福井大学 大学院工学研究科 研究報告 第 60 巻 2012 年 3 月
Mem. Grad. Eng. Univ. Fukui, Vol. 60 (March 2012)



# 圧縮 Gist ランドマークの研究
## 時系列圧縮 Gist に基づくモンテカルロ自己位置推定

近藤 賢佑* 田中 完爾* 池田 剛一郎*

# Study on Compressed Gist Landmark
## Monte Carlo Localization Using Compressed Gist Sequence

Kensuke KONDO* and Kanji TANAKA* and Kouichirou IKEDA*

(Received January 23, 2012)

This paper is concerned with the problem of mobile robot localization using a novel compact representation of visual landmarks. With recent progress in lifelong map-learning as well as in information sharing networks, compact representation of a large-size landmark database has become crucial. In this paper, we propose a compact binary code (e.g. 32bit code) landmark representation by employing a compressed Gist scene descriptor from web-scale image retrieval. We show how well such a binary representation achieves compactness of a landmark database while preserving efficiency of the localization system. Experiments using a high-speed car-like mobile robot evaluate effectiveness of the presented techniques in terms of cost-performance, semantic gap, saliency evaluation using the presented techniques as well as challenge to further reduce the resources (#bits) per landmark.

***Key words*** : monte carlo localization, visual landmark, compressed Gist

## 1. はじめに

ランドマークに基づく自己位置推定問題は，移動ロボット学の最も基礎的な問題の一つである．本問題は，作業環境の視覚特徴（ランドマーク）配置を記した地図を所与とし，ロボットが自己位置を一意に推定することを目的とする．そのために，ロボットは，移動経路上の各地点において，視覚特徴を認識し，これと類似する特徴を地図中から検索することで，次第に自己位置を絞り込んでいく．従来より，この認識・検索に有効な，様々なランドマーク（例：SIFT[1]，ビューシーケンス[2]，Bag-Of-Features[3]）の研究開発がなされてきた．その一方で，近年，移動ロボットによる地図生成技術は大きく進展し，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[4]．さらに，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技術の研究がなされている[5]．それにともない，上記の認識・検索の性能に加えて，

- コンパクト性：コンパクトであり，記憶・送受に有効であること

という新たな要求を満たすランドマーク技術が求められている．

本研究では，上記要求を満たすものとして，シーンの Gist 特徴に着目した[6]．一般に，ヒトの視覚システムは，シーンの空間表現を瞬時に獲得することができる．この空間表現は，シーンの要点（Gist 特徴）と呼ばれ，たとえば，シーンの意味（例：道路がある），主要な物体（例：道路の両側に高い壁がある），大域的な構造（例：視野の広がり）など，シーンに関する豊富な情報を含む[7]．近年，コンピュータビジョンの分野において，この Gist シーン特徴を画像処理技術として工学的に実装する試みがなされている[8]-[11]．Oliva ら[12]は，画像の低空間周波数成分を抽出するフィルタを用いて，

*知能システム工学科
*Dept. of Human & Artificial Intelligent Systems

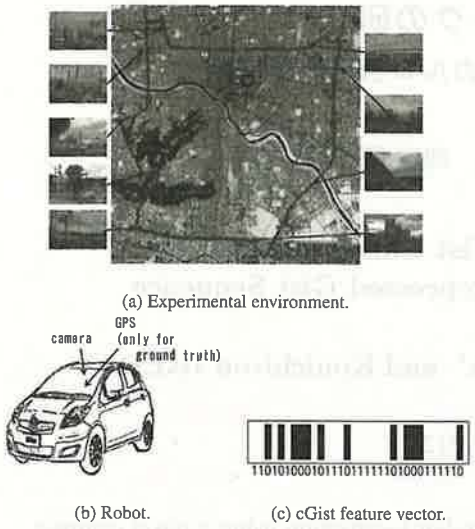



図 1: Experimental platform.

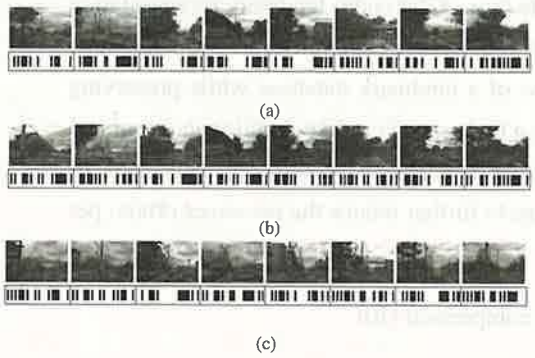



図 2: Compressed Gist sequence.

Gist シーン記述子を開発した．文献[13]では，多層グラフィカルモデルに基づく量子化技術，セマンティックハッシング（SH）[14]を利用して，この Gist シーン記述子を，識別性を保ったまま，コンパクトな 32bit の二値コード（以下，圧縮 Gist）へ量子化する方法を開発した．この Gist および圧縮 Gist は，近年，画像補完[11]や画像検索[13]などの応用において，高い認識性能・検索性能を達成している．

本論文では，ロボットがナビゲーション中に取得する時系列圧縮 Gist を用いる自己位置推定問題を考える．実験プラットホームとして，図 1 のように，視覚センサを搭載した自家用車を使用し，街中の約 20km の道路を，0-50km/h の速度で走行し，自己位置推定を行う．このような，自家用車の自己位置推定タスクは，視点間隔が大きい，カメラが高速振動する，などの理由から，依然として挑戦的な課題である．図 2 a,b,c に，それぞれ異なる日時に取得した時系列圧縮 Gist の 3 つの例を示す．a と b は同じ場所，c は異なる場所で取得した．図からも分かるように，同じ場所で取得した圧縮 Gist は互いに類似している場合が多いが，完全に一致してはおらず，また，異なる走行経路の圧縮 Gist とも部分的に類似しているため，個々の圧縮 Gist だけでは自己位置を一意に決定することができない．本研究のように，圧縮 Gist を時系列で用いることで，単一の圧縮 Gist よりも多くの情報を得ることを期待できる．また，時系列圧縮 Gist の持つ冗長な情報量を利用して，さらなるコンパクト性の向上を期待できる．

