



局所特徴量を用いた特定物体認識に関する研究

メタデータ	言語: jpn 出版者: 公開日: 2012-11-21 キーワード (Ja): キーワード (En): 作成者: 井上, 勝文 メールアドレス: 所属:
URL	https://doi.org/10.24729/00000068

大阪府立大学博士論文

局所特徴量を用いた特定物体認識に関する研究

2012年2月

井上 勝文

目次

第1章 緒論	1
第2章 局所特徴量の取捨選択によるメモリ使用量削減法	7
2.1 緒言	7
2.2 関連研究	8
2.3 提案手法	10
2.3.1 考え方	11
2.3.2 野口らの手法	12
2.3.3 物体認識シミュレーション	17
2.3.4 プラス関与性とその性質	17
2.3.5 登録処理の流れと前提条件	18
2.3.6 登録のための評価基準1	20
2.3.7 登録のための評価基準2	22
2.4 実験	23
2.4.1 実験条件	24
2.4.2 実験結果	25
2.4.3 考察	27
2.5 結言	28
第3章 Bloomier Filter を用いたメモリ使用量削減法	29
3.1 緒言	29
3.2 従来手法	31
3.2.1 従来研究の主な取り組み	31
3.2.2 木構造に基づくメモリ使用量削減法	32
3.2.3 ハッシュ表に基づくメモリ使用量削減法	33

3.3	提案手法	36
3.3.1	Bloom Filter	36
3.3.2	Bloomier Filter	38
3.3.3	符号の定義	39
3.3.4	符号化関数の作成	39
3.3.5	パリティビットに基づく誤り検出用 Bloomier Filter の作成	41
3.3.6	認識処理	42
3.4	1万個の二次元物体を用いた実験	44
3.4.1	実験条件	44
3.4.2	従来手法との性能比較	45
3.4.3	誤り検出用 Bloomier Filter の有効性に関する実験	49
3.4.4	偽陽性の発生を容認する <i>bit ID</i> の桁数 <i>t</i> に関する実験	50
3.4.5	ビット反転の操作を行う次元数 <i>b</i> に関する実験	51
3.5	55個の三次元物体を用いた実験	56
3.5.1	実験条件	56
3.5.2	従来手法との性能比較	57
3.6	結言	60
第4章	データセットの違いが物体認識手法に与える影響の解析	61
4.1	緒言	61
4.2	物体データセットと物体に関する性質	62
4.2.1	一般物体認識のための物体データベース	63
4.2.2	特定物体認識のための物体データベース	64
4.2.3	1002個の三次元物体	66
4.3	物体データセットの構成の違いに関する実験	70
4.3.1	実験条件	70
4.3.2	COIL-100を用いた実験	71
4.3.3	1002個の三次元物体を用いた実験	73
4.4	物体データセットの物体数の違いに関する実験	75
4.4.1	実験条件	76
4.4.2	実験結果	76

4.5 結言	79
第5章 結論	81
謝辞	85
参考文献	87
付録	97
付録 A 第 2.4 節の実験で用いた三次元物体	97
付録 B 第 3.5 節の実験で用いた三次元物体	99
付録 C 第 4.2.3 節で述べた 1002 個三次元物体データセット	101

目 次

2.1	局所特徴量のクラスタリングによる Visual Word の作成	8
2.2	投票処理に基づく特定物体認識手法	9
2.3	局所特徴量の取捨選択に基づくメモリ使用量削減法の処理の流れ	12
2.4	リスト構造による物体 ID の登録	14
2.5	特徴ベクトルの変動への対処	15
2.6	多段階化による認識	16
2.7	複数の局所特徴量とプラス関与する局所特徴量	18
2.8	物体モデル登録処理の流れ	19
2.9	カレントモデルを用いた場合の正しい投票数	21
2.10	近傍点との距離計算	22
2.11	55 物体の例	24
2.12	COIL-100 のフレーム画像の例	25
2.13	実験結果 (55 物体)	26
2.14	実験結果 (COIL-100)	27
3.1	階層的 k -means クラスタリングによる Vocabulary Tree の作成	32
3.2	ハッシュ表へのデータ登録 (ハッシュ(距離計算なし))	34
3.3	Bloom Filter の処理の流れ	37
3.4	偽陽性の発生確率 $P_{fp}(k, m, n)$	38
3.5	Bloomier Filter への登録方法	40
3.6	誤り検出用 Bloomier Filter の作成方法の略図	41
3.7	誤り検出用 Bloomier Filter によるスコア更新 ID の選定 . はその Bloom Filter にクエリベクトルが登録されていることを意味し, \times は登録されて いないことを意味する	43
3.8	登録画像の例	45

