

## 3D NDT 法による自己位置推定のための環境地図の圧縮

松崎 康平<sup>†,††a)</sup> 田坂 和之<sup>†b)</sup> 川西 康友<sup>†,††c)</sup> 村瀬 洋<sup>†,††d)</sup>

Environmental Map Compression for Localization based on 3D NDT

Kohei MATSUZAKI<sup>†,††a)</sup>, Kazuyuki TASAKA<sup>†b)</sup>, Yasutomo KAWANISHI<sup>†,††c)</sup>,  
and Hiroshi MURASE<sup>†,††d)</sup>

あらまし 3次元点群を用いた自己位置推定における課題の一つは、大規模な環境地図のデータ量の大きさである。自己位置推定の精度を維持しつつ環境地図のデータ量を低減する代表的な手法に、3D NDT (Normal Distributions Transform) 法がある。しかしながら、この手法によるデータ量の低減率は限定的であり、大規模な環境地図に対しては依然としてデータ量が大きい。本論文では、3D NDT 法によって構築された環境地図のデータ量を低減するために、ベクトル量子化に基づくデータ圧縮手法を提案する。ただし、ボクセルごとにベクトル量子化を適用すると、個々のボクセルのもつパラメータ数が少ないために圧縮比は低い。提案手法では、占有と非占有の状態が混在する複数のボクセルから単一のベクトルを合成し、占有/非占有状態と正規分布のパラメータを同時に符号化することによって圧縮比を改善する。実験では、提案手法は自己位置推定の精度を維持しながら、無圧縮の3D NDT と比べて地図データ量を2桁から3桁低減することを示した。圧縮後の地図データ量は元の3次元点群地図のデータ量と比べて5桁程度小さく、経路1 [km] あたりでは約10 [kB] となった。

キーワード 3D NDT, 3次元点群, 自己位置推定, 地図圧縮, ベクトル量子化

## 1. ま え が き

自己位置推定は、車両の自動走行や運転支援における重要な課題である。近年では、高精度な自己位置推定を実現する技術として3D LiDAR (Light Detection and Ranging) センサに基づくスキャンマッチング [1]~[4] が注目を集めている。この技術では3次元点群を環境地図とし、現在位置で計測したスキャン点群とのマッチングを行うことで環境地図上での自己位置を推定する。

3D LiDAR センサに基づく自己位置推定の課題の一つは、環境地図のデータ量の大きさである。3次元点群は大量の  $(x, y, z)$  座標値の集合として表現され、

時には反射強度やRGB色などの付加的な情報も含む。そのため、環境地図のデータ量は膨大となり、車載のストレージに格納しきれない恐れがある。

この課題に対処するために、よりデータ量の小さい地図に変換してスキャンマッチングを行う手法が提案されている。最も代表的な手法は、正射投影画像に基づく手法 [2] と3D NDT (Normal Distributions Transform) 法 [3] である。前者の手法は3次元点群を正射投影された俯瞰画像に変換する。後者の手法は3次元点群を粗いボクセルで分割し、各ボクセル内の点群を正規分布のパラメータに変換する。これにより、自己位置推定精度を維持しつつ、保持すべき地図のデータ量を低減することができる。しかしながら、これらの地図のデータ量は3次元点群地図を基準として2桁から3桁小さい程度であり、大規模な環境地図を想定する場合には数TBから数百TBのオーダーとなる可能性がある [5], [6]。

そのため、データ量の小さい形式に変換された環境地図に対してデータ圧縮を適用し、更にデータ量を低減する試みも検討されている。X. Wei ら [5] は、自己位置推定のために最適化されたConvolutional Neural Network を用いて、正射投影画像を圧縮する手法を提

<sup>†</sup> (株) KDDI 総合研究所, ふじみ野市  
KDDI Research Inc., 2-1-15 Ohara, Fujimino-shi, 356-8502 Japan

<sup>††</sup> 名古屋大学大学院情報学研究科, 名古屋市  
Graduate School of Information Science, Nagoya University,  
Furo-cho, Chikusa-ku, Nagoya-shi, 464-8601 Japan

a) E-mail: ko-matsuzaki@kddi-research.jp

b) E-mail: ka-tasaka@kddi-research.jp

c) E-mail: kawanishi@i.nagoya-u.ac.jp

d) E-mail: murase@i.nagoya-u.ac.jp

DOI:10.14923/transinfj.2019JDP7078

案した。これにより、無圧縮の場合と同等の自己位置推定を達成しつつ、正射投影画像のデータ量を2桁低減した。我々の先行研究 [6] では、3次元点群を二値占有モデルに基づいてベクトル表現に変換し、ベクトル量子化を用いて圧縮する地図表現手法を提案した。更に、その表現形式に適した自己位置推定手法もあわせて提案し、3D NDT法と比べて同程度の推定精度を達成しつつ地図データ量を2桁低減することを示した。しかしながら我々の知る限り、3D NDT法によって構築された地図データを圧縮する試みはない。

本論文では、3D NDT法によって構築された地図を効率的に圧縮する手法を提案する。提案手法では、効率的な非可逆圧縮手法であるベクトル量子化を用いてボクセルのもつ正規分布パラメータを圧縮する。ただし、ボクセルごとに圧縮する場合には個々のボクセルのもつパラメータ数が少ないため、圧縮比が限定的になる。そこで、空間をブロックと呼ばれる単位で分割し、各ブロックに含まれる複数のボクセルのもつ正規分布パラメータを連結してベクトル化することにより、圧縮比を改善する。しかしながら、3D NDT法によって構築された地図では点群の有無に応じて占有/非占有状態のボクセルが混在するため、これらの状態を復元するためにブロック内の各ボクセルの占有/非占有を1/0で表現した状態情報を追加する。本論文では、ベクトル量子化に基づくブロック単位のデータ圧縮を実現するために、状態情報と複数のボクセルの正規分布パラメータを同時に符号化する手法を提案する。

本研究は我々の先行研究 [6] と同様に、地図データのベクトル表現とベクトル量子化による圧縮に基づく。ただし、先行研究では地図の表現形式と自己位置推定手法を独自に設計したのに対し、本研究の提案手法で圧縮・復元された地図は3D NDT法による地図との互換性を持ち、既存の自己位置推定手法をそのまま利用できる。すなわち、本研究による地図データの圧縮・復元機能は、3D NDT法による既存の地図生成と自己位置推定のシステムに容易に統合可能である。3D NDT法による自己位置推定は幅広く利用されており、地図データベースや関連するソフトウェアも多数存在するため、本研究の意義は大きいと考えられる。

本研究の貢献は、以下に要約される。

(1) 3D NDT法によって構築される地図データに対してベクトル量子化を適用する、データの圧縮と復元のためのフレームワークを提案した。

(2) 占有状態と非占有状態が混在する複数のボク

セルデータから単一のベクトルを合成し、ボクセルの占有/非占有状態と正規分布パラメータを同時に符号化する手法を提案した。

(3) 提案手法の有効性を検証するための包括的な実験を行い、自己位置推定の精度を維持しながら地図データの圧縮比を改善することを示した。

本論文の残りの構成は以下のとおりである。2.にて関連研究について概説し、3.で3D NDT法によって構築された地図データの新たな圧縮手法を提案する。4.では地図データの圧縮比と自己位置推定の精度について評価を行い、5.でまとめを述べる。

## 2. 関連研究

本章では、本研究との関連が大きい3D NDT法とベクトル量子化について概説する。

### 2.1 3D Normal Distributions Transform

3D NDT法 [3] は、3次元点群地図  $\mathcal{F}$  をボクセル内の3次元点群を表現する正規分布 (Normal Distribution, ND) パラメータの集合に変換する地図生成処理と、センサから計測されたスキャン点群  $\mathcal{M}$  を  $\mathcal{F}$  に整理させる自己位置推定処理で構成される。

