RGB-D カメラを用いた テクスチャの特徴点と3次元点群との対応付けによる リアルタイム3次元形状復元

KinectFusion Aided by Matching feature points In Color Images

鈴木 遼平, 徐 剛
Ryohei Suzuki and Gang Xu
立命館大学大学院 情報理工学研究科, 草津市野路東 1-1-1
Ritsumeikan University, 1-1-1 Nojihigashi, Kusatu, Shiga

あらまし: 本稿では, 3次元点群とテクスチャ情報を用いた3次元形状モデルの生成と自己位置姿勢推定手法を提案する. 距離センサより得られる3次元点群を用いた手法では, 壁や床のような平面の場所ではセンサの位置姿勢を求めることが困難である. また, 同形状の物体や回転体といった形状が変わらない物体などに対しても位置姿勢の推定が困難となる. そこで本提案手法では, テクスチャ情報の特徴点のフレーム間での対応付け, 同フレームにおける特徴点と3次元点群の対応付け, フレーム間の特徴点マッチングの結果を利用し, 3次元点群の位置合わせを行う. これにより, 2次元特徴または3次元特徴のどちらか一方がある場所でのセンサの位置姿勢を求めることができる.

Summary: As robots develop, importance of autonomous mobile robots in various fields increases. This paper proposes a new approach to real-time pose estimation and dense 3D reconstruction using both color images and 3D point clouds obtained by Kinect. At places with no 3D features such as walls or floors, it is difficult to estimate sensor poses by using only 3D point clouds. However, even in places with no 3D features, there are still 2D features in many cases which can be detected from color images. To aid point cloud fusion, we add feature point matching between two consecutive frames. 3D coordinates can be obtained from the depth image for these matched feature points. The 3D coordinates can then be used to align 3D point clouds. Thus, we can estimate sensor poses at places with either 3D features or 2D features. This new function is added to KinectFusion.

キーワード: 3次元形状復元, 自己位置姿勢推定, 特徴点, SLAM
Keywords: 3D reconstruction, self-localization, feature points, SLAM

1. はじめに

1.1. 研究背景

近年の社会問題となっている少子高齢化により近い将来、老人の介護や施設内での作業を行う人手の減少が予想される. また、福島原発の事故現場のような人間の活動が制限された場所での作業の必要性が高まってきている. これにより、現在の周辺環境を把握し, 自律移動を行うロボットの期待が高まっている. このような自律移動ロボットの実用化に向けた課題の1つに, 身の回りの物体や置物などの環境と自分の位置を認識するといったものがある. これは周辺環境の3次元モデルを作成し、さらに自分の位置姿勢を推定するSLAM(Simultaneous Localization and Mapping)[1]問題と置き換えることが出来る. SLAMの手法には入力情報にテクスチャ情報を用いるものと, 距離情報を用いるものとが挙げられる. ここで3次元モデルの作成のためには, 3次元計測が必要である. 3次元計測を行う方法として, カメラを用いる方法やレーザーレンジセンサなどの距離センサを用いる方法がある. カメラによる測定はステレオ法[2]やSfM(Structure from Motion)[3]が挙げられる. これらは複数枚のRGB画像からそのシーンの3次元形状復元

を行う. この測定は比較的手軽に行えるが, 3 次元形状推定のため計測精度があまり高くない. さらに, スケールが決まらないというスケールの不定性という問題も挙げられる. 一方, 距離センサによる測定は高速かつ正確であるが, カメラと比較すると高価である. しかし, 近年 Kinect[4]の登場により比較的安価に距離センサの入手が可能となった. Kinect は Microsoft 社製の RGB カメラや距離センサ, マルチアレイマイクロフォンなどの様々なセンサを持ち, それぞれのセンサを同時に使用することが可能なデバイスである. テクスチャ情報(図 1)と距離情報(図 2)の両方の取得が可能である. カメラによる計測手法の自然特徴点のみの 3 次元形状復元のため, 結果が疎になりやすく, 密な 3 次元形状モデルの生成は困難である. 今回の 3 次元計測には距離センサを用いることとする.

