

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,a)} 梶 克彦¹ 河口 信夫¹

受付日 2013年10月10日, 採録日 2014年1月25日

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. AR の実現手法にはいくつか種類があるが, 本研究では室内におけるマーカレス AR を対象としたロバストな位置推定を最終目標とする. 本稿では, 環境光の影響を低減させるために距離画像センサを用い, 距離画像やポイントクラウドの処理に基づいて空間情報のマッチングを行う手法を提案する. 本研究では, 天井, 壁, 机等の室内の平面オブジェクトの多さに注目する. AR の対象とその周囲の空間情報 (以下, シーン) を平面の組合せのパターンで表現し, 観測可能な平面の数, 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算出する. 本稿では, 距離画像やポイントクラウドの処理によって, 撮影したシーンから平面を抽出する手法について説明する. その後, シーンの特徴量を用いた平面オブジェクト同定およびシーン同定手法を提案する. 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. その結果, 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.

キーワード: ポイントクラウド, 平面オブジェクト, PCA, RANSAC 法

Markerless AR System Based on Spatial Information Matching Using Depth Image

KENJI YAMAKAWA^{1,a)} KATSUHIKO KAJI¹ NOBUO KAWAGUCHI¹

Received: October 10, 2013, Accepted: January 25, 2014

Abstract: Augmented Reality (AR) is a technology to overlay some information at a related position in camera images. Hence, an estimation of position and direction of camera is necessary to achieve AR. Though there are several methods, our goal is to develop a robust estimation method for indoor Markerless AR. Particularly, we use a depth image sensor to avoid the influence of environmental lights, and we propose a method to match the spatial information by processing depth images and point clouds. In indoor environments, there are many planar objects, for example ceilings, walls, tables and so on. Therefore, we extract three features from the combinations of such planar objects to describe a scene (spatial information around a target of AR). The features are as follows: the number of observed planes, the angles between planes and the distances between parallel planes. In this paper, we introduce how to extract planes at first. Then, we propose how to identify planar objects and scenes by using such features. Experiments were made to show effectiveness of the method in a real indoor environment. As a result, all of the learned scenes were recognized properly and a non-learned scene was rejected.

Keywords: point cloud, planar objects, PCA, RANSAC method

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

¹ 名古屋大学大学院工学研究科
Graduate School of Engineering, Nagoya University, Nagoya,
Aichi 464-8601, Japan

^{a)} yamaken@ucl.nuee.nagoya-u.ac.jp

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には、大きく分けて2つの必要な技術がある。1つは実世界での姿勢推定およびリアルタイムのトラッキングである。この技術によって重畳表示するイメージの姿勢を実世界と連動して変化させることが可能になり、ARに臨場感が生まれる。もう1つは実世界での位置推定およびオブジェクト同定である。この技術によってARの対象となる環境やオブジェクトごとに表示するイメージの切替えが可能になり、ARによる表現の幅が広がる。

本研究では室内におけるマーカレスARに注目する[1]。室内ではそのときの撮影時間や位置によって、窓から入る光や照明の状態が異なり、オブジェクトの見え方は変化する。さらに、時間経過にともなってオブジェクトが移動している可能性も考えられる。このように、マーカレスARの課題の1つとして環境変化による位置推定の認識率の低下があげられる。そこで、本研究ではこの課題を解決するロバストな位置推定手法の提案を最終目標とする。そのために、まず本稿では環境光や撮影位置に影響されない位置推定手法について検討する。オブジェクトの移動については今後の研究課題とし、本稿ではオブジェクトの移動がない静的な環境を対象とする。

環境光の影響を低減させるために、本研究では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り、赤外線を照射する方式の距離画像センサは、環境光の影響を受けることなく安定して距離画像を撮影できる。一般に距離画像はポイントクラウドと呼ばれる、3次元空間に分布する点の集合に変換可能であり、3次元データ処理に利用されている。本稿では室内にあるオブジェクトの中で、最も基本的であるが、多く存在する平面オブジェクトに注目し、距離画像やポイントクラウドの処理を用いた位置推定および平面オブジェクト同定手法を提案する。

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

ロケーションベースAR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。これらの位置センサにより、位置推定や姿勢推定が実現できる。しかし、民生用センサでは精度が低く大まかな推定しかできない。そのため、精密な位置推定を必要としないシステムで利用されている。たとえば、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があげられる。本研究では室内を対象としており、位置推定精度の低い上述のセンサを主軸として用いることは難しい。

