

移動端末における  
画像検索と自己位置推定のための  
大規模特徴データベースの小容量化

松崎 康平



# 要旨

情報通信技術の発展に伴い、スマートフォンを始めとする移動通信可能な端末（移動端末）が普及している．近年では、移動端末上でパターン認識を用いた様々なアプリケーションが実行されるようになった．その中に、画像検索や自己位置推定のように多数の特徴で構成されるデータベース（Database; DB）と特徴照合を行なうアプリケーションがある．これらの DB は多数の画像や広範囲の環境を表す大規模な特徴を含むため、容量が大きい．移動端末は計算資源が限られるため、大きな容量の DB は記憶装置に収まらない恐れがある．また、移動端末上で動作するアプリケーションは即応性が要求される場合が多く、特徴照合を高速に行なう必要がある．従って、これらのアプリケーションでは高速かつ高精度な特徴照合を実現しつつ、特徴 DB を小容量化することが重要な課題となる．

一般に、特徴照合においては特徴の照合精度とデータ容量はトレードオフの関係にあるため、高精度な照合と特徴の小容量化の両立は困難な問題である．同様に、特徴照合のための計算量を削減するほど照合精度は低下する恐れがあるため、高速かつ高精度な特徴照合もまた難しい．そのため、本論文では (i) DB 内のデータ数の削減、(ii) DB 内の各データの小さく化、(iii) 効率的なデータの照合、の 3 つを課題とし、これらを実現することによって上記の問題を解決する．これらの課題に対しては、(a) 特徴の選択、(b) 特徴の変換、(c) 類似尺度の変更、という観点で取り組む．(a) は特徴の集合から一部の特徴を選択し、(b) は特徴をより小さいデータ量の表現に変換し、(c) は変換されたままの特徴を照合するために類似尺度を変更するものである．そして、本論文で提案する手法により、スマートフォンによる画像検索とコネクテッドカーによる自己位置推定というアプリケーションにおいて、高速かつ高精度な照合と特徴 DB の小容量化を両立できることを示す．以下、各章の要旨をまとめる．

第 1 章では、本論文の研究背景、目的、位置付け、構成を述べる．画像検索や自己位置推定のように大規模な特徴 DB を必要とするアプリケーションが盛んに開発される中で、

それらを移動端末上で動作させることの難しさについて述べる．そして、本論文の目的は、移動端末上で動作するアプリケーションにおいて高速かつ高精度な照合と特徴 DB の小容量化を両立することであり、この目的を達成するために次の 3 つの研究を行なうことを述べる．

- 研究 1：特徴の選択に基づく画像の索引付け
- 研究 2：特徴の変換に基づく環境地図の圧縮
- 研究 3：類似尺度の変更にに基づく自己位置推定

第 2 章では、本論文で着目するアプリケーションである画像検索と自己位置推定の関連研究について概説する．画像検索については、移動端末で撮影した画像をクエリとする技術が広く利用されており、特に画像から抽出した局所特徴の集合を用いる方法が成功を収めている．自己位置推定については、カメラや LiDAR で計測したセンサデータと環境地図を用いて自己位置を推定する技術が広く利用されている．本章では、これらの技術に関連する先行研究についてまとめ、いくつかの代表的な手法を紹介する．

第 3 章では、第 1 の研究である特徴の選択に基づく画像の索引付けについて述べる．本研究では、課題として (i) DB 内のデータ数の削減を、アプリケーションとして Bag-of-Visual Words に基づく画像検索を想定する．この手法は元々、(b) 特徴の変換、(c) 距離尺度の変換を利用している．本研究では新たに (a) 特徴の選択を導入することにより、画像検索に有益な特徴のみを選択した DB を構築する手法を提案する．具体的には、特徴選択を特徴空間上での最大被覆問題として定式化し、特定の制約関数の下で最も高い信頼度を得る特徴の集合を求める．この定式化のために、画像検索手法における特徴の一致条件と整合するように設計された制約関数を提案する．実験により、提案手法が従来手法と比べて DB 容量および処理時間を増加させることなく、検索精度を改善することを示す．

第 4 章では、第 2 の研究である特徴の変換に基づく環境地図の圧縮について述べる．本研究では、課題として (ii) DB 内の各データの容量化を、アプリケーションとして 3D Normal Distributions Transform (NDT) による自己位置推定を想定する．3D NDT は元々、(b) 特徴の変換、(c) 距離尺度の変更に基づいて自己位置推定を行なう手法である．この手法では 3 次元点群を集約によって変換するが、変換後であっても DB 容量は依然として大きいという問題がある．この問題に対処するために、本研究では (b) 特徴の変換に基づいて正規分布のパラメータを量子化によって更に変換することにより、DB 容量を削減する手法を提案する．実験により、提案手法は標準的な 3D NDT と同程度の自己位置



推定精度と処理時間を達成しつつ、DB を小容量化することを示す。

第 5 章では、第 3 の研究である距離尺度の変換に基づく自己位置推定について述べる。本研究では、課題として (iii) 効率的なデータの照合を、アプリケーションとして 3 次元点群を用いた自己位置推定を想定する。3 次元点群を用いた代表的な自己位置推定手法は反復的な探索に基づくため、解の収束が遅い場合には処理時間が長くなる。非反復的な探索に基づく方法によって処理時間を安定化させることができるが、従来使用される類似尺度は計算量が大きいという問題がある。この問題に対処しつつ、高精度な特徴照合と DB の小容量化を実現するために、本研究では (c) 類似尺度の変更に基づいて効率的に計算可能な 2 値ベクトル間の Hamming 距離を類似尺度として使用する手法を提案する。実験により、提案手法は従来手法と同程度の自己位置推定精度を達成しつつ、処理時間を短くするとともに DB を小容量化することを示す。

第 6 章では、本論文を総括するとともに、今後の課題と展望を述べる。上記の 3 つの研究により、本論文で課題とした (i) DB 内のデータ数の削減、(ii) DB 内の各データの小容量化、(iii) 効率的なデータの照合を実現することができる。そしてこれにより、移動端末上で動作するアプリケーションにおいて高速かつ高精度な特徴照合を実現しつつ特徴 DB を小容量化することが可能となる。今後は本研究で得られた知見を基に、移動端末によって実世界のあらゆる情報の認識を可能にしたいと考えている。



# 目次

第 1 章	序論	1
1.1	背景	1
1.1.1	移動端末の普及	2
1.1.2	パターン認識を用いた移動端末上のアプリケーション	5
1.2	目的	7
1.2.1	特徴の選択	9
1.2.2	特徴の変換	9
1.2.3	類似尺度の変更	10
1.3	本論文の位置付け	11
1.3.1	研究 1：特徴の選択に基づく画像の索引付け	13
1.3.2	研究 2：特徴の変換に基づく環境地図の圧縮	14
1.3.3	研究 3：類似尺度の変更に基づく自己位置推定	15
1.4	本論文の構成	16
第 2 章	関連研究	17
2.1	画像検索	17
2.1.1	局所特徴	18
2.1.2	局所特徴に基づく画像対応付け	19
2.1.3	画像検索の枠組み	21
2.1.4	画像表現	22
2.1.5	画像検索のためのデータセット	26
2.1.6	移動端末における画像検索	27
2.2	自己位置推定	29

2.2.1	大域的自己位置推定 . . . . .	30
2.2.2	カメラを用いた局所的自己位置推定 . . . . .	32
2.2.3	LiDAR を用いた局所的自己位置推定 . . . . .	34
2.2.4	自己位置推定のためのデータセット . . . . .	37
2.2.5	移動端末における自己位置推定 . . . . .	39
<b>第 3 章</b>	<b>特徴の選択に基づく画像の索引付け</b>	<b>41</b>
3.1	はじめに . . . . .	42
3.2	特徴設計 . . . . .	44
3.3	画像検索手法 . . . . .	46
3.4	特徴の選択に基づく画像の索引付け . . . . .	48
3.4.1	定式化 . . . . .	48
3.4.2	制約関数 . . . . .	49
3.4.3	アルゴリズム . . . . .	50
3.5	評価実験 . . . . .	56
3.5.1	実験設定 . . . . .	57
3.5.2	方向に基づく制約の影響 . . . . .	57
3.5.3	特徴平均化および非バースト性選択の評価 . . . . .	58
3.5.4	従来の画像検索手法との比較 . . . . .	60
3.5.5	従来の特徴設計手法との比較 . . . . .	63
3.5.6	人工画像生成の効果分析 . . . . .	65
3.6	まとめ . . . . .	67
<b>第 4 章</b>	<b>特徴の変換に基づく環境地図の圧縮</b>	<b>69</b>
4.1	はじめに . . . . .	70
4.2	自己位置推定とデータ圧縮 . . . . .	71
4.2.1	3D Normal Distributions Transform . . . . .	71
4.2.2	ベクトル量子化に基づくデータ圧縮 . . . . .	72
4.3	特徴の変換に基づく環境地図の圧縮 . . . . .	73
4.3.1	ベクトル合成・分解 . . . . .	74
4.3.2	ベクトル正規化・逆正規化 . . . . .	77
4.3.3	ベクトル量子化・逆量子化 . . . . .	78

4.4	評価実験 . . . . .	80
4.4.1	実験設定 . . . . .	80
4.4.2	非占有ボクセルの ND パラメータの影響 . . . . .	82
4.4.3	コードブックの大きさの影響 . . . . .	83
4.4.4	NDT 地図の定性評価 . . . . .	89
4.4.5	3 次元点群地図に対する圧縮比 . . . . .	90
4.4.6	ボクセル解像度の影響 . . . . .	93
4.4.7	逐次的な自己位置推定 . . . . .	94
4.5	まとめ . . . . .	97
<b>第 5 章</b>	<b>類似尺度の変更に基づく自己位置推定</b>	<b>99</b>
5.1	はじめに . . . . .	100
5.2	ボリューム地図の生成 . . . . .	103
5.2.1	3 次元点群のモデル化 . . . . .	104
5.2.2	ボリュームデータの構築 . . . . .	105
5.3	自己位置推定 . . . . .	106
5.3.1	定式化 . . . . .	107
5.3.2	ボリュームデータ間の類似度計算 . . . . .	108
5.3.3	探索の効率化 . . . . .	109
5.4	評価実験 . . . . .	109
5.4.1	実験設定 . . . . .	109
5.4.2	サブボリュームおよびベクトル量子化の影響 . . . . .	111
5.4.3	自己位置推定性能および地図容量の評価 . . . . .	116
5.5	まとめ . . . . .	119
<b>第 6 章</b>	<b>むすび</b>	<b>121</b>
6.1	総括 . . . . .	121
6.2	今後の課題と展望 . . . . .	125
謝辞		127
参考文献		129

---

研究業績	151
------	-----

# 表目次

1.1	本論文で行なう研究のまとめ. 1 列目から 4 列目はそれぞれ本論文で行なう研究の索引, 各研究で取り組む課題の索引, 各研究で想定するアプリケーション, 元となる手法を表す. 5 列目以降は同行の研究で利用する観点を表し, 各セルにおいて “○” は元となる手法で同列の観点が元々利用されることを, “◎” は本論文における新規手法の利用を, “—” は不利用を表す. . . . .	12
2.1	2.1 節で紹介する関連研究のまとめ. 表の 1 列目は本節における項番号を, 2 列目は紹介する関連研究の技術分類を, 3 列目は技術の概要を表す.	18
2.2	2.2 節で紹介する関連研究のまとめ. 表の 1 列目は本節における項番号を, 2 列目は紹介する関連研究の技術分類を, 3 列目は技術の概要を表す.	30
3.1	提案手法と従来の画像検索手法の比較および提案手法の各要素技術の評価. KBC は特徴点の検出に基づく制約 (Keypoint-Based Constraint; KBC), VBC は特徴ベクトルの量子化に基づく制約 (Visual word-Based Constraint; VBC), OBC は方向に基づく制約 (Orientation-Based Constraint; OBC), FA は特徴平均化 (Feature Averaging; FA), NBS は非バースト性選択 (Non-Bursty Selection; NBS) を表す. 全ての手法に対して, 画像ごとに 900 個の特徴を DB に登録する. . . . .	62
3.2	人工画像生成数を変えた場合の提案手法の処理時間 [ms]. 各 $M$ における人工画像生成 (Synthetic), 人工画像からの局所特徴抽出 (Extract), 特徴ベクトルの量子化 (Quantize), Algorithm 1 に記述される特徴設計 (Design), 特徴平均化 (FA), 非バースト性選択 (NBS) のそれぞれの処理時間をまとめる. . . . .	66

4.1	異なるボクセル解像度での自己位置推定精度. 行見出し (A) はコードブック $\mathcal{C}$ を作成するための訓練用データの解像度 $r$ を, 列見出し (B) はベクトル量子化を行なう地図データの解像度 $r$ を表す. 要素は全てのコードブックの大きさ $K$ で測定した RMSE の平均値を表し, 非対角要素の括弧内に同行の対角要素との差分を示す. . . . .	95
4.2	自己位置推定性能および地図容量の比較. 進行方向への位置誤差 (Longitudinal error), その横方向への位置誤差 (Lateral error), 進行方向に対する方向誤差 (Orientation error), 処理時間 (Computational time), 経路 1 km あたりの地図容量 (Map data capacity) を示す. . . . .	96
5.1	自己位置推定性能および地図容量の比較. 進行方向への位置誤差 (Longitudinal error), それに対して横方向への位置誤差 (Lateral error), 進行方向に対する方向誤差 (Orientation error), 処理時間 (Computational time) の平均絶対値および経路 1km あたりの地図容量 (Map data capacity) を表す. . . . .	118



# 図目次

1.1	主な情報通信機器の保有状況. 当該比率は, 各年の世帯全体における各情報通信機器の保有割合を示す. (出典: 文献 [9] より作成) . . . . .	3
1.2	インターネット接続可能な端末数の推移および予測. (出典: 文献 [9] より作成) . . . . .	4
1.3	移動通信トラヒックの推移. 3 か月ごとの総ダウンロードトラヒックおよび総アップロードトラヒックを描画した結果を表す. (出典: 文献 [14] より作成) . . . . .	5
2.1	局所特徴に基づく画像対応付けによる Inlier の例. 白い円および線はそれぞれ特徴点の座標および結果として得られた Inlier を表す. . . . .	20
2.2	一般的な画像検索の枠組み. . . . .	21
2.3	Hamming Embedding の概要. この図は特徴空間を模式的に表現しており, “.” がクラスタ中心 (Centroid) を, “□” がある画像から抽出された特徴ベクトル (Descriptor) を, “×” が他の画像から抽出された “□” に対応する特徴ベクトル (Noisy versions of the descriptor) を表す. (出典: 文献 [73] より作成) . . . . .	23
2.4	Weak Geometric Consistency の概要. 参照画像中の扇形は, クエリ画像との間で対応する局所特徴が持つ方向の差を表現している. 各参照画像上のグラフはクエリ画像との間で対応する特徴の方向の差のヒストグラムを表す. (出典: 文献 [73] より作成) . . . . .	24
2.5	Stanford mobile visual searchデータセット* [90] に含まれる画像の例. . . . .	27

2.6	物体抽出に基づくランドマークの例. (a) は環境地図の模式図であり, 道路上の白線の始点と終点, および道路標識の頂点を可視化している. (b) はカメラ画像であり, (a) 中の赤い矩形の位置から撮影されたものである. (出典: 文献 [111] より作成)	32
2.7	領域検出技術に基づくランドマークの例. 色が付いた矩形領域はランドマークを, 線はそれらが対応付けられたことを意味する. (出典: 文献 [115] より作成)	33
2.8	3次元点群の可視化. 点の色は LiDAR によって計測された物体の反射強度を表す.	34
2.9	正射投影画像の例. 画素値は反射強度を表し, 黒い領域は破棄された物体の領域に対応する. (出典: 文献 [5] より作成)	35
2.10	3D NDT によって構築された環境地図の例. 白色の楕円体はボクセルごとの正規分布を表す. (出典: 文献 [6] より作成)	36
2.11	データの収集方法. (b) において, 赤い経路は RTK による補正が得られたことを, 青い経路は補正が得られなかったことを, 黒い経路は GPS が利用できなかったことを表す. (出典: 文献 [136] より作成)	37
2.12	KITTIデータセット <sup>†</sup> [136] に含まれる画像の例.	38
3.1	提案手法における特徴の信頼度の概要. ここでは, 円は特徴点 (Key-point) を表し, 四角は特徴ベクトルの量子化結果 (Quantization result) を表す. 同一の色は物体の同一位置から検出された特徴であることを意味する.	44
3.2	画像検索のベースライン手法の枠組み.	46
3.3	人工画像生成の概要. (b) における各画像下の 2 つの数値はそれぞれ仮想視点の仰角 [°] と方位角 [°] を表す.	52
3.4	非バースト性選択 (NBS) の概要. $s$ は特徴の信頼度を表す. また, 記号 “✓” および “×” はそれぞれ選択および破棄を表す.	56
3.5	異なるしきい値 $\varepsilon_\theta$ の下での MAP の 8 クラス平均. 提案手法を “Prop”, 方向に基づく制約 (Orientation-Based Constraint; OBC) を除外した提案手法を “Prop w/o OBC” と表記する.	58

3.6	異なる DB 容量の下での特徴平均化および非バースト性選択の影響. 提案手法を “Prop”, 提案手法から特徴平均化 (Feature Averaging; FA) を除外した手法を “Prop w/o FA”, 提案手法から非バースト性選択 (Non-Bursty Selection; NBS) を除外した手法を “Prop w/o NBS”, 提案手法から特徴平均化と非バースト性選択の両方を除外した手法を “Prop w/o FA nor NBS” と表記する. . . . .	59
3.7	異なる DB 容量の下での提案手法と従来の特徴設計手法の比較. IFS は Informative Feature Selection を, MA は Multiple Assignment を表す. MA は画像ごとに 900 個の特徴を抽出し, 各特徴を $k$ 個 ( $k = 1, \dots, 10$ ) の VW に割り当てる. IFS および提案手法は, MA における各 $k$ の場合と DB 容量が等しくなるように特徴を選択する. . . . .	64
3.8	人工画像生成に用いる仮想視点の個数と検索精度の関係. . . . .	65
4.1	提案する環境地図の圧縮手法の枠組み. . . . .	74
4.2	ブロックの模式図. 黒色の線で描かれた立方体がボクセルを, 緑色の楕円体がボクセルの持つ 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 パラメータを持たない非占有ボクセルであることを意味する. . . . .	75
4.3	DCN [173] の概要. 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 $k$ -平均法に基づいてクラスタリングする. . . . .	79
4.4	異なる $\delta$ の下での自己位置推定精度. . . . .	82
4.5	異なるコードブックの大きさでの性能評価. . . . .	84
4.6	$r = 1$ の場合のボクセルの占有／非占有推定の詳細. 占有／非占有に関する推定結果を Ground-truth を用いて TP (True Positive), FP (False Positive), FN (False Negative), TN (True Negative) に分類した場合のそれらの割合を $K$ ごとに示す. . . . .	86

4.7	$r = 2$ の場合のボクセルの占有／非占有推定の詳細. 占有／非占有に関する推定結果を Ground-truth を用いて TP (True Positive), FP (False Positive), FN (False Negative), TN (True Negative) に分類した場合のそれらの割合を $K$ ごとに示す. . . . .	87
4.8	$r = 3$ の場合のボクセルの占有／非占有推定の詳細. 占有／非占有に関する推定結果を Ground-truth を用いて TP (True Positive), FP (False Positive), FN (False Negative), TN (True Negative) に分類した場合のそれらの割合を $K$ ごとに示す. . . . .	88
4.9	NDT 地図の可視化. 最上段の行は無圧縮 (w/o VW) の NDT 地図を表し, それ以外の各行は異なるコードブックの大きさ $K$ の下で提案手法によって圧縮と復元を行なった NDT 地図を表す. 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. . . . .	89
4.10	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. . . . .	92
5.1	ボリューム地図の概要. 緑色, 青色の立方体はそれぞれボリューム, サブボリュームを表す. 灰色の立方体は占有ボクセルを表し, 非占有ボクセルは図示していない. . . . .	102
5.2	提案する地図生成手法および自己位置推定手法の枠組み. . . . .	103
5.3	ベクトル量子化の例. (b) はコードブックを用いて入力ベクトルを代表ベクトルに置き換えたサブボリュームを模式的に表現している. . . . .	105
5.4	異なるパラメータを用いた際の提案手法の性能評価. . . . .	112
5.5	非類似度関数 $d(\xi)$ の可視化. . . . .	115
5.6	3D NDT および提案手法の自己位置推定性能. . . . .	117
5.7	提案手法によって生じた大きな誤差の例. 直交する矢印が進行方向およびそれに対して横方向を表し, 青と赤の色がそれぞれ真値と推定値に対応する. . . . .	119

# 第 1 章

## 序論

本論文は，移動端末における画像検索および自己位置推定アプリケーションのための，大規模な特徴データベースの小容量化に関する一連の研究成果をまとめたものである．

本章では，初めに 1.1 節で本論文における研究の背景について概説する．次に，1.2 節で本論文の目的と課題について述べ，1.3 節で本論文の位置付けを説明する．最後に，1.4 節で本論文の構成をまとめる．

### 1.1 背景

情報通信技術の発展に伴い，スマートフォンを始めとする移動通信可能な端末（移動端末）が普及している．近年では，移動端末上でパターン認識を用いた様々なアプリケーションが実行されるようになった．パターン認識を用いたアプリケーションでは通常，カメラで撮影した画像のような生データを入力とし，その入力から特徴と呼ばれる識別に有効なパターンを抽出する．そして，識別のためのモデルを用いて特徴のスコアを計算することにより，パターンを識別する．パターン認識を用いたアプリケーションには，画像検索や自己位置推定のように，モデルだけでなく多数の特徴で構成されるデータベース（Database; DB）と特徴照合を行なうものが存在する．このようなアプリケーションは移動端末を用いてユーザが物理的に移動した先で計測したデータを，予め構築した DB と照合するという利用場面に適しているため，移動端末上で動作する多数の特徴照合手法が提案されている [1–8]．この利用場面としては，例えばユーザが市街地を歩いて移動する場合に，スマートフォンで撮影した画像を用いて画像検索や自己位置推定を行なう場面が挙

げられる．これらの DB は多数の画像や広範囲の環境を表す大規模な特徴の集合を含むため，容量が大きい．移動端末は計算資源が限られるため，大きな容量の DB は記憶装置に収まらない恐れがある．また，移動端末上で動作するアプリケーションは即応性が要求される場合が多く，特徴照合を高速に行なう必要がある．従って，これらのアプリケーションでは高速かつ高精度な特徴照合を実現しつつ，特徴 DB を小容量化することが重要な課題となる．本論文では移動端末上で動作する画像検索および自己位置推定アプリケーションに焦点を当て，それらの特徴 DB を小容量化する技術を提案する．

以下本節では，初めに近年の移動端末の普及について 1.1.1 項で概説する．そして，移動端末上で動作するパターン認識技術を用いた主要なアプリケーションとその課題について 1.1.2 項で述べる．

### 1.1.1 移動端末の普及

移動端末とは移動通信システムを介して通信可能な端末の総称であり，例として携帯電話やスマートフォン，タブレット，ウェアラブルコンピュータが挙げられる．移動端末の中で，最も普及しているものはスマートフォンである．Fig. 1.1 に示す我が国における情報通信機器の保有状況の調査 [9] によれば，2018 年にはスマートフォンの世帯保有率が 79.2 % となり，パソコンやタブレットなどと比べて最も普及していることがわかる．世界規模での調査 [9] によれば，スマートフォンの出荷台数は近年では年間約 15 億台に上る．この出荷台数はパソコンの約 2 億 5000 万台やタブレットの約 2 億台を大きく上回っており，世界的にもスマートフォンの普及が進んでいることがわかる．

また，インターネットに接続可能な端末数においても，スマートフォンは大きな割合を占める．Fig. 1.2 は，インターネットに接続可能な端末数の推移と予測 [9] を示している．この図において「通信」はスマートフォンや通信機器などの端末を表し，2018 年時点で最も稼働数が多いことが分かる．一方，2019 年以降は「自動車」，「医療」，「産業用途」の端末数が増加すると予測されている．近年では，自動車のような移動可能な装置に情報処理と通信の機能を持つ機器を搭載することにより，その装置を移動端末として扱うこともある [10, 11]．移動通信可能な自動車はコネクテッドカーと呼ばれ，車両の自動運転の実現や，緊急通報システムのような新たなサービスを創出するためのプラットフォームとして特に注目が集まっている．更に，コネクテッドカーは走行中にカメラや LiDAR (Light Detection and Ranging) のような多数のセンサから得られた情報を通信を介して収集す

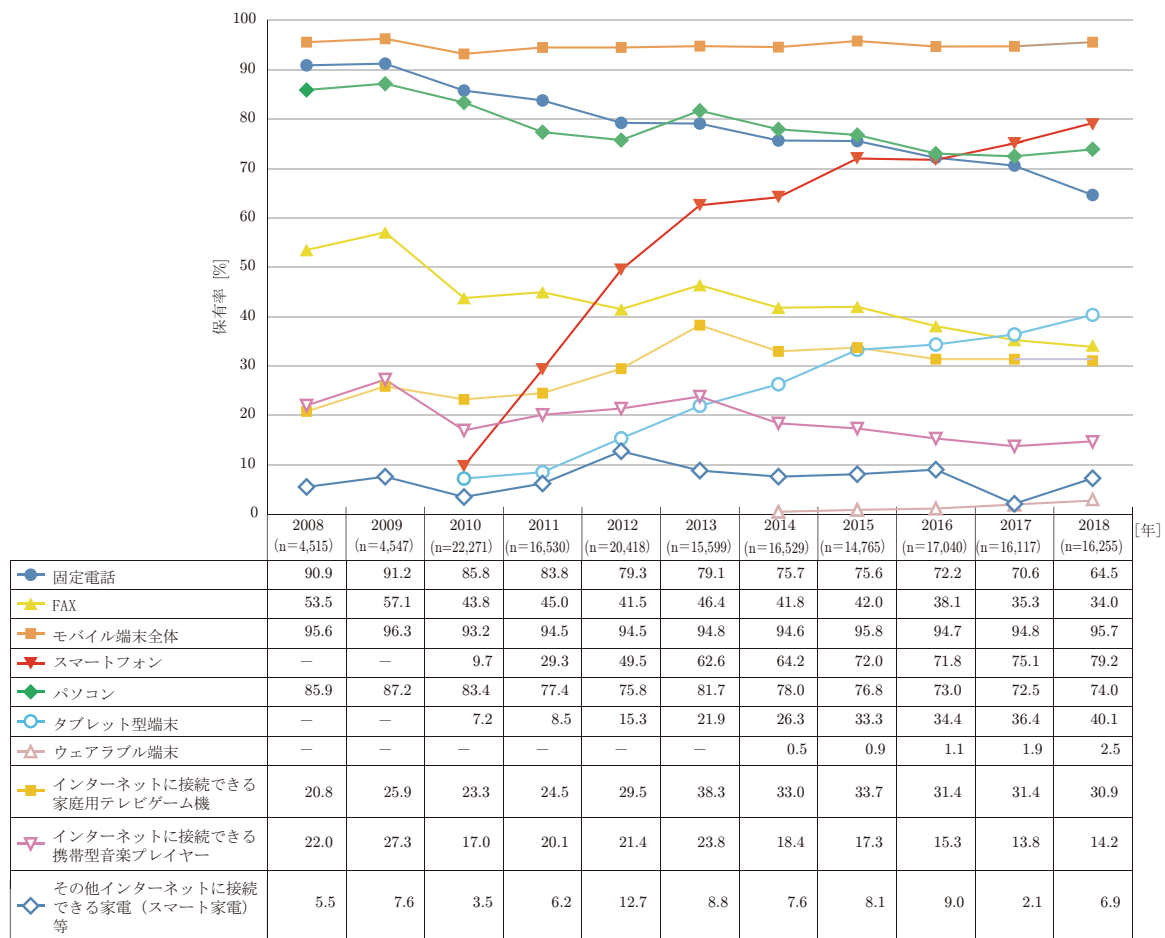


Fig. 1.1: 主な情報通信機器の保有状況. 当該比率は、各年の世帯全体における各情報通信機器の保有割合を示す. (出典：文献 [9] より作成)

ることによる、ビッグデータの蓄積に利用可能な端末としても期待されている. このような背景から、各自動車製造者が盛んにコネクテッドカーの開発を行っており、今後広く普及すると予想されている [12,13]. 以上の理由により、本論文では現在最も普及している移動端末であるスマートフォンと、今後の普及が予想される移動端末であるコネクテッドカーに注目する.

一方、移動端末の普及に伴い、移動通信ネットワーク上の通信量も急激に増加している. Fig. 1.3 は我が国における移動通信トラフィックの推移 [14] を示しており、ダウンロードトラフィックおよびアップロードトラフィックが大きく増加していることがわかる. 利用可能な帯域を超える通信を要求された場合には輻輳が発生し、パケットロスや遅延など様々な間

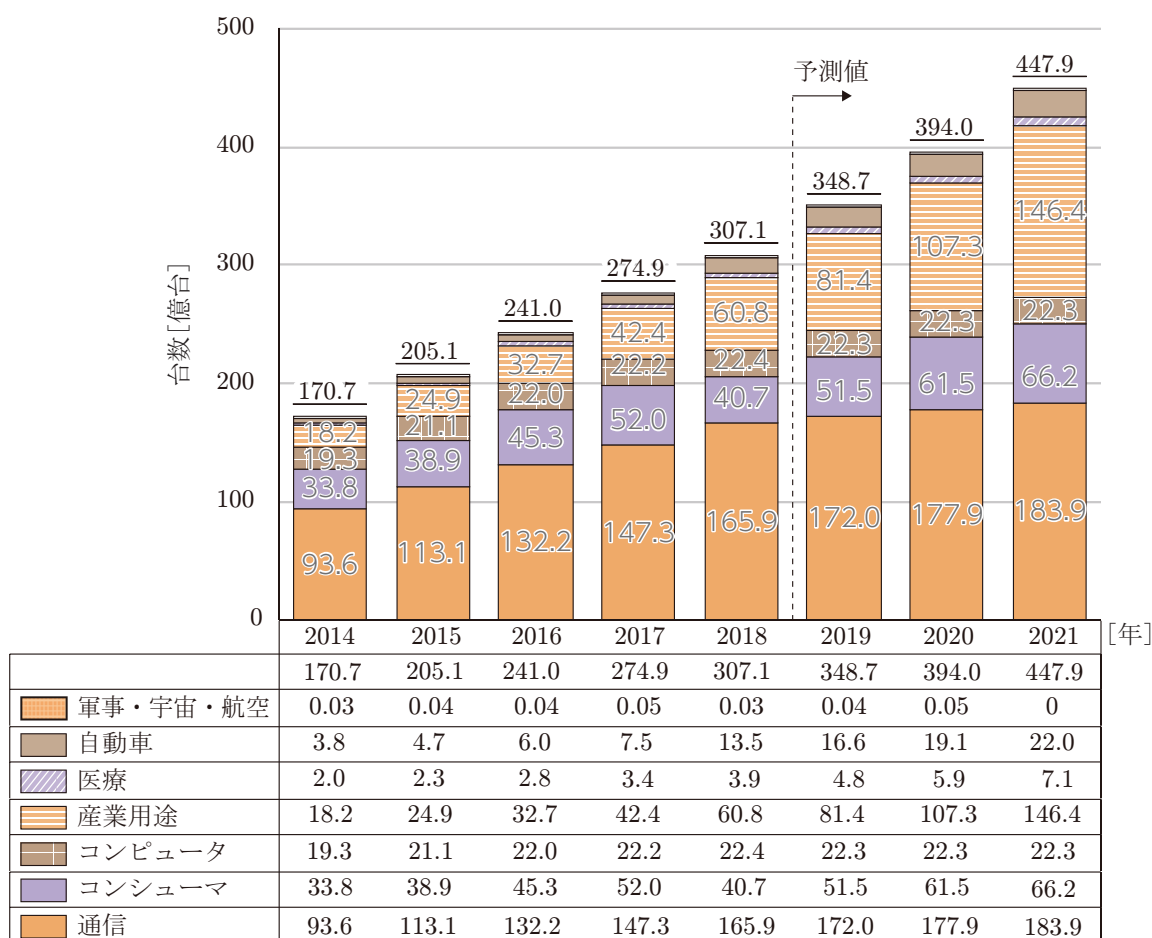


Fig. 1.2: インターネット接続可能な端末数の推移および予測. (出典：文献 [9] より作成)

題を引き起こす. この問題に対処するために, 輻輳発生回避 [15,16] や輻輳制御 [17,18] の技術が盛んに研究されている一方で, ネットワーク上の通信量自体を抑制する技術も重要な課題である. この課題に関して, 音声や画像などの汎用的なデジタル情報メディア (汎用メディア) のデータ量削減に関する研究が盛んに行なわれている [19–27]. しかし, 移動端末上のアプリケーションによっては任意のデータが伝送される可能性がある. 特に, 近年では第 5 世代 (5th Generation; 5G) 移動通信システムと呼ばれる高速大容量な通信システムの普及に伴い, 車両の自動運転や遠隔診療のような多様な利用場面が創出されると予想されている. これにより, 移動通信を介して伝送されるデータの種類も多様化すると考えられる. また, 5G は 4G と比べて 10 倍以上の最高通信速度を持つが, 本格的な IoT 社会の到来によって今後 10 年以内に現在の 1,000 倍を超える通信量が発生すると



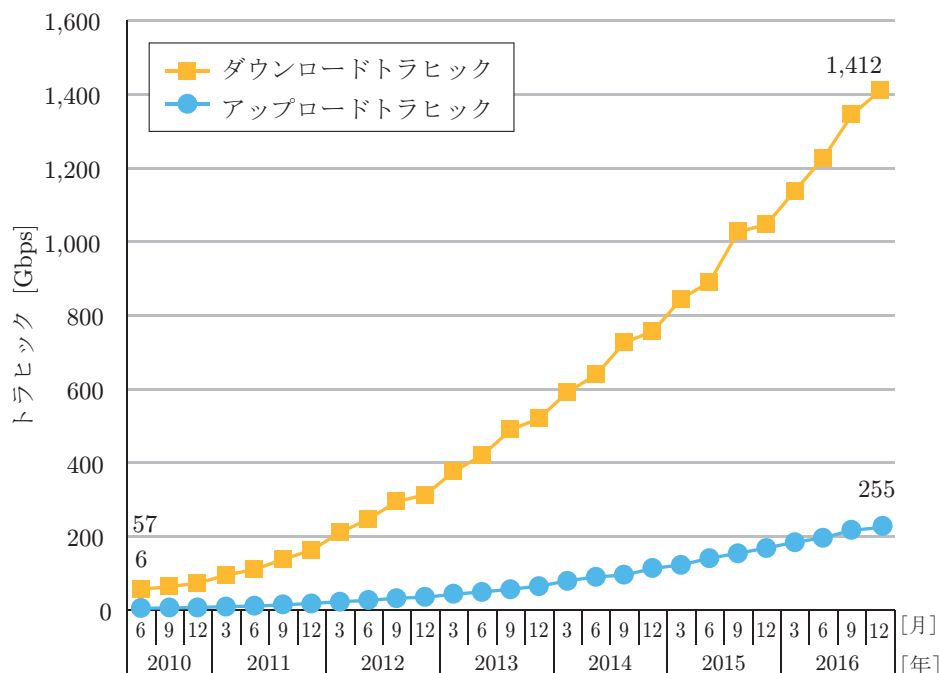


Fig. 1.3: 移动通信トラフィックの推移. 3 か月ごとの総ダウンロードトラフィックおよび総アップロードトラフィックを描画した結果を表す. (出典: 文献 [14] より作成)

予測されている [14]. そのため, 5G が普及した場合であっても, 依然として通信量の抑制が必要である. このようにして, 今後も移动通信システムの進歩が通信可能なデータ量を増加させる一方で, ユーザからの通信要求も多様化と増加を続けると予測される. 従って, 汎用メディア以外のデータに対するデータ量削減技術を確立することが, 重要な課題となる.

### 1.1.2 パターン認識を用いた移動端末上のアプリケーション

カメラや LiDAR のような視覚的センサを搭載した移動端末の普及に伴い, センサで計測したデータを入力とするアプリケーションが盛んに開発されている. 例えば Google Lens\*では, スマートフォンで撮影した画像を入力としてそこに写る外国語の自動翻訳や, 植物や動物などの種類の特定, 類似商品の検索などの様々なアプリケーションを実現している. また, Project Apollo†と呼ばれる自動運転プラットフォームを開発する計画にお

\* <https://lens.google.com/intl/en/>

† <https://apollo.auto/index.html>

いては、障害物の知覚や信号の認識、自己位置推定などの自動運転に必要な様々なアプリケーションの開発が行なわれている。このようなアプリケーション上では、パターン認識を用いた入力データの認識が重要である。パターン認識を用いたアプリケーションの種類は多岐にわたり、例えば物体検出 [28, 29]、物体分類 [30, 31]、画像検索 [32, 33]、自己位置推定 [34, 35] が挙げられる。このようなアプリケーションは、ユーザが物理的に移動しながら実行し、逐次的に認識結果を得るような状況で利用されることが多く、要件として即応性が求められる。例えば、移動しながら取得されるカメラ画像を入力として物体検出を行なうアプリケーションを実行する場合には、対象がユーザの視野から外れる前に認識することが望ましい。

パターン認識では通常、カメラや LiDAR などのセンサから取得された画像や点群のような生データを入力とし、その入力から特徴と呼ばれる識別に有効なパターンを抽出する。この処理は特徴抽出と呼ばれ、高次元の入力ベクトル空間から低次元の特徴ベクトル空間への写像とみなすことができる。パターンの識別には、機械学習によって訓練されたモデルと特徴を比較する方法が主流である。近年では、深層学習技術の発展に伴い、パターン認識を用いる多くのアプリケーションにおいて深層学習モデルを利用することが一般的となっている。しかし、深層学習に基づくアプリケーションは計算量が多い場合が多く、即応性の実現が難しいという問題がある。また、モデルのパラメータ数が多い場合にはアプリケーションのデータ量が大きくなる恐れがある。このような問題に対処するための代表的な技術として、モデルの量子化 (quantization) [36, 37] や分解 (factorization) [38, 39]、剪定 (pruning) [40, 41]、蒸留 (distillation) [42, 43] が挙げられる。このような技術によってモデルの計算量やパラメータ数を削減することにより、移動端末上で物体検出や領域分割などのアプリケーションを高速に実行することが可能になる [44–48]。また、その場合のアプリケーションのデータ量は移動端末の記憶容量に収まる程に小さい。従って、パターン認識を用いる一部のアプリケーションは移動端末上で実行可能である。

しかし、入力から抽出された特徴と DB 内の多数の特徴を照合するアプリケーションは、パターンを識別するためのモデルだけでなく、多数の特徴が格納された DB を必要とするため、移動端末上で実行することが難しい。高速な特徴照合を実現するためには、この DB は全て移動端末上の記憶装置に収めておく必要がある。また、移動端末の記憶容量を上回る場合には DB を格納することができないため、外部のサーバとの通信が必要とな

る．多数の特徴との照合を行なうアプリケーションの代表例として，画像検索と自己位置推定が挙げられる．画像検索では，多数の参照画像から抽出した特徴を DB とし，その DB とクエリ画像から抽出した特徴を照合することにより，クエリ画像と一致する参照画像を見つけることができる．自己位置推定では，大規模な環境を計測したデータから抽出した特徴を DB とし，現在位置の周辺から抽出した特徴を照合することにより，環境内での自己位置を推定することができる．DB の容量は照合の対象となるデータの規模に比例し，大規模な DB においてはテラバイト (Terabyte; TB) 以上の規模になることもある [49, 50]．自己位置推定では現在位置の周辺の環境地図のみを必要とするため，外部のサーバに DB を置き，必要なデータを逐次的にダウンロードしながらアプリケーションを実行することも可能である．しかし，容量の大きい DB を長期間ダウンロードし続ける場合，ネットワーク上の通信量へ与える影響が大きくなる．従って，DB 内の多数の特徴と照合を行なうアプリケーションでは，端末に格納する場合と外部サーバからダウンロードする場合のいずれにおいても DB の小容量化が重要な課題となる．

本論文では前項で述べた理由により，移動端末としてスマートフォンとコネクテッドカーに注目する．そして，それらの移動端末において特徴 DB と照合を行なうアプリケーションである画像検索と自己位置推定に焦点を当て，高速かつ高精度な認識を実現しつつ特徴 DB を小容量化する技術について検討する．

## 1.2 目的

本論文の目的は，移動端末上で動作するアプリケーションにおいて，高速かつ高精度な特徴照合を実現しつつ特徴 DB を小容量化することである．ここで，本論文で検討する特徴照合について説明する．前節で述べたように，本論文では画像検索と自己位置推定に焦点を当てる．これらのアプリケーションでは，2 つの特徴集合間の特徴照合処理が重要な要素技術となる．この特徴照合処理は特徴の対応付け，対応する特徴間の類似度の計量，類似度の総和の計算，で構成される．すなわち，画像検索では画像を特徴の集合として表現し，クエリ画像との間で類似度が最大となる参照画像を見つけること，一方自己位置推定では実環境を特徴の集合として表現し，環境地図と入力データとの間で類似度が最大となる位置姿勢を見つけることがこれに該当する．以下では，この特徴照合処理を定式化する．

2つの特徴集合を  $A = \{\mathbf{a}_i | i = 1, \dots, N\}$  と  $B = \{\mathbf{b}_j | j = 1, \dots, M\}$  と表記する．ここで、 $\mathbf{a}_i \in A$  および  $\mathbf{b}_j \in B$  はベクトルの形式で表現される特徴である．特徴の対応付けは、一方の特徴集合の要素に対してもう一方の特徴集合の部分集合を求める処理である．部分集合を求める際には、例えば最近傍探索のような特定の方法が利用される．説明を簡略化するために、ここでは各  $\mathbf{a}_i$  に集合  $B$  内の1つの要素  $\mathbf{b}_j$  を対応付ける方法を想定する．そして、この対応付けを実現する関数を  $f$  とし、 $\mathbf{b}_j = f_{A \rightarrow B}(\mathbf{a}_i)$  と表記する．類似度の総和は、対応付けられた特徴ベクトル間の類似度を累積加算することで計算される．この時、集合  $A$  と  $B$  の類似度  $S(A, B)$  は、次式で表現することができる．

$$S(A, B) \propto \sum_{\mathbf{a}_i \in A} g(\mathbf{a}_i, f_{A \rightarrow B}(\mathbf{a}_i)), \quad (1.1)$$

ここで  $g$  は対応する特徴間の類似度を計量する関数であり、任意の類似尺度が利用可能であるものとする．本論文で検討する特徴照合とは、式 (1.1) に示す類似度  $S$  を用いて入力側のデータと DB 側のデータを照らし合わせる処理である．一例を挙げれば、入力側の特徴の集合  $A$  と DB 側の特徴の集合  $B, C$  がある場合に、 $S(A, B)$  と  $S(A, C)$  の大きさを比較することである．これにより、 $A - B$  間の類似度と  $A - C$  間の類似度の大小関係を判断することができる．

一般に、特徴照合においては特徴の照合精度とデータ量はトレードオフの関係にあるため、高精度な照合と DB の小容量化の両立は困難な問題である．同様に、特徴照合のための計算量を削減するほど照合精度は低下する恐れがあるため、高速かつ高精度な特徴照合の実現もまた難しい．本論文では (i) DB 内のデータ数の削減、(ii) DB 内の各データの小容量化、(iii) 効率的なデータの照合、の3つを課題とし、これらを実現することによって上記の問題を解決する．課題 (i) では DB 内のデータ数を削減することによって DB の小容量化を実現するとともに、照合するデータ数が少なくなることによる特徴照合の高速化も期待される．課題 (ii) では DB に含まれる個別のデータを小容量化することによって、DB 全体の小容量化を実現する．課題 (iii) では計算方法を変更することによって、特徴照合の高速化を実現する．従って、課題 (i), (ii), (iii) を実現することができれば、本論文の目的である高速かつ高精度な特徴照合および特徴 DB の小容量化を達成することができる．本論文ではこれらの課題に対し、(a) 特徴の選択、(b) 特徴の変換、(c) 類似尺度の変更、という観点で取り組む．(a) は特徴の集合の中から一部の特徴を選択し、(b) は特徴をよりデータ量の小さい表現形式に変換し、(c) は変換されたままの特徴を照合するた

めに類似尺度を変更するものである。式 (1.1) を用いて説明すると、(a) は集合  $A$  や  $B$  の要素数を削減し、(b) は特徴ベクトル  $\mathbf{a}_i$  や  $\mathbf{b}_j$  の表現形式を変換し、(c) は関数  $g$  における類似尺度を変更する。

以下本節では、これらの 3 つの観点についてそれぞれ 1.2.1 項、1.2.2 項、1.2.3 項で説明する。

### 1.2.1 特徴の選択

1 つ目の観点は、特徴の選択に着目することである。特徴の選択とは、特徴の集合から、認識に対して特に有益な部分集合を選択する処理である。機械学習の分野においては、特徴ベクトルから一部の要素を選択する処理を、特徴の選択と呼ぶことがある [51, 52]。しかし、本論文における特徴の選択は、複数の特徴ベクトルの中から特定のベクトルを選択する処理を意味することに留意されたい。この処理を DB 側の特徴集合に適用した場合、DB の小容量化と照合の高速化という 2 つの効果が期待される。すなわち、DB 側の特徴集合をその部分集合に置き換えることにより、DB 容量を削減することができる。また、一般に特徴の対応付け関数  $f$  の計算量は特徴集合の要素数に応じて増加する。従って、DB 側の特徴集合の要素数を削減することによって、特徴照合の高速化が期待される。なお、この処理を入力側の特徴集合に適用した場合にも、式 (1.1) の計算量が削減されることによって照合が高速化される。

ただし、この処理では特徴の選択を行なう前後で特徴集合間の類似度の大小関係が変化し、特徴照合の精度が低下する恐れがある。このような変化が起こった場合、アプリケーションの認識結果に悪影響を及ぼす。従って、特徴の選択を行なう場合には、その前後で任意の特徴集合間の類似度の大小関係が変わらない手法を用いることが望ましい。このような手法は、特徴集合が冗長性を持つ場合に、非冗長な特徴を選択することで実現可能である。そして、このような手法を用いた場合、特徴照合の精度を維持しつつ DB の小容量化と処理の高速化を実現することができる。

### 1.2.2 特徴の変換

2 つ目の観点は、特徴の変換に着目することである。特徴の変換とは、特徴の表現形式を変換する処理である。変換方法によっては特徴のデータ量が増加する可能性もあるが、

本論文ではデータ量を削減する方法に注目する。この処理を DB 側の特徴集合に適用した場合、DB の小容量化の効果が期待される。変換方法には様々な種類がある。例えば、ある特徴をより低次元の特徴に変換する方法は次元削減 (dimension reduction) と呼ばれる。特徴を離散的な近似値に置き換える方法は量子化 (quantization) と呼ばれ、特に要素ごとに量子化する場合はスカラ量子化、複数の要素をまとめて量子化する場合にはベクトル量子化と呼ばれる。複数の特徴を、それらの統計量を表す単一の特徴に変換する集約 (aggregation) と呼ばれる方法も存在する。

ただし、これらの変換された特徴は、類似度の計量関数  $g$  を適用できない恐れがある。すなわち、特徴の変換によって特徴の表現形式が変化することにより、元の類似尺度を適用することが不適切になる可能性がある。そのような場合には、類似尺度を適切なものに変更するか、特徴を変換前の表現形式に逆変換し、元の関数  $g$  を用いるという 2 通りの対処法が存在する。前者については、1.2.3 項で説明する。後者の場合、変換前の特徴に対する類似尺度を適用できる。言い換えれば、特徴照合の精度を維持しつつ、記憶装置への格納や通信による伝送を行なう際の DB の容量を削減することができる。

### 1.2.3 類似尺度の変更

3 つ目の観点は、類似尺度の変更に着目することである。類似尺度の変更とは、変換された特徴に対して特徴間の類似度を定義する尺度を適切なものに変更する処理である。本論文では、ある特徴に対して適切な類似尺度が存在し、特徴の変換によってそれを適用することが不適切になる場合を想定する。この場合、類似尺度を変更することで変換後の特徴間の類似度を適切に計量できるようになる。更に、一般に特徴のデータ量が小さいほど距離の計算量が低下する。従って、特徴の変換と類似尺度の変更を組み合わせることにより、DB の小容量化と照合の高速化という 2 つの効果が期待される。

ただし、この処理で不適切な類似尺度に変更してしまった場合には特徴照合の精度が低下し、アプリケーションの認識結果に悪影響を及ぼす可能性がある。従って、1.2.1 項の場合と同様に、変更の前後で任意の特徴集合間の類似度の大小関係が変わらない類似尺度が求められる。このような類似尺度を用いることにより、特徴照合の精度を維持しつつ DB の小容量化と処理の高速化を実現することができる。

## 1.3 本論文の位置付け

前節までに述べたように、本論文では多数の特徴との照合を行なうアプリケーションにおいて、高速かつ高精度な特徴照合および特徴 DB の小容量化の実現を目的とする。実際のアプリケーションとして、スマートフォンを用いた画像検索とコネクテッドカーを用いた自己位置推定に着目する。それぞれの応用において多数の特徴との照合を行なう際には、画像検索については BoVW (Bag-of-Visual Words) [53] が、自己位置推定については 3D NDT (Normal Distributions Transform) [6] が広く利用される手法の 1 つであるため、これらを元となる手法とする。ただし、3D NDT による自己位置推定は反復的な最適化に基づくため、解の収束が遅い場合には処理時間が長くなるという問題がある。そのため、自己位置推定については 3D NDT だけでなく、この問題に対処することが可能な新規手法を提案する。そして、これらの手法においてそれぞれ (a) 特徴の選択, (b) 特徴の変換, (c) 類似尺度の変更, の観点に基づいて, (i) DB 内のデータ数の削減, (ii) DB 内の各データの小容量化, (iii) 効率的なデータの照合, の課題を実現することにより、高速かつ高精度な照合を実現しつつ特徴 DB を小容量化することを目指す。本論文では、これらの課題 (i), (ii), (iii) への取り組みを、それぞれ研究 1, 研究 2, 研究 3 と呼ぶものとする。

Table 1.1 に、本論文で行なう研究をまとめる。研究 1 では課題として (i) DB 内のデータ数の削減を、アプリケーションとして画像検索を、元となる手法として BoVW を想定する。この手法は元々、(b) 特徴の変換, (c) 類似尺度を利用している。そして、本研究では新たに (a) 特徴の選択を導入することにより、更なる性能改善を目指す。研究 2 では課題として (ii) DB 内の各データの小容量化を、アプリケーションとして自己位置推定を、元となる手法として 3D NDT を想定する。この手法は元々、(b) 特徴の変換, (c) 類似尺度の変更を利用している。ここで、元々利用される特徴の変換の種類は、複数の特徴を単一の統計量に変換する集約である。本研究ではそれとは異なる種類の変換、すなわち離散的な近似値に置き換える量子化を新たに導入することにより、DB の小容量化を目指す。なお、環境地図を表す DB において、特徴は実世界の地形に対応している。この DB に対して (a) 特徴の選択を導入した場合、環境地図の変化検出や、障害物の回避といった自己位置推定以外のアプリケーションに悪影響を及ぼす恐れがある。そのため、本研究におい

Table 1.1: 本論文で行なう研究のまとめ. 1 列目から 4 列目はそれぞれ本論文で行なう研究の索引, 各研究で取り組む課題の索引, 各研究で想定するアプリケーション, 元となる手法を表す. 5 列目以降は同行の研究で利用する観点を表し, 各セルにおいて “○” は元となる手法で同列の観点が元々利用されることを, “◎” は本論文における新規手法の利用を, “-” は不利用を表す.

