

要旨

本研究では BIM の継続的更新を実現させるため、建物内をレーザ計測した点群データを用いて変更情報を抽出し、BIM に反映させることを目的とする。本報では、RANSAC や R-RANSAC により計測点群を BIM 構成要素へ自動的に、かつ高速にレジストレーションし、対応付けの手法を開発した。さらに、変更情報抽出の初期段階として、BIM へ関連付けられない点を差分点として抽出する手法を開発した。

1. はじめに

建築プロジェクトにおけるプロダクトモデルである BIM (Building Information Model) の利用機会が、急速に普及している。BIM の活用は竣工後の建物の維持管理等に有効であるが、そのためには BIM を現況に合わせて継続的に更新し、変更情報を反映していくことが必要である。その解決方法として、現況のレーザ計測点群に基づいて、BIM を更新する案が考えられる。

そこで本研究では、前々報でレーザ計測点群データと BIM データ間に論理的な関連を持たせ統合管理するため、BIM のデータモデル標準である IFC (Industry Foundation Classes) 形式のデータモデルを、レーザ計測点群モデルも統合管理できるように独自に拡張したシステムを開発した [1]。さらに前報 [2] では、RANSAC を用いて多数のソリッドから成る BIM データと、屋内の現況を計測したレーザ計測点群データとのレジストレーションを自動化するアルゴリズムを提案したが、マッチングに多大な処理時間がかかるといった問題が残されていた。

そこで本報では、まず RANSAC 自体の効率化や R-RANSAC 導入による計測点群と BIM 間のレジストレーションの効率化を試みた。さらに、レジストレーション後の点群を BIM の構成要素へ対応付けるとともに、差分点の抽出を行う方法も開発したので報告する。

2. レジストレーションの効率化

2.1 従来手法の問題点とその解決方法

図 1 に BIM の自動更新における本報の位置づけを示す。前報 [2] では、BIM を構成する全ソリッドと、平面領域にセグメンテーションされた計測点群の両者から全ての平面を抽出し、RANSAC により平面对平面でレジストレーションを行うアルゴリズムを開発した (図 1 ①部)。しかし、2.4 節に示すように、RANSAC のフィットネス評価部分に処理時間を要していたため、本報ではその効率化を行った。また、RANSAC に  $T_{d,a}$  テストを加えた R-RANSAC を用い、更なる効率化を図った。さらに、レジストレーション後、BIM 構成要素内の平面に近い計測点群を、BIM 平面に対応付け (図 1 ②部)、対応付けられなかった点を差分点として抽出した (図 1 ③部)。

2.2 平面情報を用いたレジストレーションの利点

IFC に準拠した正式な BIM は図 2 左のように、床、壁、ドアなどが一般に全て異なるソリッドで構成されているモデルであるため、システムやモデリング手法の違いにより厳密に接しているべきソリッド間の境界面が完全に一致していなかったり、干渉したままモデリングされている可能性も多い。このため、各ソリッド内の頂点を抽出したり、ソリッドの面分間の交点を計算する単純な方法では、BIM 内から点群と一致する頂点を必ずしも抽出できないという問題点がある。

一方、BIM が詳細度 LOD200 [3] を持てば、建物内の空間との境界面はほぼモデリングされており、また図 2 右のように計測点群からも平面領域は比較的容易に抽出できる。従って本研究では平面对平面でのレジストレーションを採用した。

2.3 RANSAC による平面-平面レジストレーション

RANSAC (Random Sample Consensus) とは、仮説の検証と反復によるノイズにロバストなモデルのフィッティング手法である [4]。本研究では、BIM に含まれる全てのソリッドの平面集合  $A$  を

$$A = \{a_i | i = 1, \dots, n_a\}, \quad a_i = \langle p_i, b_i \rangle \quad \dots(1)$$