3.9	質問質問の例	46
3.10	誤り検出用 Bloomier Filter の有無による性能比較	49
3.11	偽陽性の発生を容認する <i>bit ID</i> の桁数 t と認識率の関係	51
3.12	再現率と適合率の関係	52
3.13	再現率と認識率の関係	53
3.14	適合率と認識率の関係	53
3.15	b を増加させることで認識できるようになった検索質問画像の例	54
3.16	b を増加させることで認識できなくなった検索質問画像の例	54
3.17	b を増加させても全く認識できなかった検索質問画像の例	55
3.18	55 個の三次元物体の例	56
3.19	提案手法で認識できなかった検索質問画像	59
4.1	三次元物体の画像例．三次元物体は，撮影される角度が変化すると，物体の見え方が大きく変化する．	62
4.2	Caltech-256 の画像例	63
4.3	COIL-100 の画像例	64
4.4	ETH-80 の画像例	64
4.5	物体構造の複雑さの判断基準．上段に物体の例，下段にその物体をターンテーブルで回転させた際にできるシルエットを示す．これらの形に近似できる物体を物体構造の単純な物体とする．	65
4.6	1002 個の三次元物体の例	68
4.7	見た目が類似する三次元物体の例	69
4.8	両手法で誤認識した検索質問画像の例 (COIL-100)	72
4.9	COIL-100 において，見た目が類似する物体の検索質問画像．左側二枚と右側二枚の検索質問画像は，それぞれ BF 法と Hash 法で誤認識した検索質問画像である．	73
4.10	両手法で誤認識した検索質問画像の例 (DB1002)	74
4.11	DB1002 において，見た目が類似する物体の検索質問画像．左側二枚と右側二枚の検索質問画像は，それぞれ BF 法と Hash 法で誤認識した検索質問画像である．	75
4.12	物体データベースの規模よらず両手法で誤認識した検索質問画像の例	77

4.13 物体数に対するメモリ使用量の推定値	79
A.1 第 2.4 節の実験で用いた三次元物体 (1 ~ 30)	97
A.2 第 2.4 節の実験で用いた三次元物体 (31 ~ 55)	98
B.1 第 3.5 節の実験で用いた三次元物体 (1 ~ 30)	99
B.2 第 3.5 節の実験で用いた三次元物体 (31 ~ 55)	100
C.1 第 4.2.3 節で述べた 1002 個三次元物体データセット	101

表 目 次

3.1	実験に用いた V-Tree(Vocabulary Tree) のパラメータ (1 万個の二次元物体)	47
3.2	各手法のパラメータ (1 万個の二次元物体)	47
3.3	各手法の性能評価 (1 万個の二次元物体)	48
3.4	誤り検出用 Bloomier Filter の有無による検索質問画像の正誤枚数	50
3.5	物体 ID の平均検索回数	55
3.6	実験に用いた V-Tree(Vocabulary Tree) のパラメータ (55 個の三次元物体)	57
3.7	各手法のパラメータ (55 個の三次元物体)	58
3.8	各手法の性能評価 (55 個の三次元物体)	58
4.1	各物体属性に属する物体の数 (COIL-100)	66
4.2	各物体属性に属する物体の数 (1002 個の三次元物体)	66
4.3	両手法の最大認識率 (COIL-100)	71
4.4	物体属性ごとの誤認識率 (COIL-100)	71
4.5	両手法の最大認識率 (DB1002)	73
4.6	物体属性ごとの誤認識率 (DB1002)	74
4.7	物体データベースの規模に対する認識率・処理時間・メモリ使用量の関係 . 両手法において最も認識率の高かった結果を示す .	76
4.8	BF 法において , 最高認識率を達成するパラメータ b の平均値	78

第1章 緒論

近年，デジタルカメラやスマートフォン等の携帯端末が老若男女を問わずに，多くの人々に普及してきている．また，このような携帯端末機器の高性能化が著しく，特にこれらに付与されているカメラ機能の性能には目を見張るものがある．例えば，現在開発されている携帯端末のほとんどには，顔検出機能^[1]や顔認識機能^{[2],[3]}が付与されており，中には表情認識機能^[4]が付いたものもある．このような撮影時のサービスに加え，近年これらの携帯端末機器を用いて撮影された画像を入力キーとした，新しい情報検索サービスが開始されつつある．このようなサービスの一つに，ユーザがある物体に関して十分な知識を有していなくも，ユーザがその物体を携帯端末機器を用いて撮影することにより，インターネットから有益な情報を入手できるものがある．代表的なサービス例として，“Google goggles”^[5]が知られている．このようなサービスを提供する場合，例えばランドマークといった物体の認識^{[6]-[8]}では，GPS^{[9]-[11]}やジオタグ^{[12],[13]}を利用することなどが考えられるが，認識対象物体を限定せずにサービスを提供するには，撮影対象物体が何であるかを認識する物体認識技術が必須となる．

物体認識は，画像中に存在する物体のカテゴリを認識する一般物体認識^{[14],[15]}と物体のインスタンスを認識する特定物体認識^{[16]-[18]}に大別される．前者は，椅子，自動車などの物体のクラスを結果として返すのに対して，後者は例えば自動車の特定のモデルなど，インスタンスを結果として返すものである．本論文では，上述のようなある特定の物体の情報を得るサービスを実現するために，後者の特定物体認識に焦点を当てて話を進める．その中でも特に，SIFT(Scale-Invariant Feature Transform)^{[19],[20]}やPCA-SIFT^[21]といった局所特徴量^{[22]-[27]}を用いる特定物体認識手法に着目する．この手法は，高い認識精度を得ることができるため，近年コンピュータビジョンやパターン認識の分野で盛んに研究されているものである．