研究	課題	アプリケーション	元となる手法	観点		
				(a) 特徴 の選択	(b) 特徴 の変換	(c) 類似尺 度の変更
1	(i)	画像検索	BoVW [53]	◎	○	○
2	(ii)	自己位置推定	3D NDT [6]	-	◎	○
3	(iii)	自己位置推定	新規手法	-	◎	◎

ては (a) 特徴の選択は導入しないものとする. 研究 3 では課題として (iii) 効率的なデータの照合を, アプリケーションとして自己位置推定を想定し, 地図生成と自己位置推定のための新たな手法を提案する. この手法は (b) 特徴の変換, (c) 類似尺度の変更を利用することにより, 高速かつ高精度な自己位置推定および DB の小容量化を目指す. 本研究においては主に (c) 類似尺度の変更に着目し, (b) 特徴の変換は補助的に利用するものとする. なお, 研究 2 の場合と同じ理由で (a) 特徴の選択は導入しないものとする.

研究 2 および研究 3 ではともに自己位置推定に関する手法を提案するが, それぞれの手法は異なる意義を持つ. 研究 2 で元となる手法の 3D NDT は, 最も広く利用される自己位置推定手法の 1 つである. 提案手法によって小容量化された DB は従来の 3D NDT による DB と互換性を持つように設計されるため, 既存の DB や関連するソフトウェアを利用する多くのユーザに恩恵を与えることができる. 一方, 研究 3 では 3D NDT が持つ問題に対処した新たな手法を提供するという意義がある. すなわち, 3D NDT は反復的な方法に基づいて位置姿勢を探索するため, 解の収束が遅い場合には処理時間が長くなるという問題がある. それに対し, 新たな手法では予め設定した範囲の位置姿勢パラメータを高速に探索することにより, 安定的に処理時間を短くすることができる.

以下本節では, 上記の研究 1, 2, 3 についてそれぞれ 1.3.1 項, 1.3.2 項, 1.3.3 項で概要を述べる.



### 1.3.1 研究 1：特徴の選択に基づく画像の索引付け

本研究では、アプリケーションとしてスマートフォン上での局所特徴に基づく画像検索に着目する。画像検索手法としては、BoVW に基づく手法を想定する。BoVW による画像の索引付けでは、初めに画像から多数の局所特徴を抽出し、画像をそれらの集合として表現する。ここで、局所特徴は位置座標を表す特徴点と、特徴を表す特徴ベクトルを持つ。その後、特徴ベクトルを VW (Visual Word) と呼ばれる代表ベクトルへ量子化し、画像を VW のヒストグラムとして表現する。BoVW では、参照画像とクエリ画像の間で特徴点が物体の同一位置から検出され、特徴ベクトルが同一の VW に量子化される局所特徴のみが検索に貢献する。

物体をある 1 つの視点から撮影した画像を参照画像とする場合、スマートフォンで撮影したクエリ画像との間で視点が異なり、検索精度が低下するという問題がある。この問題に対しては、同一の物体を複数の視点から撮影した画像集合（多視点画像）から抽出した特徴を DB に登録することで対処できるが、その場合には DB の容量が大きくなる。これらの問題に対処するために、本研究では BoVW における画像の索引付けに対して特徴の選択処理を導入する。すなわち、多視点画像から抽出した特徴の集合からその部分集合を選択し、集合の要素数を減らすことによって DB の容量を削減する。ここで、無作為に選択した場合には特徴照合の精度を低下させる恐れがあるため、特徴照合に関して特に有益な特徴を選択することにより、精度を維持しつつ容量を削減する。

具体的には、本研究では特徴点の検出の再現性と特徴ベクトルの量子化の再現性に基づく特徴信頼度と、それを用いた特徴選択手法を提案する。すなわち、多視点画像を構成する各画像にわたって特徴点が物体の同一位置から検出され、特徴ベクトルが同一の VW に量子化される頻度を計算し、それを信頼度として使用する。そして、信頼度が高い特徴を優先的に選択した DB を構築することによって、より少数の特徴のみを用いて視点変化に頑健な特徴照合を実現する。提案手法では、特徴選択を特徴空間上での最大被覆問題として定式化し、特定の制約関数の下で最高の信頼度を得る特徴の集合を求める。この目的のために、BoVW における特徴の一致条件に対して整合性を持つ制約関数を設計する。また、最大被覆問題は NP 困難であり、多項式時間で実行不可能であることから、貪欲法に基づく近似解法を提案する。

評価実験として、スマートフォンで撮影されたクエリ画像を用いて、画像検索の精度を

評価する．実験結果として，従来手法と比べて DB 容量や検索時間を増加させずとも，提案手法により検索精度が改善することを示す．更に，提案手法によって DB を構築するための処理時間を測定し，多項式時間で実行可能であることを確認する．

### 1.3.2 研究 2：特徴の変換に基づく環境地図の圧縮

本研究では，アプリケーションとしてコネクテッドカー上での環境地図に基づく自己位置推定に着目する．環境地図の生成および自己位置推定の手法としては，3D NDT を想定する．3D NDT では，環境地図を生成する際に 3 次元点群を入力とし，粗いボクセルを用いてそれを分割する．そして，各ボクセルに含まれる 3 次元点群を多変量の正規分布表現に変換し，得られた正規分布パラメータの集合を環境地図とみなす．この変換は複数の特徴を単一の統計的な特徴に置き換えるため，集約と呼ばれる変換に分類される．これにより，環境地図の容量を元の 3 次元点群の容量に比べて 100 分の 1 以下にすることができる．

しかし，大規模な環境を想定する場合には，3D NDT による圧縮をしても，環境地図の容量は数 TB から数百 TB の規模になる恐れがあるため，更なる容量削減が必要である．この問題に対処するために，3D NDT によって構築される環境地図に対して特徴の変換を導入する．本研究では 3D NDT が元々行なっている集約の変換に加えて，特徴を離散的な近似値に置き換える，量子化と呼ばれる変換を利用する．すなわち，集約と量子化という異なる種類の複数の変換を組み合わせることにより，自己位置推定の精度を維持しつつ元の 3 次元点群に比べて大幅に地図の容量を削減する．

具体的には，本研究では正規分布パラメータの集合として表現される環境地図に対する特徴変換手法として，環境地図を構成する正規分布パラメータをベクトル表現に変換し，ベクトル量子化によってデータを圧縮する手法を提案する．ただし，ボクセルごとにベクトル量子化を適用すると，個々のボクセルが持つパラメータ数が少ないために圧縮比は低い．そこで，圧縮比の改善のために占有／非占有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手法を提案する．このとき，高次元ベクトルに対しても量子化誤差が少ないベクトル量子化を実現するために，コードブックの学習に深層クラスタリング技術を適用する．自己位置推定の実行時には，圧縮前の表現に逆変換することによって 3D NDT による環境地図との互換性を担保する．これにより，3D NDT に基づいて特徴照合を行なうことが可能に

なる。

評価実験として、車両に搭載された LiDAR で計測した 3 次元点群を用いて、自己位置推定の精度および環境地図の容量を評価する。実験結果として、無圧縮の場合の 3D NDT と比べて提案手法が自己位置推定の精度や処理時間を維持しつつ、環境地図の容量を削減することを示す。

### 1.3.3 研究 3：類似尺度の変更に基づく自己位置推定

本研究では 1.3.2 項と同様に、アプリケーションとしてコネクテッドカー上での 3 次元点群に基づく自己位置推定に着目する。3 次元点群に基づく自己位置推定における最も代表的な手法は、ICP (Iterative Closest Point) [54] と 3D NDT である。しかし、これらの手法は環境地図の容量が大きいことに加え、自己位置推定手法が反復的な方法に基づいているため、解の収束が遅い場合には処理時間が長くなるという問題がある。

非反復的な解の探索方法として、標本化された位置姿勢変換パラメータを用いてデータを変換し、類似度を比較するものがある。しかし、ICP や 3D NDT で使用される類似度の計量にはそれぞれ最近傍点間の Euclidean 距離の計算や、点と正規分布を用いた尤度の計算が含まれており、計算量が大きい。そのため、この方法を適用した場合には即応性の実現が困難である。この問題に対処するために、類似尺度の変更ににより高速な類似度の計量を実現する。本研究では、3 次元点群間の類似尺度を、従来の手法で使用されるものから、2 値ベクトル間の Hamming 距離の総和へ変更する。

具体的には、本研究では 3 次元点群を 2 値ベクトルの集合に変換する地図生成手法と、高速に計算可能な類似尺度を用いる自己位置推定手法を提案する。提案する地図生成手法においては、初めにボクセルを用いて 3 次元点群を分割し、2 値占有モデルに従って各ボクセルの占有情報を 2 値で表現する。そして、複数のボクセルの占有情報を空間的位置関係に従って結合することにより、環境地図を 2 値ベクトルの集合に変換する。更に、2 値ベクトルをベクトル量子化によって圧縮することにより、環境地図の容量を削減する。自己位置推定の実行時には、環境地図を圧縮前の 2 値ベクトル表現に逆変換し、2 値ベクトルの集合向けに変更した類似尺度を用いて特徴を照合する。すなわち、類似尺度として 2 値ベクトル間の Hamming 距離の総和を使用し、予め標本化した離散的な位置姿勢変換パラメータを探索する。これにより、環境地図の容量を削減しつつ、高速かつ高精度な自己位置推定を実現する。

評価実験として，1.3.2 項と同様に，車両に搭載された LiDAR で計測した 3 次元点群を用いて，自己位置推定の精度および環境地図の容量を評価する．実験結果として，従来手法と比べて提案手法が同程度の自己位置推定精度を維持しつつ，環境地図の容量を削減することを示す．更に，提案手法による自己位置推定の処理時間を測定し，実時間処理が可能であることも示す．

## 1.4 本論文の構成

本論文は 6 つの章で構成される．第 1 章は序論であり，本論文における研究の背景および本論文の目的，本論文の位置付けについて述べた．第 2 章では関連研究として，画像検索と自己位置推定に関する従来手法およびデータセットについて紹介する．第 3 章では，研究 1 として特徴の選択に基づく画像の索引付け手法を提案する．第 4 章では，研究 2 として特徴の変換に基づく環境地図の圧縮手法を提案する．第 5 章では，研究 3 として類似尺度の変更に基づく自己位置推定手法を提案する．最後に，第 6 章で本論文を総括し，今後の課題と展望について述べる．

## 第 2 章

# 関連研究

本章では、本論文で着目するアプリケーションである画像検索と自己位置推定の関連研究およびデータセットについて各々概説する。これらのアプリケーションは移動端末との親和性が高く、様々な利用例が存在する。例えば画像検索では、スマートフォンを持って移動した先で気になる物体を撮影した場合に、その場で情報を検索することができる。あるいは自己位置推定では、コネクテッドカーで移動している最中に自身の位置情報を高精度に知ることができる。以下では、2.1 節で画像検索について、2.2 節で自己位置推定について述べる。

### 2.1 画像検索

本節では、主にスマートフォンでの使用を想定した画像検索技術について紹介する。画像検索とは、画像 DB の中からクエリに合致する画像を見つける技術である。クエリとしては、文字入力や音声入力によるキーワードやカメラで撮影された画像を利用することが代表的である。移動端末上で動作するアプリケーションにおいては、端末に搭載されたカメラを用いて簡便に画像を撮影できることから、画像をクエリとする技術が広く利用されている。画像をクエリとする場合、画像から抽出した特徴を照合し、クエリ画像と類似度が高い画像を検索する方法が成功を収めている。ただし、例えば色ヒストグラム [55] のような大域特徴は遮蔽や大きさの違いによって検索精度が低下するという問題がある。この問題を解決するために、SIFT (Scale-Invariant Feature Transform) [56,57] に代表される局所特徴の集合によって画像を表現する方法が提案されている。この方法では画像を特徴の集合として表現し、特徴集合間の類似度を比較することで画像を検索する。ただ

Table 2.1: 2.1 節で紹介する関連研究のまとめ. 表の 1 列目は本節における項番号を, 2 列目は紹介する関連研究の技術分類を, 3 列目は技術の概要を表す.

項	分類	概要
2.1.1	局所特徴	画像中の局所的な小領域から, 回転や大きさの変化, 照明変化, 雑音に対して頑健な特徴を抽出する.
2.1.2	画像対応付け	複数の画像を画素単位で対応付ける. 代表的な手法では, 局所特徴を用いて特徴点の位置合わせを行なう.
2.1.3	画像検索	局所特徴を用いた画像対応付けの考えに基づいて, DB 内からクエリ画像と合致する画像を見つける.
2.1.4	画像表現	高速な画像検索を実現するために, 画像から抽出された局所特徴の集合を異なる表現に変換する.
2.1.5	データセット	画像検索の性能を評価するためのデータセット. 移動端末を用いた画像検索に特化したデータセットもある.

し, 特徴集合間の類似度の計算は計算量が大きいため, 高速な検索を実現するために特徴の集合を異なる表現に変換する方法が採用される.

以下本節では, 2.1.1 項から 2.1.5 項のそれぞれにおいて局所特徴, 局所特徴に基づく画像対応付け, 画像検索の枠組み, 画像表現, 画像検索のためのデータセットについて関連研究を紹介する. これらの各項で紹介する技術の概要は, Table 2.1 にまとめられる. 最後に, 2.1.6 項において移動端末上で動作する画像検索アプリケーションの課題について述べる.

### 2.1.1 局所特徴

局所特徴は画像全体ではなく, 特徴点の周辺の小領域のみから抽出される特徴である. 最も代表的な局所特徴の抽出手法は SIFT である. 以下では, SIFT のアルゴリズムについて簡潔に述べる. 初めに, 入力画像を様々なスケールパラメータ  $\sigma$  を用いて平滑化した画像を作成し, それらの差分として表現される DoG (Difference of Gaussian) 画像上の極値を検出する. 次に, 得られた極値から, 雑音に強く, 角や縁のような輝度値の変化が大きい領域に位置する点のみを選択し, 特徴点 (Keypoint) とする. その後, 特徴点の周辺の画素から勾配方向ヒストグラムの最大を算出し, 方向 (Orientation) とする. 最

後に、特徴点を中心として方向が揃うように回転し、スケールに応じた小領域から 128 次元の特徴ベクトルを算出する。ここで、照明変化に対して頑健にするためにベクトルの正規化を行なう。結果として、入力画像からは特徴点の座標、スケール、方向、特徴ベクトルの組として表現される局所特徴の集合が抽出される。このようなアルゴリズムにより、SIFT は画像の回転や大きさの変化、照明変化、雑音に対して頑健な局所特徴を抽出することができる。SIFT の高速化や精度改善を目的とする様々な手法 [58–63] が提案されているが、それらは見えの変化に対して頑健な局所特徴を抽出するという共通の方針に基づいている。また、特徴ベクトルを実数値ではなく 2 値で表現することにより、計算効率性の改善や特徴のデータ量削減を実現する手法も提案されている [64–66]。

### 2.1.2 局所特徴に基づく画像対応付け

画像対応付け [67,68] とは、同一の物体を写した複数の画像を画素単位で対応付ける技術である。画像を局所特徴の集合として表現した場合、局所特徴の特徴点の位置合わせを行なうことによって画像対応付けを実現できる。この技術は、結果として特徴集合間の類似度が得られることから、画像検索や物体認識に利用される。本論文においては移動端末によって同一の物体を異なる視点から撮影する場合を想定する。この場合、画像の幾何的な変化は相似変換や Affine 変換、Homography 変換のような幾何変換モデルで近似することができる。従って、同一の物体を写した 2 枚の画像から局所特徴を抽出し、それらに対応付けた特徴対応の集合から幾何変換モデルを推定することによって、特徴点の位置合わせを実現することができる。しかし、局所特徴の対応付けには誤りが含まれる可能性があるため、正しい幾何変換モデルを推定することは難しい。

この問題に対処し、局所特徴に基づく画像対応付けを実現する代表的な手法が、RANSAC (RANdom SAmple Consensus) [69] による手法である。この手法は幾何変換モデルの推定と、正しい対応と誤った対応の分類を同時に行なうことにより、画像の幾何的な変化に頑健な画像対応付けを実現する。具体的には、初めに無作為に選択した少数の対応を用いて幾何変換モデルの仮説を立て、その仮説に従う対応 (Inlier) とそれ以外の外れ値 (Outlier) に分類することによって、仮説の妥当性を検証する。この仮説と検証の処理を繰り返すことにより、最も妥当な幾何変換モデルとそれに従う対応を推定することができる。以下では、RANSAC による画像対応付けのアルゴリズムを簡潔に説明する。初めに、1 組の局所特徴の集合から特徴ベクトルの対応付けによって特徴対応の集

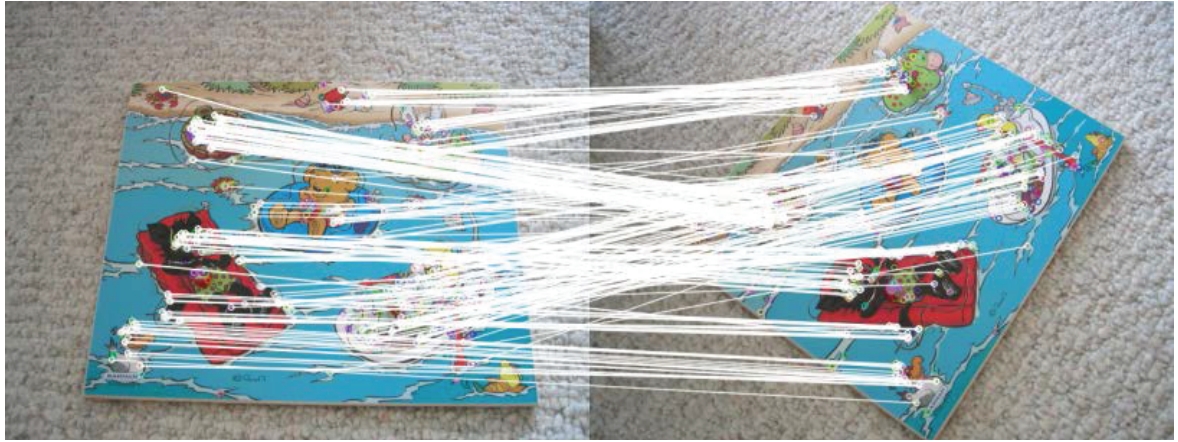


Fig. 2.1: 局所特徴に基づく画像対応付けによる Inlier の例. 白い円および線はそれぞれ特徴点の座標および結果として得られた Inlier を表す.

合を構築する. 特徴対応の集合から無作為に  $n$  組を選択し, 幾何変換モデルのパラメータを計算する. ここで,  $n$  は想定する幾何変換モデルの自由度以上の拘束条件を得るための最小の整数であり, 例えば相似変換なら  $n = 2$ , Affine 変換なら  $n = 3$ , Homography 変換なら  $n = 4$  となる. 次に, 得られた幾何変換モデルに従って一方の画像の特徴点をもう一方の画像の座標系へ再投影する. そして, 対応する特徴点間の誤差がしきい値以内であれば Inlier, そうでなければ Outlier に分類する. 上記の処理を反復し, 得られる Inlier の個数が最大となる場合の幾何変換モデルと特徴対応の分類を出力する. この時に得られる Inlier の個数は, 特徴集合間の類似度とみなすことができる.

Fig. 2.1 に, 幾何変換モデルに Homography 変換を想定した場合の Inlier の例を示す. ここでは, 同一の物体を写した 2 枚の画像間で局所特徴に基づく画像対応付けを行なっている. 白い円および線はそれぞれ特徴点の座標および結果として得られた Inlier を表す. この図より, 画像間で物体の回転や大きさの変化があるにも関わらず, 物体の同一位置から抽出された局所特徴同士が線で結ばれていることがわかる. この Inlier の個数が, 画像間の類似度となる. すなわち, この類似度は特徴ベクトルと特徴点の座標の両方が一致する局所特徴の組を数えたものである. 言い換えれば, 局所特徴に基づく画像対応付けは, 特徴ベクトルと特徴点の両方を利用して類似度を求める手法である.



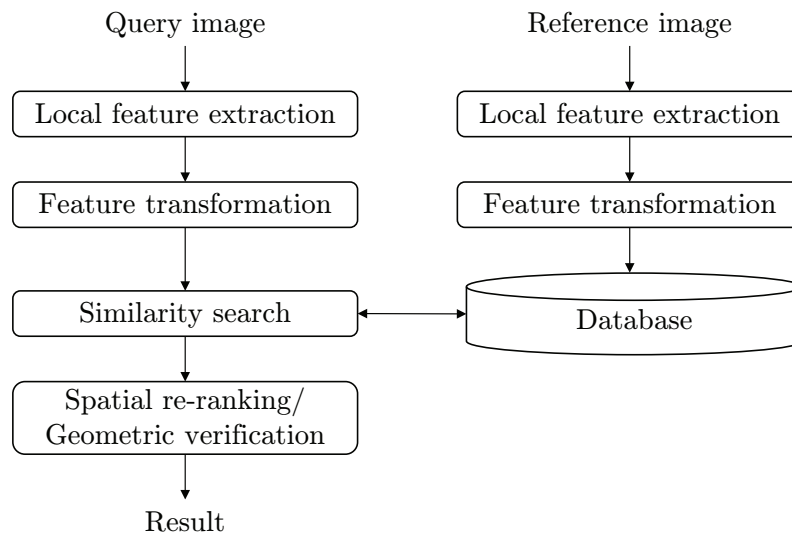


Fig. 2.2: 一般的な画像検索の枠組み.

### 2.1.3 画像検索の枠組み

局所特徴に基づく画像検索は、あるクエリ画像と DB 内の参照画像の間で特徴集合の類似度を計算し、類似度が高い参照画像を見つける。最も単純な検索方法は、クエリ画像と参照画像の間で画像対応付けを網羅的に行なうことであろう。しかし、RANSAC による画像対応付けは計算量が大きいため、大規模な画像 DB に適用することは現実的ではない。特徴集合間の類似度を高速に計算するために、局所特徴の集合を異なる表現に変換した上で、類似度を計算する手法が提案されている。これらの手法は、特徴点の座標を利用せず、特徴ベクトルのみを利用することで高速な検索を実現している。ただし、特徴点の座標を利用した方が類似度の精度が高いため、高速な検索による類似度が高い少数の画像に対してのみ画像対応付けを適用することで、検索時間を抑制しつつ精度を改善する。

Fig. 2.2 に、一般的な画像検索の枠組み [70–72] を示す。初めに、クエリ画像 (Query image) と参照画像 (Reference image) のそれぞれから局所特徴を抽出する (Local feature extraction)。そして、局所特徴の集合を変換した特徴によって、それぞれの画像を表現する (Feature transformation)。このようにして画像を異なる特徴で表現することは画像表現と呼ばれ、ここでは任意の画像表現手法が適用可能である。その後、参照画像を表す特徴を集積した DB とクエリ画像を表す特徴の間で類似検索 (Similarity search) を行ない、類似度が高い順に参照画像を並べた初期ランキングを得る。その後、初期ランキン

グの上位に位置する少数の参照画像に対してのみ、クエリ画像との間で画像対応付けによる類似度を求め、それに従ってランキングを修正する。この処理は特徴点の座標、すなわち空間的情報 (Spatial information) を利用することから空間的再ランキング (Spatial re-ranking) と呼ばれる。あるいは、幾何変換モデル (Geometric transformation model) を利用することから幾何検証 (Geometric verification) とも呼ばれる。幾何検証はランキングの修正だけでなく、クエリ画像と参照画像が同一の物体を写すか否かを検証するためのしきい値判定を含む場合もある。ここでは、参照画像から得られる最大の類似度がしきい値未満である場合、クエリ画像に合致する画像は存在しないと判定する。これにより、DB の中にクエリ画像と同一の物体を写す参照画像が存在しない場合であっても、偽陽性の誤検出を防ぐことができる。

#### 2.1.4 画像表現

本項では、効率的な検索を実現するために、局所特徴の集合を変換する画像表現手法を紹介する。最も代表的な手法は、BoVW (Bag-of-Visual Words) [53] である。この手法は、局所特徴が持つ特徴点の座標や方向、スケールを破棄し、特徴ベクトルのみを用いて画像を表現する。具体的には、特徴ベクトルを VW (Visual Words) と呼ばれる代表ベクトルの集合を用いてベクトル量子化する。この VW は多数の訓練用特徴ベクトルを  $K$  個のグループにクラスタリングして得られる、クラスタ中心の集合である。最終的に、1 枚の画像を表現する局所特徴の集合を、VW のインデックスのヒストグラムへ変換する。画像間の類似度は、2 つのヒストグラムの余弦類似度である。この余弦類似度を効率的に計算するために、BoVW では転置インデックスと呼ばれる DB を構築する。すなわち、通常は参照画像のインデックスごとに、その画像を表現する VW のインデックスのリストを格納する。それに対し、転置インデックスは VW のインデックスごとに、その VW で表現される参照画像のインデックスのリストを格納する。検索の実行時には、クエリ画像を表現する VW のインデックスに対応する参照画像のインデックスにのみ、類似度スコアを投票する。これにより、全ての参照画像に対して網羅的に余弦類似度を計算する場合に比べて、効率的に類似度を計算することができる。このようにして、BoVW は小容量な特徴 DB を用いて高速な画像検索を実現する。

しかし、BoVW では余弦類似度の識別性と、ベクトル量子化の再現性がトレードオフの関係にある。すなわち、クラスタ数  $K$  が小さいほど密なヒストグラムを構築し、結果

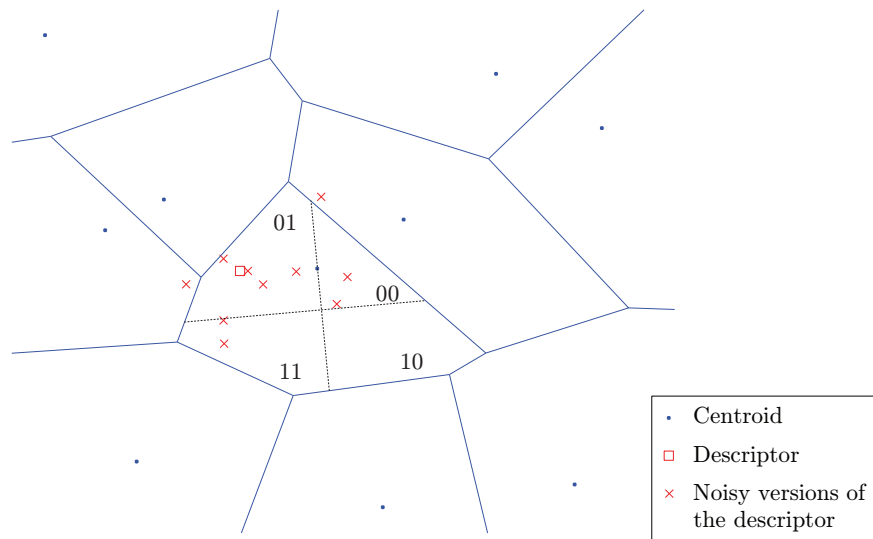


Fig. 2.3: Hamming Embedding の概要. この図は特徴空間を模式的に表現しており, “.” がクラスタ中心 (Centroid) を, “□” がある画像から抽出された特徴ベクトル (Descriptor) を, “×” が他の画像から抽出された “□” に対応する特徴ベクトル (Noisy versions of the descriptor) を表す. (出典: 文献 [73] より作成)

として余弦類似度による識別性が低下する. 一方, クラスタ数  $K$  が大きいほど疎なヒストグラムを構築するために識別性が改善するが, ベクトル量子化の再現性が低下する. すなわち, わずかな視点変化や照明変化などに起因する特徴ベクトルの変化によって, 実際には対応している局所特徴が同一の VW に割り当てられなくなる. これは結果として, 余弦類似度による識別性を低下させる. このような理由により, BoVW はクラスタ数  $K$  を小さくしても大きくしても検索精度が限定的となる.

この問題に対処するために, 比較的小さい  $K$  を用いることでベクトル量子化の再現性を担保しつつ, 高精度な検索を可能にする手法が提案された [72, 73]. この手法は HE (Hamming Embedding) と WGC (Weak Geometric Consistency) で構成される. HE はある特徴ベクトルを変換した 2 値ベクトルを用いて, より高精度な局所特徴の対応付けを実現する. この 2 値ベクトルは, VW を表す特徴空間上での Voronoi セル上における特徴ベクトルの位置を表すように設計された射影関数を用いて, 特徴ベクトルを変換することで得られる. Fig. 2.3 に, HE による 2 値ベクトルの概略を示す. この図は特徴空間を模式的に示し, 破線で区切られた空間とそこに記述されたビット列が, 特徴ベクトルを変換して得られる 2 値ベクトルを表している. 類似度スコアを投票する際には, 同一の

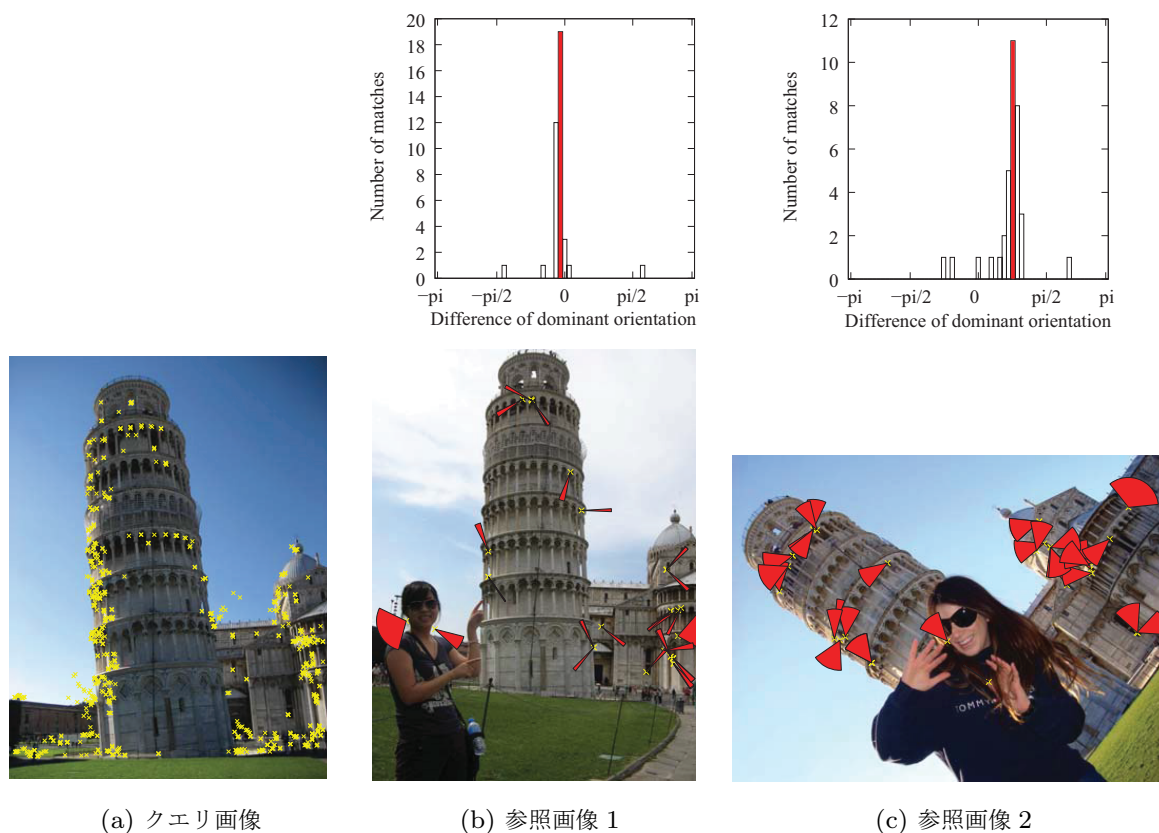


Fig. 2.4: Weak Geometric Consistency の概要. 参照画像中の扇形は、クエリ画像との間に対応する局所特徴が持つ方向の差を表現している. 各参照画像上のグラフはクエリ画像との間に対応する特徴の方向の差のヒストグラムを表す. (出典：文献 [73] より作成)

VW に割り当てられたクエリ側と参照側の局所特徴が持つ 2 値ベクトル間で Hamming 距離を計算し、その距離がしきい値未満または  $k$  近傍以内である場合にのみ、局所特徴を対応付ける. 正しい対応の Hamming 距離は小さくなる可能性が高いため、この対応付けは誤対応を除去する処理とみなすことができる. そして、クエリ側と対応付けられた局所特徴を持つ参照画像にのみ類似度スコアを投票することにより、検索精度が改善される.

WGC は局所特徴が持つ方向とスケールを用いて、簡易的な幾何検証を高速に実現する. 始めに、上述した HE によってクエリ側と参照側で局所特徴を対応付ける. そして、対応付けられた局所特徴が持つ方向の差、およびスケールの比をビンに投票したヒストグラムを構築する. Fig. 2.4 に、方向の差の場合のヒストグラムの概略を示す. 参照画像中に示した扇形は、クエリ画像との間に対応する局所特徴が持つ方向の差を表現している.

参照画像 1 はクエリ画像と比べて物体の回転がほとんどないため、扇形の中心角が小さい場合が多いが、参照画像 2 は物体の回転角が大きいため、扇形の中心角も大きくなる。各参照画像に対する方向の差のヒストグラムを見ると、物体の回転角に近いビンに度数が集中していることがわかる。局所特徴の方向は回転不変性を持つため、局所特徴が正しく対応付けられた場合には、その差は物体の回転角に等しくなると考えられる。この考えに従い、度数の最大値をとるビンに属する対応以外を除去することにより、誤りの可能性が高い対応を類似度スコアの投票から除外する。スケールの比の場合も同様である。これにより、方向の差とスケールの比の整合性に基づく簡易的な幾何検証が実現され、検索精度を改善することができる。

また、よりデータ量の小さい特徴を用いて高精度な検索を実現するために、局所特徴の集合を集約し、単一の大域特徴として表現する手法も提案されている。代表的な手法として、Fisher vector [74, 75] と VLAD (Vector of Locally Aggregated Descriptors) [76] が挙げられる。局所特徴の集合を Fisher vector として表現する場合、各局所特徴は確率密度関数  $p$  に従い独立に生成されると仮定する。典型的に  $p$  には混合正規分布が選択され、各正規分布は局所特徴が各 VW に割り当てられる確率と想定される。この混合正規分布においては、混合重みは局所特徴が  $K$  通りの VW に割り当てられる確率、平均は各 VW に割り当てられる特徴ベクトルの平均ベクトル、分散共分散の対角成分は各 VW に割り当てられる特徴ベクトルの次元ごとの分散を表すベクトルとして表現される。これらのパラメータは、訓練用の多数の局所特徴を用いて最尤推定を行なうことによって得られる。そして、この確率密度関数  $p$  から導出される勾配ベクトルを  $K$  個連結することにより、Fisher vector とする。ただし、Fisher vector は一般に高次元となるため、データ量が大きく、検索速度が低速になるという問題がある。これらの問題に対処するために、Perronnin ら [75] は Fisher vector を 2 値ベクトルに変換する手法を提案した。

VLAD は、初めに小さな  $K$  (例えば  $K = 16$ ) を用いて特徴ベクトルを VW に割り当てる。各 VW において、それに割り当てられた特徴ベクトルと VW 自身の残差ベクトルを計算する。その後、各 VW において残差ベクトルの総和を取り、最後にその総和ベクトルを自身の  $L_2$  ノルムを用いて正規化する。そして、 $K$  個の正規化された総和ベクトルを連結したベクトルを合成し、単一の大域特徴ベクトルとする。更に、主成分分析によってこの大域特徴ベクトルを次元削減することにより、データ量を削減する。DB を構築する際には、参照側のベクトルに直積量子化を適用することにより、更にデータ量を削減す

る。画像検索の実行時には、クエリ側のベクトルと、参照側の直積量子化されたベクトルの間で高速に類似度を計算するために、非対称距離計算手法 [77] を適用する。このようにして、これらの手法では小容量な特徴 DB を用いて高速かつ高精度な画像検索を実現することができる。

近年では、CNN (Convolutional Neural Network) を用いた画像表現による手法が優れた検索精度を達成している [32]。多くの手法は CNN を用いて抽出した単一の大域特徴によって画像を表現する [78, 79]。これらの手法では畳み込みフィルタによって局所的な領域におけるパターンを記述し、Pooling によってそれらを集約したベクトルを抽出する。従って、CNN に基づく大域特徴は局所特徴の集約とみなすことができ、これによって遮蔽や視点変化に頑健な検索が可能となる。画像表現に CNN を用いる場合、ImageNet [80] のような大規模なデータセットを用いて訓練された CNN に対して、Fine-tuning を行なうことによって検索精度を改善する手法も検討されている [81–83]。これらの手法では同一の物体を表す特徴ベクトル間の距離が近く、また異なる物体を表す特徴ベクトル間の距離が遠くなるように Fine-tuning を行なう。このような Fine-tuning の多くは、Siamese network を用いて計量される Contrastive loss [84] や、Triplet loss [85] を用いて実現される。また、CNN を用いて局所特徴を抽出する手法も存在する [86, 87]。これらの手法では明示的に特徴点とその周辺領域から抽出される局所特徴の集合として画像を表現する。Noh らの研究 [87] においては、局所特徴を用いて CNN に基づく大域特徴と同等以上の検索精度を達成することが示されている。しかし、CNN に基づく特徴抽出は計算量が大きい [88] ため、移動端末上で実時間処理を実現することは困難である。

### 2.1.5 画像検索のためのデータセット

画像検索の性能を評価するために、多くのデータセットが公開されている [70, 73, 88–91]。しかし、一般的なデータセットは必ずしも移動端末で撮影した画像をクエリとしないため、移動端末を用いた画像検索の評価には適さない。移動端末を用いた画像検索の評価のために設計されたデータセットに、SMVS (Stanford Mobile Visual Search) データセット\* [90] がある。このデータセットは、物体を正面視点から写す参照画像と、移動端末を用いてその物体を撮影したクエリ画像で構成されている。Fig. 2.5(a) に参照画像の

---

\* <https://purl.stanford.edu/rb470rw0983/>



Fig. 2.5: Stanford mobile visual searchデータセット\* [90] に含まれる画像の例.

例を, Fig. 2.5(b) にクエリ画像の例を示す. 参照画像はインターネット上で収集した画像であり, クエリ画像は実際の物体を様々な条件の下で撮影した画像である. Fig. 2.5(b) より, クエリ画像には, 参照画像と比べた場合の照明変化や遮蔽, 視点変化などによる外乱が生じていることがわかる. 撮影する物体のクラスは, “Book covers”, “Business cards”, “CD covers”, “DVD covers”, “Landmarks”, “Museum paintings”, “Print documents”, “Video frames” の 8 つである. これらのクラスは, 商品や名刺, 建築物, テキスト, 映像などの様々な物体を認識する画像検索アプリケーションの性能を評価するために設計されている. Landmarks を除く全てのクラスは, 100 枚の参照画像と, 4 通りの移動端末でそれぞれ 100 枚撮影された計 400 枚のクエリ画像からなる. Landmarks クラスのみ, 500 枚の参照画像と 1 通りの移動端末で撮影された 500 枚のクエリ画像からなる. 従ってこのデータセットは, 全てのクラスを合わせて 1,200 枚の参照画像と 3,300 枚のクエリ画像を含む. 以上のように, 様々な物体クラスにわたって, 移動端末で撮影されたクエリ画像とその参照画像が提供されていることから, SMVS データセットは移動端末を用いた画像検索の評価に適している.

### 2.1.6 移動端末における画像検索

本項では, 移動端末上で動作する画像検索アプリケーション, すなわちモバイル画像検索 (Mobile Visual Search; MVS) アプリケーションとその課題について整理する. 高解



像度カメラを搭載した移動端末の進歩に伴い、画像検索技術に基づく MVS アプリケーションに注目が集まっている。これらのアプリケーションでは事前に構築された DB から、移動端末のカメラで撮影された特定の物体の情報を検索することができる。MVS をサーバクライアントシステムとみなすいくつかの研究 [1, 2] が存在し、これらは画像の伝送を想定する。従って画像のデータ量を削減することによって通信量を抑制することができるが、その検討は本論文の範囲外である。一方、MVS をスタンドアローンシステムとみなす場合、それは DB を含むモバイルアプリケーションとして実装され、移動端末上に搭載される。一般に、ユーザはデータ量の大きいアプリケーションをインストールすることを忌避する傾向があるため、アプリケーションのデータ量は小さい方が好ましい。MVS アプリケーションにおける DB 以外の構成要素（例えば検索関数）のデータ量を大幅に削減することは通常困難である。そのため、アプリケーションのデータ量を小さくするための最も単純な方法は、DB 容量を制限することである。しかし、高い検索精度と小さい DB 容量の両立は、困難な課題である。

典型的な MVS の枠組みにおいては、画像の検索および索引付けに局所特徴が利用される。最も代表的な局所特徴は、SIFT である。しかし、計算資源が乏しい移動端末上で実時間処理を実現するには、SIFT 特徴の抽出アルゴリズムは計算量が大きい。この点をふまえて、SIFT のような実数値特徴ではなく、BRISK [64] や ORB [65], FREAK [66] のような 2 値特徴を用いた手法が注目を集めている。このような 2 値特徴は、実数値特徴と比べて、性能を維持しつつ 1 桁から 2 桁倍高速に処理することが可能である。従って、2 値特徴は移動端末上で動作する画像検索アプリケーションに対して、実数値特徴よりも適している。

多くの画像検索の枠組みにおいて、局所特徴は BoVW [53] や Fisher vector [74, 75, 92], VLAD [76] のような画像表現へ符号化される。DB 内の参照画像と移動端末で撮影されたクエリ画像間の類似度は、それらの表現を比較することによって計算される。そして、類似度が高い参照画像に対して幾何検証を行なうことによって、特徴点对応付けの成否を判定する。Fisher vector や VLAD は複数の局所特徴を単一の画像表現に集約することによって、高い検索精度と小さい DB 容量を両立している。しかし、これらの手法は画像から抽出される全ての局所特徴を集約するため、画像単位の対応付けはできても、局所特徴単位での対応付けを実現することができない。そのため、幾何検証を含む画像検索の枠組みに対してこれらの手法を適用する場合、局所特徴単位での対応付けを行なうために、追



加の情報が必要である。それに対し、BoVW は VW の対応付けに基づいて局所特徴単位での対応付けをすることができるため、Fisher vector や VLAD に比べて必要な DB 容量が小さくなる。

従って、移動端末上で動作する MVS アプリケーションに対しては、2 値特徴と BoVW に基づく画像検索手法が最も適切と考えられる。BoVW では通常、画像から抽出された全ての特徴を用いて DB を構築する。しかし、BoVW では DB 容量が画像ごとの特徴数に依存するため、特徴数が多いほど DB が大容量化するという問題がある。

## 2.2 自己位置推定

本節では、主にコネクテッドカーでの使用を想定した自己位置推定技術について紹介する。自己位置推定とは、環境地図上における自身の位置と姿勢を推定する技術である。車両の自動走行や運転支援を実現するためには周囲の環境を認識することが必須であるが、実時間で全ての対象を認識することは難しい。そのため、予め構築した環境地図と自己位置推定によって環境認識の補助をすることが重要となる。例えば、自己と信号機の相対的な位置や姿勢を予め知ることができれば、カメラで撮影した画像から信号機を認識することが容易になる。

代表的な自己位置推定技術は GPS (Global Positioning System) のような GNSS (Global Navigation Satellite System) である。しかし、これらは建築物による電波の反射や遮断などに起因して車線単位での高精度な自己位置推定は困難である [93]。そのため、車両に搭載されたセンサと事前に作成された環境地図を用いた高精度な自己位置推定手法が検討されている。ここで利用される代表的なセンサとして、カメラと LiDAR が挙げられる。カメラは色や高解像度なテクスチャを計測することができる一方で、LiDAR は広範囲にわたって深度を計測できるなど、それぞれ異なる利点を持っている。そのため、カメラと LiDAR のそれぞれに対して自己位置推定手法が研究されている。

また、センサと環境地図に基づく自己位置推定は大域的自己位置推定と局所的自己位置推定に大別される。前者は探索範囲の制限なしに環境地図上での自己位置を推定する。後者は初期姿勢と呼ばれる位置と姿勢の初期値を基準として、その周辺の局所的な範囲のみを探索することにより、初期姿勢に対する相対的な自己位置を推定する。カメラと LiDAR を用いた代表的な大域的自己位置推定手法は、それぞれ画像検索と 3 次元点群検

Table 2.2: 2.2 節で紹介する関連研究のまとめ. 表の 1 列目は本節における項番号を, 2 列目は紹介する関連研究の技術分類を, 3 列目は技術の概要を表す.

項	分類	概要
2.2.1	大域的自己位置推定	探索範囲の制限なしに環境地図上での自己位置を推定する. 代表的な手法では, 画像検索や 3 次元点群検索技術を用いて DB に格納された位置情報を検索する.
2.2.2	カメラを用いた局所的自己位置推定	カメラを用いて, 環境地図上の局所的な範囲内で自己位置を推定する. 代表的な手法では, 道路上の白線や画像中の小領域のようなランドマークの対応付けを行なう.
2.2.3	LiDAR を用いた局所的自己位置推定	LiDAR を用いて, 環境地図上の局所的な範囲内で自己位置を推定する. 代表的な手法にスキャン照合があり, 3 次元点群やそれを変換した特徴の位置合わせを行なう.
2.2.4	データセット	自己位置推定の性能を評価するためのデータセット. 実環境を車両で走行しながら収集したデータで構成される.

索である. また, 代表的な局所的自己位置推定手法として, カメラに対しては白線や標識のようなランドマークの対応付け, LiDAR に対しては 3 次元点群のスキャン照合が挙げられる.

以下本節では, 2.2.1 項で大域的自己位置推定について述べた後に, 2.2.2 項と 2.2.3 項のそれぞれでカメラを用いた局所的自己位置推定と LiDAR を用いた局所的自己位置推定について概説する. そして, 2.2.4 項で自己位置推定のためのデータセットについて関連研究を紹介する. これらの各項で紹介する技術の概要は, Table 2.2 にまとめられる. 最後に, 2.2.5 項において移動端末上で動作する自己位置推定アプリケーションの課題について述べる.

