

# 消失点を用いた RGB-D センサによる単調な環境での 6 自由度 SLAM

池 勇勲<sup>\*1</sup>, 山下 淳<sup>\*1</sup>, 浅間 一<sup>\*1</sup>

## RGB-D Sensor-based 6DOF SLAM for a Monotonic Environment with a Vanishing Point

Yonghoon JI<sup>\*1</sup>, Atsushi YAMASHITA<sup>\*1</sup>, and Hajime ASAMA<sup>\*1</sup>

<sup>\*1</sup> Department of Precision Engineering, The University of Tokyo  
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan

This paper deals with RGB-D sensor-based 6DOF SLAM (simultaneous localization and mapping) in a monotonic environment where few features for matching are extracted. A challenging point for the SLAM problem is matching the spatial alignment of the sensor data. To deal with this problem, this paper proposes a novel matching scheme using a vanishing point which utilized for the observation model of the extended Kalman filter-based SLAM, so-called EKF SLAM. The 6DOF pose data can be corrected by matching the extracted vanishing point from the image data and the precision 3D map is represented by octree structure from the 3D point cloud data. The experimental results in the monotonic environments show that the proposed vanishing point matching scheme can enhance performance of indoor SLAM even if any other robust features for matching are hard to be extracted.

**Key Words** : SLAM, Kalman filter, RGB-D sensor, point cloud

### 1. 序論

環境のモデルとして用いられる地図は、移動ロボットの位置推定、経路計画などの様々な走行アルゴリズムのベースとなり、人間とロボット間の相互作用の側面においても重要な役割を果たす。そのため、精密な地図の生成は、移動ロボットの研究において最も重要である。

地図生成について重要な課題は、地図の全体的な位置ずれを減少させるため、取得したデータ間のマッチングを行うことである。従来の研究では、主に ICP (iterative closest points) などのスキャンマッチングと呼ばれる手法がよく用いられている<sup>(1,2)</sup>。しかし、ICP などのスキャンマッチングの方法は、すべてのデータ間の対応関係の探索を繰り返すため、膨大な計算量が必要になり、リアルタイムに動作させることが困難であるという問題点がある。また、スキャンマッチング手法の多くは、データの形状情報を利用してマッチングを行うため、形状に変化が少ない単調な環境、すなわち幾何特徴の少ない環境に対応不可能な問題がある。例えば、建物内部の廊下など、凹凸のない壁が続く環境などにおいては、スキャンマッチング手法では正確なマッチングができない。これらの問題の解決のため、レーザーレンジファインダ (laser range finder) の反射

強度情報を用い、幾何特徴の少ない環境においても、ロバストな ICP を可能にした intensity-ICP が提案された<sup>(3)</sup>。しかし、反射強度情報の取得が不可能な測距センサへの適用は不可能である。

以上のような問題をふまえ、本研究では、特徴の少ない単調な廊下環境での正確な 3 次元地図を生成するため、画像情報から抽出した消失点を用いた SLAM (simultaneous localization and mapping) 手法を提案する。SLAM とは、自己位置と地図の状態を同時に推定する地図作成の一般的な手法であり、本研究では、EKF SLAM と呼ばれる拡張カルマンフィルタ (extended Kalman filter) に基づいた SLAM 手法を使用する<sup>(4)</sup>。各画像の消失点の位置を求め、マッチングに用いることで、幾何特徴の少ない環境においても、センサの位置と姿勢を 6 自由度で正確に推定することができ、ずれの小さい地図の生成が可能である。また、EKF SLAM の場合、特徴の数が増加するにつれ計算時間が指数関

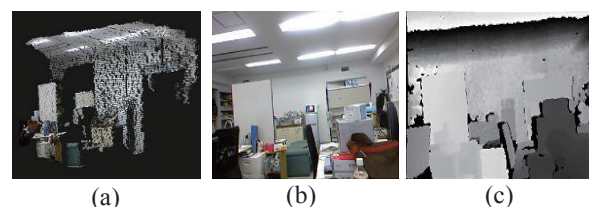


Fig. 1 Data from RGB-D sensor: (a) point cloud, (b) RGB image, and (c) range image.

