

RGB-D カメラを用いた屋内環境マッピングの高速化

北海道大学 ○奥野洋朗, 金井理, 伊達宏昭

要旨

本研究では, RGB-D カメラから得られる連続的なデプス画像と RGB 画像を用いたリアルタイム 3 次元マッピングの手法を開発した. SIFT-GPU による RGB 画像の特徴抽出と Screw 理論を用いたデプス画像の座標変換により従来の手法と比べてマッピングを高速化した.

1. はじめに

近年, Kinect の様な画素と深度情報を同時に取得できる安価な RGB-D カメラの登場により, 屋内環境の 3 次元環境モデルを作る屋内環境マッピングの研究が活発になされている. 例えば, ユーザが RGB-D カメラを持ち歩きながら 3 次元環境モデルをその場で構築可能なモバイルマッピングシステムの研究例がある[1]. しかし, この研究では, マッピングの処理速度が 3~4fps にとどまっており, RGB-D カメラからの連続するフレームの位置合わせが安定的に行えず, また作成された環境モデルの精度検証も行われていないといった問題がある.

そこで本研究では, 精度をそれほど犠牲にせず, より高い処理速度と位置合わせの安定性を実現する RGB-D カメラによる屋内環境マッピングアルゴリズムを開発し, 環境モデルの精度検証を行ったので報告する.

2. 提案する屋内環境マッピング手法

2.1 提案手法の概要

提案するマッピング手法の流れを図 1 に示す. 図 1 はある時刻フレームで取得したターゲットデータ (RGB 画像 I_{RGB}^t , デプス画像 I_{depth}^t) に次のフレームで取得したソースデータ (RGB 画像 I_{RGB}^s , デプス画像 I_{depth}^s) を 2 段階のレジストレーション処理を用いてレジストレーション (位置合わせ) するアルゴリズムである. 以下に各処理の詳細を述べる.

2.2 デプスキャリブレーション

安価な RGB-D カメラから得られるデプス値には無視できない系統的なうねり等の誤差が含まれていることが知られている. そこでこの系統誤差の補正を行うため, [2]の手法に基づき, 取得したデプス画像の各画素のデプス値を, 事前に求めた画素毎のデプス補正テーブルを用いて補正し, 補正デプス画像 ${}^cI_{depth}^t, {}^cI_{depth}^s$ へ変換する.

2.3 3次元点群生成

$I_{RGB}^s, {}^cI_{depth}^s, I_{RGB}^t, {}^cI_{depth}^t$ から, 属性として付加したカメラ座標系におけるデプス画像の各画素の X, Y 座標を式(1)で求め, デプス値を 3 次元点群へ変換する. さらに, デプス画像の各画素 (深さ値) と対応する RGB 画像の各画素の R, G, B 値を点群の各点に属性として付加したターゲット点群 P^t , ソース点群 P^s を生成する.

$$X = \frac{Z}{f_x}(x_{image} - c_x), Y = \frac{Z}{f_y}(y_{image} - c_y) \quad (1)$$

ここで, Z は補正デプス画像内の画素のデプス値, f_x, f_y は x, y 軸方向の画素単位での RGB カメラの焦点距離, x_{image}, y_{image} は注目画素の x, y 座標, c_x, c_y は画像平面とカメラ光軸の交点の画素単位での x, y 座標である.

2.4 画像特徴点抽出 (SIFT-GPU)

次に I_{RGB}^s と I_{RGB}^t から SIFT-GPU を用いてそれぞれの 2 次元の SIFT 特徴点集合 $S^t = \{s_i^t\}, S^s = \{s_j^s\}$ を抽出し, これらに対応している点群内の点より, カメラ座標系におけるタ