我々の周囲には無数の物体インスタンスが存在するため，特定物体認識技術を用いたサービスを真に実用的なものにするには，識別できる物体数の大規模化が必須である．SIFT

等の局所特徴量は，物体の各部位（局所領域^{[19],[28]-[31]}）を数十から数百の次元を持つ実数値ベクトルで表現するものであり，高い識別性を有しているため，多数の物体を識別する目的に適している．従来の局所特徴量を用いた物体認識手法では，局所特徴量を組み合わせて「物体モデル」を作成する．そして，作成された各物体の物体モデルと検索質問画像とを照合することによって，最も類似度の高い物体を認識結果とする．これまでに様々な物体モデル作成方法が提案されている．例えば，様々な角度から撮影された物体の画像から抽出された特徴ベクトルの位置関係をもとに，物体の三次元形状を復元し，これを認識に用いるもの^{[32]-[35]}がある．以後，三次元形状を復元した物体モデルを「三次元物体モデル」と呼ぶ．この手法では，三次元物体モデルを所持しているため，高精度に認識できるが，三次元物体モデルの作成に時間がかかるという問題がある．これに対して，三次元物体モデルを用いずに，画像から抽出した特徴ベクトルを物体モデルとして未知の画像と照合するもの^{[36]-[39]}がある．この手法では，物体を撮影した画像があれば，特徴ベクトルを抽出するのみで物体モデルを構築できるという利点がある．本研究では，上記利点から後者のスタンスをとる．

三次元物体モデルを構築しない手法では，未知の物体から得られる全特徴ベクトルと，既知の物体から得られる特徴ベクトルとを照合し，その結果を統合することで物体を認識する．特徴ベクトルを照合する最も単純で実装の容易な方法は，最近傍探索^{[40],[41]}である．最近傍探索とは，ある距離尺度において，既知の特徴ベクトルの中から，クエリとなる特徴ベクトルと最も距離の近いものを照合結果とする方法である．文献^[40]と文献^[41]では，距離尺度として，特徴ベクトルの分布を考慮する必要がなく，計算の容易なユークリッド距離が用いられている．また，最近傍探索には，照合の容易さに加え，対応する既知の特徴ベクトルを一意に決定できる利点もある．しかしこの照合方法には，最近傍の特徴ベクトルが，必ずしも同じ物体の同じ箇所から抽出されたものとは限らないという問題点がある．このため，物体を正しく認識するには，上記のような照合誤りを考慮して，全ての特徴ベクトルの照合結果を統合する必要がある．この問題を解決する最も単純な統合方法に，投票処理^{[16],[42]}がある．この統合方法では，あらかじめ既知の特徴ベクトルにどの物体から抽出されたかを示す物体 ID を付与しておき，検索質問画像から得られた特徴ベクトルとの照合で，対応する物体に投票する．そして，最も得票数の多い物体を統合結果とする．上記の枠組みを用いると，いくつかの特徴ベクトルの照合に失敗しても，正解物体と他の物体の得票数が逆転しなければ，物体を正しく認識できる．

本論文で提案する特定物体手法は全て上記の概念に基づく。この手法は、単純で高い認識精度を得ることができるが、大規模な物体認識を行うためには、まだいくつかの障害を乗り越える必要がある。考えられる最も大きな障害は、以下の三つである。

一つ目の障害は、処理時間である。一般的に、特徴ベクトルは一枚の画像より数百から数千個抽出されるため、認識対象物体が増加すると、特徴ベクトル数が膨大となる。特徴ベクトルの照合に最近傍探索を用いる手法では、最近傍となる特徴ベクトルを求めるのに、膨大な数の特徴ベクトルとの距離を計算しなければならない。このため、大規模な物体認識を行うには、特徴ベクトルの最近傍探索の効率化が必須となる。この問題を解決するアプローチには、個々の距離計算自体を高速化するものと、距離計算の対象を絞り込むものがある。特に後者のアプローチは、「近似最近傍探索」と呼ばれ、絞り込みを大胆に行うことで、照合処理にかかる時間を大幅に改善できる^[43]。しかし、この代償として、最近傍となる特徴ベクトルが、距離計算の対象から外れてしまい、求まらない可能性がある。このため、近似最近傍探索では、最近傍となる特徴ベクトルが距離計算の対象から外れないように特徴ベクトルを絞り込む必要がある。幸いこれまでに、この問題を解決する手法として、k-d木などの木構造に基づくもの^{[19],[44]-[46]}や、ハッシュ表に基づくもの^{[47]-[51]}などが多数提案されている。本研究では、処理時間の問題は、既存のハッシュ表に基づく近似最近傍探索を導入することにより対処する。