<sup>\*1</sup> 東京大学大学院工学系研究科精密工学専攻 (〒113-8656 東京都文京区本郷 7-3-1) {ji, yamashita, asama}@robot.u-tokyo.ac.jp

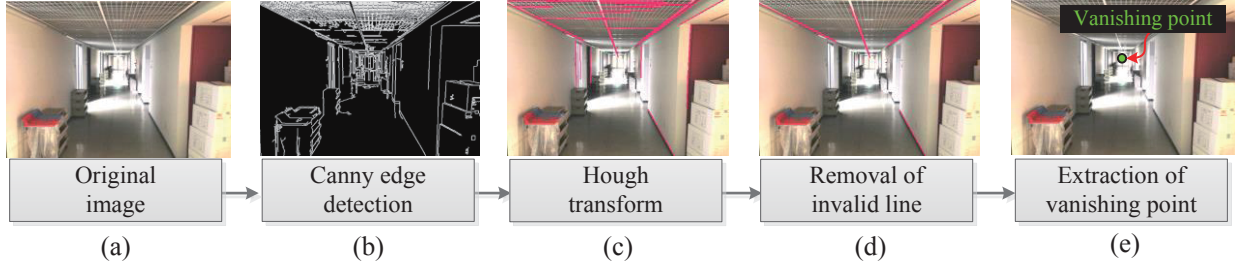


Fig. 2 Procedure of the line-based vanishing point extraction.

数的に増加する欠点が存在するが、本研究では消失点という 1 つの代表点のみを使用して対応関係を求めるため、非常に高速なリアルタイム地図生成が可能であるという利点がある。

一方、従来の屋内環境での自律走行機能のみ行われる単純なサービスロボットの場合は、2 次元の地図で環境の表現が十分であった。しかし、複雑な空間上で作業を行うサービスロボットの場合、その優れた作業に対応可能な精密な 3 次元地図が不可欠である。このため本研究においては、3 次元情報の取得が可能な RGB-D センサのみを用い、3 次元地図を生成する SLAM を行う。RGB-D センサからは、図 1(a)~(c)に示すように、環境に対する 3 次元点群データ、RGB 画像、そして距離画像をリアルタイムで取得可能である。

## 2. 消失点抽出

消失点とは遠近法において、実際には平行であるものを平行でなく描く際に、その線が交わる点であり、理論的に、この点は無限遠点である<sup>(5)</sup>。廊下のような直線方向の長い通路は、1 つの消失点に収束する特徴を持つ。したがって、消失点はずれた位置を補正する際、非常に有用な特徴となる。消失点を抽出する過程を図 2 に示す。まず、図 2(b)に示すように Canny エッジを抽出したバイナリ画像を生成する。そして、図 2(c)に示すように Hough 変換によって画像内の直線の候補群を抽出する。続いて、図 2(d)に示すように画像差表

系  $u, v$  にそれぞれ平行である直線を検出し、ドア、ポスターなどの人工物から抽出される直線（消失点に収束しない直線）を削除する。最終的に残った直線から唯一の消失点を抽出することが可能である。消失点に収束する直線が与えられたとき、消失点を抽出するための直線の方程式は次のように表現される。

$$a_i u + b_i v = c_i \quad (1)$$

ただし、 $i=1, 2, \dots, M$  である。ここで、 $(a_i, b_i, c_i)$  は、各直線の方程式の係数であり、 $M$  は直線の数、 $(u, v)$  は画像座標を意味する。検出された直線の係数を式(1)に代入し、行列—ベクトルの形で表現すると、次のとおりである。

$$A u = c \quad (2)$$

$$\begin{bmatrix} a_1 & b_1 \\ a_2 & b_2 \\ \vdots & \vdots \\ a_M & b_M \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_M \end{bmatrix} \quad (3)$$

ここで、消失点の推定値  $u=(u, v)$  は次式のように Moore Penrose pseudo-inverse 行列を用いた最小二乗法により推定される。