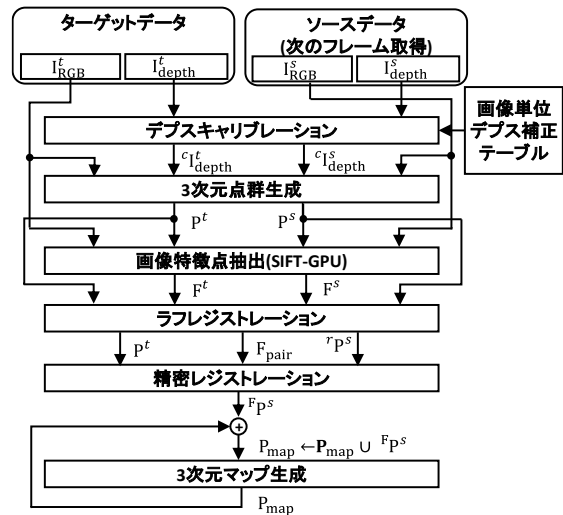


図 1 提案マッピング処理フロー

ーゲットとソースの 3 次元 SIFT 特徴点集合 $F^t = \{f_i^t\}, F^s = \{f_j^s\}$ を取得する. SIFT-GPU[3]は SIFT 特徴点抽出処理を GPU を用いて高速化 (20FPS) したライブラリである. なお予備実験より, DoG 画像生成の標準偏差を表すスケールの大きな SIFT 特徴点は, 時刻が異なる RGB 画像上でも安定して抽出されることが判ったため, 閾値以上のスケールを持つ SIFT 特徴点のみを F^t, F^s とし, レジストレーションに用いる.

2.5 ラフレジストレーション

F^t, F^s を用いて, ソース点群 P^s をターゲット点群 P^t にほぼ一致させる剛体変換 T_r を算出し, P^s, F^s を T_r で剛体変換した rP^s, rF^s と, ターゲット画像とソース画像の SIFT 特徴点ペア集合 F_{pair} を抽出する. まず前処理として F^t の kd 木を構築し, その後以下の手順によりレジストレーションを行う.

Step1: F^t 内からの SIFT 特徴点 3 点のランダム選択

SIFT 特徴点集合 F^t から互いに 3 次元実距離が d_{min} 以上離れた SIFT 特徴点 3 点 $\{f_1^t, f_2^t, f_3^t\}$ をランダムに選択する.

Step2: 対応 SIFT 特徴点候補の選択

選択された各 SIFT 特徴点 f_i^t ごとに, $\|SIFT(f_i^t) - SIFT(f_j^s)\| < \epsilon_s$ を満たす SIFT 特徴点 f_j^s を F^s から全探索し f_i^t の対応 SIFT 特徴点候補 $\{f_j^s\}_i$ として登録する. ここで, $SIFT(f)$ は点 f の 128 次元ベクトルの SIFT 特徴量を表す.

Step3: SIFT 特徴点对応ペアからの剛体変換算出

ある f_i^t に対し $\{f_j^s\}_i$ 内からランダムに 1 点を選択し, 3 特徴点間の対応 $(f_1^t, f_2^t, f_3^t) - (f_{j,1}^s, f_{j,2}^s, f_{j,3}^s)$ よりこの 3 点同士の剛体変換 T_r を算出する.

Step4: ソース画像の SIFT 特徴点の剛体変換

T_r を用いてソース画像の F^s 内の全 SIFT 特徴点を剛体変換し, 変換後のソース画像の特徴点集合 rF^s を算出する.

Step5: SIFT 特徴点ペア取得

F^t に含まれ、SIFT 特徴点 ${}^r f_i^s \in {}^r F^s$ の最近傍点となる SIFT 特徴点 f_j^t を、 F^t に対する kd 木を用いて探索し、それら点間の実距離が閾値 d_{pair} 以下であれば正しい対応ペアとし、ペア集合 F_{pair} へ含める。

Step6: 反復処理

- $|F_{\text{pair}}| \geq N_{\text{pair}}$ (N_{pair} : 閾値) の場合は処理を終了する。
- $|F_{\text{pair}}| \leq N_{\text{pair}}$ 場合は、以下に従う。
 - Step3~の反復数が閾値 N_{3-5} 以下であれば、Step3 へ、以上であれば Step1 へ戻る。
 - Step1~の反復数が閾値 N_{1-5} 以上であれば、ソースフレームを棄却し、次の時刻のソースフレームを取得する。