地図生成処理では、初めに解像度  $r$  のボクセルを用いて  $\mathcal{F}$  を分割する。ここで点が割り当てられたボクセルを占有ボクセル、空のボクセルを非占有ボクセルと呼ぶ。その後、各占有ボクセル内の3次元点群  $\mathcal{P} = \{\mathbf{p}_i \in \mathbb{R}^3 | i = 1, \dots, n\}$  を次式に従ってNDパラメータ  $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$  に変換する。

$$\boldsymbol{\mu} = \frac{1}{n} \sum_{i=1}^n \mathbf{p}_i = r \begin{pmatrix} \mu_x \\ \mu_y \\ \mu_z \end{pmatrix}, \quad (1)$$

$$\begin{aligned} \boldsymbol{\Sigma} &= \frac{1}{n-1} \sum_{i=1}^n (\mathbf{p}_i - \boldsymbol{\mu})(\mathbf{p}_i - \boldsymbol{\mu})^\top \\ &= r^2 \begin{pmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{xy} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{xz} & \sigma_{yz} & \sigma_{zz} \end{pmatrix}, \end{aligned} \quad (2)$$

ここで  $\boldsymbol{\mu} \in \mathbb{R}^3$  は平均、 $\boldsymbol{\Sigma} \in \mathbb{R}^{3 \times 3}$  は分散共分散であり、それらのスケールは  $r$  に依存する。地図データは通常、各占有ボクセルの座標  $\mathbf{v} \in \mathbb{R}^3$  も含む。本論文では、平均  $\boldsymbol{\mu}$  はこの座標  $\mathbf{v}$  を基準点とした相対座標として表現されるものとする。したがって、 $\mathcal{F}$  を3D NDT法に基づいて変換した地図データ (以下、NDT地図と呼ぶ) は、占有ボクセルのパラメータ  $\Theta = \{\boldsymbol{\mu}, \boldsymbol{\Sigma}, \mathbf{v}\}$

の集合として表現される。

自己位置推定処理では、スキャン点  $m \in \mathcal{M}$  と NDT 地図の間の距離の総和を最小化する並進と回転の姿勢変換パラメータ  $\xi = (t_x, t_y, t_z, \theta_x, \theta_y, \theta_z)$  を求める。初めに、座標  $v$  を用いて点  $m$  を NDT 地図のボクセルに割り当て、点と ND パラメータの対応を構築する。その後、それらの対応間の距離を表すスコア関数  $S$  を計算する。

$$P(m) \propto \exp\left(-\frac{(m - \mu)^\top \Sigma^{-1} (m - \mu)}{2}\right), \quad (3)$$

$$S(\xi) = - \sum_{m \in \mathcal{M}} P(T(\xi, m)), \quad (4)$$

ここで  $P$  は各ボクセルのもつ ND パラメータ  $(\mu, \Sigma)$  を用いて定義された確率密度関数、 $T$  は姿勢変換関数である。そして、ニュートン法を用いてパラメータ  $\xi$  を反復的に最適化する。

## 2.2 ベクトル量子化に基づくデータ圧縮

ベクトル量子化 [7], [8] とは、所与のベクトル  $\mathbf{x} \in \mathbb{R}^D$  を  $K$  個のベクトルの集合  $\mathcal{C} = \{\mathbf{c}_k \in \mathbb{R}^D\}_{k=1}^K$  (コードブック) を用いて量子化するデータ圧縮手法である。 $\mathcal{C}$  は、訓練用のベクトルの集合を  $K$  通りのグループにクラスタリングした際のクラスタ中心の集合である。ベクトル  $\mathbf{x}$  を量子化するには、 $\mathbf{x}$  との距離  $d(\mathbf{x}, \mathbf{c}_i)$  が最小となるベクトル  $\mathbf{c}_i$  を探索し、そのインデックス  $i \in \mathbb{Z}^K$  に置き換える。ここで、 $d$  は任意の距離尺度であり、量子化の際には  $d(\mathbf{x}, \mathbf{c}_i)$  の量子化誤差が生じる。また、 $\mathcal{C}$  を用いて  $i$  を逆量子化することにより、ベクトル  $\hat{\mathbf{x}} = \mathbf{c}_i$  を復元することができる。

ベクトル量子化は、ベクトル形式で表現可能な任意のデータに対して適用可能であり、画像や音声などの様々なデータの圧縮に利用されている [9], [10]。例えば画像の圧縮を行う際には、 $n_x \times n_y$  個のピクセルをまとめたブロックと呼ばれる単位で画像を分割する。そして、各ブロック内の画素値を並べたベクトルをベクトル量子化することにより、画像を圧縮する。したがって、圧縮する対象とコードブックの間で、ベクトル単位での類似性が高いほど圧縮効率が向上する。

3D NDT 法によって構築される環境地図では、例えば道路や壁面のようなボクセル単位で類似する形状のパターンを多く含むことが考えられる。したがって、実環境とコードブックの間のベクトルが類似し、高い圧縮効率を得ることが期待される。更に、ベクトル量子化は復元時の計算量が  $O(1)$  であり、自己位置推定

のように実時間処理が求められるアプリケーションに適している。そのため、本論文ではベクトル量子化を用いた NDT 地図の圧縮を検討する。

## 3. 提案手法

本章では、NDT 地図をベクトル量子化に基づいて圧縮及び復元する手法を提案する。最も単純な考えは、占有ボクセルごとの ND パラメータをベクトル化し、ベクトル量子化を適用することである。しかし、個々の ND パラメータはパラメータ数が少ないため、データの圧縮比が限定的になる。そのため、画像の圧縮からのアナロジーとして、ブロック内の複数のボクセルの ND パラメータを連結することによって  $D$  次元のベクトルを合成し、ベクトル量子化を適用することを考える。ただし、NDT 地図では占有ボクセルが非一様に分布するため、ブロック内には占有と非占有のボクセルが混在する可能性がある。そこで、常に  $D$  次元のベクトルを合成するために、非占有ボクセルに対して仮の ND パラメータを与える。この場合、復元時にはブロック内の各ボクセルの占有/非占有を表す状態情報が必要となる。したがって本論文では、NDT 地図に対するデータ圧縮比を改善するために、ブロック単位で状態情報及び ND パラメータを連結し、ベクトル量子化を適用する手法を提案する。

この手法では、以下の問題への対処が求められる。

1. **状態情報の連続値表現** ベクトル量子化に基づいて圧縮・復元された状態情報は  $[0, 1]$  の連続値で表現されるため、本来は  $1/0$  の離散値で表現される占有/非占有の状態情報と表現形式が一致しない。
  2. **異なる特徴量の連結** 合成されたベクトルは状態情報や平均、分散、共分散という異なる特徴量で構成されるため、各要素の単位やスケールが一致しない。
  3. **高次元ベクトルのクラスタリング** 複数のボクセルのもつパラメータを連結したことによりベクトルが高次元となり、クラスタリング精度が低下する。
- 上記の問題 1, 2, 3 は、それぞれ **3.1** のベクトル合成・分解、**3.2** のベクトル正規化・逆正規化、**3.3** のベクトル量子化・逆量子化において対処される。

図 1 は提案手法のフレームワークを表しており、左側が圧縮処理、右側が復元処理に対応する。提案手法は NDT 地図が入力として与えられることを前提とし、ベクトル合成、ベクトル正規化、ベクトル量子化を介して NDT 地図を圧縮する。圧縮されたデータが与えられた場合、ベクトル逆量子化、ベクトル逆正規化、

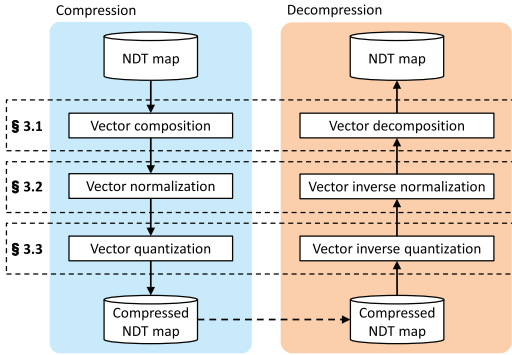


図1 提案手法のフレームワーク  
Fig.1 Framework of the proposed method.

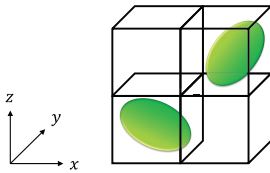


図2 ブロックの模式図  
Fig.2 Illustration of a block.