$$u = A^\Psi c \quad (4)$$

ただし、 $A^\Psi = (A^T A)^{-1} A^T$  である。上記の方法で抽出された消失点を、図 2(e)に示す。

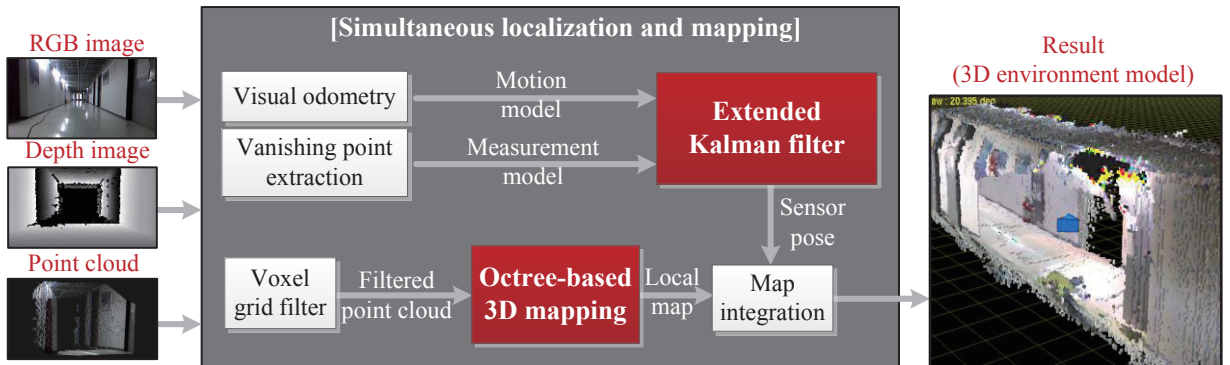


Fig. 3 Overview of proposed vanishing point-based EKF SLAM.

### 3. EKF SLAM

本研究で提案する全体的な地図生成過程の概要を図3に示す。センサの姿勢は画像データを用いて推定され、この際、提案手法であるEKF SLAMに基づく消失点マッチングが行われる。また3次元点群データからは8分木(octree)構造の3次元地図が作成される。

EKF SLAMは運動モデルによる予測ステップと、観測モデルによる更新ステップに分けられ、本研究では、運動モデルと観測モデルとして、それぞれ visual odometry からの制御入力情報と、消失点情報を用いる。EKF SLAMの推定対象となる、センサと消失点に対する状態ベクトル $\mathbf{x}_t$ と、不確実性を表す共分散行列 $\mathbf{P}_t$ は、次のとおりである。

$$\mathbf{x}_t = [\mathbf{x}_{S,t} \quad \mathbf{x}_{V,t}]^T \quad (5)$$

$$\mathbf{x}_{S,t} = [x_{S,t} \quad y_{S,t} \quad z_{S,t} \quad \psi_{S,t} \quad \theta_{S,t} \quad \phi_{S,t}]^T \quad (6)$$

$$\mathbf{x}_{V,t} = [x_{V,t} \quad y_{V,t} \quad z_{V,t}]^T \quad (7)$$

$$\mathbf{P}_t = \begin{bmatrix} \mathbf{P}_{S,t} & \mathbf{P}_{SV,t} \\ \mathbf{P}_{VS,t} & \mathbf{P}_{V,t} \end{bmatrix} \quad (8)$$

ここで、 $\mathbf{x}_{S,t}$ は時間 $t$ でのセンサの位置と姿勢、 $\mathbf{x}_{V,t}$ は3次元空間上の消失点の位置を意味する状態ベクトルであり、グローバル座標系に対する状態変数である。また、共分散行列 $\mathbf{P}_t$ の対角要素 $\mathbf{P}_{S,t}$ と $\mathbf{P}_{V,t}$ はそれぞれセンサと消失点の位置と姿勢に対する共分散であり、非対角要素 $\mathbf{P}_{SV,t}$ と $\mathbf{P}_{VS,t}$ は、それぞれ $\mathbf{P}_{S,t}$ と $\mathbf{P}_{V,t}$ に関する交差共分散(cross covariance)を意味する。本章ではEKF SLAM アルゴリズムを予測ステップと更新ステップに分けて簡単に説明する。

**3.1 予測ステップ(prediction step)** EKF SLAMの予測ステップでは、visual odometry 情報に基づいた運動モデルを定義し、これによりセンサの姿勢と不確実性を計算する。Visual odometry の基本的な概念は、図4に示すように、連続する画像内の特徴点間の変化

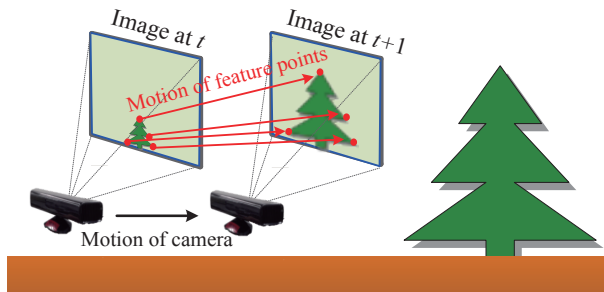


Fig. 4 Illustration of core idea of visual odometry

量を計算することで、センサの運動を推定する手法である。数多くの関連研究の中で、本研究では、MITで開発された Fovis (fast odometry from vision) ライブラリを使用する<sup>(6,7)</sup>。具体的には、予測ステップでは、時間 $t$ で推定された状態ベクトル $\mathbf{x}_t$ に、visual odometry 情報から求められた時間 $t+1$ での制御入力情報 $\Delta\mathbf{x}_{t+1}$ を適用し、時間 $t+1$ での状態ベクトル $\mathbf{x}_{t+1}$ と共分散 $\mathbf{P}_{t+1}$ を次のように予測する。

