

距離画像による空間情報マッチングに基づく マーカレス AR システムの設計と実装

山川 健司¹ 梶 克彦¹ 河口 信夫¹

概要：実世界の映像にデジタル情報を重畳表示する AR (拡張現実感) には、姿勢追従や位置推定が必要となる。本研究では室内におけるマーカレス AR を取り扱う。マーカレス AR の課題の 1 つとして環境の変化が挙げられる。例えば、環境光の変化、オブジェクトの見え隠れ、移動などにより姿勢追従や位置推定の精度が低下する。これらに強いロバストな位置推定手法について検討する。室内には天井、壁、机などの平面オブジェクトが多く存在する。そこで、これらの平面の組み合わせのパターンに注目する。本稿では、平面同定と、平面によって構成される 3 次元空間情報 (以下、シーン) の同定手法を提案する。シーンに含まれる平面の数、平面間の角度、平行な平面間の距離をシーンの特徴量として用いる。空間情報の取得には、距離画像センサを用いる。評価実験では、実際の室内環境を用いてマッチングを行った。結果としてはすべての学習済み環境が正しく同定でき、未学習の環境は該当なしの結果を得た。

Markerless AR System Based on Spatial Information Matching Using Depth Image

KENJI YAMAKAWA¹ KATSUHIKO KAJI¹ NOBUO KAWAGUCHI¹

1. はじめに

近年、拡張現実感 (Augmented Reality, AR) と呼ばれる技術に注目が集まっている。AR とは、実世界の映像にデジタル情報を重畳表示する技術である。主にナビゲーションシステムやコンピュータゲーム等で、現実には人間が知覚できない情報を具現化するために利用されている。AR によってデジタル情報を適切に表示するためには、大きく分けて 2 つの必要な技術がある。1 つが実世界での姿勢の推定及びリアルタイムのトラッキングである。重畳表示するイメージの姿勢が実世界と連動して変化することで臨場感のある表現が可能である。この技術はシステム実行時に得られる情報を用いて実現が可能である。もう 1 つが実世界での位置推定及びオブジェクト同定である。AR 対象毎に適切なデジタル情報を切り替えて表示するためには欠かせない。この技術は事前に学習した位置情報と同定することで実現が可能である。

本研究では室内におけるマーカレス AR に注目する [1]。

マーカレス AR の課題の 1 つとして環境の変化が挙げられる。時間や撮影位置によって日差しや照明等の環境光及びオブジェクトの見え方は変化する。更に、オブジェクトが移動する可能性も考えられる。これにより、位置推定の認識率が低下してしまう。そこで、本研究では環境変化に対してロバストなマーカレス AR のための位置推定を目標とする。本稿では、特に環境光や撮影位置の影響を低減することを目標とし、AR 対象はオブジェクトの移動等がない静的な環境を前提とする。

環境光の影響を低減するために、距離画像センサを用いる。距離画像とは、撮影点から物体表面までの距離によって構成される画像である。距離画像センサは赤外線を照射し、その反射光から距離を計測するため、環境光の影響を受けない。距離画像はポイントクラウドに変換可能である。ポイントクラウドとは 3 次元空間に分布する点の集合であり、オブジェクトの 3 次元形状を表現するためによく利用される。

本稿では、室内の数あるオブジェクトの中で、最も基本的だが多く存在する平面オブジェクトに注目し、3 次元空

¹ 名古屋大学大学院工学研究科
Graduate School of Engineering, Nagoya University

間情報を用いた位置推定及び平面オブジェクト同定の手法を提案する。

2. 関連研究

ARの実現手法はロケーションベースの手法とビジョンベースの手法の2つに大別される。

2.1 ロケーションベース AR

本手法は加速度センサ・ジャイロセンサ・GPS等を用いる手法である。これら位置センサにより、大規模環境での位置や姿勢の推定が容易に実現できる。しかし、センサの誤差の蓄積により、大まかな推定しかできない。現在向いている方向にある建物は何か等を示す屋外ナビゲーションシステムによく利用されている手法である。本研究では室内を対象としているため位置センサを主軸として用いることは難しい。

2.2 ビジョンベース AR

本手法はカメラを用いる手法である。前述の手法に対して、小規模環境における精密な位置及び姿勢の推定が可能である。次の2種類の手法に分けられる。

2.2.1 マーカ型 AR

マーカ型 AR は実世界に AR マーカを設置し、それをカメラで認識する手法である。AR マーカとは特殊なパターンが印刷されたマーカのことである。パターン認識により位置推定が、そしてマーカの変形によって姿勢推定が可能である。しかし、マーカを事前に設置する手間が生じ、またマーカ設置が困難な場所への AR が不可能となる。さらに、必ずマーカが見えていなくてはならず認識可能範囲が制限される。

加藤ら [2] は ARToolKit を開発した。ARToolKit は黒い太枠の内側に特殊なパターンが描かれたマーカを画像処理によって認識し、位置及び姿勢の推定を行う。これにより、マーカの座標系に合わせた 3 次元モデルのリアルタイム重畳表示が可能である。