以上を踏まえ，時系列圧縮 Gist をランドマークとして利用する自己位置推定システムの有効性を検証することを本論文の目的とする．この目的のために，ランドマーク以外の実験条件は，先行研究と同等の条件とした．具体的に，自己位置推定アルゴリズムには，標準的なモンテカルロ自己位置推定[15]を用いる．また，地図には，事前にロボットが同じ経路を走行し各視点について圧縮 Gist および視点番号を記録したものを用いる．また，性能指標には，車載 GPS により計測した車体位置を基準とした推定位置の誤差を用いる．さらに，時系列圧縮 Gist の有用性を，識別性，パラメータ依存性，汎化性，顕著性，情報圧縮，従来法との性能比較など，様々な観点から検証する．

### 1.1 関連研究と位置付け

自己位置推定問題の先行研究は，多岐に渡って行われている．ここでは，関連研究に焦点を絞って，本研究の位置付けを示す．

一般に，自己位置推定問題は，与えられる地図の種類により，位相地図に基づくものと，計量地図に基づくものとに大別される．前者の位相地図は，ロボットにより識別可能な場所のリスト，および，場所間の接続関係を表す移動可能経路の情報を記す．後者の計量地図は，これらの情報に加えて，物体・経路の位置・形状などの計量情報を記す．本論文は，前者の位相地図に基づく自己位置推定問題に分類される．

位相地図に基づく自己位置推定問題の研究は，使用する視覚特徴の種類により，さらに，局所特徴および大域特徴という 2 つのアプローチに分類される．前者は，識別性に優れる局所特徴を用いて，シーンを記述する．たとえば，[1] の方法は，回転・伸縮・遮蔽・照明などの変化に対して頑健な SIFT 特徴を用いて視覚画像を記述する．このアプローチは，シーンの大域的な変化に対して頑健であるが，その反面，シーンの意味や空間構造などの大域的な特徴を捉えることができないという課題がある．[16] 対照的に，後者の大域特徴のアプローチは，シーンの意味や構造を一つの大域特徴により記述する．たとえば，[2] の方法は，ビューシーケンスを直接に大域的な特徴として用いて視覚画像（列）

を記述する．以上のように，2つのアプローチは相補的であり，両者を組合せる方法についても研究がなされている[16]．本論文は，後者の大域特徴のアプローチに分類される．

視覚画像に基づく自己位置推定問題は，コンピュータビジョンの分野において長年研究がなされており，近年，モバイルや移動ロボットの分野においても，研究が活発になってきている．[17]では，都市規模の大規模環境において，vocabulary tree および枝刈に基づく効率的な自己位置推定方法を提案している．[18] は，三次元点群に基づく，物体の位置推定，セグメンテーション，意味ラベリング，および，分類のための方法を提案している．[19]では，航空写真などの2次元平面地図を所与とし，一枚の全方位画像から，geometric hashing と投票に基づいて，頑健に自己位置推定を行う方法を提案している．[20]では，都市規模の大規模環境において，携帯電話の画像から，ランドマークを認識する方法，および，データセットを公開している．以上を踏まえた上で，本論文は，視覚特徴のコンパクト性に焦点を当てる点に特色がある．

移動ロボットの分野において，局所特徴群をコンパクトに表現する方法として，Bag-Of-Features (BOF) 手法（FABMAP 等[3]）が広く用いられている．その基本となるアイデアは，局所特徴アプローチにおいて，視覚画像を局所特徴の集合により表現することにある．この表現を利用して，視覚画像を量子化された局所特徴（視覚単語）のヒストグラムによりコンパクトに表現し，転置ファイルを利用して高速な検索を実現することができる．しかし，ヒストグラム表現は，依然として，多次元データであり，[21]でも指摘されているように，大規模な画像集合を扱うのに十分なほどコンパクトではない．このような観点から，情報検索の分野において，二値 BOF[22]，圧縮転置ファイル[23]，miniBOF[24] などの研究がなされているが，自己位置推定問題に応用した事例は少ない．

識別性を保ったまま視覚特徴を二値コードへ量子化する技術は，コンピュータビジョンの分野において，研究が盛んになってきている．[25]は，先駆的な研究であり，いくつかの有望な量子化手法を比較検討した．これまでに，spectral hashing[26]，カーネル学習[27]，半教師有り学習[28]などの手法が提案されている．本論文は，画像検索や画像補完の応用において有効性が確認されている[25] の二値コード（圧縮 Gist）を利用する．以上を踏まえた上で，本研究は，二値コードを時系列で利用する自己位置推定の研究としては先駆的なものであり，時系列圧縮 Gist ランドマークの有効性を検証する研究として位置付けられる．

### 1.2 本論文の構成

本論文の構成を以下に示す．2. では，本論文におけるランドマークの認識・検索の方法について説明する．3. では，実験システムの構成を示す．4. では，検証実験を示し，5. において結論を述べる．

## 2. ランドマークの認識と検索

### 2.1 ランドマークの認識

本研究においてランドマークを認識する手続きは，以下の流れとなる．

1. Gist シーン記述子を利用して視覚画像からシーンの Gist（要点）特徴を抽出する．
2. セマンティックハッシングを利用して Gist 特徴を $k = 32$bit の圧縮 Gist

$$z = [z^1, ..., z^k] \tag{1}$$

へ量子化する．

前述のように，この圧縮 Gist を走行経路上の各視点について記録したものを地図とする．この地図は，図 3 のように，視点当り $k$[bit] の情報をもつ．

Gist シーン記述子[12] は，シーン画像を入力とし，知覚次元と呼ばれる，自然性（naturalness），開放性（openness），粗野性（roughness），拡張性（expansion），起伏性（ruggedness）などの大域的な特徴を捉え，シーンの空間的な構造を記述する．そのために，画像のスペクトルや粗い位置情報を利用して特徴抽出を行う．具体的には，4 つの異なる解像度（スケール）の画像を用意しておき，各画像を 4x4 のグリッドに分割し，各々のセル上で，特徴抽出を行う．この特徴抽出においては，画像に，ガボールフィルタを施し，8 つの方向成分（45deg 刻み）の大きさを算出する．その結果は，4x(4x4)x8=512 次元の特徴ベクトルとなる．