二つ目の障害は、メモリ使用量である。ここで、本論文では主記憶をメモリと表記する。また本論文では、携帯端末等に搭載されている主記憶の容量をメモリ容量、物体認識システムに必要なメモリ量をメモリ使用量と表記する。上述のように、認識対象物体が増加すると、特徴ベクトルの数が膨大となるため、最近傍探索を用いる手法では、これらを記憶するのに莫大なメモリが必要となる。これまでに、特徴ベクトルのベクトル量子化^{[52]-[54]}や、スカラ量子化^{[43],[55]}などのメモリ使用量削減法が提案されている。特徴ベクトルを量子化すると、異なる特徴ベクトルを一つの代表ベクトルでまとめて表現するためメモリ使用量を削減できる。しかし、量子化誤差が大きくなると、正しく対応付いていた特徴ベクトルが、誤ったものに対応付く可能性が高くなる。本研究では、上記のように量子化誤差によって、誤った特徴ベクトルに対応付く可能性が高くなることを、“特徴ベクトルの識別能力が低下する”という。特徴ベクトルを量子化する手法では、一つの代表ベクトルで表現する特徴ベクトルの数を増やせば増やすほど、メモリ使用量を削減できるが、同時に特徴ベクトルの識別能力も低下するため、高い物体認識率を維持すること

が難しい^[56]。このほか、主記憶ではなく補助記憶を利用するもの^{[57],[58]}もあるが、補助記憶へのアクセス時間がボトルネックとなり、認識処理に時間がかかるという問題点がある。このため、認識処理の時間を考慮すると、補助記憶よりも主記憶を用いることが望まれる。しかし、メモリ容量が数百 GB 以上の計算機は高価であるため、大規模物体認識の実現は容易ではない。このため、大規模な物体認識を安価に実現するには、物体認識システムに必要なメモリ使用量を削減する必要がある。本論文では、この問題に対して、以下の二つのアプローチに基づくメモリ使用量削減法を提案する。

第一のアプローチは、特徴ベクトルの取捨選択によるもの^{[59]-[61]}である。上述したように、投票処理に基づく特定物体認識手法では、最近傍探索によって特徴ベクトルを対応付ける。提案手法ではこの点に着目し、特徴ベクトルの物体認識に対する貢献度の違いを見つけ、この性質を利用して局所特徴量の保存に必要なメモリ使用量を削減する。具体的には、物体モデル構築用画像から得られた全ての特徴ベクトルを用いて物体認識シミュレーションを行い、どの物体の特徴ベクトルに対応付いたかを調べる。この情報をもとに、特徴ベクトルの物体認識に対する貢献度の違いを調べ、より貢献度の高いものから順にメモリ使用量が限度に到達するまで物体モデルへ登録する。但し、物体をどのような角度からでも認識できるようにするために、あらゆる角度から得られる特徴ベクトルを満遍なく登録する。

第二のアプローチは、特徴ベクトルの照合手法を距離計算に基づかないものに変更することである^[62]。具体的には、各特徴ベクトルに対して ID を割り振り、この ID が一致するかで照合する。本研究では、このような特徴ベクトルの照合方法を、特徴ベクトルの「一致検索」と呼ぶ。一致検索では、距離計算に必要な特徴ベクトル自体の記録は必要なく、特徴ベクトルの ID のみを記録すれば良いため、大幅にメモリ使用量を削減できる。この考えのもと、本論文では空間効率の良い Bloomier Filter^[63] と呼ばれる確率的データ構造を用いる手法を提案する^{[64]-[67]}。Bloomier Filter とは、登録される要素に関連する値を返す連想配列である。提案手法では、これを用いて特徴ベクトルを符号化し、この符号が一致するかどうかで特徴ベクトルを照合する。この手法では、小さなハッシュ表を用いることで、認識処理全体に必要なメモリ使用量も削減しつつ、複数のハッシュ関数を用いて特徴ベクトルの識別性を高めている。

三つ目の障害は、認識精度である。認識精度を低下させる要因は、物体の類似性である。認識対象物体が増加すると、それらの中に互いの見た目が酷似するものが含まれる可能性

が高くなる．これらの個体差を区別するには，データベースに登録される特徴ベクトルの数は多い方が良いことが知られている^{[52],[56]}．しかし，特徴ベクトルの登録数が増加すると，処理時間やメモリ使用量に影響する．このため，これらにできるだけ影響を与えずに多くの特徴ベクトルの情報を扱う必要がある．上述した提案手法のような特徴ベクトルの一致検索に基づく認識手法は，距離計算の必要がなく，特徴ベクトルの存在のみを記録すれば良いため，少ないメモリ使用量でリアルタイム認識が可能である^[62]．しかし，特に三次元物体を対象としたこれまでの研究^{[61],[64]–[67]}では，検証のために用いられている三次元物体データセットは小規模なものが多い．例えば，COIL-100^[68]等の一般的な三次元物体データセットには，100物体程度しか含まれておらず，物体数の違いが手法に与える影響を検証するには十分であるとは言い難い．このため，上述の影響を議論するには，より大規模な三次元物体データセットが必要である．そこで，本研究では独自に1002個の三次元物体データセットを作成し，物体数の違いが手法に与える影響について検証する^{[69]–[71]}．また，COIL-100に含まれる物体に着目すると，COIL-100には物体構造の単純なものが多く含まれている．しかし，一般的に物体数が増加すると，物体データセットの構成が構造の単純なものに偏るとは限らない．そこで，本研究では作成する1002個の三次元物体データセットの構成をCOIL-100と異なるものとし，物体データセットの構成の違いが手法に与える影響についても検証する^[69]．