と定義する。ここで  $n_a$  は BIM ソリッドの平面領域数、 $a_i$  は  $i$  番目の平面領域を表し、式 (2) に示すような平面方程式  $p_i$  と、 $a_i$  の境界線ポリライン  $b_i$  をもつ。

$$b_i = (b_{i1}, b_{i2}, \dots, b_{im_i}, b_{i1}), \quad p_i = \langle n_i, d_i \rangle \quad \dots(2)$$

ここで  $b_i$  の各要素  $b_{i1}, b_{i2}, \dots, b_{im_i}, b_{i1}$  は  $b_i$  の各頂点を表し、 $m_i$  は平面  $a_i$  の境界線ポリライン頂点数、 $n_i$  は平面  $a_i$  の外向き単位法線ベクトル、 $d_i$  は  $a_i$  の原点までの距離を意味する。

また、レーザ計測点群データからセグメンテーションにより得られた平面領域集合も同様に

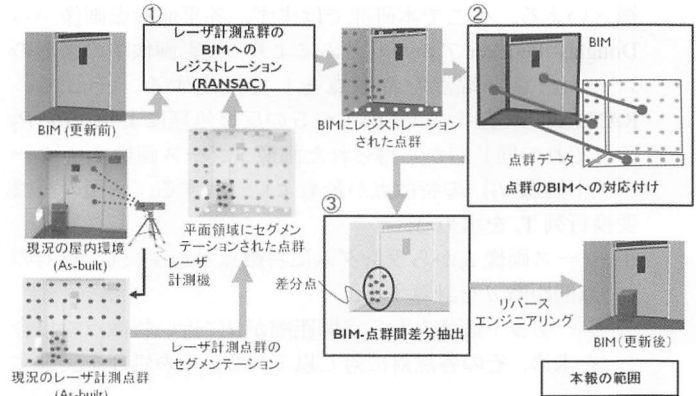


図 1 BIM の自動更新プロセスと本報の位置づけ

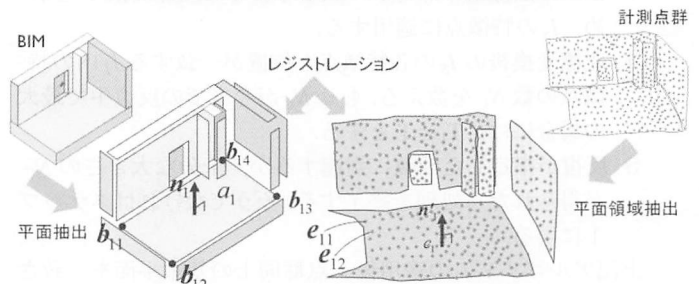


図 2 RANSAC によるレジストレーションの概要

$$C = \{c_j | j = 1, \dots, n_c\}, \quad c_j = \langle r_j, e_j \rangle \quad \dots(3)$$

と定義する。 $n_c$  はレーザ計測の平面領域数であり、平面領域  $c_j$  は式 (4) に示すような平面方程式  $r_j$  と、 $c_j$  の各計測点  $e_j$  をもつ。

$$e_j = (e_{j1}, e_{j2}, \dots, e_{jl_j}), \quad r_j = \langle n'_j, d'_j \rangle \quad \dots(4)$$

ここで  $l_j$  は  $c_j$  の平面領域上の計測点数、 $n'_j$  は平面  $c_j$  の外向き単位法線ベクトル、 $d'_j$  は平面の座標系の原点からの距離である。