### 2.2.1 大域的自己位置推定

大域的な自己位置推定手法とは, 探索範囲の制限なしに環境地図上での自己位置を推定する手法である. カメラを用いた場合の代表的な手法は画像検索であり, 2.1 節で紹介した内容とほぼ同様の技術が利用される. ただし, 自己位置推定の文脈における画像検索は場所認識 (Place recognition) もしくは位置認識 (Location recognition) [94–97] と呼ば

れ、一般的な画像検索とは区別されることが多い。場所認識では、DB に実世界の建物を撮影した参照画像と、その撮影場所の情報を紐付けて登録する。そして、現在位置で撮影した画像をクエリとして、参照画像に紐付いている位置情報を検索することにより、自己の位置や姿勢を得る。通常、参照画像とクエリ画像はほぼ同じ視点から撮影されるが、近年では衛星画像を参照画像とし、地上で撮影したカメラ画像をクエリ画像とする手法も提案されている [98–100]。これにより、地上で撮影された参照画像が存在しない場合にも自己位置推定が可能になる。

また、より高精度に位置や姿勢を推定するために、RANSAC を用いた姿勢推定を行なう場合もある。これは 2.1.2 項で述べた画像対応付けとほぼ同様の技術であるが、クエリ画像から得た 2 次元の特徴点と参照画像から得た 3 次元の特徴点の間で構築される 2D-3D 点対応を利用する点が異なる。ここでは、予め SfM (Structure-from-Motion) や SLAM (Simultaneous Localization and Mapping) を用いて参照画像の特徴点を 3 次元復元する。そして、クエリ画像が持つ 2 次元の特徴点と、参照画像が持つ 3 次元の特徴点を、それらに関する特徴ベクトルを用いて対応付けることにより、2D-3D 点の対応関係を構築する。幾何変換モデルとしては、次式に示す透視投影変換行列  $\mathbf{P}$  を用いる。

$$\mathbf{P} = \mathbf{AT} = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{pmatrix}, \quad (2.1)$$

ここで行列  $\mathbf{A}$  はカメラの内部パラメータ行列であり、 $f_x$  および  $f_y$  はそれぞれ  $x$  軸方向と  $y$  軸方向の焦点距離、 $c_x$  および  $c_y$  はそれぞれ  $x$  軸方向と  $y$  軸方向の画像主点を表す。また、行列  $\mathbf{T}$  はカメラの外部パラメータを表す行列であり、 $r_{11}$  から  $r_{33}$  は回転行列を、 $t_1$  から  $t_3$  は並進ベクトルを表す。そして、この行列  $\mathbf{P}$  を用いて 3 次元の特徴点をクエリ画像上に再投影し、対応する 2 次元の特徴点との誤差をしきい値判定することによって Inlier と Outlier へ分類する。これにより、地図に対する自己の相対的な位置と姿勢を高精度に推定することができる。また、画像検索を介さずに、直接的に局所特徴を対応付けることによって 2D-3D 点の対応関係を構築する手法もある [101, 102]。これらの手法においても、同様にして式 (2.1) から位置と姿勢を推定することができる。

LiDAR を用いた場合においても、画像検索のアナロジーとしての 3 次元点群検索が検討されている [103–106]。代表的な手法として、PointNetVLAD [103] が挙げられる。この手法は LiDAR で計測した 3 次元点群から多数の局所特徴を抽出し、それらを 1 本の大域特徴ベクトルへ集約する。そして、現在位置のスキャンデータから得られた特徴ベクトル

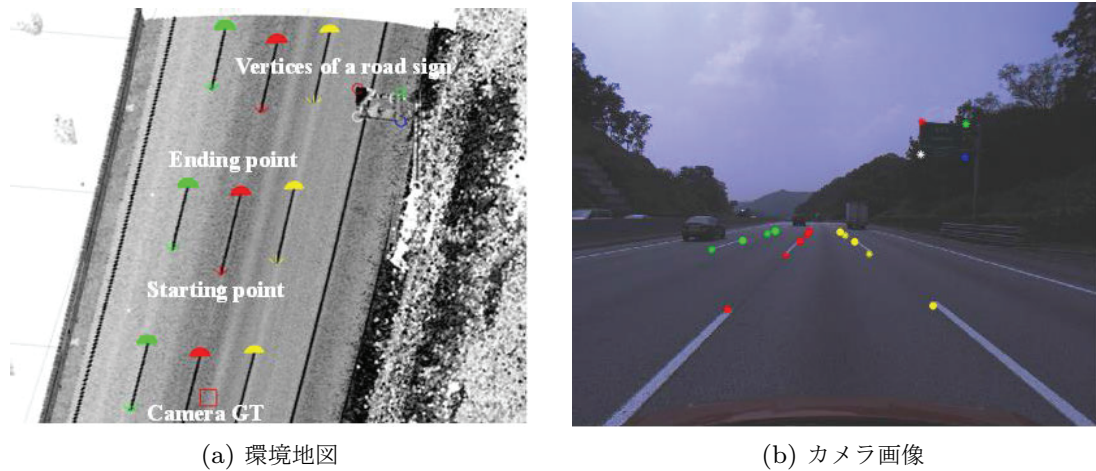


Fig. 2.6: 物体抽出に基づくランドマークの例. (a) は環境地図の模式図であり，道路上の白線の始点と終点，および道路標識の頂点を可視化している. (b) はカメラ画像であり，(a) 中の赤い矩形の位置から撮影されたものである. (出典：文献 [111] より作成)

ルを用いて地図側の特徴ベクトルを検索し，それに紐付く位置と姿勢を得る．また，3次元点群から局所特徴を抽出する手法も多数ある [107–110]．これらを用いて 3D-3D 点の対応関係を構築し，RANSAC を用いてより高精度な位置と姿勢を推定することもできる．

上記のような大域的自己位置推定は，例えば移動を開始する前に環境地図上での現在位置を推定するために利用される．そのため，必ずしも実時間処理が要求されるわけではない．一方，局所的自己位置推定は移動中に逐次的に実行する必要があるため，実時間処理が必須である．これを実現するために，一般に局所的自己位置推定手法は位置と姿勢の初期値を持ち，その初期値を基準として局所的な範囲のみを探索する．

## 2.2.2 カメラを用いた局所的自己位置推定

カメラを用いた場合の代表的な局所的自己位置推定手法は，ランドマーク対応付け [111–114] である．この手法は，道路上の白線，交通標識，路面標示，ポールのような特定クラスの物体をランドマークとみなし，カメラ画像からそれらの特徴を抽出する．Fig. 2.6 に，物体抽出に基づくランドマークの例を示す．環境地図は，実世界を撮影した画像または 3 次元点群から特徴を抽出し，その特徴と位置情報を紐付けた DB として構築される．自己位置推定の実行時には，現在位置で撮影したカメラ画像から抽出した特徴と，環境地図が含む特徴の位置合わせを行なうことによって自己位置を推定する．この

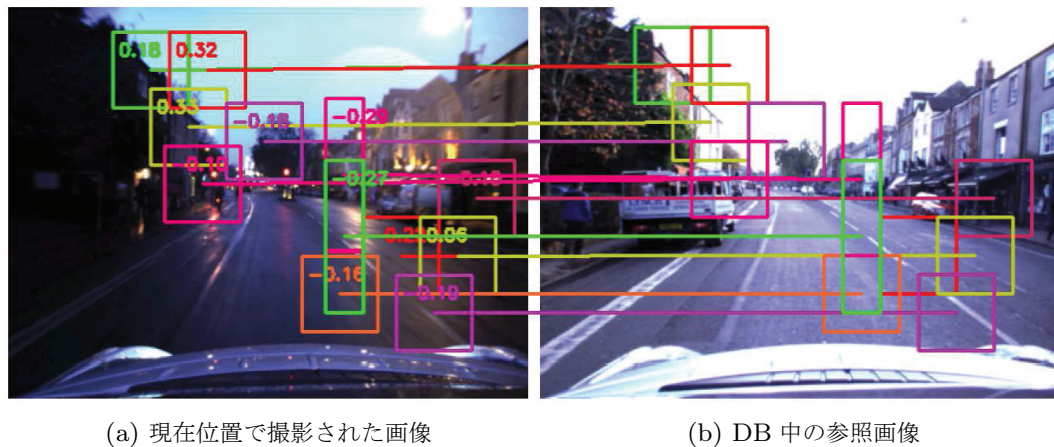


Fig. 2.7: 領域検出技術に基づくランドマークの例. 色が付いた矩形領域はランドマークを, 線はそれらが対応付けられたことを意味する. (出典: 文献 [115] より作成)

際, 前フレームの結果から予測される初期値を用いて, パーティクルフィルタや Kalman フィルタのような局所的な領域を高速に探索可能な手法により逐次的に自己位置を推定することができる.

また, 物体の種類を制限せずに, 領域検出技術によって画像から得られる局所領域をランドマークとする手法もある [115–117]. これらの手法では, Fig. 2.7 に示すような矩形から抽出された特徴と, その位置情報を紐付けた DB を環境地図とする. 自己位置推定の実行時には, 現在位置で撮影したカメラ画像から抽出した特徴と, 現在位置の近傍にある DB 中の特徴を対応付けることによって自己位置を推定する. 例えば Linegar ら [115] は, センサ側と地図側で 2D-3D 点の対応関係を構築し, RANSAC を用いて高精度に位置と姿勢を推定している.

また, Visual SLAM [118, 119] や Visual Odometry [120, 121] もカメラを用いた代表的な自己位置推定手法である. これらの手法は, 時系列的に連続する 2 フレームの画像間の相対的な位置と姿勢を推定する. これを反復することにより, ある時点のカメラ姿勢を基準として, その後のカメラの相対的な位置姿勢の軌跡を得ることができる. ただし, これらは事前に構築した環境地図を利用しないため, 各フレームにおいて生じる誤差が時間とともに累積する. そのため, 長時間にわたる推定を行なった場合には大きな誤差が発生するという問題がある. 更に, 環境地図を利用しないことにより, 上述した環境認識の補助が困難になる. 従って, これらの手法は車両の自動走行や運転支援を目的とした自己位置推定には適さない.



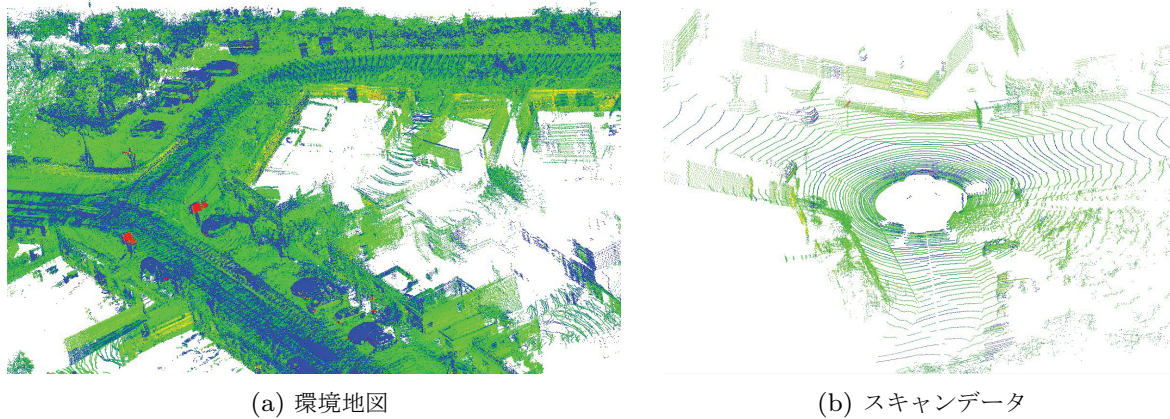


Fig. 2.8: 3次元点群の可視化. 点の色はLiDARによって計測された物体の反射強度を表す.

### 2.2.3 LiDAR を用いた局所的自己位置推定

LiDAR を用いた代表的な局所的自己位置推定手法は、スキャン照合 [6–8, 54, 122–128] である. この手法は、LiDAR で計測した 3 次元点群（スキャンデータ）の位置合わせを行なうことにより、自己位置を推定する. Fig. 2.8 に、3 次元点群を可視化した例を示す. 環境地図は、実世界の広範囲の環境から計測された 3 次元点群を表す. これは、LiDAR に基づく SLAM [129] や位置姿勢計測センサの利用によって複数のスキャンデータを統合することによって構築される. スキャンデータは、LiDAR による一度のスキャンによって得られた 3 次元点群を表す. 自己位置推定の実行時には、環境地図とスキャンデータの位置合わせを行なうことでそれらの間の回転と並進の変換を推定する. ここでは、現在の位置と姿勢を表す初期値を利用し、その初期値に対して推定された回転と並進の変換を加えることによって、自己の位置と姿勢を推定する.

3 次元点群の位置合わせを行なう最も代表的な手法は、ICP (Iterative Closest Point) [54] とその変形 [125, 126] である. ICP のアルゴリズムを簡潔に紹介する. 2 つの点群  $A$ ,  $B$  が与えられたとき、 $A$  の各点から最近傍となる  $B$  の点を探索し、対応付ける. そして、対応付けられた全ての点の間の Euclidean 距離の総和  $e$  を計算する. 次に、 $e$  を最小化するような回転と並進の変換パラメータを推定する. このとき、回転と並進の変換を加えた  $A$  の点と、それに対応する  $B$  の点の間の Euclidean 距離を表す目的関数の微分を用いて、解析的に変換パラメータが求まる. そして、推定された変換パラメータを用いて

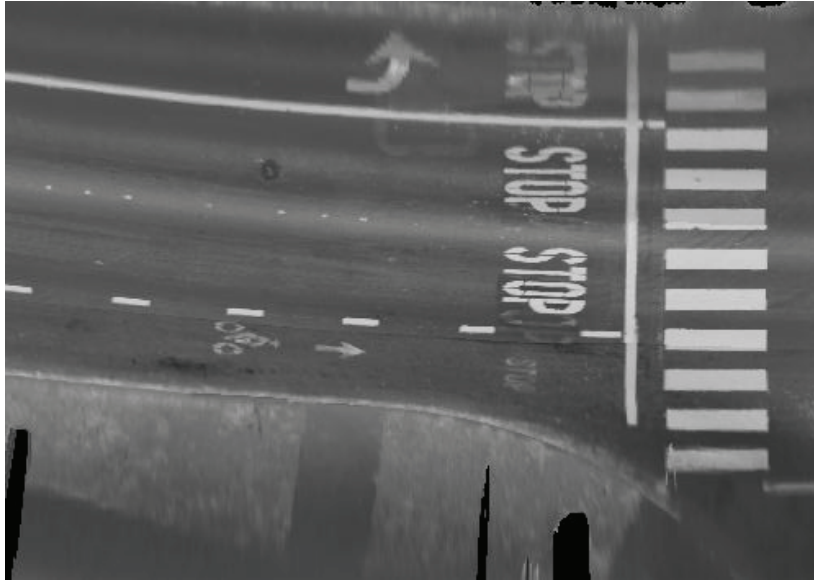


Fig. 2.9: 正射投影画像の例. 画素値は反射強度を表し, 黒い領域は破棄された物体の領域に対応する. (出典: 文献 [5] より作成)

$A$  の全ての点を変換する. その後, 変換された  $A$  を用いて上記の処理を反復する. この反復は, 反復の前後で得られる  $e$  の差がしきい値未満となるまで継続する. これにより, 点群  $A$  と  $B$  が形状的に合致するための回転と並進の変換を求めることができる.

ICP はアルゴリズムが単純で汎用性が高いため, 3次元モデルの位置合わせのために幅広く利用されている [130–133]. しかし, 車両の自己位置推定に利用する際には, 大きく2つの問題がある. 1つ目は, 環境地図の容量が大きいことである. 3次元点群は多数の座標値の集合として表現されるため, 例えば国規模のような大規模な環境地図を想定する場合にはその容量は膨大となる. 2つ目は, 計算量が大きいことである. ICP には片方の点群の点からもう片方の点群の点への最近傍探索を行なう処理が含まれるが, この処理は一般に点群の点数が多いほど計算量が大きくなる. 車両の自己位置推定においては地図側とセンサ側の両方とも点群の点数が多いため, 実時間処理が困難となる恐れがある.

従って, 車両の自己位置推定に対しては, ICP と比べて小容量な地図を用いて高速に計算可能なスキャン照合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 [5] と 3D NDT (Normal Distributions Transform) [6] である. 前者は, 3次元点群を正射投影された路面の俯瞰画像に変換する. Fig. 2.9 に, この画像の例を示す. この画像では, 画素値は LiDAR によって計測された物体の反射強度を表し, 路面以外を

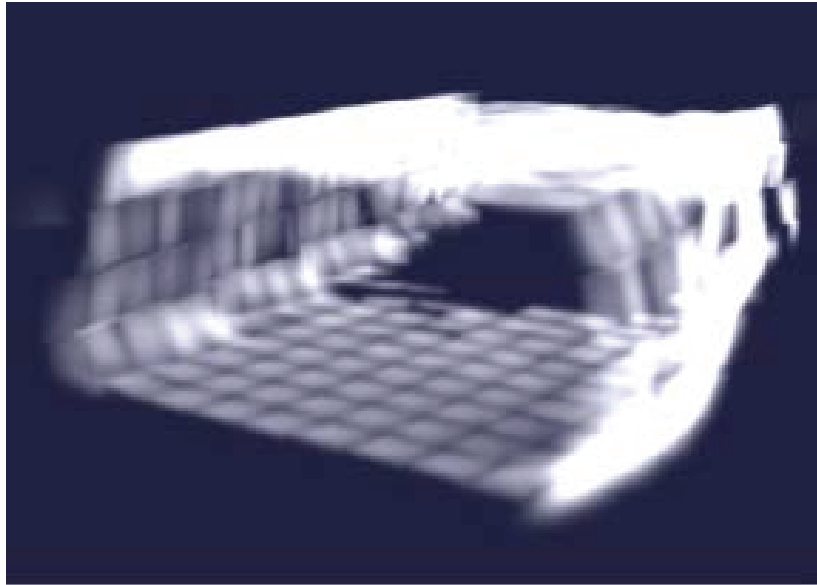


Fig. 2.10: 3D NDT によって構築された環境地図の例．白色の楕円体はボクセルごとの正規分布を表す．（出典：文献 [6] より作成）

表す物体は破棄される．自己位置推定の実行時には，パーティクルフィルタを用いて実時間で位置と姿勢の変換パラメータを求める．これにより，この手法は ICP に比べて小容量な地図データを用いて高速に自己位置推定を実現することができる．しかし，この手法は路面の反射強度，すなわちテクスチャを手掛かりに位置合わせを行なうため，路面が遮蔽される場合や路面のテクスチャが乏しい場合には推定に失敗するという問題がある [8, 127]．一方，後者は 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．Fig. 2.10 に，この手法によって構築された環境地図の例を示す．この図では，白色の楕円体はボクセルごとの正規分布を表す．自己位置推定の実行時には，ICP と同様の方法で回転と並進の変換パラメータを推定する．ただし，ICP と比べて 2 つの点で異なる．1 つ目は，ICP が点間の対応関係を構築するのに対して，3D NDT は点と正規分布の対応関係を構築することである．この処理はボクセルを用いてスキャンデータを分割し，各点をボクセルに割り当てることで実現できる．従って，点の最近傍探索を行なう必要がないため，ICP に比べて計算量が削減される．2 つ目は，ICP が類似尺度に点間の Euclidean 距離を利用するのに対して，3D NDT は点と正規分布から計算できる尤度を利用することである．すなわち，正規分布に従って点が観測されるという仮説の尤度を，点と正規分布の対応間の類似尺度とみなす．このような類似尺度の変更



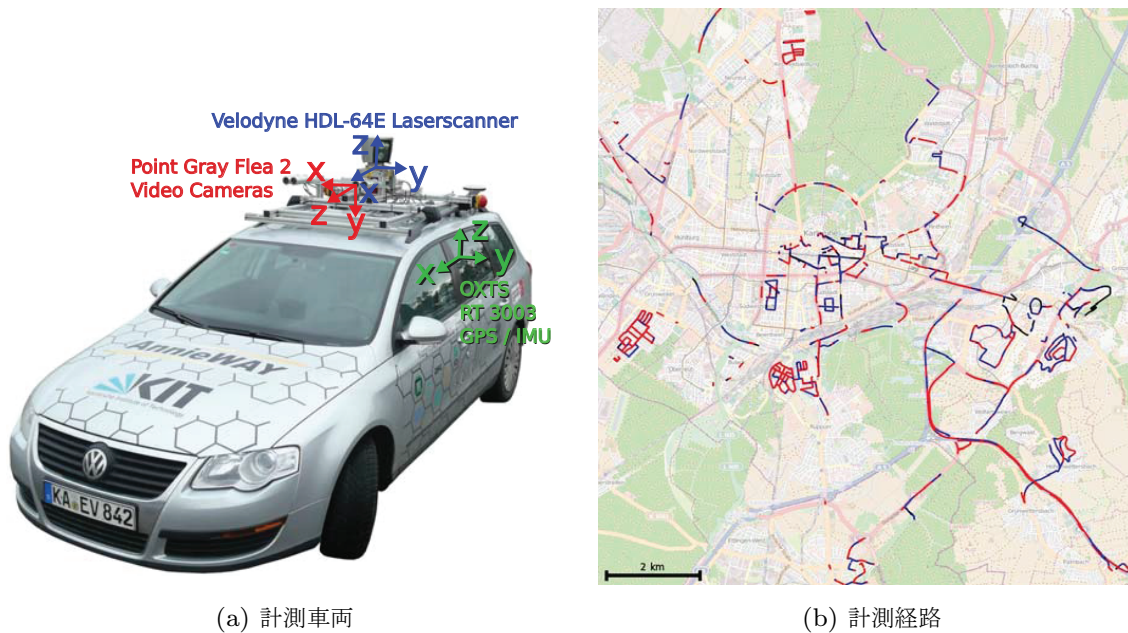


Fig. 2.11: データの収集方法. (b) において、赤い経路は RTK による補正が得られたことを、青い経路は補正が得られなかったことを、黒い経路は GPS が利用できなかったことを表す. (出典：文献 [136] より作成)

により、点群と正規分布の集合の間の類似度を高速に計量することができる. この手法は路面だけでなくその周囲の立体物も利用して位置合わせを行なうため、正射投影画像に基づく手法と比べて頑健に自己位置推定を実現することができる.

#### 2.2.4 自己位置推定のためのデータセット

自己位置推定を含む、車両の自動走行に関わるアプリケーションの性能を評価するためのデータセットが多数公開されている [134–139]. 最も代表的なデータセットとして、KITTI データセット<sup>†</sup> [135, 136] が挙げられる. このデータセットは、Visual Odometry や Visual SLAM, 物体検出, 深度推定, オプティカルフロー推定などの様々なアプリケーションの性能を評価することができるため、幅広く利用されている. データは、Fig. 2.11(a) に示す車両に搭載されたセンサを用いて、市街地を走行しながら収集された. この車両には、ステレオカメラ, LiDAR センサ, GPS (Global Positioning System),

<sup>†</sup> <http://www.cvlibs.net/datasets/kitti/>



Fig. 2.12: KITTIデータセット<sup>†</sup> [136] に含まれる画像の例.

IMU (Inertial Measurement Unit) が搭載されている. ステレオカメラはステレオ画像を, LiDAR センサは 3 次元点群 (スキャンデータ) を計測する. GPS と IMU は統合されたシステムとして, 3 次元の位置や姿勢を計測する. なお, このシステムでは RTK (Real Time Kinematic) を用いて位置情報の補正を行なう. データの収集はドイツの Karlsruhe 市内において, Fig. 2.11(b) に示す経路に沿って行なわれた. この図においては, 赤い経路は RTK による補正が得られたことを, 青い経路は補正が得られなかったことを, 黒い経路は GPS が利用できなかったことを表す. 収集されたデータは上記のアプリケーションごとに分類され, 性能を評価するための真値と合わせて提供されている. この中でも, Visual Odometry 向けのデータセットは数 km の経路にわたって位置と姿勢の真値が提供されているため, 自己位置推定の性能評価に適している. Fig. 2.12 に, このデータセットに含まれる画像の例を示す. この図より, 住宅や街路樹のような様々な物体が存在する環境においてデータが計測されていることがわかる. 以上のように, 車両の自動走行に関わる様々なアプリケーションの性能評価に適したデータが提供されていることから, KITTI データセットは幅広く利用されている.

### 2.2.5 移動端末における自己位置推定

本項では、移動端末上で動作する自己位置推定アプリケーションとその課題について整理する。自己位置推定は、車両の自動走行や運転支援を実現するための基礎的要件の 1 つである。GNSS は車線単位の自己位置推定の実現が困難である [93] ため、車載のセンサと事前に作成した環境地図を用いた高精度な自己位置推定手法が検討されている。

自己位置推定は大域的自己位置推定と局所的自己位置推定に大別できる。前者は移動を開始する前などに利用されるため、必ずしも実時間で処理できなくても良い。一方、後者はセンサからの入力を実時間で処理する必要があるため、通信による遅延を回避するためにはアプリケーションを車両上で動作させることが望ましい。従って、車両上で動作する局所的自己位置推定アプリケーションの検討は、重要な課題である。

環境地図を用いた自己位置推定手法は地図データの表現形式に適するように設計されるため、自己位置推定手法を検討する上では地図の表現方法も重要である。2.2.2 項で紹介したランドマークで構成される環境地図は、代表的な表現形式の 1 つである。しかし、このような表現形式による自己位置推定は地図側とセンサ側の両方でランドマークが抽出される場合にしか実現できない。例えば、地図側に対象となるランドマークが存在しない場合や、センサ側で遮蔽などの外乱に起因してランドマークを検出できない場合には推定が困難である。また、物体検出手法を用いて地図を生成する場合には未検出や誤検出の問題が不可避免的に発生する [140–145]。従って、誤りがない地図を作るためには少なからず人手による作業を要するため、地図生成を完全に自動化することは難しい。

一方、ランドマークの抽出を伴わずに環境地図を生成する手法もある。代表的なものとして、LiDAR で計測した 3 次元点群を環境地図とする方法がある。ランドマークの抽出に基づく手法とは異なり、この手法は特定のランドマークに依存せずに頑健に自己位置推定を実現することができる。また、LiDAR はレーザ照射を利用するため、この手法は照明の影響を受けにくいという利点を持つ。しかし、3 次元点群で表現される環境地図は容量が大きいという問題がある。特に、車両の自動走行のように大規模な環境地図を要する場合においては、その容量は膨大である。車両で地図データを利用するためには、2 つの方法が考えられる。1 つ目は、例えば国規模のような大規模な地図データを予め記憶装置に格納する方法である。この方法では単位当たりの容量が大きい場合、記憶装置の空き容量（例えば 100 GB）に収まらない恐れがある。2 つ目は、通信を介して地図データを車

両ヘストリーミング配信する方法である．この方法では電波の乱れなどにより通信帯域が狭くなった場合でも遅延なく配信する必要があるため，容量が小さな地図データが求められる．従って，いずれの方法に対しても可能な限り容量は小さいことが望ましい．

この問題に対処するために，より小さな容量の地図に変換してスキャン照合を行な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 [5] と 3D NDT [6] 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，保持すべき地図の容量を削減することができる．しかし，これらの地図の容量は 3 次元点群地図の容量を基準として 100 分の 1 から 1,000 分の 1 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 の規模になる可能性がある．

そのため，容量が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，更に容量を削減する試みも検討されている．Wei ら [50] は，自己位置推定のために最適化された CNN を用いて，正射投影画像を圧縮する手法を提案した．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，正射投影画像地図の容量を元の大きさの 100 分の 1 以下にした．しかし，これまでのところ，3D NDT によって構築された地図データを圧縮する試みはない．また，3D NDT は最も広く利用される自己位置推定手法の 1 つであるが，推定が反復的な方法に基づくために処理速度が不安定になりやすいという問題もある．

## 第 3 章

# 特徴の選択に基づく画像の索引付け

本論文ではスマートフォンによる画像検索とコネクテッドカーによる自己位置推定というアプリケーションにおいて、高速かつ高精度な照合と特徴 DB の小容量化を両立することを目指す。この目的を達成するために、本論文では (a) 特徴の選択, (b) 特徴の変換, (c) 類似尺度の変更, の観点に基づいて, (i) DB 内のデータ数の削減, (ii) DB 内の各データの小容量化, (iii) 効率的なデータの照合, の課題に取り組む。この内、本章ではスマートフォンによる画像検索に着目し, (a) 特徴の選択に基づいて (i) DB 内のデータ数の削減を実現する。具体的には, 2 値特徴と BoVW に基づくモバイル画像検索 (Mobile Visual Search; MVS) における画像の索引付け処理を改善するための, 特徴設計手法を提案する。本手法では, 画像検索に特に有益な特徴を選択することにより, DB 容量や処理時間の増加を抑制しつつ検索精度を改善する。

MVS ではカメラの視点変化に伴って局所特徴の検出誤りと量子化誤りが不可避免的に発生し, 検索精度が低下する。典型的な画像検索手法では単一視点の参照画像から局所特徴を抽出するが, そのような局所特徴では検出誤りおよび量子化誤りに対処するには不十分である。この問題は同一の物体を複数の視点から撮影した画像集合 (多視点画像) から抽出した局所特徴を DB に登録することによって対処できるが, その場合には DB の容量が大きくなる。そこで本章では, 多視点画像から抽出した局所特徴を選択的に採用することにより, 特徴 DB を小容量化する手法を提案する。具体的には, 多視点画像の各画像にわたる同一位置における特徴点の検出および量子化した特徴ベクトルの再現性に基づいて特徴の信頼度を定義し, その信頼度に従って特徴を選択する。本研究では, この特徴選択を最大被覆問題とみなし, 特定の制約関数の下で最も高い信頼度を得る特徴の集合を求め

る。この問題は NP 困難であり、多項式時間で実行できないことから、貪欲法による近似解法を提案する。また、この目的のために、画像検索手法における一致条件と整合するように設計された新たな制約関数を提案する。この一致条件は、参照画像とクエリ画像の間で対応する局所特徴が一致するか否かを識別するものである。評価実験として、提案手法が DB 容量や検索時間を増加させずとも、検索精度を改善することを示す。

本研究の貢献は、以下のように要約される。

1. 多様な視点変化に対する特徴点の検出と特徴ベクトルの量子化の再現性に基づく特徴の信頼度を提案する。
2. 画像検索の文脈における特徴選択を最大被覆問題として定式化し、最も高い信頼度を得る特徴の集合を求める手法を提案する。
3. スマートフォンに搭載されたカメラで撮影したクエリ画像を用いた評価実験を行ない、提案手法が多様な視点変化に対して頑健な局所特徴を選択することを示す。

以下、3.1 節で提案手法の基本的な考え方について述べる。その後、3.2 節で画像検索および画像対応付けの文脈における特徴設計について概説し、3.3 節で提案手法を適用する画像検索手法を紹介する。そして、3.4 節で画像索引付けのための特徴設計手法を提案する。その後、3.5 節で提案手法の有効性を評価し、3.6 節で本章をまとめる。

### 3.1 はじめに

スマートフォン上で動作する画像検索アプリケーションを実現するために、本研究では 2 値特徴と BoVW に基づく MVS に焦点を当てる。BoVW は効率的な検索を実現するために、局所特徴の特徴点を利用せず、特徴ベクトルを量子化した VW の一致のみに基づいて画像間類似度を計量する。しかし、その背景にある技術は局所特徴に基づく画像対応付け、すなわち特徴点と特徴ベクトルの両方の一致に基づく類似度の計算である。従って、BoVW においては参照画像とクエリ画像の間で次の一致条件を満たす局所特徴のみが、検索に貢献すると考えられる。

1. 特徴点の一致：対応する特徴点が物体の同一位置から検出される
2. 特徴ベクトルの一致：対応する特徴ベクトルが同一の VW に量子化される

本研究においては、これらの条件が満たされなくなることをそれぞれ特徴点の検出誤り、特徴ベクトルの量子化誤りと呼ぶ。これらの誤りは、参照画像とクエリ画像間の視点変化に伴って発生する恐れがある。そして、これらの誤りのいずれかが発生した場合には、正しい参照画像とクエリ画像の対であっても画像間類似度が低くなり、検索精度に悪影響を及ぼす。

BoVW のもう 1 つの特性は、DB 容量が画像ごとの特徴数に依存することである。従って、信頼度に基づく特徴選択は、DB 容量を削減するための効果的な戦略である。この戦略においては、検索に貢献する可能性が高い特徴を選択する必要がある。言い換えれば、特徴点の検出の再現性と特徴ベクトルの量子化の再現性のいずれかが低い場合、その特徴を DB に登録するべきではない。

本章では、以上の一致条件を満たす確率を向上させるような、特徴設計手法を提案する。提案手法では、実際の画像集合の代わりに ASIFT (Affine-SIFT) [146] のように視点変化を模擬する人工画像を使用する。これにより、実世界で物体を様々な視点から撮影せずとも、多視点画像を構築することができる。提案手法は単一の参照画像を変形させることで人工画像を生成するため、平面物体のみを認識対象とする。すなわち、提案手法には平面物体の画像に対してのみ適用可能であるという制限がある。ただし、対象物体の 3D モデルが存在する場合には、3D 画像技術を介して適用可能である。

Fig. 3.1 に提案手法における特徴の信頼度の概要を示す。提案手法は初めに、単一の参照画像から様々な人工画像を生成し、それらから個別に局所特徴を抽出する。次に、各人工画像から検出された特徴点の座標を参照画像の座標系に再投影し、物体の同一位置にある特徴点を関連付ける。そして、特徴点によって関連付けられた特徴ベクトルが同一の VW に量子化される頻度を計量し、それを特徴の信頼度とする。例えば、Fig. 3.1 における緑色の特徴に対する信頼度は、特徴点の検出と特徴ベクトルの量子化の両方の再現性が低いために、1 となる。ここでは、VW-ID が “+” の特徴と “☆” の特徴に対して個別に信頼度が計量される。青色の特徴に対する信頼度は、特徴点の検出の再現性が高いが、特徴ベクトルの量子化の再現性が低いために、2 となる。同様に、ここでは VW-ID が “+” の特徴と “△” の特徴に対して個別に信頼度が計量される。橙色の特徴に対する信頼度は、特徴点の検出と特徴ベクトルの量子化の両方の再現性が高いために、最も高い 3 となる。このように信頼度に基づいて特徴を選択することによって、DB 容量の増加を防ぎつつ、検索に貢献する可能性の高い特徴を用いて参照画像を索引付ける。具体的には、特徴が一



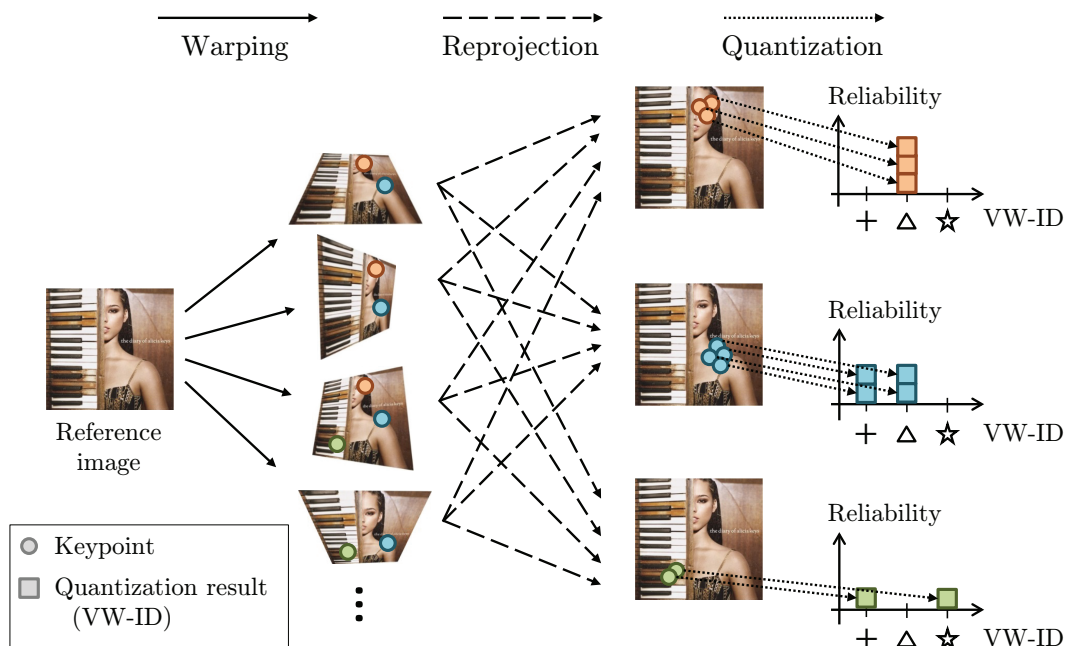


Fig. 3.1: 提案手法における特徴の信頼度の概要. ここでは、円は特徴点 (Keypoint) を表し、四角は特徴ベクトルの量子化結果 (Quantization result) を表す. 同一の色は物体の同一位置から検出された特徴であることを意味する.

致条件を満たすことを特徴空間上での被覆とみなし、特徴選択を最大被覆問題として定式化する. これにより、最も高い信頼度を得る特徴の集合を求めることができる.

## 3.2 特徴設計

画像検索または画像対応付けの文脈において、特徴設計に関する多くの研究が存在する. 本節では、特徴設計の一般的な方法と MVS における実用上の問題について概説する.

Wang ら [147] は、Informative feature と呼ばれる、視点変化に対して頑健かつ検索への有効性が高い特徴を選択する手法を提案した. この手法は、検出誤りと量子化誤りの両方を考慮するという点で、本研究の提案手法に最も近いものである. この手法では、初めに RANSAC [69] により幾何検証を行なうことによって、多視点画像の各画像にわたって特徴点を関連付ける. ここで、他の特徴点と関連付けられていない孤立した特徴点は破棄される. そして、残った特徴点のみを用いて、関連する特徴ベクトルを VW へ量子化する. その後、各特徴は TF-IDF (Term Frequency-Inverse Document Frequency) に類似した方法によって順位付けされる. 最後に、参照画像の索引付けを行なうために、上



位順位の特徴が選択される．これにより，この手法は検出誤りと量子化誤りの両方を軽減することができる．しかし，異なる視点から同一の物体を撮影した多数の画像を必要とするため，実用的ではない．

複数割り当て (Multiple Assignment; MA) [71] は， $k$  近傍となる複数の VW に単一の特徴を割り当てる手法であり，量子化誤りを軽減するためにしばしば使用される．Mikulík ら [148] はこの手法よりも更に高精度な MA を実現した．彼らの手法は，異なる視点の画像から抽出した特徴を用いて，代替的な  $k$  個の VW を学習する．この代替的な  $k$  個の VW は，クエリ画像からある VW  $W_a$  が観察された場合に，参照画像の VW  $W_b$  と一致する確率  $P(W_b|W_a)$  に基づいている．しかし，これらの手法は DB 容量を  $k$  倍に増大させるという問題がある．

Chen ら [149] は，クエリ画像で予想される視点を模擬するために参照画像を変形し，その画像から局所特徴を抽出する手法を提案した．具体的には，参照画像に対する正面，上面，右面，左面，底面を模擬する人工画像を生成する．その後，これらの画像から得られる 5 通りの個別のデータセットを構築する．局所特徴がそれぞれの人工画像から抽出されるため，この手法は検出誤りと量子化誤りの両方を軽減することができる．しかし，この手法は DB 全体の容量を 5 倍に増大させるにも関わらず，その効果は上記の 5 通りの視点に限定されているため，効果が低い．

ASIFT [146] は，視点変化に頑健な画像対応付けを実現する．この手法は画像を変形させることによって様々な視点変化を模擬する．元々の ASIFT には量子化処理が含まれてはいないものの，Chen らの手法と同様の理由で，検出誤りと量子化誤りの両方を軽減できる可能性がある．しかし，ASIFT の戦略を画像検索に適用した場合，全ての人工画像を DB に登録する必要がある．その場合，模擬した視点数倍に DB 容量が増大するという問題がある．

本研究では，多視点画像から観察される VW に焦点を当て，画像の変形に対する局所特徴の信頼度を考慮した手法を提案する．提案手法は DB に登録される特徴が一致条件を満たす確率を改善するために，より効果的な MA を実現する．また，信頼度に従って特徴を選択することにより，DB 容量を効率的に調整することができる．更に，提案手法は，異なる視点から実物体を撮影した多数の画像を収集する必要がないという利点も持つため，実用性が高い．評価実験では，DB 容量が等しい場合，標準的な MA よりも提案手法が優れた検索精度を達成することを示す．

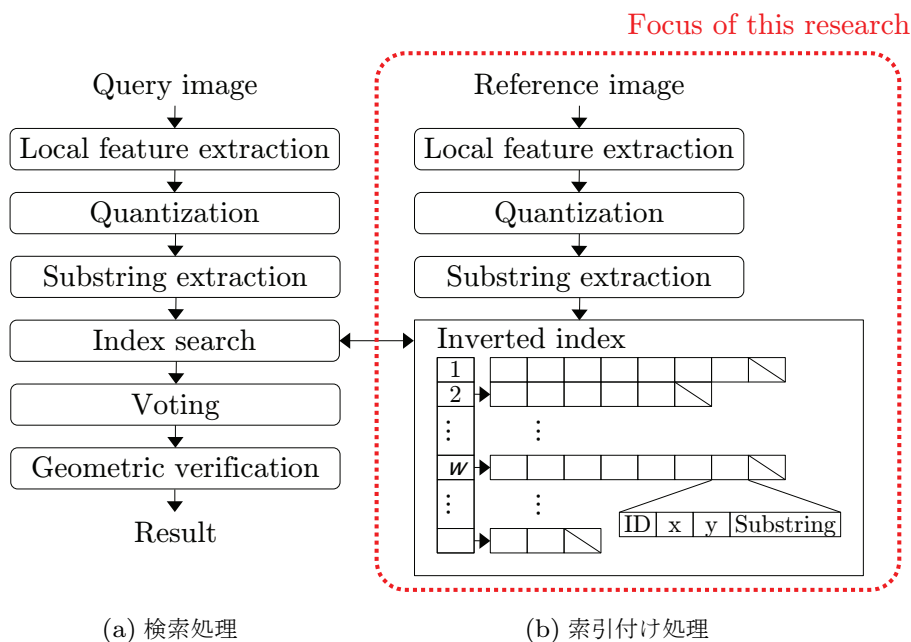


Fig. 3.2: 画像検索のベースライン手法の枠組み.

### 3.3 画像検索手法

BoVW は代表的な画像検索手法であるが、一般に実数値特徴が適用されており [70, 72, 150], 2 値特徴を適用した研究はわずかである. Gálvez-López ら [151] は, BoBW (Bag-of-Binary Words) と呼ばれる, BoVW に 2 値特徴を適用する手法を提案した. Zhou ら [152] は, メモリ使用量を抑制しつつ検索精度を改善するために, 2 値特徴に対する HE (Hamming Embedding) [72] の変形を提案した. この手法は, 特徴ベクトルの最初の  $a$  [bit] の 2 値文字列を VW とみなし, その次の  $b$  [bit] の 2 値文字列を転置インデックスに格納する. Uchida ら [153] は, スマートフォン上で実時間 MVS を実現するための, 2 値特徴に基づく画像検索手法を提案した. この手法は, DB 容量を増やすことなく, 従来手法と比べて検索精度を改善できることが示されている. 数十 MB の記憶容量を必要とする従来のスタンドアローン MVS 手法 [1, 2] とは対照的に, この手法は数百通りの物体を認識するために, 数 MB の記憶容量しか必要としないという利点を持つ. 従って, 本研究では Uchida ら [153] の手法を画像検索のベースライン手法とみなす. すなわち, 提案する画像索引付けのための特徴設計手法は, この手法を基に構築する. Fig. 3.2 にベースライン手法の枠組みを示す. これについては, 以下で詳細に説明する.

標準的な BoVW に従い、画像から局所特徴を抽出 (Local feature extraction) し、VW と呼ばれる代表ベクトルに量子化 (Quantization) する。クエリ画像の各 VW が参照画像にスコアを投票 (Voting) することによって画像間類似度を計算し、それによって初期順位が得られる。投票処理の後に幾何検証 (Geometric verification) によって再ランキングが行なわれ、初期順位を改善する [70]。更に、この枠組みは以下の拡張手法を含む。

#### 適応的サブストリング抽出

サブストリング (Substring; SS) [153] とは、HE [72] と同様に、特徴ベクトルをコンパクトな符号に変換する手法である。この手法は Fig. 3.2 のサブストリング抽出 (Substring extraction) で実行し、索引付け処理の場合は転置インデックス (Inverted index) に符号を格納する。SS は 2 値特徴ベクトルの特定位置のビットを抽出することによって、短い 2 値文字列を生成する。これらの位置は、識別的なビットを抽出するために VW ごとに適応的に変化する。Fig. 3.2 の索引検索 (Index search) においては、クエリ画像 (Query image) の特徴と同一 VW に割り当てられた参照画像 (Reference image) の特徴を転置インデックスから検索し、それらの特徴から抽出された 2 値文字列間の Hamming 距離を計量する。

#### 修正 Local NBNN (Naive-Bayes Nearest-Neighbor) スコアリング

同一 VW に割り当てられたクエリ画像の特徴  $\mathbf{q}$  および参照画像の特徴  $\mathbf{r}$  から抽出された 2 値文字列が与えられた場合、スコア関数は  $w(d_l) = (d_L/d_l)^2 - 1$  と表現される。ここで  $d_l$  は、 $\mathbf{q}$  の 2 値文字列と、その  $l$  番目の近傍となる  $\mathbf{r}$  の 2 値文字列の間の Hamming 距離である。 $L$  はスコアリングに使用する近傍数を指定するパラメータである。Fig. 3.2 の投票 (Voting) においては、このスコア関数を用いて参照画像にスコアを投票し、画像間類似度を計算する。本研究では、原論文 [153] と同じく  $L = 2$  とする。この手法は、Local NBNN [154] の修正版である。

#### WGC (Weak Geometric Consistency)

WGC [72] は、方向とスケールの情報に基づく幾何的制約を与える。この手法は、Fig. 3.2 の投票 (Voting) において、方向の差とスケール比に関して一貫性がない特徴の対応を投票から除外する。本研究では、Tsai らの研究 [155] においてスケール情報が効果的でないことが示されているため、方向情報のみを使用する。

### 3.4 特徴の選択に基づく画像の索引付け

本節では、信頼度に従って選択した特徴を用いた、画像索引付けのための特徴設計手法を提案する。この信頼度とは、クエリ画像が実際に与えられたときに一致条件を満たす頻度  $[0, \infty]$  を表す。しかし、参照画像を索引付けする際に、実際のクエリ画像は不明なため、代わりに実際のクエリ画像を模擬した多視点画像を使用する。実世界で撮影することなく多視点画像を構築するために、参照画像を変形させることによって多視点の人工画像群を生成する。これにより、実際のクエリ画像が人工画像の1つに類似している場合には、視点変化に対して頑健な認識を実現する参照画像の索引付けが可能となる。本研究の目的は、より少数の参照特徴によって、より多くのクエリ特徴に対する一致条件を満たすことである。そのために、本研究では特徴選択を特徴空間上での最大被覆問題として定式化し、解を求める。索引付け処理においては、SS および WGC の性能を改善するために最適化した特徴を DB に登録する。また、BoVW においては同一 ID の VW が大量に出現するバースト性が画像間類似度に悪影響を与える [156]。これを抑制しつつ DB 容量を削減するために、非バースト性選択を行なう。提案手法は、Fig. 3.2(b) の索引付け処理のみを改良し、Fig. 3.2(a) の検索処理は変更しない。

#### 3.4.1 定式化

求める  $N$  個の局所特徴の集合を  $X = \{\mathbf{x}_i | i = 1, \dots, N\}$  とする。また、単一の参照画像を変形させることによって生成した多視点の人工画像から抽出した  $M$  個の局所特徴の集合を  $Q = \{\mathbf{q}_j | j = 1, \dots, M\}$  とする。本研究では、特定の制約の下での目的関数の最大化問題として特徴選択を次式のように定式化する。

$$\max \sum_{\mathbf{x}_i \in X} \sum_{\mathbf{q}_j \in Q} z(X, \mathbf{q}_j) f(\mathbf{x}_i, \mathbf{q}_j), \quad (3.1)$$

$$\text{s.t.} \quad \sum_{\mathbf{x}_i \in X} f(\mathbf{x}_i, \mathbf{q}_j) \leq N, \quad \forall \mathbf{q}_j \in Q, \quad (3.2)$$

$$f(\mathbf{x}_i, \mathbf{q}_j) \in \{0, 1\}, \quad \forall \mathbf{x}_i \in X, \quad \forall \mathbf{q}_j \in Q, \quad (3.3)$$

$$z(X, \mathbf{q}_j) \in \{0, 1\}, \quad \forall \mathbf{q}_j \in Q, \quad (3.4)$$

ここで  $f$  は想定する画像検索手法における一致条件に対応する制約関数,  $z(X, \mathbf{q}_j)$  は  $\mathbf{q}_j$  に対して一致条件を満たす  $X$  内の一つの  $\mathbf{x}_i$  に対して 1 となり, それ以外の  $\mathbf{x}_i$  に対して 0 となる関数である. この関数  $z(X, \mathbf{q}_j)$  は,  $X$  の冗長性を排除するために使用される. すなわち, 複数の  $\mathbf{x}_i$  が同一の  $\mathbf{q}_j$  に対して一致条件を満たすことを防ぐことにより, 有限個の  $\mathbf{x}_i$  が可能な限り多くの異なる  $\mathbf{q}_j$  に対して一致条件を満たすようになる. 言い換えれば,  $X$  は検索に貢献する  $Q$  内の最大個数の  $\mathbf{q}_j$  を被覆する. これは NP 困難であることが知られている最大被覆問題であり, 多項式時間で実行不可能である [157]. そのため, 実現可能な時間で解を求めるために近似解法を提案する.

