

改良型 RANSAC マップマッチングに基づく辞書式地図圧縮

長坂朋実[†] 田中完爾[†] 橋本拓[†] 角谷崇徳[†] 早水大智[†][†] 福井大学工学部 〒910-8507 福井市文京 3-9-1

E-mail: †{nagasaka,hashimoto,kadoya,hayamizu}@rc.his.u-fukui.ac.jp, ††tnkknj@u-fukui.ac.jp

あらまし 本論文は、点群地図が与えられるとき、これを高速かつ正確にデータ圧縮する、地図圧縮問題を考える。本問題は、自律移動ロボットの分野において、大規模地図生成、ロボット・センサネットワークやコルモゴロフ複雑性など、様々な応用を持つ。本論文の主要な貢献は、辞書式地図圧縮と呼ぶ、新しい地図圧縮手法を提案し、RANSAC マップマッチングに基づく効率的な地図圧縮手法を提案する点にある。実験により、地図圧縮率および計算効率の観点から、提案方法の有効性を検証する。

キーワード 自律移動ロボット, 地図圧縮, 辞書式圧縮, マップマッチング

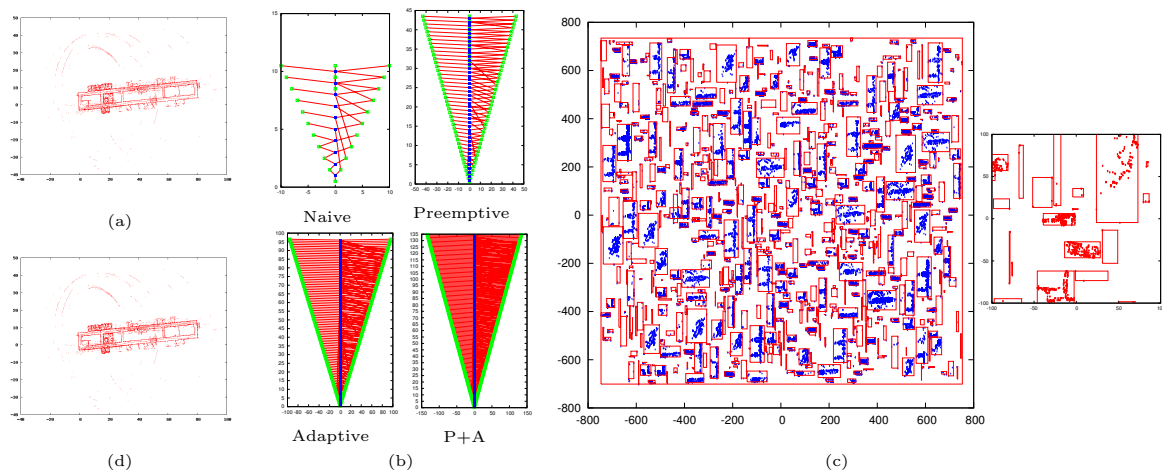


図1 Map compression task.

(a) An original map. (b) A graphical representation of the map compression task. "Naive", "Preemptive", "Adaptive" and "P+A" are four different map-matching algorithms. Each green and blue node respectively indicate a map and a transformation. The bottom node at (0,0) corresponds to the original input map. As a result of a map-matching process, each map is explained by two or three smaller submaps and a transformation. (c) A collection of maps learned during an entire task. (The details are better seen by zooming on a computer screen.) (d) The decompressed map.

1. はじめに

本論文は、点群地図が与えられるとき、これを高速かつ正確にデータ圧縮する、地図圧縮問題 [1]-[3] を考える。近年、SLAM 技術の進展により、自律移動ロボットのセンサデータ群をもとに大規模環境の地図をリアルタイムに生成することが可能になってきた (e.g. "大規模 SLAM [4]"). また、この技術を基盤として、不特定多数のロボットが互いの地図を共有利用する自律分散ネットワークの研究がなされている (e.g. "ロボットセンサネットワーク [5]"). 以上のような、容量制約の厳しい応用において、高速・正確な地図圧縮技術が望まれる (e.g. "圧縮 SLAM [3]"). もう一つの応用は、汎用的な類似度尺度としてのコルモゴロフ複雑性である ("圧縮距離 [6]"). 汎用類似度には、情報検索 [7], 意味ラベリング [8], 類似クラスタリング [9] のように、様々なアプリケーション