$$\mathbf{x}_{t+1}^- = \mathbf{g}(\mathbf{x}_t, \Delta\mathbf{x}_{t+1}) \quad (9)$$

$$\mathbf{P}_{t+1}^- = \mathbf{G}_{t+1} \mathbf{P}_t \mathbf{G}_{t+1}^T + \mathbf{M}_{t+1} \quad (10)$$

ここで、 $\mathbf{g}(\cdot)$ はシステムの運動モデルを意味し、 $\mathbf{G}_{t+1} = \partial\mathbf{g}/\partial\mathbf{x}_t$ は非線形関数 $\mathbf{g}(\cdot)$ の状態ベクトルに対するヤコビ行列、 $\mathbf{M}_{t+1}$ はプロセスノイズ行列である。また、上付き文字 ‘-’は観測情報(本研究では消失点情報)によって更新されていない状態であることを示す。このように予測ステップでは、制御入力とプロセスノイズを使用し、センサの姿勢や不確実性の大きさを予測する。

**3.2 更新ステップ(update step)** EKF SLAMの更新ステップでは、抽出された消失点の情報を使用し、予測ステップで推定された状態ベクトル $\mathbf{x}_t^-$ と、共分散行列 $\mathbf{P}_t^-$ を更新する。センサ座標系とグローバル座標系の関係は、予測された状態ベクトルに基づいた観測モデルによって定義される。本研究では、図5のように、抽出された消失点の画像座標による、位置情報を観測情報 $\mathbf{z}_t = (z_{u,t}, z_{v,t})$ として用いる。この値を観測情報として使用することで、特に、地図の位置ずれに大きな影響を与える $y, z$ 軸方向の誤差がうまく補正される。消失点による観測モデル $\mathbf{h}(\cdot)$ は、次のように定義される。

$$\hat{\mathbf{z}}_t = \mathbf{h}(\mathbf{x}_t) \quad (11)$$

$$\hat{\mathbf{z}}_t = \begin{bmatrix} \hat{z}_{u,t} \\ \hat{z}_{v,t} \end{bmatrix} = \frac{f}{d - x_{S,t}} \mathbf{R}_{(2:3),t}^{-1} \begin{bmatrix} x_{V,t} - x_{S,t} \\ y_{S,t} - y_{V,t} \\ z_{V,t} - z_{S,t} \end{bmatrix} \quad (12)$$

$$\mathbf{R}_t = \begin{bmatrix} c_\phi c_\theta & c_\phi s_\theta s_\psi - s_\phi c_\psi & c_\phi s_\theta c_\psi + s_\phi s_\psi \\ s_\phi c_\theta & s_\phi s_\theta s_\psi + c_\phi c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi \\ -s_\theta & c_\theta s_\psi & c_\theta c_\psi \end{bmatrix} \quad (13)$$

ただし $\mathbf{R}_t$ はセンサの姿勢に関するオイラー角の回転行列、 $\mathbf{R}_{(2:3),t}^{-1}$ は $\mathbf{R}_t$ の逆行列から1行目を除いた行列で、 $s_\psi = \sin\psi_{S,t}$ ,  $s_\theta = \sin\theta_{S,t}$ ,  $s_\phi = \sin\phi_{S,t}$ ,  $c_\psi = \cos\psi_{S,t}$ ,  $c_\theta = \cos\theta_{S,t}$ ,  $c_\phi = \cos\phi_{S,t}$ である。ここで、 $\hat{\mathbf{z}}_t$ は時間 $t$ で予測された状態ベクトル $\mathbf{x}_t$ に基づいて予測される消失点の情