点群を BIM にレジストレーションする場合、RANSAC でのモデルは、点群の位置姿勢を移動させる座標変換行列となる。まずこれを求めるため、式 (5) で示されるように、BIM 平面集合  $A$  と点群平面集合  $C$  から、それぞれ 3 面をサンプル  $s$  としてランダムに選択する。ただしサンプルのうち 1 面は、必ず水平面  $a_{zb}$ 、 $c_{zp}$  を含むように制限する。

$$s = [(a_r, a_g, a_{zb}), (c_u, c_v, c_{zp})] \quad \dots(5)$$

サンプル  $s$  に基づき、点群の 3 平面を BIM の 3 平面にマッチングさせるためのモデルパラメータは、式 (6) の 3 次元剛体変換行列  $q$  として表現できる。

$$q = \langle R, t \rangle \quad \dots(6)$$

ここで、 $R$  は  $3 \times 3$  の回転行列、 $t$  は 3 次元並進ベクトルを表す。 $R, t$  はサンプル  $s$  内の 6 平面の法線ベクトルにより、前報 [2] で述べた通り容易に求めることができる。

この  $q$  により点群内の全平面領域を座標変換し、どの程度点群が BIM と一致しているかを、以下のフィットネス  $G$  により評価する。ここで  $G$  は、点群の平面領域が BIM 平面に接近するにつれて大きくなるべきなので、式 (7) のように  $G$  を定義した。

$$G = \sum_{a_i \in A, c'_j \in C'} g(q(a_i, c'_j), \gamma) \quad \dots(7)$$

ここで  $C'$  は  $C$  内の領域を  $q$  により座標変換した領域からなる集合であり、関数  $g(q(a_i, c'_j), \gamma)$  は、 $c'_j$  の点  $e'_j = (e'_{j1}, e'_{j2}, \dots, e'_{jl'_j})$  のうち、BIM 平面  $a_i$  上への投影点のうち、 $a_i$  の境界内部に入っており、かつ投影点と平面間の距離が  $\gamma$  以下のものの点数を与える。なお  $e'_{jk}$  が

複数の BIM 平面においてフィットネスの評価に関与することが原因で、フィットネスが局所的な点により高くなるのを避けるため、点と BIM 平面との対応は 1 対 1 に限定するようにした。この処理を繰り返し、フィットネス  $G$  が最大値となる  $q$  を最適パラメータ  $q^*$ 、点群を BIM にマッチングする最適座標変換として採用する。

2.4 RANSAC の効率化

RANSAC の実行時間  $J_R$  は、一般に式(8)で評価できる[6]。

$$J_R = N(t_M + \bar{t}) \quad \dots(8)$$

ここで、 $t_M$  は 1 回の試行においてモデルパラメータを求めるのに要する時間、 $\bar{t}$  はフィットネス評価を行うのに要する時間である。また  $N$  は試行回数であり、[5]に示す理論式より求めることができる。サンプル時に点群と BIM の領域同志が正しく対応をもつ確率(正対応確率) $\epsilon$ は、BIM 平面全領域のうち、いずれかの点群平面領域と対応をもつ領域数を  $\bar{n}_a$  とすると、 $\epsilon = \bar{n}_a/n_a$  となる。従って、全ての領域同志の正対応確率は  $P_I = \epsilon^3$  と表すことができ、誤対応確率は  $P_o = 1 - \epsilon^3$  となる。

前報[2]では 2.3 節で述べた各 BIM 平面に対する全ての点の包含テストを行っていたため式(8)の  $\bar{t}$  に時間を要していた。その解決のため、まず BIM 平面の表裏に厚さ  $w$  の BoundingBox を作成し、座標変換された各計測点のうち、この Box に含まれる点のみに対し、式(7)の評価を行わせることとした(評価方法 I とする)。さらに効率化を図るため、点群平面に対して BoundingBox を作成し、Box 同士が干渉しているときのみ式(7)の評価を行わせるようにした(評価方法 II とする)。