2.2.2 マーカレス AR

マーカレス AR は上述のようなマーカを用いず、撮影した画像中に存在する自然特徴点のパターンを認識する手法である。事前準備の必要がなく実世界との親和性が高い。

Klein ら [3] は未知の小規模空間における単眼カメラの姿勢推定手法、Parallel Tracking and Mapping (PTAM) を提案した。画像から 1000 個程度の自然特徴点を抽出し、その中で繰り返し観測できる特徴点から 3 次元情報を計算し、実世界とカメラの座標関係をリアルタイムに推定する。Castle ら [4] は PTAM にマップの記憶及び識別機能を追加した Parallel Tracking and Multiple Mapping (PTAMM) を開発した。記憶した画像と現在の入力画像の特徴量記述子を比較することにより、マップを同定する。これにより、

場所に応じたマーカレス AR が可能である。しかし、これらの手法は、撮影画像中に安定して観測可能な自然特徴点が多く含まれていなければならない、環境光の変化等により認識精度が悪くなる。

Newcombe ら [5] は単眼カメラによって位置推定を行うと同時に、実世界の 3 次元形状をリアルタイムに復元する Dense Tracking and Mapping (DTAM) を提案した。DTAM では画像中の特徴点による疎な特徴量比較ではなく、全画素に対して隣接フレームとの違いを密に比較する。計算量が多いため、GPU を利用してリアルタイムな処理を実現している。

最近では、低価格化に伴って距離画像センサも AR に利用されている。Izadi ら [6] は距離画像センサによって位置推定を行うと同時に、実世界の 3 次元形状をリアルタイムに復元する KinectFusion を発表した。距離画像センサから取得したポイントクラウドの全点に対して Iterative Closest Point (ICP) アルゴリズム [7] を適用し、位置合わせを行う。リアルタイム処理のため GPU を利用している。復元した 3 次元形状を用いて物の見え隠れや物理法則を反映した AR が可能である。しかし、復元した 3 次元形状を学習し、後に識別及び同定する手法については言及されていない。

3. 提案手法

本研究では距離画像から得られる室内 3 次元空間情報を用いて位置推定及び平面オブジェクト同定を行う。AR 対象とその周囲の空間情報をポイントクラウドで取扱い、以下ではこのようなポイントクラウドをシーンと呼ぶ。事前に収集したシーンと現在見ているシーンのマッチングを行うことで、位置推定及びオブジェクト同定を行う。

シーン同定について、Iterative Closest Point (ICP) アルゴリズム等の点レベルでの同定はコストが高い。シーンに特徴量を見出し同定を行わなくてはならない。本章では、シーン中の特徴量について述べた後、それを用いた空間情報のマッチング手法について説明する。

3.1 シーンのモデル化

図 1 に示すように、室内には天井や床、壁、机、家電製品等、平面が多く存在する。平面は大きければ大きいほど移動する可能性が少ない。そこで、本研究ではそのような複数の平面の組み合わせのパターンによってシーン及び平面の同定を行う。本稿では、撮影するシーンに複数の平面が含まれていることを前提条件とし、取扱うオブジェクトはすべて平面に限定する。

シーン中の平面の組み合わせパターンの特徴として、次の 3 つを利用する。

- 平面の数
- 平面間のなす角度

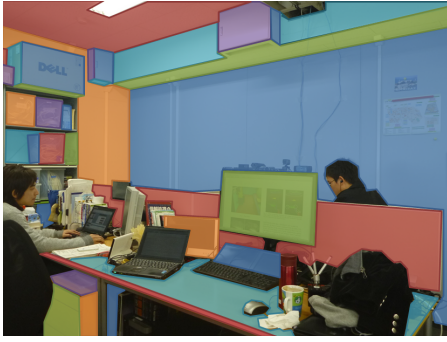


図 1 室内の平面オブジェクトの例

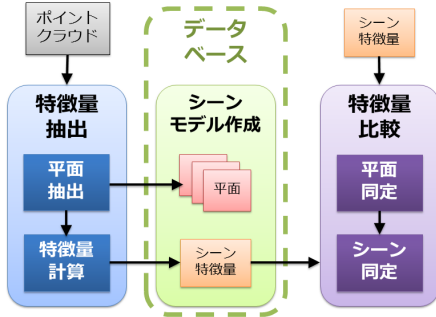


図 2 空間情報マッチング手法の概要

- 平行な平面間の距離

平面の数は撮影位置や姿勢によって観測領域が変動する特徴量であり、シーンモデルの候補をだまかに絞ることができる。平面間のなす角度は撮影位置によらない。ただし、室内の平面は 0° または 90° をなしている場合が多い。前者の 0° の場合に対応するため、3つ目の特徴量として平行平面間の距離を用いる。これも撮影位置によらない特徴量である。