距離センサを用いて物体の 3 次元形状復元を行う. このとき距離センサで取得できる距離情報は 1 方向からの情報, センサが向いている部分のみのため, 図 3 のように側面や反対側の形状は取得できない. このため, 物体の周囲を抜けがないように複数回計測し, 複数計測データの統合を行う必要がある(図 5). 計測データの統合を行う場合, 図 4 のように計測データ間の対応を求め, 位置合わせを行う. 位置合わせとは共通部分の対応により, 計測データ間の回転と並進の変換行列を求めることである. つまり位置合わせは距離センサ間の相対的な幾何関係, 自己位置姿勢が求めることである. これらの位置合わせや統合手法はいくつか存在する[2]. しかし, 計測データは 3 次元点群のため, 3 次元における幾何関係を求める必要があり, 重複領域の計測誤差や形状をどのようにするかという問題がある. また多くの計測データの位置合わせを行う場合には, 累積誤差が発生する. 累積誤差とは, 2 つの計測データ間では無視出来るような誤差でも, 全体となると累積し大きな誤差となる. そのため 2 つのデータ間計測データ全体で累積誤差が最小になるように最適化する必要がある. このように多くの処理を 3 次元形状復元では行うため, メモリ及び計算に膨大なコストがかかる. そのためリアルタイムに位置合わせや自己位置姿勢推定を行う場合には GPU (Graphics Processing units)を用いた並列処理が必要となる. また, 平面や球体のように 3 次元形状特徴が少ない環境では, 位置姿勢推定が困難という問題が存在する.

図 1 テクスチャ情報(RGB 画像)

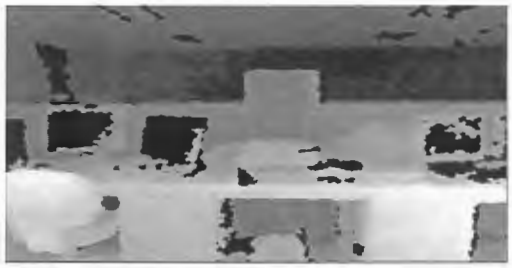

図 2 距離情報(深度画像)

図 3 Kinect が取得する
1 フレームの 3 次元点群

図 4 3 次元点群同士の対応付け

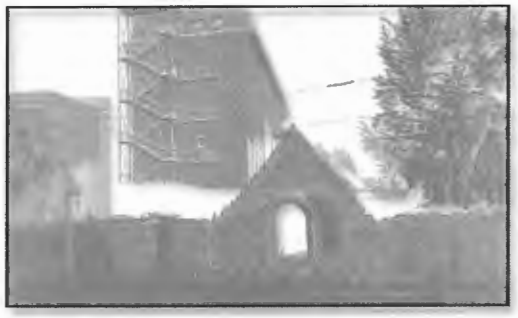

図 5 3 次元点群の位置合わせ

1.2. KinectFusion

距離センサを用いた手法に KinctFusion[5] がある. Kinect から取得出来る距離情報(3 次元情報)を用いてリアルタイムに自己位置姿勢推定と図 4 のような実環境の密な 3 次元形状復元を行う. 出力結果は 3 次元点群だけでなく, メッシュ化したモデルも出力可能となっている. 通常, 3 次元形状復元は計算コストが高く, リアルタイムでの処理は困難となっている. しかしながら, Coarse-to -Fine や GPGPU を用いることでリアルタイム処理を実現している. Frame-to-Model(フレームとモデル間の)位置合わせを用いることで, 累積誤差が少なく, 安定して位置姿勢の推定が可能となっている. また Truncated Signed Distance Function(TSDF)を用いることで, モデルの表面推定や外れ値の除去を行っている(図 5). しかし, 距離情報を用いているため, 壁や床のような平面や球体のように回転しても形状の変わらないものなどを対象とした場合, 位置姿勢推定が困難になる問題がある.