セマンティックハッシング[29] は，多層グラフィカルモデルに基づく量子化手法であり，識別性を保ったまま視覚特徴を二値コードへ量子化する．このモデルは，512 次元の実数ノードを最下層とし，32 bit の二値ノードを最上層とする多層ニューラルネットワークとなっている．多層ネットワークの訓練アルゴリズムは，個々の層を個別に学習していく事前トレーニング（pre-training），および，バックプロパゲーションに基づいてネットワークの重みを補正していくファインチューニング（fine-tuning）の処理からなり，計算時間を要する．一方，多層ネットワークを用いる識別アルゴリズムは，掛け算，および，各層毎に非線形関数を計算する処理からなり，高速に実行できる．

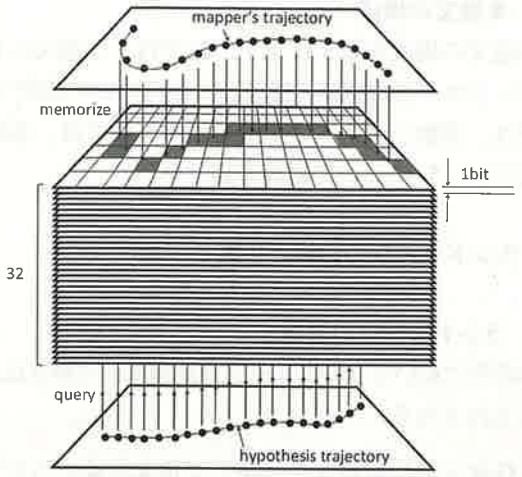



図 3: Binary landmark map.

本論文では，ランドマーク地図の記憶・送受の観点から，ランドマークのコンパクト性をさらに向上させる方法についても検討する．本論文の実験では，各ランドマークの $k = 32$ bit の全てを利用するケースだけでなく，$\delta k$[bit] を間引いた $k'$（$k' = k - \delta k < k$）bit を利用するケースについても，性能検証を行い，地図のコンパクト性と有用性との間のトレードオフについて調査を行う．

### 2.2 ランドマークの検索

本論文で扱う，モンテカルロ自己位置推定など多くの自己位置推定アルゴリズムにおいて，ランドマーク地図を検索する場面には，主に，下記の 2 通りがある[30]．

- 尤度評価：ランドマーク観測の結果をもとに，「ある視点（自己位置）にロボットがいる」という仮説の確からしさを評価する．ここでは，その視点での観測結果を予測するために，地図を検索する．
- 仮説生成：ランドマーク観測の結果をもとに，既存の仮説集合とは独立に，尤もらしい新規仮説（群）を生成する．ここでは，その観測を与える視点を予測するために，地図を検索する．

しかしながら，いずれのケースについても，本研究のような二値コードをランドマークとして利用した研究事例は少ない．そこで，以下では，尤度評価および仮説生成に二値コード（圧縮 Gist ランドマーク）を利用する方法について述べる．

#### 2.2.1 尤度評価への利用

一般に，尤度評価の処理は，自己位置の仮説 $x$ およびランドマーク観測 $z$ を所与とし，ロボットが $x$ にいる条件下で観測 $z$ が起こる条件付き確率密度（尤度）$P(z|x)$ を評価することを目的とする．いま，$x$ を自己位置の仮説，$z$ を観測された圧縮 Gist とし，自己位置が $x$ である条件下で観測される圧縮 Gist を地図に基づき予測したものを $z_{map,x}$ とする．このとき，$z$ と $z_{map,x}$ とのハミング距離 $\Delta z$ をもとに，尤度を

$$P(z|x) = l_o l^{-\Delta z/k} \tag{2}$$

のように算出するものとする．$l_o$ は，$P(z|x)$ が確率密度関数の条件を満足するために導入する正規化定数であり，推定結果に影響を及ぼすことはない．

式 (2) 中で，$l$ は，ハミング距離の差異を重み付けする定数である．実験では，定数 $l$ の値を 1.2, 1.5, 2, 3, 5 のように複数通りに変化させて，推定結果に及ぼす影響を調査する．

上記の尤度評価において，最も計算時間を要するのは，ハミング距離を算出する処理である．この処理は，仮説数と同じ回数だけ繰り返すことになるので，大規模な仮説集合を扱うためには，処理を高速化することが重要になる．その高速化の手段として，ビットカウント操作を利用する方法が有効である．一般に，ハミング距離の算出は，ビットカウント操作に帰着することができ，ビットカウント操作の高速アルゴリズムを利用することで，高速に実行できる．

#### 2.2.2 仮説生成への利用

一般に，仮説生成の処理は，観測されたランドマーク $z$ を所与とし，$z$ の下で尤もらしい仮説群を生成することを目的とする．ランドマークとして二値コードを用いる本研究の場合，この処理は，二値コード $z$ からのハミング距離 $\Delta z$ が $z_o$ 以下となる二値コード，すなわち，不等式

$$\Delta z \leq z_o \tag{3}$$

を満足する全ての二値コードを地図中から検索する処理となる．

上記の仮説生成では，与えられる二値コード $z$ からのハミング距離が $z_o$ 以下となる全ての二値コードを地図中から検索する処理に，最も計算時間を要する．高速化の手段として，参照テーブルを利用する方法が有効である．いま，この参照テーブルには，零ベクトルからのハミング距離が $z_o$ 以下である全ての二値コードを記録しておく．このとき，上記の検索処理は，参照テーブル内の各要素と $z$ との排他的論理和を算出する処理に帰着することができ，高速に実行できる．