大淵ら[2]は旅行用ナビゲーション機器、旅ナビを開発した。旅ナビでは位置センサから端末の位置やカメラの方向を検出し、内蔵されている地図データから適切な地名やランドマーク、ガイドブックデータを参照する。取得した情報はカメラ映像中の適切な位置に重畳表示される。

2.2 ビジョンベース AR

ビジョンベースAR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ミリメートルオーダの精密な位置推定や姿勢推定が可能である。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は実世界に特殊なパターンが印刷されたARマーカを設置し、それをカメラで認識する手法である。画像認識により、マーカに印刷されたパターンから位置推定が、そしてマーカの変形から姿勢推定が可能である。

Katoら[3]はマーカ型ARシステム、ARToolKitを開発した。ARToolKitでは黒い太棒とその内側に特殊なパターンが描かれたARマーカを用いる。このマーカから位置推定や姿勢推定を行い、マーカのパターンの種類に応じたイメージをマーカの姿勢に合わせてリアルタイムに表示する。しかし、ARマーカを事前に設置する手間が生じ、またマーカの設置が困難な場所へのARが不可能である。さらに、必ずマーカが見えていなくてはならないため、認識可能範囲が制限される。

2.2.2 マーカレス AR

マーカレスARは上述のような人工的なマーカを用いず、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。ARの対象となる環境への事前準備が必要なく、実世界との親和性が高い。

Kleinら[4]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。PTAMでは画像から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[5]はPTAMにマップの記憶および識別機能を追加したParallel Tracking and Multiple Mapping (PTAMM)を開発した。PTAMMでは記憶した画像と現在の入力画像の特徴量比較により、マップを同定し、場所に応じたマーカレスARを実現している。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[6]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では画像中のいくつかの特徴点を用いた疎な特徴量比較ではなく、全画素を用いた密な特徴量比較を行う。

計算量は多いが、GPU の利用によりリアルタイムな処理を実現している。

最近では、低価格化にともなって距離画像センサも AR に利用されている。Izadi ら [7] 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。KinectFusion では距離画像センサから取得したポイントクラウドと過去に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 [8] を適用し、位置合わせを行う。こちらも計算量は多いが、GPU の利用によりリアルタイムな処理を実現している。KinectFusion によって復元された 3 次元形状は、物の見え隠れや物理法則を反映した AR に利用可能である。しかし、復元した 3 次元形状を学習し、後に識別およ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

本研究では室内で撮影した距離画像から得られる 3 次元空間情報を用いて位置推定および平面オブジェクト同定を行う。具体的に我々が取り扱う 3 次元空間は、AR の対象とするオブジェクトとその周囲の空間である。以下ではこれを距離画像センサから得られる「シーン」と呼ぶ。

提案手法の大まかな手順は次のとおりである。まず最初にポイントクラウドの処理により、シーンに関する情報を計算する。次に、事前に収集したシーン情報と現在見ているシーン情報のマッチングによって、位置推定およびオブジェクト同定を行う。

シーンの同定手法として、ICP アルゴリズムによるポイントクラウドの位置合わせが利用可能であるが、この手法を数万点からなる複数のポイントクラウドに対して同時に適用した場合、明らかにリアルタイムなシーン同定は実現できない。そこで、シーンに何らかの特徴量を見出し、低いコストで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、家電製品等の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合せのパターンによってシーンおよ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、取り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合せパターンの特徴として、次の 3 つを利用する。

観測可能な平面の数

撮影位置や姿勢によって変動する特徴量であり、シー

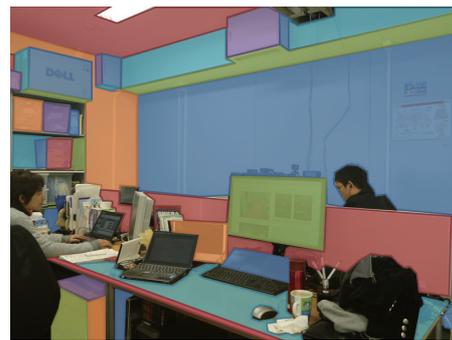


図 1 室内の平面オブジェクトの例

Fig. 1 Example of planar objects in a room.

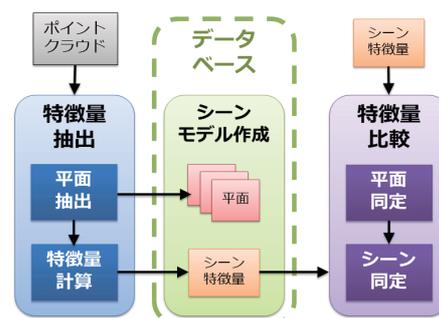


図 2 空間情報マッチング手法の概要

Fig. 2 Overview of the method of spatial information matching.