図 6 KinectFusion の実行結果
左：取得した 3 次元点群
右：生成された 3 次元モデル

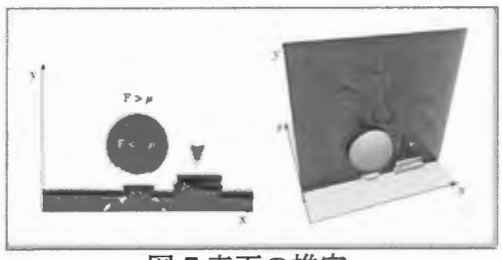

図 7 表面の推定

1.3. 問題定義

テクスチャ情報を用いた手法では, 同形状の物体でも, 色や模様といったテクスチャによる判別が可能である. しかし, 3 次元モデルのスケールが定まらないスケールの不定性や照明変化に弱いという問題が挙げられる. 一方, 距離情報を用いた手法では, 完全な 3 次元形状情報の取得が可能であり, 照明変化にも強いという利点がある. しかし, 平面や球体などの 3 次元特徴が少ない場合, 自己位置姿勢推定が困難であることや, 同形状や回転体の場合に誤対応が起きるという問題が存在する. 本研究では, テクスチャ情報か 3 次元形状のどちらかに特徴が存在する場所での自己位置姿勢推定を行うことを目指す.

1.4. 研究目的

本研究では, テクスチャ情報と 3 次元情報の両方を用いて, 自己位置姿勢推定及び 3 次元形状復元をリアルタイムで行うことを目的とする, テクスチャ情報と距離情報の両方を用いることで, 一方の問題を他方の情報で補う. 今回, RGB-D カメラからテクスチャ情報と 3 次元情報の両方を同時に取得する. 1 つのピクセルが RGB-D, 4 つの情報を保持している(図 8). 今回はテクスチャ情報(RGB 画像)より特徴点の抽出を行う. フレーム間で特徴点のマッチングを行う. その後, 特徴点マッチングによる対応を用いた3次元点群同士の位置合わせを行うことで自己位置姿勢の推定を行う. また, 対応点が少ない場合は3次元点群のみを利用し, ICP アルゴリズムを用いて, 位置合わせを自己位置姿勢推定を行う. 得られた結果を用いて, 世界座標系へと点群の統合やモデルの表面の推定を行う. 今回, 3 次元点群同士の位置合わせや表面推定の処理は KinectFusion を利用する. そのため, テクスチャ情報と 3 次元点群を用いた位置合わせ処理を KinectFusion に追加することで実現する.

本稿では, 2 節で特徴点の処理について, 3 節で 3 次元点群の位置合わせ, 4 節で実験結果, 5 節で結論及び考察を述べる.

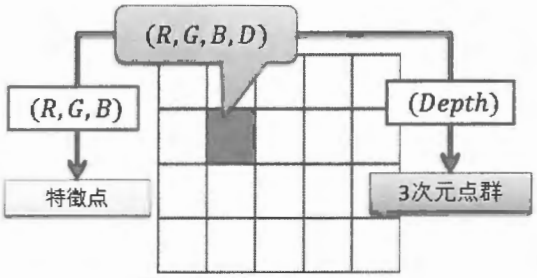

図 8 RGB-D 画像

2. 特徴点

テクスチャ情報である画像を用いて，自己位置姿勢を求める場合には画像同士の対応する特徴点の対応を求める必要がある．1999 年に Lowe らによって SIFT が提案された．SIFT は回転不変性，スケール不変性を持ちなおかつ高精度にマッチング可能である．しかし，SIFT は処理コストが高くリアルタイムなマッチング困難であった．そこで積分画像を利用し BOX フィルタを用いて高速化した SURF が提案された．近年，より省メモリかつ高速なキーポイント抽出や特徴量の記述が求められ，高速なキーポイントの抽出手法として FAST が提案された．また，高速なマッチングを行うためにバイナリコードを特徴量記述に利用した BRIEF[6]も近年提案された．しかし，BRIEF には回転不変性がないという問題点がある．