がある。以上を踏まえ、辞書式地図圧縮と呼ぶ、汎用的な、新しい地図圧縮方式を提案することを本研究の目的とする。

データ圧縮技術において、辞書式圧縮は、標準的な圧縮方式の一つである [10]. その基本的なアイデアは、データ中に繰り返し出現するパターン (繰り返しパターン) を辞書化・符号化することにある。たとえば、テキスト圧縮は、テキスト中に繰り返し出現するパターン (単語) を辞書化・符号化することで、圧縮を実現する。繰り返しパターンの辞書は、所与とするケース、処理中に学習するケースの 2 通りの方法がある。辞書が与えられるとき、入力データ中のパターンは、辞書中のパターンに置き換えられる。以上のような、辞書学習およびパターン照合が、データ圧縮処理の基本的な構成要素となる。

本研究では、RANSAC マップマッチングと呼ぶ、地図

データの標準的なパターン照合方法を応用する [11]. 一般に, マップマッチング処理は, 地図対を入力とし, 一方を他方へ最も大きく重ね合わせるような変換 (回転・並進) を探索することを目的とする. そのために, RANSAC マップマッチングは, 変換の仮説をランダムに複数生成し, 各々の仮説の下での地図対の重なり度合いを評価し, 最も評価値の高い仮説を探索する [12]. これまでに著者らは, この RANSAC マップマッチングに関して, いくつかの改善・拡張を行ってきた. 文献 [13] では, preemption スキームと呼ぶ, RANSAC アルゴリズムの効率化手法を導入し, 地図生成タスクに有効な, 逐次型 RANSAC と呼ぶ拡張手法を提案した. また, 文献 [11] では, LSH 視覚検索技術を利用して, 逐次性を保ったまま, 逐次型 RANSAC を高速化する手法を提案した. 文献 [2] では, 問題規模に応じて, マップマッチングに使用する計算資源の量を制御する方法を提案した. 本論文では, 様々なマップマッチング手法を実装し, radish データセットを用いた地図圧縮実験により, 提案方式の圧縮性能 (圧縮速度, 圧縮率) を検証する.

関連研究として, ロボット分野の様々な認識タスク (例: 自己位置推定, 地図生成) にマップマッチングを利用した事例 [11] や, コンピュータビジョン分野の一般的なパターン照合問題に RANSAC アルゴリズムを利用した事例 [14] [15] [16], 様々な認識タスク (画像認識 [17], 画像補完 [18], 超解像度化 [19]) に繰返しパターン (自然シーンの自己相似性, 同一対象物の重複観測) を利用した事例, データ圧縮分野の画像圧縮・動画像圧縮などの事例を挙げることができる [20]. 以上のような研究事例を踏まえた上で, 本研究は, 地図圧縮問題という新しいアプローチとして位置付けられる. 他の多くのデータ圧縮問題と異なり, 地図データには, 大小様々な繰返しパターン (例: 建築物, 道路標識) が混在するという特色がある. 疎符号化など, 繰返しパターンを辞書化・符号化する方法について, 文献 [3] で検討している. マップマッチングは, ロボット分野の様々な認識タスクに用いられており, それらのタスクと本タスクとの間で, データ構造やアルゴリズムを共有することにより, ロボットシステム全体としての計算資源を節減することができる. 辞書式圧縮方式は, 入出力のデータ形式が同一であり, 原理的に, 他の様々な圧縮手法と直交・相補的な関係にある.

2. 地図圧縮アプローチ

マップマッチング処理は, 二枚の地図 M_i, M_j を入力とし, 一方を他方へ最も大きく重ねるような変換 ψ (回転・並進) を探索することを目的とする. この重なり領域 S_{ji}^ψ を, 繰返しパターンと呼ぶ. マップマッチング処理の出力は, 変換 ψ , 重なり領域 S_{ji}^ψ , 入力地図中の非重なり領域 R_i^ψ, R_j^ψ となる. いま, ある表現のデータサイズを $Size(\cdot)$ のように表すことにする. 通常, 地図のデータサイズと比べて変換 ψ のデータサイズは無視できるほ

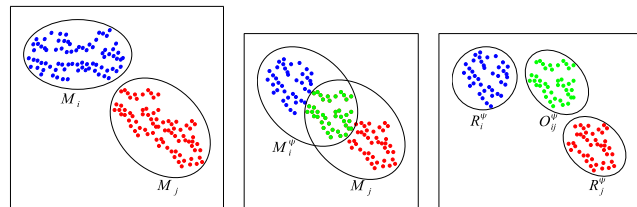


図2 Map compression process (left and middle: before and after a map matching process. right: output maps.

