

床指紋照合における傾斜補正手法

藤田 悟¹ 内田 薫²

概要: 本研究は, 床面の画像から特徴点抽出を行い, 床指紋として画像照合することで, 屋内でも利用可能な位置推定システムを実現することを目的としている. 高速な画像照合のためには, 床面を2次元平面として扱える必要があり, 3次元歪が無視できる程度の画像が撮影できることが求められる. しかし, 実際にスマートフォンを自然な形で保持して撮影した画像は, 端末の傾斜によって大きな歪を含んでいる. そこで, 本論文では, ARCore を利用して床面の形状とカメラの姿勢推定を行い, 3次元歪を補正する手法について述べる. この手法では, 3次元空間のモデルを計算することでホモグラフィ行列を生成し, 前後や左右に傾けた端末で撮影した床画像を, 直上から撮影した画像に変換することができる. 画像補正の過程で, 画質の劣化が生じるが, 実験を通して, 床指紋照合に十分な情報を含んでいることを確認した. さらに, 床画像データベースと照合することにより, 画像の補正無しで照合した場合と比較して, より高い確率で位置推定に成功することを確認した.

Rectification of Distorted Images in Floor Fingerprint

SATORU FUJITA¹ KAORU UCHIDA²

1. 序論

床表面には, 模様や汚れ, 傷などが多数存在し, 一見, 同一に見える床であっても, それぞれに微妙に異なるパターンを保有している. このように床表面画像に現れる固有の特徴を抽出し, 唯一に同定することができれば, 床画像から, 特定の位置を推定することが可能になる. 著者らは, この技術を, 床指紋照合と名付け, 特徴点の効果的抽出手法や, 床画像データベースとの高速照合技術について研究を進めてきた [2][3]. 床画像は, 一般に強い特徴点が少なく, 照合が困難である. これに対して, 過去の研究成果の中で, ORB[7] を改良した特徴点の抽出手法や, RANSAC[1] を用いた高速なマッチング手法に関して提案してきた. 実際に 87.5m の廊下を用いて, 床指紋照合による位置推定実験を実施し, 正解率 99.5%, 1 画像当たりの照合時間 0.5 秒の結果を得て, 本位置推定手法の実用性を示してきた.

これらの実験では, 実用的な利用シーンを想定し, 床画

像をスマートフォンで撮影した. しかし, カメラレンズが床面に平行になることを意識して撮影しており, スマートフォンを傾けて保持して撮影した場合には, 画像に3次元の透視投影の効果による歪が生じ, 認識精度が低下した. これは, 画像照合の手法が, 歪の無い2次元的な床平面を仮定していて, 画像間で同じ性質を持つ特徴点の相似三角形を探索するアルゴリズムとなっていることが原因である. そこで, スマートフォンに備わっている加速度センサーを用いて重力方向ベクトルを求めることにより, 画像の傾斜歪を補正する方法について, 検討を進めてきた [4]. しかし, カメラの床面からの高さが未知のパラメータであり, 一般的なスマートフォンの保有姿勢を想定して, 高さを 1.3m に仮定した上での補正手法であった.

一方, Google 社が提供を開始した ARCore プラットフォームでは, 撮影映像中から水平な平面を同定し, カメラ姿勢やカメラ位置を推定する機能を持つ. このプラットフォームを利用すれば, カメラの高さを計測することができ, また, 画像の歪補正についても, より容易に正確に実現できることが期待できる. そこで, 本稿では, 加速度センサーに加えて, ARCore を用いた歪補正手法を提案し, 実証実験により性能評価を行う.

¹ 法政大学情報科学部
Computer and Information Sciences, Hosei University

² 法政大学大学院情報科学研究科
Graduate School of Computer and Information Sciences, Hosei University

以下、2章で、既存の歪補正手法について概要を説明し、3章で ARCore を利用した歪補正手法を提案する。続いて、4章で性能評価を行い、5章で課題などの考察を行う。最後に、6章で結論をまとめる。