ベクトル分解を介して NDT 地図を復元する。

### 3.1 ベクトル合成・分解

所与の NDT 地図を、占有ボクセルのパラメータ  $\Theta = \{\boldsymbol{\mu}, \boldsymbol{\Sigma}, \mathbf{v}\}$  の集合とみなす。初めに、 $x, y, z$  方向にそれぞれ  $n_x, n_y, n_z$  個のボクセルをまとめたブロックと呼ばれる単位で空間を分割する。各ブロックが含むボクセルの個数を  $N = n_x \times n_y \times n_z$  とする。図2に、 $n_x = 2, n_y = 1, n_z = 2$  ( $N = 4$ ) の場合のブロックの模式図を示す。この図では、黒色の線で描かれた立方体がボクセルを、緑色のだ円体がボクセルのもつ ND パラメータを表現している。何も描かれていないボクセルは、ND パラメータをもたない非占有ボクセルであることを意味する。なお、提案手法では非占有ボクセルのみで構成されたブロックは無視する。

ベクトル合成処理では、ブロック内でのボクセルの位置に対応するインデックスの順に各ボクセルの ND パラメータを連結することによって、単一のベクトルを合成する。提案手法では、非占有ボクセルに対して仮の ND パラメータを与える。この場合、元のボクセルにおける占有と非占有の状態を復元するためには、それらを表す状態情報が必要となる。しかし、状態情報を個別に保存した場合、そのデータ量によってブロック単位の圧縮比が制限される恐れがある。この問題に対処するために、状態情報も上記のベクトルに含

めてベクトル量子化することにより、ブロック単位のベクトル量子化を実現する。データの復元時には、逆量子化されたベクトルにおける占有/非占有を表す要素は誤差を含むため、しきい値を用いて 1/0 で表される状態情報を推定する。これにより、占有/非占有の状態を誤る可能性が生じる一方で、ブロック単位で高い圧縮比を実現する。

初めに、ブロック内の各ボクセルの占有と非占有の状態をそれぞれ 1 と 0 で表現した二値ベクトル  $\mathbf{B} \in \{0, 1\}^N$  を構築する。この  $\mathbf{B}$  は、ブロック内の各ボクセルにインデックス  $i \in \{1, \dots, N\}$  を割り当て、各インデックスを次元とみなして対応するボクセルの状態を表現する要素を並べたベクトルである。

次に、占有ボクセルごとに ND パラメータをベクトル化する。具体的には次式で表すように、平均  $\boldsymbol{\mu} \in \mathbb{R}^3$  の要素と分散共分散  $\boldsymbol{\Sigma} \in \mathbb{R}^{3 \times 3}$  の上三角要素を並べることによってベクトル  $\mathbf{v}_i \in \mathbb{R}^9$  を構築する。

$$\mathbf{v}_i = h(\boldsymbol{\mu}^T, r^2(\sigma_{xx}, \sigma_{xy}, \sigma_{xz}, \sigma_{yy}, \sigma_{yz}, \sigma_{zz})), \quad (5)$$

ここで  $i$  はブロック内でのボクセルのインデックス、 $h$  はベクトルの結合関数である。式(2)より  $\boldsymbol{\Sigma}$  は上三角要素から全ての要素を復元可能であるため、上三角以外の要素はこのベクトルには含めないものとする。

更に、非占有ボクセルに ND パラメータを与え、ベクトル化する。提案手法ではボクセルの占有/非占有の状態を推定によって求めるため、推定を誤った場合にはこの ND パラメータをもつボクセルが復元される。本論文では、標準的な ND パラメータとして平均  $\boldsymbol{\mu}$  にはボクセルの中心座標を、分散共分散  $\boldsymbol{\Sigma}$  には単位行列  $\mathbf{I}$  を与えるものとする。ただし、式(2)より  $\boldsymbol{\Sigma}$  は  $r^2$  に比例するため、 $\mathbf{I}$  には  $r^2$  を乗算する。また、式(3)、(4)より  $\boldsymbol{\Sigma}$  のスケールは 3D NDT 法のスコア関数に影響を及ぼす。したがって、スケールを調整するための係数  $\delta$  も乗算し、 $\boldsymbol{\Sigma} = \delta r^2 \mathbf{I}$  とする。その後、占有ボクセルの場合と同様に  $\mathbf{v}_i$  を構築する。

そして、 $\mathbf{B}$  と  $N$  個の  $\mathbf{v}_i$  を結合することによって、次式で表される単一のベクトル  $\mathbf{x}$  を合成する。

$$\mathbf{x} = h(\mathbf{B}, \mathbf{v}_1, \dots, \mathbf{v}_N). \quad (6)$$

これは  $N$  次元の  $\mathbf{B}$  と、9次元の  $\mathbf{v}_i$  を  $N$  本結合したベクトルであるため、常に  $D = 10N (= N + 9N)$  次元となる。また、各ブロックの座標  $\mathbf{v}' \in \mathbb{R}^3$  も計算す

る。ここまでの処理により、NDT 地図は占有ボクセルのパラメータ  $\Theta$  の集合から、ブロックを表現するベクトル  $\mathcal{X}$  と座標  $\mathbf{v}'$  の集合に変換される。

提案手法では座標  $\mathbf{v}'$  のデータ量を低減するために、可逆圧縮手法である八分木圧縮 [11], [12] を利用する。結果として、NDT 地図はブロックを表現するベクトル  $\mathcal{X}$  の集合と、八分木圧縮されたブロックの座標情報として表現される。

ベクトル分解処理では、ベクトル合成処理と逆の手順によって NDT 地図を復元する。ただし、ブロックを表現するベクトルは圧縮と復元を経た後の  $\hat{\mathcal{X}}$  であり、 $\mathcal{X}$  に対する誤差を含む。この誤差は、ボクセルの占有と非占有の状態を決定することを困難にする。状態情報はベクトル量子化を介して連続値の誤差が加わることにより、 $[0, 1]$  の連続値として表現される。しかし、状態情報は元々は  $1/0$  の離散値で表現されるものであるため、復元後には表現形式が一致しないという問題が生じる。この問題に対処するために、次式のしきい値処理によって復元後の状態情報を離散化し、ベクトル  $\mathcal{B}$  を推定する。

$$b_i^* = \begin{cases} 1 & \text{if } \hat{b}_i \geq 0.5 \\ 0 & \text{otherwise} \end{cases}, \quad (7)$$

ここで  $\hat{b}_i$  は  $\hat{\mathcal{B}}$  の  $i$  番目の要素、 $0.5$  は  $[0, 1]$  の連続値を離散化するためのしきい値である。このしきい値は、推定結果のバランスを調整するパラメータとみなせる。本論文では、想定される範囲  $[0, 1]$  の中間であり、なおかつ 4.3 の図 6 において優れたバランスを示すことが実験的に確かめられたため、このしきい値を  $0.5$  に設定する。

### 3.2 ベクトル正規化・逆正規化

ベクトル正規化処理では、ブロックを表現するベクトル  $\mathcal{X}$  に対する正規化を行う。この正規化は、単位の正規化と範囲の正規化で構成される。

$\mathcal{X}$  の中で、分散共分散  $\Sigma$  の要素の単位は、平均  $\mu$  の要素の単位の 2 乗となる。したがって、単位の正規化では  $\Sigma$  の要素の平方根をとることによって単位系を統一する。ただし、共分散の要素は符号が負である可能性をもつため、符号付き平方根をとるものとする。

範囲の正規化では、ベクトル  $\mathcal{X}$  の全ての要素を  $[0, 1]$  の範囲に収めるために min-max 正規化を行う。本手法では  $\mathcal{X}$  の要素の種類、すなわち平均、分散、共分散ごとに最大値と最小値を求めて正規化する。しか

しながら、個々の  $\mathcal{X}$  に対して正規化パラメータを定めた場合、地図データ全体でのパラメータのデータ量が大きくなる恐れがある。そのため、訓練用のベクトルの集合に対して単位の正規化を行った後に、平均、分散、共分散の要素の集合を構築し、その中から正規化パラメータ  $\Phi = \{\alpha_{max}, \alpha_{min}, \beta_{max}, \beta_{min}, \gamma_{max}, \gamma_{min}\}$  を求める。そして、全ての  $\mathcal{X}$  に対して  $\Phi$  を用いて正規化を行う。なお、 $\mathcal{B}$  の要素については、常に  $[0, 1]$  の範囲に収まるため、正規化を行わない。

したがって、提案手法におけるベクトル正規化処理は次式でまとめられる。

$$\alpha_{norm} = \frac{\alpha - \alpha_{min}}{\alpha_{max} - \alpha_{min}}, \quad (8)$$

$$\beta_{norm} = \frac{\sqrt{\beta} - \beta_{min}}{\beta_{max} - \beta_{min}}, \quad (9)$$

$$\gamma_{norm} = \frac{\text{sgn}(\gamma) \sqrt{|\gamma|} - \gamma_{min}}{\gamma_{max} - \gamma_{min}}, \quad (10)$$