ど小さい. その場合,

$$Size(\langle \psi, S_{ij}^\psi, R_i^\psi, R_j^\psi \rangle) \quad (1)$$

$$\simeq Size(\langle S_{ij}^\psi, R_i^\psi, R_j^\psi \rangle) \quad (2)$$

$$< Size(\langle R_i^\psi, S_{ij}^\psi, R_j^\psi, S_{ji}^\psi \rangle) \quad (2)$$

$$= Size(\langle M_i, M_j \rangle). \quad (3)$$

が成り立つので, 出力 $\langle \psi, S_{ji}^\psi, R_i^\psi, R_j^\psi \rangle$ は, 入力 $\langle M_i, M_j \rangle$ の圧縮表現とみなすことができる.

便宜上, 異なる二枚の地図 $M_i, M_j (i \neq j)$ を入力とするマップマッチング処理, および, 同一の地図 M_i, M_i を入力とするマップマッチング処理を, 互いに区別し, それぞれ, 相互マッチングおよび自己マッチングと呼ぶ. 後者の場合, 零回転・零並進の自明な解は, 解候補に含めない.

地図には, 大小様々な繰返しパターン (例: 建築物, 道路標識) が混在するという特色があり, 一度のマップマッチング処理では, それらのパターンを網羅的に検出することができない. そこで, 地図圧縮タスクでは, あるマップマッチング処理の出力 (パターン) に対し, 再びマップマッチング処理を施す, という再帰的处理を行う. そのために, パターンを記憶するための辞書を導入し, 各々のステップで検出されたパターンを次々と辞書に追加していく. 具体的に, 本アプローチでは, まず最初に, 入力地図に対して, 自己マッチングを行い, その結果として得られる2つの地図を要素として辞書を初期化する. そして, 計算時間が尽きるまで, 辞書中のパターンを入力としてマップマッチング処理を実行し, その結果のパターンを辞書へ追加する, という手続きを繰り返すことで, 辞書を逐次的に構築していく. 図1(c)に, ある圧縮タスクの間に学習されたすべてのパターンを示す. 本アプローチの最終的な出力は, 地図集合, および, 変換の列となる.

展開アルゴリズムは, 地図集合および変換の列 (Fig. 2a,b) を入力とし, 逆変換を逆順に実行する手続きであり, 高速に実行できる. なお, この展開手続きは, 地図毎に独立に行うこともできる. また, 変換の表現方法として, 同次変換行列を用いることで, 変換列の長さによらず, その結合変換を一定の空間コストで記憶することができる.

2.1 RANSAC マップマッチング

RANSAC マップマッチングは, 地図対 M_i, M_j を

入力とし、 M_i を M_j に最も大きく重ね合わせる変換（回転・並進）を探索することを目的とする。基本的な RANSAC アルゴリズムを、以下に説明する。いま、 n_i を地図 M_i 中の点数とする。 $U_i = \{x_l\}_{l=1}^{n_i}$ を点集合とする。 $f_m : S \rightarrow p$ を、 U の部分集合をもとに仮説 p を算出するようなモデル関数とする。 $\rho(p, x)$ をスコア関数とする。この関数は、点 x および仮説 p を入力とし、 x がモデル p に対し整合（インライア）ならば、1 を返し、それ以外ならば、0 を返す。実装では、入力 x （地図 M_i 上の点）がインライアかどうかを判定するための条件として、 $|x - x'| < d_{th}$ を満たすような点 x' が地図 M_j 上に存在するかどうか、という条件を用いる。ただし、 d_{th} は、事前に決定しておく閾値パラメータである。RANSAC の処理は、まず、スコア変数 C^* を 0 へ初期化し、以下を一定回数繰り返す。