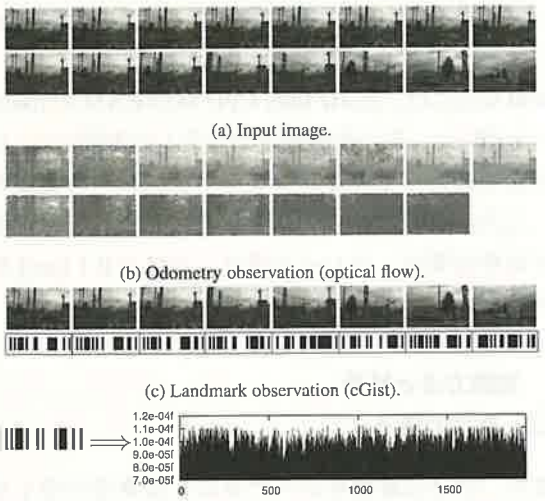



図 4: Self-localization process.

## 3. 自己位置推定システムの構成

本実験で使用する自己位置推定システムは，モンテカルロ自己位置推定[15]の処理の流れに従う．モンテカルロ自己位置推定は，パーティクルフィルタに基づく自己位置推定アルゴリズムであり，時系列観測データに基づく自己位置の推定量を，一つ前の時刻の推定量を用いて，効率的に算出することができる．各時刻 $t$ の観測データには，オドメトリ（自己移動量）$a_t$ およびランドマーク $z_t$ の 2 種があり，それぞれ，運動モデル $P(x_t|x_{t-1}, a_t)$ および知覚モデル $P(z_t|x_t)$ を介して，自己位置 $x_t$ に関する情報を与える．推定量は，時系列観測データ $a^t = (a_1, \cdots, a_t), z^t = (z_1, \cdots, z_t)$ に基づく，条件付き確率密度 $Bel(x_t) = P(x_t|a^t, z^t)$ であり，自己位置の重み付きサンプルの集合 $\{\langle w^{(j)}, x_t^{(j)}\rangle\}_{j=1}^N$ により表される．各時刻 $t$ の推定 $Bel(x_t)$ は，

$$Bel(x_t) = \eta P(z_t|x_t) \int_{x_{t-1}} P(x_t|x_{t-1}, a_t) Bel(x_{t-1}) dx_{t-1} \quad (4)$$

のように前時刻の推定 $Bel(x_{t-1})$ を用いて再帰的に表される．具体的な処理は，以下の手続きからなる．まず，初期時刻 $t = 1$ において自己位置のサンプルを無作為に生成し，各サンプルの重みを同じ値に初期化する．そして，各々の時刻 $t$ において，オドメトリおよびランドマークの観測結果を取得するたびに，それぞれ下記のようにして各サンプルおよび重みを更新する．

1. 運動更新：オドメトリ観測に基づいて，各サンプルの位置を移動する．
2. 知覚更新：ランドマーク観測に基づいて，各サンプルの重みを更新する．

図 4 に，本システムにおける運動更新および知覚更新の処理の様子を示す．以下では，これらの処理について，より詳細に述べる．

### 3.1 運動更新

多くの自己位置推定システムにおいて，オドメトリの手段として車輪エンコーダやジャイロセンサなどの内界センサが広く用いられている．一方，ハンドカメラに基づく SLAM など，内界センサを搭載していない自己位置推定システムにおいては，視覚画像列から自己移動量を推定する，視覚オドメトリのアプローチが一般的であり，たとえば，局所特徴の追跡に基づく方法（例：monoSLAM[31]）やキーフレームの照合に基づく方法（例：PTAM[32]）などの方法が提案されている．しかし，いずれの方法も，追跡・照合する局所特徴やキーフレームが存在することを前提にしており，本実験のように，ロボットが高速移動する場合，観測地点の間隔が大きい，ロボット本体が高速振動する，などの理由により，安定してオドメトリデータを得ることができない．

本研究で対象としている自動車ロボットは，内界センサを搭載しておらず，かつ，安定に視覚オドメトリを行うことはむずかしい．そこで，本研究では，オプティカルフローに基づいて，各時刻 $t$ において車体が静止しているのか否かの二値データを安定に取得する，簡便な観測方法を開発した．この方法は，次の手順からなる．

1. 現在および一つ前のフレーム画像からなる画像対を入力としオプティカルフローを算出し，すべてのフローについてベクトル長を算出する．
2. それらのベクトル長の中間値が閾値を越えるかどうかを判定し，その結果を，車体が静止している (0) か否 (1) かを表す二値データ（自己移動量）とする．

その上で，運動更新の処理では，

$$a_t^{(j)} = \begin{cases} U(0, l_a) & (a_t = 1) \\ 0 & (a_t = 0) \end{cases} \quad (5)$$

のように，サンプル毎の自己移動量を生成する．ただし，$U(,)$ は一様分布を表し，移動量の上限値 $l_a$ は，自家用車の最高時速 50km/h をもとに決定する．