3.2 空間情報マッチング手法

シーン同定及び平面同定手法について述べる。提案手法は図2に示すように、特徴量抽出、シーンモデル作成、特徴量比較の3段階からなる。

3.3 特徴量抽出

特徴量抽出は、事前処理、平面抽出と特徴量計算の3つの手順からなる。事前処理及び平面抽出ステップではRusu [8]の手法を参考に行っている。ただし、Rusuはすべての処理をポイントクラウドを用いて行っている。一般にポイントクラウドの処理はコストが高い。そこで、本研究ではコストの低い画像処理を可能な限り用いる。図3に示すように、ポイントクラウドは距離画像中の画素 $f(i, j)$ を3次元空間中の点ベクトル $\mathbf{p}(i, j)$ に変換したものの集合である。すなわち、ポイントクラウドの各点における処理は画像処理に対応させることができる。

3.3.1 事前処理

平面抽出に先立ち、ポイントクラウドの平滑化と法線方

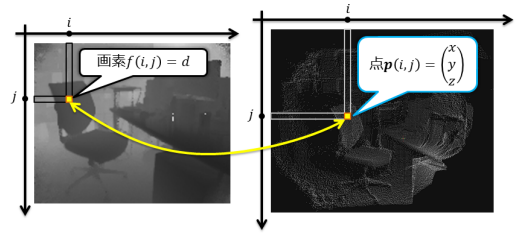


図 3 距離画像とポイントクラウドの対応関係

向計算を行う。

平滑化は画像処理を適用する。距離画像またはポイントクラウドの視点から見た奥行方向成分に対してバイラテラルフィルタ [9] を適用する。バイラテラルフィルタとは、画像において変化の急峻な部分を避け、ノイズ除去を行う手法である。入力画素を $f(i, j)$ とすると、平滑化後の画素 $g(i, j)$ は式1, 2で表される。

$$g(i, j) = \frac{\sum_{m=-w}^w \sum_{n=-w}^w f(i+m, j+n)h(i, j, m, n)}{\sum_{m=-w}^w \sum_{n=-w}^w h(i, j, m, n)} \quad (1)$$

$$h(i, j, m, n) = \exp\left(-\frac{m^2 + n^2}{2\sigma_1^2}\right) \cdot \exp\left(-\frac{f(i, j) - f(i+m, j+n)}{2\sigma_2^2}\right) \quad (2)$$

ただし、 w はカーネルサイズ、 σ_1 がガウシアンフィルタを制御する正規分布の標準偏差、 σ_2 が輝度差を制御する正規分布の標準偏差である。本研究では経験的に $w = 1, \sigma_1 = \sigma_2 = 1000$ としている。

その後、ポイントクラウドを用いた処理により、法線方向計算を行う。平滑化したポイントクラウドの各点 $\mathbf{p}(i, j)$ における法線 $\mathbf{n}(i, j)$ を求める。 $\mathbf{p}(i, j)$ の近傍点 $\mathbf{p}_k(i, j) (k = 1, 2, \dots, N)$ を探索し、集合 \mathbf{X} を作成する。 \mathbf{X} が形成する局所的な面に対して主成分分析 (Principal Component Analysis, PCA) を行い、第3主成分 \mathbf{v}_3 を求める。そのためには、 \mathbf{X} の重心点 $\boldsymbol{\mu}$ と共分散行列 \mathbf{C} を式3, 4により求める。

$$\boldsymbol{\mu} = \frac{1}{N} \sum_{i=1}^N \mathbf{X} \quad (3)$$

$$\mathbf{C} = \sum_{i=1}^N (\mathbf{X} - \boldsymbol{\mu})^T (\mathbf{X} - \boldsymbol{\mu}) \quad (4)$$

\mathbf{C} の最小固有値に対応する固有ベクトルが \mathbf{v}_3 である。本研究では経験的に $N = 20$ としている。

\mathbf{v}_3 の向きは一意的に定まらない。そこで、式5により、視点 \mathbf{v}_{vp} 方向を向いた法線ベクトル $\mathbf{n}(i, j)$ を求める。

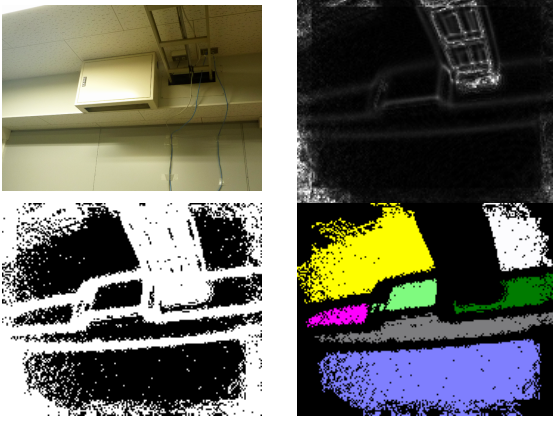


図 4 平面領域推定の流れ: 対象環境 (左上), 曲率画像 (右上), 2 値化画像 (左下), ラベリング処理後 (右下)

$$\mathbf{n}(i, j) = \begin{cases} -\mathbf{v}_3 & (\mathbf{v}_{vp} - p_z < 0) \\ \mathbf{v}_3 & (\text{otherwise}) \end{cases} \quad (5)$$

ただし, p_z は点 $\mathbf{p}(i, j)$ の視点 \mathbf{v}_{vp} から見た奥行方向成分である.