(1) 部分集合 $S \subset U$ をランダムに選択し、仮説 $p = f_m(S)$ を生成する。

(2) スコアを $C = \sum_{x \in U} \rho(p, x)$ のように算出する。

(3) もし $C > C^*$ ならば、 $C^* \leftarrow C$ 、 $p^* \leftarrow p$ のように最良仮説およびスコア変数を更新する。

閾値パラメータ d_{th} を、RANSAC マップマッチングの空間解像度と呼ぶ。このパラメータにより、地図圧縮タスクのコンパクト性と精度の間のトレードオフを制御することができる。3. の実験では、3 つの異なる d_{th} の値 0.1, 0.2, 0.3 に対し、圧縮性能を調査する。

2.2 preemption スキーム

preemption スキームは、RANSAC アルゴリズムの効率性を向上させることを目的として考案された拡張スキームである。一般に、RANSAC では、評価する点仮説ペア $\{(p, x)\}$ の数に比例して、処理時間がかかる。2.1 で述べた基本的な RANSAC は、各ペア (p, x) を評価する順序は事前に定められている。pRANSAC では、この制約を取り除き、ペア (p, x) の評価順序を動的に計画する。本論文では、その計画手法として、幅優先ルール [21] と呼ぶ手法を用いる。本手法は、次のような望ましい性質を備えている。第一に、ユーザ側で計算時間の上限を制御することができる。第二に、競争原理により、仮説を相対的に比較する。このため、絶対的な評価指標が与えられることを前提としない。第三に、数値解析により、安定性が明らかになっている [21]。本手法の基本的なアイデアは、競争原理に基づいて、仮説数を段階的に減少させていくことにある。ある時点で生き残っている仮説をアクティブ仮説と呼ぶ。その上で、各時点 i でのアクティブ仮説数の上限を、ユーザ定義の単調減少関数 $f_b(i)$ により制御する。初期化ステージ $i = 0$ において、本手法は、入力の点列 $1, \dots, n$ および仮説 ID 列 $1, \dots, m$ をランダムに置換し、全ての仮説 $j (1 \leq j \leq m)$ についてスコア変数を $C_j = 0$ のように設定する。そして、以下のステップ $i (\geq 1)$ を、アクティブ仮説が一つになるまで

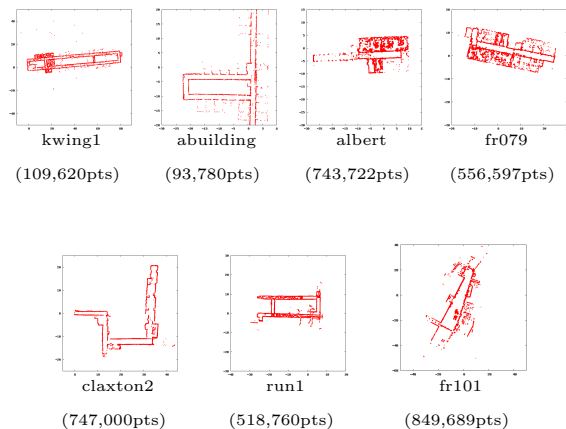


図 3 Datasets.

繰り返す。

(1) 各々のアクティブ仮説 $j (1 \leq j \leq f_b(i-1))$ について、スコアを $C_j \leftarrow C_j + c(p_j, x_i)$ のように算出する。

(2) 仮説 ID 列 $1, \dots, f_b(i-1)$ を、並び替える。この並び替えは、累積スコア C_j に基づいて、区間 $1, \dots, f_b(i)$ の部分が最良の $f_b(i)$ 個のアクティブ仮説を含む、という条件を満足するように行う。

関数 f_b は、

$$f_b(i) = \lfloor m2^{-\lfloor \frac{i}{b} \rfloor} \rfloor, \quad (4)$$

という形をしている。ここで、 $\lfloor \cdot \rfloor$ は、切下げ関数であり、 b はブロックサイズと呼ばれる定数である。 b 回繰り返しのたびに、仮説集合のサイズは半分になるので、アルゴリズムの時間コストは $O(bm)$ となる。従って、ユーザ側でパラメータ m を増減させることにより、計算時間の上限を制御することができる。

