について投票することで、各端末の位置が確定し、また、推定頻



図 11 2台識別実験

Fig. 11 Experiment to discriminate two devices.



図 12 パターン画像 ($y_3 = 0, y_4 = 128$)

Fig. 12 Pattern image ($y_3 = 0, y_4 = 128$).

度（系列長に依存）によって端末の判定も行えることが分かった。

4.4 画面向き推定

経験的にパターン表示半幅 $N = 5$ 画素として画面向き推定手法について実験した。パターン画像を図 12 に示す。表示パターンの端末画面占有率は約 1% である。端末の撮影画像内角度がいずれの場合でも正しく向きを推定できた。端末角度ごとの撮影画像を図 13 に示す。推定アルゴリズムで差分を計算した、計算処理上の仮の上下方向を赤い矢印で示している。その結果、端末の上側と推定した端末上の辺部分を赤い丸で示している。計算時の仮の順序とは無関係に適切に端末上側を判定できていることが分かる。

端末角度 $0^\circ, 180^\circ, 225^\circ$ の撮影画像は、端末画面およびベゼルで照明の映り込みが見られ、画面向き推定での誤推定が心配されたが、向き検出自体は問題なく行われた。ただし、式 (8) での値 O の絶対値が他の角度より小さめであった。これは、画面向き推定のための隣接画素輝度値の差分を適切に求められていない画素列が多めであったことを意味する。しかし、それは信頼度の低い列を除外していてもいえ、判定自体に問題あるほどの状態ではなかった。

なお、この O の絶対値が最小であったのは、端末角度 90° の場合である。この場合も判定自体に問題を及ぼすような値の状況にはなかったが、詳細の状況の分析については今後とも行っていきたい。

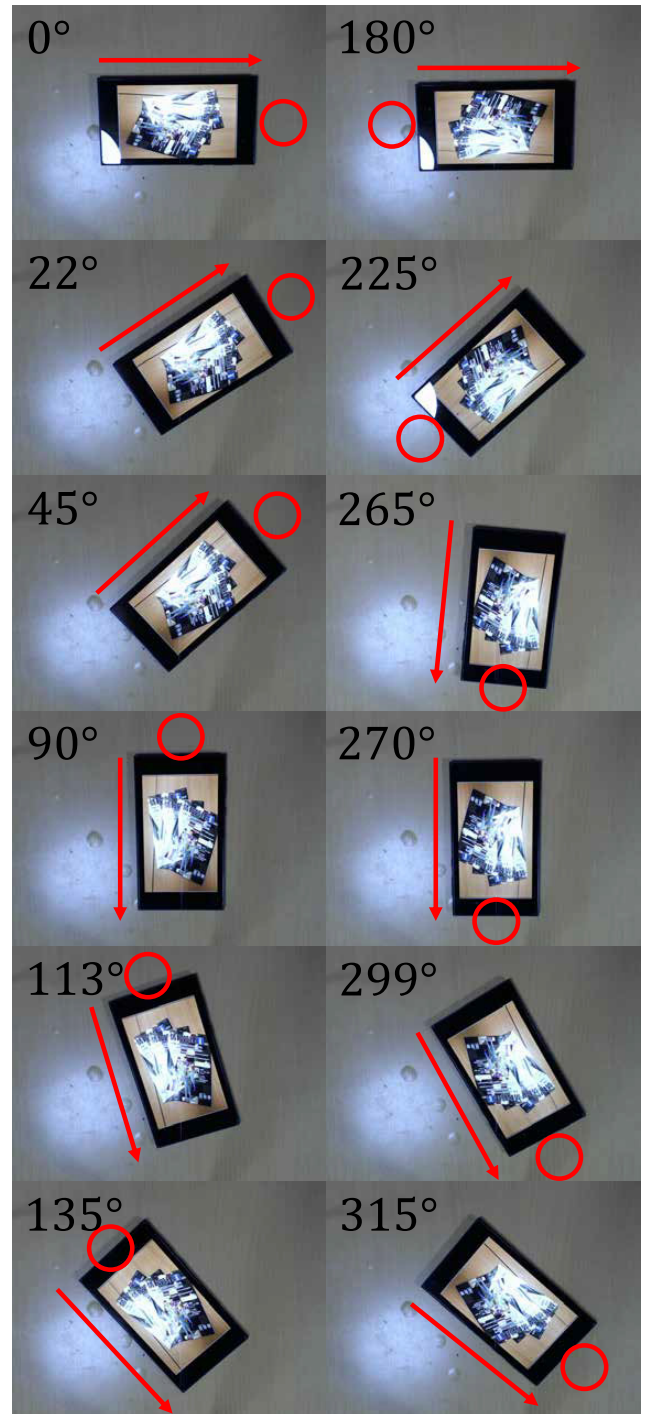


図 13 画面向き推定実験画像

Fig. 13 Images of screen orientation estimation.