3.3.2 平面抽出

本手法では, 画像処理により大まかに平面の存在位置を絞り, その後ポイントクラウドでの処理により, 適当な平面を抽出する.

まず, ポイントクラウドの各点 $\mathbf{p}(i, j)$ について, それぞれ $\mathbf{p}(i+1, j), \mathbf{p}(i, j+1)$ が存在する場合, 法線ベクトル $\mathbf{n}(i, j)$ と $\mathbf{n}(i+1, j), \mathbf{n}(i, j+1)$ のなす角度を求め, これらの平均から成る曲率画像を作成する. 次に, この画像を閾値 $a_{\text{rgn,th}}$ で 2 値化する. その後, 曲率が小さい部分に対してラベリング処理を行い, $N_{\text{rgn,th}}$ pixel 以上の平面領域 n_{rgn} 個を推定する. この一連の流れの様子を図 4 に示す. 本研究では経験的に $a_{\text{rgn,th}} = 3.5^\circ, N_{\text{rgn,th}} = 100$ としている.

上述のラベリング処理では, 見え隠れによって同一の平面が異なる領域に分割されている場合がある. また, 分割された各領域内のすべての点が必ずしも同一平面を形成しているとは限らない. そこで, ラベリングした n_{rgn} 個の領域を基にポイントクラウドを用いて平面を抽出する. その処理手順を表 1 に記す.

まず初めに, 平面方程式 (式 6) を推定する. 平面方程式推定手法には, RANSAC 法 [10] がある. RANSAC 法では, 同一線上にない 3 点をランダムに選択し, 平面を作成する. それを繰り返して最適な平面を推定する. しかし, ポイントクラウド全体から 3 点を選択するため, 不適当な平面を推定する試行が多い. 選択される 3 点は離れていない方が好ましい. Dube ら [11] はポイントクラウド全体からランダムに 1 点を選択し, それを中心とする所定の半径の円上から, 残り 2 点を選択している. しかし, ある 1 点から指定距離だけ離れた点の探索コストは高い. そこで本

表 1 平面推定アルゴリズム

Algorithm Plane_Estimation

$(R, C, a_{\text{inl,th}}, d_{\text{inl,th}}, K, N_{\text{blk}}, N_{\text{inl,th}})$:

```

1:   $L := \emptyset$ 
2:  for every region  $r_i \in R$  in descending order by area do
3:     $N_{\text{inl,best}} = N_{\text{inl,th}}$ 
4:     $P_{\text{inl,best}} = \emptyset$ 
5:    for  $k = 1$  to  $K$  do
6:      find  $N_{\text{blk}} \times N_{\text{blk}}$  block  $b \subset r_i$  randomly
7:      estimate equation coefficients  $\mathbf{n}_P, \delta_P$  for plane
        by computing PCA on  $b$ 
8:       $N_{\text{inl}} = 0$ 
9:       $P_{\text{inl}} = \emptyset$ 
10:     for every point  $\mathbf{p} \in C$ 
11:       if  $\mathbf{n}_P \cdot \mathbf{n} > \cos a_{\text{inl,th}} \cap \mathbf{n}_P \cdot \mathbf{p} + \delta_P < d_{\text{inl,th}}$  do
12:         increment  $N_{\text{inl}}$ 
13:          $P_{\text{inl}} := P_{\text{inl}} \cup \{\mathbf{p}\}$ 
14:       end if
15:     end for
16:     if  $N_{\text{inl,best}} < N_{\text{inl}}$  do
17:        $N_{\text{inl,best}} := N_{\text{inl}}$ 
18:        $P_{\text{inl,best}} := P_{\text{inl}}$ 
19:     end if
20:   end for
21:   extract  $P_{\text{inl,best}}$  from  $C$ 
22:    $L := L \cup \{P_{\text{inl,best}}\}$ 
23: end for
24: return  $L$ 

```

研究では, 領域に内包される N_{blk} pixel 四方のブロックをランダムに選択し (6 行目), それの主成分分析により平面 P を推定する (7 行目). 本研究では, $N_{\text{blk}} = 4$ としている.

$$\alpha_P x + \beta_P y + \gamma_P z + \delta_P = 0 \quad (6)$$

なお, $\alpha_P, \beta_P, \gamma_P, \delta_P$ はヘッセ標準形で表現される. すなわち, 平面 P の法線ベクトル \mathbf{n}_P , 点 \mathbf{p} と平面 P の間距離 d はそれぞれ式 7, 8, 9 で表される.

$$\mathbf{n}_P = (\alpha_P, \beta_P, \gamma_P) \quad (7)$$

$$|\mathbf{n}_P| = 1 \quad (8)$$

$$d = \mathbf{n}_P \cdot \mathbf{p} + \delta_P \quad (9)$$

次に, 推定した平面方程式に含まれる点をポイントクラウド全体 C から探索する (10-15 行目). 具体的には, 平面の法線 \mathbf{n}_P とのなす角度が $a_{\text{inl,th}}$ 以下かつ平面 P からの距離が閾値 $d_{\text{inl,th}}$ 以下となる点 \mathbf{p} の数 N_{inl} を数える (11-13 行目). 図 5 にその様子を示す.