2.3 適応サンプルサイズ

地図圧縮タスクにおいて、各々のマップマッチング処理が扱う入力地図は、大小様々であり、入力地図が大きいほど、より多くの計算資源（時間資源）が必要になる。この計算資源は、前述のように、仮説数 m により制御することができる。本節では、この仮説数（サンプルサイズ） m を適応的に制御する適応サンプルサイズと呼ぶ方法を示す。

マップマッチングは、仮説群の中から適合仮説（インライア仮説）を探索する処理であり、その時間計算量は、全体の仮説数をインライア仮説数で除算したものである、とモデル化することができる。インライア仮説数は、入力の地図対 M_i, M_j によらず、一定である、とモデル化することができる。全体の仮説数は、入力地図 M_i, M_j の点数を、それぞれ n_i, n_j とすると、 $O(n_i n_j)$ である。以上の考察をもとに、パラメータ m を、地図サイズ n_i, n_j に応じて、下式のように制御する。

$$m_{ij} = \gamma n_i n_j. \quad (5)$$

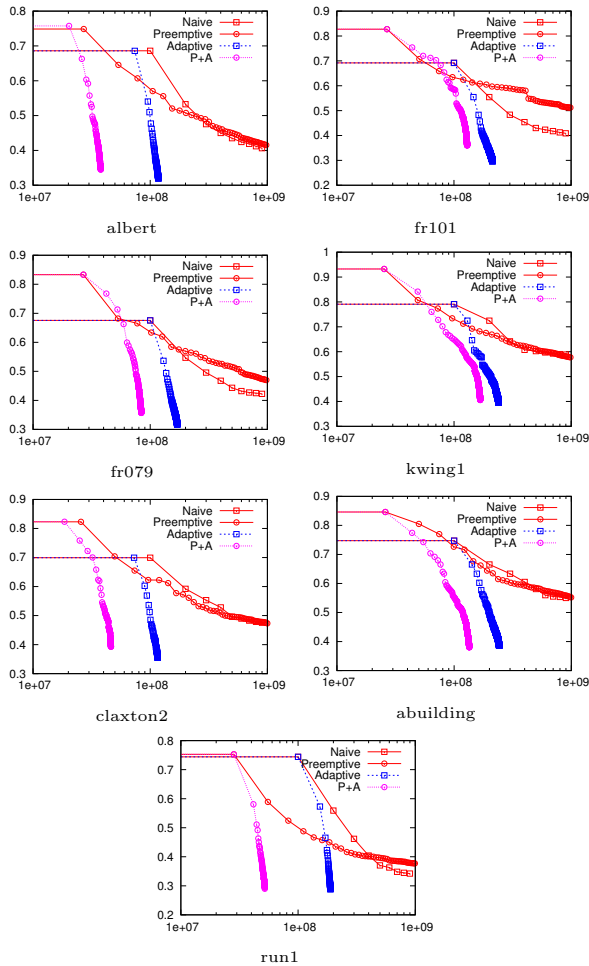


図 4 Compression performance (res: 0.1m).

ただし, γ は, 事前に学習しておく係数である. 以上の方法を, 適応サンプルサイズと呼ぶ.

3. 地図圧縮実験

提案法の有効性を検証するために, 地図圧縮実験を行った. ここでは, 圧縮対象の地図として, radish [22] データセットを用いる. radish データセットは, 屋内移動ロボットにより収集したオドメトリおよびレーザー計測の時系列センサデータからなる. それらの中から, "albert", "fr101", "fr079", "kwing1", "claxton2", "abuilding", "run1" の 7 つのデータセットを使用する. 各々のデータセットをもとに, スキャンマッチングと呼ばれる標準的な地図生成アルゴリズムを用いて, 図 3 に示すような, 7 つの異なる地図を生成した. これらを, 本実験の圧縮対象地図として用いる. 性能指標として, 圧縮率を用いる. これは, 圧縮前後のデータサイズの比をとったものである. 比較手法として, "Naive", "Preemptive", "Adaptive", "P+A" の 4 つの方法を用いる. "Naive" は, ナイブな RANSAC マップマッチング手法である. "Preemptive" は, pRANSAC マップマッチング手法である. "Adaptive" は, ナイブな RANSAC マップマッチングと適応サンプルサイズを組合せたもの

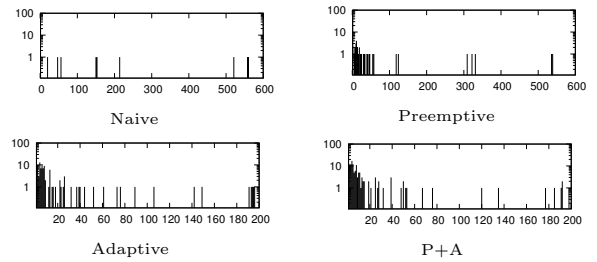


図 5 Histogram of map size in terms of #datapoints (albert dataset).