ン探索の際に候補を大まかに絞ることができる。

平面間のなす角度

撮影位置によらずに一定値をとる特徴量である。室内にある平面間の角度は 0° または 90° である場合が多い。

平行な平面間の距離

平面間の角度が 0° の場合に利用可能な、撮影位置によらずに一定値をとる特徴量である。

3.2 空間情報マッチング手法

本節では、シーン同定およ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。事前処理および平面抽出ステップでは Rusu [9] の手法を参考にしている。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ユークリッド距離の計算や行列演算をとまなう場合が多いため、コストが高い。そこで、本研究ではコストの低い 2 次元画像処理を可能な限り用いる。図 3 に示すように、距離画像から生成される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

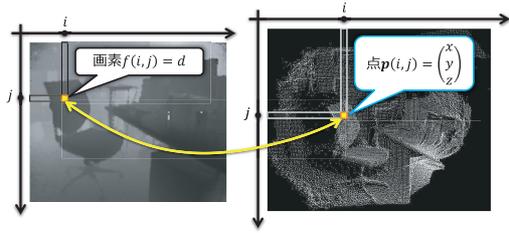


図 3 距離画像とポイントクラウドの対応関係

Fig. 3 Depth image and its corresponding point cloud.

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、距離画像から生成されるポイントクラウドの各点における 3 次元データ処理は 2 次元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向計算を行う。

平滑化は画像処理により実現する。具体的には、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 [10] 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て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。本研究では入力画素の周囲 1 [pixel] を利用するために $w = 1$ とし、経験的に $\sigma_1 = \sigma_2 = 1000$ 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。具体的には、平滑化したポイントクラウドの各点 $\mathbf{p}(i, j)$ について法線 $\mathbf{n}(i, j)$ を求める。まず $\mathbf{p}(i, j)$ の N 個の近傍点 $\mathbf{q}_k(i, j)$ ($k = 1, 2, \dots, N$) を探索し、これを集合 $\mathbf{X}(i, j)$ とする。次に $\mathbf{X}(i, j)$ 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 $\mathbf{v}_3(i, j)$ を求める。すなわち、 $\mathbf{X}(i, j)$ の重心点 $\boldsymbol{\mu}(i, j)$ と共分散行列 $\mathbf{C}(i, j)$ を式 (3), (4) により求め、 $\mathbf{C}(i, j)$ の 3 番目に大きい (最小の) 固有値に対応する固有ベクトル $\mathbf{v}_3(i, j)$ を求める。

$$\boldsymbol{\mu}(i, j) = \frac{1}{N} \sum_{k=1}^N \mathbf{q}_k(i, j) \quad (3)$$

$$\mathbf{C}(i, j) = \sum_{k=1}^N (\mathbf{q}_k(i, j) - \boldsymbol{\mu}(i, j))^T (\mathbf{q}_k(i, j) - \boldsymbol{\mu}(i, j)) \quad (4)$$

$\mathbf{v}_3(i, j)$ の向きは一意に定まらない。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。

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

処理の高速化のため、以上の各点における法線方向計算は並列化して行う。

本研究では $N = 20$ としている。Rusu [9] が指摘しているとおり、考慮する近傍点の数が少ない場合はノイズの影響が大きくなるため、全体として法線の推定精度が下がってしまう。しかし逆に近傍点の数が多き場合は、平面が交わる角や辺等の、急激に法線方向が変化する部分で推定精度が下がってしまう。この解決策として、徐々に N を増加させる試行を繰り返す方法があげられるが、計算コストが多くなってしまったため、本稿ではこの手法を採用せず、経験的に N を決定した。

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を形成する点を絞り、その後ポイントクラウドの処理により、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、これらの平均からな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 $r_i \in R$ ($i = 1, 2, \dots, n_{\text{rgn}}$) 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、 $N = 20$ の場合に天井や壁等の平面が正しく抽出できるように、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、ラベリングした領域群 R 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。以下では、処理手順の詳細について述べる。

まず初めに、3 次元空間 (x, y, z) における平面方程式 (式 (6)) を推定する (7 行目)。ただし、原点は距離画像センサ撮影時の視点とする。また、 α_P , β_P , γ_P , δ_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。