### 3.2 知覚更新

知覚更新は，サンプル毎に，

$$w^{(j)} \leftarrow w^{(j)} P(z|x^{(j)}) \quad (6)$$

のように重みを更新する処理となる．ここで，$P(z|x)$ は，2.2.1 の尤度評価により得られる．本実験では，この知覚更新に関連して，モンテカルロ自己位置推定において広く用いられている 2 種類の改善方法，リサンプリングおよびセンサリセットを導入する．リサンプリングは，確率密度を精度よく推定することを目的として，有効サンプル数[33] に基づき適時に，サンプル集合を分布 $Bel(x_t)$ からサンプルしなおす．センサリセット[34]は，大域自己位置推定やエラーリカバリを目的として，ランドマーク観測のたびに，サンプル群の一部分（全体の 1%）をランダムに選択し，同数の新規サンプルにより置き換える．ただし，この新規サンプルは，2.2.2 の仮説生成により得られる．[35] では，このセンサリセットの様々な戦略を実装し，比較検討を行っている．以上のリサンプリングおよびセンサリセットの実装方法は，著者らの研究[30] および[36] においても同様に実装し有効性を確認している．

## 4. 評価実験

時系列圧縮 Gist ランドマークの有効性を検証するために実験を行った．本章では，まず実験の基本的な設定について述べる．その後，実験方法と結果を示す．

### 4.1 設定

実験プラットホームとして，単眼カメラを搭載した自家用車（以下，ロボット）を用いる．ロボットの外観を図 1 に示す．カメラは，車内上部に前方を向けて設置した．

自己位置推定タスクの性能指標として，車載 GPS により計測した車体位置を基準とした推定位置の誤差を用いる．ただし，GPS の外れ値および誤差に対処するため，簡便な線形補間法を用いた．

自己位置推定システムのうち，量子化を行う多層ニューラルネットワーク（セマンティックハッシング）は，2.1 で述べたように，事前に学習しておく．この学習のために，LabelMe ウェブサイト[37] からダウンロードした 70,000 枚の画像を用いる．

図 1 の経路をロボットが 2 周走行し 2 本の視覚画像列を取得した．1 本目の視覚画像列は，ランドマーク地図を生成するのに用いた．もう 1 本は，後述のように，自己位置推定タスクの視覚画像列として用いる．走行経路長は約 20km，ロボットの走行速度は 0-50km/h，画像取得のフレームレートは 10fps であった．以下では，それぞれの画像列を，“mapping” および “localization” と呼んで区別する．

特に断りがない場合，モンテカルロ自己位置推定のサンプル数を 10,000，重み係数を $l = 2$，圧縮 Gist ランドマークのビット数を $k = 32$ とした．

計算処理には，2GHz Intel CPU 8GB RAM の linux マシンを用いた．自己位置推定システムの実装には，C++ 言語を用いた．セマンティックハッシングを実装する際，公開 Matlab コード[13] を参考にした．自己位置推定の処理時間は，フレーム当り，合計で 0.1 [sec] 前後であった．

### 4.2 実験方法と結果

#### 4.2.1 推定の様子

まず，自己位置推定システムによる推定の様子を示す．ここでは，“localization” から長さ 100 の視点列を，無作為に 100 通り抽出し，各々を視覚画像列として用いて自己位置推定タスクを実行した．さらに，各々のタスクを，最終的な推定誤差が 200m（経路長の 1%相当）よりも小さくなったかどうかを基準にして，成功例と失敗例とに分類した．成功例および失敗例から，それぞれ 5 通りずつを無作為に抽出し，推定誤差の時間変化をプロットしたものを図 5a,b に示す．

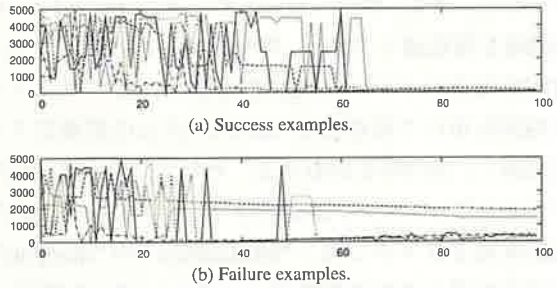
図からも分かるように，自己位置推定タスクの開始時には，自己位置は完全に未知であり，推定誤差は，非常に大きい．成功例の場合，この推定誤差が徐々に減少していき，最終的には，ほぼ零へと収束した．また，多くの場合，各々の場所で観測された圧縮 Gist は地図中で同じ場所にある圧縮 Gist と一致し，その結果として，モンテカルロ自己位置推定において正しい仮説に高い尤度が与えられた．一方，失敗例の場合，推定誤差は，全く収束しないか，あるいは，十分に小さくならないか，のいずれかであった．

#### 4.2.2 成功率

つぎに，自己位置推定の成功率について，下記のパラメータを変化させながら，調査した．

1. サンプル数 $N$
2. ビット数 $k$
3. 重み付けパラメータ $l$

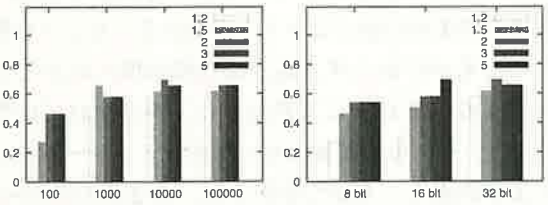
(1) は，モンテカルロ自己位置推定において生成する自己位置の仮説数である．この値を大きくすると，自己位置の多様な仮説を考慮することになり，精度やエラーリカバリの面で信頼性が向上するが，その反面，モンテカルロ自己位置推定の処理コストは線形的に増大する．(2) は，圧縮 Gist の bit 数であり，セマンティック



(a) Success examples.

(b) Failure examples.

図 5: Estimation error [m].



(a) vs #samples (b) vs #bits

図 6: Success ratio vs. #samples (left) and success ratio vs. #bits (right) over the 100 tasks. Each plot corresponds to each value of parameter $l$.