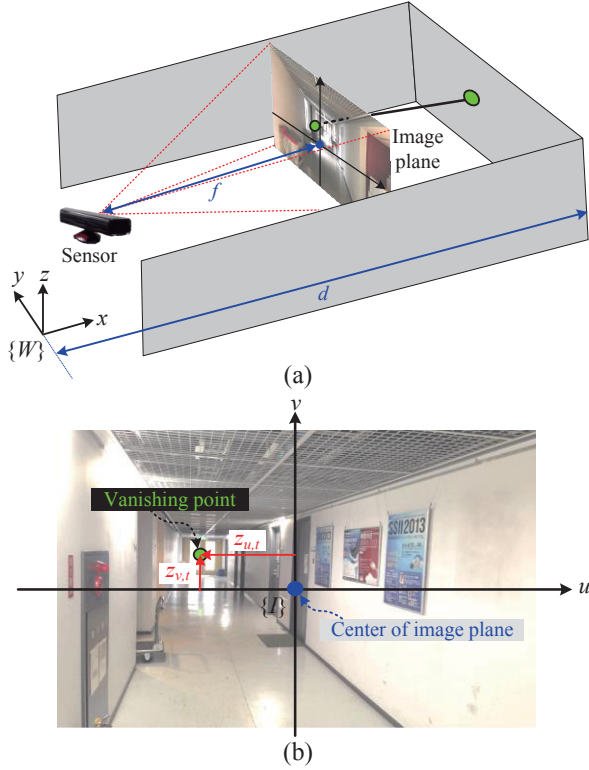


Fig. 5. Vanishing point-based observation model of EKF SLAM: (a) image plane with global coordinate, and (b) detailed representation of image plane.

報を意味し,  $f$  はカメラの焦点距離 (focal length) を意味する. また, 消失点とは実空間では無限遠に相当するが, 無限遠である仮想の点を観測情報として用いるのは困難であるため, 本研究では, この消失点を実環境に投影される有限距離の点を観測情報として用いる. つまり  $d$  は, 図 5(a) に示すように, グローバル座標系の原点と仮想の消失点が投影される環境の点までの  $x$  軸方向の距離で, 既知である.

□消失点情報に基づく, 状態ベクトルと共分散の更新は情報咬合 (data association) を通じて, 以前に抽出された消失点と同じものであると判別された場合のみ実行される. 情報咬合は次のようにマハラノビス距離 (Mahalanobis distance) に基づいて行われる.

$$\mathbf{S}_t = \mathbf{H}_t \mathbf{P}_t^- \mathbf{H}_t^T + \mathbf{Q} \quad (14)$$

$$(\mathbf{z}_t - \hat{\mathbf{z}}_t)^T \mathbf{S}_t^{-1} (\mathbf{z}_t - \hat{\mathbf{z}}_t) < \pi \quad (15)$$

ここで  $\mathbf{H}_{t+1} = \partial \mathbf{h} / \partial \mathbf{x}_t$  は, 状態ベクトルに対する観測モデルのヤコビ行列,  $\mathbf{Q}$  は観測値に対するノイズ行列,  $\mathbf{S}_t$  はセンサの状態に対する共分散とセンサノイズを含む消失点の共分散を意味する. もし,

式(15)で計算したマハラノビス距離があらかじめ定義したしきい値  $\pi$  より大きい場合は, 誤って抽出された消失点と判断し, 更新処理を実行しない. 一方, マハラノビス距離が  $\pi$  より小さい場合は, 情報咬合に成功したと判断し, 次のようにシステムの状態を更新する. すなわち, 更新ステップでは, 情報咬合に成功した場合, 時間  $t+1$  で測定された観測情報  $\mathbf{z}_{t+1}$  を使用し, 予測ステップで推定された状態ベクトル  $\mathbf{x}_t^-$  と共分散  $\mathbf{P}_t^-$  を次のように  $\mathbf{x}_{t+1}$  と  $\mathbf{P}_{t+1}$  に更新する.

$$\mathbf{K}_{t+1} = \mathbf{P}_{t+1}^- \mathbf{H}_{t+1}^T [\mathbf{H}_{t+1} \mathbf{P}_{t+1}^- \mathbf{H}_{t+1}^T + \mathbf{Q}]^{-1} \quad (16)$$

$$\mathbf{x}_{t+1} = \mathbf{x}_{t+1}^- + \mathbf{K}_{t+1} [\mathbf{z}_{t+1} - \mathbf{h}(\mathbf{x}_{t+1}^-)] \quad (17)$$

$$\mathbf{P}_{t+1} = [\mathbf{I} - \mathbf{K}_{t+1} \mathbf{H}_{t+1}] \mathbf{P}_{t+1}^- \quad (18)$$

ここで,  $\mathbf{K}$  はカルマンゲインである. このように提案したEKF SLAMは, 毎時刻, 消失点を一致させるマッチングを行うことにより, センサのより正確な姿勢を推定することが可能である.

