

# RGB-D センサを用いた単調な環境における 消失点マッチングによるボクセル地図生成

○池 勇勳 (東京大学), 山下 淳 (東京大学), 浅間 一 (東京大学)

## Vanishing Point Matching-based Voxel Mapping Using an RGB-D Sensor in a Monotonic Environment

○Yonghoon JI (The University of Tokyo), Atsushi YAMASHITA (The University of Tokyo),  
and Hajime ASAMA (The University of Tokyo)

Abstract: An environment model is a crucial part of a mobile robot navigation or human robot interaction. A challenging point for the precision 3D environment modeling is matching the spatial alignment of consecutive data from the sensors. To deal with this problem, this paper proposes extended Kalman filter-based vanishing point matching that utilizes a novel pose correction algorithm in a monotonic environment where robust features for matching are not extracted. In this paper, the precision 3D map is represented by filtered voxel grids from 3D point cloud data. The experimental results show that the accuracy of map increases and the proposed matching method is useful for robust 3D mapping in the monotonic environment using an RGB-D sensor.

### 1. 序論

移動ロボットの研究において、環境のモデルは、位置認識、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、ロボットと人間との相互作用においても重要な役割を果たす。環境のモデルとしての三次元の地図生成について最も重要な課題は、地図の全体的な位置ずれを減少させるため、取得した連続的なセンサ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[1,2]。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、データの形状情報を利用してマッチングを行うため、建物内部の廊下など、幾何特徴の少ない単調な凹凸のない壁が続く環境に対応できない問題がある。

以上のような問題をふまえて、本研究では、特徴の少ない単調な廊下環境での正確な地図を生成するため、画像情報から消失点を抽出し、マッチングを行う手法を提案する。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ずれの小さい地図の生成ができる。また一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。

本研究においては、RGB-D センサのみを用い、ボクセルベースの三次元地図を生成する。RGB-D センサからは Fig. 1(a)~(c)に示すように、環境に対する三次元点

群データと、RGB 画像、そして距離画像をリアルタイムで取得する。

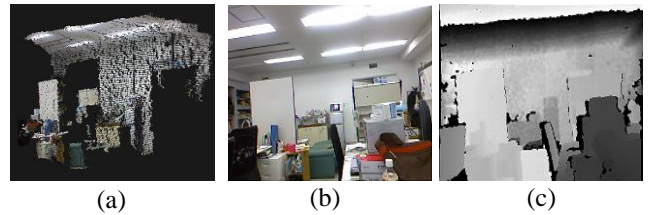


Fig. 1 Data from RGB-D sensor: (a) point cloud, (b) RGB image, and (c) depth image.

### 2. 提案システム

本研究で提案する地図生成過程の概要を Fig. 2 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 (Extended Kalman filter) に基づく消失点マッチングが行われる。また、三次元点群データからはボクセル地図が作成される。

#### 2.1 姿勢推定 (localization)

本研究では、visual odometry からの制御入力情報と、消失点情報を融合し、センサの姿勢を推定するため EKF を用いる。Visual odometry は、連続する画像内の特徴点間の変化量を計算し、カメラの運動を推定する手法であり、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[3,4]。EKF は運動モデルによる予測ステップと、計測モデルによる更新ステップに分けられる[5]。

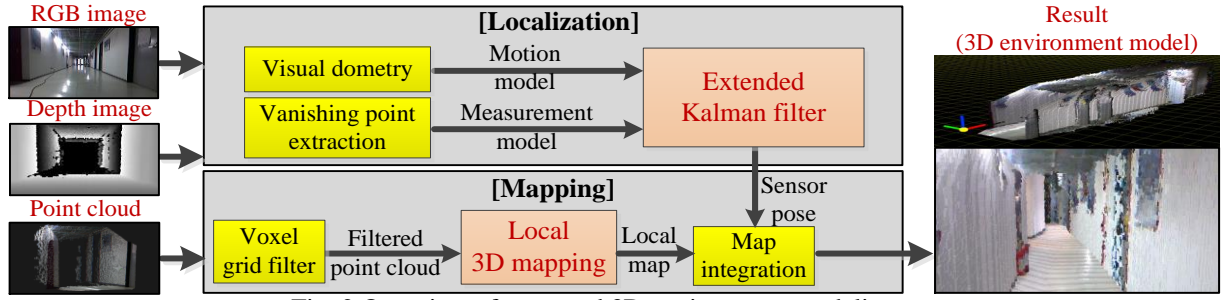


Fig. 2 Overview of proposed 3D environment modeling.