本研究では，リアルタイムでの処理を目的としているため，高速に抽出が可能であることと特徴量によるマッチングが可能であり，スケールや回転に対しロバストな手法が望ましい．そこで，高速に特徴点抽出や特徴量マッチングを行うことが可能な ORB(Oriented FAST and Rotated BRIEF)[7]を用いて特徴点の処理を行う．ORB は以下のような特徴を持つ．

・特徴点抽出が高速
・スケールに対しロバスト
・回転不変性
・特徴量をバイナリコードにより記述
・ハミング距離による高速なマッチング

以下の項から ORB について説明していく．

2.1. キーポイント検出手法

ORB のキーポイント検出手法には，高速に検出可能な FAST を用いる．FAST について簡潔に説明する．FAST は，パッチ中心の輝度値と周囲の画素の輝度値を比較し，明るいまたは暗い画素が一定以上続く場合はコーナーやエッジとして検出する．周囲 16 ピクセルの連続性を確認するために，上下左右 4 点の輝度値を調査することで，コーナーかどうかの判定を行う．更に高速な検出を行うために，学習した 3 分木によってコーナーの検出をする．木構造の検出器のため，高速に検出が可能となる．しかし，FAST はオリエンテーションの算出が出来ないため，回転に弱いという問題がある．回転に対してロバスト性を得るためには，オリエンテーションを算出する必要がある．オリエンテーションを基準にすることで，入力画像が回転をしていても，同一な特徴量として検出することができる．そこで ORB では，回転不変性を得るために Oriented FAST を提案している．オリエンテーションの算出にはパッチの輝度の重心とキーポイントの中心のベクトルを用いる方法である．モーメント算出を以下に示す．

$$m_{pq} = \sum_{x,y} x^p y^q I(x,y) \quad (1)$$

ここで，パッチ内の座標を(x, y)，p, qは各軸方向のモーメントを求めるための値であり，0 または 1 を取る．式(1)を用いて式(2)によりパッチの重心位置を算出する．

$$C = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}}\right) \quad (2)$$

算出した重心位置とパッチ中心との方向ベクトルをオリエンテーション方向とする．パッチの中心をO，パッチの重心をCとするとオリエンテーションは$\overrightarrow{OC}$ となる．そして，オリエンテーションが作る角度は以下の式(3)で求められる．

$$\theta = \text{atan2}(m_{01}, m_{10}), \quad (3)$$

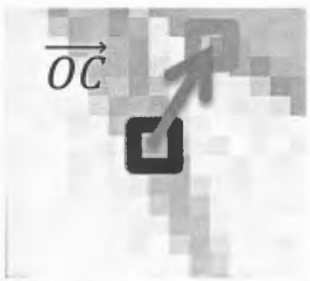

図 9 オリエンテーション方向

2.2. 特徴量記述

バイナリコードによってキーポイントの特徴量記述を

行う手法として，Binary Robust Independent Elementary Features (BRIEF)が提案されている．従来の SIFT や SURF では高次元の実数を用いていた．しかし，高次元の実数を使用した場合，メモリ容量の増加やマッチングの計算コストの増大などの問題がある．そこで，バイナリコードによって特徴量記述を行うことで，省メモリ化を図り，類似度計算にハミング距離を用いることで高速化を実現した．BRIEF について説明する．ノイズに対するロバスト性を得るために，パッチにスムージングをかけておく．パッチ内において選択された 2 点の輝度値の符号によりバイナリコードを作成する．以下に式を示す．

$$\tau(\mathbf{p};\mathbf{x},\mathbf{y}) = \begin{cases} 1 : \mathbf{p}(\mathbf{x}) < \mathbf{p}(\mathbf{y}) \\ 0 : \mathbf{p}(\mathbf{x}) \geq \mathbf{p}(\mathbf{y}) \end{cases} \quad (4)$$