#### 4. 地図生成

地図生成において, RGB-D センサから取得したすべての点群データを使用すると, 非常に計算の負担がかかる問題がある. 例えば,  $640 \times 480$  の解像度の画像

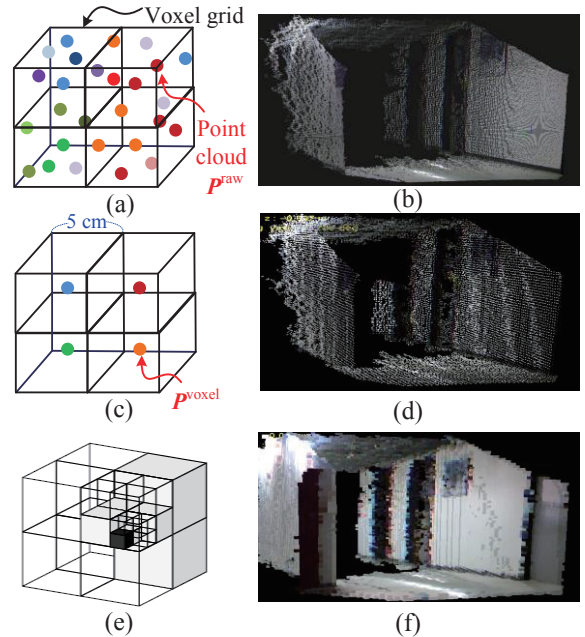


Fig. 6 3D mapping: (a), (b) original point cloud (size: about 300,000 points), (c), (d) after applying voxel grid filter (size: about 20,000 points), (e), and (f) related OctoMap.

を使用する場合、1 フレームごとに 307,200 個の点群データを処理しなければならない。これは、データの処理速度において非常に非効率的である。そのため、本研究では、図 6(a)~(d)のように、数多くの点群データに 3 次元立方体形状の格子を導入し、中心に近い 1 個の点のみをデータとして取る voxel grid filter を用いる。これにより均一に点群の密度を減少させることが可能である。本研究では、1 辺 5cm の立方体を使用した。したがって、以降の作業で実行する演算量を削減することが可能である。そして、毎時刻、各センサの姿勢から生成されたローカル地図の空間的な統合のため、センサ座標系に対するすべての点群  $^S p_t^{\text{voxel}}$  を、次式のようにグローバル座標系に対する  $p_t^{\text{voxel}}$  に座標変換を行う。

$$p_t^{\text{voxel}i} = T_t^S {}^S p_t^{\text{voxel}i} \quad (19)$$

$$\begin{bmatrix} x_t^i \\ y_t^i \\ z_t^i \\ 1 \end{bmatrix} = \begin{bmatrix} R_t & x_{S,t} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} {}^S x_t^i \\ {}^S y_t^i \\ {}^S z_t^i \\ 1 \end{bmatrix} \quad (20)$$

ここで、 $p_t^{\text{voxel}} = \{(x_t^i, y_t^i, z_t^i) | i=1, 2, \dots, N^{\text{voxel}}\}$  はフィルタリングされた点群、 $N^{\text{voxel}}$  はボクセルの数である。最終的に本研究では、Fribrug 大学で開発された OctoMap ライブラリを使用し、点群データ  $p_t^{\text{voxel}}$  の確率的更新による 3 次元地図を生成する<sup>(8,9)</sup>。生成された OctoMap の概念図と様子をそれぞれ図 6(e), (f) に示す。これらの 8 分木の形のデータ構造は、メモリの側面で非常に効率的である。このようにすべての時刻で生成されたローカル地図を 1 つの空間に統合することでグローバル地図を生成する。



Fig. 7 ASUS Xtion Live Pro.

Table 1 Specifications of ASUS Xtion Live Pro.

|                                        |                                                               |
|----------------------------------------|---------------------------------------------------------------|
| Maximum range                          | 3.5 m                                                         |
| Field of view                          | 58 deg (horizontal)<br>45 deg (vertical)<br>70 deg (diagonal) |
| Frame rate<br>(depth and color stream) | 30 FPS (640 × 480)<br>60 FPS (320 × 240)                      |
| Size                                   | 180 × 35 × 50 mm                                              |
| Data interface                         | USB 2.0                                                       |