2. 歪補正手法

Kurz らは、重力ベクトルを用いて、水平な平面画像の3次元歪を補正する手法 GREED (gravity-rectified feature descriptors) を提案した [5][6]。この手法では、端末の座標系で表現された重力ベクトル $\mathbf{g} = (g_x, g_y, g_z)^T$ を、地表面のグローバル座標系 $\mathbf{z} = (0, 0, 1)^T$ に対応させるように、ホモグラフィ行列 H を求めている。まず、重力ベクトル $\mathbf{g}$ を法線とする平面を $\mathbf{g}_1$ と $\mathbf{g}_2$ という直交ベクトルで表現する。

$$\mathbf{g}_1 = (-g_z, 0, g_x)^T \quad (1)$$

$$\mathbf{g}_2 = \mathbf{g} \times \mathbf{g}_1 \quad (2)$$

この時、ホモグラフィ行列 H_{kurz} は、次のように与えられるとした。

$$H_{kurz} = [\mathbf{g}_1 \ \mathbf{g}_2 \ \mathbf{z}]^{-1} \quad (3)$$

この手法では、 H_{kurz} によるホモグラフィ変換結果の表示位置や表示倍率が思った結果にならない場合があり、Kurz らの論文でも、経験的に、式 (4) のような微修正を行って利用している [5][6]。

$$H_{scaled} = [\mathbf{g}_1 \ \mathbf{g}_2 \ \sqrt{|\mathbf{g}|\mathbf{z}}]^{-1} \quad (4)$$

一方、著者らは、表示位置や表示倍率について、より正確に定義したホモグラフィ行列の計算方法を示した [4]。まず、端末の local 座標系から見た重力ベクトル $\mathbf{g} = (g_x, g_y, g_z)^T$ の向きを z 軸とし、地面に水平な x 軸と y 軸を持つような global な座標系の基本ベクトル $\mathbf{b}_x, \mathbf{b}_y, \mathbf{b}_z$ を、端末の local 座標系を用いて、次のように定義する。

$$\mathbf{b}_z = (g_x/G, g_y/G, g_z/G)^T \quad (5)$$

$$\mathbf{b}_x = (-g_z/G', 0, g_x/G')^T \quad (6)$$

$$\mathbf{b}_y = \mathbf{b}_z \times \mathbf{b}_x \quad (7)$$

ただし、 $G = \sqrt{g_x^2 + g_y^2 + g_z^2}$ 、 $G' = \sqrt{g_x^2 + g_z^2}$ である。この時、ホモグラフィ行列 H_0 は、上記の3個の基本ベクトルから構成できる。

$$H_0 = [\mathbf{b}_x \ \mathbf{b}_y \ \mathbf{b}_z] = \begin{pmatrix} \frac{g_z}{G'} & -\frac{g_x g_y}{G'G} & \frac{g_x}{G} \\ 0 & \frac{G'}{G} & \frac{g_y}{G} \\ \frac{g_x}{G'} & -\frac{g_y g_z}{G'G} & \frac{g_z}{G} \end{pmatrix} \quad (8)$$

しかし、このままでは、ホモグラフィ変換の中心は画像の原点 (左上) であり、また、画像の表示倍率も制御できていない。そこで、原点を画像の中心に移動し、表示倍率を元画像の倍率に制御するために、 H_0 の左右から、下記の行列を積算する。ただし、 l_x は、1m の距離から撮影した時の画像の x 軸の実世界での長さ、 r_x, r_y は、画像の x 軸、 y 軸の画素数である。 h は、撮影時の画像の地表面からの高さである。

$$H_g = L H_0 R \quad (9)$$

$$R = \begin{pmatrix} \frac{l_x}{r_x} & 0 & -\frac{l_x}{2} \\ 0 & \frac{l_x}{r_x} & -\frac{l_x r_y}{2r_x} \\ 0 & 0 & 1 \end{pmatrix} \quad (10)$$

$$L = \begin{pmatrix} \frac{r_x}{l_x} & 0 & \frac{r_x}{2} - \frac{h r_x g_x}{l_x g_z} \\ 0 & \frac{r_x}{l_x} & \frac{r_y}{2} - \frac{h r_x g_y}{l_x g_z} \\ 0 & 0 & 1 \end{pmatrix} \quad (11)$$

以上により計算したホモグラフィ行列 H_g を用いることで、前後に傾斜した端末から撮影された画像が、カメラを水平な位置に保って撮影した画像に変換されることを確認した。ただし、 h は地表からカメラまでの撮影時の高さであり、 $h = 1.3$ などの値を暫定的に利用してきた。また、端末の前後方向の傾きは補正できるものの、左右方向の傾きについては、考慮していなかった。