4.5 2次元コード手法との比較

2次元コードの例として ArUco マーカを利用した画面領域認識と比較する。提案手法同様に、撮影画面内の端末の角度を変化させた 12 種類の撮影画像を用いて、端末画面領域認識を行った。環境は、表 2 と同一である。ただし、ArUco マーカは印刷利用を想定しているため、マーカの作成は自作プログラムによるものである。推定したマーカ位置を緑枠で描画したものを図 14 に示す。結果は、端末角度がいずれの場合でもマーカを検出できた。しかし、マー

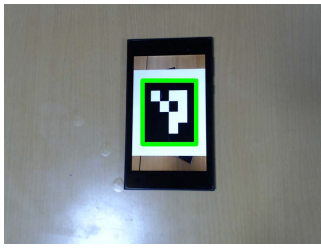


図 14 端末位置推定画像 (ArUco マーカー)

Fig. 14 Estimated image of device position (ArUco marker).

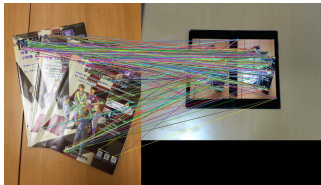


図 15 2台識別実験 (ORB)

Fig. 15 Experiment to discriminate two devices (ORB).

カが表示コンテンツを妨害していることは図 14 から明らかであり、実験例での端末画面に占めるマーカー領域面積比は、約 0.75 であった。また、端末画面領域 (大きさ) の情報は取得することができない。もし、これをマーカー表示で実現しようとする、全画面を占有しなければならない。

これに対して、提案方式では、 $N = 5$ の条件で十分に画面領域に認識ができた。このとき、パターンの画面占有率は、約 1.3% に過ぎず可用性を維持しているといえる。また、2次元コードではパターン領域を検出、確定するための余白領域を要求するため、端末識別等のための符号量以外の面積を要してしまう。短時間で輝度変化する物体が通常は存在しないことを用いてパターンを検出する本方式ではこのような領域を時空間的に要しない点は原理的な優位性である。

4.6 画像認識による追跡手法との比較

通常の端末移動認識には画像認識手法が想定される。しかし、見失うことや、各端末での表示画像が識別できない内容であるため失敗する場合もある。ここでは、2台の端末上に同一内容が表示されている場合に端末 ID 識別が確実にできないことを確認する実験を行った。認識のための画像特徴、マッチングには、それぞれ ORB, BFMatcher を使用した。同一画像を 2 台の端末それぞれに表示した状態で、物体認識を行うための特徴点どうしの対応関係を、表示の元画像とのマッチングとして見ている。

結果は、図 15 で、マッチした点どうしを線で結んでいる。当然のことながら両端末上の画像に同一箇所のマッチが多数見られる。これは、端末画面に、白紙等を含む繰り返しパターンや、よく似た写真を多数表示させている場合に想定される状況である。このほか、撮影画像の解像度が表示コンテンツの特徴に対して不足している状況 (書類や

地図等の細かな描画のコンテンツ表示) でも起こりうると考えられるが、これらの具体的シーンについては未調査である。

提案手法は、これと同等な同一のコンテンツ内容を表示した状態で実験を行ったが、識別用パターンとして端末ごとに異なるパターンを表示させることで端末 ID 識別を実現していた。このことから、提案手法は、表示コンテンツに依存なく確実な端末 ID 識別を実現できる手法であるといえる。