ここで  $\alpha, \beta, \gamma$  はそれぞれベクトル  $\mathcal{X}$  における平均、分散、共分散の要素であり、 $\text{sgn}$  は符号関数である。

ベクトル逆正規化処理では、次式に従って単位とスケールを復元する。

$$\alpha = \alpha_{norm} (\alpha_{max} - \alpha_{min}) + \alpha_{min}, \quad (11)$$

$$\beta = \{\beta_{norm} (\beta_{max} - \beta_{min}) + \beta_{min}\}^2, \quad (12)$$

$$\gamma'_{norm} = \gamma_{norm} (\gamma_{max} - \gamma_{min}) + \gamma_{min}, \quad (13)$$

$$\gamma = \text{sgn}(\gamma'_{norm}) \gamma'^2_{norm}. \quad (14)$$

### 3.3 ベクトル量子化・逆量子化

ベクトル量子化処理では、事前にコードブック  $\mathcal{C}$  を学習する。そして、圧縮の際には正規化済みのベクトル  $\mathcal{X}_{norm}$  を、ユークリッド距離の 2 乗が最小となる  $\mathcal{C}$  のベクトルのインデックスに置き換える。逆量子化処理では、各インデックスを  $\mathcal{C}$  内の対応するベクトルに置き換えることによって、ベクトル  $\mathcal{X}_{norm}$  を復元する。ただし、ベクトル量子化ではベクトルの次元が高いほど圧縮比の向上が期待されるが、一方で  $\mathcal{C}$  を学習するためのクラスタリングの精度が低下するという問題がある。本研究では、高次元ベクトルに対しても高精度なクラスタリングを実現する手法を適用することによって、この問題に対処する。以下では、 $\mathcal{C}$  の学習方法について説明する。

$\mathcal{C}$  の学習は、次元が等しいベクトルの集合を  $K$  個のグループにクラスタリングすることで実現される。提案手法ではベクトルの次元はブロックが含むボクセ

ルの個数  $N$  に依存するため、圧縮の実行時に想定する  $N$  を用いて訓練用のベクトル集合を構築する。そして、任意の手法を用いて訓練用のベクトル集合をクラスタリングする。一般的には  $k$ -means 法が用いられることが多い。しかしながら、この手法はベクトルの次元が大きい場合にはデータ間の距離の差が小さくなり、クラスタリング精度が低下する恐れがある [13]。提案手法では想定する  $N$  が大きい場合にはベクトルの次元が大きくなるため、高次元なベクトルに対しても高精度なクラスタリングを実現する DCN (Deep Clustering Network) [14] を使用する。この手法では、自己符号化器型の DNN (Deep Neural Network) を用いて次元削減したベクトルをクラスタリングする。

図 3 に、ネットワークの概要図を示す。入力ベクトルを符号化器を用いて潜在空間に写像することによって次元削減し、復号化器を介して入力と同次元のベクトルを復元する。そして、次元削減されたベクトルを  $k$ -means 法に基づいてクラスタリングする。入力ベクトルは一つのブロックを表現する  $D = 10N$  次元のベクトル  $\mathcal{X}_{norm}$  である。ネットワークを学習する際には、次式に示す入力ベクトルの復元損失と、潜在空間でのクラスタリング損失の和  $\mathcal{L}$  を用いてネットワークパラメータを最適化する。

$$z = f(\mathcal{X}_{norm}), \quad (15)$$

$$\mathcal{L} = \|g(z) - \mathcal{X}_{norm}\|_2^2 + \frac{\lambda}{2} \|z - c_k\|_2^2, \quad (16)$$

ここで  $f$  及び  $g$  はそれぞれ符号化器及び復号化器による写像関数、 $z$  は次元削減されたベクトル、 $c_k$  は  $z$  が割り当てられるクラスタの中心、 $\|\cdot\|_2^2$  はユークリッド距離の 2 乗を表す。 $\lambda$  は各損失のバランスを調整するためのパラメータである。これにより、 $k$ -means 法によるクラスタリングに適した潜在空間を学習すると同時に、データのクラスタリングも実現する。提案手

法では、潜在空間でのクラスタ中心を復元したベクトルの集合を  $\mathcal{C}$  とする。また、提案手法ではブロックの時系列及び空間上の配置を考慮せず、全てのブロックを同等かつ独立に扱う。

近年では、DNN を用いてクラスタリングの高精度化を達成する手法が多数提案されているが、これらの手法は DCN に比べると計算量が大きくなるものが多い [15]~[17]。本論文では 100 万以上の大規模なデータのクラスタリングを想定するため、データ量に対するスケーラビリティが高い DCN を選択する。ただし、提案手法には任意のクラスタリング手法が適用可能であり、DCN を他の手法に置き換えても構わない。

我々は予備実験として、 $D = 10, 80, 270, 640$  (i.e.,  $N = 1, 8, 27, 64$ ) の場合に  $k$ -means 法及び DCN で学習した  $\mathcal{C}$  を用いて、テスト用ベクトルの集合をベクトル量子化した際の量子化誤差を比較した。その結果、全ての  $D$  において DCN の方が量子化誤差が少なくなることが確認されたため、本研究では想定する  $D$  にかかわらず常に DCN を使用する。

## 4. 評価実験

提案手法の有効性を評価するために、データの圧縮と復元を経た NDT 地図を自己位置推定へ適用する。

### 4.1 実験設定

評価のために、LiDAR センサや GNSS/INS を搭載した車両によって市街地で計測したデータで構成される KITTI データセット [18] を用いる。また、コードブックを作成するために、KITTI データセットとは異なる実環境で計測された 3 次元点群を含む Málaga Urban データセット [19] を使用する。

NDT 地図を構築する際には、1 辺の長さが  $r$  [m] の等方性のボクセルを使用する。提案手法におけるボクセル合成数  $N$  には  $N = 8, 27, 64$  を使用し、それぞれ  $n_x = n_y = n_z = 2$ ,  $n_x = n_y = n_z = 3$ ,  $n_x = n_y = n_z = 4$  に設定した。また、DCN の実装には著者らが公開したソースコード<sup>(注1)</sup>を利用した。

自己位置推定精度の指標として RMSE (Root Mean Square Error) を使用する。初めに 100 通りのスキャンデータをランダムに選択し、真値から離れる姿勢変換を施した位置姿勢を初期値として地図データに対する自己位置推定を行う。姿勢変換では、半径 2 [m] 以内のランダムな平行移動と  $[-5^\circ, 5^\circ]$  の範囲の  $z$  軸周り

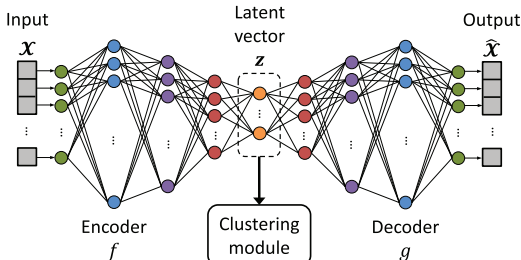


図 3 ネットワークの概要図  
Fig. 3 Overview of the network.