3. ARCore を用いた歪補正手法

ARCore は、Google 社の開発した Augmented Reality を実現するための Android 向けのプラットフォームである。画像内の特徴点をトラッキングすることにより、水平な平面の抽出や、カメラの位置と傾きを推定する機能を持つ。床指紋の研究の観点では、床表面の抽出が可能であり、カメラの傾きや高さを測定できることは有効であるが、次の2点を解決する必要がある。

- ARCore がカメラデバイスを利用するため、Android 上で動作する通常の撮影 API を利用できない。
- ARCore のカメラ Pose などの情報から、ホモグラフィ行列を求める方法が明らかでない。

両者について、以下の節で解決手法を示す。

3.1 画像の取得

ARCore の起動中は、ARCore がカメラデバイスを占有しているため、カメラ API を利用しようとしても、デバイスが busy となり、利用できない。ARCore を一時停止するなど、いくつかの方法を試したが、著者らの知る限りで

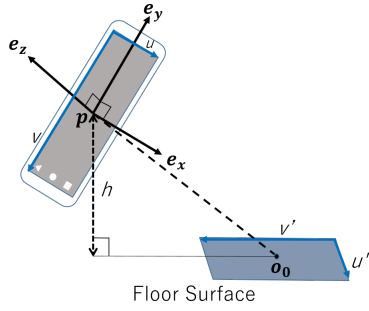


図 1 端末と床面の矩形領域の位置関係

は、カメラを直接利用することはできなかった。そこで、ARCore が利用しているカメラ画像情報から、必要な画像を抜き出して利用する手法を検討した。

ARCore では、カメラ画像は OpenGL の Texture 情報として取り込まれる。ただし、このカメラ画像からの Texture は内部拡張機能として提供されていて、直接、画像情報に参照できない。そこで、Texture を、いったん、Frame Buffer の矩形領域に書き出し、それを読み込む手法を用いた。実際には、カメラ画像の Texture はスマートフォン画面に表示する必要もあることから、次の手順で実装した。

- (1) Renderer を用いて、Texture をスマートフォン画面に表示する。
- (2) 画面から、OpenGL の glReadPixels 関数を呼び出して、pixel 情報として画像を抽出する。
- (3) Android の createBitmap 関数を用いて、bitmap 情報に変換し、ファイルに出力する。

本手法の問題点は、画像の解像度である。ARCore は Preview モードの動画を用いて画像解析や画像表示を行っているため、低解像度の画像しか利用できない。例えば、Pixel 2 では、写真モードであれば 3024×4032 の画素数の画像が撮影可能であるが、Preview モードからの画像の画素数は、 1080×1920 である。また、この数値以上に、画素が荒く、床指紋照合に支障をきたす恐れがある。

3.2 ホモグラフィ行列の計算

ARCore は、撮影画像の特徴点をトラッキングすることで、水平な床面を認識し、また、カメラ (端末) の位置と姿勢を取得できる。これらを用いると、床画像を直上から撮影した歪のない 2 次元画像に変換できる。以下、図 1 を参考にして、詳細を説明する。

ARCore の座標系は、初期値として、ある基準点を中心に、床面の法線を y 軸にし、右側に x 軸、画面の手前側に向けて z 軸が設定される。これが global 座標系であり、以後、端末を移動させても固定化される。認識された平面の位置、端末の位置 (p) と姿勢 (Pose) もこの global 座標系で表現される。端末の姿勢は、カメラのレンズから見た local 座標系の基本ベクトルで表現され、右に x 軸、上に y

軸、画面の前面方向に z 軸を配した直交座標系の基本ベクトルを global 座標系のベクトルとして表現したものになる。これらを、 e_x, e_y, e_z とする。姿勢行列 E は、式 (12) で表現される。

$$E = [e_x \ e_y \ e_z] \quad (12)$$

この基本ベクトルに沿って端末の local 座標系の座標を $s = (s_x, s_y, s_z)^T$ とし、端末の位置座標を p とすると、カメラから見える物体の座標 o は、式 (13) で表される。

$$o = p + Es \quad (13)$$

一方、端末のスクリーン座標 (u, v) を、 u 軸が右向き、 v 軸が下向きに定義し、端末のスクリーンの画素数を、縦 r_y 、横 r_x であるとし、端末から、1m 離れた面を撮影した時の縦方向の実世界の長さを l とすると、 s_x, s_y, s_z と u, v の関係は、次のように表現できる。

$$\begin{pmatrix} s_x \\ s_y \\ s_z \end{pmatrix} = L_1 \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} s_z \quad (14)$$

$$L_1 = \begin{pmatrix} -\frac{l}{r_y} & 0 & \frac{r_x l}{2r_y} \\ 0 & \frac{l}{r_y} & -\frac{l}{2} \\ 0 & 0 & 1 \end{pmatrix} \quad (15)$$