以上の平面方程式推定と含有点探索を K 回繰り返す (5-19 行目). N_{inl} が最も大きくなる平面方程式を正しい推

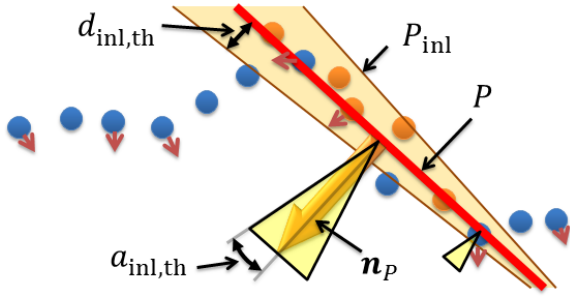


図 5 平面の含有点の探索の様子

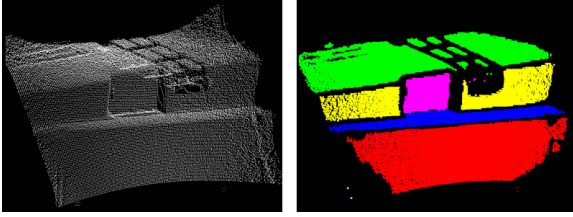


図 6 平面抽出の成功例: ポイントクラウド(左)と抽出平面(右)の様子

定と見なす. このときの含有点を C から抽出する (20 行目). N_{inl} が $N_{\text{inl,th}}$ 以下のとき, 適切な平面はないと見なす (3 行目).

以上の操作をすべての領域に対して実行し (2-22 行目), 最終的に n_{mrg} 個の平面を推定する (23 行目). $a_{\text{inl,th}}, d_{\text{inl,th}}$ は式 10, 11 のように定義する.

$$a_{\text{inl,th}} = \phi_{\text{inl,th}} \sqrt{p_z} \quad (10)$$

$$d_{\text{inl,th}} = \delta_{\text{inl,th}} p_z \quad (11)$$

ただし, p_z は点 p の視点から見た奥行方向成分である. 本研究では経験的に $K = 100, N_{\text{inl,th}} = 200, \phi_{\text{inl,th}} = 10^\circ/m, \delta_{\text{inl,th}} = 0.02$ としている. 正しく平面抽出された例を図 6 に示す.

3.3.3 特徴量計算

3.1 章で挙げた 3 つの特徴量:

- 平面の数 n
- 平面間のなす角度の LUT A
- 平行な平面間の距離の LUT D

を計算する. LUT は Look-up Table である.

平面 P_i, P_j 間の角度 $a_{i,j} (i, j = 1, 2, \dots, n; i < j)$ は平面の全組み合わせ ${}_n C_2$ に対して計算し, 式 12 に示す上三角行列を作成する.

$$A = \begin{pmatrix} 0 & a_{1,2} & \cdots & a_{1,n} \\ & 0 & \ddots & \vdots \\ & & & 0 & a_{n-1,n} \\ 0 & & & & 0 \end{pmatrix} \quad (12)$$

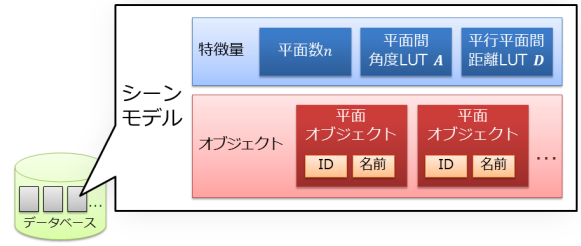


図 7 シーンモデル

平行平面 P_i, P_j 間の距離 $d_{i,j} (i, j = 1, 2, \dots, n; i < j)$ は平面方程式 (式 6) の左辺の第 4 項 $\delta_{P_i}, \delta_{P_j}$ より求める. ヘッセ標準形では $|\delta_{P_i}|, |\delta_{P_j}|$ は原点と平面間の距離に等しい. 平面が平行の場合, 2 つの距離の差は平面間距離とみなしてよい. そこで, 平行平面間距離は式 13 とする.

$$d_{i,j} = \begin{cases} \left| |\delta_{P_i}| - |\delta_{P_j}| \right| & (\text{if parallel}) \\ 0 & (\text{otherwise}) \end{cases} \quad (13)$$

本稿では, 平面間の角度が 15° 以下のとき平行とみなしている. これらから, 式 14 に示す上三角行列を作成する.

$$D = \begin{pmatrix} 0 & d_{1,2} & \cdots & d_{1,n} \\ & 0 & \ddots & \vdots \\ & & & 0 & d_{n-1,n} \\ 0 & & & & 0 \end{pmatrix} \quad (14)$$

3.4 シーンモデル作成

抽出した各平面オブジェクトに ID や名前等の識別情報を入力し, 計算した特徴量と併せて 1 つのシーンモデルとする. これをデータベースへ登録する. その様子を図 7 に示す.