4.7 可用性評価

画面領域認識、向き推定いずれも、端末画面に占める表示パターン面積比は、約 0.01 である。これは、既存研究の多くで用いられている 2次元コードの、端末画面に占める面積比に比べて、明らかに小さく、また、大部分のコンテンツ内容を表示し続けることが可能であるといえるため、端末位置推定時にも可用性を維持している。この面積比はパターン表示幅依存であり、一般に表示幅が広いほど認識成功率は高まると予想される。実験で用いた値は経験的に実験機材、環境で十分な数値として設定したものであり、現実には認識の難易度に合わせて増減されるべきものである。しかし、実験で用いた機材はなんら特殊なものではなく、経年も経ており現在では低性能なものに分類され、このうでこの程度の設定で認識できたため、仮に表示パターン面積が実験の数倍であっても、全体の 9 割以上の画面面積でコンテンツを表示し続けることが可能であるため、全体として可用性を維持したパターン表示、認識手法であるといえる。

認識時間は表示パターンの半系列長と撮影フレームレートに依存するが、実験時パラメータでは、画面領域認識については、1 回の検出に $\frac{3}{10}$ [s]、この検出結果の投票を行うので、1, 2 秒程度で端末画面領域を特定することが可能である。画面向き推定に関しては、撮影動画像の 1 [frame] で検出が可能であるため、 $\frac{1}{20}$ [s] とプログラムの処理時間約 0.12 [s] を足した、0.17 [s] という短時間で、端末向きを特定することができる。これらは、十分実用的な認識時間であり、システムの利用を妨げない点において、従来手法より優れているため、提案手法は、システム利用者の利便性を向上できる手法である。

5. おわりに

複数台のタブレット端末を机上で並べて仮想大型画面を構成するマルチタブレットシステム研究について、サーバクライアント方式の一般的な構成について紹介し、その中で必要となる端末位置、向きの推定手法について提案した。これまで、この応用分野では普及した 2次元コード様の端末画面を占有するパターンを表示させ、これを外部カメラで撮影して認識するような手法が用いられてきた。

しかし、提案手法では、端末画面上のほとんどの領域をコンテンツ表示を保ったままこれらの端末状況認識を行うことができ、システム利用時の可用性を維持したままとすることのできる手法であることを明らかにした。

本論文では、端末画面端のみへのパターン表示での認識ができることを原理的に示したものである。したがって実用にあたっては、外部カメラによる端末の撮影倍率によって、パターン表示幅を変動させる等の適応制御が望ましい。これを含み、パターン自体の縮小や半透明とする等、さらにコンテンツ表示を阻害しないパターン表示方式は今後の課題である。