これに基づいて、点 o は、式 (16) に展開できる。

$$o = p + EL_1 \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} s_z \quad (16)$$

次に、カメラ端末からの視線が地表面と交差する点の座標を o_0 とする。この o_0 をカメラの local 座標系で表すと、 $(0, 0, s_{0z})^T$ になる。端末と地表面の高度差を h とし、 $e_z = (e_{zx}, e_{zy}, e_{zz})^T$ とすると、 s_{0z} は、式 (17) で計算することができる。

$$s_{0z} = -h/e_{zy} \quad (17)$$

以上により、交点 o_0 は、式 (18) で表される。

$$o_0 = p + e_z s_{0z} \quad (18)$$

次に、ホモグラフィ変換を行いたい床面の領域を視線と床面との交点 o_0 を中心に矩形領域として設定する。矩形領域の座標 (u', v') は、それぞれ、 u' 軸が右向きとし、 $(e_{zz}, 0, -e_{zx})^T$ 方向にとり、 v' 軸が下向きとし、 $(e_{zx}, 0, e_{zz})^T$ 方向にとる。単位ベクトル化するため、 $d = \sqrt{e_{zz}^2 + e_{zx}^2}$ とすると、床面上の点 o は式 (19) で表現される。

$$\mathbf{o} = \mathbf{p} + TL_2 \begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix} \quad (19)$$

$$L_2 = \begin{pmatrix} \frac{l}{r_y} & 0 & \frac{r_x l}{2r_y} \\ 0 & \frac{l}{r_y} & -\frac{l}{2} \\ 0 & 0 & 1 \end{pmatrix} \quad (20)$$

$$T = \begin{pmatrix} \frac{e_{zz}}{d} & \frac{e_{zx}}{d} & -\frac{e_{zx}}{e_{zy}} h \\ 0 & 0 & -h \\ -\frac{e_{zx}}{d} & \frac{e_{zz}}{d} & -\frac{e_{zz}}{e_{zy}} h \end{pmatrix} \quad (21)$$

床面上の点において、式 (16) と式 (19) は等しくなることから、式 (22) を導くことができる。

$$EL_1 \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} s_z = TL_2 \begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix} \quad (22)$$

よって、床面の矩形領域の座標系と、それが対応するスクリーン座標の関係は、式 (23) で表現できることがわかる。

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} s_z = L_1^{-1} E^{-1} TL_2 \begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix} \quad (23)$$

ホモグラフィ行列 H_{ar} は、式 (24) を満たす行列として定義される。

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} s_z = H_{ar}^{-1} \begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix} \quad (24)$$

よって、最終的に、ARCore を用いた歪補正のためのホモグラフィ行列 H_{ar} は、式 (25) にて計算できる。

$$H_{ar} = (L_1^{-1} E^{-1} TL_2)^{-1} \quad (25)$$

3.3 画像変換例

撮影用アプリは、ARCore のサンプルコードを拡張して開発した。図 2 に、ARCore を用いた撮影シーンと画面のスナップショットを示す。利用者は、普通に端末を傾けたままで床面を撮影する。画面の下方に、緑色の網目がかかっている部分が、平面を認識したことを表している。この時、画面をタッチすることで、画像の取得と、ARCore の必要パラメータの取得を行い、別個のファイルに保存することができる。

図 3 に、重力ベクトルを用いた手法と、ARCore を用いた手法による画像変換例を示す。撮影対象は、画像の歪みがわかりやすいように、小さな 1 辺 10 の正方形のタイルを敷き詰めたフロアとした。上段の例は、カメラを前後に傾けただけで、左右には傾けずに撮影した例である。いずれ



図 2 床画像の撮影シーンと画面ショット

の方法も、ほぼ、画像変換に成功しているが、重力ベクトルによる方法の方が、若干、歪が大きい。これは、高さ h を計測できずに、標準値を利用しているためであると考えられる。一方、中段の例は、カメラを左右方向にも傾けた例である。重力ベクトルによる方法では、左右の傾きは考慮できていないため、変換後の画像は斜めに歪んでいる。ARCore を用いた手法では、左右の傾きにも対応して、画像変換に成功していることがわかる。下段の例は、高さ 0.69m という想定外の位置で撮影した時の結果である。重力ベクトルによる方法では、高さが想定と大きく異なるため、正しく歪補正できていないが、ARCore を用いた手法では、高さも検知して、正しく補正を行うことができた。なお、変換画像の左右の下側に現れた黒色の三角形領域は、画像の撮影範囲外であることを表している。