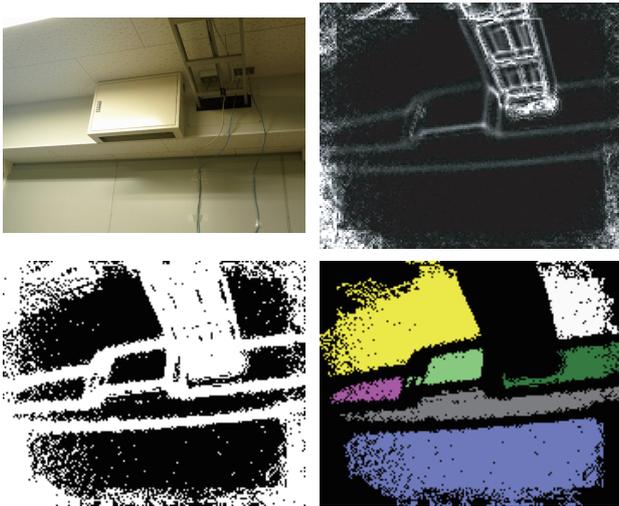


図 4 平面領域推定の流れ：対象環境（左上），曲率画像（右上），2値化画像（左下），ラベリング処理後（右下）

Fig. 4 Planar area estimation: Target environment (top left), curvature image (top right), binary image (bottom left) and labeled image (bottom right).

表 1 平面抽出アルゴリズム
Table 1 Algorithm of plane extraction.

Algorithm Plane_Extraction ($R, C, K, N_{\text{blk}}, N_{\text{inl.th}}$).	
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, δ_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_{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

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

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

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

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

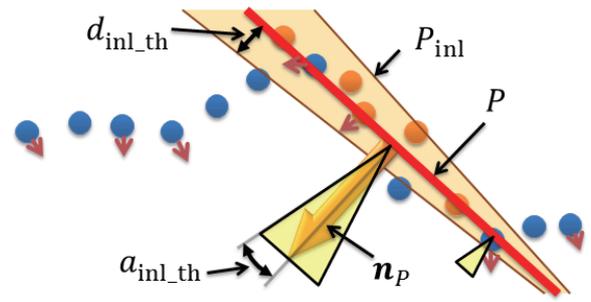


図 5 平面の含有点の探索の様子
Fig. 5 Search for points contained in a plane.

平面方程式推定手法の1つに、RANSAC法 [11] があげられる。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。これに対して、Dubeら [12] 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。たとえば、 N_{rgn} 個の点を含むポイントクラウドからkd木を用いて指定距離の点を探るコストは $O(\log N_{\text{rgn}})$ となる。そこで本研究では、空間的に隣接する点は強い相関を持つことに注目し、ラベリングされた各領域に内包される N_{blk} [pixel] 四方のブロックをランダムに選択する (6行目)。この手法の探索コストは $O(1)$ である。ここで、 N_{blk} は小さすぎるとノイズの影響が大きく平面方程式に反映されてしまう。逆に、 N_{blk} が大きすぎると、ブロックが領域に内包されなくなってしまふ。この点に留意し、本稿では予備実験を行った結果から $N_{\text{blk}} = 4$ とした。その後、選択した点集合の主成分分析により平面 P を推定する (7行目)。

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 (11-14行目)。図 5 にその様子を示す。橙色の点が条件を満たす点であることを表している。

以上の平面方程式推定と含有点探索を K 回繰り返す (5-20行目)。 N_{inl} が最も大きくなる平面方程式を正しい推定と見なす。このときの含有点を C から抽出する (21行目)。 N_{inl} が $N_{\text{inl.th}}$ 以下のとき、適切な平面ではないと見なす (3, 16行目)。

以上の操作をすべてのラベリング領域 r_i に対して実行し (2-23行目)、最終的に n 個の平面からなる集合 L を推定する (24行目)。閾値 $a_{\text{inl.th}}$, $d_{\text{inl.th}}$ は式 (10), (11) のように定義する。

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

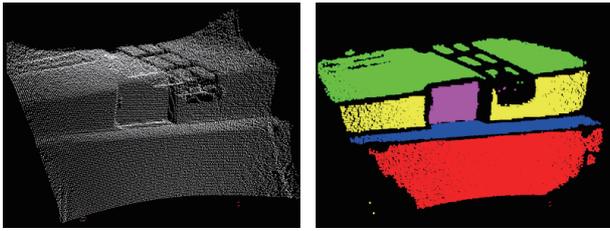


図 6 平面抽出の成功例：ポイントクラウド（左）と抽出平面（右）の様子

Fig. 6 Successful results of plane extraction: Point cloud (left) and extracted planes (right).