具体的には, 最大被覆問題に対する多項式時間近似アルゴリズムとして知られる貪欲法 [158, 159] に基づく近似解法を提案する. 最大被覆問題に対してこのような近似アルゴリズムを適用する考え自体は新しくはなく, 無線センサ網の設計 [160] やテキスト要約 [161] の文脈において広く適用されている. しかし, 画像検索の文脈における特徴選択を最大被覆問題として定式化するのは, 本研究が初の試みである. この目的のために, 3.4.2 項で新たな制約関数を導入し, 3.4.3 項で近似解法のアルゴリズムを提案する.

### 3.4.2 制約関数

制約関数は, 3.3 節で紹介されたベースライン手法における一致条件と整合性を持つように設計する. すなわち, 参照特徴  $\mathbf{r}$  とクエリ特徴  $\mathbf{q}$  における, 特徴点の検出および特徴ベクトルの量子化と整合性を持つ制約である. また, 本研究では WGC に基づく一致条件と整合性を持つ制約も含める.

個々の局所特徴  $\mathbf{l}$  は空間的座標  $c(\mathbf{l})$  および量子化された特徴ベクトル, すなわち VW  $v(\mathbf{l})$ , 方向  $\theta(\mathbf{l})$  を持つものとする. 特徴点の検出に基づく制約 (Keypoint-Based Constraint; KBC) 関数は, 次式で定義される.

$$f_c(\mathbf{r}, \mathbf{q}) = \begin{cases} 1 & \text{if } \|c(\mathbf{r}) - \mathbf{P}c(\mathbf{q})\| \leq \varepsilon_c, \\ 0 & \text{otherwise} \end{cases}, \quad (3.5)$$

ここで  $\varepsilon_c$  は再投影誤差のしきい値であり,  $\mathbf{P}$  はクエリ画像の座標系から参照画像の座標系への既知の投影行列 (例えば Homography 行列) である. また, 特徴ベクトルが同一の VW に割り当てられるか否かを表す, 特徴ベクトルの量子化に基づく制約 (Visual

word-Based Constraint; VBC) 関数は次式で定義される.

$$f_v(\mathbf{r}, \mathbf{q}) = \begin{cases} 1 & \text{if } v(\mathbf{r}) = v(\mathbf{q}) \\ 0 & \text{otherwise} \end{cases}. \quad (3.6)$$

WGC のための, 方向に基づく制約 (Orientation-Based Constraint; OBC) は次式で定義される.

$$f_\theta(\mathbf{r}, \mathbf{q}) = \begin{cases} 1 & \text{if } |\theta(\mathbf{r}) - \theta(\mathbf{q})| \leq \varepsilon_\theta \\ 0 & \text{otherwise} \end{cases}, \quad (3.7)$$

ここで  $\varepsilon_\theta$  は方向の差のしきい値であり, 量子化された角度の解像度 [72] と整合性を持つ. この OBC 関数は, WGC による特徴対応のフィルタリングを模擬している. すなわち, OBC 関数は WGC に基づく一致条件と整合性を持つように設計された制約関数である.

ベースライン手法では, 上記の制約の全てが満たされた場合にのみ, 投票が行なわれる. 従って, 制約関数は次式のように要約される.

$$f(r, q) = \begin{cases} 1 & \text{if } f_c(\mathbf{r}, \mathbf{q}) \wedge f_v(\mathbf{r}, \mathbf{q}) \wedge f_\theta(\mathbf{r}, \mathbf{q}) \\ 0 & \text{otherwise} \end{cases}. \quad (3.8)$$

式 (3.6) では, 特徴ベクトルが単一の VW に割り当てられると想定している. しかし, 単一の局所特徴を異なる  $v(\mathbf{l})$  と同一の  $c(\mathbf{l})$  および  $\theta(\mathbf{l})$  を持つ複数の局所特徴とみなすことによって, MA によって単一の特徴ベクトルが複数の VW に割り当てられる場合にもこの制約関数を使用することができる. 例えば,  $\mathbf{l}$  の特徴ベクトルが  $v_1(\mathbf{l})$  および  $v_2(\mathbf{l})$  に割り当てられた場合,  $\mathbf{l}$  はそれぞれ  $\{c(\mathbf{l}), v_1(\mathbf{l}), \theta(\mathbf{l})\}$  および  $\{c(\mathbf{l}), v_2(\mathbf{l}), \theta(\mathbf{l})\}$  を持つ 2 つの局所特徴とみなす. その後, これらの局所特徴を個別に制約関数に入力する.

### 3.4.3 アルゴリズム

提案する画像索引付けのための特徴設計手法は, 以下の 4 つの処理で構成される.

1. 移動端末によって撮影される実際のクエリ画像を模擬する人工画像の生成 (3.4.3.1).
2. 人工画像から抽出した局所特徴を用いた信頼度の計算 (3.4.3.2). この処理では, 特定の制約の下で目的関数を最大化する局所特徴の集合を探索する.
3. 人工画像から抽出された特徴ベクトルおよび方向の平均化 (3.4.3.3). この処理の目的は, SS および WGC に対する最適な特徴を得るためである.

4. 同一の VW を複数回選択することを回避しながら，信頼度に従って特徴を選択する非バースト性選択 (3.4.3.4)．この処理の目的は，DB 容量の削減と VW のバースト性の抑制を同時に実現するためである．

### 3.4.3.1 人工画像生成

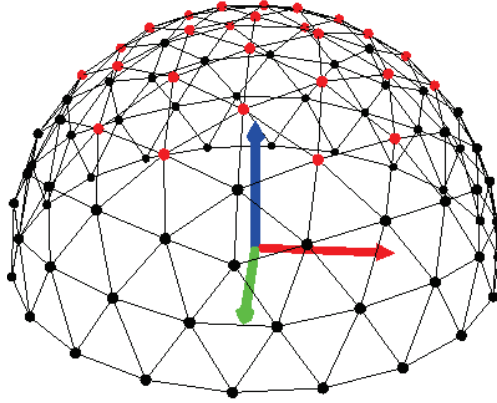
移動端末によって撮影された様々なクエリ画像を模擬するために，単一の参照画像を変形させることによって仮想視点  $V_i$  からの人工画像を生成する．Hinterstoisser らの研究 [162] で行なわれたように，仮想視点位置の様な標本化を行なう．標本間隔を細かくすると計算量が増加するため，本研究では 71 通りの仮想視点 [163] のうち，仰角が  $45^\circ$  を超える 26 通りの仮想視点を使用する．Fig. 3.3 に人工画像生成の概要を示す．Fig. 3.3(a) は仮想視点の模式図であり，赤い点が仰角が  $45^\circ$  を超える仮想視点を表している．Fig. 3.3(b) は人工画像の例と，それを生成する際に用いた仮想視点の仰角  $[\circ]$  と方位角  $[\circ]$  を表している．投影行列としては，Hinterstoisser らの研究 [162] で行なわれたように仮想視点  $V_i$  に対応する Homography 行列  $\mathbf{P}_i$  を計算する．また，スケール変化を模擬するために，各仮想視点に対してスケール因子 1,  $1/\sqrt{2}$ ,  $1/2$  を用いて複数スケール画像を生成する．従って，26 通りの仮想視点と 3 通りのスケール因子に対応する Homography 行列を用いて， $26 \times 3 = 78$  通りの人工画像を生成する．

### 3.4.3.2 信頼度の計算

3.4.1 項で述べた最大被覆問題を実現可能な時間で解くために，貪欲法に基づく近似解法を提案する．すなわち，各反復において最も多くの特徴に一致する特徴を繰り返し選択することによって， $Q$  から  $N$  個の特徴集合を得る．始めに， $Q' = \{\mathbf{q}'_i | i = 1, \dots, M\}$  を  $Q$  の複製とする．そして， $Q$  内の特徴と  $Q'$  内の特徴が一致条件を満たすか否かを判定し，満たす場合には  $Q'$  内の特徴に対してスコアを与える．これにより， $Q'$  の  $i$  番目の特徴  $\mathbf{q}'_i$  は次の値を得，これを特徴信頼度と呼ぶ．

$$s_i = \sum_{\mathbf{q}_j \in Q} z(Q', \mathbf{q}_j) f(\mathbf{q}'_i, \mathbf{q}_j), \quad (3.9)$$

ここで  $f$  は式 (3.8) の制約関数であり， $z(Q', \mathbf{q}_j)$  は現在までの反復において選択された  $Q'$  内のいかなる  $\mathbf{q}'_i$  に対しても  $\mathbf{q}_j$  が一致条件を満たしていない場合に 1 となり，それ以外の場合に 0 となる関数である．全ての  $z(Q', \mathbf{q}_j)$  は初期値を 1 とする．そして，最も

(a) 仮想視点の模式図（赤い点が仰角が  $45^\circ$  を超える仮想視点）

(b) 仮想視点から生成された人工画像の例

Fig. 3.3: 人工画像生成の概要. (b) における各画像下の 2 つの数値はそれぞれ仮想視点の仰角  $[\circ]$  と方位角  $[\circ]$  を表す.

高い信頼度を持つ特徴  $\mathbf{q}'_i$  の選択と、信頼度の再計算の処理を反復する. この時、選択された  $\mathbf{q}'_i$  と  $\mathbf{q}_j$  が一致条件を満たしていた場合、それ以降の反復ではその  $\mathbf{q}_j$  に対応する  $z(Q', \mathbf{q}_j)$  は 0 となる. これは組み合わせ最適化問題であり、多項式時間で計算可能である. Algorithm 1 に、このアルゴリズムの詳細を示す.

特徴  $\mathbf{q}_j$  が特徴  $\mathbf{q}'_i$  と一致条件を満たした場合（つまり  $f(\mathbf{q}'_i, \mathbf{q}_j) = 1$ ）、 $\mathbf{q}_j$  は  $\mathbf{q}'_i$  に関連付けられる.  $C_i$  を  $\mathbf{q}'_i$  に関連付けられた  $\mathbf{q}_j$  の集合、 $C = \{C_i | i = 1, \dots, M\}$  を  $C_i$  の集



---

**Algorithm 1** Feature design method for image indexing

---

**Input:**  $Q = \{\mathbf{q}_1, \dots, \mathbf{q}_M\}$ **Output:**  $X = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ 

```

1: Set  $X \leftarrow \emptyset, Q' \leftarrow Q$ 
2: for  $i = 1 \rightarrow M$  do
3:   Set  $C_i \leftarrow \emptyset$ 
4:   for  $j = 1 \rightarrow M$  do
5:     if  $f(\mathbf{q}'_i, \mathbf{q}_j) = 1$  then
6:       Vote a reliability score to  $\mathbf{q}'_i$ 
7:       Insert  $\mathbf{q}_j$  to  $C_i$ 
8:     end if
9:   end for
10: end for
11: for  $i = 1 \rightarrow M$  do
12:   Select  $\mathbf{q}'_i$  with the highest reliability score:  $\mathbf{q}'_h$ 
13:   Average  $\mathbf{q}'_h$  using  $C_h$ 
14:   if a feature with the same VW as  $\mathbf{q}'_h$  is not included in  $X$  then
15:     Insert the averaged  $\mathbf{q}'_h$  to  $X$  as  $\mathbf{x}$ 
16:   else
17:     Store  $C_h$  as discarded features
18:   end if
19:   Set each  $z(Q', \mathbf{q}_j)$  corresponding  $\mathbf{q}_j \in C_h$  to 0
20:   Re-calculate reliability scores for each  $\mathbf{q}'_i \in Q'$ 
21:   Remove each  $\mathbf{q}_j \in C_h$  from all  $C_i$ 
22:   if  $|X| \geq N$  then
23:     break
24:   end if
25: end for
26: while  $|X| < N$  do
27:   Repeat processing of lines 11–25 using only the discarded features
28: end while

```

---

合とする． $C_i$  は，特徴空間上で  $\mathbf{q}'_i$  によって被覆された特徴  $\mathbf{q}_j$  の集合を意味する．全ての特徴に対する信頼度の計算が完了した後に，最も高い信頼度を持つ特徴  $\mathbf{q}'_i$  を選択する．これを  $Q'$  の  $h$  番目の特徴  $\mathbf{q}'_h$  と表記する．同時に，集合  $C_h$  を用いて特徴  $\mathbf{q}'_h$  を平均化する．この処理の詳細については，3.4.3.3 で説明する．そして，元々の特徴  $\mathbf{q}'_h$  の代わりに，平均化された特徴  $\mathbf{q}'_h$  を特徴  $\mathbf{x}$  として集合  $X$  に追加する．その後，集合  $C_h$  内の特徴  $\mathbf{q}_j$  に対応する  $z(Q', \mathbf{q}_j)$  を 0 にする．そして，式 (3.9) に従って特徴  $\mathbf{q}'_i$  に対する信頼度  $s_i$  を再計算し，全ての集合  $C_i$  ( $i = 1, \dots, M$ ) から集合  $C_h$  内の特徴  $\mathbf{q}_j$  を取り除く．上記の処理を， $N$  個の特徴  $\mathbf{x}$  の集合が得られるまで反復する．

この特徴信頼度に従って指定個数の特徴を選択した場合，次の効果が期待される．

#### 有益な特徴の選択

提案手法では，実際のクエリ画像を模擬した全ての人工画像にわたって一致条件を満たす可能性が高い特徴を選択する．従って，これらの特徴は実際の検索において正しいクエリ特徴と一致する傾向があり，検索結果が大幅に改善されることが期待される．

#### 適応的複数割り当て

提案手法は複数の制約を同時に満たす特徴を関連付けるため，結果として単一の特徴点に対して複数（非固定の  $n$  個）の VW が割り当てられる可能性がある．Fig. 3.1 はこの MA を模式的に表している．この図の中央右側では，同一位置にある青い特徴点を KBC に従って単一の特徴点と見なすことができる．そして，VBC に従ってこの特徴点に “+” および “△” の 2 つの VW を関連付ける．これは MA の結果に似ているが，単一の特徴点に関連付けられた VW の数  $n$  は適応的に決定される．KBC を満たす確率が高いが他の制約を満たす確率が低い特徴に対しては， $n$  が大きくなる．対照的に，全ての制約を満たす確率が高い特徴に対しては， $n$  が小さくなる．従って，同等の DB 容量でより効果的な MA を実現できる．

#### 3.4.3.3 特徴平均化

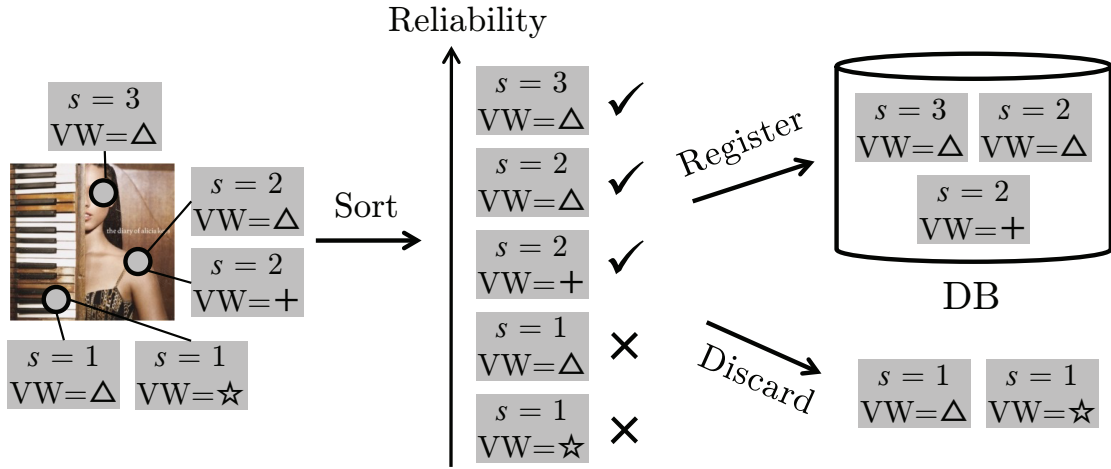
3.3 節で紹介した SS および WGC の性能を改善するために，Algorithm 1 における特徴  $\mathbf{q}'_h$  と関連付けられた全ての特徴の平均を出力する．本研究では，この処理を特徴平均化 (Feature Averaging; FA) と呼ぶ．通常，SS および WGC には正面視点の参照画像か

ら抽出した特徴が使用される．しかし，提案手法においては様々な人工画像から抽出した複数の特徴が  $\mathbf{q}'_h$  に関連付けられ，集合  $C_h$  の要素となる．そのため，集合  $C_h$  内の特徴量（特徴ベクトルおよび方向）を平均化し，特徴  $\mathbf{q}'_h$  の特徴量を得られた平均値に置き換える．これは様々な視点にわたる特徴量の最尤推定であるため，検索の頑健性を向上させることが期待される．すなわち，3.4.2 項で述べた制約の下で最尤の特徴ベクトルおよび方向を用いることによって，SS および WGC の性能を改善する．特徴ベクトルに関しては，次元ごとに値を平均化し，得られた平均値を 2 値化する．方向は一般に，基準方向に対する角度として表現されるため，単純に平均化すると角度の周期性に起因する問題が生じる．この問題に対処するために，単位ベクトル表現に変換した後に全てのベクトルを合算し，これを方向へ逆変換することによって平均化する．

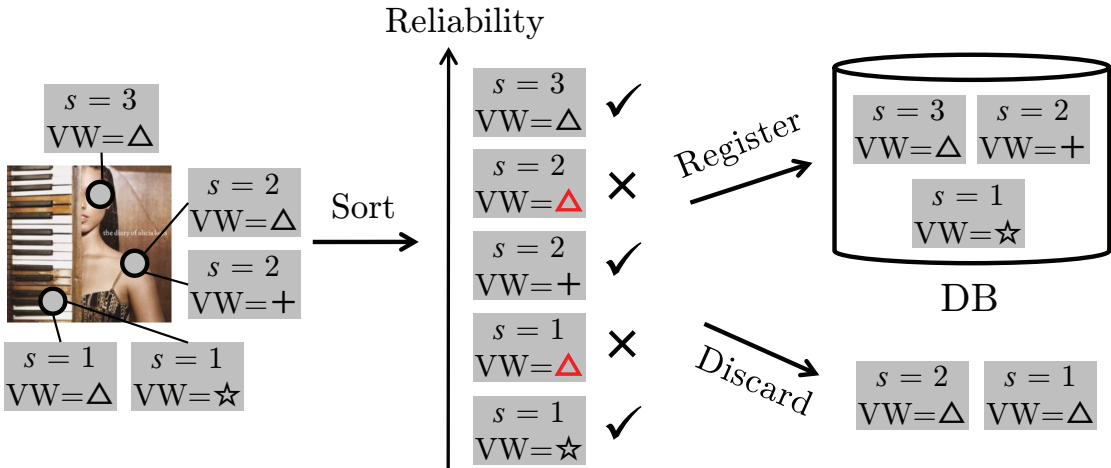
#### 3.4.3.4 非バースト性選択

Jégou らは，VW のバースト性が画像間類似度に悪影響を与えると主張した [156]．そして，このバースト性を抑制するためにスコアリング手法を提案したが，この手法では画像の索引付け処理を変更しないため，DB 容量は削減されない．そのため，本研究では DB 容量の削減と VW のバースト性の抑制を同時に実現するための特徴選択手法を提案し，これを非バースト性選択（Non-Bursty Selection; NBS）と呼ぶ．この選択手法においては，複数の特徴が同一 VW に量子化される場合，信頼度が最も高い特徴のみが DB に登録され，他の特徴は破棄される．

Fig. 3.4 に，NBS の概要を示す．Fig. 3.4(a) および Fig. 3.4(b) はそれぞれ，NBS を適用しない場合とする場合の特徴選択に対応している．Fig. 3.4(a) においては，特徴は信頼度が高い順に単純に選択される．一方，Fig. 3.4(b) においては，信頼度  $s = 3$  および  $VW = \triangle$  を持つ特徴が最初に選択される．信頼度  $s$  に従った場合， $s = 2$  および  $VW = \triangle$  を持つ特徴が次に選択されるべきであるが， $s = 2$  および  $VW = \triangle$  を持つ特徴は同一の VW “ $\triangle$ ” を持つ特徴が既に登録されているために破棄される．この手法を用いて指定個数の特徴を選択することによって，VW のバースト性を直接的に抑制すると同時に，DB 容量を削減することができる．登録された特徴の個数が指定された数に達していない場合，新たに特徴を選択するために直前の処理で破棄された特徴を用いて同様の処理を繰り返す．この手法は，Algorithm 1 の 14–18 行目および 26–28 行目に対応している．



(a) 非バースト性選択なしの特徴選択



(b) 非バースト性選択を適用した特徴選択

Fig. 3.4: 非バースト性選択（NBS）の概要.  $s$  は特徴の信頼度を表す. また, 記号 “✓” および “✗” はそれぞれ選択および破棄を表す.

### 3.5 評価実験

公開データセットを用いた実験により, 提案した特徴設計手法の有効性を評価する. 本実験では, 提案手法は従来手法と比べて移動端末を用いた画像検索に適した DB を構築することを示す.

### 3.5.1 実験設定

本実験では、Stanford mobile visual search データセット\* [90] を使用する。第 2 章で述べた通り、このデータセットは移動端末で撮影されたクエリ画像とその参照画像を含むため、移動端末を用いた画像検索の評価に適している。このデータセットは“Book covers”, “Business cards”, “CD covers”, “DVD covers”, “Landmarks”, “Museum paintings”, “Print documents”, “Video frames” の 8 クラスで構成されている。異なる大きさの画像が含まれているため、参照画像とクエリ画像の両方を VGA (640×480 画素) に収まるように拡張縮する。

検索性能の指標として、MAP (Mean Average Precision) [70] を使用する。また、2 値特徴として、広く知られた手法の中で一般的な画像変形に対して最も頑健 [164] である ORB 特徴 [65] を使用する。本実験では、各画像において平均で 4 つのスケールから 900 個の特徴を抽出する。また、コードブックの VW 数および SS の長さをそれぞれ 1,024 および 64 bit に設定する。式 (3.5) においては、RANSAC の実装 (例えば OpenCV ライブラリ†) における再投影誤差のしきい値として典型的に使用される値に倣い、 $\varepsilon_c = 3$  に設定する。全ての実験は、クロック周波数 3.4 GHz の Intel Core i7-6800K CPU と 32GB のメモリを搭載した PC 上で実行した。

本節においては式 (3.5), 式 (3.6), 式 (3.7) で定義した制約をそれぞれ “KBC”, “VBC”, “OBC” と表記する。また、3.4.3.3 および 3.4.3.4 で述べた処理をそれぞれ “FA” および “NBS” と表記する。

### 3.5.2 方向に基づく制約の影響

式 (3.7) で定義した OBC の有効性を評価する。ここでは、OBC の評価に焦点を当てるために、提案手法から FA および NBS の処理を除外した。本実験では、この提案手法を “Prop”, OBC を除外した提案手法を “Prop w/o OBC” と表記する。

これらの性能を比較するために、式 (3.7) におけるしきい値  $\varepsilon_\theta$  を変化させた場合の MAP の 8 クラス平均を Fig. 3.5 に示す。Prop と Prop w/o OBC を比較すると、OBC

---

\* <https://purl.stanford.edu/rb470rw0983/>

† <https://opencv.org/>

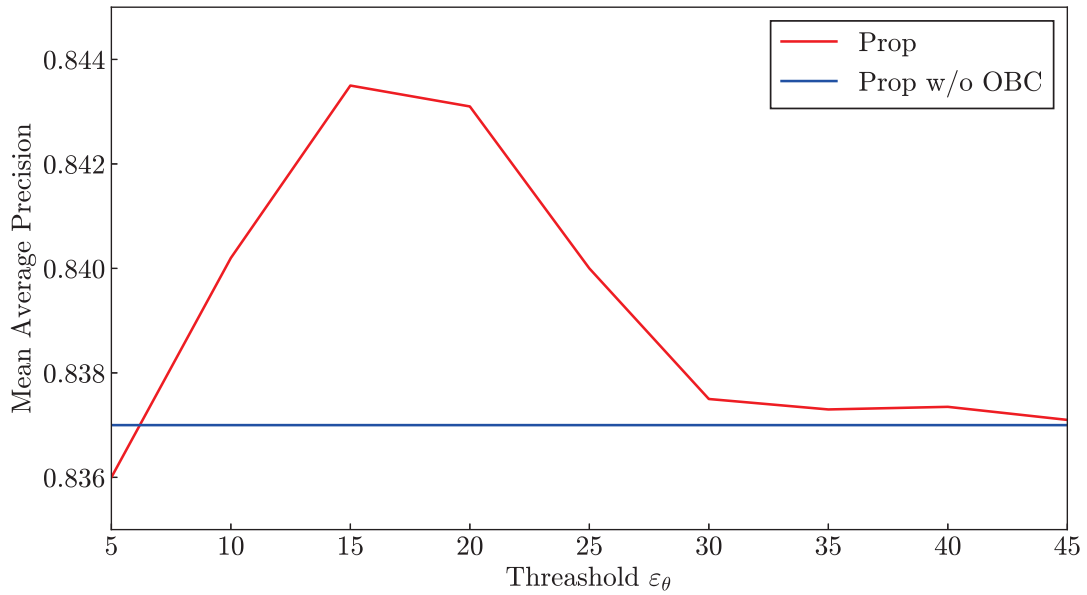


Fig. 3.5: 異なるしきい値  $\varepsilon_\theta$  の下での MAP の 8 クラス平均. 提案手法を “Prop”, 方向に基づく制約 (Orientation-Based Constraint; OBC) を除外した提案手法を “Prop w/o OBC” と表記する.

が検索精度を改善することがわかる. これは, OBC が方向の再現性が高い特徴に対して高い信頼度を与えたためである. すなわち, DB に視点変化によって方向がほとんど変化しない安定した特徴を優先的に登録したためである. 対照的に, 不安定な特徴は DB に登録されない. WGC は方向の差に関して整合性がない特徴を投票から除外するため, OBC によって検索精度が改善する. 平均 MAP の最大値はしきい値が  $\varepsilon_\theta = 15^\circ$  の場合に得られた. このしきい値が小さすぎると, 実際の検索で一致条件を満たす可能性がある特徴が除外される恐れがある. 一方, しきい値が大きすぎると, OBC の効果が低下する. 従って, 検索精度を高めるために以降の実験では常に  $\varepsilon_\theta = 15^\circ$  を使用する.

### 3.5.3 特徴平均化および非バースト性選択の評価

DB 容量が増加する場合の FA と NBS の影響を評価する. 本実験においては, 提案手法を “Prop”, 提案手法から FA を除外した手法を “Prop w/o FA”, 提案手法から NBS を除外した手法を “Prop w/o NBS”, 提案手法から FA と NBS の両方を除外した手法を “Prop w/o FA nor NBS” と表記し, 各手法による検索精度を計測する. これらの手法で

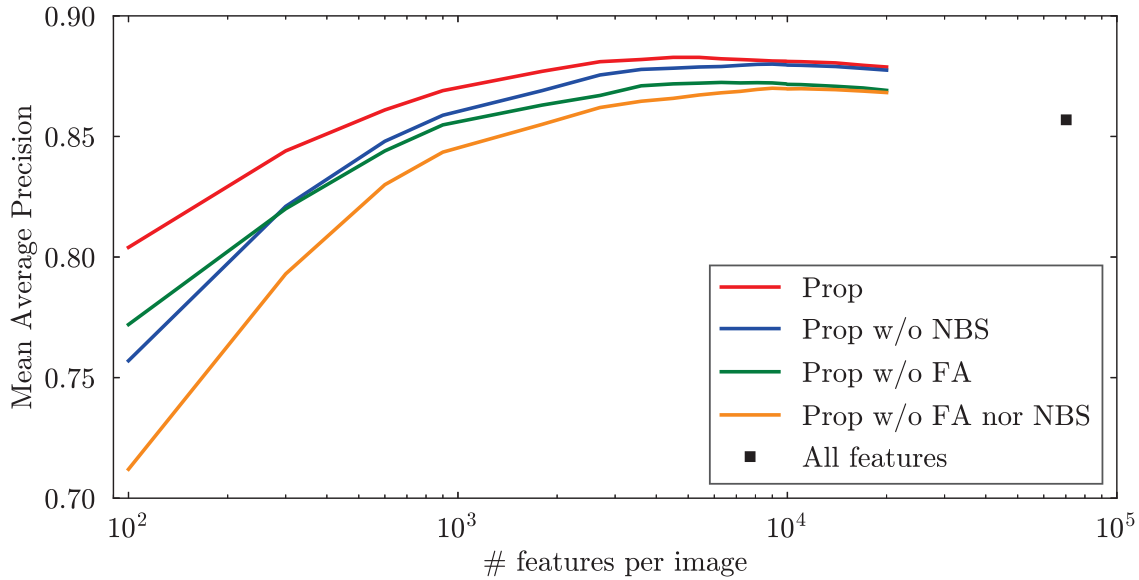


Fig. 3.6: 異なる DB 容量の下での特徴平均化および非バースト性選択の影響. 提案手法を“Prop”, 提案手法から特徴平均化 (Feature Averaging; FA) を除外した手法を“Prop w/o FA”, 提案手法から非バースト性選択 (Non-Bursty Selection; NBS) を除外した手法を“Prop w/o NBS”, 提案手法から特徴平均化と非バースト性選択の両方を除外した手法を“Prop w/o FA nor NBS”と表記する.

は 1 枚の参照画像から 78 通りの人工画像を生成し, それぞれから 900 個の特徴を抽出する. すなわち, 合計で  $78 \times 900 = 70,200$  個の特徴を抽出する. しかし, 3.4.3.2 で述べた処理において, 最も信頼度の高い特徴を選択する際に, その特徴と関連付けられた特徴は以降の処理から除外される. そのため, DB に登録可能な特徴数は, 人工画像から抽出した特徴数よりも少なくなる. DB への特徴の登録数が 20,000 を超えるころから, それ以上登録できなくなる事例が発生し始めたため, 本実験ではこの登録数の上限を 20,000 に設定する. 更に, 参考のために人工画像から抽出した全ての局所特徴を DB に登録する手法の検索精度も計測し, これを“All features”と表記する. この手法では, 画像ごとの特徴の登録数は 70,200 である.

Fig. 3.6 に, 画像ごとの特徴の登録数を変化させ, 各手法の MAP の 8 クラス平均をとった結果を示す. この図では, FA の有無が特徴平均化の効果を, NBS の有無が非バースト性選択の効果を表している. FA の有無を比較することによって, DB 容量とは無関係に, 特徴平均化が常に有効であることがわかる. これは, 平均化された特徴が SS の効

果を改善したため、すなわち 3.3 節で説明したスコアリング手法に従って、様々な視点にわたるクエリ画像から投票されるスコアを改善したためである。平均化された方向もまた、様々な視点にわたってクエリ特徴と参照特徴の方向の差を軽減させることによって、WGC の性能を向上させた。従って、人工画像群から抽出した特徴集合の平均は、それらの内の 1 つから抽出した特徴よりも信頼度が高いと言える。一方、NBS の有無を比較することによって、非バースト性選択が検索精度が改善することがわかる。この結果は、制約を設けずに信頼度が高い順に特徴を登録するよりも、非バースト性制約を課す方が検索に有益な特徴を選択できることを示している。

提案手法では、DB 容量が大きくなるにつれて検索精度が改善する傾向が見られた。これは、DB に登録する特徴数が増えるほど、クエリ特徴と一致条件を満たす可能性が高くなるためである。しかし、特徴の登録数が 3,000 程度になるころに最大値に達し、それ以降は緩やかに低下し始めた。これは、DB 容量の増加が、検索精度の低下を引き起こすためである。言い換えれば、増加した参照特徴が、検索処理において特定のクエリ特徴に対するディストラクタとして振る舞うためである。All features は全ての特徴を DB に登録するにもかかわらず、より少数の特徴のみを登録する提案手法よりも検索精度が低くなった。この原因は、提案手法と同様に、DB 容量の大きさによるものと考えられる。一方、提案手法における NBS の有無と DB 容量の関係に着目すると、DB 容量が大きくなるにつれて検索精度の差が少なくなることがわかる。NBS は特徴を選択する順番を変更する処理であるため、登録数が大きい場合には NBS の有無に関わらず同じ特徴が選択される。そのため、DB 容量の増加に伴って NBS の効果が小さくなる。以上より、提案手法では特徴の登録数が 3,000 程度になるまでは、DB 容量と検索精度の間でトレードオフの関係が成り立つ。提案手法は FA と NBS の効果により、DB 容量が小さい場合であっても高い検索精度を達成する。従って、提案手法は容量が小さい DB の構築に特に適している。

### 3.5.4 従来の画像検索手法との比較

提案手法と従来の画像検索手法である BoBW [151], Zhou ら [152] の手法、ベースライン手法 [153] の検索精度を比較する。提案手法が人工多視点画像から抽出された局所特徴を使用するのに対して、従来の手法は全て正面視点の参照画像から抽出された局所特徴を用いて DB を構築する。更に本実験では、提案手法の各要素技術の効果を調査する。この目的のために、提案手法から各要素技術を除外した場合の検索精度を計測する。検討す



る要素技術は、KBC、VBC、OBC の制約と、FA と NBS の特徴設計処理である。ただし、KBC と VBC は提案手法の最小構成であるため、それらは常に使用する。全ての手法に対して、画像ごとに 900 個の特徴を DB に登録する。

Table 3.1 に、全ての結果をまとめる。ベースライン手法は、3.3 節で紹介した拡張手法が奏功して、他の従来手法よりも高精度を示した。提案手法は、最小構成（つまり KBC と VBC のみ）の場合でさえ、ベースライン手法と比べて精度を 9.6% 改善することができた。これは、様々なクエリ特徴に対して一致条件を満たす可能性が高い（つまり信頼度が高い）特徴を選択することによって参照画像を索引付けするためである。

また、提案手法においては、要素技術 OBC、FA、NBS を追加することによって、最小構成の場合に比べて精度を改善することがわかる。これは、OBC は方向の再現性が高い特徴を選択し、FA が SS および WGC に最適な特徴ベクトルおよび方向を提供し、NBS が DB における VW のバースト性を抑制するためである。これらの要素技術は相補的であり、組み合わせることで効果を高めることができる。全ての要素技術を含めた場合、提案手法はベースライン手法と比べて精度を 12.7% 改善することができた。ただし、NBS は Museum painting クラスや CD covers クラスでは有効ではなかった。これは、これらのクラスがテクスチャに乏しい画像を多く含むために、特徴点の再現性が特に低いためである。NBS は特徴点の再現性が高い特徴を破棄する可能性があるために、このクラスでは精度が低下したと考えられる。

なお、ベースライン手法と提案手法における検索時の平均処理時間はそれぞれ 17.8 ms と 17.5 ms であり、ほぼ同程度であった。これは、ベースライン手法と提案手法の違いが索引付け処理のみであり、検索処理は共通しているためである。提案手法ではベースライン手法によって構築される DB と同一の形式になるように DB を構築するため、検索時に追加的な処理を行なう必要はない。また、本実験においては DB に登録する特徴数を 900 個に統一しているため、DB 容量やメモリ使用量も同値となる。そのため、ベースライン手法と提案手法の検索時間は同程度になったと考えられる。

以上より、提案手法はベースライン手法と比べて DB 容量や処理時間、メモリ使用量に悪影響を及ぼすことなく、より高精度な検索を実現できる。更に、提案手法は人工画像を利用するため、参照物体を異なる視点から実際に撮影した画像のような、追加的な情報も必要としない。これらは、提案手法の実用上、大きな利点である。

Table 3.1: 提案手法と従来の画像検索手法の比較および提案手法の各要素技術の評価．KBC は特徴点の検出に基づく制約 (Keypoint-Based Constraint; KBC), VBC は特徴ベクトルの量子化に基づく制約 (Visual word-Based Constraint; VBC), OBC は方向に基づく制約 (Orientation-Based Constraint; OBC), FA は特徴平均化 (Feature Averaging; FA), NBS は非バースト性選択 (Non-Bursty Selection; NBS) を表す．全ての手法に対して、画像ごとに900個の特徴をDBに登録する．

Method	Element technology					Class								Average
	KBC	VBC	OBC	FA	NBS	Book covers	Business cards	CD covers	DVD covers	Land marks	Museum paintings	Print documents	Video frames	
BoBW [151] Zhou et al. [152] Baseline						0.610	0.173	0.427	0.465	0.080	0.486	0.125	0.584	0.369
						0.874	0.463	0.752	0.811	0.197	0.671	0.423	0.824	0.627
						0.935	0.616	0.882	0.945	0.292	0.743	0.609	0.923	0.742
Prop	x	x				0.944	0.839	0.949	0.977	0.362	0.861	0.780	0.987	0.838
Prop	x	x	x			0.949	0.848	0.946	0.976	0.382	0.877	0.785	0.981	0.843
Prop	x	x	x	x		0.956	0.867	<b>0.965</b>	0.980	0.407	0.890	0.801	0.983	0.856
Prop	x	x	x	x		0.951	0.871	0.959	0.982	<b>0.419</b>	<b>0.895</b>	0.811	0.986	0.859
Prop	x	x			x	0.960	0.854	0.951	0.975	0.379	0.849	0.805	0.992	0.846
Prop	x	x	x		x	0.957	0.882	0.953	0.973	0.380	0.866	0.825	<b>0.999</b>	0.855
Prop	x	x		x		0.961	0.900	0.961	0.983	0.417	0.874	0.850	0.989	0.867
Prop	x	x	x	x	x	<b>0.964</b>	<b>0.901</b>	0.961	<b>0.984</b>	<b>0.419</b>	0.876	<b>0.855</b>	0.989	<b>0.869</b>

### 3.5.5 従来の特徴設計手法との比較

提案手法と従来の特徴設計手法である IFS (Informative Feature Selection) [147] および参照側の MA [71] とを比較する。本実験では、これらの特徴設計手法を従来 of 画像検索手法に適用する。IFS は実際に撮影した多視点画像を必要とし、それらの幾何的關係を推定するために RANSAC を使用する。提案手法との公正な比較のために、本実験では人工画像および幾何的關係の真値を IFS に適用する。Zhou ら [152] の手法は特徴ベクトルの部分集合を直接的に VW として使用するため、単一の特徴ベクトルを  $k$  近傍の VW に割り当てる MA は、この手法には適用することができない。従って、MA を Zhou らの手法に適用する手法はこの比較には含めないものとする。MA は単一の特徴ベクトルを  $k$  個の VW に割り当てるため、DB 容量を  $k$  倍 ( $k = 1, \dots, 10$ ) に増大させる。IFS および提案手法は、DB 容量が MA のものと等しくなるように特徴を選択する。Hessian Affine 共変検出器 [165] と SIFT 記述子の組み合わせは視点変化に対して頑健であるため、Wang らの研究 [147] で使用されているが、それらは計算量が大きいため、本実験では ORB 特徴に対して IFS を適用する。

Fig. 3.7 に、全ての実験結果を示す。この図では、MA は画像ごとに 900 個の特徴を抽出し、各特徴を  $k$  個 ( $k = 1, \dots, 10$ ) の VW に割り当てる。IFS および提案手法は、MA における各  $k$  の場合と DB 容量が等しくなるように特徴を選択する。この結果より、提案手法が DB 容量にかかわらず最良の精度を達成することがわかる。IFS は DB 容量の増加に伴って徐々に精度を改善するが、その精度は常に提案手法よりも低い。これは、IFS が常に特徴の検出と特徴ベクトルの量子化の再現性が高い特徴を選択するわけではないためである。すなわち、IFS は再現性のない特徴点を破棄するが、特徴点の検出の再現性が高い特徴を優先的に選択するわけではない。更に、IFS は TF-IDF の考えに基づいて特徴を選択するが、同一位置にある特徴点に関連付けられた特徴ベクトルの量子化の再現性は考慮していない。MA は量子化誤りを軽減することによって従来 of 画像検索手法の精度を改善する。しかし、十分に大きい  $k$  を用いた場合でさえ、MA は DB 容量が最小の場合の提案手法の精度を上回ることができない。これは、提案手法が特徴ベクトルの量子化誤りだけでなく、特徴点の検出誤りと方向の量子化誤りまで軽減するためである。従って、提案手法は画像の索引付けに関して MA よりも効果的な特徴設計を実現することができる。

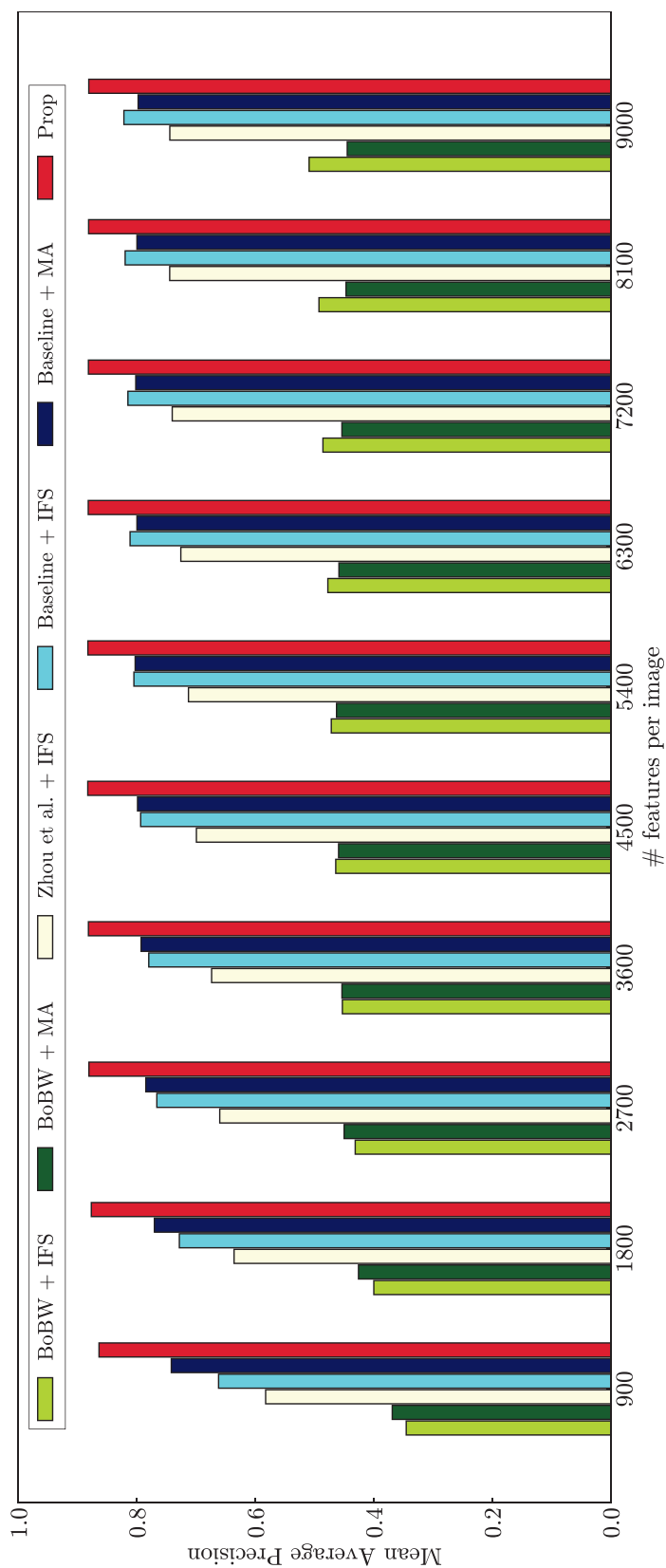


Fig. 3.7: 異なる DB 容量の下での提案手法と従来の特徴設計手法の比較. IFS は Informative Feature Selection を, MA は Multiple Assignment を表す. MA は画像ごとに 900 個の特徴を抽出し, 各特徴を  $k$  個 ( $k = 1, \dots, 10$ ) の VW に割り当てる. IFS および提案手法は, MA における各  $k$  の場合と DB 容量が等しくなるように特徴を選択する.

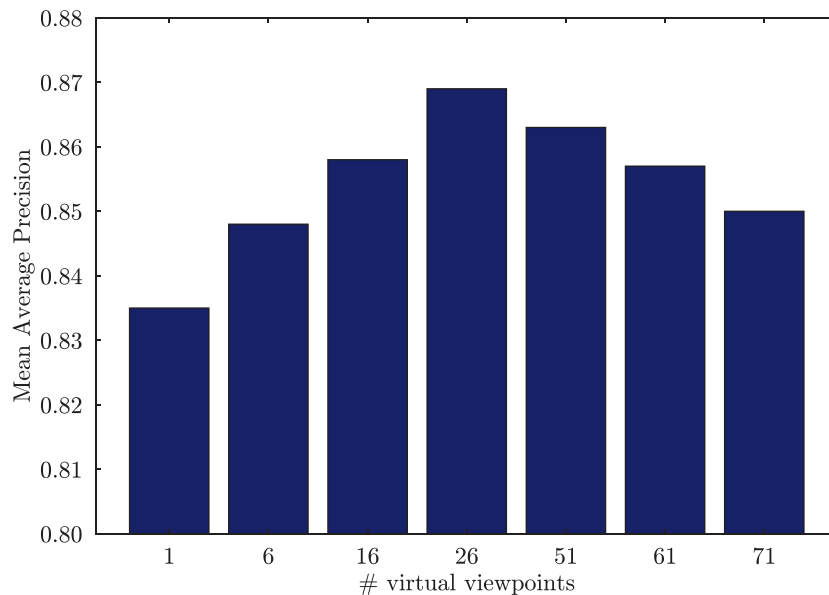


Fig. 3.8: 人工画像生成に用いる仮想視点の個数と検索精度の関係.

IFS と MA を比較すると、IFS は DB 容量が小さい場合には MA に劣るが、DB 容量が大きい場合には MA を上回る。これは、DB 容量が小さい場合には、IFS は特徴点の検出の再現性が高い特徴が必ずしも多く登録されていないためである。しかし、DB 容量の増加に伴って IFS においても徐々にそのような特徴が登録されるため、結果的に精度が改善する。Wang らの研究 [147] では、DB 容量が小さい場合でも IFS は精度を向上させるが、これは Hessian Affine 共変検出器が視点変化に対して頑健に特徴点を検出できるためである。一方、MA は共通の特徴点を使用するため、DB 容量への依存性が相対的に小さくなる。

### 3.5.6 人工画像生成の効果分析

人工画像生成に用いる仮想視点の個数と検索精度の関係を調査する。本実験では、Fig. 3.3(a) に示す仮想視点の内、仰角がしきい値以上となるものを用いて人工画像を生成する。このしきい値は  $90^\circ$ ,  $75^\circ$ ,  $60^\circ$ ,  $45^\circ$ ,  $30^\circ$ ,  $15^\circ$ ,  $0^\circ$  に設定する。これらを設定した場合の仮想視点の個数は、それぞれ 1 個, 6 個, 16 個, 26 個, 46 個, 61 個, 71 個となる。提案手法によって DB に登録する特徴の個数は、常に 900 個とする。

Fig. 3.8 に、仮想視点の個数ごとの MAP の 8 クラス平均を示す。この図より、仮想視点の個数を増やすほど検索精度が改善する傾向があることがわかる。これは、仮想視点の

Table 3.2: 人工画像生成数を変えた場合の提案手法の処理時間 [ms]. 各  $M$  における人工画像生成 (Synthetic), 人工画像からの局所特徴抽出 (Extract), 特徴ベクトルの量子化 (Quantize), Algorithm 1 に記述される特徴設計 (Design), 特徴平均化 (FA), 非バースト性選択 (NBS) のそれぞれの処理時間をまとめる.