### 2.2.1 予測ステップ (prediction step)

EKF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予測する。センサの姿勢と不確実性を表す変数  $\mathbf{x}_t$  と  $\mathbf{P}_t$  は、次のとおりである。

$$\mathbf{x}_t = \begin{bmatrix} x_t & y_t & \varphi_t \end{bmatrix}^T \quad (1)$$

$$\mathbf{P}_t = \begin{bmatrix} \sigma_x^2 & \sigma_{xy}^2 & \sigma_{x\varphi}^2 \\ \sigma_{yx}^2 & \sigma_y^2 & \sigma_{y\varphi}^2 \\ \sigma_{\varphi x}^2 & \sigma_{\varphi y}^2 & \sigma_\varphi^2 \end{bmatrix}^T \quad (2)$$

ここで、 $\mathbf{x}_t$  は時間  $t$  でのロボットの位置と姿勢を示す状態ベクトル、 $\mathbf{P}_t$  は  $\mathbf{x}_t$  の不確実性を示す共分散行列である。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、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。

$$\mathbf{x}_{t+1}^- = \mathbf{g}(\mathbf{x}_t, \Delta\mathbf{x}_{t+1}) \quad (3)$$

$$\begin{bmatrix} x_{t+1}^- & y_{t+1}^- & \varphi_{t+1}^- \end{bmatrix}^T = \begin{bmatrix} x_t + \Delta x_{t+1} \cos\left(\varphi_t + \frac{\Delta\varphi_{t+1}}{2}\right) - \Delta y_{t+1} \sin\left(\varphi_t + \frac{\Delta\varphi_{t+1}}{2}\right) \\ y_t + \Delta x_{t+1} \sin\left(\varphi_t + \frac{\Delta\varphi_{t+1}}{2}\right) + \Delta y_{t+1} \cos\left(\varphi_t + \frac{\Delta\varphi_{t+1}}{2}\right) \\ \varphi_t + \Delta\varphi_{t+1} \end{bmatrix} \quad (4)$$

$$\mathbf{P}_{t+1}^- = \mathbf{G}_{t+1} \mathbf{P}_t \mathbf{G}_{t+1}^T + \mathbf{M}_{t+1} \quad (5)$$

$$\mathbf{M}_{t+1} = \begin{bmatrix} k_x |\Delta x_{t+1}| & 0 & 0 \\ 0 & k_y |\Delta y_{t+1}| & 0 \\ 0 & 0 & k_\varphi |\Delta\varphi_{t+1}| \end{bmatrix}^T \quad (6)$$

ここで、 $\mathbf{g}$  はシステムの運動モデルを示し、制御入力  $\Delta\mathbf{x}_{t+1} = (\Delta x_{t+1}, \Delta y_{t+1}, \Delta\varphi_{t+1})$  は visual odometry 情報から求められる。また  $\mathbf{P}_{t+1}$  は  $\mathbf{x}_{t+1}$  の共分散行列であり、 $\mathbf{G}_{t+1} = \partial\mathbf{g}/\partial\mathbf{x}_t$  は非線形関数  $\mathbf{g}$  の状態ベクトルに対するヤコビ行列、 $\mathbf{M}_t$  はプロセスノイズであり、各要素は、各制御入力の移動量に比例する変数で構成される。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。

### 2.2.2 更新ステップ (update step)

EKFの更新ステップでは、計測した観測情報  $\mathbf{z}_t$  を使用し、予測ステップで推定したセンサの姿勢  $\mathbf{x}_t^-$  と不確実性  $\mathbf{P}_t^-$  を補正する。まず、観測情報として使用する消失点を毎時刻抽出する。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る。理論的にはこの点は無限遠点である[6]。廊下のような直線方向の長い通路は、一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 Fig. 3 に示す。

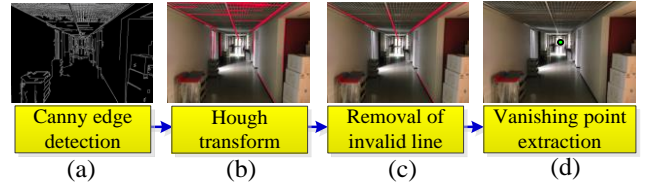


Fig. 3 Line-based Vanishing point extraction.