ハッシングの最上位層のノード数に等しい．この値を大きくすると，個々の圧縮 Gist ランドマークの情報量が多くなるが，その反面，ランドマークの記憶に係る空間コストは線形的に増大する．(3) は，圧縮 Gist による観測結果を推定量に反映させる際の重み付けを行う係数である．この値を大きくするほど，リサンプリングにおいて有望なサンプルに計算資源を集中させることができるが，その反面，サンプル集合全体としての多様性が失われやすくなる．4.2.1 で使用した 100 通りの自己位置推定タスクのうち，最終的に推定誤差が収束したもの（全体の 9 割程度）を母集団とし，成功率を調べた．図 6a および 6b に，それぞれ，(1) および (2) と成功率との関係を，(3) のいくつかの設定値について示す．

図より，サンプル数 10,000 以上の場合，ほぼ 0.7 程度の成功率が得られている．また，ビット数が 32bit の場合に最も安定した結果が得られている．この結果から，0.7 の成功率で，自己位置の候補を 1% に絞り込むことに成功しており，圧縮 Gist が，ランドマークの認識・検索に有効であることを確認することができた．また，セマンティックハッシングの最上位層のノード数として 32 が適切であることが分かった．

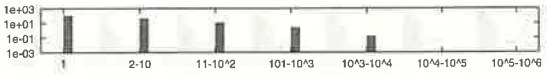


図 7: cGist frequency of “LabelMe” (left) vs. “localization” (right). cGists are sorted and grouped into 7 groups according to number of their corresponding landmarks. Each bar indicates average number of corresponding landmarks for each of the 7 groups.

### 4.2.3 汎化性能

圧縮 Gist ランドマークの汎化性能について考察する．本研究において量子化の手段として利用したセマンティックハッシングは，未知データに対しても高い汎化性能を有することが知られている．すなわち，セマンティックハッシングは，訓練データとして使用した “LabelMe” 画像群だけでなく，自己位置推定に使用した “localization” 画像群に対しても，識別性の高い二値コード（圧縮 Gist）を生成することが期待される．

実際に，本実験において訓練データ “LabelMe” と自己位置推定タスク “localization” との間で，各々の二値コードの出現頻度にどの程度の差異があったのかを可視化してみる．具体的に，“localization” データセットの $2.2 \times 10^4$ 個の圧縮 Gist を対象とし，可視化のために，これらを出現頻度の順にソートした上で 7 つのグループに分け，各々のグループ毎の平均出現回数を集計した．その結果を図 7 に示す．

この結果より，訓練データ “LabelMe” と自己位置推定タスク “localization” との間には，圧縮 Gist の出現頻度に大きな差異があることが分かる．たとえば，訓練データと自己位置推定タスクとの間で，重複した二値コードは 331 個のみであった．これは，“LabelMe” の 4.7% にあたり，また，“localization” の 13.3% に過ぎない．また，出現した二値コードは，2486 種類であり，これら 2486 通りの少数の二値コードが複数回（平均 8.8 回）出現したことになる．以上のような，出現頻度の差異があるにもかかわらず，圧縮 Gist ランドマークによる自己位置推定タスクに成功したことが分かる．

### 4.2.4 顕著性

圧縮 Gist ランドマークの顕著性について考察する．一般に，移動経路上で全てのランドマークが同じ頻度で出現する訳ではなく，ランドマーク間で顕著性に違いがある．この顕著性は，自己位置推定の分野において，ランドマークの価値を計る尺度として広く用いら

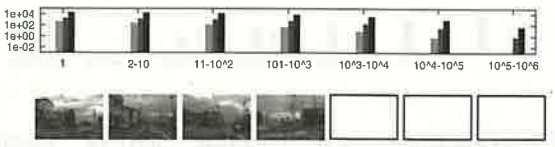



図 8: Saliency of cGist landmark. Top: Each bar indicates average number of corresponding landmarks for each group's 0bit, 1bit, 2bit, 3bit near neighbors from left to right. Bottom: Images corresponding to the central cGist (ID: 1, 5, 50 and 500) of each group.

れており，ランドマーク選択[38]やランドマーク学習[39]などの応用がある．

そこで，自己位置推定タスク "localization" において出現した二値コードを図 7 と同様にグループ分けした上で，各グループ毎に二値コードの平均出現回数を集計してみる．ただし，ここでは，同一の二値コード（ハミング距離 0bit）が出現したケースだけでなく，ハミング距離 1bit,2bit,3bit 内の二値コードが出現したケースについても集計する．その結果を，図 8 に示す．参考のため，図の下段に各グループの代表画像* を示す．

この結果より，二値コード間で出現頻度に大きな差異があることが分かる．たとえば，最頻出の上位 100 の二値コードは，その他のものと比べて，10 倍から 100 倍の高い出現頻度となった．すなわち，少数の高頻出特徴が重要な特徴の大部分を占めていることが分かる．

### 4.2.5 情報圧縮

情報圧縮の観点から，自己位置推定の性能を保ったまま，圧縮 Gist ランドマークのビットをさらに間引くことは可能かどうかについて考察する．そのために，間引くビット数 $\delta k$ を様々に変化させながら，ビット数 $k-\delta k$ のランドマーク地図を用いて自己位置推定タスクを実施し，間引くビット数と自己位置推定性能との間にどのような関係があるのかについて調べた．

その際，自己位置推定の性能指標として，4.2.2 と同様，100 回の自己位置推定タスクを実施した際の成功率を用いた．

また，ビット数 $k-\delta k$ のランドマーク地図には，複数通り（${}_kC_{\delta k}$ 通り）の候補があるが，それらの中から，最も高い有用度を持つものを選択した．ただし，ランドマーク地図の有用度を評価する方法としては，汎用的なシミュレーション経験に基づく方法を利用した[40]．

すなわち，まず，間引くビットのパタン（ビットマスク）の候補を複数通り生成し，各々のビットマスクについて，対応するランドマーク地図を生成し，そのランドマーク地図を用いて仮想的な 100 回の自己位置推定タスクを実施し，成功率を集計する．そして，最も高い成功率を与えたビットマスクを選択する．ただし，仮想的な自己位置推定タスクでは，"localization" や "mapping" とは独立の第 3 の視覚画像列（"validation"）を使用した．