4. 実験

4.1 画像照合実験

初期の性能評価実験として、2 枚の画像間の照合実験を行う。2 枚のうち、1 枚の画像は標準画像を想定して、床面に水平に保持したスマートフォンのカメラアプリを用いて、高精細に撮影した画像とした。他方は、今回実装した ARCore を利用して撮影した床画像である。実験には、Pixel 2 を利用し、標準画像の画素数は 3024×4032 、ARCore から撮影した画像の画素数は、 1080×1920 である。標準画像は 4 分の 1 に縮小し、比較画像は 2 分の 1 に縮小することで、画像間の倍率の違いを最小限に制限した。撮影場所は、スーパーマーケット、家電ショップ、レストランモールなどである。特徴点抽出には、著者らが拡張した Balanced ORB [3] を用い、RANSAC による相似三角形をなす 3 点を探索する手法を用いた。

図 4 に照合例を示す。撮影場所は、スーパーマーケットの雑貨売り場であり、床面はタイル地であるが、経年変化で傷や汚れがあり、特徴点抽出の困難さは中程度である。(2) の比較画像は、自然な形で端末を保持して撮影した画像であり、3 次元の歪が大きい。(3) 以降の照合結果の図では、抽出された特徴点を小円で示し、2 画像の照合に成功した場合に、小円の間を直線で連結している。(3) の歪補正無しの場合、照合に成功しているのは、比較的、歪が小さい下側の画像だけであり、奥に行くにつれ、歪が大き

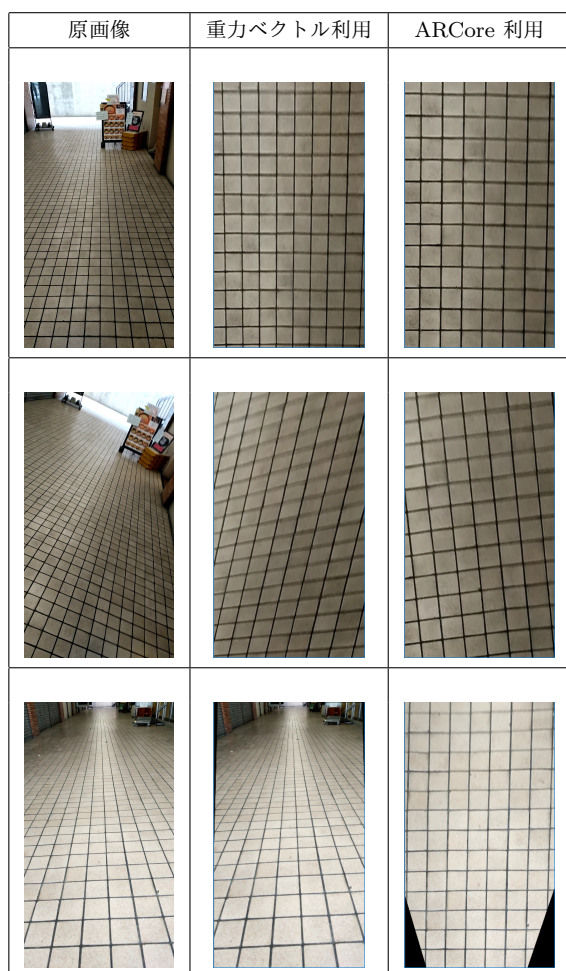


図 3 各手法による床画像の歪補正例

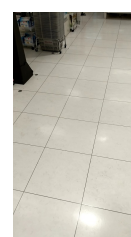
なって、照合が成立しなくなる。照合に成功した相似三角形の数は 42 個であった。(4) の重力ベクトルを利用した補正では、ある程度までの画像の補正に成功し、相似三角形の照合数も 92 個に増加する。ただし、照合している特徴点の位置は、ある程度、まとまった領域に集中してしまっている。(5) の ARCore を利用した補正では、補正が適切に行え、タイルが正方形に近い形に復元できている。これにより、歪が小さい領域が増え、相似三角形の照合数も 266 個になり、多くの対応点が検出できている。