2.6 精密レジストレーション

最後に、SIFT 特徴点ペア F_{pair} とターゲット点群 P^t 、ラフレジストレーション後のソース点群 ${}^r P^s$ を用いて P^s を P^t に精密に位置合わせする最適剛体変換 T_F を式(3)により算出し、 T_F を用いて剛体変換されたソース点群 ${}^r P^s$ を取得する。

$$T_F = \arg \min_T \left\{ \alpha \left\{ \frac{1}{|F_{\text{pair}}|} \sum_{i \in (i, g(i))} w_i |T({}^r f_i^s) - f_{g(i)}^t|^2 \right\} + (1 - \alpha) \left\{ \frac{1}{|P_{\text{pair}}|} \sum_{j \in (j, h(j))} w_j |(T({}^r P_j^s) - P_{h(j)}^t) \cdot \mathbf{n}_j|^2 \right\} \right\} \quad (3)$$

式(3)の右辺の目的関数の第 1 項は、剛体変換 T を適用した際のソースとターゲットの RGB 画像中の対応 SIFT 特徴点間の重み付き 2 乗平均誤差を、第 2 項は、ソースとターゲット点群の最近傍対応点間の法線方向の重み付き 2 乗平均誤差を表す。また、 $g(i)$ は SIFT 特徴点 i に対応する特徴点番号を、 $h(j)$ は点 j の最近傍点を与える関数で P^t に対する kd 木から高速に探索する。また、 w_i, w_j は各点間距離に対する重みを表し、 α ($0 \leq \alpha \leq 1$) は第 1 項と第 2 項を制御するパラメータである。精密レジストレーションでは式(3)の第 2 項が閾値 ϵ 以下となるまで、最近傍点 $h(j)$ を更新しながら ${}^r f_i^s \leftarrow T({}^r f_i^s)$ 、 ${}^r P_j^s \leftarrow T({}^r P_j^s)$ とし反復する。しかし、式(3)は非線形であるため、[1]のように Levenberg-Marquardt 法のような反復計算で解を求めると、処理速度が低下する問題がある。そこで、本研究では、Screw 座標[4]を用いて式(3)の剛体変換 T を式(4)で線形近似する。

$$T(\mathbf{p}) \approx \mathbf{p} + \mathbf{c} \times \mathbf{p} + \bar{\mathbf{c}} \quad (4)$$

ここで、 $(\mathbf{c}, \bar{\mathbf{c}}) \in R^6$ は剛体変換で 3 次元運動を表す Screw 座標である。式(4)を用いて式(3)を変形すると式(5)となる。

$$(\mathbf{c}, \bar{\mathbf{c}}) = \arg \min_{(\mathbf{c}, \bar{\mathbf{c}})} \left\{ \alpha \left\{ \frac{1}{|F_{\text{pair}}|} \sum_{i \in (i, g(i))} w_i |{}^r f_i^s + \mathbf{c} \times {}^r f_i^s + \bar{\mathbf{c}} - f_{g(i)}^t|^2 \right\} + (1 - \alpha) \left\{ \frac{1}{|P_{\text{pair}}|} \sum_{j \in (j, h(j))} w_j |({}^r P_j^s + \mathbf{c} \times {}^r P_j^s + \bar{\mathbf{c}} - P_{h(j)}^t) \cdot \mathbf{n}_j|^2 \right\} \right\} \quad (5)$$

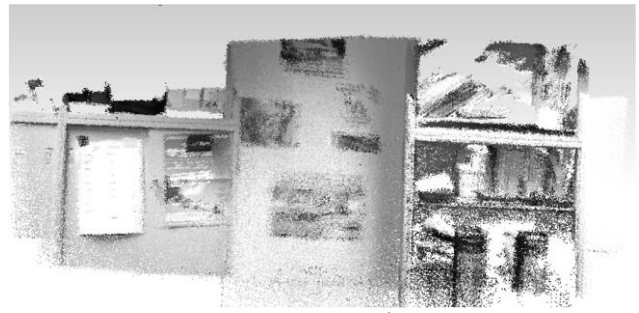
式(5)の第 1 項を $E_1(\mathbf{c}, \bar{\mathbf{c}})$ 、第 2 項を $E_2(\mathbf{c}, \bar{\mathbf{c}})$ をすると、 $E_1(\mathbf{c}, \bar{\mathbf{c}}) + E_2(\mathbf{c}, \bar{\mathbf{c}})$ は、 $\mathbf{c}, \bar{\mathbf{c}}$ の 2 次関数となり、その最小化は無制約 2 次問題に帰着する。従って、式(5)の解は $\mathbf{c}, \bar{\mathbf{c}}$ を変数とする連立 1 次方程式の解として簡単に求めることができ、反復計算を行う必要がなく処理の高速化が図れる。