Processing	$M$ (# features extracted from synthetic images)				
	9,000	18,000	27,000	36,000	45,000
Synthetic	$3.85 \times 10^1$	$7.84 \times 10^1$	$1.17 \times 10^2$	$1.52 \times 10^2$	$1.86 \times 10^2$
Extract	$1.00 \times 10^2$	$2.02 \times 10^2$	$3.01 \times 10^2$	$3.97 \times 10^2$	$4.84 \times 10^2$
Quantize	$2.42 \times 10^1$	$4.81 \times 10^1$	$7.24 \times 10^1$	$9.58 \times 10^1$	$1.20 \times 10^2$
Design	$1.62 \times 10^2$	$5.70 \times 10^2$	$1.25 \times 10^3$	$2.30 \times 10^3$	$3.67 \times 10^3$
FA	$2.28 \times 10^1$	$4.21 \times 10^1$	$6.17 \times 10^1$	$8.00 \times 10^1$	$1.09 \times 10^2$
NBS	$3.91 \times 10^{-1}$	$5.98 \times 10^{-1}$	$8.22 \times 10^{-1}$	$1.04 \times 10^0$	$1.27 \times 10^0$
Total	$3.50 \times 10^2$	$9.41 \times 10^2$	$1.80 \times 10^3$	$3.02 \times 10^3$	$4.57 \times 10^3$

パターンが多いほど、多様な視点変化に対して頑健な特徴を選択できるためである。しかし、仮想視点の個数が 26 個（つまりしきい値  $45^\circ$  の場合）より大きくなる場合には、検索精度が低下した。この理由は、仮想視点が実際のクエリ画像の視点と類似しなくなるためである。すなわち、ほとんどのクエリ画像の視点は仰角が  $45$  度以上の仮想視点で模擬できるために、それ以下の仰角を持つ仮想視点を追加しても、検索精度が改善しないと考えられる。一方、DB に登録する特徴の中に、小さい仰角の視点からの特徴が増えることによって、大きい仰角の視点からの特徴が相対的に少なくなる。これにより、実際のクエリ画像と類似した視点からの特徴が DB に登録されにくくなるため、検索精度が低下する。

また、人工画像の生成数異なる場合の、提案手法の処理時間を計測する。初めに、10 枚、20 枚、30 枚、40 枚、50 枚の人工画像を生成し、それぞれから画像ごとに 900 個の特徴を抽出する。結果として、それぞれの場合の特徴数  $M$  は 9,000, 18,000, 27,000, 36,000, 45,000 となる。

Table 3.2 に、人工画像生成数を変えた場合の人工画像生成 (Synthetic), 人工画像からの局所特徴抽出 (Extract), 特徴ベクトルの量子化 (Quantize), Algorithm 1 に記述される特徴設計 (Design), 特徴平均化 (FA), 非バースト性選択 (NBS) のそれぞれの

処理時間をまとめる．FA および NBS については，それらが Design には含まれないように独立に計測した．

Synthetic および Extract は人工画像の枚数に対して線形に増加した．また，Quantize は  $M$  に対して線形に増加した．この傾向は，それらのアルゴリズムを考慮すれば明らかである．一方，Design は  $M$  に対して 2 次的に増加した．Algorithm 1 に示す通り，特徴設計処理は  $M$  に対する 2 重反復を含むため，Design の計算量は  $O(M^2)$  となる．本研究の実装では，Algorithm 1 の 12 行目の処理は線形探索に基づくため，その計算量は  $O(M)$  である．Algorithm 1 の 19 から 21 行目の計算量は無視できるほど小さい．3.4.3.2 で述べた  $C_h$  の大きさは線形に増加する傾向があるため，FA も線形に増加した．NBS は  $M$  の増加に伴ってわずかに増加した．これは  $M$  が大きいほど VW にバースト性が生じる可能性が高くなるためである．以上をまとめると，提案手法は多項式時間，すなわち  $O(M^2)$  で処理することが可能である．

## 3.6 まとめ

本章では，本論文で取り組む 3 つの課題である (i) DB 内のデータ数の削減，(ii) DB 内の各データの小容量化，(iii) 効率的なデータの照合の内，(i) に取り組んだ．すなわち，移動端末上で動作する画像検索アプリケーションにおいて，特徴の選択に基づいてデータ数を削減することにより，高速かつ高精度な照合と特徴 DB の小容量化を両立した．従来の特徴設計手法においては，画像の索引付けに用いる特徴の個数を増加させることによって画像検索の精度を改善することができるが，一方で DB 容量が大きくなるという問題があった．この問題に対処するために，画像検索に対して特に有益な特徴を選択することによって，検索精度を改善しつつ DB を小容量化する手法を提案した．

具体的には，本章では，2 値特徴と BoVW に基づくモバイル画像検索のための特徴設計手法を提案した．本研究では特徴選択を最大被覆問題として定式化し，実現可能な計算時間で解決するための近似解法を導入した．この定式化において，目的関数は画像検索手法における一致条件と整合性を持つように設計した．視点変化を含む画像で構成された公開データセットを用いた実験により，提案手法の有効性を示した．すなわち，ベースラインとなる画像検索手法と比べて DB 容量および処理時間を増加させることなく，検索精度を改善することを示した．





## 第 4 章

# 特徴の変換に基づく環境地図の圧縮

本論文ではスマートフォンによる画像検索とコネクテッドカーによる自己位置推定というアプリケーションにおいて、高速かつ高精度な照合と特徴 DB の小容量化を両立することを目指す。この目的を達成するために、本論文では (a) 特徴の選択, (b) 特徴の変換, (c) 類似尺度の変更, の観点に基づいて, (i) DB 内のデータ数の削減, (ii) DB 内の各データの小容量化, (iii) 効率的なデータの照合, の課題に取り組む。この内、本章ではコネクテッドカーによる自己位置推定に着目し, (b) 特徴の変換に基づいて (ii) DB 内の各データの小容量化を実現する。具体的には、3D NDT によって構築した地図データを、よりデータ量の小さい表現形式に変換する、環境地図の圧縮手法を提案する。本手法では、集約と量子化という異なる種類の複数の変換を組み合わせることにより、自己位置推定の精度や処理時間を維持しつつ DB 容量を削減する。

車両の自己位置推定手法の 1 つに、予め生成された環境地図と現在位置からセンサで計測した 3 次元点群を照合するスキャン照合がある。大規模な環境を表現する 3 次元点群は容量が大きいため、記憶容量や通信帯域の制限に対処するためにはデータ量の小さい地図表現に変換する必要がある。自己位置推定の精度を維持しつつ環境地図の容量を削減する代表的な手法に 3D NDT があるが、この手法による容量の削減率は限定的であり、大規模な環境地図の容量は依然大きい。本章では、3D NDT によって構築された環境地図の容量を削減するために、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、個々のボクセルが持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 と比べて環境地図の容量を 100 分の 1 から 1,000 分の 1 以下にすることを示す．

本研究の貢献は，以下のように要約される．

1. 3D NDT によって構築した地図データに対してベクトル量子化を適用することで，データの圧縮と復元をする枠組みを提案する．
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特徴の変換手法を提案する．
3. 無圧縮の 3D NDT との性能比較を行ない，提案手法が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す．

以下，4.1 節で本研究の基本的な考え方について述べる．その後，4.2 節で本研究と関連が強い 3D NDT とベクトル量子化について概説し，4.3 節で 3D NDT によって構築した地図データの新たな圧縮手法を提案する．そして，4.4 節では地図データの圧縮比と自己位置推定の精度を評価し，4.5 節で本章をまとめる．

## 4.1 はじめに

コネクテッドカー上で動作する自己位置推定アプリケーションを実現するために，本研究では 3D NDT によるスキャン照合に焦点を当てる．3D NDT は高速かつ高精度な自己位置推定と環境地図の圧縮を両立する手法である．データの圧縮効率の指標として，次式で表される圧縮比（Compression ratio）がある．

$$\text{Compression ratio} = \frac{\text{Uncompressed capacity}}{\text{Compressed capacity}}. \quad (4.1)$$

これは，無圧縮の場合の容量（Uncompressed capacity）と圧縮後の容量（Compressed capacity）の比であり，高いほど圧縮効率が良いことを意味する．3D NDT による環境地図の圧縮比は 100 から 1,000 程度であり，地図の容量を大きく削減する．しかし，大規模な環境を想定する場合には，圧縮後であっても地図の容量が数 TB から数百 TB の規模になる恐れがあるため，更なる容量削減が必要である．従って本研究では，3D NDT 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、ボクセルが持つ正規分布パラメータを表すデータを圧縮する．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が持つパラメータ数が少ないため、圧縮比の大きさは限定的である．そこで、空間をブロックと呼ばれる単位に分割し、各ブロックに含まれる複数のボクセルが持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、3D NDT 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 2 値で表現した状態情報を追加する．以下、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．

## 4.2 自己位置推定とデータ圧縮

本節では、本研究との関連が強い 3D NDT とベクトル量子化について概説する．

### 4.2.1 3D Normal Distributions Transform

3D NDT [6] は、3 次元点群  $\mathcal{F}$  をボクセル単位で正規分布 (Normal Distribution; ND) パラメータ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  $\mathcal{S}$  と  $\mathcal{F}$  の位置合わせを行なうことによって自己位置を求め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 1 辺の長さが  $r$  (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  $\mathcal{P} = \{\mathbf{p}_m \in \mathbb{R}^3 | m = 1, \dots, M\}$  を次式に従って ND パラメータ  $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$  に変換する．

$$\boldsymbol{\mu} = \frac{1}{M} \sum_{m=1}^M \mathbf{p}_m = r \begin{pmatrix} \mu_x \\ \mu_y \\ \mu_z \end{pmatrix}, \quad (4.2)$$

$$\boldsymbol{\Sigma} = \frac{1}{M-1} \sum_{m=1}^M (\mathbf{p}_m - \boldsymbol{\mu})(\mathbf{p}_m - \boldsymbol{\mu})^\top = r^2 \begin{pmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{xy} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{xz} & \sigma_{yz} & \sigma_{zz} \end{pmatrix}. \quad (4.3)$$

ここで  $\boldsymbol{\mu} \in \mathbb{R}^3$  は平均、 $\boldsymbol{\Sigma} \in \mathbb{R}^{3 \times 3}$  は分散共分散行列である．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．ここでは、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．従って、 $\mathcal{F}$  を 3D NDT 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. この時,  $\boldsymbol{\mu}$  の要素,  $\boldsymbol{\Sigma}$  における分散の要素, 共分散の要素が取りうる値の範囲はそれぞれ  $[0, r)$ ,  $[0, r^2)$ ,  $(-r^2, r^2)$  であり, ボクセルの解像度  $r$  に依存する.

自己位置推定処理では, スキャン点  $\mathbf{s} \in \mathcal{S}$  と NDT 地図の間で評価関数を最小化する並進と回転を表す位置姿勢変換パラメータ  $\boldsymbol{\xi} = (t_x, t_y, t_z, \theta_x, \theta_y, \theta_z)^\top$  を求める. 初めに, 座標  $\mathbf{v}$  を用いて点  $\mathbf{s}$  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, それらの対応間の負の尤度スコアを表す評価関数  $E$  を計算する.

$$P(\mathbf{s}) \propto \exp\left(-\frac{(\mathbf{s} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{s} - \boldsymbol{\mu})}{2}\right), \quad (4.4)$$

$$E(\boldsymbol{\xi}) = -\sum_{\mathbf{s} \in \mathcal{S}} P(T(\boldsymbol{\xi}, \mathbf{s})), \quad (4.5)$$

ここで  $P$  は各ボクセルが持つ ND パラメータ  $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$  を用いて定義された確率密度関数,  $T$  は位置姿勢変換関数である. そして, Newton 法を用いてパラメータ  $\boldsymbol{\xi}$  を反復的に最適化する.

#### 4.2.2 ベクトル量子化に基づくデータ圧縮

ベクトル量子化 [166, 167] 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  $\mathcal{C} = \{\mathbf{c}_k \in \mathbb{R}^D | k = 1, \dots, K\}$  (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  $d(\mathbf{x}, \mathbf{c}_k)$  が最小となるベクトル  $\mathbf{c}_k$  を探索し, そのインデックス  $k \in \{1, \dots, K\}$  に置き換える. ここで,  $d$  は任意の距離尺度である. また,  $\mathcal{C}$  を用いて  $k$  を逆量子化することにより, ベクトル  $\hat{\mathbf{x}} = \mathbf{c}_k$  を復元することができる. この逆量子化の際には, ベクトル  $\mathbf{x}$  と  $\hat{\mathbf{x}}$  の間で  $d(\mathbf{x}, \mathbf{c}_k)$  の誤差が生じる.

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, 画像や音声など様々なデータの圧縮に利用されている [168, 169]. 例えば画像の圧縮を行なう際には,  $n_x \times n_y$  個の画素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. 従って, 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 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。従って、実環境とコードブックの間でベクトルが類似し、高い圧縮効率を得られることが期待される。また、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、本章ではベクトル量子化を用いた NDT 地図の圧縮を検討する。

### 4.3 特徴の変換に基づく環境地図の圧縮

本節では、NDT 地図をベクトル量子化に基づいて圧縮およ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、データの圧縮比の大きさが限定的である。そのため、画像圧縮技術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。この状態情報は、占有／非占有を 2 値で表現した離散的なスカラ値である。従って本研究では、NDT 地図に対する圧縮比を改善するために、ブロック単位で状態情報および 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必要である。

#### 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および復元された状態情報は量子化誤差に起因して  $[0, 1]$  の連続値で表現されるため、本来は 2 値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

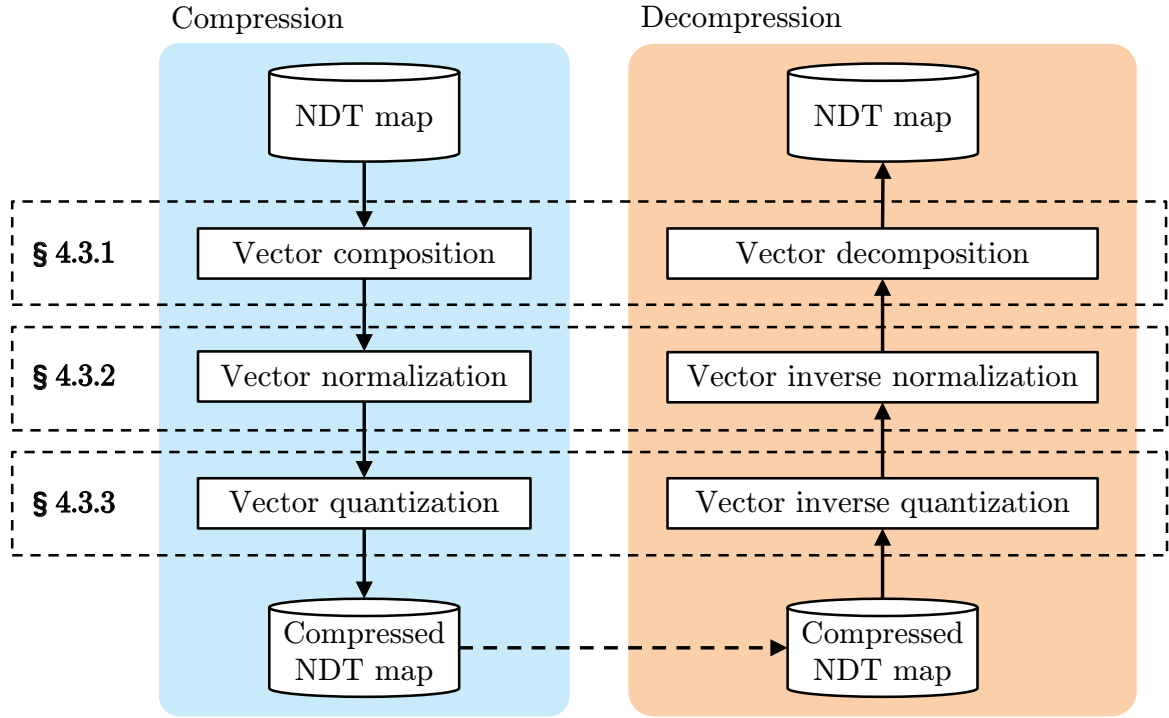


Fig. 4.1: 提案する環境地図の圧縮手法の枠組み.

れるため、各要素の単位や取りうる値の範囲が一致しない。

### 3. 高次元ベクトルのクラスタリング

複数のボクセルが持つパラメータを連結する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。

Fig. 4.1 に圧縮処理と復元処理からなる提案手法の枠組みを示す。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、圧縮処理を介して NDT 地図を圧縮する。また逆に、圧縮されたデータが与えられた場合、復元処理を介して NDT 地図を復元する。上記の各問題はそれぞれ、この図におけるベクトル合成・分解（4.3.1 項）、ベクトル正規化・逆正規化（4.3.2 項）、ベクトル量子化・逆量子化（4.3.3 項）によって対処する。

#### 4.3.1 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、合計  $N = n_x \times n_y \times n_z$  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。Fig. 4.2 に、 $n_x = 2, n_y = 1, n_z = 2$

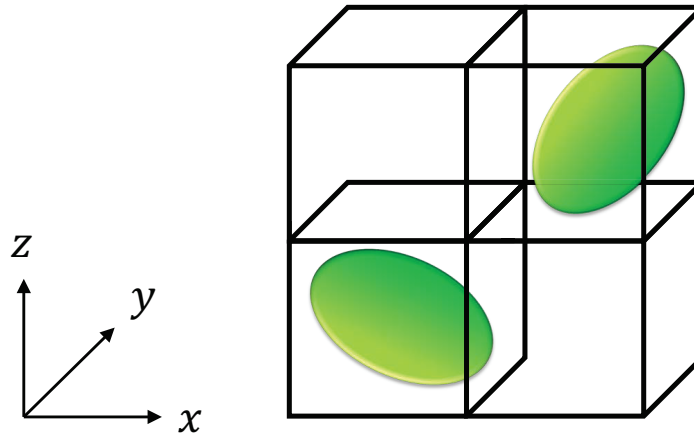


Fig. 4.2: ブロックの模式図．黒色の線で描かれた立方体がボクセルを，緑色の楕円体がボクセルの持つ ND パラメータを表現している．何も描かれていないボクセルは，ND パラメータを持たない非占有ボクセルであることを意味する．

( $N = 4$ ) 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，緑色の楕円体がボクセルの持つ 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 パラメータを持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，1 つ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である．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抑制される恐れがある．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量子化誤差を含むため，しきい値を用いて 2 値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 または 0 で表現した 2 値ベクトル  $\mathbf{B} \in \{0, 1\}^N$  を構築する．この  $\mathbf{B}$  は，ブロック内のボクセルにインデックス  $i \in \{1, \dots, N\}$  を割り当て，各ボクセルが持つ占有／非占有を表す 2 値を  $i$  番目の要素としたベクトルである．

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．ただし、式 (4.3) より分散共分散行列  $\Sigma$  は対称行列であり、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．具体的には次式で表すように、平均  $\mu \in \mathbb{R}^3$  の要素と、分散共分散行列  $\Sigma \in \mathbb{R}^{3 \times 3}$  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．

$$\mathbf{V}_i = h(\mu^T, r^2(\sigma_{xx}, \sigma_{xy}, \sigma_{xz}, \sigma_{yy}, \sigma_{yz}, \sigma_{zz})), \quad (4.6)$$

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．

次に、非占有ボクセルに仮の ND パラメータを与え、ベクトル化する．ここで、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れを復元した ND パラメータを持つボクセルが得られる．本章では、仮の ND パラメータとして平均  $\mu$  にはボクセルの中心座標を、分散共分散行列  $\Sigma$  には単位行列  $\mathbf{I}$  を与えるものとする．ただし、式 (4.3) より  $\Sigma$  は  $r^2$  に比例するため、 $\mathbf{I}$  には  $r^2$  を乗ずる．また、式 (4.4)、(4.5) より、 $\Sigma$  の要素の取りうる値の範囲は 3D NDT の評価関数に影響を及ぼす．従って、この範囲を調整するための係数  $\delta$  も乗じ、 $\Sigma = \delta r^2 \mathbf{I}$  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  $N$  個のボクセルの  $\mathbf{V}_i$  を結合することによって、次式で表されるベクトル  $\mathbf{X}$  を合成し、ブロックを表現する．

$$\mathbf{X} = h(\mathbf{B}, \mathbf{V}_1, \dots, \mathbf{V}_N). \quad (4.7)$$

このベクトルは  $N$  次元の  $\mathbf{B}$  と、9 次元の  $\mathbf{V}_i$  を  $N$  個結合したものであるため、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．ここでは、1 つのブロックに含まれる全てのボクセルの座標の中から、 $x, y, z$  の全ての方向に対して最小の値を持つ 1 つの座標  $\mathbf{v} \in \mathbb{R}^3$  を選択する．そして、このボクセルの座標  $\mathbf{v}$  をブロックの座標  $\mathbf{v}'$  とする．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  $\mathbf{X}$  と座標  $\mathbf{v}'$  の集合に変換される．

提案手法では座標  $\mathbf{v}'$  のデータ量を削減するために、可逆圧縮手法である 8 分木圧縮 [170, 171] 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  $\mathbf{X}$  の集合と、8 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  $\hat{\mathbf{X}}$  であり、 $\mathbf{X}$  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  $\mathbf{B}$  を推定する。

$$b_i^* = \begin{cases} 1 & \text{if } \hat{b}_i \geq 0.5 \\ 0 & \text{otherwise} \end{cases}, \quad (4.8)$$

ここで  $\hat{b}_i$  は、圧縮と復元を経た後の 2 値ベクトル  $\hat{\mathbf{B}}$  の  $i$  番目の要素である。離散化する際のしきい値は、推定結果のバランスを調整するパラメータとみなせる。本研究では、想定される範囲  $[0, 1]$  の中間であり、なおかつ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。

#### 4.3.2 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  $\mathbf{X}$  に対する正規化を行なう。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。

$\mathbf{X}$  のうち、分散共分散行列  $\Sigma$  の要素の単位は、平均  $\mu$  の要素の単位の 2 乗である。従って、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があるため、符号付き平方根をとるものとする。

次に、ベクトル  $\mathbf{X}$  の全ての要素を  $[0, 1]$  の範囲に収めるために min-max 正規化を行なう。ここでは  $\mathbf{X}$  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、各々の範囲を正規化する。

しかし、個々の  $\mathbf{X}$  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なった後に、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  $\mathbf{X}$  に対して  $\Phi$  を用いて正規化を行なう。なお、 $\mathbf{B}$  の要素については、常に  $[0, 1]$  の範囲に収まるため、正規化を行なわない。

従って、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．

$$\alpha_{\text{norm}} = \frac{\alpha - \alpha_{\min}}{\alpha_{\max} - \alpha_{\min}}, \quad (4.9)$$

$$\beta_{\text{norm}} = \frac{\sqrt{\beta} - \beta_{\min}}{\beta_{\max} - \beta_{\min}}, \quad (4.10)$$

$$\gamma_{\text{norm}} = \frac{\text{sgn}(\gamma) \sqrt{|\gamma|} - \gamma_{\min}}{\gamma_{\max} - \gamma_{\min}}, \quad (4.11)$$

ここで  $\alpha$ ,  $\beta$ ,  $\gamma$  はそれぞれベクトル  $\mathbf{X}$  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範囲を復元する．

$$\alpha = \alpha_{\text{norm}} (\alpha_{\max} - \alpha_{\min}) + \alpha_{\min}, \quad (4.12)$$

$$\beta = \{\beta_{\text{norm}} (\beta_{\max} - \beta_{\min}) + \beta_{\min}\}^2, \quad (4.13)$$

$$\gamma'_{\text{norm}} = \gamma_{\text{norm}} (\gamma_{\max} - \gamma_{\min}) + \gamma_{\min}, \quad (4.14)$$

$$\gamma = \text{sgn}(\gamma'_{\text{norm}}) \gamma'^2_{\text{norm}}. \quad (4.15)$$

### 4.3.3 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  $\mathbf{X}_{\text{norm}}$  を, Euclidean 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  $\mathbf{X}_{\text{norm}}$  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

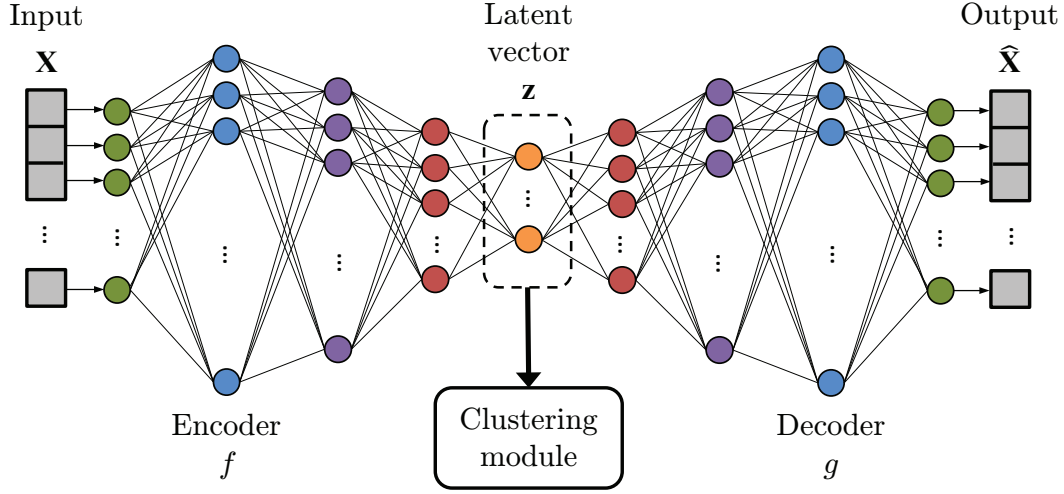


Fig. 4.3: DCN [173] の概要. 入力ベクトルを符号化器を用いて潜在空間に写像することによって次元削減し, 復号化器を介して入力と同一次元のベクトルを復元する. そして, 次元削減されたベクトルを  $k$ -平均法に基づいてクラスタリングする.

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  $K$  個の部分集合にクラスタリングすることで実現する. 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 (一般には  $k$ -平均法が用いられることが多い) を用いて訓練用のベクトル集合をクラスタリングする. しかし, この手法はベクトルの次元が高い場合には, いわゆる次元の呪いによって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 [172]. 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) [173] 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.

Fig. 4.3 に, DCN の概要を示す. 入力ベクトルを符号化器 (Encoder) を用いて潜在空間に写像 (Latent vector) することによって次元削減し, 復号化器 (Decoder) を介して入力と同一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  $k$ -平均法に基づいてクラスタリングする. 入力ベクトルは 1 つのブロックを表現する  $D = 10N$  次元のベクトル  $\mathbf{X}_{\text{norm}}$  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．

$$\mathbf{z} = f(\mathbf{X}_{\text{norm}}), \quad (4.16)$$

$$\mathcal{L} = \|g(\mathbf{z}) - \mathbf{X}_{\text{norm}}\|_2^2 + \frac{\lambda}{2} \|\mathbf{z} - \mathbf{c}_k\|_2^2, \quad (4.17)$$

ここで  $f$  および  $g$  はそれぞれ符号化器および復号化器による写像関数， $\mathbf{z}$  は次元削減されたベクトル， $\mathbf{c}_k$  は  $\mathbf{z}$  が割り当てられるクラスタの中心， $\|\cdot\|_2^2$  は Euclidean 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， $k$ -平均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およ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 [174–176]．本研究では 100 万個以上の大規模なデータのクラスタリングを想定しているため，データ量に対する拡張性が高い 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，これを他の手法に置き換えても構わない．

## 4.4 評価実験

公開データセット中の実データに対して，データの圧縮と復元を経て NDT 地図を作成し，提案する自己位置推定手法の有効性を評価する．

### 4.4.1 実験設定

本実験では，LiDAR センサや GPS，IMU を搭載した車両によって市街地で計測したデータから構成される KITTI データセット [135, 136] を用いる．第 2 章で述べた通り，このデータセットは車両の自動走行に関する様々なアプリケーションの性能評価に広く利用されるものである．本実験では，Visual Odometry 向けのデータセットを用いて，自己位置推定の性能を評価する．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 [137] を用いる.

初めに, GPS および IMU データが含む位置と姿勢を用いて複数のスキャンデータを統合することにより 3 次元点群地図を構築した. その後, この 3 次元点群を元に NDT 地図を生成した. 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した. 提案手法におけるボクセル合成数には  $N = 8, 27, 64$  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\*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が無作為に選択し, 真値から離れる位置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なう. 位置姿勢変換では, 半径 2 m 以内の無作為な並進と  $[-5^\circ, 5^\circ]$  の範囲の  $z$  軸周りの無作為な回転を組み合わせる. そして各スキャンデータに対して 100 回の自己位置推定試行を行ない, 真値との RMSE を求める.

提案手法では, ボクセルの占有/非占有を推定するため, これを誤る可能性がある. この推定精度を評価するための指標として, 次式で表されるボクセル IoU (Intersection over Union) を使用する.

$$\text{IoU} = \frac{\sum_i \mathbb{1}[Y_{\text{gt}}(i)Y_{\text{est}}(i) > 0]}{\sum_i \mathbb{1}[Y_{\text{gt}}(i) + Y_{\text{est}}(i) > 0]}. \quad (4.18)$$

ここで  $i$  はボクセルのインデックス,  $\mathbb{1}$  は条件が成立すれば 1 を, そうでなければ 0 をとる指示関数,  $Y_{\text{gt}}$  および  $Y_{\text{est}}$  はそれぞれボクセルの占有/非占有に関する真値と推定値である.

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  $\hat{\mathbf{V}}_i$  の最近傍となる真値のベクトル  $\mathbf{V}_i$  との Euclidean 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 4.3.2 項で述べた単位の正規化を施す.

---

\* <https://github.com/boyangumn/DCN/>

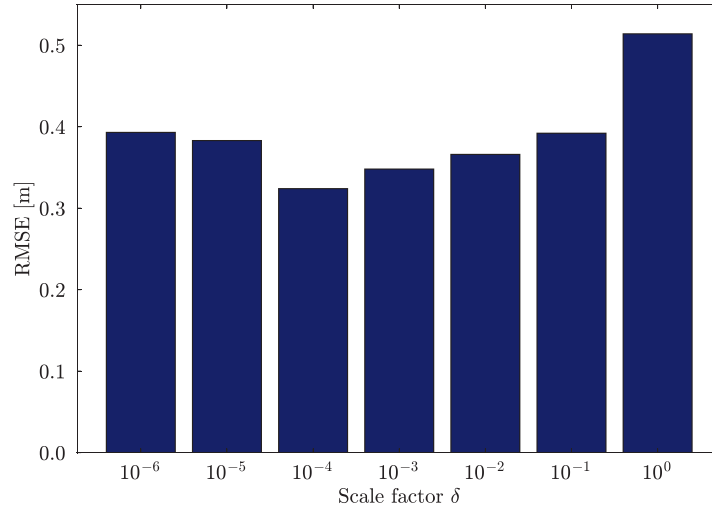


Fig. 4.4: 異なる  $\delta$  の下での自己位置推定精度.

#### 4.4.2 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行列  $\Sigma = \delta r^2 \mathbf{I}$  の係数  $\delta$  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  $\delta \in \{10^{-6}, \dots, 10^0\}$  を選択した．この実験ではボクセルの解像度  $r$  を 2 m，コードブックの大きさ  $K$  を  $2^{16}$ ，ボクセル合成数  $N$  を 64 に固定し、各  $\delta$  における RMSE を調査した．

結果を Fig. 4.4 に示す．このグラフより、 $\delta = 10^{-4}$  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関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．4.2.1 項で説明したように、3D NDT では点と ND パラメータの対応を構築し、式 (4.4), (4.5) 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．従って、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  $\mathbf{s}$  に対して  $P(\mathbf{s}) = 0$  であれば自己位置推定への悪影響を抑制できる．式 (4.4) より、任意の点  $\mathbf{s}$  が与えられた場合、 $\Sigma^{-1}$  がスカラ値の大きいスカラ行列に近似するほど  $P(\mathbf{s}) = 0$  に近づけることができ、結果として式 (4.5) 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 が悪化した．これは、復元された分散共分散行列  $\hat{\Sigma}$  の対角要素と非対角要素の大きさ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。この復元誤差の大きさが  $\Sigma = \delta r^2 \mathbf{I}$  の対角要素にとって相対的に小さい場合には、 $\hat{\Sigma}^{-1}$  がスカラ行列に近似する。しかし、 $\delta$  が小さくなるほどこの対角要素の値が小さくなるため、 $\hat{\Sigma}^{-1}$  がスカラ行列に近似しなくなる。結果として式 (4.4) における  $P(\mathbf{s})$  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。

#### 4.4.3 コードブックの大きさの影響

次に、コードブックの大きさ  $K$  の性能への影響を検証する。検証するパラメータには、 $K \in \{2^1, \dots, 2^{18}\}$  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。Fig. 4.5(a) から Fig. 4.5(d) に  $K$  とボクセル IoU、復元損失、地図圧縮比、自己位置推定精度各々の関係を描画したグラフを示す。

Fig. 4.5(a) より、 $K$  が大きくなるほどボクセル IoU が改善することが分かる。これは、コードブックが含む  $\mathbf{B}$  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、ボクセル IoU は  $r$  に依存しないことが分かる。これは、4.3.2 項で述べた正規化によって、クラスタリングを行なう際の  $r$  依存性を解消しているためである。

Fig. 4.5(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を確認できる。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 の大きさが  $r$  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 IoU が小さくなり、結果として NND が大きくなるためである。

Fig. 4.5(c) は、無圧縮の地図の容量を圧縮後の地図の容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  $r$  でもほぼ同等の値が得られることより、圧縮比はあまり  $r$  に依存しないことが分かる。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高いことが分かる。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。

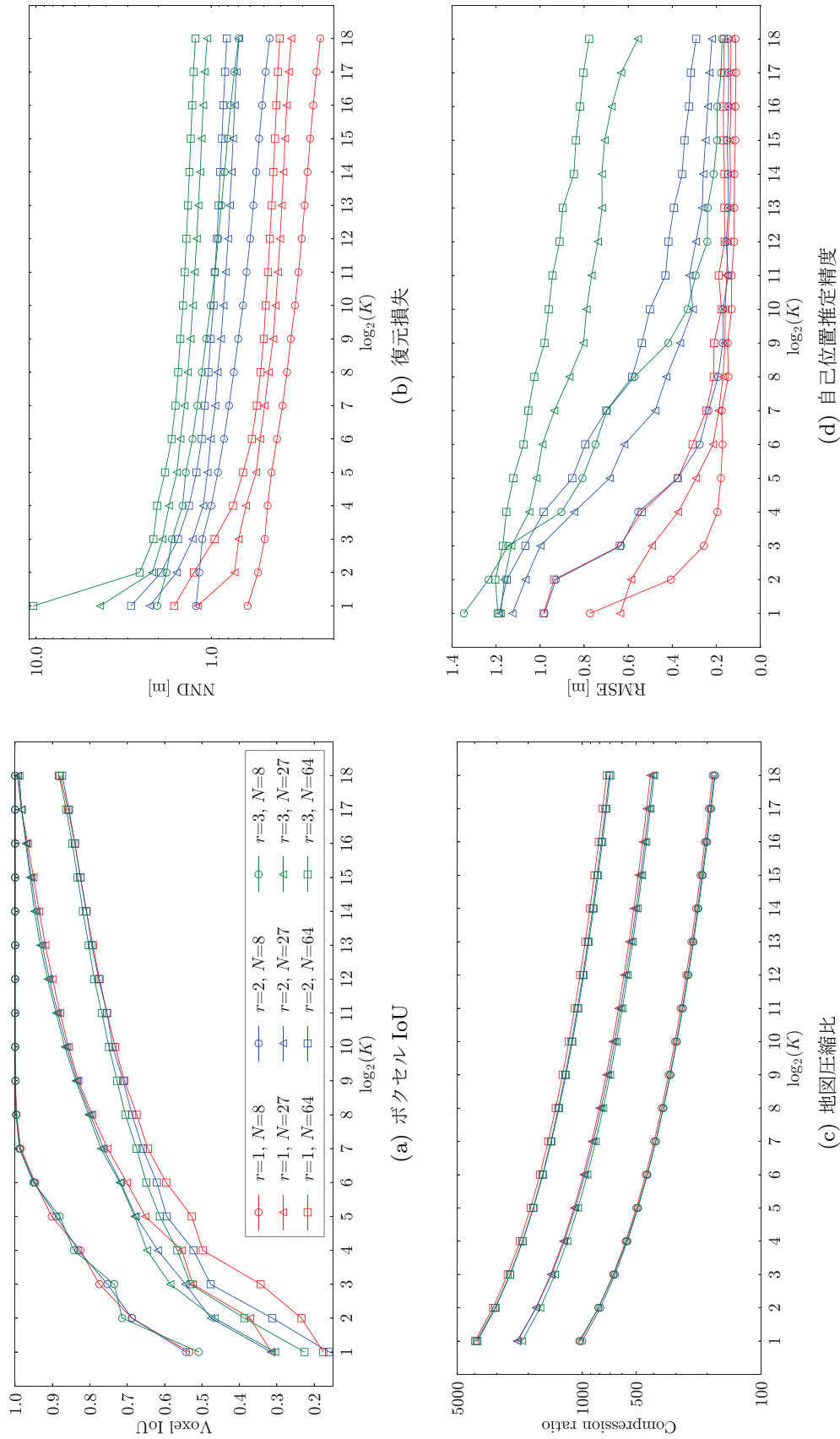


Fig. 4.5: 異なるコードブロックの大きさでの性能評価.



Fig. 4.5(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, 上述のボクセル IoU および 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 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, Fig. 4.5(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. 従って, 任意の  $K$  を選択することによって RMSE と地図容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について, 真値を用いて TP (True Positive), FP (False Positive), FN (False Negative), TN (True Negative) に分類し, Fig. 4.6 に  $r = 1$  m の場合におけるそれらの割合を  $K$  ごとに示す. Fig. 4.6(a), Fig. 4.6(b), Fig. 4.6(c) はそれぞれ  $N = 8, 27, 64$  の場合に対応する. 同様に, Fig. 4.7 および Fig. 4.8 に,  $r = 2$  m および  $r = 3$  m の場合の結果をそれぞれ示す.

全ての  $r$  においてほぼ同様の結果が得られることから, ボクセルの占有／非占有の推定精度は  $r$  にほとんど影響されないことがわかる. これは, 4.3.2 項で述べたベクトル正規化・逆正規化の効果である. 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.  $N$  が大きいほど失敗率が低下しにくくなるのは,  $N$  個のボクセルの占有／非占有のパターンが増加するためである. すなわち, ベクトル  $\mathbf{B}$  のパターンには  $2^N - 1$  通りの可能性があり,  $N$  が大きいほど  $K$  通りのコードブックで表現することが困難になるためである. また,  $N$  が大きいほど TN の割合が増加するのは, より大きなブロックで空間を分割するために, ブロック内の非占有ボクセルの割合が増加するためである. その原因は, 評価用データセットから構築された実環境を表す NDT 地図において, ボクセルが空間的非一様に分布することである. 同様の理由で,  $N$  が大きいほど FP に比べて FN の割合が大きくなることを確認できる. すなわち, コードブックの学習に用いたブロック内の非占有ボクセルの割合が増加するために, 推定結果が非占有となる頻度が高くなるためである. ただし, 最も  $N$  が大きい  $N = 64$  の場合であっても,  $K$  が大きくなるにつれて FP と FN の割合の差が少なくなることがわかる. 従って, 大きな  $K$  を用いた場合には, FP と FN の間で適切なバランスが保たれる.

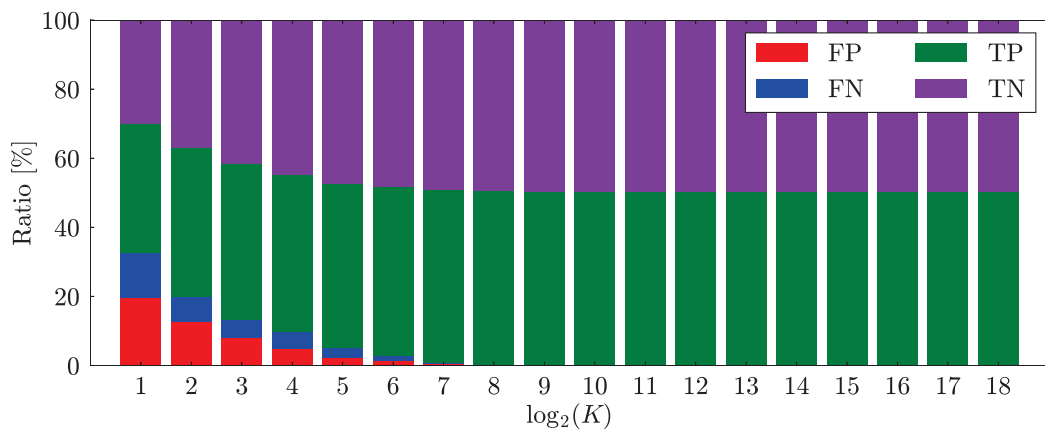
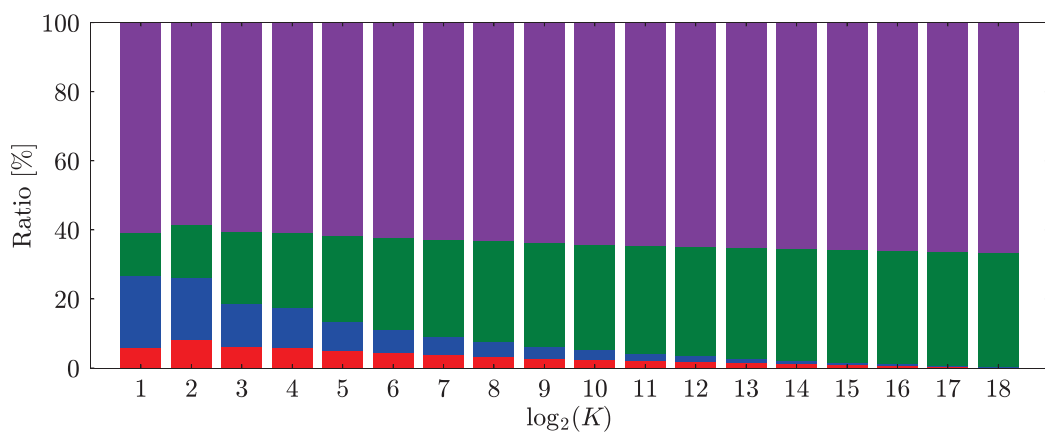
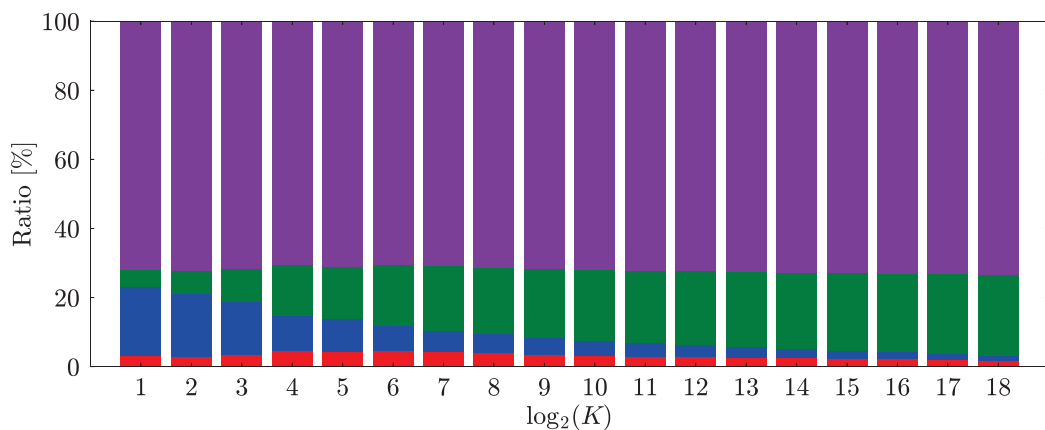
(a)  $N = 8$ (b)  $N = 27$ (c)  $N = 64$ 

Fig. 4.6:  $r = 1$  の場合のボクセルの占有／非占有推定の詳細. 占有／非占有に関する推定結果を Ground-truth を用いて TP (True Positive), FP (False Positive), FN (False Negative), TN (True Negative) に分類した場合のそれらの割合を  $K$  ごとに示す.

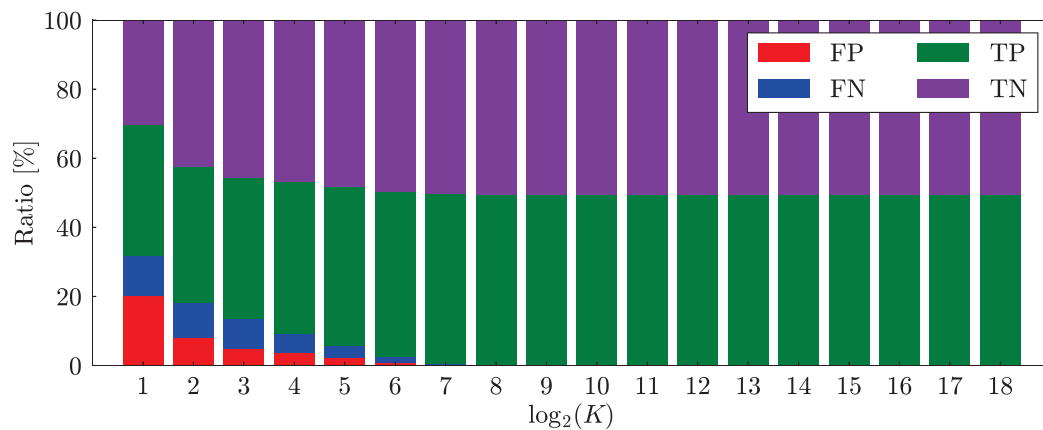
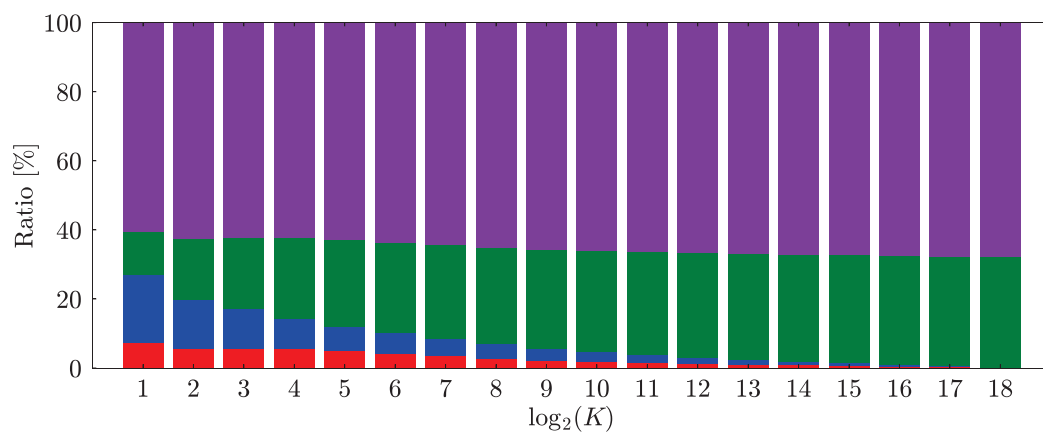
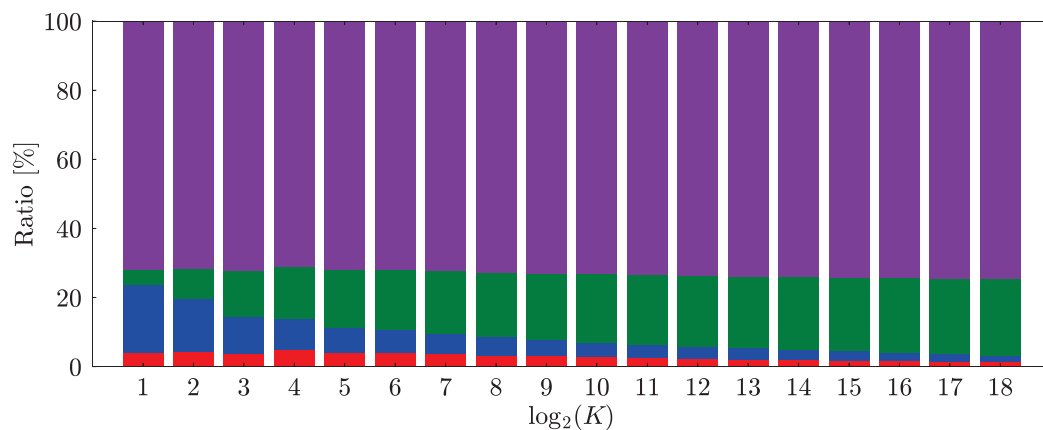
(a)  $N = 8$ (b)  $N = 27$ (c)  $N = 64$ 

Fig. 4.7:  $r = 2$  の場合のボクセルの占有／非占有推定の詳細. 占有／非占有に関する推定結果を Ground-truth を用いて TP (True Positive), FP (False Positive), FN (False Negative), TN (True Negative) に分類した場合のそれらの割合を  $K$  ごとに示す.