2.5 R-RANSAC による効率化

R-RANSAC(Randomized RANSAC)[6]とは、各試行においてフィットネス評価の前に  $T_{d,d}$  テストを行い、テストを通過した時のみ全計測点(点数  $n_{pts} = \sum_{l=1}^{n_c} l$ )に対してフィットネス評価を行うものである。 $T_{d,d}$  テストでは、全測定点からランダムに選択した  $d$  点 ( $d \ll n_{pts}$ ) のみに対し、まず式(7)のフィットネス評価を行い、 $d$  点全てが inlier となるとき、テストを通過させる。この R-RANSAC の実行時間  $J_d$  は式(9)のように表される[6]。

$$J_d = \frac{1}{\epsilon^3 \alpha} \{ t_M + P_I(\alpha n_{pts} + (1 - \alpha)\bar{t}_\alpha) + P_o(\beta n_{pts} + (1 - \beta)\bar{t}_\beta) \} \quad \dots(9)$$

ここで  $\alpha$  はサンプルが全て正対応となる試行において、 $\beta$  はサンプルが誤対応となる試行において、 $T_{d,d}$  テストを通過する確率を表す ( $\alpha \gg \beta$ )。また、 $\bar{t}_\alpha, \bar{t}_\beta$  はこれらそれぞれの場合において  $T_{d,d}$  テストが失敗するときのテストに要する時間を表す。 $\alpha = \epsilon^d, \beta = \delta^d$  とおくと、 $\bar{t}_\alpha, \bar{t}_\beta$  は次式のように近似できる。

$$\bar{t}_\alpha \approx \frac{1}{1 - \epsilon}, \quad \bar{t}_\beta \approx \frac{1}{1 - \delta} \quad \dots(10)$$

以上より、式(9)は

$$J_d(d) = \frac{1}{\epsilon^{d+3}} \{ t_M + 1 + (\delta^d + \epsilon^{d+3})n_{pts} \} \quad \dots(11)$$

と  $d$  の関数で表すことができる。 $J_d(d)$  はただ 1 つの極値を持ち、 $d \rightarrow \pm\infty$  のとき  $J_d(d) \rightarrow \infty$  となるので、 $\partial J_d(d)/\partial d = 0$  となる  $d$  を求めることで  $J_d$  の最小値を与える点数  $d$  を求めることができる。

3. 点群の BIM 構成要素への対応付け・差分点抽出

前述の最適レジストレーション時にフィットネス評価を行った際、平面領域上の点と BIM 平面は 1 対 1 対応をもつため、点群の中で BIM 平面に接近しており、包含されたもののみ、BIM 平面に点が一致していると考え、その平面が属する BIM 構成要素への対応付けを行う。また点群において、平面領域として抽出されなかった点に対してもこの処理を適用し、BIM 構成要素への対応付けを行う。一方、BIM 平面の近くにない点は、元々の BIM からの変更が行われた差分点として抽出できる。

4. 実験結果

提案手法の有効性を簡単な例題で確認した。独自開発した計測点群モデル・BIM 統合管理システム[1]内に本レジストレーション処理を組み込み、実装を行った。

例題として、本学情報科学研究科棟 4 階廊下部分(約 11m×7m を対象に、図 3(a)のように、BIM を建築用 CAD(MicroStation)でモデリングし、平面抽出を行った。BIM ソリッドの平面領域数  $n_a$  は 4598 である。

初めにレジストレーションの効率化を確認するため、同一環境の点群(約 27.8 万点)を地上型レーザスキャナ(Focus3D)で計測し、

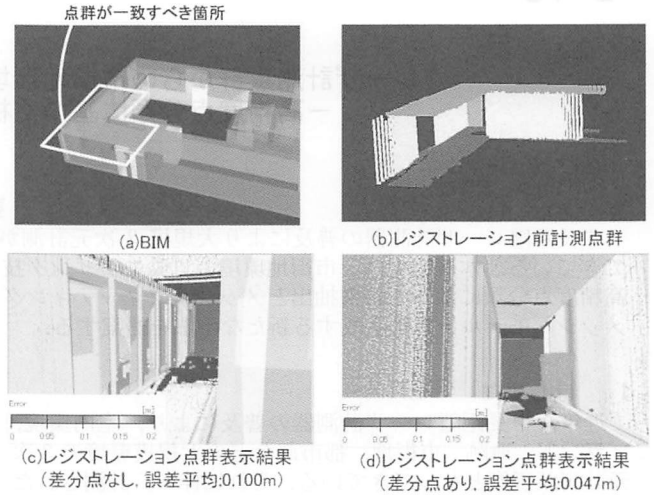


図 3 入力データと出力結果

表1 レジストレーション実行時間比較(試行回数100回あたり)