(注1) : <https://github.com/boyangumn/DCN>

のランダムな回転を組み合わせた。そして各スキャンデータに対して 100 回の試行を行い、RMSE 求める。

提案手法では、ボクセルの占有/非占有を推定によって求めるため、誤りを含む可能性がある。この推定精度を評価するための指標として、文献[20]に記載のボクセル IoU (Intersection over Union) を使用する。

また、データの圧縮と復元に伴う復元損失を評価するために、NND (Nearest Neighbor Distance) を使用する。ここでは、復元後の各ベクトル  $\hat{\mathbf{V}}_i$  の最近傍となる ground-truth ベクトル  $\mathbf{V}_i$  とのユークリッド距離を計算する。ベクトルの各要素の単位を統一するために、距離を計算する際には全てのベクトルに対して 3.2 に記載の単位の正規化を行った。

#### 4.2 非占有ボクセルの ND パラメータの影響

初めに、非占有ボクセルの分散共分散  $\Sigma = \delta r^2 \mathbf{I}$  のスケール  $\delta$  について、自己位置推定精度への影響を検証する。検証するパラメータには  $\delta \in \{10^{-6}, 10^{-5}, \dots, 10^0\}$  を選択した。この実験では  $r = 2$  [m],  $K = 2^{16}$ ,  $N = 64$  に固定し、各  $\delta$  における RMSE を調査した。

結果を図 4 に示す。この図より、 $\delta = 10^{-4}$  までは  $\delta$  が小さいほど RMSE が改善していく傾向が見られた。これは、 $\delta$  が小さいほど、元々は非占有であるにもかかわらず占有と推定されたボクセルに起因する悪影響が減少するためである。2.1 より、3D NDT では点と ND パラメータの対応を構築し、式 (3), (4) によりスコアを計算する。言い換えれば、点との対応を構築できない非占有ボクセルはスコアに影響を与えない。したがって、元々は非占有のボクセルを占有と推定した場合、任意の点  $m$  に対して  $P(m) = 0$  であれば自己位置推定への悪影響を抑制できる。式 (3) より、任意の点  $m$  が与えられた場合、 $\Sigma^{-1}$  のスケールが大きいほど  $P(m) = 0$  に近づけることができ、結果として式 (4) のスコアに与える影響が少なくなる。そのため、 $\delta$  が小さいほど RMSE が改善したと考えられる。

一方で、 $\delta = 10^{-5}$  以下では  $\delta$  が小さいほど RMSE

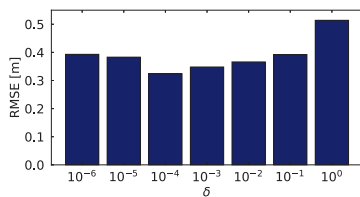


図 4 異なる  $\delta$  の下での自己位置推定精度  
Fig. 4 Localization accuracy under different  $\delta$ .

が悪化した。これは、復元された分散共分散  $\hat{\Sigma}$  の対角成分と非対角成分のスケールの差が小さくなったためである。提案手法では潜在空間上のクラスタ中心を復元することによって  $\mathcal{C}$  を作成するため、 $\mathcal{C}$  内のベクトルは微小な復元誤差を含む。したがって  $\delta$  が復元誤差のスケールよりも小さくなった場合、 $\hat{\Sigma}^{-1}$  がスケールの大きいスカラ行列に近似しない可能性がある。結果として式 (3) における  $P(m)$  の値が大きくなり、RMSE が悪化したと考えられる。

以降の実験では、自己位置推定精度を高めるために常に  $\delta = 10^{-4}$  を使用する。

#### 4.3 コードブックサイズの影響

次に、コードブックサイズ  $K$  について、性能への依存性を検証する。検証するパラメータには、 $K \in \{2^1, 2^2, \dots, 2^{18}\}$  を用いた。また、ボクセルの解像度には  $r \in \{1\text{m}, 2\text{m}, 3\text{m}\}$  を、ボクセル合成数には  $N \in \{8, 27, 64\}$  を用いた。図 5 (a) から図 5 (d) に  $K$  とボクセル IoU, 復元損失, 地図データ圧縮比, 自己位置推定精度の関係をプロットした結果を示す。

図 5 (a) より、 $K$  が大きくなるほど IoU が改善することが分かる。これは、コードブックが含む  $\mathcal{B}$  における占有/非占有のパターンが増加するためである。また、 $N$  が等しい場合は異なる  $r$  でもほぼ同等の値を得ることより、IoU は  $r$  に依存しないことが分かる。これは、3.2 に記載の正規化によって、クラスタリングを行う際の  $r$  依存性を解消しているためである。

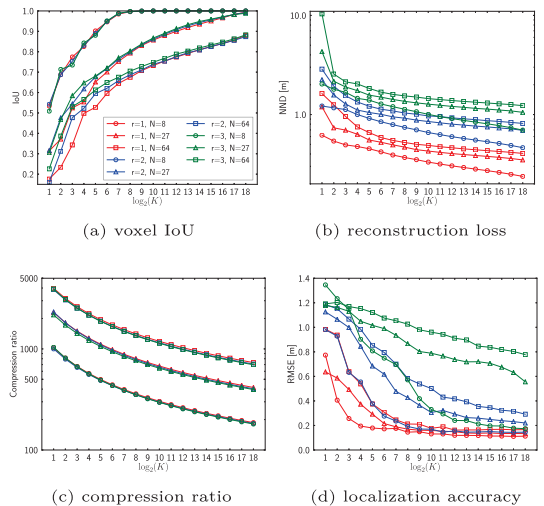


図 5 異なるコードブックサイズでの性能評価  
Fig. 5 Performance evaluation under different codebook sizes.

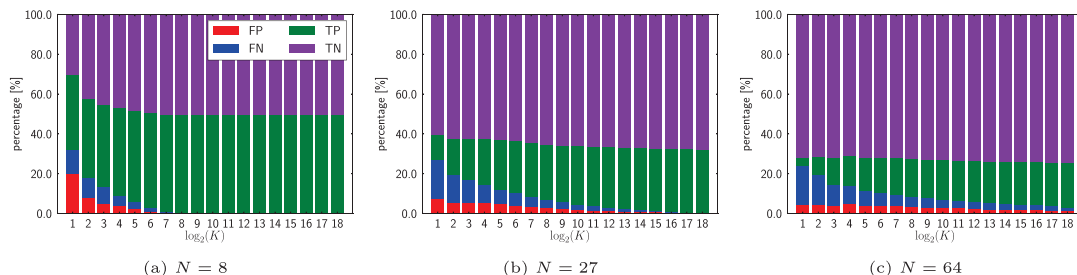


図 6 ボクセルの占有/非占有推定の詳細  
Fig. 6 Details of voxel occupied/unoccupied state estimation results.

図 5(b) より, 全ての  $r$  と  $N$  において  $K$  が大きくなるほど NND が改善していくことが確認できる. これは,  $K$  が大きくなるほどベクトル量子化における量子化誤差が減少するためである.  $r$  が大きいほど NND が悪化する傾向が見られるのは, NND のスケールが  $r$  のスケールに比例するためである. また,  $r$  が等しい場合には  $N$  が大きいほど復元損失が大きくなる. これは,  $N$  が大きいほどボクセルの IoU が小さくなり, 結果として NND が大きくなるためである.

図 5(c) は, 無圧縮の地図データ量を圧縮後の地図データ量で除算した圧縮比を示している.  $K$  が大きくなるほど圧縮比が低下するのは, コードブックのインデックスを表すビット列の長さ  $\log_2(K)$  が増加するためである. また,  $N$  が等しい場合は異なる  $r$  でもほぼ同等の値を得ることより, 圧縮比は  $r$  依存性に乏しいことが分かる. これは, 異なる  $r$  でもベクトル  $\mathbf{X}$  の次元が等しく  $10N$  であるためである. 一方,  $r$  が等しい場合には  $N$  が大きいほど圧縮比が大きくなる. これは,  $N$  が大きいほど所与の NDT 地図を表現するベクトルの本数が少なくなるためである.

図 5(d) より,  $K$  の増加が RMSE の改善をもたらすことが分かる. これは, 上述の IoU 及び NND の改善に起因する. すなわち, 占有ボクセルの推定精度と各占有ボクセルのもつ ND パラメータの復元精度が改善することにより, RMSE が改善したと考えられる. また, 図 5(c) と合わせて見ることで, RMSE と圧縮比がトレードオフの関係にあることがわかる. したがって, 任意の  $K$  を選択することによって RMSE と地図データ量のバランスを調整することができる.

ここで, 本実験におけるボクセルの占有/非占有の推定結果についてより詳細に分析する. 占有/非占有に関する推定結果を正解を用いて TP (True Positive), FP (False Positive), FN (False Negative), TN (True

Negative) に分類し, 図 6 に  $r = 2$  [m] の場合のそれらの割合を  $K$  ごとに示す. 図 6(a), 図 6(b), 図 6(c) はそれぞれ  $N = 8, 27, 64$  の場合に対応する. なお,  $r = 1$  [m] 及び  $r = 3$  [m] の場合については,  $r = 2$  [m] の場合とほぼ同様の結果となったため割愛する. これらの図においては, FN と FP が推定失敗を表し, それらの割合の和は失敗率とみなせる. どの  $N$  においても  $K$  の増加とともに失敗率が低下し, ほとんどの場合に失敗率が 10% を下回ることが確認できる.