$$d_{\text{inl_th}} = \delta_{\text{inl_th}} p_z \quad (11)$$

ただし、 p_z は点 p の視点から見た奥行方向成分である。

本稿では経験的に $K = 100$, $N_{\text{inl_th}} = 200$, $\phi_{\text{inl_th}} = 10^\circ/m$, $\delta_{\text{inl_th}} = 0.02$ 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 節であげた 3 つの特徴量：

- 観測可能な平面の数 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。ただし、LUT は Look-up Table である。

平面 P_i, P_j 間の角度 $a_{i,j}$ ($i, j = 1, 2, \dots, n; i < j$) は平面の全組合せ nC_2 に対して計算し、式 (12) に示す上三角行列を作成する。

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

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ので、平面が平行の場合、2 つの距離の差は平面間距離と見なしてよい。そこで、平行平面間距離は式 (13) とする。

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

これらから、式 (14) に示す上三角行列を作成する。

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

本稿では、壁と家電機器や机と床等の間の角度を繰り返し計算する予備実験の結果から、平行平面間の角度が 15° 以下のとき平行と見なす。

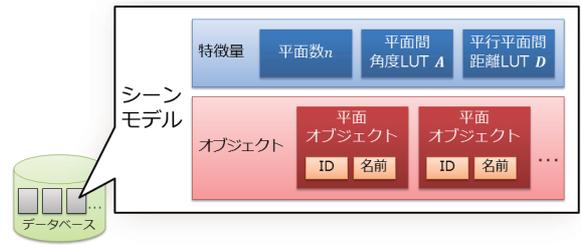


図 7 シーンモデル

Fig. 7 Scene model.

表 2 平面同定アルゴリズム (シーン特徴量比較アルゴリズム)

Table 2 Algorithm of plane identification (scene feature matching).

Algorithm Feature_Matching (m_i, m_{curr}).	
1:	$a_{\text{best}} := a_{\text{th}}, p_{\text{best}} := \text{null}$
2:	if neither m_i nor m_{curr} include any parallel planes do
3:	$d_{\text{best}} := 0$
4:	else
5:	$d_{\text{best}} := d_{\text{th}}$
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:	$a_{\text{max}} := \max \left\{ s_{u,v} \mid S = [s_{u,v}] = A_i - A_{\text{curr}}^{p_n^k} \right\}$
13:	$d_{\text{max}} := \max \left\{ t_{u,v} \mid T = [t_{u,v}] = D_i - D_{\text{curr}}^{p_n^k} \right\}$
14:	if $a_{\text{th}} > a_{\text{max}} \cap d_{\text{best}} \geq d_{\text{max}}$ do
15:	$p_{\text{best}} := p_n^k$
16:	$a_{\text{best}} := a_{\text{max}}$
17:	$d_{\text{best}} := d_{\text{max}}$
18:	end if
19:	end for
20:	return ($p_{\text{best}}, a_{\text{best}}, d_{\text{best}}$)

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、計算した特徴量とあわせて 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、左側、正面、右側の 3 つの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。この 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 $a_{\text{th}} = 15^\circ, d_{\text{th}} = 0.3m$ 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

表 3 シーン同定アルゴリズム

Table 3 Algorithm of scene identification.

Algorithm Scene_Identification (M, m_{curr}).	
1:	$d_{best} := d_{th}, m_{best} := null$
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 $a < a_{th} \cap d < d_{best}$ do
7:	$m_{best} := m_i$
8:	$d_{best} := d$
9:	end if
10:	end if
11:	end for
12:	return m_{best}

行平面間距離を無効にするために $d_{best} = 0$ とする (2-6 行目). k は順列のインデックス番号 ($k = 1, 2, \dots, n!$) である (10 行目). 行列 $A_{curr}^{p_n^k}$, $D_{curr}^{p_n^k}$ 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 (12, 13 行目). 返される a_{best} , d_{best} はマッチングスコアとして利用される (20 行目). このスコアが小さいほどシーンの一致度が高いことを表す. また, スコアが最小の場合に, 2つのシーンに含まれるすべての平面は同定されている.

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出し (6-9 行目), 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および平面同定の実験とその結果について述べる. 実験は Intel Xeon CPU E5-2609 2.40 GHz 2 プロセッサ (8 コア 8 スレッド) 搭載の Windows 7 上で行った. 距離画像センサは MesaImaging 社の SwissRanger SR4000 (5m レンジ, 広角レンズ版) を用いた.