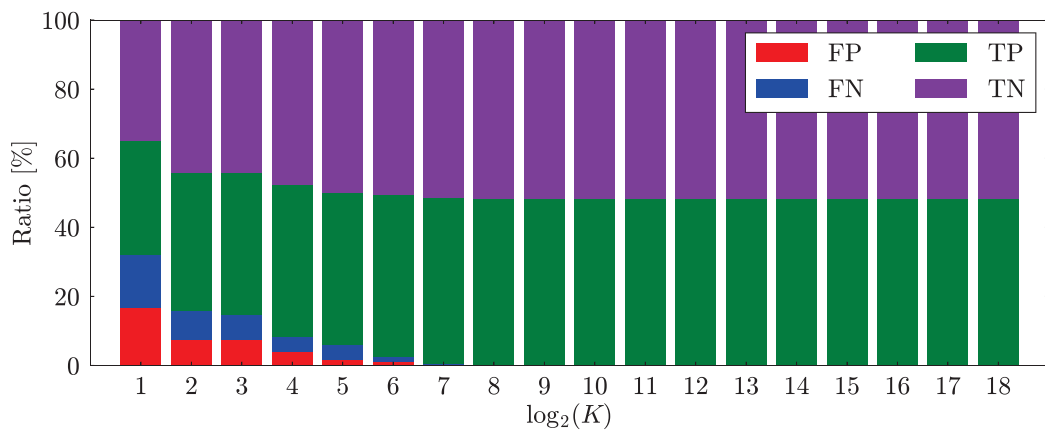
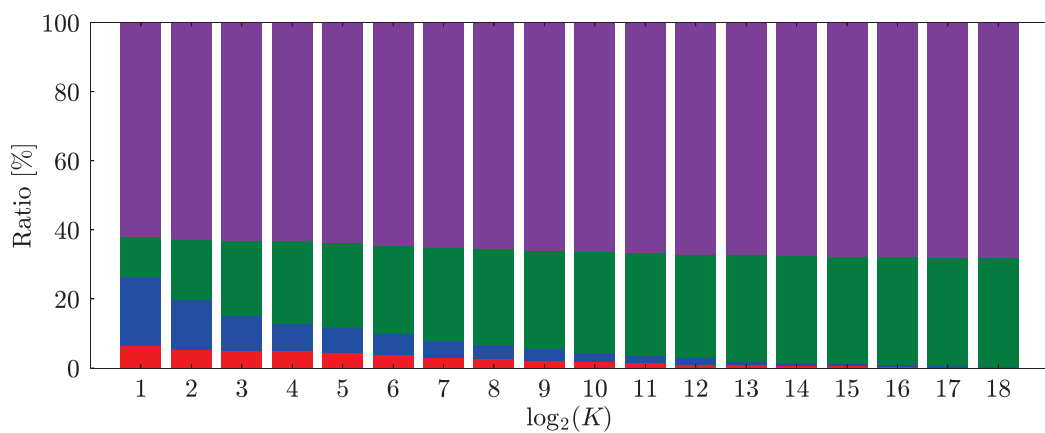
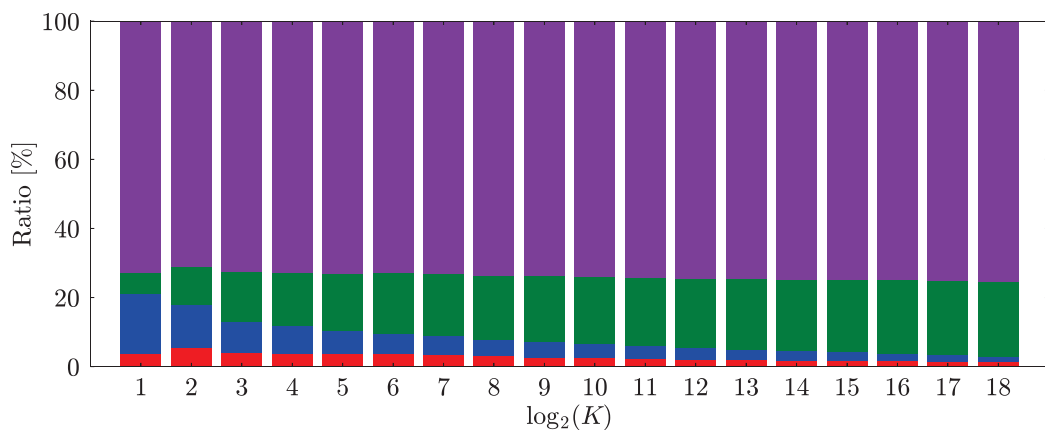
(a)  $N = 8$ (b)  $N = 27$ (c)  $N = 64$ 

Fig. 4.8:  $r = 3$  の場合のボクセルの占有／非占有推定の詳細. 占有／非占有に関する推定結果を Ground-truth を用いて TP (True Positive), FP (False Positive), FN (False Negative), TN (True Negative) に分類した場合のそれらの割合を  $K$  ごとに示す.

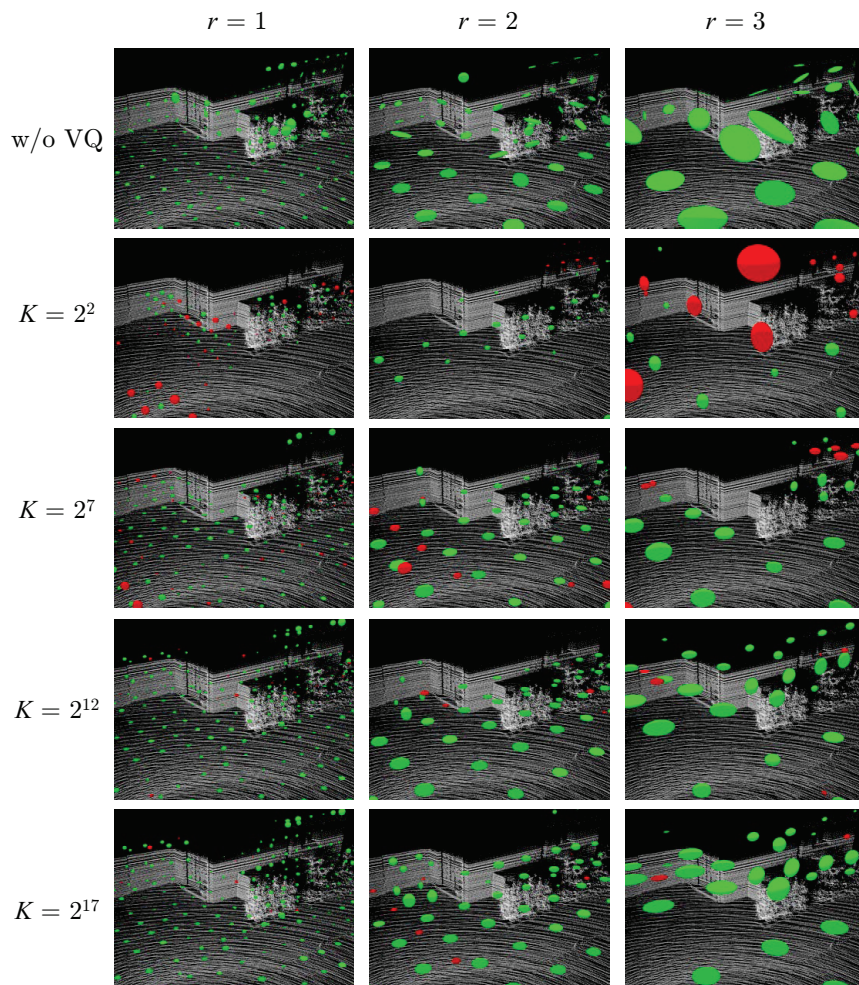


Fig. 4.9: NDT 地図の可視化. 最上段の行は無圧縮 (w/o VW) の NDT 地図を表し, それ以外の各行は異なるコードブックの大きさ  $K$  の下で提案手法によって圧縮と復元を行った NDT 地図を表す. また, NDT 地図の元となった点群も白色の点として重畳する.

#### 4.4.4 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 ND パラメータを楕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楕円体を表現した. 使用するパラメータにはボクセル合成数  $N = 64$ , ボクセル解像度  $r \in \{1 \text{ m}, 2 \text{ m}, 3 \text{ m}\}$ , コードブックの大きさ  $K \in \{2^2, 2^7, 2^{12}, 2^{17}\}$  を選択した.

Fig. 4.9 に, 視覚化の結果を示す. 最も上段の行は無圧縮 (w/o VQ) 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なった 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。緑色の楕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、赤色の楕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す。 $r$  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、楕円体の個数が少なくなる。また、 $K$  が大きくなるほどボクセル IoU が改善するため、赤色の楕円体が減少することが確認できる。緑色の楕円体については、 $K$  が小さい時には無圧縮の地図とは形状差が大きい、 $K$  が大きくなるほど類似していく傾向が見られた。

#### 4.4.5 3次元点群地図に対する圧縮比

本節では、3次元点群地図の容量を基準とした圧縮比を評価する。次の5通りの手法を用いた際の圧縮比と自己位置推定精度を調査する。

##### 1. NDT

標準的な 3D NDT [6] である。本章では、NDT 地図は  $\Theta = \{\mu, \Sigma, \mathbf{v}\}$  の集合として表現する。

##### 2. NDT-Octree

NDT 地図に対して 8 分木圧縮 [170, 171] を適用する手法である。本手法は、ボクセルの座標  $\mathbf{v}$  のみを可逆圧縮する。

##### 3. NDT-DS

NDT 地図に対してダウンサンプリングを適用する手法である。本手法は、 $\Theta = \{\mu, \Sigma, \mathbf{v}\}$  の集合から無作為にダウンサンプリングを行なう。

##### 4. 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  $\mu$  と分散共分散行列  $\Sigma$  を単一のベクトルで表現し、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  $\mathbf{v}$  への 8 分木圧縮なども適用する。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。

##### 5. Prop

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ボクセルの解像度を  $r \in \{1 \text{ m}, 2 \text{ m}, 3 \text{ m}\}$  の 3 通りに、NDT-VQ および Prop におけるコードブックの大きさを  $K \in \{2^1, \dots, 2^{18}\}$  に設定した。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。車線単位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 [177] であり、先行研究 [114, 178] 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 Fig. 4.10 に示す。

NDT の圧縮比は  $r = 1$  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するためである。その一方で 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ためである。

NDT-Octree の圧縮比は  $r = 1$  で 99.2,  $r = 2$  で 388.6,  $r = 3$  で 858.8 であった。また、8 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圧縮比は全ての  $r$  において 1.24 であり、効果は限定的であった。この原因は、占有ボクセルのパラメータ  $\Theta = \{\mu, \Sigma, \mathbf{v}\}$  の 15 パラメータの内、座標  $\mathbf{v}$  の 3 パラメータに対してのみデータ圧縮を行なっているためである。

NDT-DS ではダウンサンプリング率が高いほど圧縮比が高くなり、それに伴って RMSE が悪化した。許容可能な RMSE の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。ダウンサンプリング率はそれぞれ 90%, 90%, 85% に対応する。

NDT-VQ では  $K$  が小さいほど圧縮比が高くなり、それに伴って RMSE が悪化した。許容可能な RMSE の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  $r$  を持つ NDT を基準とした場合の圧縮比は  $r = 1$  で 261.5,  $r = 2$  で 167.0,  $r = 3$  で 142.6 であった。この結果より、 $r$  が小さいほど NDT を基準とした圧縮比が高くなる傾向があることがわかる。これは、RMSE の大きさが  $r$  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。

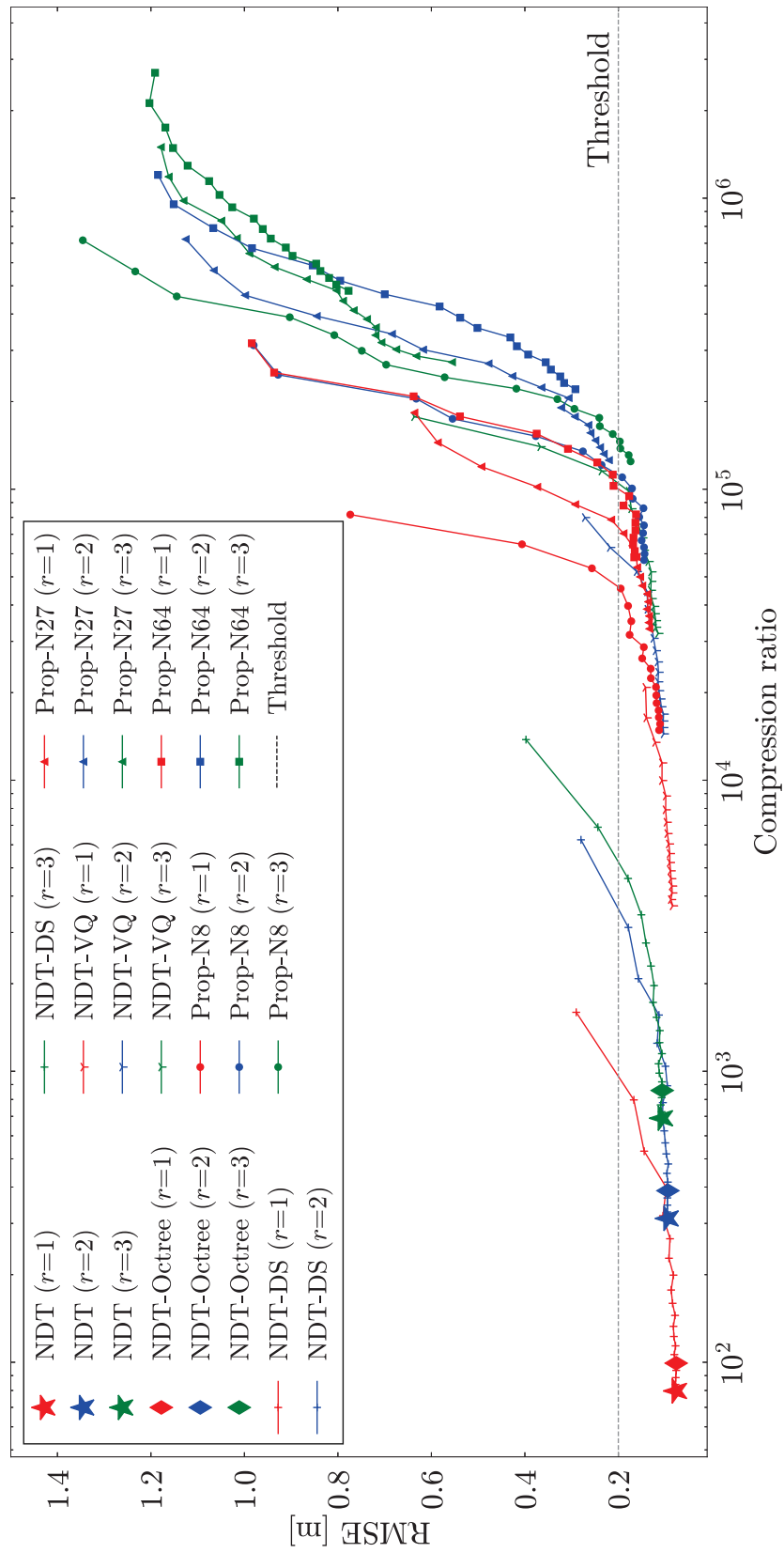


Fig. 4.10: 3次元点群地図を基準とした圧縮比と自己位置推定精度の関係.



提案手法 (Prop-\*) の圧縮比は使用した  $K$  と  $N$  に依存する．許容可能な RMSE の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，地図容量を元の大きさのおよそ 10 万分の 1 にする．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 よりも圧縮比を高めることができる．また，この時のボクセルの占有／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高くなることからわかる．式 (4.4), (4.5) に示す通り，3D NDT では全てのスキャン点  $\mathbf{s}$  と ND パラメータの対応から得られる  $P(\mathbf{s})$  の総和をスコアとする．提案手法における FP に対しては，4.4.2 項で述べた通り， $P(\mathbf{s}) = 0$  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．従って提案手法では， $r = 1$  において 7.5% という高い失敗率を示していても，自己位置推定精度を維持することができる．一方， $r = 2$  および  $r = 3$  における許容可能な失敗率が低い理由は，FP や FN の影響というよりも，むしろ許容可能なボクセル合成数  $N$  に起因すると考えられる．Fig. 4.5(d) より， $r$  が等しい場合には  $N$  が大きいほど RMSE が大きくなることからわかる．提案手法では RMSE の大きさが  $r$  に比例するため，固定された RMSE のしきい値を下回するためには， $r$  が大きいほど小さな  $N$  が必要となる可能性が高い．そのため，本実験における  $r = 2$  および  $r = 3$  の場合には  $N = 8$  の場合にしか RMSE のしきい値を下回ることができなかったと考えられる．Fig. 4.7(a) および Fig. 4.8(a) より  $N = 8$  の場合には多くの  $K$  で失敗率が低くなることからわかる．従って， $r = 2$  および  $r = 3$  の場合には許容可能な  $N$  が小さいため，結果として失敗率も低くなったと考えられる．

#### 4.4.6 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なう地図データの  $r$  を常に一致させていた．しかし，提案手法はベクトル正規化によって  $r$  への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

化を行なうことができる．本実験では，これらの  $r$  が異なる場合に RMSE に生じる影響を調査する．検証するパラメータにはボクセル解像度  $r \in \{1\text{ m}, 2\text{ m}, 3\text{ m}\}$ ，ボクセル合成数  $N \in \{8, 27, 64\}$ ，コードブックの大きさ  $K \in \{2^1, \dots, 2^{18}\}$  を用いた．

Table 4.1(a) から Table 4.1(c) に， $N$  ごとの結果を示す．この結果より， $\mathcal{C}$  を作成するための訓練用データの  $r$  (A) とベクトル量子化を行なう地図データの  $r$  (B) を一致させると，ほとんどの場合に最良の RMSE が得られることがわかる．一方，A と B の値が離れるほど，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なう地図データの間で，ベクトルの分布が異なるためと思われる．両者のベクトルは実環境から計測された 3 次元点群地図から作成されるため，A と B の不一致は 3 次元点群地図の縮尺の変化に相当する．縮尺の変化は作成されるベクトルの分布を変化させるため，学習したベクトル集合  $\mathcal{C}$  とベクトル量子化を行な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， $\mathcal{C}$  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，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影響も小さい傾向がある．そのため，A と B をわずかに変化させる場合には， $\mathcal{C}$  を再学習せずとも実用上の問題はないと考えられる．

#### 4.4.7 逐次的な自己位置推定

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市街地における 3.7 km の経路にわた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なう．誤差を計算する際には，各スキャン点群に対応する，GPS および IMU によって計測された位置と姿勢を真値とみなす．評価指標として，進行方向への位置誤差 (Longitudinal error)，その横方向への位置誤差 (Lateral error)，進行方向に対する方向誤差 (Orientation error)，処理時間 (Computational time)，経路 1 km あたりの地図容量 (Map data capacity) を用いる．提案手法のパラメータとしては，4.4.3 項の実験結果に基づいて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，クロック周波数 3.4 GHz の Intel Core i7-6800K CPU と 32GB のメモリを搭載した PC 上で実行した．

Table 4.1: 異なるボクセル解像度での自己位置推定精度. 行見出し (A) はコードブック  $\mathcal{C}$  を作成するための訓練用データの解像度  $r$  を, 列見出し (B) はベクトル量子化を行なう地図データの解像度  $r$  を表す. 要素は全てのコードブックの大きさ  $K$  で測定した RMSE の平均値を表し, 非対角要素の括弧内に同行の対角要素との差分を示す.

(a)  $N = 8$ 

$\begin{smallmatrix} \text{A} \\ \text{B} \end{smallmatrix}$	$r = 1$	$r = 2$	$r = 3$
$r = 1$	0.195	0.234 (0.0386)	0.247 (0.0519)
$r = 2$	0.299 (-0.0172)	0.316	0.336 (0.0200)
$r = 3$	0.588 (0.0363)	0.537 (-0.0139)	0.551

(b)  $N = 27$ 

$\begin{smallmatrix} \text{A} \\ \text{B} \end{smallmatrix}$	$r = 1$	$r = 2$	$r = 3$
$r = 1$	0.244	0.279 (0.0352)	0.286 (0.0419)
$r = 2$	0.483 (-0.0156)	0.499	0.510 (0.0109)
$r = 3$	0.904 (0.0468)	0.871 (0.0147)	0.857

(c)  $N = 64$ 

$\begin{smallmatrix} \text{A} \\ \text{B} \end{smallmatrix}$	$r = 1$	$r = 2$	$r = 3$
$r = 1$	0.331	0.348 (0.0173)	0.372 (0.0412)
$r = 2$	0.605 (-0.0185)	0.623	0.648 (0.0244)
$r = 3$	0.997 (0.0108)	1.001 (0.0142)	0.987

Table 4.2: 自己位置推定性能および地図容量の比較. 進行方向への位置誤差 (Longitudinal error), その横方向への位置誤差 (Lateral error), 進行方向に対する方向誤差 (Orientation error), 処理時間 (Computational time), 経路 1 km あたりの地図容量 (Map data capacity) を示す.

	$r = 1$		$r = 2$		$r = 3$	
	NDT	Prop	NDT	Prop	NDT	Prop
Longitudinal error [m]	0.0771	0.150	0.0995	0.153	0.100	0.158
Lateral error [m]	0.0997	0.153	0.115	0.153	0.120	0.176
Orientation error [°]	0.274	0.278	0.308	0.296	0.308	0.436
Computational time [s]	0.147	0.198	0.0345	0.0589	0.0410	0.0568
Map data capacity [MB/km]	13.0	0.0101	3.32	0.0103	1.50	0.00832

Table 4.2 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なった結果を示す. 全ての  $r$  に対して提案手法は位置誤差を増加させたが, それらは増加後であっても 0.2 m を下回った. 方向誤差は同等か, わずかに悪化する程度であった. 処理時間は提案手法によってやや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, Newton 法に基づく最適化の収束が遅くなったためと思われる. 地図容量に関しては, 標準的な NDT では経路 1 km あたり数 MB から十数 MB の規模であったのに対して, 提案手法では数 kB から十数 kB の規模であり, 標準的な NDT と比べて容量を更に 100 分に 1 から 1,000 分の 1 以下にすることができた.

ここで, 3D NDT 以外の環境地図の圧縮方法とも定性的に比較する. NDT 地図とともに最も代表的な環境地図は正射投影画像地図 [5] である. この地図は道路の路面のみを画像化し, 画素単位の照合によって自己位置を推定する. この手法は 3D NDT 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, Magnusson ら [6] の研究では 1 辺が数 m のボクセルを用いるのに対し, Levinson ら [5] の研究では 1 辺が数 cm のピクセルを用いて環境を表現する. Wei ら [50] は, 正射投影画像地図を圧縮する手法を提案している. この研究では, 地図容量は道路 150 m ごとに

4 MB, すなわち 26.7 MB/km と想定される. Wei 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. この場合の地図容量は  $26.7/500 = 0.0534$  MB/km であり, 本研究の地図容量の 5 倍以上となる. 従って, 元となる環境地図の特性が異なるため公正な比較は困難ではあるものの, 提案手法は精度に関しては Wei らの手法に劣る一方で, 地図容量に関しては優位性を持つと考えられる.

## 4.5 まとめ

本章では, 本論文で取り組む 3 つの課題である (i) DB 内のデータ数の削減, (ii) DB 内の各データの小容量化, (iii) 効率的なデータの照合の内, (ii) に取り組んだ. すなわち, コネクテッドカーによる自己位置推定において, 特徴の変換に基づいて各データを小容量化することによって, 高速かつ高精度な照合と特徴 DB の小容量化を両立した. 従来の手法においては, 3 次元点群を正規分布のパラメータに変換 (集約) することによって環境地図の容量を削減しているが, 大規模な環境地図を想定する場合には容量が大きいという問題があった. この問題に対処するために, 正規分布のパラメータを離散的な近似値に置き換える変換, すなわち量子化を導入することにより, 環境地図の更なる小容量化を実現する手法を提案した.

具体的には, 本章では, 3D NDT によって構築された環境地図の容量を削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持つ占有/非占有の状態およ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を圧縮する. 公開データセットを用いた評価実験により, 提案手法が標準的な 3D NDT と比べて同程度の推定精度と処理時間を達成しつつ, DB を小容量化することを示した. ただし, 3D NDT は反復的な最適化に基づくため, 解の収束が遅い場合には処理時間が長くなるという問題がある. 次章ではこの問題に対処しつつ, 高精度な特徴照合と特徴 DB の小容量化を両立する手法を検討する.



## 第 5 章

# 類似尺度の変更に基づく自己位置推定

本論文ではスマートフォンによる画像検索とコネクテッドカーによる自己位置推定というアプリケーションにおいて、高速かつ高精度な照合と特徴 DB の小容量化を両立することを目指す。この目的を達成するために、本論文では (a) 特徴の選択, (b) 特徴の変換, (c) 類似尺度の変更, の観点に基づいて, (i) DB 内のデータ数の削減, (ii) DB 内の各データの小容量化, (iii) 効率的なデータの照合, の課題に取り組む。この内, 第 4 章ではコネクテッドカーによる自己位置推定に着目し, (b) 特徴の変換に基づいて (ii) DB 内の各データの小容量化を実現することにより, 3D NDT [6] によって構築される環境地図を小容量化した。提案手法による地図は 3D NDT による地図と互換性があり, 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, 提案した地図データの圧縮・復元機能は, 3D NDT 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 による自己位置推定は幅広く利用されており, 地図 DB や関連するソフトウェアも多数存在するため, その意義は大きいと考えられる。しかし, 3D NDT による自己位置推定は反復的な最適化に基づくため, 解の収束が遅い場合には処理時間が長くなる。そこで本章では, (c) 類似尺度の変更に基づいて (iii) 効率的なデータの照合を実現することにより, 処理時間を安定的に短くしつつ, 高精度な照合と特徴 DB の小容量化を両立する自己位置推定手法を提案する。本手法では, 第 4 章と同様に (b) 特徴の変換に基づいて環境地図を小容量化し, その地図に適した類似尺度を用いて効率的に特徴の照合を行なう。

具体的には, 3 次元点群を一部の空間に対応するベクトルデータの集合として表現する

地図生成手法および自己位置推定手法を提案する．この手法は，ボクセルを用いて 3 次元点群をモデル化した後に，ボクセルデータの集合を 2 値ベクトルとして表現する．そして，ベクトル量子化によって圧縮されたデータを効率的に格納することによって地図容量を削減する．自己位置推定を行なう際には，2 値ベクトル間の類似度に基づいて 2 つのデータを照合する．ここでは，2 つのデータ間の類似尺度として 2 値ベクトル間の Hamming 距離の総和を利用する．評価実験では，従来手法である 3D NDT に基づく地図と比べて小容量の地図データを生成可能であることを示す．また，その地図データを用いて高精度な自己位置推定を実時間に実現可能であることを実証する．

本研究の貢献は，以下のように要約される．

1. 3 次元点群として表現される環境地図に対して，自己位置推定の精度を維持しつつ，容量を削減するための地図表現を提案する．
2. 安定的に高速な自己位置推定を実現するために，類似尺度の変更とそれに基づく特徴照合手法を提案する．
3. 車両に搭載された LiDAR で計測した 3 次元点群を用いた実験を行ない，提案手法が高速かつ高精度な自己位置推定と地図データの小容量化を両立できることを示す．

以下，5.1 節で 3D LiDAR を用いたスキャン照合について整理する．その後，5.2 節および 5.3 節で新たな地図生成手法および自己位置推定手法を提案する．そして，5.4 節で地図容量および自己位置推定性能に関して評価し，5.5 節で本章をまとめる．

## 5.1 はじめに

第 2 章で述べた通り，車両の自己位置推定手法として 3D LiDAR を用いたスキャン照合が注目を集めている．本節では，既存のスキャン照合手法について簡潔にまとめた後に，本章で提案する手法の概要を述べる．

最も一般的なスキャン照合手法は，ICP (Iterative Closest Point) [54] とその変形 [125,126] である．これらの手法では，3 次元点群そのものを地図データとみなす．スキャン照合の実行時には，地図データとスキャンデータ（3 次元点群）の間で最近傍点の対応付けと位置姿勢変換を交互に反復する．そして，全ての対応点間の距離の総和を最小化す



る位置と姿勢を求めることにより、高精度な自己位置推定を実現する。一方、これらの手法は地図の容量が非常に大きい。近年の LiDAR センサは 1 秒間に数十万点以上の 3 次元点を計測するため、構築される 3 次元点群の容量は経路 1 km あたり数百 MB から数 GB の規模になり得る。

点群に対するデータ圧縮は、地図容量削減という課題に対する解決策の 1 つである。Sim ら [179] は QSplat 法 [180] によって表現された点群を対象に、個々の点データに対して線形量子化を適用することでデータ量を削減する手法を提案した。更に、Sim ら [181] は、線形量子化の代わりにベクトル量子化を利用することによって、点群データをより効率的に圧縮する手法も提案した。ベクトル量子化とは、あるベクトルの集合を事前に用意された代表ベクトルの集合（コードブック）を用いて近似するデータ圧縮手法である。Golla ら [182] は 8 分木ボックスを用いて 3 次元空間を再帰的に分割し、画素値を量子化した高さ地図画像および占有地図画像で各ボックスに含まれる点群を表現することによって点群データを圧縮する手法を提案した。これらの手法は点群の詳細を保持しつつデータ量を削減することを目指している。すなわち、点群の局所的な密度や表面形状の曲率などの諸量が損なわれないようにデータを圧縮する。そのため、スキャン照合の精度を維持しつつ地図容量を削減するという本研究の目的に対して必ずしも最適な方法とは言えず、十分な削減率が得られない可能性がある。

3D NDT [6] は、ボックスごとの 3 次元点群を 3 変量正規分布モデルで表現することにより、元の 3 次元点群と比べて地図容量を大きく削減することができる。また、最近傍点の対応付けを行わず、スキャンデータをボックスへ割り当てることによって点と正規分布モデルを対応付けるため、ICP と比べて計算量が小さいという利点もある。このような近似表現を用いた地図の容量は、経路 1 km あたり数 MB から数十 MB の規模になる。国規模の地図を想定する場合、道路の総延長は数十万 km から数百万 km に及ぶため、単純に換算した場合の地図容量は数 TB から数百 TB の規模となる。車両に搭載された記憶装置に格納することを考えると、この容量は小さいとはいえない。

第 4 章では、3D NDT による環境地図の容量を削減した。しかし、3D NDT による自己位置推定は反復的な最適化に基づくため、解の収束が遅い場合には処理時間が長くなる。自己位置推定ではセンサデータの入力に対して逐次的に処理を行なう必要があるため、処理時間は安定的に短い方が望ましい。また、走行中の車線と自車を対応付けるためには、誤差 1 m 未満の位置精度が必要である [177]。従って、これらの要件を満たしつつ、

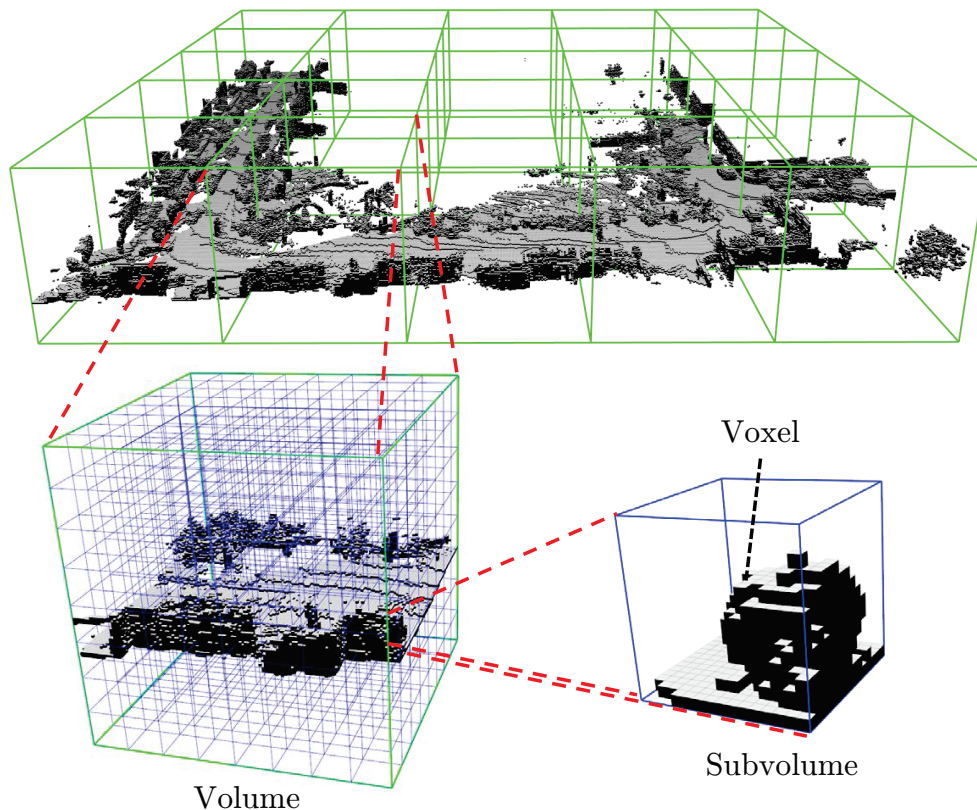
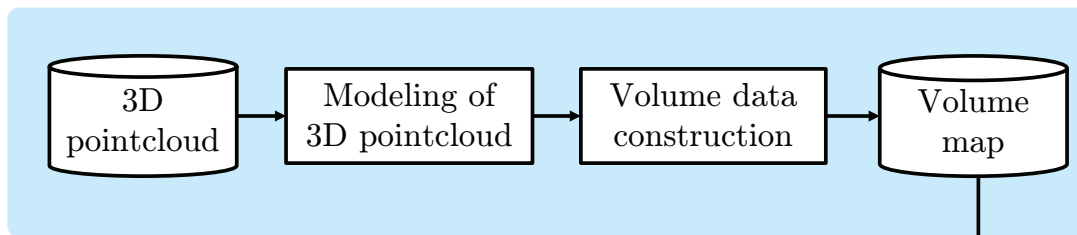


Fig. 5.1: ボリューム地図の概要．緑色，青色の立方体はそれぞれボリューム，サブボリュームを表す．灰色の立方体は占有ボクセルを表し，非占有ボクセルは図示していない．

環境地図の容量を削減する手法が必要である．

そこで本章では，LiDAR データからの新たな地図生成手法，および高速なスキャン照合に基づく自己位置推定手法を提案する．提案手法は，車両の自動走行に要求される精度と計算効率性を実現しつつ，可能な限り環境地図の容量を削減することを目的とする．この内，地図生成手法は，ボクセルを用いて 3 次元点群を表現した後に，それらのデータを量子化することによって地図データを圧縮する．また，更に容量を削減するために，Fig. 5.1 に示すようなボリューム地図を構築する．この図では緑色，青色の立方体はそれぞれボリューム，サブボリュームを表す．灰色の立方体は占有ボクセルを表し，非占有ボクセルは図示しない．一方，自己位置推定手法は，同一体積の空間に対応するデータの類似性に基づいてスキャン照合を行なう．ここでは，地図データが含む量子化インデックスを復号することにより，各ボクセル内のモデルを復元する．提案手法は，標本化されたパ

## Map generation (offline)



## Localization (online)

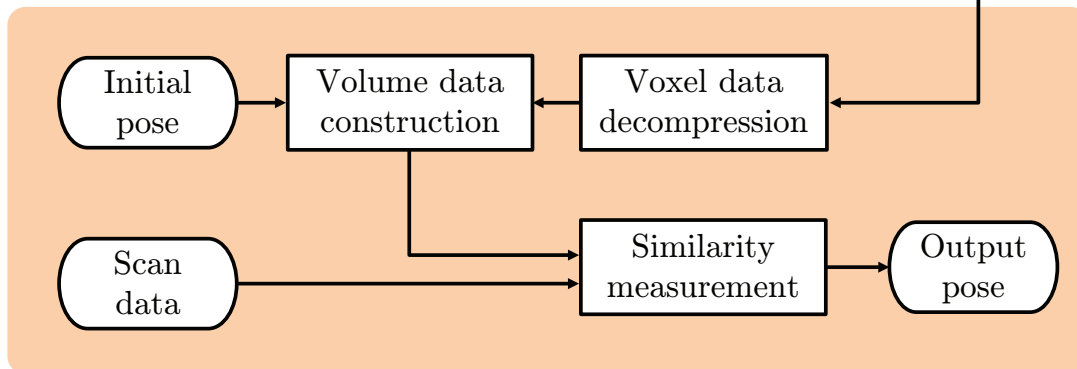


Fig. 5.2: 提案する地図生成手法および自己位置推定手法の枠組み.

ラメータの範囲を探索することにより，処理時間を安定させることができる．この探索では，地図データとスキャンデータをボクセル単位で照合することにより，10 cm 単位の精度で自己位置推定を実現する．また，高速な照合のために，有限の空間を表現するベクトル間の類似度計算，および粗密探索法に基づく位置姿勢パラメータの探索を行なう．

## 5.2 ボリューム地図の生成

自動走行に要求される精度と計算効率性を達成しつつ，地図容量を削減するための新たな地図生成手法を提案する．Fig. 5.2 の上段に提案する地図生成手法の概要を示す．提案手法は，SLAM 技術や位置姿勢計測センサの利用によって生成された 3 次元点群地図が入力として与えられることを前提とする．

提案手法は，初めに空間をボクセルで分割し，各ボクセル内のデータをモデル化する．その後，指定した大きさのボリューム内のボクセルの集合をベクトル表現に変換する．次に，地図容量を削減するために，予め作成したコードブックを用いてこのベクトルを代表ベクトルに置き換える量子化処理を行なう．この際，大きさが小さいコードブックを用い

て効率的にベクトルを表現するために、サブボリュームごとにベクトルの計算と量子化を行なう。更に、各サブボリュームの絶対位置をボリューム内の相対位置で表現することによって、位置情報に関するデータ量を削減する。以上の処理は事前にオフラインで実行しておく。以下各々の処理について詳細を述べる。

### 5.2.1 3次元点群のモデル化

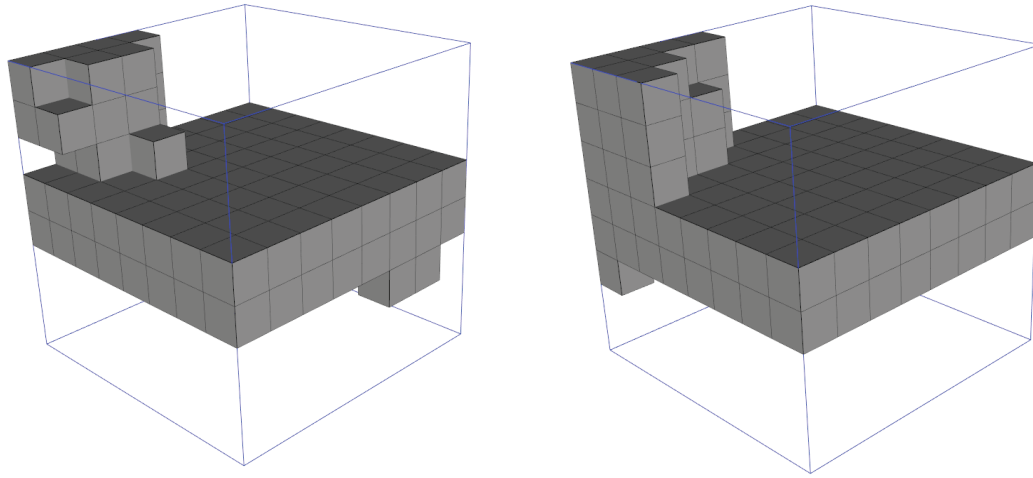
3次元点群が与えられた場合、初めに1辺の長さが $r$ のボクセルを用いて空間を分割する。そして、ボクセルの占有／非占有を2値で表現する2値占有モデルにより所与の3次元点群をモデル化する。

得られたボクセルデータは、有限個のボクセルで構成されるボリュームデータとしてモデル化する。この際、後述のベクトル表現における次元数を削減するとともに、非占有ボクセルのみで構成される一部の空間を効率的に格納するために、ボリュームデータの部分集合であるサブボリュームを用いて階層的にモデル化する。すなわち、初めに $x$ ,  $y$ ,  $z$ 方向にそれぞれ $m$ 個、合計 $M = m^3$ 個のボクセルを1つのサブボリュームとしてモデル化する。その後、同様に一方向に $n$ 個、合計 $N = n^3$ 個のサブボリュームから1つのボリュームを形成する。そして、サブボリューム内のボクセルの集合を1つのベクトルとして表現する。すなわち、サブボリューム内で $(x, y, z)$ 方向に0から数えて $(u, v, w)$ 番目に位置するボクセルに対して、次式のインデックスを与える。

$$i = u + vm + wm^2 + 1. \quad (5.1)$$

そして、ボクセルが持つ占有／非占有を表す2値を $i$ 番目の次元の要素としたベクトルによりサブボリュームを表現する。その後、地図容量を削減するために、各サブボリュームのベクトルをベクトル量子化する。コードブックは、訓練用の3次元点群から計算されたサブボクセルを表現するベクトル群を $K$ 個にクラスタリングし、各クラスターの代表を求めることで作成する。そして、各サブボリュームを、それが割り当てられた代表ベクトルのインデックス $k \in \{1, \dots, K\}$ で表現する。全てのボクセルが非占有であるサブボリュームに対してはこの処理を省略し、N/A (Not Available) と表現する。

Fig. 5.1 に、提案手法によって生成するボリューム地図の概要を示す。提案手法においては、最終的に得られたボリュームデータの集合を地図データとする。Fig. 5.3(a) はある入力ベクトルで表現されたサブボリュームの例を、Fig. 5.3(b) はコードブックを用いて



(a) 入力ベクトルに対応するサブボリューム

(b) 代表ベクトルに対応するサブボリューム

Fig. 5.3: ベクトル量子化の例. (b) はコードブックを用いて入力ベクトルを代表ベクトルに置き換えたサブボリュームを模式的に表現している.

代表ベクトルへ置き換えられたサブボリュームを模式的に表現している. 代表ベクトルは量子化誤差に相当する形状的な誤差を含むものの, 入力ベクトルに近似した形状を表現することができる.

### 5.2.2 ボリュームデータの構築

各ボリュームデータは, 次の 3 通りの情報で構成される.

1. ボリュームの中心座標
2. N/A ではないサブボリュームの相対位置を表すビット列  $B_P$
3. N/A ではないサブボリュームを表現する代表ベクトルのインデックスで構成されたビット列  $B_I$

地図容量を削減するために, 各サブボリュームの位置情報を絶対位置として表現する代わりに, ボリューム内での相対位置のインデックスとして表現する. 初めに, ボリューム内で  $(x, y, z)$  方向に  $(u, v, w)$  番目に位置するサブボリュームに対して, 式 (5.1) と同様の様式でインデックス  $j$  を与える. そして,  $j$  番目の要素に対して, サブボリュームが 1 つ以上の占有ボクセルを含めば '1', そうでなければ '0' が割り当てられるビット列  $B_P$  を生

成する.

サブボリュームを表現する代表ベクトルのインデックス  $k$  は, コードブックの大きさ  $K$  を表現可能な最短長, すなわち長さ  $\text{ceil}(\log_2(K))$  のビット列  $b_j$  (量子化インデックス) に変換される. ここで,  $\text{ceil}$  は切り上げ関数である. その後, 対応する  $j$  の小さい順に  $b_j$  を並べたビット列  $B_I$  を生成する. このビット列には,  $N/A$  のサブボリュームに対応する  $b_j$  を含まない.

### 5.3 自己位置推定

地図データとスキャンデータの間の類似度に基づく自己位置推定手法を提案する. Fig. 5.2 の下段に示すように, 提案手法への入力は地図側の座標系における位置と姿勢を表す回転行列  $\mathbf{R}$  および並進ベクトル  $\mathbf{t}$  の初期値  $\Xi_0 = [\mathbf{R}|\mathbf{t}]$ , 5.2 節で用意したボリューム地図, LiDAR センサによって計測されたスキャンデータである.

提案手法は Wolcott ら [8] と同様に標本抽出に基づく探索によって, 初期値の誤差を補正する位置姿勢変換パラメータを求める. 前述の ICP や 3D NDT は地図データとスキャンデータの対応付けと位置姿勢変換パラメータの最適化を交互に反復するため, 解の収束が遅い場合には処理時間が長くなる. それに対し, 標本抽出に基づく手法は予め設定した離散的な位置姿勢変換パラメータを探索することによって, 非反復的に解を求めることができる. 従って, 提案手法は ICP や 3D NDT に比べて得られる結果が離散的ではあるものの, 処理時間を安定化させることができるという利点がある.

具体的には, 初めにコードブックを用いて量子化インデックス  $b_j$  を復号することにより, ボリューム地図からボクセルデータを復元する. その後, 復元されたデータから初期値に基づく自己位置を中心としたボリュームデータを生成する. この処理は, 復元されたボクセルデータに対して式 (5.1) のインデックスを再計算するだけで実現される. また, 様々な位置姿勢変換を施されたセンサ側の 3 次元点群からボリュームデータを生成する. そして, 地図側のボリュームデータと最も類似するセンサ側のボリュームデータを検索し, それに対応する位置姿勢変換パラメータを求める.

なお, 提案手法によって構築される地図データは 3 次元点群地図に基づく既存の自己位置推定手法でも利用可能である. しかし, 本研究では前述の計算効率性を実現するために, 新たな自己位置推定手法を提案する.