従来評価手法[2]	RANSAC		R-RANSAC
	評価手法 I	評価手法 II	評価手法 II
199[s]	12.15[s]	11.59[s]	1.95[s]

平面領域集合を得るため、図 3(b)のように PCL(Point Cloud Library)の関数を用いてセグメンテーション処理を行った。このとき、点群の平面領域数  $n_c$  は 11 であった。この組合せに対し、RANSAC 試行回数を理論式[5]より求めた。成功率を 90% と仮定すると、試行回数は約 533 万回と試算される。しかし実用上の観点から、試行回数を下げるため、点群は BIM の大きい平面と対応を持つ確率が高いことを利用し、BIM 平面面積  $5m^2$  以上のもののみをサンプルすることで、 $n_a=139$  と減少させた。これより試行回数は 5415 回に減少した。2.4 節の評価方法 I, II, 2.5 節の R-RANSAC における実行時間を比較した結果が表 1 である。なお、R-RANSAC では、式(11)より  $d = 1$  とした。これより、レジストレーションに要する時間は、前報[2]に比べ約 100 分の 1 に短縮された。また図 3(c)に示すように、上述の試行回数内で、妥当な座標変換行列が出力されたことも分かる。

次に差分点抽出のため、同一環境内の一部に元々 BIM には含まれない椅子を配置した環境の点群(約 45.7 万点)を計測した。点群の平面領域数  $n_c$  は 26 となった。この入力点群にも、上述と同様の試算で求めた試行回数で R-RANSAC( $d = 1$ )によりレジストレーション処理を行ったが、妥当な座標変換行列は導出できなかった。これは BIM との差分点領域がある場合、 $n_d$  を点群内の差分点の推定される領域数とすると、正対応確率は  $(\bar{n}_d/n_a)^3 \{ (n_c - n_d)/n_c \}^3$  に減少し、前述の試算のままでは試行回数が不足するためである。そこで、試行回数を 10 万回に増やしたところ、妥当な座標変換行列が出力された。実行時間は約 34 分であった。図 3(d)は最適レジストレーション後の点群を、BIM 平面とのエラーマップによって示したものである。この結果より、椅子が正しく差分点として抽出されていることが分かる。

今後の課題として、差分点領域数の見積、測定点群の微小領域の排除、RANSAC 後の ICP による高精度なマッチング、対応付け誤差閾値  $\gamma$  の妥当な推定が挙げられる。

参考文献

- 山岡晋也 他 “BIM 自動更新のための計測点群モデル・BIM 統合管理システムの構築”, 2012 年度精密工学会北海道支部講演論文集, pp.25-26 (2012)
- 山岡晋也 他 “RANSAC を用いた屋内レーザ計測点群と BIM とのレジストレーション”, 2013 年度精密工学会春季大会学術講演会講演論文集, pp.879-880 (2013)
- 足達嘉信, “BIM の概要と IFC について”, [http://blchiba2012.up.secsaa.net/benkyokai/BIM\\_IFC\\_intro\\_20120806.pdf](http://blchiba2012.up.secsaa.net/benkyokai/BIM_IFC_intro_20120806.pdf) (2012)
- M. Fischler et al. “Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography”, Comm. of the ACM, 24(6), pp.381-395, (1981)
- Konstantinos G. Derpanis, “Overview of the RANSAC Algorithm”, Computer Science and Engineering, 1.2, (2010)
- O.Chum et al. “Randomized RANSAC with Td,d test”, Image and Vision Computing (IVCNZ), pp. 448-457 (2002)