バイナリセットを τ，スムージングをかけたパッチを $\mathbf{p}$ とする．選択された 2 点の座標をそれぞれ$\mathbf{x},\mathbf{y}$ とし，$\mathbf{p}(\mathbf{x})$ はある $\mathbf{x}$ 座標における輝度値を示す．特徴量は以下の式を用いて，n 列のバイナリコードとする．

$$f_n(\mathbf{p}) := \sum_{i=1}^{n} 2^{i-1}\tau(\mathbf{p};\mathbf{x}_i,\mathbf{y}_i) \ , \quad (5)$$

バイナリコードの長さは$n = 256$ している．なお，選択するピクセルは，キーポイント位置を中心としたガウス分布により選択している．また，ノイズに対するロバスト性を得るために，パッチにスムージングをかけておく．

ORB では，高精度にマッチングをするために学習を用いてピクセルを選択している．選択するピクセル位置は，ペアのビット分散が大きく，なおかつ N 組みのペアの相関が低いときに特徴量記述能力が高いバイナリとして特徴記述に使用する．また，バイナリコードのパターン例を以下の図 10 で示す．相関の少ないペアの例である．ORB では，Greedy アルゴリズムを用いて，学習し選択を行う．

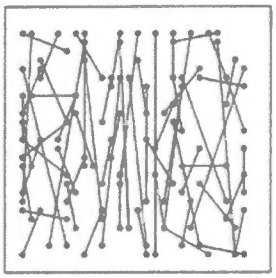

図 10 学習したパターン例

2.3. 特徴点マッチング

前述の特徴量を持った特徴点同士のマッチングを行う．各特徴点はバイナリコードで記述された特徴量を保持している．ある時刻Mにおける特徴点のセットをF_Mとする．F_MとF_{M-1}のフレーム間でのマッチングを行う．このとき，バイナリコード同士を排他的論理和(XOR)で演算することができ，ハミング距離での高速な評価が可能となる．図 11 に例を示す．本手法では，特徴点マッチングの結果を位置合わせの処理で利用する．

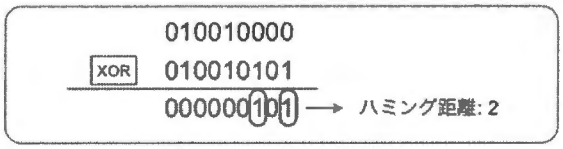

図 11 ハミング距離

3. 位置合わせと3次元統合

3次元モデルを生成するには，計測データの統合が不可欠となる．各カメラ座標系の計測データを同一座標系，つまり世界座標系へ変換することが必要となる．各カメラ座標系から世界座標系への変換には，自己位置を示す並進ベクトル $\boldsymbol{t}$，姿勢を表す回転行列 $\mathbf{R}$ を用いる．

本手法では，2 種類の手法を用いて，自己位置姿勢である回転行列と並進ベクトルの算出を行う．

3.1. 特徴点対応を利用した位置合わせ

ここでは前述で求めた特徴点対応を利用し，3 次元点群同士の位置合わせについて述べる．壁や床のような 3 次元特徴がない場所でも，その場所に絵や模様のような 2 次元特徴が存在している場合に対応できるようにするためである．また，特徴点マッチングを行