#### 4.4 NDT 地図の定性評価

提案手法によるデータ圧縮の効果を定性評価する. この評価のために, 占有ボクセルのもつ ND パラメータをだ円体で表現することによって NDT 地図を視覚化した. 具体的には中心を  $\mu$  とし, 3 軸の向きと半径を  $\Sigma$  の固有ベクトルと固有値とすることでだ円体を表現した. 使用するパラメータには  $N = 64$  と,  $r \in \{1\text{m}, 2\text{m}, 3\text{m}\}$ ,  $K \in \{2^2, 2^7, 2^{12}, 2^{17}\}$  を選択した.

図 7 に, 視覚化の結果を示す. 最も上段の行は無圧縮の NDT 地図を表し, それ以外は提案手法によって圧縮と復元を行った NDT 地図を表している. また, NDT 地図の元となった点群も白色の点として重畳する. 緑色のだ円体は圧縮前後でボクセルの位置が一致した占有ボクセルを表し, 赤色のだ円体は圧縮前は非占有ボクセルであった占有ボクセルを表している.  $r$  が大きくなるほどより粗いボクセルで空間が分割されるため, だ円体の個数が少なくなる. また,  $K$  が大きくなるほど IoU が改善するため, 赤色のだ円体が減少することが確認できる. 緑色のだ円体については,  $K$  が小さいときには無圧縮の地図とは形状差が大きいが,  $K$  が大きくなるほど類似していく傾向が見られた.

#### 4.5 3次元点群地図に対するデータ圧縮比

本節では, 3次元点群地図のデータ量を基準とした



データ圧縮比を評価する。次の5通りの手法を用いた際のデータ圧縮比と自己位置推定精度を調査する。

**NDT** 標準的な3D NDT法[3]である。本論文では、NDT地図は $\Theta = \{\mu, \Sigma, v\}$ の集合として表現される。

**NDT-Octree** NDT地図に対して八分木圧縮[11], [12]を適用する手法である。本手法は、ボクセルの座標 $v$ のみを可逆圧縮する。

**NDT-DS** NDT地図に対してダウンサンプリング(Down Sampling, DS)を適用する手法である。本手法は、 $\Theta = \{\mu, \Sigma, v\}$ の集合からランダムDSを行う。

**NDT-VQ** NDT地図に対して単純にベクトル量子化を適用するベースライン手法である。本手法は、各占有ボクセルの平均 $\mu$ と分散共分散 $\Sigma$ を単一のベク

トルとして表現し、ベクトル量子化する。提案手法との公正な比較のために、ベクトル正規化や座標 $v$ への八分木圧縮等も適用する。言い換えれば、本手法は $N = 1$ に設定した、提案手法の特殊な例である。

**Prop** 本論文における提案手法である。本実験においてはボクセル合成数を $N = 8, 27, 64$ に設定し、それぞれ“Prop-N8”, “Prop-N27”, “Prop-N64”と表記する。

本実験では、全ての手法に対して $r \in \{1m, 2m, 3m\}$ の3通りを、NDT-VQ及びPropにおけるコードブックサイズに $K \in \{2^1, 2^2, \dots, 2^{18}\}$ を設定した。また、各手法の圧縮効率を比較するために、許容可能なRMSEのしきい値を設定する。車線レベルの自己位置推定を実現するためには誤差1[m]未満の精度が必要[21]であり、先行研究[22], [23]では誤差0.2[m]以下の精度を目指していることから、本実験ではこのしきい値を0.2[m]に設定する。全ての結果を図8に示す。この図では、横軸は3次元点群地図を基準としたデータ圧縮比、縦軸はRMSEとし、しきい値を破線で表現する。

NDTのデータ圧縮比は $r = 1$ で79.7,  $r = 2$ で311.9,  $r = 3$ で689.5であり、RMSEはそれぞれ7.6[cm], 9.3[cm], 10.6[cm]であった。 $r$ が大きいほどデータ圧縮比が向上するのは、結果的に得られる占有ボクセルの個数が減少したためである。一方、RMSEが悪化するの、より大きなボクセル内の点群を共通のモデルで表現することに起因する。

NDT-Octreeのデータ圧縮比は $r = 1$ で99.2,  $r = 2$ で388.6,  $r = 3$ で858.8であった。また、八分木圧縮は可逆圧縮であるため、RMSEはNDTと同値であった。NDTを基準とした場合のデータ圧縮比は全ての

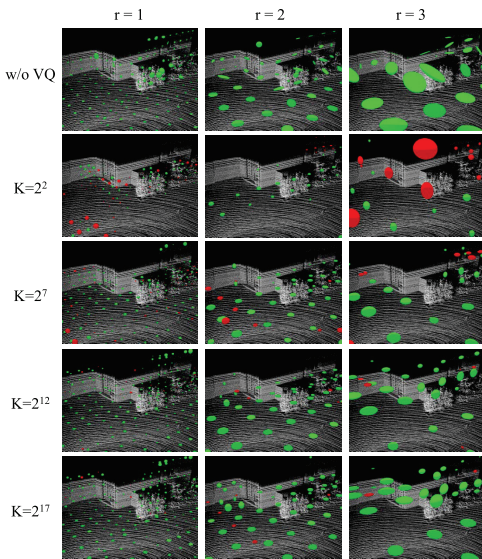


図7 NDT 地図の可視化  
Fig. 7 Visualization of NDT maps.

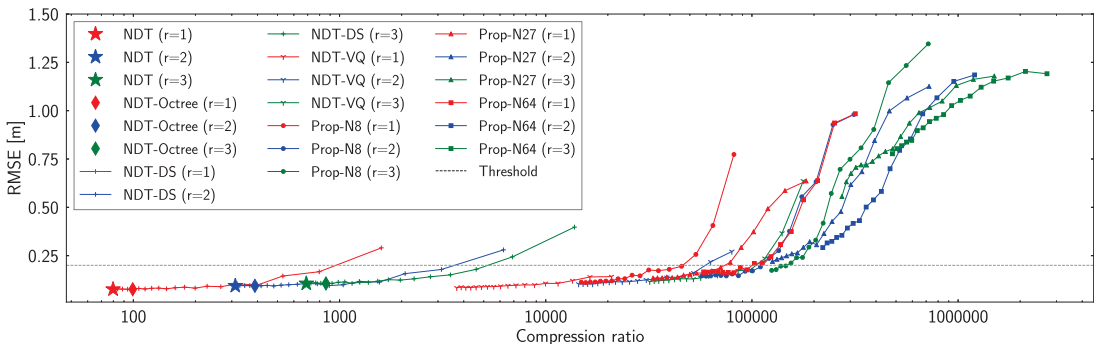


図8 3次元点群地図を基準とした圧縮比と自己位置推定精度の関係  
Fig. 8 Effect of compression ratio for 3D point cloud map on localization accuracy.

$r$  において 1.24 となり、圧縮比は限定的であった。この原因は、占有ボクセルのパラメータ  $\Theta = \{\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{v}\}$  の 15 パラメータの内、座標  $\boldsymbol{v}$  の 3 パラメータに対してのみデータ圧縮を行っていることである。

NDT-DS では DS 率が大きいほど圧縮比が大きくなり、それに伴って RMSE が悪化した。破線で表現されたしきい値を下回る最大の圧縮比は  $r = 1$  で 797.0,  $r = 2$  で 3,119.9,  $r = 3$  で 4,596.9 であった。DS 率はそれぞれ 90%, 90%, 85% に対応する。

NDT-VQ では  $K$  が小さいほど圧縮比が大きくなる一方で、RMSE が悪化した。破線で表現されたしきい値を下回る最大の圧縮比は  $r = 1$  で 20,848.7,  $r = 2$  で 52,108.3,  $r = 3$  で 98,376.6 であった。また、同一の  $r$  をもつ NDT を基準とした場合のデータ圧縮比は  $r = 1$  で 261.5,  $r = 2$  で 167.0,  $r = 3$  で 142.6 となった。この結果より、 $r$  が小さいほど NDT を基準とした圧縮比が大きくなる傾向があることがわかる。これは、RMSE のスケールが  $r$  に比例し、結果として  $K$  が小さい場合でも RMSE を維持できるためである。