2.7 3次元マップ生成

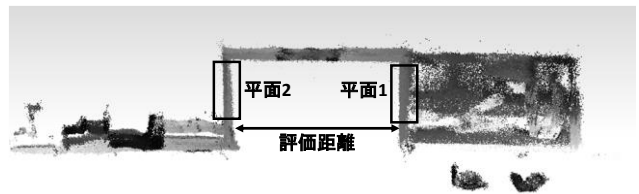
精密レジストレーション後のソース点群 ${}^r P^s$ を、ソースデータ取得前の 3 次元マップ P_{map} に加えることにより取得後の 3 次元マップを逐次生成する。

3. マッピング結果と評価

RGB-D カメラとして Xtion PRO LIVE を用い、提案手法により作成した 3 次元マップを図 2 に示す。図 2 は式(5)における α, w_i, w_j をそれぞれ式(6)のように設定し、連続する 90 フレーム間のレジストレーションを行った研究室内本



(a) 屋内環境マップ(正面)



(b) 屋内環境マップ(上面)

図 2 提案マッピング手法での 3 次元マップ生成結果

表 1 寸法計測精度

	距離平均値 [m]	距離標準偏差 [mm]	平均値 - 真値 [mm]
提案手法	0.802	13.5	5.20
Skannect	0.820	15.0	12.60

棚付近計測点群(9 万点)の点群である。カメラは直線に沿ってほぼ 0.1m/s の定速で 1.2m 移動させた。この場合の処理速度は P^s の点数が既存研究[1]の約 3 分の 1 の場合、約 7-8fps であった。

$$w_i = \frac{1}{1 + ({}^r f_i^s - f_{g(i)}^t / c)^2}, \quad w_j = \frac{1}{1 + ({}^r P_j^s - P_{h(j)}^t / c)^2}, \quad c = 2.4 \quad (6)$$

$$\alpha = 0.5,$$

提案手法と Kinect を用いる市販マッピングソフトの Skannect (version 1.3)[5], open space モードから得られる点群の寸法測定精度を、図 2 中の平行と見られる平面 1, 2 に対し、平面 1 の最小 2 乗フィット平面と平面 2 の各点群間の距離の分布、及び真値(0.807m)との誤差により評価した。表 1 より提案手法は、Skannect と比べ距離標準偏差が 1.50mm、平均値と真値の誤差も 7.40mm と少なく、利用しているカメラ自体の計測性能を考慮すると良好な結果であるといえる。

4. まとめ

本報では、RGB-D カメラによるリアルタイムでの屋内環境マッピングの手法を開発し、生成した 3 次元環境モデルの寸法測定誤差評価を行い、市販システムと比較し、良好なマッピング精度を持つことを示した。また、提案手法の処理速度は 7-8fps である。しかし、一部不安定な症状があるためその改善を図る。

参考文献

- [1] Hao Du et al. "Interactive 3D Modeling of Indoor Environments with a Consumer Depth Camera", Proceedings of the 13th international conference on Ubiquitous computing, pp.75-84, (2011).
- [2] 佐藤雄隆 et al. "RGB-D カメラから得られる Depth データの歪み補正", Vision Engineering Workshop 2013, (2013).
- [3] 「SIFT on GPU (siftgpu)」
<http://cs.unc.edu/~ccwu/siftgpu/>(2014/07/31 アクセス)
- [4] Tae-wan Kim et al. Simultaneous registration of multiple views with markers, Computer-Aided Design Volume 41, Issue 4, pp. 231-239, (2010).
- [5] 「skannect 3D Scanning Software By Occipital」
<http://skannect.occipital.com/>(2014/07/31 アクセス)