同じ対象環境であっても, 撮影位置や姿勢により含有平面数が異なる. そこで本稿では, いくつか異なる視点からシーンを撮影し, シーンモデルを作成する.

3.5 特徴量比較

現在撮影しているシーンの特徴量抽出を行い, 入力シーンモデル m_{curr} を作成する. そして, m_{curr} とデータベース M 中の各モデル m_i の特徴量を比較する. 2 つのシーンの比較手順を表 2 に記す. 本稿では, 閾値 $a_{\text{th}}, d_{\text{th}}$ はそれぞれ $a_{\text{th}} = 20^\circ, d_{\text{th}} = 0.3m$ としている. m_{curr}, m_i のどちらにも平行な平面の組が含まれていない場合は, 平行平面間距離を無効にするために $d_{\text{best}} = 0$ とする. k は順列のインデックス番号 ($k = 1, 2, \dots, n!$) である. 12, 13 行目の行列 $A_{\text{curr}}^k, D_{\text{curr}}^k$ は行列の要素が順列 p_n^k に応じて再配置されていることを表す. 返される $a_{\text{best}}, d_{\text{best}}$ はマッチングスコアとして利用される. このスコアが小さいほど一致度が高いことを表す. スコアが最小となるとき, 平面の同定が可能である.

表 2 平面同定アルゴリズム

Algorithm Feature_Matching(m_i, m_{curr}) :

- 1: $a_{best} := a_{th}, p_{best} := null$
- 2: if neither m_i nor m_{curr} include any parallel planes do
- 3: $d_{best} := 0$
- 4: else
- 5: $d_{best} := d_{th}$
- 6: end if
- 7: if m_i doesn't have the same number of planes n as m_{curr}
- 8: return *null*
- 9: end if
- 10: create permutations p_n^k ($k = 1, 2, \dots, n!$)
- 11: for $k = 1$ to $n!$ do
- 12: $a_{max} := \max \left\{ |s_{u,v}| \mid S = [s_{u,v}] = A_i - A_{curr}^{p_n^k} \right\}$
- 13: $d_{max} := \max \left\{ |t_{u,v}| \mid T = [t_{u,v}] = D_i - D_{curr}^{p_n^k} \right\}$
- 14: if $a_{best} \geq a_{th} \cap d_{best} \geq d_{max}$ do
- 15: $p_{best} := p_n^k$
- 16: $a_{best} := a_{max}$
- 17: $d_{best} := d_{max}$
- 18: end if
- 19: end for
- 20: return ($p_{best}, a_{best}, d_{best}$)

表 3 シーン同定アルゴリズム

Algorithm Approximate_Localization(M, m_{curr}) :

- 1: $L := \emptyset$
- 2: for every model $m_i \in M$ do
- 3: $r := \text{Feature_Matching}(m_i, m_{curr})$
- 4: if $r \neq null$ do
- 5: $(p, a, d) := r$
- 6: if $a < a_{th} \cap d < d_{th}$ do
- 7: $L := L \cup \{m_i\}$
- 8: end if
- 9: end if
- 10: end for
- 11: return L

以上の比較手順をデータベース M 中の全モデルに対して適用する。その手順を表 3 に記す。マッチングスコアが最小となるモデルを探索し、シーン同定を行う。

4. 評価実験

本章では、3章で提案した手法を用いたシーン同定の実験とその結果について述べる。実験は Intel Xeon CPU 2.40GHz 搭載の Windows 7 上で行った。距離画像センサとして MesaImaging 社の SwissRanger SR4000(5m レンジ, 広角レンズ版)を用いた。

4.1 実環境でのシーン及び平面同定実験

実験に用いた環境の様子を図 8 に示す。対象環境として具体的には机 2 つと冷蔵庫, 配電盤を選択した。以下では

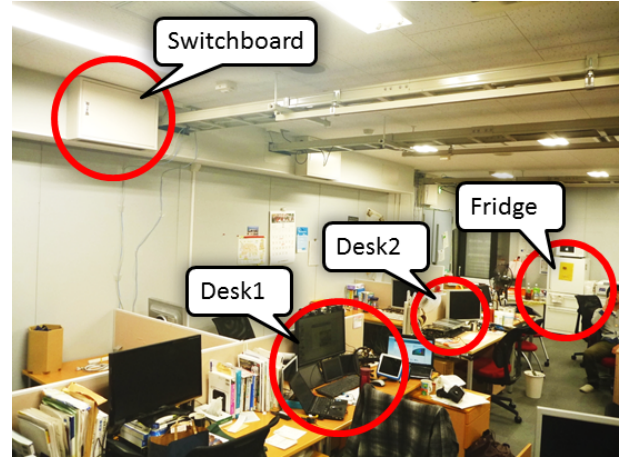


図 8 実験に用いた環境



図 9 対象環境: 机 1(左上), 机 2(右上), 机 3(左中央), 冷蔵庫(右中央), 配電盤(下)