## 5. 実験結果

本研究においては、提案したアルゴリズムの実装のため、図 7 に示す ASUS 社の Xtion Pro Live センサと、公開されている OpenNI ライブラリを使用した<sup>(10)</sup>。Xtion Pro Live の主な仕様を表 1 に示す。

実験は、図 9(a) に示すように、マッチングに使用することが可能な幾何特徴がほとんど存在しない、単調な廊下環境で行われた。誤差を意図的に多く発生させるため、人がセンサを持ち、約 0.8m/s の速度でジグザグ運動をし、センサデータの取得周期は 300ms とした。上記のような同一の実験条件で消失点の情報を用いず実験を行った場合、提案手法により消失点を用いてマッチングを行った場合のセンサの軌跡を、それぞれ図 8 に示す。消失点の情報を用いず、実験を行った場合は、実行速度は最も速いが、特別なマッチングを行わないため、大きな位置ずれのある不正確な軌跡が推定された。例えば、ほぼ同じ高さを維持しながら、データを取得したにもかかわらず、開始点と終了点の間に約 5m の  $z$  軸方向の誤差が発生した。これに対し、提案手法により消失点を用いてマッチングを行った場合は、位置ずれの誤差がほぼ存在しない直線に近い軌跡が推定され、図 9 に示すように、実際の廊下の環境と同じような地図が生成された。また、マッチングを行わない場合に比べて若干実行時間は増加し、260ms であったが、センサデータの取得周期内に行われるため、リアルタイムの地図生成が可能であった。

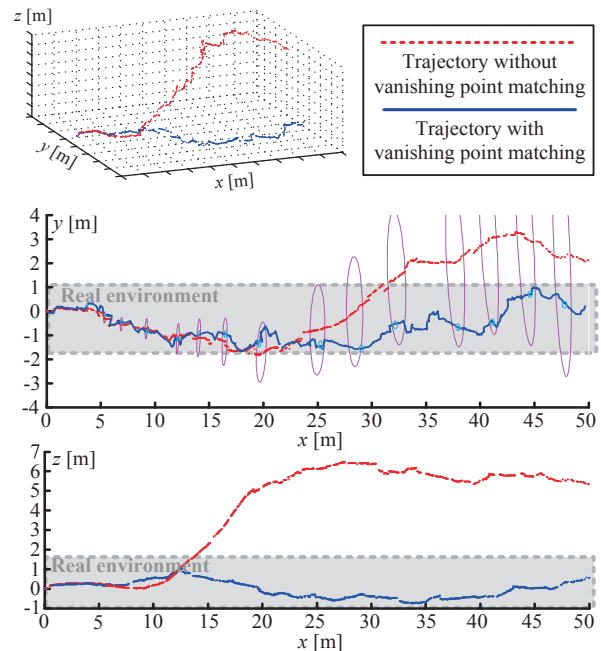


Fig. 8 Comparison of estimated trajectories. Ellipses mean position uncertainty of sensor state.