った結果，特徴点対応が 3 点以上ある場合にこの処理を行う．図 12 で示すように 3 つの対応を用いて，位置合わせする．1 つ目の対応は RGB-D 対応である．今回 Kinect を用いているため，RGB カメラと Depth カメラの幾何対応が既知となっている．そのため，同時刻のフレームMおいて，RGB-D のデータが得られる．これにより，特徴点とそれに対応する 3 次元点の登録することができる．2 つ目の対応は，ある時刻Mと $M-1$ のフレーム間における特徴点の対応である．2 節での処理により，対応がわかる．F_Mのi 番目の特徴点$F_M(i)$が持つ 3 次元点をX_{iM}とする．ここで$F_M(i)$に対応するF_{M-1}上の特徴点を$F_{M-1}(i)$とすると，$F_{M-1}(i)$が保持している 3 次元点X_{iM-1}とX_{iM}も対応していると考えられる．この 3 次元点同士の対応が 3 つ目の対応である．これらを用いて，以下の評価式で変換行列を求める．

$$E = \sum_{i=1}^{n} \|\mathbf{X}_{iM-1} - (\mathbf{R}\mathbf{X}_{iM} + \boldsymbol{t})\|^2 \quad (6)$$

評価式 E を最小化することで，回転行列と並進ベクトルを得る．また，位置合わせの際に，RANSAC を用いることで，ロバストに特徴点の誤対応を除去している．

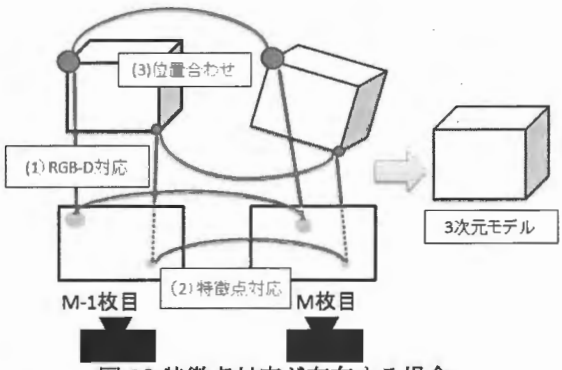

図 12 特徴点対応が存在する場合

3.2. 3 次元点群のみの位置合わせ

ここでは，特徴点対応が 3 点未満の場合の位置合わせについて述べる．図 13 のように特徴点の対応がない場合，3 次元点群同士のみの位置合わせにより．回転行列と並進ベクトルを求める．特徴点の対応が 3 点未満しか存在しない場合，センサの位置姿勢を決定するのに，特徴点の対応だけでは十分な拘束を得られることが出来ない．そのため，位置姿勢の推定には，オリジナルの KinectFusion を使用する．しかし，2 点の対応が存在している場合，特徴点の対応を利用して，並進ベクトルが定まる．そのため，拘束条件を用いて回転行列の算出ができる．

KinectFusion では，フレーム-モデル間で ICP アルゴリズムを用いて，位置合わせを行っている．フレーム-モデル間での位置合わせのため，累積誤差を考えずに安定して 3 次元モデルの生成が可能となっている．このとき，画像ピラミッドをモデル，フレームでそれぞれ作成する．そして，画像ピラミッドを用いて，Coarse-to-Fine の考え方でセンサの位置姿勢の推定を行っている．また，画像ピラミッドの各レベルにて，距離関数を使用している．距離関数にモデルの点や表面からどれくらいの距離があるのかを登録しておくものである．表面を 0 とする．これにより計算コストの削減や，外れ値の除去を行っている．