順に“机 1”, “机 2”, “冷蔵庫”, “配電盤”と呼ぶ。これら 4 つの環境に対して、それぞれ左, 正面, 右の 3 視点からシーンを撮影し, 合計 12 個のシーンモデルを作成しデータベースへ登録する。その後, 正面から撮影した新しいシーンと比較を行う。データベースに登録された環境 4 つと登録されていない環境 1 つ(以下, “机 3”), 合計 5 個を新しい入力シーンとした。これらの様子を図 9 に示す。本実験では, シーン中のすべてのオブジェクトは固定されており, 平面オブジェクトはその多くの部分を露出していることを前提条件とする。

4.2 実験結果

ポイントクラウドからのシーンモデル 1 個の作成の平均所要時間は 765[ms] であった。入力ポイントクラウド 1 個とシーンモデル 12 個の比較の平均所要時間は 203[ms] で

表 4 シーン及び平面オブジェクト同定結果:
行は登録シーン, 列は検索対象.

データ ベース	入力 シーン (平面数)	学習済み				未登録 機 3 (4)
		機 1 (5)	機 2 (5)	冷蔵庫 (3)	配電盤 (5)	
機 1	左 (5)	O ₁	X		X	
	正面 (5)	O ₂	X		X	
	右 (4)					X
機 2	左 (7)					
	正面 (5)	X	O ₃		X	
	右 (6)					
冷蔵庫	左 (3)			O ₄		
	正面 (2)					
	右 (6)					
配電盤	左 (5)	X	X		O ₅	
	正面 (5)	X	X		O ₆	
	右 (7)					
推定結果		機 1	機 2	冷蔵庫	配電盤	なし

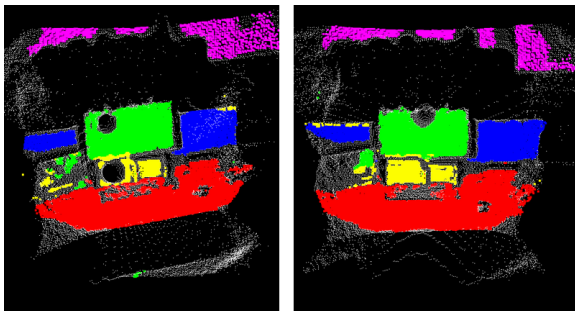


図 10 シーン同定及び平面同定の成功例: 入力シーン (左) とデータベース登録シーン (右). 同じ色は同じ平面であることを表す.

あった. すなわち, 距離画像センサを対象環境へ向けてから位置推定が完了するまで1[s]程度要することになる.

表 4 に同定結果を示す. 丸括弧内は平面数を表す. 空白欄は平面数の違いにより不一致と見なされた組み合わせを表す. 表中の X は平面間角度及び平行平面間距離の差の最大値が閾値を超えたために除外された組み合わせを表す. 表中の O は一致と見なされた組み合わせを表す. 表から, 4 つの環境すべてが正しく推定されていることがわかる. 成功例を図 10 に示す. また, 登録されていない“機 3”は該当なしと推定された. ただし, O₅ 及び O₆ では, 図 11 に示すように, 平面の同定が正しく行われなかった.

4.3 考察

本実験では, 平行な 2 平面が反対に同定される場合があった. これは特徴量として用いている距離や角度が絶対値で評価されており, 平行な平面が 2 枚しか存在しなかったためと考えられる. 平行でない平面間についても距離を計算, 評価対象とすることで改善されると思われる.

本実験環境では, シーンモデル 1 個とのマッチング処理に 17[ms] 程度要した. 比較するシーンモデル数に応じて

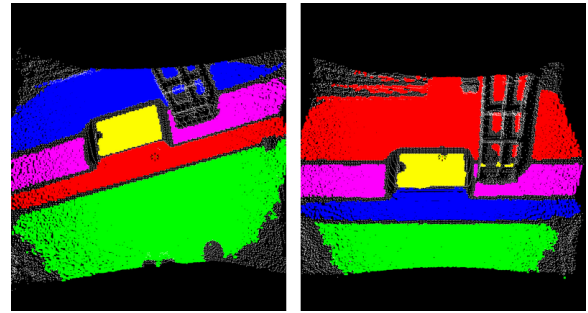


図 11 平面同定の失敗例: 入力シーン (左) とデータベース登録シーン (右).

表 5 デバイス一覧

種類	製品名
距離画像センサ	MesaImaging SR4000 (5m レンジ, 広角レンズ版)
Web カメラ	Logicool HD Pro Webcam C920
ディスプレイ	Century plus one DVI (LCD-8000DA)
ゲームパッドコントローラ	Elecom JC-U3312S シリーズ

処理時間は線形に増加する. よって登録するシーンモデル数が多い場合は, 本手法をそのまま用いることは難しい. 他の位置推定技術等を併用し, 比較するシーンモデル数を限定する必要があると考えられる. 例えば, 建物内の複数の部屋でのシーンモデルが登録されている場合, GPS や無線 LAN 電波強度による位置推定技術等を用いて比較対象を限定することが可能である.