本論文の提案手法では、ARCore のプレビュー画面から画像を抽出することにより、解像度の低い画像となり、照合に不安を残していたが、特徴点の計算および特徴点マッチングを行うための十分な情報量を含んでいることを示すことができた。また、歪補正無しの場合には、一部の画像だけが照合したことに対し、補正後の画像では、広い領域で照合に成功しており、補正が効果的に働いたことを実証することができた。

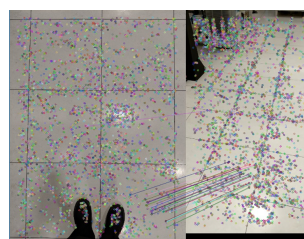
他の画像比較においても、上記と同様の傾向が見られた。さらに、補正無し画像で照合に失敗した場合であっても、ARCore による補正で、特徴点マッチングに成功する例も観察され、歪補正の有効性を示した。



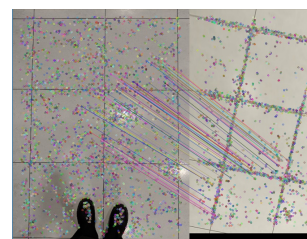
(1) 標準画像



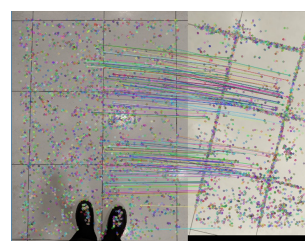
(2) 比較画像



(3) 歪補正無し



(4) 重力ベクトル利用



(5) ARCore 利用

図 4 床表面の照合例

表 1 位置推定実験の結果

データベース	歪補正	照合率	平均スコア	偏差
小規模	無し	23/26	0.173	0.0504
	重力	23/26	0.221	0.0551
	ARCore	26/26	0.240	0.0375
中規模	無し	72/118	0.0765	0.0228
	重力	56/118	0.0884	0.0247
	ARCore	77/118	0.0932	0.0194

4.2 位置推定実験

床画像データベースと、撮影画像を比較して、照合結果から位置推定を行う実験を実施した。実験では、小規模と中規模の 2 種類の床画像データベースを用意した。

(1) 小規模: スーパーマーケットの雑貨売り場の床画像データベース。横 2m, 縦 7.5m のタイル地の床に対し、10 枚の標準画像を持つ。

(2) 中規模: 法政大学西館 4 階廊下の床画像データベース。幅 2m, 全長 87.5m の石材調の床に対し、116 枚の標準画像を持つ。

比較対象の画像は、小規模データベースに対して 26 枚、中規模データベースに対して 118 枚を用意した。照合結果を表 1 に示す。

ARCore を利用した歪補正手法については、小規模、中規模のいずれのケースについてにおいても、補正無し、あるいは、重力ベクトルによる歪補正手法より高い照合率を示

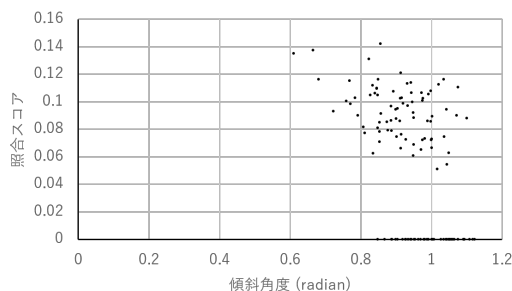


図 5 端末の傾斜角度と照合スコア

した．特に，小規模データベースについては，全ての例で，位置推定に成功している．これに加えて，この実験では，照合率以外で，照合スコアについても補正手法の差異が出た．照合スコアとは，著者らの論文 [3] で示した妥当性評価指標であり，照合画像結果を，ホモグラフィ変換で重ね合わせた時に，対応点の特徴量が十分に近い値になった率で表している．その論文では，正例の照合結果において，0.1 以上の数値になり，負例において，0.03 以下となり，閾値を 0.06-0.08 に設定すると良いと考察した．ARCore を利用した歪補正では，小規模データベースに対して，平均照合スコアが 0.240，中規模データベースに対して，0.0932 あり，他の手法より，高いスコアを残すと同時に，照合の閾値の基準も満たしている．以上の結果から，ARCore を利用した歪補正手法は，他手法に比較して，有効な手法であると言える．