である. "P+A" は, pRANSAC マップマッチングと適応サンプルサイズを組合せたものである. 各々のマップマッチング処理では, その時点において, 最も大きなサイズの地図を入力として選択するものとした. 特に断りがないかぎり, RANSAC マップマッチングの空間解像度は 0.1m とした. 仮説数の上限を 1×10^6 とし, 評価に用いる点仮説ペア数の上限を, 仮説当たり 100 に設定した. 適応サンプルサイズの係数 γ は 0.01 とした.

図 4 に結果を示す. 図中の横軸は, 点仮説ペアの数を表す. 縦軸は, 圧縮率を表す. 図より, "Adaptive" および "P+A" の 2 つの手法が, 他よりも高い性能を示していることが分かる. これらの手法は, 2×10^8 未満のコストで, 0.4 よりも良い圧縮率に到達している. 一方, "Naive" および "Preemptive" は, より低い性能となった. これらの手法では, マップマッチング処理当りのコストが高く, 実用的な計算時間内では, 少ない回数のマップマッチング処理しか実行できなかった. "P+A" と "Adaptive" を比べると, 前者の方が, より良い性能となっていることが分かる. この結果は, pRANSAC および適応サンプルサイズの両方の戦略が, 高速化に有効であったことを示している.

各々の手法について, 各々のマップマッチング処理の入力地図のサイズの分布を調べたものを, 図 5 に示す. 図より, "Adaptive" や "P+A" の場合に, より多くの地図が生成されている. また, 地図サイズの分布は一様ではなく, より小さな地図サイズに偏った分布となっている.

図 6 に, より低い空間解像度 0.2m および 0.3m を用いて地図圧縮タスクを実施した結果を示す. 空間解像度 0.1m の場合と比べて, 10-20 倍の圧縮速度となっていることが分かる.

先に述べたように, マップマッチング処理の時間コストは, 点仮説ペアの数に, ほぼ比例する. その比例定数は, 点仮説ペア当たり, 2.2×10^{-7} sec であった. たとえば, 1×10^6 個の点仮説ペアを評価するコストは, 0.22 sec となる. ただし, 計算機は, メモリ 8GB, キャッシュ 6.14, CPU 2GHz, Intel (R)Core (TM) 2Duo であった.

図 1(b) は, 各々のマップマッチング処理において, 入力・出力に用いられた地図を, グラフにより可視化したものである. 図中で, 下端中央 (0,0) のノードが地図圧