図 9 (a)"selection" に間引くビット数 $\Delta k$ と成功率の関係を調べたものを示す．図 9 (a)"selection" より，$\Delta k$ が 0-12 の広い範囲において，ビットを間引くことによる成功率の劣化が殆どないことが分かる．ビットを間引いたのにもかかわらず，高い成功率が保たれたことから，元々の圧縮 Gist に冗長なビットが含まれていたこと，また，それらの冗長なビットをシミュレーション経験により間引くことができたこと，の 2 点が分かった．

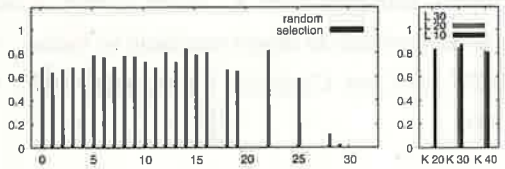
比較のため，どの $\delta k$ ビットを間引くのかを，シミュレーション経験に基づいて決定するのではなく，単純にランダムに決定した場合についても，間引くビット数 $\delta k$ と自己位置推定性能の関係を調査した．その結果を，図 9(a)"random" に示す．この場合，間引くビット数が増えると著しく成功率が低下することが分かる．この比較の結果から，どのビットを間引くかによって自己位置推定性能に大きな影響があることが分かった．

### 4.2.6 従来法[30]との比較

本研究で対象としているセマンティックハッシングのように，識別性を保ったまま視覚特徴を量子化する代表的な方法に，Locality Sensitive Hashing (LSH)[41] がある．この LSH に関連して，著者らは，文献[30] において，LSH に基づく自己位置推定システムを提案している．

ここでは,[30] の LSH に基づく自己位置推定システムを比較手法とし，本論文の自己位置推定システムの性能を再度考察してみる．まず，比較手法の成功率を評価した結果を図 9(b)"LSH" に示す．ただし，図中において，$K$ および $L$ は，LSH の次元数およびハッシュテーブル数であり，それぞれ，値が大きいほど，誤検出および検出漏れを抑制する効果が大きくなる．図より，圧縮 Gist ランドマークに基づく本提案手法は，広い範囲のパラメータについて，比較手法と同等の性能を示している．ただし，パラメータ設定に依存して，比較手法の方が本提案手法よりも高い性能を示しているケースがある．その理由として，LSH は，Gist シーン記述子を視点番号へ直接に写像するため，汎化誤差の影響を受けないことなどが考えられる．

*ここでは，出現頻度がグループ内で中央値 (1,5,50,500) をとる cGist に対応する画像を代表画像とした．代表画像が存在しないグループについては空欄とした．




(a) cGist (b) LSH-Gist

図 9: Performance comparison.

一方，提案方法の特色は，特徴当りの空間コストが非常に低い点にある．LSH は，1 つの視覚特徴を複数のハッシュテーブルに記録するマルチデータベース手法の一種であり，特徴当り 40Byte の空間コストがかかる．これに対し，圧縮 Gist ランドマークは，特徴当り 32bit であり，高い推定性能を保ちながらコンパクトなランドマークを実現できていることが分かる．

## 5. むすび

本論文では，移動ロボットの自己位置推定問題を対象として，時系列圧縮 Gist ランドマークという新しいアプローチを提案した．圧縮 Gist ランドマークは，シーンの意味や空間構造を捉える Gist（要点）記述子を，汎化性能に優れるセマンティックハッシングを利用して，コンパクトな 32bit の二値コードへ量子化したものである．個々の圧縮 Gist ランドマークから得られる情報量は少ないが，本研究では，圧縮 Gist ランドマークを時系列で用いることで，識別性を向上させるアプローチを提案した．実験により，識別性，パラメータ依存性，汎化性，情報圧縮，コンパクト性などの観点から，時系列圧縮 Gist ランドマークが自己位置推定タスクに有効であることを確認した．今後の展望として，複数種のランドマークを組合せる coarse-to-fine 法[42] 等の仕組みを利用して本アプローチの推定精度を向上させることが考えられる．また，本研究のように，識別性を保ったまま視覚特徴を二値コードへ量子化する技術は，近年，画像検索などの分野において，理論的な研究が活発になってきており，今後，自己位置推定の分野においても，大規模地図の記憶・送受などの観点から，重要な役割を担うと考える．

**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，および，立石科学技術振興財団研究助成によった．

## 参考文献

[1] Stephen Se, David Lowe, and Jim Little. Vision-based mobile robot localization and mapping using scale-invariant features. In *In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA*, pages 2051–2058, 2001.

[2] Y. Matsumoto, M. Inaba, and H. Inoue. Visual navigation using view-sequenced route representation. *Proc. IEEE Int. Conf. Robotics and Automation*, pages 83–88, 1996.

[3] M. Cummins and P. Newman. Accelerated appearance-only slam. *Proc. IEEE Int. Conf. Robotics and Automation*, pages 1828–1833, 2008.

[4] Viorela Ila Kai Ni Frank Dellaert, Justin Carlson and Charles E. Thorpe. Subgraph-preconditioned conjugate gradients for large scale slam. In *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. In *Proc. IEEE Int. Conf. Robotics and Automation*, 2010.

[6] K. Ikeda and K. Tanaka. Visual robot localization using compact binary landmarks. In *Proc. IEEE Int. Conf. Robotics and Automation*, 2010.

[7] A. Torralba. How many pixels make an image? *Visual Neuroscience*, 26:123–131, 2009.

[8] Aude Oliva and Antonio Torralba. Modeling the shape of the scene: a holistic representation of the spatial envelope. *Int. J. Computer Vision*, 42(3):145–175, 2001.

[9] J. Hays and A.A. Efros. Im2gps: estimating geographic information from a single image. In *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, 2008.