4.1 実環境でのシーンおよび平面同定実験

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, 正面から新しく撮影した入力シーンと比較を行う. 入力シーンとしては, データベースに登録されている環境 4 つと登録されていない環境 1 つ (以下, “机 3”) の合計 5 つを用いる. これら 5 つの環境の様子を図 9 に示す. ただし, 本実験ではシーン中のすべてのオブジェクトが固定されていること, および平面オブジェクトは 600 cm^2 以上を露出していることを前提条件と



図 8 実験に用いた環境

Fig. 8 Experimental environments.

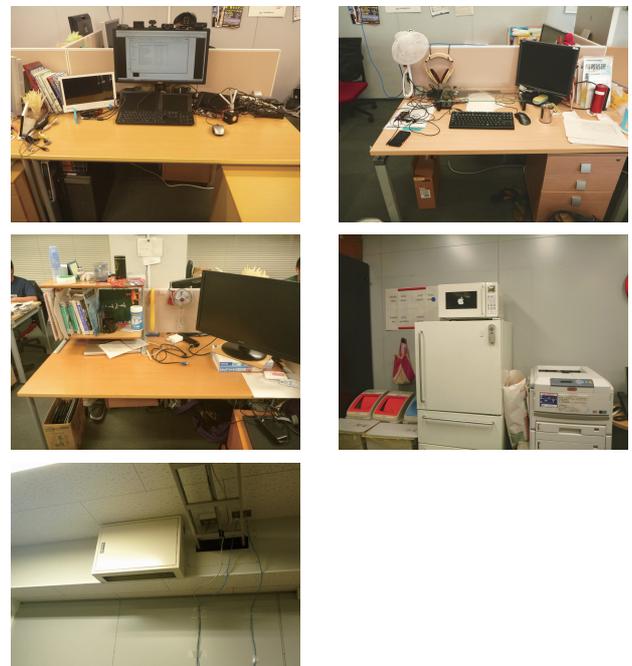


図 9 対象環境: 机 1 (左上), 机 2 (右上), 机 3 (左中央), 冷蔵庫 (右中央), 配電盤 (下)

Fig. 9 Target environments: Desk 1 (top left), Desk 2 (top right), Desk 3 (middle left), Fridge (middle right) and Switchboard (bottom).

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 個作成の平均所要時間は 765 [ms] であった. 入力シーン 1 個とシーンモデル 12 個の比較の平均所要時間は 4 [ms] であった.

表 4 に同定結果を示す. 丸括弧内は平面数を表し, 空白欄は平面数の違いにより不一致と見なされた組合せを表す. 表中の \times は平面間角度およ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合せを表し, \circ は一致と見なされた組合せを表す. この表から, 4 つの環

表 4 シーンおよび平面オブジェクト同定結果：行は登録シーン，列は入力シーン

Table 4 Results of scene identification and plane identification: Learned scenes in the rows and input scenes in the columns.

	入力 シーン (平面数)	学習済み				未登録 机 3 (4)
		机 1 (5)	机 2 (5)	冷蔵庫 (3)	配電盤 (5)	
机 1	左 (5)	○ 1	×		×	
	正面 (5)	○ 2	×		×	
	右 (4)					×
机 2	左 (7)					
	正面 (5)	×	○ 3		×	
	右 (6)					
冷蔵庫	左 (3)			○ 4		
	正面 (2)					
	右 (6)					
配電盤	左 (5)	×	×		○ 5	
	正面 (5)	×	×		○ 6	
	右 (7)					
推定結果		机 1	机 2	冷蔵庫	配電盤	なし

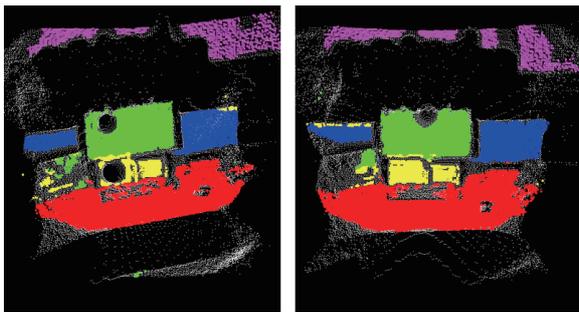


図 10 シーン同定および平面同定の成功例 (机 1)：入力シーン (左) とデータベース登録シーン (右)．同じ色は同じ平面であることを表す

Fig. 10 Successful results of scene identification and plane identification (Desk 1): Input scene (left) and its corresponding learned scene (right). Planes in same colors correspond to each other.