ここで，本研究の位置づけについてまとめる．本研究では，局所特徴量を用いた大規模物体認識システムを実現するために障害となる三つの課題に対して，二つ目のメモリ使用量の課題と，三つ目の認識精度の課題に焦点を置く．特に二つ目の課題に対して本論文では，局所特徴量の取捨選択によるアプローチと，一致検索に基づくアプローチに基づく手法を提案する．また，認識精度の課題に対して，本論文では，データベースの登録物体数を変化させた実験より，物体数の違いが一致検索に基づく物体認識手法に対してどのような影響を与えるかを検証した結果について述べる．さらに，物体の類似性といった物体形状が一致検索に基づく認識手法に対してどのような影響を与えるのかを検証した結果についても述べる．但し，一つ目の処理時間の課題については，これまでに多数の解決策が提案されているため，本研究では重きを置かず，既存手法の概念を導入するのみにとどめる．つまり本研究では，既存手法の概念を導入して処理を効率化した物体認識システムを対象とする．本研究の目的は，この物体認識システムに対して，二つのアプローチによりメモリ使用量を削減することである．また，提案手法によってメモリ使用量を削減した物

体認識システムの認識精度を検証することである。

本論文の構成は以下の通りである。

第2章では、局所特徴量の取舍選択に基づくメモリ使用量削減法について詳しく説明する。提案手法のアプローチは、物体認識に有効な局所特徴量のみを物体モデルに登録することである。提案手法では、物体モデル構築用画像から得られる局所特徴量を用いて、物体認識シミュレーションを行い、局所特徴量の物体認識に対する貢献度の違いを調べる。そして、貢献度の高いものから順に物体モデルへ登録する。これにより、認識精度をできるだけ落とすことなく、物体モデルに必要なメモリ使用量を削減する。第2章では、局所特徴量の物体認識に対する貢献度や、物体認識シミュレーションについて詳しく述べる。また、第2章では、上記提案手法の有効性を確かめるために、55個の三次元物体データセット及びCOIL-100を用いた実験とその結果、考察について述べる。

第3章では、Bloomier Filterを用いたメモリ使用量削減法について詳しく説明する。Bloomier Filterを用いた提案手法は、特徴ベクトルの照合に一致検索を用いてメモリ使用量削減を実現するものである。Bloomier Filterとは、登録される要素に関連する値を返す連想配列である。提案手法では、これを用いて特徴ベクトルを符号化し、この符号が一致するかどうかで特徴ベクトルを照合する。第3章では、Bloomier Filterについて詳細を述べ、このFilterを用いたメモリ使用量削減法を提案する。また第3章では、提案手法の有効性を確かめるために、1万個の二次元物体データセットと55個の三次元物体データセットを用いた実験とその結果、考察について詳しく述べる。

第4章では、データセットの違いが、第3章で提案したような一致検索に基づく物体認識手法に対して与える影響について検証する。本論文では、物体データセットの違いを、物体数及び物体データセットの構成の違いと定義し、これらが物体認識手法に与える影響について議論する。第4章では、一般的に公開されている物体データセットに言及した後、上述の影響を検証するために独自に作成した1002個の三次元物体データセットの詳細について述べる。そして、COIL-100と作成したデータセットを用いて、物体数の違いや物体データセットの構成の違いが、一致検索に基づく物体認識手法に与える影響について検証した結果について述べる。

最後に第5章では、本研究で得られた結果を総括すると共に、今後の研究課題について述べる。

第2章 局所特徴量の取捨選択によるメモリ使用量削減法

2.1 緒言

局所特徴量を用いた特定物体認識では，一般的に局所特徴量の照合に最近傍探索^{[40],[41]}を用いるものが多い．このような特定物体認識手法には，認識対象物体が増加すると，局所特徴量の数が膨大となり，距離計算を行うための情報を記憶するのに莫大なメモリが必要となる問題がある．この問題に対して本章では，局所特徴量の取捨選択によるメモリ使用量削減法を提案する^{[59]-[61]}．提案手法の特徴は，物体モデル構築用画像から得られる局所特徴量を用いて，物体認識シミュレーションを行うことである．この物体認識シミュレーションから，局所特徴量の物体認識に対する貢献度の違いを見つけ，物体毎に貢献度の高いものから順に物体モデルに登録する．また，物体をどのような角度からでも認識できるようにするために，あらゆる角度から得られる特徴ベクトルを満遍なく物体モデルに登録する．これにより，認識精度をできるだけ落とすことなく，局所特徴量自体の保存に必要なメモリ使用量を削減することを目指す．