まず、Fig. 3(a)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、Fig. 3(b)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、Fig. 3(c)に示すように角度条件によって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。

$$a_i u + b_i v = c_i \quad (7)$$

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(7)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。

$$\mathbf{A}\mathbf{u} = \mathbf{c} \quad (8)$$

$$\begin{bmatrix} a_1 & b_1 \\ a_2 & b_2 \\ \vdots & \vdots \\ a_M & a_M \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_M \end{bmatrix} \quad (9)$$

ここで、消失点の推定値  $\mathbf{u}=(u, v)$  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。

$$\mathbf{u} = \mathbf{A}^\Psi \mathbf{c} \quad (10)$$

ただし、 $\mathbf{A}^\Psi = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T$  である。上記の方法で抽出された消失点を Fig. 3(d) に示す。

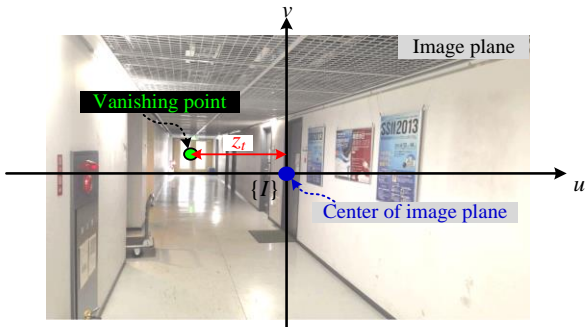


Fig. 4 Vanishing point-based EKF observation model.

本研究では Fig. 4 のように、抽出された消失点の画像座標による、水平方向の画素位置情報を観測情報  $z_t$  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回転角の誤差がうまく補正される。消失点による観測モデル  $\mathbf{h}$  は、次のように定義される。

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

$$\hat{z}_t = \{(x_t - d) \sin \varphi_t - y_t \cos \varphi_t\} \frac{f}{d - x_t} \quad (12)$$

ここで、 $\hat{z}_t$  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報を意味する。また、 $d$  はグローバル座標系の原点と仮定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、事前に認知していると仮定する。すなわち、更新ステップでは、時間  $t+1$  で測定された観測情報  $z_{t+1}$  を使用し、予測ステップで推定された状態ベクトル  $\mathbf{x}_{t+1}^-$  と共分散  $\mathbf{P}_{t+1}^-$  を次のように  $\mathbf{x}_{t+1}$  と  $\mathbf{P}_{t+1}$  に更新する。

$$\mathbf{K}_{t+1} = \mathbf{P}_{t+1}^- \mathbf{H}_{t+1}^T [\mathbf{H}_{t+1} \mathbf{P}_{t+1}^- \mathbf{H}_{t+1}^T + \mathbf{q}]^{-1} \quad (13)$$

$$\mathbf{x}_{t+1} = \mathbf{x}_{t+1}^- + \mathbf{K}_{t+1} [z_{t+1} - \mathbf{h}(\mathbf{x}_{t+1}^-)] \quad (14)$$

$$\mathbf{P}_{t+1} = [\mathbf{I} - \mathbf{K}_{t+1} \mathbf{H}_{t+1}] \mathbf{P}_{t+1}^- \quad (15)$$

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、 $\mathbf{K}$  はカルマンゲイン、 $\mathbf{q} = \sigma_u$  は観測値に対するノイズである。このように EKF を使用し、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。

## 2.2 地図生成 (mapping)

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、Fig. 5 のように、数多くの点群データに三次元ボックス形状の格子を導入し、中心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たボクセルベースの地図を生成することが可能である。本研究では、一辺 5cm のボクセルを使用した。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。

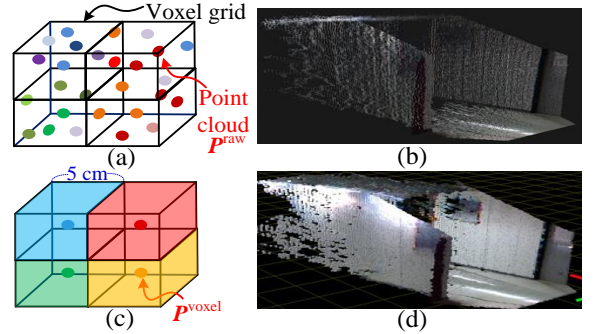


Fig. 5 Voxel-based 3D mapping: (a), (b) original point cloud (size: about 300,000 points), (c) and, (d) after applying voxel grid filter (size: about 20,000 points).