### 5.3.1 定式化

自己位置推定問題を，地図側とセンサ側における同一体積の空間を表現するベクトル間の類似検索問題として定式化する．初期値の位置成分  $\mathbf{t}$  を地図側のボリウムデータの中心座標  $\mathcal{C}$  とする．地図側の 3 次元点群  $\mathcal{P}$  から生成され，パラメータ  $\Theta = \{\mathcal{C}, r, m, n\}$  を用いて構築されたボリウムデータを次式で表す．

$$V(\mathcal{P}, \Theta) = \{\mathbf{p}_j | j = 1, \dots, N\}, \quad (5.2)$$

ここで  $\mathbf{p}_j = \{p_{j,i} | i = 1, \dots, M\}$  はボリウム内の  $j$  番目のサブボリウムを表現するベクトルである． $N/A$  のサブボリウムに対しては， $\mathbf{p}_j$  を全ての要素が 0 のベクトルとみなす．センサ座標系の原点を，センサ側のボリウムデータの中心座標  $\mathcal{C}'$  とする．初期値が誤差を含まない場合， $\mathcal{C}$  と  $\mathcal{C}'$  は実世界の同一位置を表す．しかし，一般に初期値は誤差を含むため，必ずしもそうなるとは限らない．初期値の誤差を補正するために，並進と回転を表す位置姿勢変換パラメータ  $\xi = (t_x, t_y, t_z, \theta_x, \theta_y, \theta_z)^T$  を用いてセンサ側の 3 次元点群  $\mathcal{P}'$  に様々な位置姿勢変換を施す．その後，5.2.1 項で述べた方法でパラメータ  $\Theta' = \{\mathcal{C}', r, m, n\}$  を用いてボリウムデータを生成する．ただし，ここでは容量を削減する必要がないため，サブボリウムを表すベクトルは代表ベクトルには置き換えない．

$$V'(T(\mathcal{P}', \xi), \Theta') = \{\mathbf{q}_j | j = 1, \dots, N\}. \quad (5.3)$$

ここで， $T$  は位置姿勢変換関数である．また，式 (5.2) と式 (5.3) で表されるボリウムデータが含むボクセルやサブボリウムの大きさを一致させるために， $\Theta'$  内の  $r, m, n$  は，常に  $\Theta$  内のものと同一になるように設定する．

提案手法は，ボリウムデータが含むボクセルの内，ボリウム内での相対位置が一致し，かつ同一の表現を持つボクセルの対に対して，地図データとスキャンデータの類似性を表す類似度を与える．効率的な計算のために，サブボリウムを表す 2 値ベクトル間の Hamming 距離を非類似度尺度として利用する．従って，初期値の誤差を最も良く補正する位置姿勢変換パラメータ  $\xi$  を求める問題を，次式で表される非類似度関数の最小化問題として定式化する．

$$\arg \min_{\xi} d(\xi) = \sum_{j=1}^N \sum_{i=1}^M |p_{j,i} - q_{j,i}|_{\oplus}, \quad (5.4)$$

ここで， $|p_{j,i} - q_{j,i}|_{\oplus}$  は 2 つの 2 値表現の間の Hamming 距離である．

### 5.3.2 ボリュームデータ間の類似度計算

初期値の誤差を補正するために、予め指定した範囲から位置姿勢変換パラメータを標本抽出する。初めに、様々な位置姿勢変換が施されたセンサ側の 3 次元点群からボリュームデータを生成し、地図側のボリュームデータと最も類似するものを求める。そして、そのボリュームデータに対応する位置姿勢変換パラメータを用いて初期値を補正し、自己位置を推定する。

以下、5.3.2.1 および 5.3.2.2 でセンサ側の 3 次元点群に回転変換および並進変換を施す方法を説明する。その後、5.3.2.3 で自己位置を推定する方法を述べる。

#### 5.3.2.1 回転変換

初めに、回転に関してのみセンサ側の座標系を地図側の座標系に一致させるために、初期値  $\mathbf{R}$  を用いてスキャンデータ  $P'$  を回転変換する。その後、初期値の誤差を補正するために、 $\mathbf{R}$  を基準とした回転オフセット  $o_\theta$  を用いて  $P'$  を更に変換する。回転オフセット  $o_\theta$  は、回転パラメータに関する標本抽出の範囲  $[-\theta, \theta]$  内で等間隔に位置する指定個数の回転角度である。以降の処理は、異なるオフセットで変換されたセンサ側の 3 次元点群ごとに独立に行なう。

#### 5.3.2.2 平行移動による並進変換

初期値の並進誤差を補正するために、並進オフセット  $o_t$  を用いて並進変換された 3 次元点群からボリュームデータを生成する。ただし、並進変換を効率的に計算するために、実際に 3 次元点群を並進変換する代わりに、求めるサブボリュームおよびボリュームと同じ大きさの 3 次元ウィンドウを平行移動させることによってボリュームデータを計算する。

具体的には、初めに 5.2.1 項と同様にしてボクセル群を構築する。その後、サブボリュームと同じ大きさの 3 次元ウィンドウをボクセル単位で各方向に平行移動させることによってサブボリューム群を構築する。すなわち、サブボリュームを構築する際には、各サブボリューム内でのボクセルの相対位置を表すインデックス  $i$  のみを計算する。従って、同一のボクセルであっても、異なるサブボリュームにおいては異なる  $i$  が割り当てられる。同様にして、ボリュームと同じ大きさの 3 次元ウィンドウをサブボリューム単位で平行移動



させることによってボリューム群を構築する．並進オフセット  $o_t$  は，各サブボリュームおよび各ボリュームを構築する際の平行移動回数の組み合わせから計算できる．

### 5.3.2.3 初期値の補正

センサ側の 3 次元点群から生成された全てのボリュームデータの中から，式 (5.4) の非類似度を最小化するものを求める．その後，得られたボリュームデータに対応する位置姿勢変換パラメータ  $\hat{\xi}$  は，上述のオフセット  $o_\theta$  および  $o_t$  から計算できる．最後に， $\hat{\xi}$  を用いて初期値を補正することによって，自己の位置と姿勢を得る．

### 5.3.3 探索の効率化

効率的な探索のために，粗密戦略を採用する．すなわち，徐々に細くなるオフセットを用いて段階的に位置姿勢変換パラメータを求める．この処理は 1 つの地図データしか利用せず，追加のデータを必要としない．

最初の段階においては，初期値  $\Xi_0$  と最も粗いオフセット  $o_{\theta,1}$  および  $o_{t,1}$  を用いて，指定された範囲を網羅的に探索する．結果として，推定された位置姿勢  $\Xi_1$  を得る．次の段階においては，より細かいオフセット  $o_{\theta,2}$  および  $o_{t,2}$  を用いて，より狭い範囲のみを探索する．ここでは， $\Xi_1$  を初期値とみなす．結果として，推定された位置姿勢  $\Xi_2$  を得る．これ以降の段階においても，同様にオフセットを細かくしながらより狭い範囲を探索する．

## 5.4 評価実験

本節では，提案した地図生成手法および自己位置推定手法の性能を評価する．提案手法は従来手法と比べて遥かに容量が小さい地図を用いて，車両の自己位置推定に関する要件を満たす性能を実現することを示す．

### 5.4.1 実験設定

提案手法の性能を評価するために，第 4 章と同様に KITTI データセット [135, 136] を利用する．このデータセットは，LiDAR センサや GPS, IMU を搭載した車両を用いて市街地で計測したデータを含む．実験では，初めに GPS および IMU データが含む位置と姿勢を用いて複数のスキャンデータを統合することにより 3 次元点群地図を構築した．そ

の後，提案した地図生成手法を用いて地図データを作成した．本実験において，単位長当たりの地図容量はこの 3 次元点群を構築する際に使用した GPS および IMU データの軌跡の総延長で地図容量を除算することによって求める．コードブックは，Málaga Urban データセット [137] から訓練用の 3 次元点群を構築し， $k$ -medoids 法を用いて作成した．

提案手法では，求める位置姿勢の並進成分はボクセル単位での探索によって決定する．そのため，ボクセルの 1 辺の長さ  $r$  が長いほど並進成分の誤差が大きくなる一方で計算量が削減される．車線単位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 [177] であるため，本実験では許容可能な並進成分の誤差を 1 m までと想定する．そして，それより更に小さい並進誤差で計算量を削減するために， $r = 25$  cm に設定した．また，提案手法ではボリュームの 1 辺の長さが長いほど，5.2.2 項で定義したビット列  $B_P$  のデータ量が大きくなる．一方，長さが短いほどボリューム内での N/A ではないサブボリュームの割合が平均的に高くなり，結果として全データに対するビット列  $B_I$  のデータ量の割合が大きくなる．本研究では地図の容量を最小にする長さを調査し，ボリュームの大きさを  $24\text{ m} \times 24\text{ m} \times 24\text{ m}$  に設定した．提案手法による自己位置推定では，実行時に構築するボリュームが大きいくほど利用可能な地図データおよびスキャンデータの範囲が広がる一方で処理時間が増加する．処理時間の増加を抑えつつ広範囲のデータを利用するために，地図側とセンサ側の両方でボリュームの大きさを  $96\text{ m} \times 96\text{ m} \times 24\text{ m}$  に設定した．すなわち， $x$  方向と  $y$  方向のそれぞれに  $24\text{ m} \times 24\text{ m} \times 24\text{ m}$  のボリュームの 4 倍の大きさを設定した．ただし， $z$  方向に対しては，多くの場合に地図上に 1 つのボリュームデータしか存在しないため，等倍に設定した．

提案手法における位置姿勢変換パラメータ  $\xi$  の探索範囲については，LiDAR センサの計測を入力として実時間処理を行なう前提で設定した．すなわち，KITTI データセットで使用された LiDAR センサの反復率である 10 Hz の周期で逐次的に自己位置推定を行なう場合に，初期値の誤差がとり得る範囲を含むように探索範囲を設定した．また，本実験では Wolcott ら [8] と同様に，車両に対するロール角およびピッチ角は微小な値であると想定し，探索範囲に含めないものとした．具体的には， $x$  方向および  $y$  方向にそれぞれ  $[-3\text{ m}, 3\text{ m}]$ ， $z$  方向に  $[-0.5\text{ m}, 0.5\text{ m}]$ ，そしてヨー角に対して  $[-5^\circ, 5^\circ]$  を探索範囲とした．また，本実験では粗密戦略に基づき 2 段階の探索を行なった．1 段階目は上記の範囲に対して  $x$  方向， $y$  方向， $z$  方向にそれぞれ 0.5 m 刻みで探索する  $o_{t,1}$  を，ヨー角には  $1^\circ$  刻みで探索する  $o_{\theta,1}$  を設定した．2 段階目では  $x$  方向および  $y$  方向にそれぞれ

$[-1 \text{ m}, 1 \text{ m}]$ ,  $z$  方向に  $[-0.25 \text{ m}, 0.25 \text{ m}]$  の範囲を  $0.25 \text{ m}$  刻みで探索する  $o_{t,2}$  を, ヨー角には  $[-1^\circ, 1^\circ]$  の範囲を  $0.5^\circ$  刻みで探索する  $o_{\theta,2}$  を設定した. 2段階目のオフセットには, 1段階目で誤った場合にも正しく推定できるようにするために, 冗長性を持たせる設定とした.

全ての実験は, クロック周波数  $3.4 \text{ GHz}$  の Intel Core i7-6800K CPU と  $32\text{GB}$  のメモリを搭載した PC 上で実行した. 提案手法では, 位置姿勢変換パラメータごとに独立して類似度計算を実行することができる. そこで, 高速化のために OpenMP ライブラリ\*による CPU 上での並列処理技術を導入し,  $6$  コア  $12$  スレッドを用いて並列処理するように実装した.

#### 5.4.2 サブボリュームおよびベクトル量子化の影響

提案手法で使用されるパラメータであるサブボリュームの 1 辺の長さ  $L (= r \times m)$  およびコードブックの大きさ  $K$  について, 性能への依存性を検証する. 初めに  $100$  通りのスキャンデータが無作為に選択し, 真値から離れるような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なう. ここでは, 初期値の誤差を模擬する目的で 5.4.1 項で記した探索範囲内での無作為な変換を施した. 各スキャンデータに対してそれぞれ  $100$  回の試行を施し, RMSE を計算した. 更に, 参考のために 3 次元点群に対する PSNR (Peak Signal-to-Noise Ratio) [182] を用いてベクトル量子化の前後における地図データの劣化度合を定量的に示す. この PSNR は, 地図データを占有ボクセルの中心位置を座標とする 3 次元点群とみなして計算する. また, 経路  $1 \text{ km}$  当たりの地図容量および処理時間も計測した.

検証するパラメータとしては, サブボリュームの 1 辺の長さ  $L \in \{1 \text{ m}, 2 \text{ m}, 3 \text{ m}\}$  およびコードブックの大きさ  $K \in \{2^1, \dots, 2^{20}\}$  を用いた. 各  $L$  に対して, ボリュームの大きさが同一になるようにサブボリューム数を設定した. また,  $L$  は, 地図生成時と自己位置推時に共通の値を用いた. 更に, ベクトル量子化の有無が性能へ及ぼす影響を検証するために, 地図生成においてベクトル量子化を行わない場合の RMSE も合わせて測定した. 以上全ての結果を Fig. 5.4 に示す.

---

\* <https://www.openmp.org/>

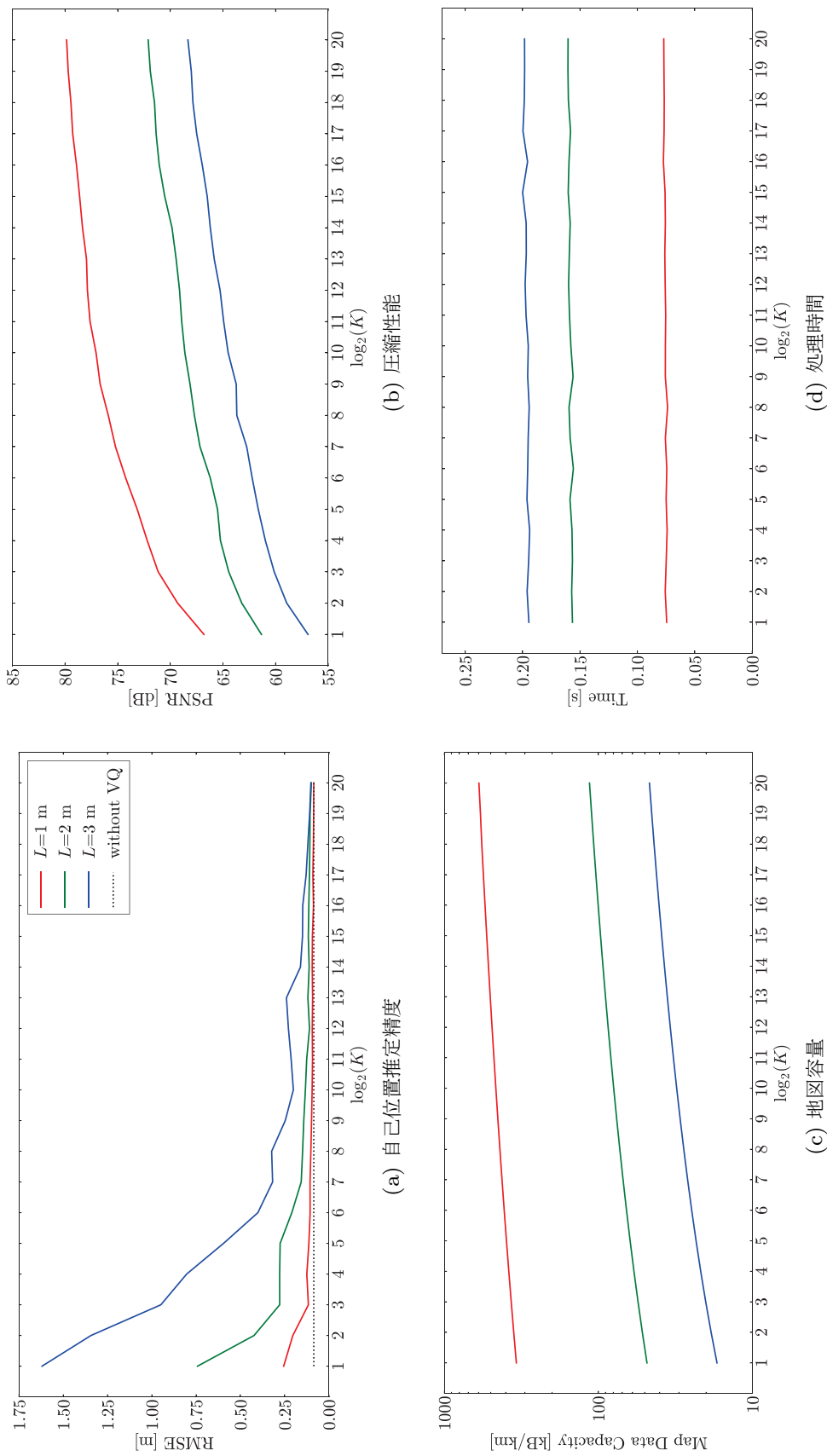


Fig. 5.4: 異なるパラメータを用いた際の提案手法の性能評価.

Fig. 5.4(a) は異なる  $L$  および  $K$  に対する自己位置推定精度を示す。このグラフより、 $K$  が等しい場合は  $L$  が小さいほど自己位置推定の精度が向上する傾向があることがわかる。また、同一の  $L$  を用いた場合は  $K$  が大きいほど自己位置推定の精度が改善する。これらは、サブボリュームの体積に対して  $K$  が大きくなるほど、量子化誤差が減少するためである。各  $L$  において、ベクトル量子化を行わない場合の精度と比較した場合、 $L=1, 2, 3$  のそれぞれに対して  $K = 2^6, 2^{12}, 2^{20}$  の時点ではほぼ同等の精度を達成した。すなわち、大きさがこれらの値以上のコードブックを用いた場合には、自己位置推定の精度への悪影響を抑えつつ地図容量を削減できる。

Fig. 5.4(b) にベクトル量子化の前後の PSNR を示す。このグラフでは、縦軸の値が大きいほど地図データの損失が少ないことを表す。Fig. 5.4(a) と合わせて見ることで、自己位置推定の精度と地図データの損失は対応する傾向があることが分かる。すなわち、 $K$  が等しい場合は  $L$  が小さいほど PSNR が大きく、各  $L$  において  $K$  の増加が PSNR を改善する。これは、前述した理由によって量子化誤差が減少するためである。従って、各  $L$  において  $K$  の増加が地図データの損失を軽減させ、それによって自己位置推定の精度が改善したと考えられる。

Fig. 5.4(c) は異なる  $L$  および  $K$  に対する地図容量を示す。このグラフより、地図容量が  $L$  に大きく依存することがわかる。これは、同一の体積の空間に含まれるサブボリュームの個数が  $L$  の 3 乗に反比例するためである。一方、 $K$  への依存性は小さい。これは、サブボリュームを表現するベクトル量子化インデックスの大きさが  $\log_2(K)$  であり、 $K$  の増加量に対してインデックスの大きさの増加が緩やかであるためである。従って、地図容量を削減するためには  $L$  を大きくすることが効果的であることがわかる。ただし、これはサブボリュームを表現するベクトルの次元を増加させることになるため、いわゆる次元の呪いに起因して適切なコードブックの大きさ  $K$  を指数関数的に増加させる。それにより、いずれは  $K$  の増加に伴う地図データの増加量が、 $L$  の増加に伴う削減量を上回ることが予想される。

ここで、3 次元データの圧縮に広く利用される 8 分木を用いてサブボリュームを表現する場合との比較を検討する。例えば  $r = 25$  cm,  $L = 2$  m の場合、提案手法は 1 つのサブボリュームを最小 12 bit で表現できる。一方、8 分木を用いた場合はこのサブボリュームを 3 階層の 8 分木構造で表現する。その際のデータ量の最小値は、各階層で 1 つのノードのみが子ノードを持つ場合の  $8 + 8 + 8 = 24$  bit、最大値は全てのノードが子ノードを

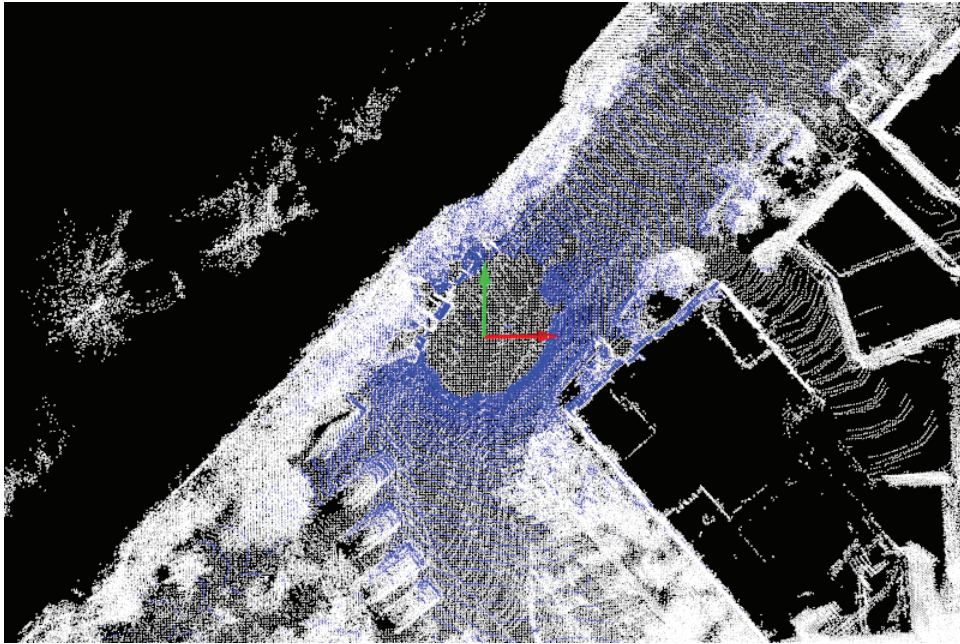
持つ場合の  $8 + 8^2 + 8^3 = 584$  bit となる．従って，常に提案手法の方がデータ量が小さくなる．ただし，提案手法は非可逆圧縮手法であるのに対し，2 値占有モデルへの 8 分木表現は可逆圧縮手法であるため，復元される地図データには劣化が生じないという利点がある．

Fig. 5.4(d) に，異なる  $L$  および  $K$  に対する平均処理時間を示す．このグラフより， $L$  が大きくなるほど処理時間が増加する傾向があることがわかる．これは， $L$  が大きくなるほど N/A であるサブボリュームが減少していくことに起因する．本実験の実装では，地図側とセンサ側のサブボリュームが両方とも N/A であった場合，それらのサブボリュームを表現するベクトル間の Hamming 距離を 0 とみなして計算を省略する．ここでは，異なる  $L$  に対してボリュームの大きさを共通にしているため，入力点群が同一であれば  $L$  が大きいほどサブボリュームが N/A となる割合が減少する．従って， $L$  が大きいほどビット演算の回数が増加し，処理時間が長くなる．ただし， $L$  が小さいほどボリューム内のサブボリューム数が増えるため，N/A であるか否かの判定に要する処理時間が増加していく．そのため，必ずしも  $L$  が小さいほど高速化できるとは限らない．一方，処理時間は  $K$  には影響を受けないことがわかる．これは，ビット演算を行なう回数が  $K$  に依存しないためである．

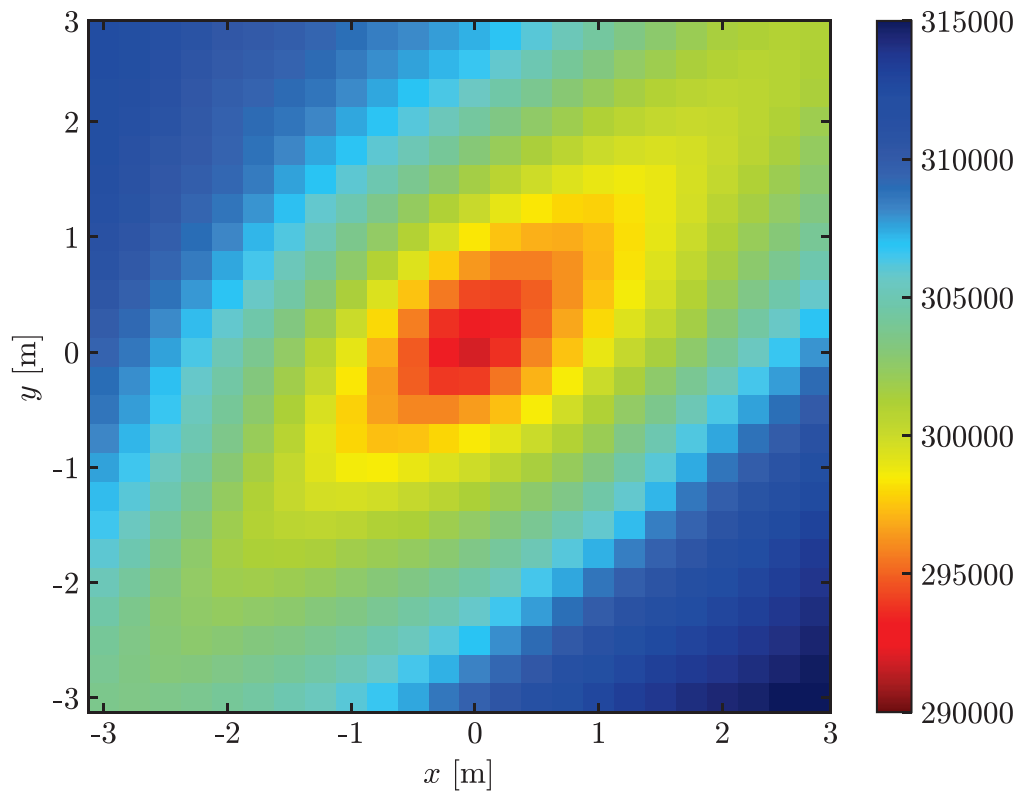
高精度な自己位置推定を実現しつつ地図容量を削減するために，地図生成時に利用するパラメータとして  $L = 3$ ， $K = 2^{20}$  を選択する．これらのパラメータに対応するコードブックの容量は 226.5 MB であり，自己位置推定時に利用するために地図データとともに追加的に格納される．Fig. 5.5 にこれらのパラメータを用いて生成した地図データとスキャンデータ間の非類似度関数  $d(\xi)$  の値の例を示す．Fig. 5.5(b) を見ると，量子化された地図データであっても真値において非類似度が最小となるため，高精度な自己位置推定が可能であることを確認できる．また，右斜め方向への変位に対しては，地図データの形状的な変化が少ないために非類似度の変化が緩やかであることがわかる．

提案手法では，地図生成時に量子化したボクセルデータを自己位置推定時に復元する．その後，復元されたボクセルデータから再度サブボリュームを構築する．そのため，地図生成時と自己位置推定時に必ずしも共通の  $L$  を利用する必要はない．従って高速化のために，自己位置推定時に利用するパラメータには  $L = 1$  を選択した．

以降の実験では，これらのパラメータを使用する．



(a) スキャンデータの例。白い点と青い点でそれぞれ地図データとスキャンデータを示す。



(b) ヒートマップ。スキャンデータの各軸方向に与えた変位に対応する非類似度を示す。

Fig. 5.5: 非類似度関数  $d(\xi)$  の可視化.

### 5.4.3 自己位置推定性能および地図容量の評価

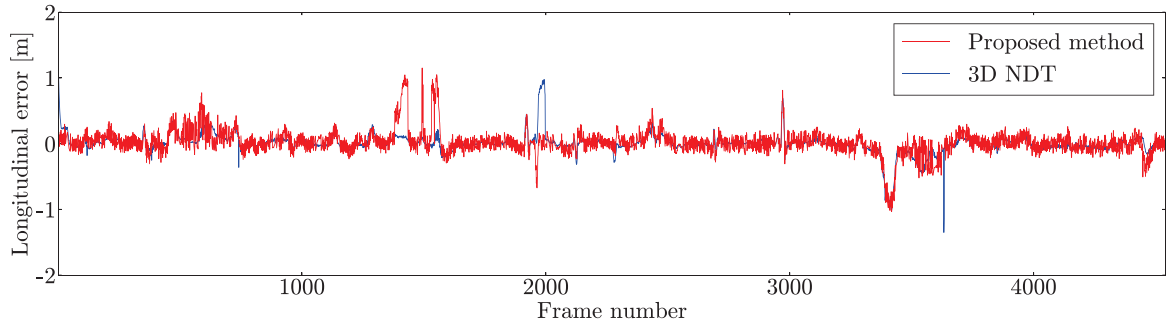
連続的に計測されたスキャンデータを用いて、提案した自己位置推定手法の性能を検証する。また、自己位置推定に利用した地図の容量も測定する。この実験では、KITTI データセットに含まれる市街地内の 3.7 km の経路にわたって計測されたスキャンデータを用いる。ここでは、入力フレームより前のフレームの推定結果を用いて線形予測した位置姿勢を初期値として利用する。推定誤差を計算する際には、各スキャンデータに対応する GPS および IMU データを真値とみなす。また、地図データがコンパクトであり、最も広く利用されるスキャン照合手法の 1 つである 3D NDT との比較も行なう。3D NDT に対するボクセルの 1 辺の長さは、Stoyanov ら [183] の調査において最良の精度を達成した 1.6 m に設定した。3D NDT に対しては、処理を高速化するために、スキャンデータに対して 1 辺が 1.6 m のボクセルグリッドフィルタ [184] を適用した。

評価指標として、進行方向への位置誤差 (Longitudinal error)、それに対して横方向への位置誤差 (Lateral error)、進行方向に対する方向誤差 (Orientation error)、処理時間 (Computational time) を用いる。Fig. 5.6 に各指標に対して横軸をフレーム番号としたグラフを、Table 5.1 にこれらの平均絶対値および地図容量 (Map data capacity) を示す。提案手法については、参考のために Fig. 5.6(a) の 1,500 フレーム付近に見られる大きな位置誤差の例、Fig. 5.6(c) の 3,000 フレーム付近に見られる大きな方向誤差の例をそれぞれ Fig. 5.7(a) および Fig. 5.7(b) に示す。

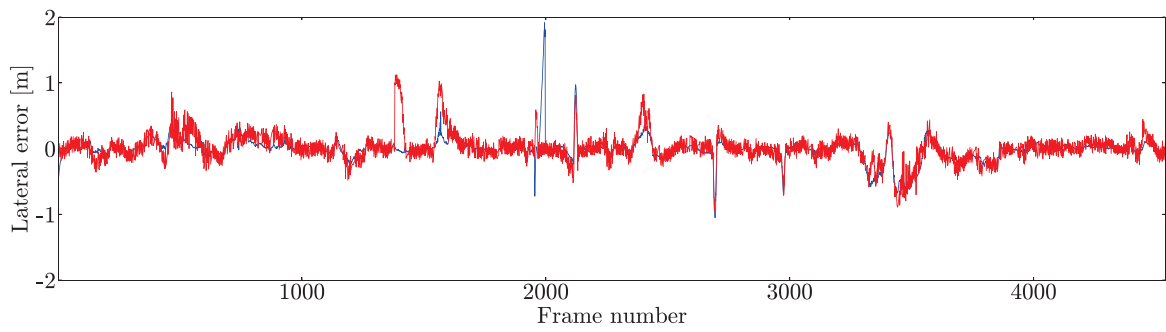
進行方向および横方向への位置誤差に関して、提案手法は 3D NDT と類似した傾向を示すが、わずかに大きな誤差を持つことが示された。これは、提案手法がボクセル単位の離散的な探索を行なうために、ボクセルを形成する際の空間の離散化誤差の影響を受けることに起因する。しかし、提案手法の位置に関する平均誤差は 15 cm 未満であり、車線単位の位置推定には影響しないと考えられる。どちらの手法においても、道路脇に建物の壁面のような表面が平坦な構造物が並ぶ環境においては大きな誤差を持つことがあった (Fig. 5.7(a) 参照)。これは、どちらも物体の 3 次元形状を手掛かりとする手法であり、形状的な変化に乏しい環境では正確な推定が困難なためである。方向誤差に関して、両手法で類似した精度を示した。どちらの手法も角を曲がる際に大きな方向誤差を持つ場合もあった (Fig. 5.7(b) 参照) が、その後のフレームにおいては精度良く推定された。

処理時間に関して、提案手法は KITTI データセットで利用された LiDAR センサの計

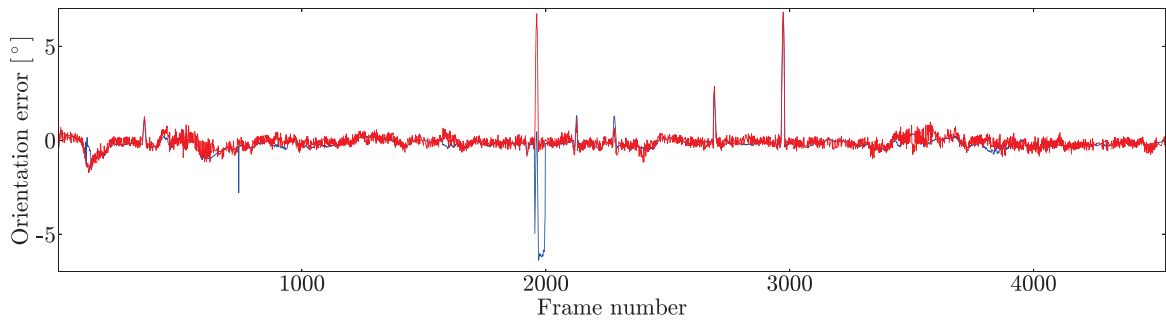




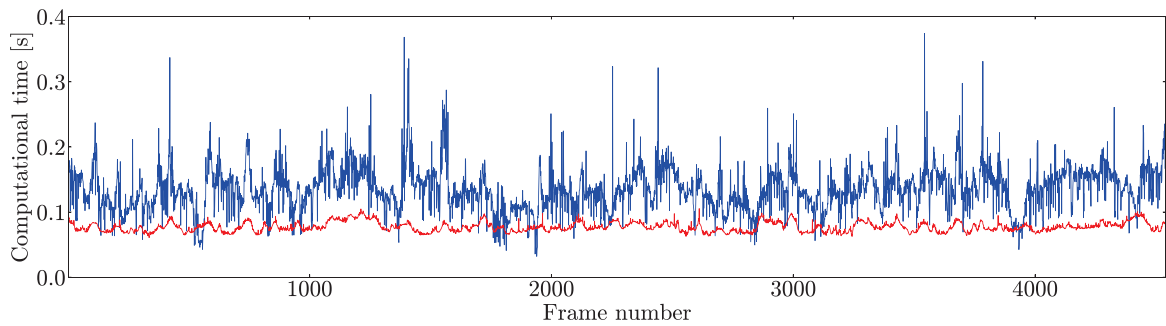
(a) 進行方向への位置誤差



(b) 進行方向に対して横方向への位置誤差



(c) 進行方向に対する方向誤差



(d) 処理時間

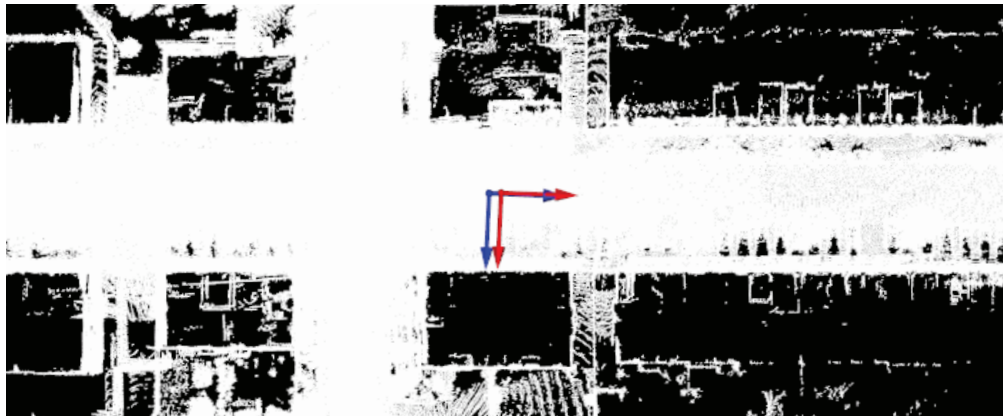
Fig. 5.6: 3D NDT および提案手法の自己位置推定性能.

Table 5.1: 自己位置推定性能および地図容量の比較. 進行方向への位置誤差 (Longitudinal error), それに対して横方向への位置誤差 (Lateral error), 進行方向に対する方向誤差 (Orientation error), 処理時間 (Computational time) の平均絶対値および経路 1km あたりの地図容量 (Map data capacity) を表す.

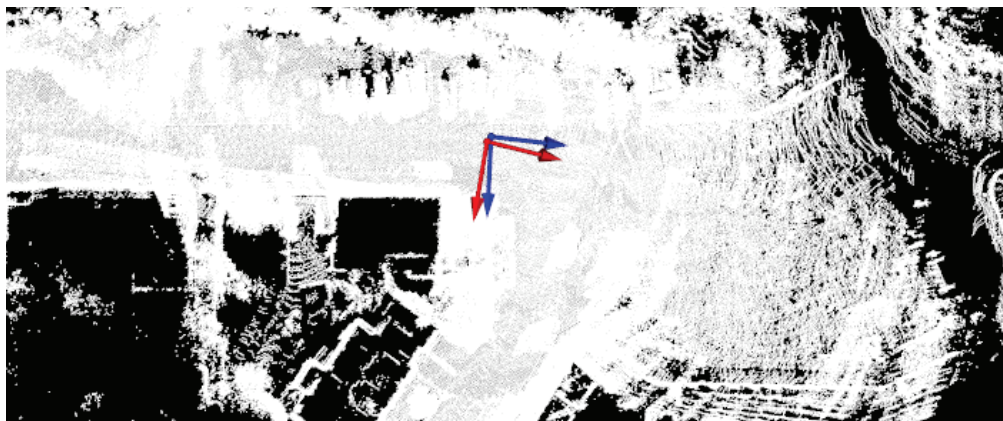
	3D NDT	Proposed method
Longitudinal error [m]	0.0936	0.126
Lateral error [m]	0.107	0.143
Orientation error [°]	0.312	0.273
Computational time [s]	0.173	0.0772
Map data capacity [MB/km]	8.06	0.0473

測間隔である 0.1 s を安定的に下回っており, 実時間処理が可能であることがわかる. 提案手法によって生成される地図の経路 1 km 当たりの容量は 0.0473 MB/km であり, 3D NDT の 8.06 MB/km に比べて 170 倍小容量な地図を生成できた. これは, ベクトル量子化に基づくデータの圧縮およびボリューム地図表現による効率的な格納のためである. 従って, 提案手法は 3D NDT と比べて容量が 100 分の 1 以下の地図を用いて同程度の精度を実現しており, その有効性を確認できる.

提案手法によって生成される地図の容量に関して, 例えば日本の道路の実延長である 122 万 km [185] に対する地図容量を単純換算すると約 57 GB となる. これは車載の記憶装置に格納する目的に対して妥当な大きさであり, 大規模な範囲を地図化する場合であっても全ての地図データを車両に搭載可能である. 一方, 走行中の車両へストリーミング配信をする場合を想定すると, 例えば車両が時速 100 km で走行する際に必要な通信帯域は約 10 kbps となる. この値は G.729 による音声通話に要求される通信帯域 (8 kbps) に匹敵するほど小さいため, 利用可能な通信帯域が狭い場合であっても安定的にストリーミング配信が可能であると思われる.



(a) Fig. 5.6(a) における 1,500 フレーム付近の大きな位置誤差



(b) Fig. 5.6(c) における 3,000 フレーム付近の大きな位置誤差

Fig. 5.7: 提案手法によって生じた大きな誤差の例. 直交する矢印が進行方向およびそれに対して横方向を表し, 青と赤の色がそれぞれ真値と推定値に対応する.

## 5.5 まとめ

本章では, 本論文で取り組む 3 つの課題である (i) DB 内のデータ数の削減, (ii) DB 内の各データの小容量化, (iii) 効率的なデータの照合の内, (iii) に取り組んだ. すなわち, コネクテッドカーによる自己位置推定において, 類似尺度の変更に基いて効率的なデータの照合を実現することにより, 高速かつ高精度な照合と特徴 DB の小容量化を両立した. 第 4 章の手法は自己位置推定処理が 3D NDT の方法に基づくため, 解の収束が遅い場合には処理時間が長くなるという問題があった. この問題に対処するために, 効率的に計算可能な 2 値ベクトル間の Hamming 距離を類似尺度として利用し, 標本化した離散的

なパラメータを非反復的に探索することによって、処理時間を安定的に短くする手法を提案した。なお、本章では特徴の変換に基づいて 3 次元点群を 2 値ベクトルの集合に変換する地図生成手法も合わせて提案した。

具体的には、本章では、2 値ベクトルの集合として表現される小容量な地図を用いて高速かつ高精度な自己位置推定を実現する手法を提案した。提案手法は同一体積の空間を表現する 2 値ベクトル間の類似度最大化問題として自己位置推定問題を定式化する。提案した地図生成手法は空間全てを地図化するため、困難な物体抽出処理を回避することができる。公開データセットを用いた評価実験により、提案手法は従来手法と同程度の自己位置推定性能を持ちつつ、地図容量を 100 分の 1 以下にすることを示した。更に、提案手法の処理時間は LiDAR センサの計測間隔を安定的に下回り、実時間処理が可能であることを示した。従って、提案手法はベースラインとなる自己位置推定手法と比べて同程度の推定精度を達成しつつ、処理時間および DB 容量を削減することが可能と言える。

## 第 6 章

# むすび

本章では，本論文の内容を簡潔にまとめた後に，今後の課題と展望を示す．

### 6.1 総括

情報通信技術の発展に伴い，スマートフォンを始めとする様々な移動端末が普及している．近年では，移動端末上でパターン認識技術に基づく様々なアプリケーションが実行されるようになった．その中でも，多数の特徴で構成される DB と特徴照合を行なうアプリケーションがある．このような DB は容量が大きいため，移動端末上で動作させるためには DB の小容量化が重要な課題となる．そこで本論文では，移動端末上で動作するアプリケーションにおいて，高速かつ高精度な特徴照合を実現しつつ特徴 DB を小容量化することを目的とした．特徴照合においては特徴の照合精度と計算量およびデータ量はトレードオフの関係にあるため，高速かつ高精度な特徴照合と特徴 DB の小容量化の両立は困難である．そのため，(i) DB 内のデータ数の削減，(ii) DB 内の各データの小容量化，(iii) 効率的なデータの照合，の 3 つを課題とし，これらを実現することによって上記の目的の達成を目指した．これらの課題を実現するために，(a) 特徴の選択，(b) 特徴の変換，(c) 類似尺度の変更，という観点で取り組んだ．(a) は特徴の集合から一部の特徴を選択し，(b) は特徴をより小さいデータ量の表現に変換し，(c) は変換されたままの特徴を照合するために類似尺度を変更するものである．本論文では，移動端末において多数の特徴と照合を行なう代表的なアプリケーションである画像検索と自己位置推定に焦点を当て，上記の課題への取り組みとして次の 3 つの研究を行なった．

- 研究 1：特徴の選択に基づく画像の索引付け
- 研究 2：特徴の変換に基づく環境地図の圧縮
- 研究 3：類似尺度の変更にに基づく自己位置推定

以下、各章で得られた成果をまとめる。

第 2 章では、本論文で着目するアプリケーションである画像検索と自己位置推定の関連研究について概説した。画像検索については、移動端末で撮影した画像をクエリとする技術が広く利用されている。画像をクエリとする場合、画像から抽出した局所特徴の集合を用いる方法が成功を収めている。この方法では、特徴ベクトルの照合によって特徴点の対応付けを行ない、幾何検証によって推定された幾何的に整合性を持つ特徴点の対応数を画像間類似度とする。この幾何検証により画像間類似度を高精度に計量することができるが、計算量が大きいという問題がある。この問題に対処するために、特徴点の情報を使用せず、特徴ベクトルの集合を異なる表現に変換する様々な手法が提案されている。これらの手法により、大規模な画像データベースに対して高速かつ高精度な検索が可能となっている。更に、類似度が高い少数の画像に対してのみ幾何検証を行なうことにより、検索精度を改善するとともに偽陽性の誤検出を防ぐことができる。自己位置推定については、カメラや LiDAR で計測したセンサデータと環境地図を用いて自己位置を推定する技術が広く利用されている。これらの技術は大域的自己位置推定と局所的自己位置推定に大別できる。前者は探索範囲の制限なしに環境地図上での自己位置を推定する。後者は初期姿勢と呼ばれる位置と姿勢の初期値を基準として、その周辺の局所的な範囲のみを探索することにより、初期姿勢に対する相対的な自己位置を推定する。カメラと LiDAR を用いた代表的な大域的自己位置推定手法は、それぞれ画像検索と 3 次元点群検索である。また、代表的な局所的自己位置推定手法として、カメラに対しては白線や標識のようなランドマークの対応付け、LiDAR に対しては 3 次元点群のスキャン照合が挙げられる。

第 3 章では、第 1 の研究である特徴の選択に基づく画像の索引付けについて述べた。ここでは、アプリケーションとして BoVW に基づく画像検索を想定した。BoVW は元々、(b) 特徴の変換、(c) 類似尺度の変更に観点を利用している。すなわち、局所特徴を VW に変換し、類似尺度を局所特徴集合間の Inlier の個数から VW ヒストグラム間の余弦類似度に変換する。この手法では、画像の索引付けに用いる特徴の個数を増加させることによって画像検索の精度を改善することができるが、一方で DB 容量が大きくなるという問

題があった。そこで、(a) 特徴の選択に基づいて有益な特徴のみを選択した DB を構築することにより、高速かつ高精度な照合および DB の小容量化を実現した。具体的には、特徴点の検出と特徴ベクトルの量子化の再現性に基づく特徴信頼度と、それを用いた特徴選択手法を提案した。すなわち、多視点画像の各画像にわたって信頼度が高い特徴を選択した DB を構築することによって、より少数の特徴のみを用いて視点変化に頑健な特徴照合を実現した。提案手法では、特徴選択を特徴空間上での最大被覆問題として定式化し、特定の制約関数の下で最も高い信頼度を得る局所特徴の集合を求めた。この目的のために、BoVW における特徴の一致条件と整合性を持つ制約関数を設計した。また、最大被覆問題は NP 困難であり、多項式時間で実行不可能であることから、貪欲法に基づく近似解法を提案した。評価実験ではスマートフォンで撮影されたクエリ画像を用いて、画像検索の精度を評価した。実験結果は、提案手法が従来手法と比べて DB 容量および検索時間を増加させることなく、検索精度を 12.7% 改善することを示した。更に、提案手法によって DB を構築するための処理時間を測定し、多項式時間で実行可能であることを確認した。

第 4 章と第 5 章はともにアプリケーションとして自己位置推定について検討したが、それぞれ異なる意義を持つ。第 4 章では元となる手法として 3D NDT を想定した。これは幅広く使用される手法であるため、この手法による地図と互換性がある地図を構築することにより、既存の地図 DB やソフトウェアを利用できるという意義がある。一方、第 5 章では、3D NDT が持つ問題に対処した新たな手法を提案するという意義がある。すなわち、3D NDT は解の収束が遅い場合に処理時間が長くなるという問題があるのに対し、提案手法では安定的に処理時間を短くすることができる。

第 4 章では、第 2 の研究である特徴の変換に基づく環境地図の圧縮について述べた。3D NDT は元々、(b) 特徴の変換、(c) 類似尺度の変更の観点を利用している。すなわち、3 次元点群を正規分布のパラメータに集約し、点と点の距離に基づく類似尺度から点と正規分布から計算される尤度に基づく類似尺度に変更する。しかし、この手法では 3 次元点群を集約した後であっても DB 容量は依然として大きいという問題がある。この問題に対処するために、(b) 特徴の変換に基づいて正規分布のパラメータを量子化によって更に変換することにより、DB 容量を削減する手法を提案した。具体的には、正規分布パラメータを持つボクセルの集合として表現される環境地図に対して、正規分布パラメータをベクトル表現に変換し、ベクトル量子化によってデータを圧縮する手法を提案した。ただし、ボクセルごとにベクトル量子化を適用すると、個々のボクセルが持つパラメータ数が少な

いために圧縮比は低い。そのため、圧縮比の改善のために占有／非占有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手法を提案した。自己位置推定の実行時には、圧縮前の表現に逆変換することによって 3D NDT による環境地図との互換性を実現する。これにより、3D NDT を実装した既存のソフトウェアを用いて自己位置推定を行なうことが可能となる。評価実験により、提案手法は無圧縮の場合の 3D NDT と比べて同程度の自己位置推定精度および処理時間を達成しつつ、DB 容量を 100 分の 1 から 1,000 分の 1 以下にすることを示した。

第 5 章では、第 3 の研究である類似尺度の変更に基づく自己位置推定について述べた。3 次元点群を用いた代表的な自己位置推定手法は反復的な探索に基づくため、解の収束が遅い場合には処理時間が長くなるという問題がある。非反復的な探索に基づく方法によって処理時間を安定化させることができるが、従来使用される類似尺度は計算量が大きいという問題がある。この問題に対処しつつ、高精度な特徴照合と DB の小容量化を実現するために、(b) 特徴の変換、(c) 類似尺度の変更に基づく新たな手法を提案した。具体的には、(b) 特徴の変換に基づいて 3 次元点群を 2 値ベクトルの集合に変換する地図生成手法と、(c) 類似尺度の変更に基づいて効率的に計算可能な 2 値ベクトル間の Hamming 距離を類似尺度として使用する自己位置推定手法を提案した。地図生成手法は、初めに 3 次元点群をボクセルを用いて分割し、2 値占有モデルに従って各ボクセルの占有情報を 2 値で表現する。そして、複数のボクセルの占有情報を空間的位置関係に従って結合することにより、環境地図を 2 値ベクトルの集合に変換する。自己位置推定の実行時には、予め設定した離散的な位置姿勢変換パラメータを用いてスキャンデータを変換し、地図データとの間で Hamming 距離の総和を計量する。そして、この Hamming 距離の総和が最小となる位置姿勢変換パラメータを探索する。これにより、非反復的かつ高速に解を得ることができる。評価実験により、提案手法は従来手法と比べて同程度の自己位置推定精度をより短い処理時間で達成しつつ、DB 容量を 100 分の 1 以下にすることを示した。

上記の 3 つの研究により、本論文で課題とした (i) DB 内のデータ数の削減、(ii) DB 内の各データの容量化、(iii) 効率的なデータの照合を実現することができた。そして、これらの課題を実現することによって、移動端末上で動作するアプリケーションにおいて高速かつ高精度な特徴照合を実現しつつ特徴 DB を小容量化するという、本論文の目的を達成することができた。すなわち、特徴の照合精度と計算量およびデータ量がトレードオフ



の関係にある中で、高速かつ高精度な特徴照合と小容量な DB を両立することができた。課題 (i), (ii), (iii) に対応する提案手法は、アプリケーションの処理時間と DB 容量を維持しつつ精度を改善する、あるいは精度を維持しつつ処理時間と DB 容量を削減するなど、要件に応じて精度、処理時間、DB 容量を調整できることを示した。そのため、例えば移動端末の記憶容量が十分にある場合には精度改善を、ない場合には DB 容量の削減を行なうなどの使い分けも考えられる。また、課題 (i), (ii), (iii) を実現するための観点である (a) 特徴の選択、(b) 特徴の変換、(c) 類似尺度の変更は、1 つの手法に対して複数導入することが可能であり、それによって手法の性能を改善することができた。従って、課題 (i), (ii), (iii) の全てを実現することは、本論文の目的を達成するための重要な手段として位置付けられる。以上より、本論文は移動端末上で動作するアプリケーションに関する研究に対して、有益な知見を与えられたい。

## 6.2 今後の課題と展望

本論文では、移動端末における画像検索と自己位置推定アプリケーションのための、大規模な特徴 DB の小容量化に取り組んだ。

画像検索については、多視点画像を単一視点画像の変形によって生成するために、対象を平面物に制限した。しかし、実世界には 3 次元形状を持つ物体が多数存在するため、対象を立体物へ拡張することは今後の課題の 1 つである。また、本論文では特徴の選択に際して視点変化に対する再現性のみを考慮した。しかし、実利用を行なう上ではそれ以外にもボケや白飛び、遮蔽、背景など様々な外乱が考えられるため、それらに対しても頑健な検索を可能にする特徴の選択が必要である。移動端末を用いて任意の形状を持つ物体の検索を実現することにより、ユーザは移動先で見かけた物体の情報を容易に取得できるようになるであろう。今後は本研究を更に発展させ、多様な条件の下で高精度な画像検索を実現する手法を検討していきたい。

自己位置推定については、本論文ではその方法を LiDAR によるスキャン照合に限定した。しかし、車両はカメラや加速度センサを始めとする多数のセンサを搭載できるため、それらと連携することによる頑健な自己位置推定の実現が今後の課題の 1 つである。また、本論文では環境地図に対して特徴の変換を適用する際に自己位置推定に利用することのみを考えた。しかし、環境地図は領域分割や変化検知などの様々なアプリケーションで

の利用が考えられるため、それらに対しても汎用的に利用可能かつ小容量な特徴への変換は重要な課題である。また、本論文では、2 値ベクトル間の Hamming 距離の総和に類似尺度を変更した。しかし、この尺度が 3 次元点群間の Euclidean 距離に基づく類似尺度を反映するか否かは、検証の余地がある。環境地図を表現する特徴に対して最適な類似尺度を決定する方法について検討することも興味深い。今後は本研究で提案した方法を応用し、来たるべきコネクテッドカー時代の実現に貢献したい。

情報社会の発展に伴い、実世界の情報を認識する機会は今後ますます多くなると考えられる。移動端末における画像検索と自己位置推定アプリケーションは、実世界の情報を認識するための重要な技術である。例えばスマートフォンやコネクテッドカーに搭載させたセンサを用いて画像検索や自己位置推定を行なうことにより、実世界の位置と紐づけられた情報を認識することができる。移動端末を用いて実世界の情報を認識することができれば、人々の間での情報の共有や、情報の分析による問題解決、付加的な情報の取得など、多くの価値の創出が見込まれる。しかし、現在の移動端末では、記憶装置の容量や通信帯域の制限などの要因によって、認識できる対象は限定的となっている。そのため、今後は本研究で得られた知見を基に、実世界のあらゆる情報を認識することが可能な世界を実現したい。

最後に、本研究の成果が移動端末における大規模特徴照合アプリケーションに関する研究の発展に寄与することを願い、本論文の結びとする。

# 謝辞

初めに、本論文の主査である名古屋大学大学院情報学研究科 村瀬洋 教授に深く感謝いたします。村瀬先生には研究室へ受け入れて頂くとともに、研究活動全般において多大なご支援とご協力を賜りました。ここに最大の感謝を表します。

本論文の副査である名古屋大学大学院情報学研究科 井手一郎 教授，同 戸田智基 教授，同 出口大輔 准教授に感謝いたします。皆様にはご多忙の中、本論文をご精読頂き、構成から細かい表現に至るまでの確なご指摘を頂きました。心より御礼申し上げます。

本研究を進めるにあたり、日頃から数多くのご指導ご鞭撻を賜りました名古屋大学大学院情報学研究科 川西康友 講師に感謝いたします。川西先生には筆者の研究に関して常に熱心にご討論頂き、そして数多くの有益なご助言を頂戴いたしました。

本論文の執筆の機会を与えて下さいました株式会社 KDDI 総合研究所 会長 中島康之 博士，同 所長 中村元 博士，同 総務部門および企画広報部門部門長 柳原広昌 博士，同 メディア ICT 部門部門長 内藤整 博士，同 フロンティア研究室室長 田坂和之 博士，同 超臨場感通信グループグループリーダー 河村圭 博士に感謝いたします。社会人として業務を行ないながら研究を続けることができたのは、皆様の暖かいご支援のおかげでございます。

システム知能情報学セミナーを履修する機会を与えて下さいました名古屋大学大学院情報学研究科 森健策 教授，同 小田昌宏 助教に謝意を表します。当該セミナーへの参加なくして、学位の取得はありませんでした。

最後に、長年にわたり筆者の研究活動を応援して下さいました、家族に感謝いたします。



## 参考文献

- [1] D. Chen and B. Girod, “Memory-efficient image databases for mobile visual search,” *IEEE MultiMedia Magazine*, vol.21, no.1, pp.14–23, 2014.
- [2] J. Panda, M.S. Brown, and C.V. Jawahar, “Offline mobile instance retrieval with a small memory footprint,” In *Proceedings of 14th IEEE International Conference on Computer Vision (ICCV)*, pp.1257–1264, 2013.
- [3] D. Li and M.C. Chuah, “EMOD: An efficient on-device mobile visual search system,” In *Proceedings of 6th ACM Multimedia Systems Conference (MMSys)*, pp.25–36, 2015.
- [4] A. Hartl, D. Schmalstieg, and G. Reitmayr, “Client-side mobile visual search,” In *Proceedings of 2014 International Conference on Computer Vision Theory and Applications (VISAPP)*, vol.3, pp.125–132, 2014.
- [5] J. Levinson, M. Michael, and T. Sebastian, “Map-based precision vehicle localization in urban environments,” In *Proceedings of 3rd Robotics: Science and Systems Conference (RSS)*, 8p., 2007.
- [6] M. Magnusson, A. Lilienthal, and T. Duckett, “Scan registration for autonomous mining vehicles using 3D-NDT,” *Journal of Field Robotics (JFR)*, vol.24, no.10, pp.803–827, 2007.
- [7] T. Stoyanov, M. Martin, and A.J. Lilienthal, “Point set registration through minimization of the L2 distance between 3D-NDT models,” In *Proceedings of 2012 IEEE International Conference on Robotics and Automation (ICRA)*, pp.5196–5201, 2012.