本章の構成は以下の通りである．まず，第 2.2 節にて，従来のメモリ使用量削減法について言及した後，第 2.3 節で，提案する局所特徴量の取捨選択によるメモリ使用量削減法について詳しく述べる．そして，55 個の三次元物体データセット及び，COIL-100^[68]を用いて行った実験より，第 2.4 節で，登録する局所特徴量の数と認識率，メモリ使用量の関係について検証する．最後に，第 2.5 節を結言とする．

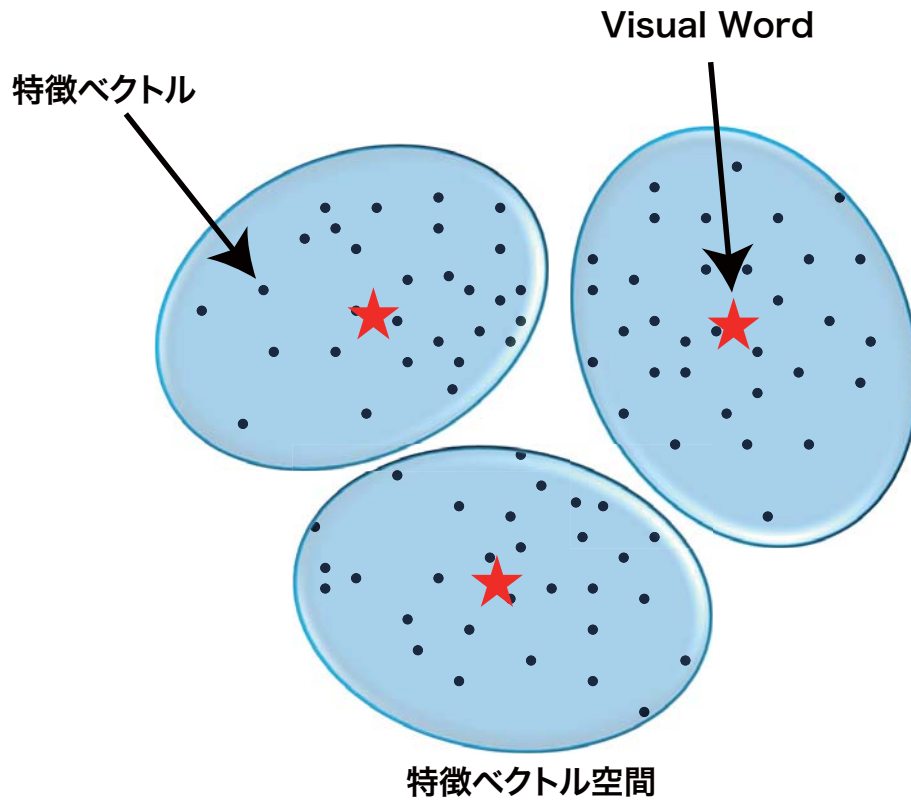


図 2.1: 局所特徴量のクラスタリングによる Visual Word の作成

2.2 関連研究

局所特徴量を用いた三次元物体認識手法のうち、物体の三次元物体モデルを構築しない手法 [36]–[39] は、物体を撮影した画像があれば、その画像から局所特徴量を抽出することによって簡単に物体モデルを構築できるという利点がある。このようなシンプルな手法で三次元物体認識の精度を得るには、様々な条件で撮影された多数の画像を物体モデル構築に用いる必要がある。一般的に一枚の画像から数十から数千個の局所特徴量が抽出されるため、ある一つの物体のモデル化には極めて多数の局所特徴量が関与することになり、その扱いが中心的な課題となる。

従来のメモリ使用量削減法 [52]–[54] の多くは、図 2.1 に示すように、局所特徴量をベクトル量子化して、Visual Word と呼ばれる代表ベクトルに置き換える方法を用いている。一般に Visual Word は、データベースに登録される物体から得られた全局所特徴量を、 k -means 法等のクラスタリング手法を用いて k 個のクラスに分類する。そして、求めたクラス