しかし，中規模データベースについての照合率が 65% という数値は，決して高い照合率ではない．これについては，約 100m の廊下を，次々と撮影していったため，撮影時の手振れが大きくなったことと，端末の傾斜角度（カメラ光軸と水平面の法線のなす角度）が大きくなったことが理由としてあげられる．図 5 に，ARCore を利用した歪補正手法について，端末の傾斜角度と照合スコアの関係を示す．照合スコアが 0 の点は，照合に失敗した点であるが，傾斜角度 0.84 radian 以下には照合の失敗はない．一方，傾斜角度が 0.9 radian を超えると，照合スコアが低下する傾向を確認できる．傾斜角度が大きくなると，必然的に，遠方の小画像を拡大することになり，画像特徴量を抽出するのに十分な解像度が得られず，照合に失敗する確率が高くなるのだと考える．

5. 考察

本稿では，画像の 3 次元歪を補正する手法について提案と実験を行った．Kurz らの手法では，重力ベクトルとホモグラフィ行列を利用して画像補正を行うものの，画像の倍率や変換画像の中心位置などについて十分な考察がされておらず，経験的な数値で実装せざるをえなかった [5][6]．これに対し，著者らの過去の論文において，重力ベクトル

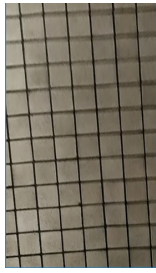
を用いて，より安定したホモグラフィ行列を計算する手法の提案を行った [4]．しかし，この手法は，端末の前後の傾きについてだけを補正しており，左右の傾きに関する補正を行っていなかった．さらに，端末と床面の高さについての情報が取得できないことにより，撮影条件によっては，補正に失敗することがあった．本稿では，これらの欠点を補い，Google 社の提供する ARCore プラットフォームを利用して，より精度の高い 3 次元歪の補正手法を提案した．ARCore は，端末の姿勢行列と，端末と床面の高さ情報を提供するため，歪補正のアルゴリズムは明確になり，精度の高い補正を実現できることが示された．

歪補正を行うためには，画素の補間が必要であり，特徴点抽出の妨げになることも懸念される．特に，ARCore を利用した実装では，カメラデバイスの利用権が ARCore 側にあり，通常の静止画の解像度を持つ画像の撮影を行えない．荒い画像で，かつ，歪補正のための画素の補間を行うことで，画像照合に必要な画像特徴が失われることも懸念された．しかし，実際には，様々な床画像データベースとの照合に成功しており，十分な画質であることが示された．

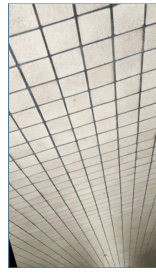
Kurz らの論文にも述べられているが，歪補正を行うことは，単に画像を見やすく整理するだけではなく，画像照合や画像トラッキングの性能向上に役立つ [5][6]．3 次元歪を持った画像を照合するには，4 点から導かれるホモグラフィ行列が必要になる．正しく照合された特徴点ペアが w の確率で存在するとすると，RANSAC で，正しい 4 点が選択される確率は， w^4 になる．床指紋照合の問題においては， $w = 0.02$ 程度であるので， 1.6×10^{-7} という極めて低い確率になり，RANSAC の選択回数を多くする必要がある．一方，3 次元歪が補正されれば，画像照合には相似三角形を利用すればよく，3 点の特徴点ペアがあればよい．よって， w^3 の確率で正しいペアが選択できるため，格段に照合速度を向上させることができる．本論文で，歪補正が画質劣化に大きな影響がないことが示され，より高速な照合システムの実現が可能になったと言える．

現在の実装では，画像の中心位置にある矩形領域を対象にして，歪補正画像を作成している．端末の傾斜角度が小さく，足元を写した写真であれば，これで十分であるが，端末の傾斜角度が多きく，視点が遠方にある場合には，画像の中心は遠方となり，画素数の小さい領域を拡大することになる．床指紋照合を行いたい画像は，遠方ではなく，足元に近い画像であることから，傾斜角度によって，画像中心ではなく，より足元に近い画像を歪補正に利用する方が，効果的であると考えられる．この点については，今後の課題としたい．

本稿は，重力ベクトルによる歪補正手法を改善して，ARCore を利用する手法について述べたが，その改善の要点は，左右の傾きに対応することと，端末の高さを知ることであった．重力ベクトルから，画像の高さを知ることが