- [8] R.W. Wolcott and R.M. Eustice, “Robust LIDAR localization using multiresolution Gaussian mixture maps for autonomous driving,” *International Journal of Robotics Research (IJRR)*, vol.36, no.3, pp.292–319, 2017.
- [9] 総務省, “令和元年版情報通信白書”, 2019. <https://www.soumu.go.jp/johotsusintokei/whitepaper/ja/r01/pdf/01honpen.pdf> [2020/12/1 アクセス] .
- [10] H. Dakroub, A. Shaout, and A. Awajan, “Connected car architecture and virtualization,” *SAE International Journal of Passenger Cars-Electronic and Electrical Systems*, vol.9, no.2016-01-0081, pp.153–159, 2016.
- [11] M.K. Svangren, M.B. Skov, and J. Kjeldskov, “The connected car: An empirical study of electric cars as mobile digital devices,” In *Proceedings of 19th ACM International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI)*, pp.1–12, 2017.
- [12] S.J. Jang, “Development trend of operating system technology for smart car,” *International Journal of Recent Trends in Engineering and Research (IJRTER)*, vol.3, no.1, pp.107–115, 2017.
- [13] 高度情報通信ネットワーク社会推進戦略本部・官民データ活用推進戦略会議, “官民 ITS 構想・ロードマップ 2019”, 2019. <https://www.kantei.go.jp/jp/singi/it2/kettei/pdf/20190607/siryou9.pdf> [2020/12/1 アクセス] .
- [14] 総務省, “平成 29 年版情報通信白書”, 2017. <https://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h29/pdf/29honpen.pdf> [2020/12/1 アクセス] .
- [15] S. Yaqoob, A. Ullah, M. Akbar, M. Imran, and M. Guizani, “Fog-assisted congestion avoidance scheme for internet of vehicles,” In *Proceedings of 14th IEEE International Wireless Communications & Mobile Computing Conference (IWCMC)*, pp.618–622, 2018.
- [16] C. Chaikalis, V. Zygouras, and I.K. Savvas, “Reconfiguration between congestion avoidance and multiplexing techniques for 5G,” In *Proceedings of 26th*

- 
- IEEE Telecommunications Forum (TELFOR), pp.1–4, 2018.
- [17] B. Han, A. DeDomenico, G. Dandachi, A. Drosou, D. Tzovaras, R. Querio, F. Moggio, O. Bulakci, and H.D. Schotten, “Admission and congestion control for 5G network slicing,” In Proceedings of 2018 IEEE International Conference on Standards for Communications and Networking (CSCN), pp.1–6, 2018.
- [18] B. Han, V. Sciancalepore, O. Holland, M. Dohler, and H.D. Schotten, “D2D-based grouped random access to mitigate mobile access congestion in 5G sensor networks,” IEEE Communications Magazine, vol.57, no.9, pp.93–99, 2019.
- [19] P. Peter, J. Contelly, and J. Weickert, “Compressing audio signals with inpainting-based sparsification,” In Proceedings of 7th International Conference on Scale Space and Variational Methods in Computer Vision (SSVM), pp.92–103, 2019.
- [20] M. Farzaneh, R.M. Toroghi, and M. Asgari, “Audio compression using graph-based transform,” In Proceedings of 9th IEEE International Symposium on Telecommunications (IST), pp.410–415, 2018.
- [21] L.L. Motta, R.S. Mendes, and M.H.M. Costa, “An audio compression method based on wavelet packet decomposition, ordering, and polynomial approximation of expressive coefficients,” In Proceedings of 11th IEEE Latin-American Conference on Communications (LATINCOM), pp.1–6, 2019.
- [22] F. Mentzer, E. Agustsson, M. Tschannen, R. Timofte, and L. van Gool, “Practical full resolution learned lossless image compression,” In Proceedings of 2019 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.10629–10638, 2019.
- [23] F. Mentzer, E. Agustsson, M. Tschannen, R. Timofte, and L. van Gool, “Conditional probability models for deep image compression,” In Proceedings of 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.4394–4402, 2018.
- [24] M. Li, W. Zuo, S. Gu, D. Zhao, and D. Zhang, “Learning convolutional net-

- works for content-weighted image compression,” In Proceedings of 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.3214–3223, 2018.
- [25] O. Rippel, S. Nair, C. Lew, S. Branson, A.G. Anderson, and L. Bourdev, “Learned video compression,” In Proceedings of 17th IEEE International Conference on Computer Vision (ICCV), pp.3454–3463, 2019.
- [26] G. Lu, W. Ouyang, D. Xu, X. Zhang, C. Cai, and Z. Gao, “DVC: An end-to-end deep video compression framework,” In Proceedings of 2019 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.11006–11015, 2019.
- [27] A. Habibian, T.V. Rozendaal, J.M. Tomczak, and T.S. Cohen, “Video compression with rate-distortion autoencoders,” In Proceedings of 17th IEEE International Conference on Computer Vision (ICCV), pp.7033–7042, 2019.
- [28] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.Y. Fu, and A.C. Berg, “SSD: Single shot multibox detector,” In Proceedings of 14th European Conference on Computer Vision (ECCV), Part I, Lecture Notes in Computer Science, vol.9905, pp.21–37, 2016.
- [29] S. Shi, X. Wang, and H. Li, “PointRCNN: 3D object proposal generation and detection from point cloud,” In Proceedings of 2019 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.770–779, 2019.
- [30] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” In Proceedings of 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.7132–7141, 2018.
- [31] C.R. Qi, H. Su, K. Mo, and L.J. Guibas, “PointNet: Deep learning on point sets for 3D classification and segmentation,” In Proceedings of 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.652–660, 2017.
- [32] L. Zheng, Y. Yang, and Q. Tian, “SIFT meets CNN: A decade survey of instance retrieval,” IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol.40, no.5, pp.1224–1244, 2017.



- 
- [33] M.M.B. Ismail, “A survey on content-based image retrieval,” *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol.8, no.5, pp.159–170, 2017.
  - [34] T. Sattler, W. Maddern, C. Toft, A. Torii, L. Hammarstrand, E. Stenborg, D. Safari, M. Okutomi, M. Pollefeys, J. Sivic, F. Kahl, and T. Pajdla, “Benchmarking 6DOF outdoor visual localization in changing conditions,” In *Proceedings of 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.8601–8610, 2018.
  - [35] W. Lu, Y. Zhou, G. Wan, S. Hou, and S. Song, “L3-Net: Towards learning based LiDAR localization for autonomous driving,” In *Proceedings of 2019 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.6389–6398, 2019.
  - [36] J. Wu, C. Leng, Y. Wang, Q. Hu, and J. Cheng, “Quantized convolutional neural networks for mobile devices,” In *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.4820–4828, 2016.
  - [37] A. Zhou, A. Yao, Y. Guo, L. Xu, and Y. Chen, “Incremental network quantization: Towards lossless CNNs with low-precision weights,” In *Proceedings of 5th IEEE International Conference on Learning Representations (ICLR)*, 12p., 2017.
  - [38] F. Chollet, “Xception: Deep learning with depthwise separable convolutions,” In *Proceedings of 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.1251–1258, 2017.
  - [39] M. Jaderberg, A. Vedaldi, and A. Zisserman, “Speeding up convolutional neural networks with low rank expansions,” In *Proceedings of 25th British Machine Vision Conference (BMVC)*, 13p., 2014.
  - [40] S. Han, H. Mao, and W.J. Dally, “Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding,” In *Proceedings of 4th IEEE International Conference on Learning Representations (ICLR)*,

- 13p., 2016.
- [41] W. Wen, C. Wu, Y. Wang, Y. Chen, and H. Li, “Learning structured sparsity in deep neural networks,” In Proceedings of 30th International Conference on Neural Information Processing Systems (NIPS), pp.2082–2090, 2016.
  - [42] G. Hinton, O. Vinyals, and J. Dean, “Distilling the knowledge in a neural network,” Computing Research Repository arXiv preprint, arXiv:1503.02531, 2015.
  - [43] J. Yim, D. Joo, J. Bae, and J. Kim, “A gift from knowledge distillation: Fast optimization, network minimization and transfer learning,” In Proceedings of 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.4133–4141, 2017.
  - [44] A.G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, “MobileNets: Efficient convolutional neural networks for mobile vision applications,” Computing Research Repository arXiv preprint, arXiv:1704.04861, 2017.
  - [45] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.C. Chen, “MobileNetV2: Inverted residuals and linear bottlenecks,” In Proceedings of 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.4510–4520, 2018.
  - [46] A. Howard, M. Sandler, G. Chu, L.C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, Q.V. Le, and H. Adam, “Searching for MobileNetV3,” In Proceedings of 17th IEEE International Conference on Computer Vision (ICCV), pp.1314–1324, 2019.
  - [47] X. Zhang, X. Zhou, M. Lin, and J. Sun, “ShuffleNet: An extremely efficient convolutional neural network for mobile devices,” In Proceedings of 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.6848–6856, 2018.
  - [48] N. Ma, X. Zhang, H.T. Zheng, and J. Sun, “ShuffleNet V2: Practical guidelines for efficient CNN architecture design,” In Proceedings of 15th European Con-

- 
- ference on Computer Vision (ECCV), Part XIV, Lecture Notes in Computer Science, vol.11218, pp.122–138, 2018.
- [49] D. Moise, D. Shestakov, G. Gudmundsson, and L. Amsaleg, “Terabyte-scale image similarity search: Experience and best practice,” In Proceedings of 2013 IEEE International Conference on Big Data (ICBD), pp.674–682, 2013.
- [50] X. Wei, I.A. Bârsan, S. Wang, J. Martinez, and R. Urtasun, “Learning to localize through compressed binary maps,” In Proceedings of 2019 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.10316–10324, 2019.
- [51] M. Dash and H. Liu, “Feature selection for classification,” *Intelligent Data Analysis*, vol.1, no.3, pp.131–156, 1997.
- [52] B. Xue, M. Zhang, W.N. Browne, and X. Yao, “A survey on evolutionary computation approaches to feature selection,” *IEEE Transactions on Evolutionary Computation (TEVC)*, vol.20, no.4, pp.606–626, 2015.
- [53] J. Sivic and A. Zisserman, “Video Google: A text retrieval approach to object matching in videos,” In Proceedings of 9th IEEE International Conference on Computer Vision (ICCV), pp.1470–1477, 2003.
- [54] P.J. Besl and N.D. McKay, “A method for registration of 3-D shapes,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol.14, no.2, pp.239–256, 1992.
- [55] A.W.M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, “Content-based image retrieval at the end of the early years,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol.22, no.12, pp.1349–1380, 2000.
- [56] D.G. Lowe, “Object recognition from local scale-invariant features,” In Proceedings of 7th IEEE International Conference on Computer Vision (ICCV), pp.1150–1157, 1999.
- [57] D.G. Lowe, “Distinctive image feature from scale-invariant keypoints,” *Interna-*

- tional Journal of Computer Vision (IJCV), vol.60, no.2, pp.91–110, 2004.
- [58] Y. Ke and R. Sukthankar, “PCA-SIFT: A more distinctive representation for local image descriptors,” In Proceedings of 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), pp.506–513, 2004.
- [59] K. Mikolajczyk and C. Schmid, “A performance evaluation of local descriptors,” IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol.27, no.10, pp.1615–1630, 2005.
- [60] H. Bay, T. Tuytelaars, and L. van Gool, “SURF: Speeded up robust features,” In Proceedings of 9th European Conference on Computer Vision (ECCV), Part I, Lecture Notes in Computer Science, vol.3951, pp.404–417, 2006.
- [61] E. Tola, V. Lepetit, and P. Fua, “DAISY: An efficient dense descriptor applied to wide-baseline stereo,” IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol.32, no.5, pp.815–830, 2009.
- [62] R. Arandjelović and A. Zisserman, “Three things everyone should know to improve object retrieval,” In Proceedings of 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.2911–2918, 2012.
- [63] J. Dong and S. Soatto, “Domain-size pooling in local descriptors: DSP-SIFT,” In Proceedings of 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.5097–5106, 2015.
- [64] S. Leutenegger, M. Chli, and R.Y. Siegwart, “BRISK: Binary robust invariant scalable keypoints,” In Proceedings of 13th IEEE International Conference on Computer Vision (ICCV), pp.2548–2555, 2011.
- [65] E. Rubee, V. Rabaud, K. Konolige, and G. Bradski, “ORB: An efficient alternative to SIFT or SURF,” In Proceedings of 13th IEEE International Conference on Computer Vision (ICCV), pp.2564–2571, 2011.
- [66] A. Alahi, R. Ortiz, and P. Vandergheynst, “FREAK: Fast retina keypoint,” In Proceedings of 2012 IEEE Conference on Computer Vision and Pattern Recog-

- 
- niton (CVPR), pp.510–517, 2012.
- [67] C. Leng, H. Zhang, B. Li, G. Cai, Z. Pei, and L. He, “Local feature descriptor for image matching: A survey,” *IEEE Access*, vol.7, pp.6424–6434, 2018.
  - [68] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid, “DeepMatching: Hierarchical deformable dense matching,” *International Journal of Computer Vision (IJCV)*, vol.120, no.3, pp.300–323, 2016.
  - [69] M. Fischler and R. Bolles, “Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography,” *Communications of the ACM*, vol.24, no.6, pp.381–395, 1981.
  - [70] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, “Object retrieval with large vocabularies and fast spatial matching,” In *Proceedings of 2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.1–8, 2007.
  - [71] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, “Lost in quantization: Improving particular object retrieval in large scale image databases,” In *Proceedings of 2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.1–8, 2008.
  - [72] H. Jégou, D. Matthijs, and S. Cordelia, “Improving bag-of-features for large scale image search,” *International Journal of Computer Vision (IJCV)*, vol.87, no.3, pp.316–336, 2010.
  - [73] H. Jégou, D. Matthijs, and S. Cordelia, “Hamming embedding and weak geometric consistency for large scale image search,” In *Proceedings of 10th European Conference on Computer Vision (ECCV)*, Part I, *Lecture Notes in Computer Science*, vol.5302, pp.304–317, 2008.
  - [74] F. Perronnin and C. Dance, “Fisher kernels on visual vocabularies for image categorization,” In *Proceedings of 2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.1–8, 2007.

- [75] F. Perronnin, Y. Liu, J. Sánchez, and H. Poirier, “Large-scale image retrieval with compressed fisher vectors,” In Proceedings of 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.3304–3311, 2010.
- [76] H. Jégou, M. Douze, C. Schmid, and P. Pérez, “Aggregating local descriptors into a compact image representation,” In Proceedings of 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.3384–3391, 2010.
- [77] H. Jégou, M. Douze, and C. Schmid, “Product quantization for nearest neighbor search,” IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol.33, no.1, pp.117–128, 2010.
- [78] A. Babenko and V. Lempitsky, “Aggregating deep convolutional features for image retrieval,” In Proceedings of 15th IEEE International Conference on Computer Vision (ICCV), pp.1269–1277, 2015.
- [79] G. Tolias, R. Sivic, and H. Jégou, “Particular object retrieval with integral max-pooling of CNN activations,” In Proceedings of 4th IEEE International Conference on Learning Representations (ICLR), 12p., 2016.
- [80] J. Deng, W. Dong, R. Socher, L.J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” In Proceedings of 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), pp.248–255, 2009.
- [81] S. Bell and K. Bala, “Learning visual similarity for product design with convolutional neural networks,” ACM Transactions on Graphics (TOG), vol.34, no.4, pp.1–10, 2015.
- [82] A. Gordo, J. Almazán, J. Revaud, and D. Larlus, “Deep image retrieval: Learning global representations for image search,” In Proceedings of 14th European Conference on Computer Vision (ECCV), Part VI, Lecture Notes in Computer Science, vol.9910, pp.241–257, 2016.
- [83] F. Radenović, G. Tolias, and O. Chum, “Fine-tuning CNN image retrieval with no human annotation,” IEEE Transactions on Pattern Analysis and Machine

- 
- Intelligence (PAMI), vol.41, no.7, pp.1655–1668, 2018.
- [84] R. Hadsell, S. Chopra, and Y. LeCun, “Dimensionality reduction by learning an invariant mapping,” In Proceedings of 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), pp.1735–1742, 2006.
- [85] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu, “Learning fine-grained image similarity with deep ranking,” In Proceedings of 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.1386–1393, 2014.
- [86] K.M. Yi, E. Trulls, V. Lepetit, and P. Fua, “LIFT: Learned invariant feature transform,” In Proceedings of 14th European Conference on Computer Vision (ECCV), Part VI, Lecture Notes in Computer Science, vol.9910, pp.467–483, 2016.
- [87] H. Noh, A. Araujo, J. Sim, T. Weyand, and B. Han, “Large-scale image retrieval with attentive deep local features,” In Proceedings of 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.3456–3465, 2017.
- [88] F. Radenović, A. Iscen, G. Tolias, Y. Avrithis, and O. Chum, “Revisiting Oxford and Paris: Large-scale image retrieval benchmarking,” In Proceedings of 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.5706–5715, 2018.
- [89] D. Nistér and H. Stewénus, “Scalable recognition with a vocabulary tree,” In Proceedings of 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), pp.2161–2168, 2006.
- [90] V. Chandrasekhar, D.M. Chen, S.S. Tsai, N.M. Cheung, H. Chen, G. Takacs, Y. Reznik, R. Vedantham, R. Grzeszczuk, J. Bach, and B. Girod, “The Stanford mobile visual search data set,” In Proceedings of 2nd ACM Multimedia Systems Conference (MMSys), pp.117–122, 2011.
- [91] T. Weyand, A. Araujo, B. Cao, and J. Sim, “Google landmarks dataset v2-A large-scale benchmark for instance-level recognition and retrieval,” In Proceed-

- ings of 2020 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.2575–2584, 2020.
- [92] Y. Uchida, S. Sakazawa, and S. Satoh, “Image retrieval with fisher vectors of binary features,” *ITE Transactions on Media Technology and Applications (MTA)*, vol.4, no.4, pp.326–336, 2016.
- [93] L.T. Hsu, F. Chen, and S. Kamijo, “Evaluation of multi-GNSSs and GPS with 3D map methods for pedestrian positioning in an urban canyon environment,” *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences (FECCS)*, vol.98, no.1, pp.284–293, 2015.
- [94] J. Hays and A.A. Efros, “IM2GPS: Estimating geographic information from a single image,” In *Proceedings of 2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.1–8, 2008.
- [95] J. Knopp, J. Sivic, and T. Pajdla, “Avoiding confusing features in place recognition,” In *Proceedings of 11th European Conference on Computer Vision (ECCV)*, Part I, *Lecture Notes in Computer Science*, vol.6311, pp.748–761, 2010.
- [96] R. Arandjelović and A. Zisserman, “DisLocation: Scalable descriptor distinctiveness for location recognition,” In *Proceedings of 12th Asian Conference on Computer Vision (ACCV)*, Part IV, *Lecture Notes in Computer Science*, vol.9006, pp.188–204, 2014.
- [97] T. Sattler, M. Havlena, K. Schindler, and M. Pollefeys, “Large-scale location recognition and the geometric burstiness problem,” In *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.1582–1590, 2016.
- [98] Y. Tian, C. Chen, and M. Shah, “Cross-view image matching for geo-localization in urban environments,” In *Proceedings of 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.3608–3616, 2017.
- [99] S. Hu, M. Feng, R.M.H. Nguyen, and G.H. Lee, “CVM-Net: Cross-view match-



- 
- ing network for image-based ground-to-aerial geo-localization,” In Proceedings of 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.7258–7267, 2018.
- [100] S. Cai, Y. Guo, S. Khan, J. Hu, and G. Wen, “Ground-to-aerial image geo-localization with a hard exemplar reweighting triplet loss,” In Proceedings of 2019 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.8391–8400, 2019.
- [101] T. Sattler, B. Leibe, and L. Kobbelt, “Fast image-based localization using direct 2D-to-3D matching,” In Proceedings of 13th IEEE International Conference on Computer Vision (ICCV), pp.667–674, 2011.
- [102] B. Zeisl, T. Sattler, and M. Pollefeys, “Camera pose voting for large-scale image-based localization,” In Proceedings of 15th IEEE International Conference on Computer Vision (ICCV), pp.2704–2712, 2015.
- [103] M.A. Uy and G.H. Lee, “PointNetVLAD: Deep point cloud based retrieval for large-scale place recognition,” In Proceedings of 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.4470–4479, 2018.
- [104] W. Zhang and C. Xiao, “PCAN: 3D attention map learning using contextual information for point cloud based retrieval,” In Proceedings of 2019 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.12436–12445, 2019.
- [105] Z. Liu, S. Zhou, C. Suo, P. Yin, W. Chen, H. Wang, H. Li, and Y.H. Liu, “LPD-Net: 3D point cloud learning for large-scale place recognition and environment analysis,” In Proceedings of 2019 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.2831–2840, 2019.
- [106] G. Kim, B. Park, and A. Kim, “1-day learning, 1-year localization: Long-term LiDAR localization using scan context image,” *IEEE Robotics and Automation Letters (RA-L)*, vol.4, no.2, pp.1948–1955, 2019.
- [107] R.B. Rusu, Z.C. Marton, N. Blodow, and M. Beetz, “Persistent point feature

- histograms for 3D point clouds,” In Proceedings of 10th International Conference on Intelligent Autonomous Systems (IAS), pp.119–128, 2008.
- [108] R.B. Rusu, N. Blodow, and M. Beetz, “Fast point feature histograms (FPFH) for 3D registration,” In Proceedings of 2009 IEEE International Conference on Robotics and Automation (ICRA), pp.3212–3217, 2009.
- [109] F. Tombari, S. Salti, and L.D. Stefano, “Unique signatures of histograms for local surface description,” In Proceedings of 11th European Conference on Computer Vision (ECCV), Part III, Lecture Notes in Computer Science, vol.6313, pp.356–369, 2010.
- [110] G. Elbaz, T. Avraham, and A. Fischer, “3D point cloud registration for localization using a deep neural network auto-encoder,” In Proceedings of 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.4631–4640, 2017.
- [111] K. Choi, J.K. Suhr, and H.G. Jung, “Map-matching-based cascade landmark detection and vehicle localization,” *IEEE Access*, vol.7, pp.127874–127894, 2019.
- [112] H. Li, F. Nashashibi, and G. Toulminet, “Localization for intelligent vehicle by fusing mono-camera, low-cost GPS and map data,” In Proceedings of 13th IEEE International Conference on Intelligent Transportation Systems (ITSC), pp.1657–1662, 2010.
- [113] T. Wu and A. Ranganathan, “Vehicle localization using road markings,” In Proceedings of 2013 IEEE Intelligent Vehicles Symposium (IV), pp.1185–1190, 2013.
- [114] R. Spangenberg, D. Goehring, and R. Rojas, “Pole-based localization for autonomous vehicles in urban scenarios,” In Proceedings of 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp.2161–2166, 2016.
- [115] C. Linegar, W. Churchill, and P. Newman, “Made to measure: Bespoke landmarks for 24-hour, all-weather localisation with a camera,” In Proceedings

- 
- of 2016 IEEE International Conference on Robotics and Automation (ICRA), pp.787–794, 2016.
- [116] C. McManus, B. Upcroft, and P. Newmann, “Scene signatures: Localised and point-less features for localisation,” In Proceedings of 10th Robotics: Science and Systems Conference (RSS), 9p., 2014.
  - [117] N. Sünderhauf, S. Shirazi, A. Jacobson, F. Dayoub, E. Pepperell, B. Upcroft, and M. Milford, “Place recognition with convnet landmarks: Viewpoint-robust, condition-robust, training-free,” In Proceedings of 11th Robotics: Science and Systems Conference (RSS), 10p., 2015.
  - [118] H. Lategahn, A. Geiger, and B. Kitt, “Visual SLAM for autonomous ground vehicles,” In Proceedings of 2011 IEEE International Conference on Robotics and Automation (ICRA), pp.1732–1737, 2011.
  - [119] Y. Zhao, S. Xu, S. Bu, H. Jiang, and P. Han, “GSLAM: A general SLAM framework and benchmark,” In Proceedings of 17th IEEE International Conference on Computer Vision (ICCV), pp.1110–1120, 2019.
  - [120] R. Wang, M. Schworer, and D. Cremers, “Stereo DSO: Large-scale direct sparse visual odometry with stereo cameras,” In Proceedings of 16th IEEE International Conference on Computer Vision (ICCV), pp.3903–3911, 2017.
  - [121] H. Zhan, R. Garg, W.C.S. Weerasekera, K. Li, H. Agarwal, and I. Reid, “Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction,” In Proceedings of 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.340–349, 2018.
  - [122] P. Biber and W. Straßer, “The normal distributions transform: A new approach to laser scan matching,” In Proceedings of 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp.2743–2748, 2003.
  - [123] E.B. Olson, “Real-time correlative scan matching,” In Proceedings of 2009 IEEE International Conference on Robotics and Automation (ICRA), pp.4387–4393, 2009.

- [124] M. Bosse and R. Zlot, “Continuous 3D scan-matching with a spinning 2D laser,” In Proceedings of 2009 IEEE International Conference on Robotics and Automation (ICRA), pp.4312–4319, 2009.
- [125] S. Rusinkiewicz and M. Levoy, “Efficient variants of the ICP algorithm,” In Proceedings of 3rd International Conference on 3-D Digital Imaging and Modeling (3DIM), pp.145–152, 2001.
- [126] A. Segal, D. Haehnel, and S. Thrun, “Generalized-ICP,” In Proceedings of 5th Robotics: Science and Systems Conference (RSS), 8p., 2009.
- [127] R.W. Wolcott and R.M. Eustice, “Fast LIDAR localization using multiresolution Gaussian mixture maps,” In Proceedings of 2015 IEEE International Conference on Robotics and Automation (ICRA), pp.2814–2821, 2015.
- [128] L. Li, M. Yang, C. Wang, and B. Wang, “Road DNA based localization for autonomous vehicles,” In Proceedings of 2016 IEEE Intelligent Vehicles Symposium (IV), pp.883–888, 2016.
- [129] J. Zhang and S. Singh, “LOAM: LiDAR odometry and mapping in real-time,” In Proceedings of 10th Robotics: Science and Systems Conference (RSS), 9p., 2014.
- [130] T. Hodan, F. Michel, E. Brachmann, W. Kehl, A. GlentBuch, D. Kraft, B. Drost, J. Vidal, S. Ihrke, X. Zabulis, C. Sahin, F. Manhardt, F. Tombari, T.K. Kim, J. Matas, and C. Rother, “BOP: Benchmark for 6D object pose estimation,” In Proceedings of 15th European Conference on Computer Vision (ECCV), Part X, Lecture Notes in Computer Science, vol.11214, pp.19–35, 2018.
- [131] M. Sundermeyer, Z.C. Marton, M. Durner, M. Brucker, and R. Triebel, “Implicit 3D orientation learning for 6D object detection from RGB images,” In Proceedings of 15th European Conference on Computer Vision (ECCV), Part VI, Lecture Notes in Computer Science, vol.11210, pp.712–729, 2018.
- [132] U. Kart, A. Lukezic, M. Kristan, J.K. Kamarainen, and J. Matas, “Object tracking by reconstruction with view-specific discriminative correlation filters,” In

- 
- Proceedings of 2019 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.1339–1348, 2019.
- [133] T. Groueix, M. Fisher, V.G. Kim, B.C. Russell, and M. Aubry, “A papier-mâché approach to learning 3D surface generation,” In Proceedings of 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.216–224, 2018.
- [134] G. Pandey, J.R. McBride, and R.M. Eustice, “Ford Campus vision and LiDAR data set,” *International Journal of Robotics Research (IJRR)*, vol.30, no.13, pp.1543–1552, 2011.
- [135] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? The KITTI vision benchmark suite,” In Proceedings of 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.3354–3361, 2012.
- [136] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, “Vision meets robotics: The KITTI dataset,” *International Journal of Computer Vision (IJCV)*, vol.32, no.11, pp.1231–1237, 2013.
- [137] J.L. Blanco-Claraco, F.A. Moreno-Dueñas, and J. González-Jiménez, “The Málaga urban dataset: High-rate stereo and LiDAR in a realistic urban scenario,” *International Journal of Robotics Research (IJRR)*, vol.33, no.2, pp.207–214, 2014.
- [138] N. Carlevaris-Bianco, A.K. Ushani, and R.M. Eustice, “University of Michigan north campus long-term vision and LiDAR dataset,” *International Journal of Robotics Research (IJRR)*, vol.35, no.9, pp.1023–1035, 2016.
- [139] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, “1 year, 1000 km: The Oxford RobotCar dataset,” *International Journal of Robotics Research (IJRR)*, vol.36, no.1, pp.3–15, 2017.
- [140] G. Mátyus, S. Wang, S. Fidler, and R. Urtasun, “HD Maps: Fine-grained road segmentation by parsing ground and aerial images,” In Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR),

- pp.3611–3619, 2016.
- [141] B. Mathibela, P. Newman, and I. Posner, “Reading the road: Road marking classification and interpretation,” *IEEE Transactions on Intelligent Transportation Systems (ITS)*, vol.16, no.4, pp.2072–2081, 2015.
- [142] A.B. Hillel, R. Lerner, D. Levi, and G. Raz, “Recent progress in road and lane detection: A survey,” *Journal of Machine Vision and Applications (MVA)*, vol.25, no.3, pp.727–745, 2014.
- [143] M. Cheng, H. Zhang, C. Wang, and J. Li, “Extraction and classification of road markings using mobile laser scanning point clouds,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing (J-STARS)*, vol.10, no.3, pp.1182–1196, 2017.
- [144] Y. Yu, J. Li, H. Guan, F. Jia, and C. Wang, “Learning hierarchical features for automated extraction of road markings from 3-D mobile LiDAR point clouds,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing (J-STARS)*, vol.8, no.2, pp.709–726, 2015.
- [145] S. Gargoum and K. E.-Basyouny, “Automated extraction of road features using LiDAR data: A review of LiDAR applications in transportation,” In *Proceedings of 4th International Conference on Transportation Information and Safety (ICTIS)*, pp.563–574, 2017.
- [146] J.M. Morel and G. Yu, “ASIFT: A new framework for fully affine invariant image comparison,” *SIAM Journal on Imaging Sciences*, vol.2, no.2, pp.438–469, 2009.
- [147] Z. Wang, Q. Zhao, D. Chu, F. Zhao, and L.J. Guibas, “Select informative features for recognition,” In *Proceedings of 18th IEEE International Conference on Image Processing (ICIP)*, pp.2477–2480, 2011.
- [148] A. Mikulík, M. Perdoch, O. Chum, and J. Matas, “Learning a fine vocabulary,” In *Proceedings of 11th European Conference on Computer Vision (ECCV)*, Part III, *Lecture Notes in Computer Science*, vol.6313, pp.1–14, 2010.

- 
- [149] D. Chen, S. Tsai, V. Chandrasekhar, G. Takacs, J. Singh, and B. Girod, “Robust image retrieval using multiview scalable vocabulary trees,” In Proceedings of 2009 IEEE International Conference on Visual Communications and Image Processing (VCIP), pp.72570V–1–72570V–9, 2009.
- [150] M. Shi, Y. Avrithis, and H. Jégou, “Early burst detection for memory-efficient image retrieval,” In Proceedings of 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.605–613, 2015.
- [151] D. Gálvez-López and J.D. Tardós, “Real-time loop detection with bags of binary words,” In Proceedings of 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp.51–58, 2011.
- [152] W. Zhou, Y. Lu, H. Li, and Q. Tian, “Scalar quantization for large scale image search,” In Proceedings of 20th ACM International Conference on Multimedia, pp.169–178, 2012.
- [153] Y. Uchida, S. Sakazawa, and S. Satoh, “Adaptive substring extraction and modified local NBNN scoring for binary feature-based local mobile visual search without false positives,” *ITE Transactions on Media Technology and Applications (MTA)*, vol.5, no.1, pp.24–34, 2017.
- [154] S. McCann and D.G. Lowe, “Local naive Bayes nearest neighbor for image classification,” In Proceedings of 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.3650–3656, 2012.
- [155] S.S. Tsai, D.M. Chen, G. Takacs, V. Chandrasekhar, R. Vedantham, R. Grzeszczuk, and B. Girod, “Fast geometric re-ranking for image based retrieval,” In Proceedings of 17th IEEE International Conference on Image Processing (ICIP), pp.1029–1032, 2010.
- [156] H. Jégou, D. Matthijs, and S. Cordelia, “On the burstiness of visual elements,” In Proceedings of 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), pp.1169–1176, 2009.
- [157] U. Feige, “A threshold of  $\ln n$  for approximating set cover,” *Journal of the ACM*

- (JACM), vol.45, no.4, pp.634–652, 1998.
- [158] D.P. Williamson and D.B. Shmoys, *The design of approximation algorithms*, Cambridge University Press, Cambridge, 2011.
- [159] D.S. Hochbaum, “Approximating covering and packing problems: Set cover, vertex cover, independent set, and related problems,” in *Approximation Algorithms for NP-Hard Problem*, pp.94–143, PWS Publishing, Boston, 1997.
- [160] M.A. Guvensan and A.G. Yavuz, “On coverage issues in directional sensor networks: A survey,” *Ad Hoc Networks*, vol.9, no.7, pp.1238–1255, 2011.
- [161] E. Filatova and V. Hatzivassiloglou, “A formal model for information selection in multi-sentence text extraction,” In *Proceedings of 20th International Conference on Computational Linguistics (COLING)*, pp.397–403, 2004.
- [162] S. Hinterstoisser, V. Lepetit, S. Benhimane, P. Fua, and N. Navab, “Learning real-time perspective patch rectification,” *International Journal of Computer Vision (IJCV)*, vol.91, no.1, pp.107–130, 2011.
- [163] D. Kurz, O. Thomas, and B. Selim, “Representative feature descriptor sets for robust handheld camera localization,” In *Proceedings of 11th International Symposium on Mixed and Augmented Reality (ISMAR)*, pp.65–70, 2012.
- [164] I. Şahin and O. Kemal, “A comparative evaluation of well-known feature detectors and descriptors,” *International Journal of Applied Mathematics, Electronics and Computers (IJAMEC)*, vol.3, no.1, pp.1–6, 2014.
- [165] K. Mikolajczyk and C. Schmid, “Scale & affine invariant interest point detectors,” *International Journal of Computer Vision (IJCV)*, vol.60, no.1, pp.63–86, 2004.
- [166] Y. Linde, A. Buzo, and R.M. Gray, “An algorithm for vector quantizer design,” *IEEE Transactions on Communications*, vol.28, no.1, pp.84–95, 1980.
- [167] R.M. Gray, “Vector quantization,” *IEEE Acoustics, Speech and Signal Processing Magazine*, vol.1, no.2, pp.4–29, 1984.



- 
- [168] P.C. Cosman, K.L. Oehler, E.A. Riskin, and R.M. Gray, "Using vector quantization for image processing," *Proceedings of the IEEE*, vol.81, no.9, pp.1326–1341, 1993.
- [169] H.B. Kekre and T.K. Sarode, "Speech data compression using vector quantization," *WASET International Journal of Computer and Information Science and Engineering (IJCISE)*, vol.2, no.4, pp.251–254, 2008.
- [170] D. Meagher, "Geometric modeling using octree encoding," *Computer Graphics and Image Processing*, vol.19, no.2, pp.129–147, 1982.
- [171] 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.
- [172] M. Steinbach, L. Ertöz, and V. Kumar, "The challenges of clustering high dimensional data," in *New Directions in Statistical Physics*, pp.273–309, Springer, Berlin, Heidelberg, 2004.
- [173] B. Yang, X. Fu, N. Sidiropoulos, and M. Hong, "Towards K-means-friendly spaces: Simultaneous deep learning and clustering," In *Proceedings of 34th International Conference on Machine Learning (ICML)*, pp.3861–3870, 2017.
- [174] 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.
- [175] T. Zhang, P. Ji, M. Harandi, W. Huang, and H. Li, "Neural collaborative subspace clustering," In *Proceedings of 36th International Conference on Machine Learning (ICML)*, pp.7384–7393, 2019.
- [176] X. Yang, C. Deng, F. Zheng, J. Yan, and W. Liu, "Deep spectral clustering using dual autoencoder network," In *Proceedings of 2019 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.4066–4075, 2019.
- [177] A. Schindler, "Vehicle self-localization with high-precision digital maps," In

- Proceedings of 2013 IEEE Intelligent Vehicles Symposium (IV), pp.141–146, 2013.
- [178] M. Lundgren, E. Stenborg, L. Svensson, and L. Hammarstrand, “Vehicle self-localization using off-the-shelf sensors and a detailed map,” In Proceedings of 2014 IEEE Intelligent Vehicles Symposium (IV), pp.522–528, 2014.
- [179] J.Y. Sim, C.S. Kim, and S.U. Lee, “Lossless compression of 3-D point data in QSplat representation,” IEEE Transactions on Multimedia, vol.7, no.6, pp.1191–1195, 2005.
- [180] S. Rusinkiewicz and M. Levoy, “QSplat: A multiresolution point rendering system for large meshes,” In Proceedings of 27th ACM Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH), pp.343–352, 2000.
- [181] J.Y. Sim and S.U. Lee, “Compression of 3-D point visual data using vector quantization and rate-distortion optimization,” IEEE Transactions on Multimedia, vol.10, no.3, pp.305–315, 2008.
- [182] T. Golla and R. Klein, “Real-time point cloud compression,” In Proceedings of 2015 IEEE International Conference on Robotics and Automation (ICRA), pp.5087–5092, 2015.
- [183] T. Stoyanov, M. Magnusson, H. Almqvist, and A.J. Lilienthal, “On the accuracy of the 3D normal distributions transform as a tool for spatial representation,” In Proceedings of 2011 IEEE International Conference on Robotics and Automation (ICRA), pp.4080–4085, 2011.
- [184] Point Cloud Library, “Downsampling a pointcloud using a voxelgrid filter.” [https://pcl.readthedocs.io/en/latest/voxel\\_grid.html](https://pcl.readthedocs.io/en/latest/voxel_grid.html) [2020/12/1 Accessed].
- [185] 国土交通省, “道路統計年報 2016”, 2016. <https://www.mlit.go.jp/road/ir/ir-data/tokei-nen/2016/nenpo02.html> [2020/12/1 アクセス] .

# 研究業績

## 本論文に関連する研究業績

### 学術雑誌

1. 松崎康平, 田坂和之, 川西康友, 村瀬洋, “3D NDT 法による自己位置推定のための環境地図の圧縮,” 電子情報通信学会論文誌 (D), vol. J103-D, no. 6, pp. 529–541, 2020.
2. K. Matsuzaki, K. Tasaka, and H. Yanagihara, “Local Feature Reliability Measure Consistent with Match Conditions for Mobile Visual Search,” IEICE Transactions on Information and Systems, vol. E101-D, no. 12, pp. 3170–3180, 2018.
3. 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,” 電子情報通信学会論文誌 (D), vol. J101-D, no. 11, pp. 1471–1481, 2018.

### 国際会議

1. K. Matsuzaki and H. Yanagihara, “A Compact Map Representation for Large-scale Environments and Localization Method based on Similarity Measure,” In Proceedings of 2018 IEEE Intelligent Vehicles Symposium (IV), pp. 674–681, 2018.
2. K. Matsuzaki, Y. Uchida, S. Sakazawa, and S. Satoh, “Local Feature Reliability Measure using Multiview Synthetic Images for Mobile Visual Search,” In Proceedings of 3rd IAPR Asian Conference on Pattern Recognition (ACPR), pp. 156–160, 2015.

## 国内会議

1. 松崎康平, 田坂和之, 川西康友, 村瀬洋, “自己位置推定のための正規分布地図の圧縮に関する一検討,” 2019 年映像情報メディア学会冬季大会, 13B-1, 2019.
2. 松崎康平, 柳原広昌, “3 次元点群を用いた類似検索に基づく自己位置推定手法,” 2017 年映像情報メディア学会年次大会, 33B-3, 2017.
3. 松崎康平, 内田祐介, 酒澤茂之, “モバイル端末を用いた画像検索のためのデータベース作成手法,” 2015 年電子情報通信学会総合大会, D-12-42, 2015.
4. 松崎康平, 酒澤茂之, “モバイル端末を用いた画像検索のための特徴点選択手法に関する一検討,” 第 13 回情報科学技術フォーラム (FIT), H-031, 2014.

## その他の研究業績

### 学術雑誌

1. 小林亜令, 松本正明, 内田祐介, 土井渉, 松崎康平, 加藤晴久, “端末とサーバによるハイブリッド大規模画像検索システムとその応用,” 映像情報メディア学会誌, vol. 69, no. 1, pp. J11–J16, 2015.

## 国際会議

1. K. Matsuzaki and K. Tasaka, “Representation Learning via Parallel Subset Reconstruction for 3D Point Cloud Generation,” In Proceedings of 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 289–296, 2019.
2. K. Matsuzaki and K. Tasaka, “Binary Representation for 3D Point Cloud Compression Based on Deep Auto-encoder,” In Proceedings of 8th IEEE Global Conference on Consumer Electronics (GCCE), pp. 498–499, 2019.
3. K. Matsuzaki Y. Uchida, S. Sakazawa, and S. Satoh, “Geometric Verification Using Semi-2D Constraints for 3D Object Retrieval,” In Proceedings of 23rd IAPR International Conference on Pattern Recognition (ICPR), pp. 2339–

2344, 2016.

4. K. Matsuzaki and T. Aoki, “A Study on Light Source Estimation Based on Diffusion Reflection of Human Faces,” Tohoku University Global COE Program Student Organizing International Mini-Conference on Information Electronics Systems (SOIM-GCOE), F7P-41, 2010.

## 国内会議

1. 松崎康平, 河村圭, 川西康友, 村瀬洋, “3次元形状特徴の位置合わせに基づくカメラの相対姿勢推定,” 2020年映像情報メディア学会創立70周年記念大会, 21C-4, 2020.
2. 松崎康平, 河村圭, 川西康友, 村瀬洋, “3次元点群の復元とレジストレーションに基づくカメラの相対姿勢推定に関する検討,” 第23回画像の認識・理解シンポジウム (MIRU), IS3-2-20, 2020.
3. 松崎康平, 田坂和之, “3次元点群生成のためのマルチタスク学習を用いた表現学習,” 第18回情報科学技術フォーラム (FIT), CH-003, 2019.
4. 松崎康平, 内田祐介, 酒澤茂之, “エピポーラ幾何に基づく画像間の幾何検証の高精度化に関する一検討,” 2016年映像情報メディア学会年次大会, 21C-1, 2016.
5. K. Matsuzaki, Y. Uchida, and S. Sakazawa, “Geometric Verification Using Semi-2D Constraints,” 第19回画像の認識・理解シンポジウム (MIRU), PS1-11, 2016.
6. 松崎康平, 内田祐介, 酒澤茂之, “特定物体認識のため見え変化を考慮した代表ベクトルの複数割り当て手法,” 2014年映像情報メディア学会冬季大会, 10-7, 2014.
7. 松崎康平, 内田祐介, 酒澤茂之, “画像検索における幾何検証のためのデータ量削減に関する一検討,” 2014年電子情報通信学会総合大会, D-12-39, 2014.
8. 松崎康平, 内田祐介, 酒澤茂之, “バイナリ局所特徴を用いた画像検索の精度改善,” 2013年映像情報メディア学会冬季大会, 1-3, 2013.
9. 松崎康平, 内田祐介, 酒澤茂之, “局所特徴の方向に着目した画像間幾何検証の高速化,” 2013年映像情報メディア学会年次大会, 6-9, 2013.
10. 松崎康平, 青木輝勝, “人物被写体を対象とした光源推定方式に関する一検討,” 映

像情報メディア学会技術報告, ME2012-69, 2012.

11. 松崎康平, 青木輝勝, “人物画像を対象とした三次元顔モデルを用いた光源推定方式に関する一検討,” 映像情報メディア学会技術報告, ME2011-133, 2011.
12. 松崎康平, 青木輝勝, “顔画像からの光源推定方式に関する一検討,” 電子情報通信学会技術研究報告, EMM2011-40, 2011.
13. 松崎康平, 青木輝勝, “人間の顔に着目した単一画像からの光源推定に関する一検討,” 2010 年映像情報メディア学会年次大会, 13-5, 2010.
14. 松崎康平, 青木輝勝, 沼澤潤二, “顔情報に着目した光源推定の一検討,” 2010 年電子情報通信学会総合大会, D-12-47, 2010.

## 表彰・受賞

1. シニア会員, 電子情報通信学会, 2020.
2. 論文賞, 電子情報通信学会, 2020.
3. 鈴木記念奨励賞, 映像情報メディア学会, 2018.
4. 学術奨励賞, 電子情報通信学会, 2016.
5. FIT 奨励賞, 情報処理学会, 2014.
6. 技術振興賞コンテンツ技術賞, 映像情報メディア学会, 2014.