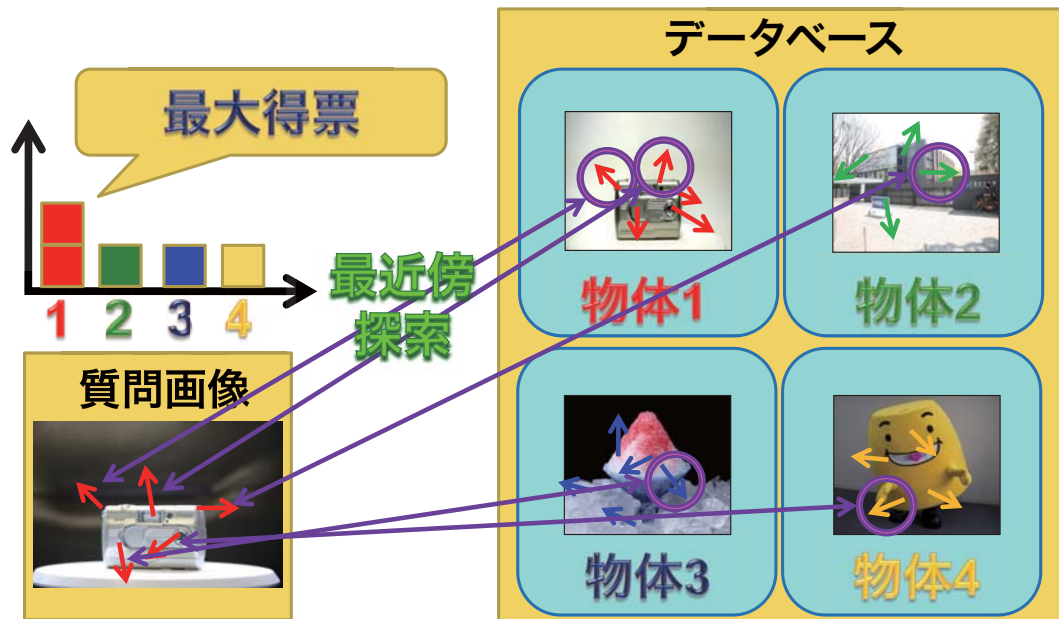


図 2.2: 投票処理に基づく特定物体認識手法

タ重心を Visual Word とする．未知の画像を認識する場合も，この画像から得た局所特徴量を Visual Word に置換し，照合する．この手法では，クラスタ数を少なくすればメモリ使用量を大幅に削減できる．しかし，物体インスタンスの識別の場合，認識対象にもよるが，Visual Word の数が増加すればそれだけ認識率も向上することが知られている．例えば，Nistèr らは，1600 万個の Visual Word を用いた例について報告している [52]．また，本道らの研究では，大規模な特定物体認識を行う場合，一つの Visual Word に 2～3 個の局所特徴量を対応づけるのが限度であり，それ以上の局所特徴量を対応づけると認識率が低下することを報告している [56]．また，大量の Visual Word を用いる場合，局所特徴量と Visual Word の照合に必要な計算時間が無視できず，様々なデータ構造 [19],[44]–[51] を用いて高速化する必要がある．

このような大量の Visual Word を用いる手法のうち，最も極端なものは，ベクトル量子化せず，局所特徴量の「事例」をすべて用いる手法である．このアプローチでは，高い認識率が期待できるものの，物体モデルの記録に必要なメモリ使用量は膨大になるという問題点が生じる．

さて，物体認識手法のうち，最も単純なものは，図 2.2 に示すような投票処理に基づくもの [16],[42] であろう．この手法では，上記の事例となる多数の局所特徴量に物体 ID を付与しておき，検索質問画像から得た局所特徴量との照合で，対応付いた局所特徴量に付与

されている物体 ID を持つ物体に投票する．照合は通常，最近傍探索^{[40],[41]}によって行われる．このようなプロセスでは，例えクエリとなる局所特徴量が，同じ物体の同じ箇所から得られるものに対応しなくても，同じ物体から得られる局所特徴量にさえ対応付けば，正解物体に投票することができる．このことから，最近傍探索と投票処理を用いる物体認識手法では，検索質問画像から得た局所特徴量に正しい物体 ID が割り当てられるかが重要となる．この点に着目すると，局所特徴量の中には，物体モデルから削除しても，クエリとなる局所特徴量の投票する物体が変化しないものが存在する．このような局所特徴量を削除すると，認識精度を落とすことなく，物体モデルのメモリ使用量を削減することができる．

すべての局所特徴量を記録した場合と全く（あるいはほぼ）同じ効果を保証しつつ，不要なものを削除する方法としては，Condensing^{[72]-[74]}と呼ばれるものが提案されている．この手法では，局所特徴量空間において，異なる物体 ID を持つ局所特徴量までの距離が等しくなる部分に識別境界線を引き，この境界線を引くのに用いられる局所特徴量のみを物体モデルへ登録する．この処理により，認識に影響を与えない局所特徴量を削除し，物体モデルのメモリ使用量を削減している．しかし，この手法には，局所特徴量の次元数が高次元になると，識別境界線を求めるのに非常に時間がかかるという問題や，登録される局所特徴量が多くなると，ほとんどの局所特徴量が識別境界線を引くのに用いられ，わずかしか物体モデルから削除できないという問題がある．

さらに，図 2.2 に示すように，多数決で物体が認識されることを考えると，すべての投票が正しくなくてもよいことがわかる．二次元平面物体の場合ではあるが，97%以上の認識率を得るために必要な照合の精度は高々15%程度であることが知られている^{[62],[75]}．この点に着目すると，Condensing で得られるものより，さらに少数の局所特徴量で物体モデルを構築可能であると考えられる．本章で提案する手法はこのような観点に基づくものである．

2.3 提案手法

局所特徴量をすべて物体モデルに登録すれば，三次元物体を高精度で認識できるが，局所特徴量の数が膨大なため，物体モデルに必要なメモリ使用量が莫大になるという問題点がある．本節では，この問題を解決するために，高精度の物体認識を実現するメモリ使用