5. AR システムの実装

現在, 本手法を用いた AR システムを実装している. 本 AR システムの概要を図 12 に示す. 空間情報マッチングが成功した場合は, 位置情報に基づき適切な AR イメージを表示する. その後は, 距離画像のトラッキング処理により, AR イメージの姿勢をリアルタイムに推定する. 本 AR システムのために, 図 13 に示す AR スコープを作成した. この AR スコープは, 表 5 に示すデバイスにより構成されている. 距離画像センサ以外は外装を取り外し, 別途 3D プリンタを用いて作成したケースでこれらのデバイスを一体化した. AR スコープの左右には持ち手がついており, 両方にゲームパッドコントローラに接続されたスイッチが搭載されている. このスイッチは, 表示コンテンツの変更等に利用する. USB ケーブルや電源ケーブル類は, AR スコープ下部からまとめて 1 本のケーブルとして取り出している.

6. まとめ

本稿では, 室内マーカレス AR における位置推定のロボスタ性の向上をめざし, 距離画像に基づく空間情報のマッチング手法を提案した. 環境光の影響を緩和するため, 距離画像センサを用いた. 室内の平面オブジェクトに注目

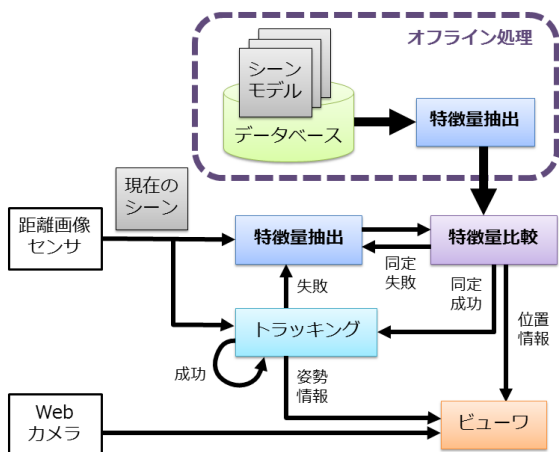


図 12 AR システムの動作フロー



図 13 AR スコープ: 表側 (上), 裏側 (中央), 動作例 (下)

し、その組み合わせのパターンからマッチングを行うため、特徴量として平面の数、平面間の角度、平行平面間の距離の3つを利用し、事前学習したシーンと現在のシーンの同

定手法について検討した。提案手法の評価実験では幾つか平面同定に失敗したもののシーン同定は良好な結果が得られた。

今後の課題として以下の点が挙げられる。

- 平面以外の形状、その他の特徴量の利用。室内には様々な形状のオブジェクトがあり、また色等の様々な特徴がある。今後はそれらを吟味し、提案手法のロバスト性を高めたい。
- 計算量の削減。平面推定や同定の段階では単純な全探索法を用いており効率が悪い。また、行列計算を伴う法線計算はコストが高い。今後はこれらアルゴリズムの改善や GPU の利用を検討したい。

参考文献

- [1] 山川 健司, 梶 克彦, 河口 信夫, 距離画像情報を用いた室内マーカレス AR とその応用, 情報処理学会 第 75 回全国大会講演論文集, pp.4.57-4.58, 2013.
- [2] H. Kato, M. Billinghurst, I. Poupyrev, K. Imamoto, K. Tachibana, Virtual Object Manipulation on a Table-Top AR Environment, *In Proceedings of the International Symposium on Augmented Reality (ISAR)*, pp.111-119, 2000.
- [3] G. Klein and D. Murray, Parallel Tracking and Mapping for Small AR Workspaces, *In Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR)*, pp.225-234, 2007.
- [4] R. Castle, G. Klein and D. Murray, Video-rate Localization in Multiple Maps for Wearable Augmented Reality, *In Proceedings of the International Symposium on Wearable Computers (ISWC)*, pp.15-22, 2008.
- [5] R. A. Newcombe, S. J. Lovegrove and A. J. Davison, DTAM: Dense Tracking and Mapping in Real-Time, *In Proceedings of the International Conference on Computer Vision (ICCV)*, pp.2320-2327, 2011.
- [6] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. A. Newcombe, P. Kohli, S. Shotton, J. Hodges, D. Freeman, A. J. Davison, and A. Fitzgibbon, KinectFusion: Real-time 3D Reconstruction and Interaction Using a Moving Depth Camera, *In Symposium on User Interface Software and Technology (UIST)*, pp.559-568, 2011.
- [7] P. J. Besl and N. D. McKay, A Method for Registration of 3-D Shapes, *In IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, Vol. 14, pp.239-256, 1992.
- [8] R. B. Rusu, Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments, *In PhD Thesis, Institut für Informatik der Technischen Universität München, Germany*, 2010.
- [9] C. Tomasi and R. Manduchi, Bilateral Filtering for Gray and Color Images, *In Proceedings of the International Conference on Computer Vision (ICCV)*, pp.839-846, 1998.
- [10] M. A. Fischler and R. C. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, *Communications of the ACM*, Vol. 24, pp.381-395, 1981.
- [11] D. Dube and A. Zell, Real-Time Plane Extraction from Depth Images with the Randomized Hough Transform, *In IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pp.1084-1091, 2011.