境すべてが正しく推定されていることが分かる．成功例を図 10 に示す．また，登録されていない“机 3”は該当なしと推定された．しかし，図 11 に示すように，○₅ および ○₆ では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 枚の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いたため，平行な平面が 1 組しか存在しなかった今回の場合に，その位置関係の特定まではできなかったと考えられる．平行でない平面間についても距離を計算し，これを評価対象とすることで改善されると思われる．

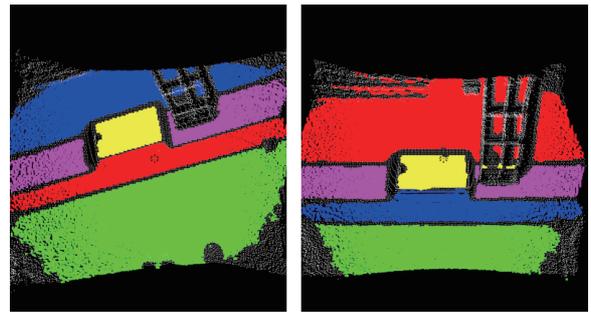


図 11 平面同定の失敗例 (配電盤)：入力シーン (左) とデータベース登録シーン (右)．青と赤の平面が誤って同定されている

Fig. 11 Failed results of plane identification (Switchboard): Input scene (left) and its corresponding learned scene (right). The blue and red planes need to be swapped.



図 12 電力情報可視化システム動作画面：認識前 (左)，認識後の AR 表示 (右)

Fig. 12 Example of power visualization system: Before recognition (left) and after (right).

現状では，比較するシーンモデル数に応じて処理時間は線形に増加してしまう．これはすべてのシーンに対して平面同定アルゴリズムが適用されているからである．平面同定に先立ってシーン特徴量の一部を比較する等の，コストの低い処理の適用によりあらかじめシーンの候補を限定すれば，処理時間の線形増加を回避できると思われる．

また，シーンモデル数の限定には他の位置推定技術等の併用も有効である．たとえば，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 電波強度を利用した位置推定技術の利用が考えられる．

5. AR システムの実装

本手法を用いた AR システムを実装した．図 12 に AR システムの動作画面例を示す．本 AR システムは家電機器や配電盤を認識し，それに対応した電力グラフ (図 13) や接続機器一覧等の情報を表示する電力情報可視化システムである．本 AR システムの動作フローを図 14 に示す．起

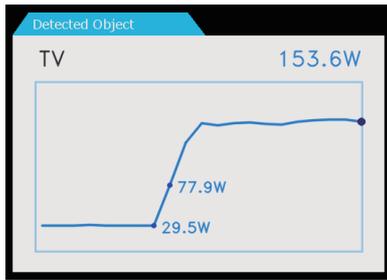


図 13 電力グラフ
Fig. 13 Power graph.



図 15 AR スコープ：表側 (上), 裏側 (中央), 動作例 (下)
Fig. 15 AR Scope: Front (top), back (middle) and example of system (bottom).

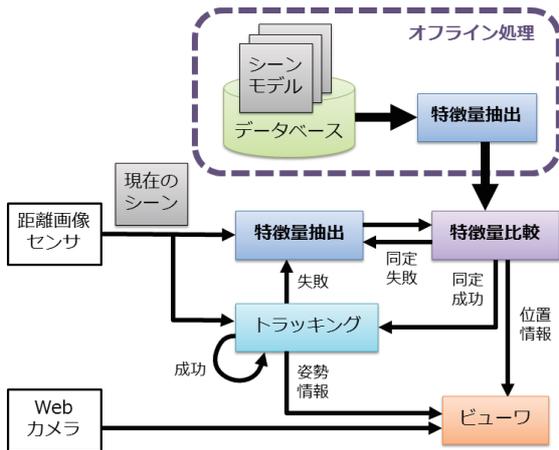


図 14 AR システムの動作フロー
Fig. 14 Workflow of AR system.

表 5 デバイス一覧
Table 5 List of devices.