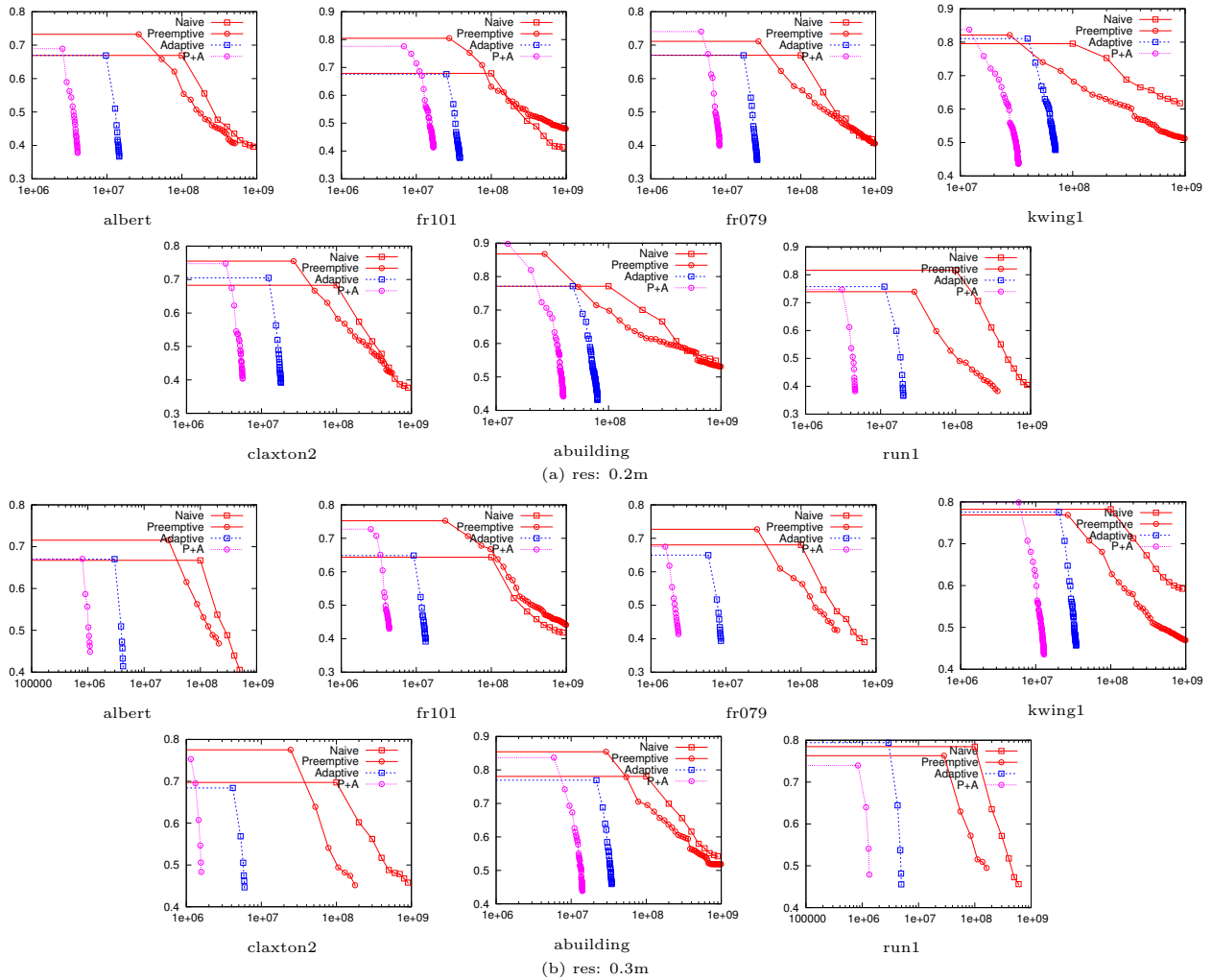


図6 Performance for coarse resolutions.

縮タスクの入力の地図を指す. y 軸上にある各々のノードは, 各々のマップマッチング処理を表す. 一方, y 軸上にないノードは, 各々のマップマッチング処理の入力・出力として用いられた地図を表す. 図より, "Adaptive" および "P+A" は, 他の手法と比べて, 非常に大きな数の地図を生成している. これは, マップマッチング処理当りの平均的な計算コストが比較的小さく, 多数のマップマッチング処理を実行できたことによる.

4. むすび

本論文では, 地図圧縮問題を定式化し, 効率的な辞書式地図圧縮手法を提案した. まず, RANSAC マップマッチングに基づく地図圧縮方法を導入し, さらに, preemption スキームおよび適応サンプルサイズによる効率化を図った. また, 地図圧縮実験において, 地図圧縮率や計算効率を調査し, 提案手法の有効性を確認した. 提案した辞書式圧縮のアプローチは, 他の多くの圧縮手法・符号化手法に対し直交・相補的な関係にある. これらの直交する手法を, 提案方式と組み合わせることにより, 今後, さらなる圧縮効率の向上を期待できる.

5. 謝 辞

本研究は, H23-25 文部科学省科学研究費補助金「移動ロボットによる軽量・精密なリアルタイム圧縮地図生成」の一環として実施した. 本研究の一部は, 倉田財団倉田奨励金の支援を受けた.