提案手法 (Prop-\*) のデータ圧縮比は使用した  $K$  と  $N$  に依存する。破線で表現されたしきい値を下回る最大の圧縮比は  $r = 1$  では  $N = 64$  の場合の 94,697.3,  $r = 2$  では  $N = 8$  の場合の 109,908.9,  $r = 3$  では  $N = 8$  の場合の 145,904.7 であった。これらの圧縮比は各  $r$  において他の手法と比べて最大となり、データ量をおよそ 5 桁低減する。その理由は、複数のボクセルを単一のベクトルで表現することによって、ボクセルの占有/非占有状態と ND パラメータを同時に符号化するベクトル量子化を実現したためである。これにより、NDT-VQ よりもデータ圧縮比を高めることができる。また、このときのボクセルの占有/非占有状態推定の失敗率は  $r = 1$  では 7.51%,  $r = 2$  では 0.148%,  $r = 3$  では 0% であった。この結果より、 $r$  が小さいほど許容可能な失敗率が大きくなることがわか

る。式 (3), (4) に示すとおり、3D NDT 法では全てのスキャン点  $\boldsymbol{m}$  と ND パラメータの対応から得られる  $P(\boldsymbol{m})$  の総和をスコアとする。提案手法における FP に対しては、4.2 で記述したとおり、 $P(\boldsymbol{m}) = 0$  に近くなるように ND パラメータを設計している。そのため、FP が発生した場合でもその悪影響が抑制され、自己位置推定精度を維持できると考えられる。また、FN が発生した場合でも、他の多くのボクセルが TP であれば正しく自己位置推定が可能である。したがって提案手法では、 $r = 1$  において 7.5% という高い失敗率をもつ場合でも、自己位置推定精度を維持することができる。

#### 4.6 ボクセル解像度の影響

これまでの実験では、コードブック  $\mathcal{C}$  を作成するための訓練用データの  $r$  と、ベクトル量子化を行う地図データの  $r$  を常に一致させていた。しかしながら、提案手法はベクトル正規化によって  $r$  依存性を解消しているため、これらの  $r$  が異なる場合でもベクトル量子化を行うことが可能である。本実験では、これらの  $r$  が異なる場合に RMSE へどのような影響が生じるかを調査する。検証するパラメータには  $r \in \{1\text{m}, 2\text{m}, 3\text{m}\}$ ,  $N \in \{8, 27, 64\}$ ,  $K \in \{2^1, 2^2, \dots, 2^{18}\}$  を用いた。

表 1(a) から表 1(c) に、 $N$  ごとの結果を示す。表における行見出しは  $\mathcal{C}$  を作成するための訓練用データの  $r$  (A) を、列見出しはベクトル量子化を行う地図データの  $r$  (B) を表している。表の要素は全ての  $K$  で測定した RMSE の平均値を表す。表の非対角要素においては、括弧内に同行の対角要素との差分を表記する。

この結果より、A と B が一致する場合に、ほとんどのケースで最良の RMSE を達成することがわかる。一方で、A と B が離れるほど、RMSE が悪化する傾向がみられた。これは、訓練用データとベクトル量子化を行う地図データの間で、ベクトルの分布が異なる

表 1 異なるボクセル解像度での自己位置推定精度  
Table 1 Localization accuracy with different voxel resolution.

(a) $N = 8$				(b) $N = 27$				(c) $N = 64$			
$\begin{matrix} A \\ B \end{matrix}$	$r = 1$	$r = 2$	$r = 3$	$\begin{matrix} A \\ B \end{matrix}$	$r = 1$	$r = 2$	$r = 3$	$\begin{matrix} A \\ B \end{matrix}$	$r = 1$	$r = 2$	$r = 3$
$r = 1$	0.195	0.234 (0.0386)	0.247 (0.0519)	$r = 1$	0.244	0.279 (0.0352)	0.286 (0.0419)	$r = 1$	0.331	0.348 (0.0173)	0.372 (0.0412)
$r = 2$	0.299 (-0.0172)	0.316	0.336 (0.0200)	$r = 2$	0.483 (-0.0156)	0.499	0.510 (0.0109)	$r = 2$	0.605 (-0.0185)	0.623	0.648 (0.0244)
$r = 3$	0.588 (0.0363)	0.537 (-0.0139)	0.551	$r = 3$	0.904 (0.0468)	0.871 (0.0147)	0.857	$r = 3$	0.997 (0.0108)	1.001 (0.0142)	0.987

ためと思われる。両者のベクトルは実環境から計測された3次元点群地図を元に作成されるため、AとBの不一致は3次元点群地図のスケール変化に相当する。環境のスケール変化はそこから作成されるベクトルの分布を変化させるため、Cが学習するベクトルとベクトル量子化を行う地図データのベクトルの間で分布の差異が大きくなる。結果として、Cによる量子化誤差が増加し、RMSEが悪化したと考えられる。

しかしながら、AとBの不一致がもたらす誤差の影響は数cmの範囲に収まっており、極端にRMSEを低下させるものではない。特に、AとBの差が小さい場合は与える誤差も小さくなる傾向がある。そのため、AとBをわずかに変化させるケースでは、Cを再学習せずとも実利用上の問題は無いと考えられる。

#### 4.7 逐次的な自己位置推定

本節では、逐次的な自己位置推定に対する提案手法の性能を検証する。この実験では、KITTIデータセットに含まれる3.7kmの経路に渡って計測されたスキャン点群を用いて、NDT地図に対する自己位置推定を行う。誤差を計算する際には、各スキャン点群に対応するGNSS/INSデータを真値とみなす。評価指標として、進行方向への位置誤差(Longitudinal error, Lon.), その横方向への位置誤差(Lateral error, Lat.), 進行方向に対する方向誤差(Orientation error, Ori.), 処理時間(Computational time, Time), 経路1[km]あたりの地図データ量(Data size, Size)を用いる。提案手法のパラメータとしては、高い自己位置推定精度と圧縮比を両立する $(r, N, K) = (1, 64, 2^9), (2, 8, 2^9), (3, 8, 2^{18})$ を選択した。本実験は、3.4GHz Intel Core i7-6800K CPUと32GBのRAMを搭載するPC上で実行した。

表2に標準的なNDT地図と、提案手法によるNDT地図を用いて自己位置推定を行った結果を示す。全ての $r$ に対して提案手法は位置誤差を増加させたが、それらは0.2[m]を下回った。方向誤差は同等か、わずかに

に悪化する程度であった。処理時間は提案手法によって増加する傾向が見られた。これは標準的なNDT地図と比べてNDパラメータが量子化誤差を含むために、ニュートン法に基づく最適化の収束が遅くなったためと思われる。地図データ量に関しては、標準的なNDTでは経路1[km]あたり数MBから十数MBのオーダーであった。一方、提案手法では数kBから十数kBのオーダーであり、標準的なNDTと比べてデータ量を更に2桁から3桁低減することができた。

ここで、3D NDT法以外の環境地図の圧縮方法との比較を検討する。NDT地図と並んで最も代表的な環境地図は正射投影画像地図[2]である。この地図は道路の路面のみを画像化し、ピクセル単位のマッチングによって自己位置を推定する。3D NDT法と比べて環境をより高精細に表現するため、自己位置推定の精度が高くなる傾向がある。例えば、文献[3]では1辺が数mのボクセルを用いるのに対し、文献[2]では1辺が数cmのピクセルを用いて環境を表現する。文献[5]では、正射投影画像地図を圧縮する手法を提案している。この文献では、地図データ量は道路150[m]ごとに4[MB]、すなわち26.7[MB/km]と想定される。そして、著者らの手法によって平均誤差0.1[m]以下の自己位置推定を実現しつつ、圧縮比500で地図データを圧縮することができる。この場合の地図データ量は53.3[kB/km]であり、我々の地図データ量の5倍以上となる。したがって、ベースとなる環境地図の特性が異なるため公正な比較が困難ではあるが、我々の手法は精度に関しては[5]の手法に劣る一方で、地図データ量に関しては優位性をもつと考えられる。

## 5. むすび

本論文では、3D NDT法によって構築された環境地図のデータ量を低減するためのデータ圧縮手法を提案した。提案手法は複数のボクセルがもつ占有/非占有の状態及び正規分布パラメータをベクトル化し、ベクトル量子化に基づいてデータ圧縮を行う。公開データセットを用いた評価実験によって、提案手法が誤差20[cm]以下の自己位置推定を達成しつつ標準的な3D NDT法と比べて環境地図のデータ量を2桁から3桁低減することを示した。今後は、点が疎であったり、ある方向には点群が存在しないなど点群に偏りのある環境で提案手法の有効性を検証する。また、3D NDT法に基づくセグメンテーション[24]や変化検知[25]等、自己位置推定以外の課題への応用を検討する。

表2 自己位置推定性能及び地図データ量の比較  
Table 2 Comparison of localization performance and map data size.