参考文献

- [1] Rädle, R., Jetter, H.C., Marquardt, N., Reiterer, H. and Rogers, Y.: HuddleLamp: Spatially-Aware Mobile Displays for Ad-hoc Around-the-Table Collaboration, *Proc. ITS '14*, pp.45-54, ACM (2014).
- [2] Zhao, Y., Campisi, P. and Kundur, D.: Dual Domain Watermarking for Authentication and Compression of Cultural Heritage Images, *IEEE Trans. IP*, Vol.13, No.3, pp.430-448 (2004).
- [3] Liu, R. and Tan, T.: An SVD-Based Watermarking Scheme for Protecting Rightful Ownership, *IEEE Trans. Multimedia*, Vol.4, No.1, pp.121-128 (2002).
- [4] Dong, P., Brankov, J.G., Galatsanos, N.P., Yang, Y. and Davoine, F.: Digital Watermarking Robust to Geometric Distortions, *IEEE Trans. IP*, Vol.14, No.12, pp.2140-2150 (2005).
- [5] Chen, S. and Leung, H.: Chaotic Watermarking for Video Authentication in Surveillance Applications, *IEEE Trans. CSVT*, Vol.18, No.5, pp.704-709 (2008).
- [6] Liu, W., Dong, L. and Zeng, W.: Optimum Detection for Spread-Spectrum Watermarking That Employs Self-Masking, *IEEE Trans. IFS*, Vol.2, No.4, pp.645-654 (2007).
- [7] Suzuki, H., Kawai, G., Fujita, T. and Kobayashi, A.: Device position estimation for Multi-tablet virtual display system, *Proc. CANDAR '20*, CANDAR (2020).
- [8] Li, M., Scharf, K.M. and Kobbelt, L.: MobileVideoTiles: Video display on multiple mobile devices, *Proc. MobileHCI '16*, pp.621-626, ACM (2016).
- [9] Kim, S., Ko, D. and Lee, W.: Utilizing Smartphones as a Multi-Device Single Display Groupware to Design Collaborative Games, *Proc. DIS '17*, pp.1341-1352, ACM (2017).
- [10] Marquardt, N., Brudy, F., Liu, C., Bengler, B. and Holz, C.: SurfaceConstellations: A Modular Hardware Platform for Ad-Hoc Reconfigurable Cross-Device Workspaces, *Proc. CHI '18*, pp.1-14, ACM (2018).
- [11] Zagermann, J., Pfeil, U., Radle, R., Jetter, H.C., Klokmose, C. and Reiterer, H.: When Tablets meet Tabletops: The Effect of Tabletop Size on Around-the-Table Collaboration with Personal Tablets, *Proc. CHI '16*, pp.5470-5481, ACM (2016).
- [12] Dippon, A., Wiedermann, N. and Klinker, G.: Seamless Integration of Mobile Devices into Interactive Surface Environments, *Proc. ITS '12*, pp.331-334, ACM (2012).
- [13] Li, M. and Kobbelt, L.: Dynamic Tiling Display: Building an Interactive Display Surface Using Multiple Mobile Devices, *Proc. MUM '12*, pp.1-4, ACM (2012).
- [14] Ballagas, R., Rohs, M. and Sheridan, J.G.: Sweep and Point & Shoot: Phocemcam-Based Interactions for Large Public Displays, *Proc. CHI EA '05*, pp.1200-1203, ACM (2005).
- [15] Ballendat, T., Marquardt, N. and Greenberg, S.: Proxemic Interaction: Designing for a proximity and Orientation-Aware Environment, *Proc. ITS '10*, pp.121-130, ACM (2010).
- [16] Dearman, D., Guy, R.T. and Truong, K.N.: Determining the Orientation of Proximate Mobile Devices using their Back Facing Camera, *Proc. CHI '12*, pp.2231-2234, ACM (2012).
- [17] Herbert, L., Pears, N., Jackson, D. and Olivier, P.: Mobile Device and Intelligent Display Interaction via Scale-invariant Image Feature Matching, *Proc. PECCS '11*, PECCS (2011).
- [18] Hazas, M., Kray, C., Gellersen, H., Agbota, H., Kortuem, G. and Krohn, A.: A Relative Positioning System for Co-located Mobile Devices, *Proc. MobiSys '05*, pp.177-190, ACM (2005).
- [19] Randell, C. and Muller, H.: Low Cost Indoor Positioning System, *Proc. UbiComp '01*, pp.42-48, ACM (2001).
- [20] Grønbaek, J.E., Knudsen, M.S., O'Hara, K., Krogh, P.G., Vermeulen, J. and Petersen, M.G.: Proxemics Beyond Proximity: Designing for Flexible Social Interaction Through Cross-Device Interaction, *Proc. CHI '20*, pp.1-14, ACM (2020).
- [21] Korzetz, M., Kühn, R. and Schlegel, T.: Turn it, Pour it, Twist it: A Model for Designing Mobile Device-Based Interactions, *Proc. CHIUXID '19*, pp.20-23, ACM (2019).
- [22] Korzetz, M., Kühn, R., Heisig, P. and Schlegel, T.: Natural collocated interactions for merging results with mobile devices, *Proc. MobileHCI '16*, pp.746-752, ACM (2016).
- [23] Ohta, T. and Tanaka, J.: MovieTile: Interactively adjustable free shape multi-display of mobile devices, *Proc. SA '15*, pp.1-7, ACM (2015).



鈴木 広人 (正会員)

2019年工学院大学工学部情報通信工学科卒業。2021年同大学大学院工学研究科電気・電子工学専攻修士課程修了。



小林 亜樹 (正会員)

2000年東京工業大学大学院理工学研究科電気・電子工学専攻博士後期課程修了。博士(工学)。同大学院助手、(独)メディア教育開発センター助教等を経て現在、工学院大学情報学部情報通信工学科准教授。主に、情報検

索・推薦、ネットワーク、画像認識、インタラクティブシステムの研究に従事。電子情報通信学会、映像情報メディア学会、日本データベース学会、ACM等各会員。