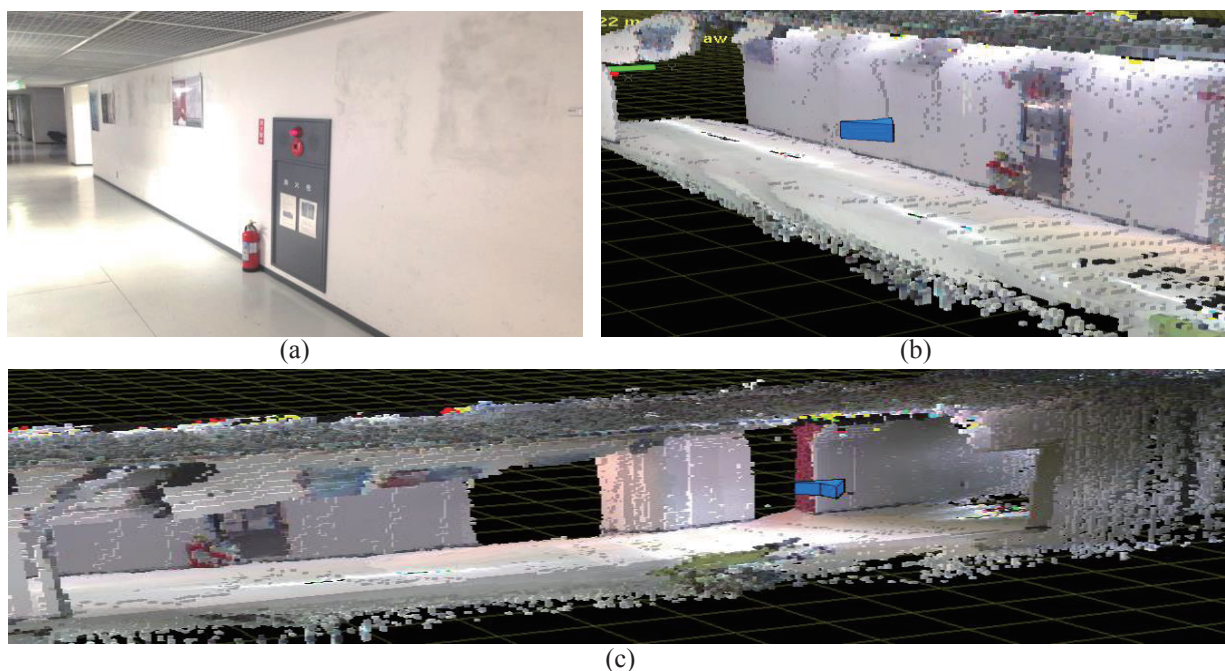


Fig. 9 SLAM result: (a) real monotonic environment where few features for matching are extracted, (b), and (c) octree-based built environment model with vanishing point matching.

## 6. 結論

本研究では、RGB-D センサから取得した RGB 画像から消失点を抽出し、SLAM の観測モデルとして用いることで、空間的に位置ずれのない、より正確な 3 次元環境モデルをリアルタイムに生成する手法を提案した。提案手法は、単調な環境においてもマッチングに対してロバストである。また、消失点は、非常に速い速度で抽出とマッチングが可能のため、非常に有用な特徴として活用されることを示した。実際に、建物の中の廊下のように形状の変化が少なく、幾何特徴のほとんど存在しない環境においても、より安定的にマッチングを行い、位置ずれの小さい精密な 6 自由度 SLAM が実現された。

今後の展望としては、提案方法は単調な直線環境でのみ対応可能であるという限界があるため、今後はより多様な環境に対応可能な手法を考案する必要がある。そのため、遠近法の種類によって複数の消失点が存在するという原理を用い、より複雑な空間でも、複数の消失点を抽出し、マッチングに活用する手法について研究を進める計画である。また、本研究で提案した消失点に基づく観測モデルは、位置ずれに影響の大きい  $y$  と  $z$  軸方向の誤差に対して、ロバストに補正が行われるが、 $x$  軸方向の移動量はうまく補正できない問題がある。そのため、他の特徴との融合モデルを構築して、全体的な性能を向上させる計画である。

## 参考文献

- (1) Szymon Rusinkiewicz and Marc Levoy, "Efficient Variants of the ICP Algorithm", *Proceeding of the International Conference on 3D Digital Imaging and Modeling*, (2001), pp. 145-152.
- (2) Peter Henry, Michael Krainin, Evan Herbst, Xiaofeng Ren and Dieter Fox, "RGB-D Mapping: Using Kinect-style Depth Cameras for Dense 3D Modeling of Indoor Environment", *The International Journal of Robotics Research*, Vol. 31, No. 5, (2012), pp. 647-663.
- (3) Yoshitaka Hara, Hirohiko Kawata, Akihisa Ohya and Shin'ichi Yuta, "Map Building for Mobile Robots using a SOKUIKI Sensor: Robust Scan Matching using Laser Reflection Intensity", *Proceeding of SICE-ICASE International Joint Conference*, (2006), pp.5951-5956.
- (4) Sebastian Thrun, Wolfram Burgard and Dieter Fox, "Probabilistic Robotics", The MIT Press, (2005).
- (5) Ernest R. Norling, "Perspective Made Easy", BN Publishing, (2008).
- (6) Albert S. Huang, Abraham Bachrach, Peter Henry, Michael Krainin, Daniel Maturana, Dieter Fox and Nicholas Roy, "Visual Odometry and Mapping for Autonomous Flight Using an RGB-D Camera", *Proceeding of the International Symposium on Robotics Research*, (2011).
- (7) Fovis: (<http://code.google.com/p/fovish/>)
- (8) Armin Horung, Kai M. Wurm, Maren Bennewitz, Cyrill Stachniss and Wolfram Burgard, "OctoMap: An Efficient Probabilistic 3D Mapping Framework Based on Octrees", *Autonomous Robots*, Vol. 34, No. 3, (2013), pp.189-206.
- (9) OctoMap: (<http://octomap.github.io/>)
- (10) OpenNI: (<http://www.openni.org/>)