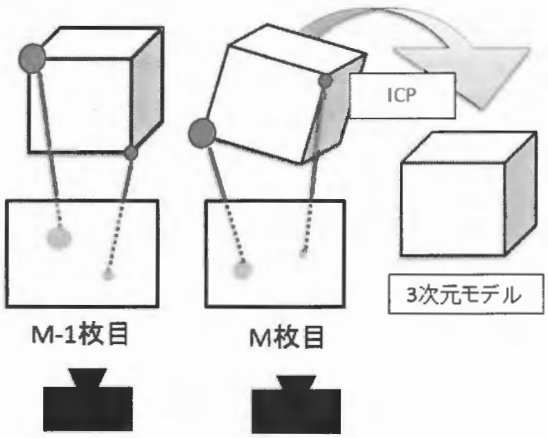

図 13 対応が 3 点未満の場合

3.3. 3 次元統合

この節では，前節で求めた変換行列を用いて，逐次的にカメラ座標系の 3 次元点群を世界座標系へと追加し，3 次元形状復元を行う．ここでの処理はオリジナルの KincetFusion の処理を利用する．これから KinectFusion の処理の簡潔に説明する．KinectFusion では，3 次元形状復元を行う際に，Truncated Signed Distance Function(TSDF)，切り捨て符号付き距離関数F，を用いて，復元されるモデルの表面推定を行う．各時刻の距離関数からも最も 0 に近い面，ゼロ交差を表面として推定する．このゼロ交差を求める際に，$F > \mu$を空間，$F < -\mu$ を物体の内部として表現している．$-\mu \leq F \leq \mu$の時に表面としている．これにより，最もそれらしい表面の推定が可能となっている．

4. 実験

手法の有効性を示すために RGB-D カメラを用いて実環境の 3 次元形状復元実験について述べる.

4.1. 実験概要

本手法の有効性を検証するために, RGB-D カメラを用いて, 実環境での 3 次元形状復元と自己位置姿勢推定を行う. 実験には, 平面での自己位置姿勢推定の困難の解決を示すため, ホワイトボードを対象とした実験を行う.

ここではホワイトボードに任意の文字や模様を描き, 2 次元特徴であるテクスチャ情報を取得できるような環境で行い, 3 次元形状復元ができているか定性的に評価する.

4.2. 実験環境

本研究での開発・実験環境を表 1 に示す. 実験で使用する RGB-D カメラは Microsoft 社の Kinect を用いる(図 14). 先にも述べたが, Kinect には IRONING プロジェクタ, IR カメラ, RGB カメラ搭載されており, 640×480 ピクセルでテクスチャ情報と距離情報を 30fpsで取得することが出来る. また, IR カメラと RGB カメラ間でのキャリブレーションができているため, 同じ位置姿勢からテクスチャ情報, 距離情報を取得することが可能である.

表 1 開発・実験環境

CPU	Intel Core i7 3.9GHz
メモリ	16GB
OS	Windows7(64bit)
開発環境	Visual Studio 2010
使用ライブラリ	OpenCV/PCL

図 14 Kinect

4.3. ホワイトボードを用いた実験

実験に用いる対象のホワイトボードを図 15 に示す. 対象の周りを中心に撮影を行い, 3 次元形状復元実験を行う. RANSAC を行い, 誤対応を除去したフレーム間の特徴点マッチングの結果を図 16 に示す. 提案手法によるホワイトボードの復元結果を図 17 に示す. 図 18 には, KinectFusion による復元結果を示す. 特徴点実験の結果を表 2, 3 にまとめる. 実験結果より, 平面において従来手法よりも 3 次元形状復元されていることがわかる.

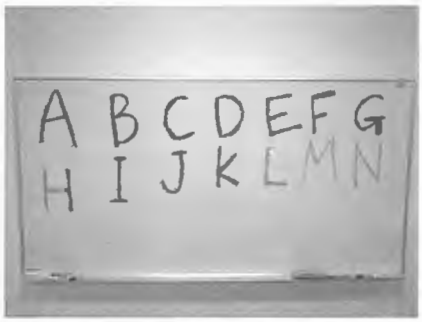

図 15 テクスチャを持つホワイトボード

図 16 特徴点マッチングの結果画像

表 2 特徴点の実験結果

	点数
特徴点(m=100)	467
全対応点	458
対応点(RANSAC 後)	349
誤対応(RANSAC 後)	2
平均抽出点数	567 / f

表 3 特徴点関連の処理速度

抽出・マッチング (平均)	22.3(ms)
抽出・マッチング(最大)	26.2(ms)
位置合わせ(平均)	35.1(ms)
位置合わせ(最大)	38.4(ms)