	$r = 1$		$r = 2$		$r = 3$	
	NDT	Prop	NDT	Prop	NDT	Prop
Lon. [m]	0.0771	0.150	0.0995	0.153	0.100	0.158
Lat. [m]	0.0997	0.153	0.115	0.153	0.120	0.176
Ori. [°]	0.274	0.278	0.308	0.296	0.308	0.436
Time [s]	0.147	0.198	0.0345	0.0589	0.0410	0.0568
Size [MB]	13.0	0.0101	3.32	0.0103	1.50	0.00832

## 文 献

- [1] P.J. Besl and N.D. McKay, "A Method for Registration of 3-D Shapes," IEEE Trans. on PAMI, vol.14, no.2, pp.239–256, 1992.
- [2] J. Levinson, M. Michael, and T. Sebastian, "Map-Based Precision Vehicle Localization in Urban Environments," Proc. RSS, 2007.
- [3] M. Magnusson, A. Lilienthal, and T. Duckett, "Scan Registration for Autonomous Mining Vehicles using 3D-NDT," JFR, vol.24, no.10, pp.803–827, 2007.
- [4] R.W. Wolcott and R.M. Eustice, "Robust LIDAR Localization using Multiresolution Gaussian Mixture Maps for Autonomous Driving," IJRR, vol.36, no.3, pp.292–319, 2017.
- [5] X. Wei, I.A. Bãrsan, S. Wang, J. Martinez, and R. Urtasun, "Learning to Localize Through Compressed Binary Maps," Proc. CVPR, pp.10316–10324, 2019.
- [6] 松崎康平, 柳原広昌, "3次元点群を用いた車両の自己位置推定のための圧縮地図表現," 信学論 (D), vol.J101-D, no.11, pp.1471–1481, Nov. 2018.
- [7] Y. Linde, A. Buzo, and R.M. Gray, "An Algorithm for Vector Quantizer Design," IEEE Trans. Commun., vol.28, no.1, pp.84–95, Jan. 1980.
- [8] R.M. Gray, "Vector Quantization," IEEE ASSP Magazine, vol.1, no.2, pp.4–29, 1984.
- [9] P.C. Cosman, K.L. Oehler, E.A. Riskin, and R.M. Gray, "Using Vector Quantization for Image Processing," Proc. IEEE, vol.81, no.9, pp.1326–1341, 1993.
- [10] H.B. Kekre and T.K. Sarode, "Speech Data Compression using Vector Quantization," IJCISE, vol.2, no.4, pp.251–254, 2008.
- [11] D. Meagher, "Geometric Modeling using Octree Encoding," Computer Graphics and Image Processing, vol.19, no.2, pp.129–147, 1982.
- [12] A. Hornung, K.M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "Octomap: An Efficient Probabilistic 3D Mapping Framework based on Octrees," Autonomous Robots, vol.34, no.3, pp.189–206, 2013.
- [13] M. Steinbach, L. Ertöz, and V. Kumar, "The Challenges of Clustering High Dimensional Data," New Directions in Statistical Physics, pp.273–309, 2004.
- [14] B. Yang, X. Fu, N.D. Sidiropoulos, and M. Hong, "Towards K-means-friendly Spaces: Simultaneous Deep Learning and Clustering," Proc. ICML, 2017.
- [15] E. Min, X. Guo, Q. Liu, G. Zhang, J. Cui, and J. Long, "A Survey of Clustering with Deep Learning: From the Perspective of Network Architecture," IEEE Access, vol.6, pp.39501–39514, 2018.
- [16] T. Zhang, P. Ji, M. Harandi, W. Huang, and H. Li, "Neural Collaborative Subspace Clustering," Proc. ICML, pp.7384–7393, 2019.
- [17] X. Yang, C. Deng, F. Zheng, J. Yan, and W. Liu, "Deep Spectral Clustering using Dual Autoencoder Network," Proc. CVPR, pp.4066–4075, 2019.
- [18] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite," Proc. CVPR, pp.3354–3361, 2012.
- [19] J.-L. Blanco-Claraco, F.-Á. Moreno-Dueñas, and J. González-Jiménez, "The Málaga Urban Dataset: High-rate Stereo and LiDAR in a Realistic Urban Scenario," IJRR, vol.33, no.2, pp.207–214, 2014.
- [20] L. Jiang, S. Shi, X. Qi, and J. Jia, "GAL: Geometric Adversarial Loss for Single-View 3D-Object Reconstruction," Proc. ECCV, pp.802–816, 2018.
- [21] A. Schindler, "Vehicle Self-localization with High-precision Digital Maps," Proc. IV, pp.141–146, 2013.
- [22] M. Lundgren, E. Stenborg, L. Svensson, and L. Hammarstrand, "Vehicle Self-Localization using Off-the-Shelf Sensors and a Detailed Map," Proc. IV, pp.522–528, 2014.
- [23] R. Spangenberg, D. Goehring, and R. Rojas, "Pole-based Localization for Autonomous Vehicles in Urban Scenarios," Proc. IROS, pp.2161–2166, 2016.
- [24] W.R. Green and H. Grobler, "Normal Distribution Transform Graph-based Point Cloud Segmentation," Proc. PRASA-RobMech, pp.54–59, 2015.
- [25] H. Andreasson, M. Magnusson, and A. Lilienthal, "Has Somethong Changed Here? Autonomous Difference Detection for Security Patrol Robots," Proc. IROS, pp.3429–3435, 2007.

(2019年10月2日受付, 12月25日再受付,  
2020年2月19日早期公開)



松崎 康平 (正員)

平 22 東北大学工学部電気情報・物理工学科卒業。平 24 同大学大学院修士課程了。同年 KDDI (株) 入社。平 25 現 (株) KDDI 総合研究所に outward。令元名古屋大学大学院情報学研究所入学。現在に至る。画像検索や自己位置推定に関する研究開発に従事。



田坂 和之 (正員)

平 14 新居浜工業高等専門学校専攻科電子工学専攻了。平 16 奈良先端科学技術大学院大学修士課程了。同年 KDDI (株) 入社。平 22 奈良先端科学技術大学院大学博士後期課程了。現在, (株) KDDI 総合研究所にて, 画像認識やダイナミックマップに関する研究開発に従事。メディア認識グループグループリーダー。博士 (工学)。平 23 電子情報通信学会学術奨励賞受賞。



川西 康友 (正員)

平 18 京大・工・情報卒。平 24 同大大学院情報学研究科博士後期課程了。博士(情報学)。平 24 同大学術情報メディアセンター特定研究員。平 26 名大未来社会創造機構特任助教。平 27 同大情報科学研究科助教。平 29 同大情報学研究科助教。現在に至る。防犯カメラ・車載カメラ映像を対象とした、人物追跡・属性認識・行動認識などの人物画像処理に関する研究に従事。平 23 年度 PRMU 研究奨励賞受賞。IEEE ITS Society Nagoya Chapter Young Researcher Award 受賞。IEEE, 画像電子学会各会員。



村瀬 洋 (正員：フェロー)

昭 53 名大・工・電気卒。昭 55 同大大学院修士課程了。同年日本電信電話公社(現 NTT)入社。平 4 から 1 年間米国コロンビア大客員研究員。平 15 から名古屋大学大学院情報科学研究科教授。現在に至る。文字・図形認識, コンピュータビジョン, マルチメディア認識の研究に従事。工博。昭 60 電子情報通信学会学術奨励賞, 平 6 IEEE-CVPR 最優秀論文賞, 平 7 情報処理学会山下記念研究賞, 平 8 IEEE-ICRA 最優秀ビデオ賞, 平 13 高柳記念奨励賞, 平 13 本会ソサエティ論文賞, 平 14 電子情報通信学会業績賞, 平 15 文部科学大臣賞, 平 16 IEEE Trans. MM 論文賞, 平 22 前島密賞, 平 24 紫綬褒章, 他受賞。IEEE フェロー, 情報処理学会フェロー。