(1) 左右の傾きの補正



(2) 高さ h が小さい時の補正

図 6 重力ベクトルを用いた改善版の歪補正例

できないが、端末の左右の傾きについては、知ることができる。以前の論文で、左右の傾き補正について考慮していなかっただけであり、本稿で述べたものと同様の手法で、左右の傾き補正を行うことができる。式 (8) の H_0 は、本質的に、式 (12) の E の逆行列と等価である。座標軸についてだけ変更を加えれば、本稿の論旨と同じ手法を適用して、歪補正可能である。図 6(1) に、重力ベクトルを用いて、左右の傾きに対応した歪補正の結果を示す。図 3 の中段と同じ画像を用いた補正結果であるが、格子模様が斜めになることなく、ARCore を用いた補正と同等の補正に成功していることがわかる。一方、図 6(2) は、高さ h が標準値より小さい場合で、図 3 の下段と同じ画像を用いた補正結果である。高さ h が標準値より大きく異なる時には、補正はうまく働いていない。以上のように、重力ベクトルだけでは、高さ h を取得できない課題が残るが、ARCore が動作しない端末で、重力ベクトルは取得できる端末においては、有効に利用できると考える。また、ARCore を起動しない場合であれば、高精細な静止画が撮影できるため、状況によって、重力ベクトルを利用する手法も有用になると考える。

ARCore を利用すると、静止画を撮影できない欠点があったが、本稿の実験を通して、プレビュー映像を利用しても、床指紋照合が行えることがわかり、新たな利用方法を検討できるようになった。すなわち、スナップ写真から位置推定をするだけでなく、連続映像を用いて、床指紋による継続的な経路トラッキングができる可能性が検討できるようになった。これについては、手振れ回避の問題も含めて、今後の検討課題である。

ARCore プラットフォームの本来の目的は、仮想現実感を実現するための支援ツールである。著者らは、これを床指紋照合に利用したわけであるが、逆に、床指紋照合に利用することで、仮想現実感システムに貢献できるとも考えている。すなわち、容易に床の位置同定を正確に行えることになれば、仮想現実感のオブジェクトを、マーカーなしに、正確な場所に設置できることになる。正確な位置を利用した仮想現実感のサービスについても、今後、検討を進めたい。

6. 結論

床面の特徴点抽出を行い、床指紋として照合する研究において課題であった画像の 3 次元歪を補正する手法について述べた。従来手法として、重力ベクトルを利用して歪補正する手法があるが、端末の床面から高さ情報が未知であること、左右方向の傾きを考慮していないことなどが課題となっていた。そこで、ARCore を利用して平面を認識し、高さとカメラの姿勢行列を取得することで、補正精度を高めることができることを示した。3 次元のモデルを忠実に計算することで、前後に加えて左右の傾きに対応したホモグラフィ行列を計算することができるようになり、床照合に十分な程度に、歪補正できることを確認した。実験を通して、歪補正を行った画像を利用することで、床画像データベースとの照合精度も高まり、位置推定の成功確率が高まることが示された。今後の課題として、撮影画像中で、足元に近い部分の画像を歪補正の対象とするように改良することや、リアルタイムに画像補正しながら、利用者の動的なトラッキングを可能にすることなどがあげられる。

謝辞 本研究は JSPS 科研費 JP17K00138 の助成を受けたものです。

参考文献

- [1] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, Vol. 24, No. 6, pp. 381–395, June 1981.
- [2] 藤田悟, 内田薫. 床指紋を用いた位置推定. マルチメディア, 分散, 協調とモバイルシンポジウム (DICOM2016), pp. 1244–1250, 2016.
- [3] 藤田悟, 藤田貴大, 内田薫. 床指紋: 床の模様に基づく位置推定. 情報処理学会論文誌, Vol. 58, No. 12, pp. 2023–2033, 2017.
- [4] Satoru Fujita, Tomoko Fujita, and Kaoru Uchida. Floor fingerprint verification using a gravity-aware smartphone. In *Proceedings of the 5th IIAE International Conference on Intelligent Systems and Image Processing*, pp. 311–318, 2017.
- [5] Daniel Kurz and Selim Benhimane. Gravity-aware handheld augmented reality. In *Mixed and Augmented Reality (ISMAR), 2011 10th IEEE International Symposium on*, pp. 111–120. IEEE, 2011.
- [6] Daniel Kurz and Selim Benhimane. Handheld augmented reality involving gravity measurements. *Computers & Graphics*, Vol. 36, No. 7, pp. 866 – 883, 2012.
- [7] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *International Conference on Computer Vision*, Barcelona, 2011.