文 献

- [1] Ikeda Kouichirou and Tanaka Kanji. Visual robot localization using compact binary landmarks. *Proc. IEEE International Conference on Robotics and Automation*, 2010.
- [2] Nagasaka Tomoni and Tanaka Kanji. Dictionary-based map compression using modified ransac map-matching. *Proc. IEEE Int. Conf ROBOTICS*, 2010. <http://rc.his.u-fukui.ac.jp/DMC.pdf>.
- [3] Nagasaka Tomoni and Tanaka Kanji. Dictionary-based map compression for sparse feature maps. *Proc. IEEE Int. Conf Robotics and Automation*, 2011. <http://rc.his.u-fukui.ac.jp/DMSC.pdf>.
- [4] Viorela Ila Kai Ni Frank Dellaert, Justin Carlson and Charles E. Thorpe. Subgraph-preconditioned conjugate gradients for large scale slam. *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 2010.
- [5] Arthur Martens Rene Iser and Friedrich M. Wahl. Localization of mobile robots using incremental local

- maps. *Proc. IEEE Int. Conf. Robotics and Automation*, 2010.
- [6] Manuel Cebrián, Manuel Alfonseca, and Alfonso Ortega. The normalized compression distance is resistant to noise. *IEEE Transactions on Information Theory*, 53(5):1895–1900, 2007.
- [7] Sivic J. and Zisserman A. Video google: a text retrieval approach to object matching in videos. *Proc. IEEE International Conference on Computer Vision*, pages 1470–1477, 2003.
- [8] Cyrill Stachniss, Oscar Martinez Mozos, Axel Rottmann, and Wolfram Burgard. Semantic labeling of places. In *International Symposium of Robotics Research (ISRR)*, 2005.
- [9] Minsu Cho et al. Unsupervised detection and segmentation of identical objects. *Proc. IEEE Int. Conf. CVPR*, 2010.
- [10] Willis Lang et al. Dictionary-based compression for long time-series similarity. *IEEE Trans. Knowl. Data Eng.*, 22(11), 2010.
- [11] Kenichi Saeki, Kanji Tanaka, and Takeshi Ueda. Lsh-ransac: An incremental scheme for scalable localization. *Proc. IEEE Int. Conf. Robotics and Automation*, pages 3523–3530, 2009.
- [12] Neira J., Tardos J.D., and Castellanos J.A. Linear time vehicle relocation in slam. *Proc. IEEE International Conference on Robotics and Automation*, 1:427– 433, 2003.
- [13] Tanaka K and Kondo E. Incremental ransac for online vehicle relocation in large dynamic environments. *Proc. IEEE Int. Conf. Robotics and Automation*, pages 1025–1030, 2006.
- [14] B. Matei, Y. Shan, H.S. Sawhney, Y. Tan, R. Kumar, D. Huber, and M. Hebert. Rapid object indexing using locality sensitive hashing and joint 3d-signature space estimation. *Trans. IEEE PAMI*, 28(7):1111–1126, 2006.
- [15] Goshen Liran and Shimshoni Ilan. Balanced exploration and exploitation model search for efficient epipolar geometry estimation. *Trans. IEEE PAMI*, 30:1230–1242, 2008.
- [16] Chum O. and Matas J. Optimal randomized ransac. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30:1472–1482, 2008.
- [17] Grant Schindler, Panchapagesan Krishnamurthy, Roberto Lubliner, Yanxi Liu, and Frank Dellaert. Detecting and matching repeated patterns for automatic geo-tagging in urban environments. *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, 2008.
- [18] James Hays and Alexei A Efros. Scene completion using millions of photographs. *Proc. ACM Transactions on Graphics*, 26(3), 2007.
- [19] Daniel Glasner, Shai Bagon, and Michal Irani. Super-resolution from a single image. *Proc. Int. Conf. Computer Vision*, 2009.
- [20] Wee Meng Woon, Anthony Tung Shuen Ho, Tao Yu, Siu Chung Tam, and Siong Chai. Achieving high data compression of self-similar satellite images using fractal. *Proc. Geoscience and Remote Sensing Symposium*, 2:609–611, 2000.
- [21] David Nister. Preemptive ransac for live structure and motion estimation. *Machine Vision and Applications*, 16:321 – 329, 2005.
- [22] Andrew Howard and Nicholas Roy. The robotics data set repository (radish). 2003.