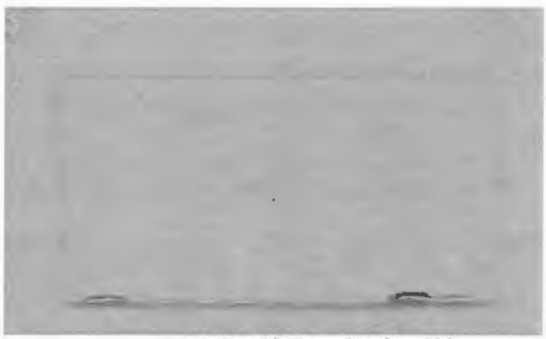

図 17 平面の復元結果 1 (提案手法)

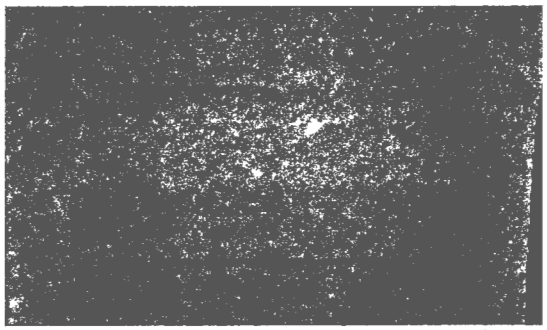

図 18　平面の復元結果 2（KinectFusion）

5.　まとめ

本稿では，テクスチャ情報，距離情報の両方を用いて 3 次元特徴の少ない場所において，センサの位置姿勢を推定し，3 次元形状復元を行う手法を提案した．テクスチャ情報において，局所特徴量の ORB を抽出された特徴点のマッチングに使用した．また，特徴点と対応した Kinect から同時に得られた 3 次元点を位置合わせに用いた．また，位置合わせの際に RANSAC を使用することで，特徴点同士の誤対応も除去することに成功した．これにより，オリジナルの KinectFusion において 3 次元特徴の少ない，平面や球体での位置姿勢推定で有効であると考えられる．2 次元特徴または 3 次元特徴のどちらか一方が存在する場所での SLAM を実現した．ただ，今後の展望としては，評価対象を増やし，定量的な評価や高速化を行っていく．

謝辞

本研究の一部は，私立大学戦略的研究基盤形成支援事業と芸術・文化分野の資料デジタル化と活用を軸とした研究資源共有化研究の支援を受けた．

参考文献

[1] B. Williams, G. Klein, and I. Reid, "Real-Time SLAM relocalisation," *In Proc.11th IEEE International Conference on Computer Vision (ICCV'07)*, pp.1-8, Rio de Janeiro, October 2007.

[2] Gang Xu, Zhengyou Zhang, "Epipolar Geometry in Stereo, Motion and Object Recognition A Unified Approach," Kluwer Academic Publishers 1996.

[3] Akihito Torii, Yuuki Hanzawa, Masatoshi Okutomi, "Easy-to-Use Online SfM System," *SSII2011*, pp.1-6, June 2011.

[4] Microsoft, "Kinect, " http://www.xbox.com/ja-JP/kinect

[5] Richard A. Newcombe, Shahram Izadi, Otmar Hilliges , David Molyneaux , David Kim , Andrew J. Davison , Pushmeet Kohli , Jamie Shotton, Steve Hodges, Andrew Fitzgibbon, "KinectFusion: Real-time dense surface mapping and tracking, "*ISMAR '11,* pp.127-136, Basel, Switzerland, 26-29th October 2011.

[6] M.Calonder, V. Lepetit and C. Strecha and P.Fua, "BRIEF: Binary Robust Independent Elementary Features, " In Proc. European Conference on Computer Vision 2010

[7] E.Rublee, V.Rabud, K.Konolige and G.Bradski, "ORB: an efficient alternative to SIFT or SURF", In Proc. International Conference on Computer Vision, 2011.

[8] Point Cloud Library: http://pointclouds.org/

[9] OpenCV: http://opencv.org/