そしてグローバル地図を生成するため、まず、センサ座標系の  $\mathbf{P}_t^{\text{voxel}}$  を、次式のようにグローバル座標系の  ${}^G \mathbf{P}_t^{\text{voxel}}$  に座標変換を行う。このようにすべての時刻で生成された地図  ${}^G \mathbf{P}_t^{\text{voxel}}$  を一つの空間に統合することでグローバル地図を生成する。

$${}^G \mathbf{P}_t^{\text{voxel},i} = \mathbf{T}_t \mathbf{P}_t^{\text{voxel},i} \quad (16)$$

$$\begin{bmatrix} G x_t^i \\ G y_t^i \\ G z_t^i \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \varphi_t & -\sin \varphi_t & 0 & x_t \\ \sin \varphi_t & \cos \varphi_t & 0 & y_t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_t^i \\ y_t^i \\ z_t^i \\ 1 \end{bmatrix} \quad (17)$$

ここで、 $\mathbf{P}_t^{\text{voxel}} = \{(x_t^i, y_t^i, z_t^i) \mid i=1, 2, \dots, N_{\text{voxel}}\}$  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。

### 3. 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 Microsoft 社の Kinect センサと、公開されている OpenNI ライブラリを使用した[7].

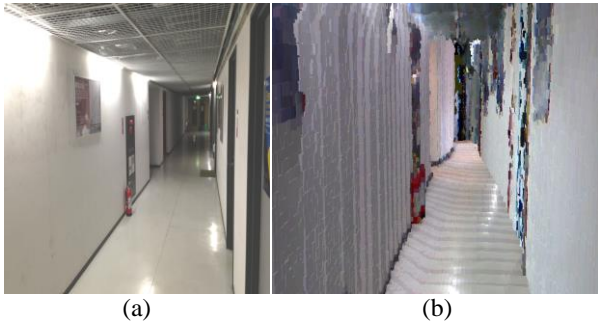


Fig. 6 Experimental result 1: (a) real environment, and (b) built environment model with proposed method.

実験は、Fig. 6(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。消失点の情報を用いず、実験を行った場合は、Fig. 7(a)のように大きな位置ずれのある不正確な地図が作成された。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、Fig. 6(b), Fig. 7(b)に示すように、直線に近い実際の廊下の環境と同じような地図が生成されたことが確認された。Fig. 7(c)は各実験に対し、センサが動いた軌跡を示している。提案手法なしに実験を行った場合は、直線環境にもかかわらず、始点と終点の間の角度誤差が約  $16\text{deg}$  となったが、提案方法での角度誤差は、ほぼ発生しなかった。

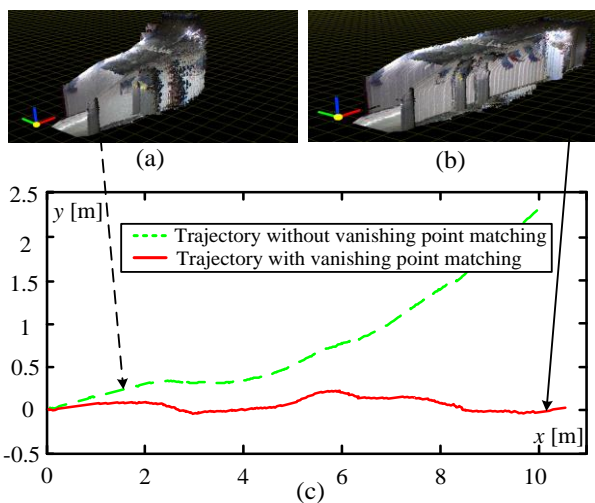


Fig. 7 Experimental result 2: (a) built environment model without vanishing point matching, (b) with vanishing point matching, and (c) comparison of estimated trajectories.

### 4. 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、マッチングを行うことで空間的に位置ずれのない、より正確な三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地図生成が実現された。

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環境モデリング手法について研究を進める計画である。

### 参考文献

- [1] Szymon RusinKiewicz, and MarcLevoy, "Efficient Variants of the ICP Algorithm," *Proceeding of the International Conference on 3D Digital Imaging and Modeling*, pp. 145-152, 2001.
- [2] Peter Henry, Michael Krainin, Evan Herbst, Xiaofeng Ren, and D. 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, pp. 647-663, 2012.
- [3] 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.
- [4] Fovis: (<http://code.google.com/p/fovis/>)
- [5] Greg Welch, and Gary Bishop, "An Introduction to the Kalman Filter," *Proceeding of the ACM SIGGRAPH*, Course 8, 2001.
- [6] Ernest R. Norling, "Perspective Made Easy", BN Publishing, 2008.
- [7] OpenNI: (<http://www.openni.org/>)