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

動後は特徴量抽出および特徴量比較アルゴリズムを繰り返して、登録されているシーンを探索する。データベース内のあるシーン情報とのマッチングが成功した場合は探索を中断し、登録された情報に基づいて適切な情報を AR 表示する。その後は、OpenCV を用いて距離画像のオプティカルフローからトラッキング処理を行い、AR イメージの位置や姿勢をリアルタイムに推定する。AR の対象となるオブジェクトが画面外に移動したとき、AR イメージの表示およびトラッキング処理を中断し、再び探索を行う。

本 AR システムのために、図 15 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、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バスト性の向上をめざし、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。本手法では、まず環境光の影響を低減するために、距離画像センサを用いて室内の平面オブジェクトを抽出する。次に、得られた平面の組合せのパターンの特徴量として、平面の数、平面間の角度、平行平面間の距離の 3 つを計算する。最後に、特徴量比較により

事前学習したシーンと現在撮影しているシーンのマッチングを行う。提案手法の評価実験ではいく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あげられる。

平面以外の形状や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、行列計算をとまな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。

閾値の設定方法の検討

本稿では多くのパラメータを経験的に求めた。使用するセンサの性能にあった最適なパラメータの計算方法を今後検討したい。

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] 大淵徹之, 岡村和男, 木村真人, 多々美滋: 旅行用ナビゲーション機器の開発と観光での利用事例, 情報処理学会デジタルプラクティス, Vol.3, No.4, pp.246-253 (2012).
- [3] Kato, H., Billingham, M., Popyrev, I., Imamoto, K. and Tachibana, K.: Virtual Object Manipulation on a Table-Top AR Environment, *Proc. International Symposium on Augmented Reality (ISAR)*, pp.111-119 (2000).
- [4] Klein, G. and Murray, D.: Parallel Tracking and Mapping for Small AR Workspaces, *Proc. International Symposium on Mixed and Augmented Reality (ISMAR)*, pp.225-234 (2007).
- [5] Castle, R., Klein, G. and Murray, D.: Video-rate Localization in Multiple Maps for Wearable Augmented Reality, *Proc. International Symposium on Wearable Computers (ISWC)*, pp.15-22 (2008).
- [6] Newcombe, R.A., Lovegrove, S.J. and Davison, A.J.: DTAM: Dense Tracking and Mapping in Real-Time, *Proc. International Conference on Computer Vision (ICCV)*, pp.2320-2327 (2011).
- [7] Izadi, S., Kim, D., Hilliges, O., Molyneaux, D., Newcombe, R.A., Kohli, P., Shotton, S., Hodges, J., Freeman, D., Davison, A.J. and Fitzgibbon, A.: KinectFusion: Real-time 3D Reconstruction and Interaction Using a Moving Depth Camera, *Symposium on User Interface Software and Technology (UIST)*, pp.559-568 (2011).
- [8] Besl, P.J. and McKay, N.D.: A Method for Registration of 3-D Shapes, *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)*, Vol.14, pp.239-256 (1992).
- [9] Rusu, R.B.: Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments, *Ph.D. Thesis, Institut für Informatik der Technischen Universität München, Germany*, (2010).
- [10] Tomasi, C. and Manduchi, R.: Bilateral Filtering for Gray and Color Images, *Proc. International Conference*

on Computer Vision (ICCV), pp.839-846 (1998).

- [11] Fischler, M.A. and Bolles, R.C.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography, *Comm. ACM*, Vol.24, pp.381-395 (1981).
- [12] Dube, D. and Zell, A.: Real-Time Plane Extraction from Depth Images with the Randomized Hough Transform, *IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pp.1084-1091 (2011).



山川 健司 (学生会員)

2013年名古屋大学工学部電気電子・情報工学科卒業。同年同大学大学院修士課程入学。ユビキタスコンピューティングに関する研究に従事。



梶 克彦 (正会員)

2002年名古屋大学工学部電気電子工学科卒業。2007年同大学大学院情報科学研究科博士課程修了。博士(情報科学)。同年NTTコミュニケーション科学基礎研究所リサーチアソシエイトを経て、2010年より名古屋大学大学院工学研究科助教。日本ソフトウェア科学会会員。屋内位置推定、遠隔コミュニケーション支援の研究に従事。



河口 信夫 (正会員)

1990年名古屋大学工学部電気電子工学科卒業。1995年同大学大学院工学研究科情報工学専攻博士課程満了。同年同大学工学部助手。同大学講師、准教授を経て、2009年より同大学大学院工学研究科教授。NPO 位置情報サービス研究機構 Lisra 代表理事。モバイルコミュニケーション、ユビキタスコンピューティング、行動センシングの研究に従事。博士(工学)。ACM, IEEE, 人工知能学会, 日本ソフトウェア科学会, 電子情報通信学会, 日本音響学会各会員。