[10] Matthijs Douze, Hervé Jégou, Harsimrat Sandhawalia, Laurent Amsaleg, and Cordelia Schmid. Evaluation of gist descriptors for web-scale image search. In *Proc. Int. Conf. Image and Video Retrieval*, 2009.

[11] James Hays and Alexei A Efros. Scene completion using millions of photographs. *ACM Transactions on Graphics (SIGGRAPH 2007)*, 26(3), 2007.

[12] A. Oliva and A. Torralba. Modeling the shape of the scene: a holistic representation of the spatial envelope. *Int. J. computer vision*, 42(3):145–175, 2001.

[13] A. Torralba, R. Fergus, and Y. Weiss. Small codes and large image databases for recognition. In *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, pages 1–6, 2008.

[14] R. Salakhutdinov and G. Hinton. Semantic hashing. *Int. J. Approximate Reasoning*, 2008.

[15] F. Dellaert, D. Fox, W. Burgard, and S. Thrun. Monte carlo localization for mobile robots. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 1322–1328, 1999.

[16] Yimeng Zhang, Zhaoyin Jia, and Tsuhan Chen. Image retrieval with geometry-preserving visual phrases. In *CVPR*, pages 809–816, 2011.

[17] G. Schindler, M. Brown, and R. Szeliski. City-scale location recognition. *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, pages 1 – 7, 2007.

[18] Jian Sun, Xiaobai Chen, and Thomas A. Funkhouser. Fuzzy geodesics and consistent sparse correspondences for: eformable shapes. *Comput. Graph. Forum*, 29(5):1535–1544, 2010.

[19] Tat-Jen Cham, Arridhana Ciptadi, Wei-Chian Tan, Minh-Tri Pham, and Liang-Tien Chia. Estimating camera pose from a single urban ground-view omnidirectional image and a 2d building outline map. In *CVPR*, pages 366–373, 2010.

[20] David M. Chen, Georges Baatz, Kevin Köser, Sam S. Tsai, Ramakrishna Vedantham, Timo Pylvänäinen, Kimmo Roimela, Xin Chen, Jeff Bach, Marc Pollefeys, Bernd Girod, and Radek Grzeszczuk. City-scale landmark identification on mobile devices. In *CVPR*, pages 737–744, 2011.

[21] Hervé Jégou, Matthijs Douze, and Cordelia Schmid. Packing bag-of-features. In *Proc. IEEE Int. Conf. Computer Vision*, 2009.

[22] Sivic J. and Zisserman A. Video google: a text retrieval approach to object matching in videos. *Proc. IEEE Int. Conf. Computer Vision*, pages 1470–1477, 2003.

[23] Justin Zobel and Alistair Moffat. Inverted files for text search engines. *ACM Comput. Surv.*, 38(2), 2006.

[24] Hervé Jégou, Matthijs Douze, and Cordelia Schmid. Packing bag-of-features. In *ICCV*, pages 2357–2364, 2009.

[25] Antonio Torralba, Robert Fergus, and Yair Weiss. Small codes and large image databases for recognition. In *CVPR*, 2008.

[26] Yair Weiss, Antonio Torralba, and Robert Fergus. Spectral hashing. In *NIPS*, pages 1753–1760, 2008.

[27] Maxim Raginsky and Svetlana Lazebnik. Locality-sensitive binary codes from shift-invariant kernels. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 1509–1517. 2009.

[28] Jun Wang, Ondrej Kumar, and Shih-Fu Chang. Semi-supervised hashing for scalable image retrieval. In *CVPR*, pages 3424–3431, 2010.

[29] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313:504–507, 2006.

[30] K. Tanaka and E. Kondo. A scalable algorithm for monte carlo localization using an incremental e2lsh-database of high dimensional features. *Proc. IEEE Int. Conf. Robotics and Automation*, pages 2784–2791, 2008.

[31] Andrew J. Davison, Ian D. Reid, Nicholas D. Molton, and Olivier Stasse. Monoslam: Real-time single camera slam. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 29:2007, 2007.

[32] Georg Klein and David Murray. Parallel tracking and mapping for small AR workspaces. In *Proc. Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'07)*, 2007.

[33] A. Doucet, N. Freitas, and N. Gordon editors. Sequential monte carlo methods in practice. *Statistics for engineering and information science*, 2001.

[34] S. Lenser and M. Velose. Sensor resetting localization for poorly modeled mobile robots. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 1225–1232, 2002.

[35] R. Ueda, T. Arai, K. Sakamoto, T. Kikuchi, and S. Kamiya. Expansion resetting for recovery from fatal error in monte carlo localization - comparison with sensor resetting methods. In *Proc. Int. Conf. Intelligent Robots and Systems*, pages 2481 – 2486, 2004.

[36] Kanji Tanaka, Yoshihiko Kimuro, Nobuhiro Okada, and Eiji Kondo. Global localization with detection of changes in non-stationary environments. In *In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 1487–1492, 2004.

[37] B. Russell, A. Torralba, and W. T. Freeman. Labelme: The open annotation tool. *http://labelme.csail.mit.edu/*.

[38] Damian M. Lyons. Selection and recognition of landmarks using terrain spatiograms. In *IROS*, pages 1–6, 2010.

[39] Hauke Strasdat, Cyrill Stachniss, and Wolfram Burgard. Which landmark is useful? learning selection policies for navigation in unknown environments. In *ICRA*, pages 1410–1415, 2009.

[40] P. Sala, R. Sim, A. Shokoufandeh, and S. Dickinson. Landmark selection for vision-based navigation. *Trans. IEEE Robotics*, 22:334–349, 2006.

[41] A. Andoni, M. Datar, N. Immorlica, P. Indyk, and V. Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. *Nearest Neighbor Methods in Learning and Vision: Theory and Practice*, 2006.

[42] Junqiu Wang, Hongbin Zha, and Roberto Cipolla. Coarse-to-fine vision-based localization by indexing scale-invariant features. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 36(2):413–422, 2006.