量の少ない物体モデルを作成する手法を提案する。

2.3.1 考え方

三次元物体を高精度で認識できるとは、物体がどのような視点から撮影された画像を検索質問画像としても、この画像中に存在する物体を認識できることである。特に、第 2.2 節で述べた投票処理に基づく物体認識手法では、どのような画像を検索質問画像としても、正解物体が他の物体よりも一票でも多く得票できると言い換えることができる。そこで本研究では、投票処理に基づく物体認識手法を用い、高精度な物体認識を実現しつつ、物体モデルに必要なメモリ使用量を削減することを目的とする。

上記の目的を達成するには、最も高い認識率を得る最も少ない局所特徴量の組合せを求めることが課題である。全ての組合せを試せば、目的を達成する物体モデルを作成することができる。しかし、一般的に扱われる局所特徴量の数は、数十万個から物体数によっては数億個以上であるため、最適な組合せを求めることは困難である。また、別のアプローチとして、ランダムに物体モデルに登録する局所特徴量を決めるというものが考えられる。このアプローチでは、高速に物体モデルを作成することができるが、無作為に物体モデルに登録する局所特徴量を決めるため、認識率が低下してしまうという問題が発生する。そこで本研究では、投票処理に基づく物体認識手法の性質に着目し、組合せを求める手法よりも高速で、ランダムな手法よりも高精度な認識を行える物体モデル作成手法を提案する。

提案手法の処理の流れを図 2.3 に示す。投票処理を用いた物体認識手法では、物体モデルに登録されている各局所特徴量に物体 ID を付与しておき、検索質問画像から得られる局所特徴量と照合する。そして、対応付いた局所特徴量に付与されている物体 ID を調べ、その物体に投票する。このため、検索質問画像から得られる局所特徴量は、自分と同じ物体の局所特徴量と対応付くことで正しい投票を行い、異なる物体の局所特徴量と対応付くことで誤った投票を行う。本研究では、物体モデルに登録した局所特徴量が、物体認識に対して、検索質問画像から得られる局所特徴量に正しい投票を行わせるように関与する性質を「プラス関与性」と呼ぶ。プラス関与性を持つ局所特徴量の中には、複数の局所特徴量にプラス関与できるものが存在する。このような局所特徴量を物体モデルに登録する方が、同じ枚数の画像の認識に必要な局所特徴量の数が少なくて済むと考えられる。また本

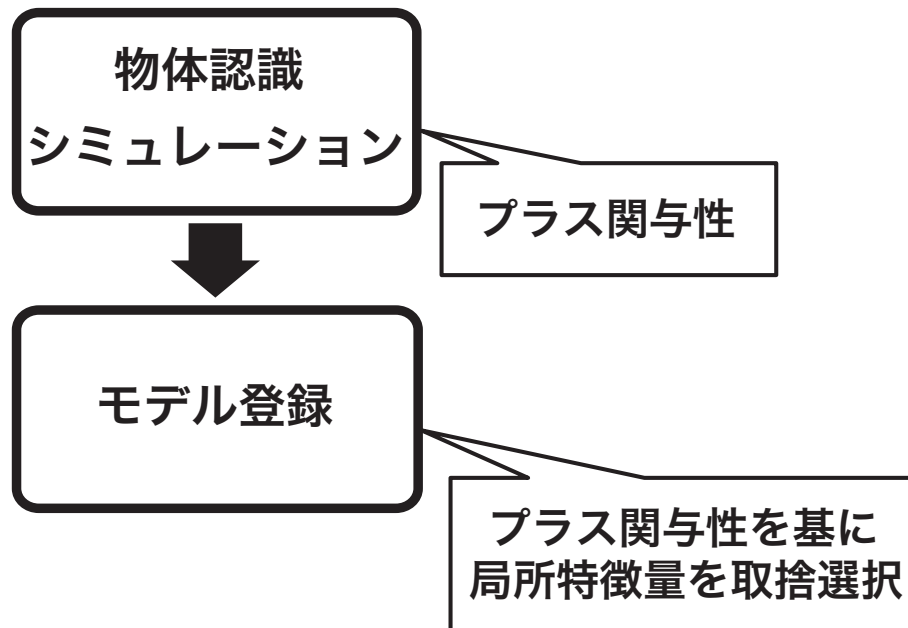


図 2.3: 局所特徴量の取捨選択に基づくメモリ使用量削減法の処理の流れ

研究では，このような局所特徴量を，目的を達成する上で有効な局所特徴量であると考えられる．そこで提案手法では，物体モデル構築用画像を用いて物体認識シミュレーションを行い，全局所特徴量のプラス関与性を調べ，物体毎に認識に有効な局所特徴量から順に物体モデルへ登録する．本研究では，物体を回転させて撮影した画像をフレーム画像と呼び，物体モデル構築用画像に用いる．局所特徴量には PCA-SIFT^[21] を用いる．物体モデル構築用画像から得られる局所特徴量には，どの物体の画像から抽出されたかを示す物体 ID と，その物体のどのフレーム画像から抽出されたかを示す画像 ID と，その画像のどの局所特徴量かを示す特徴量 ID を付与しておく．以下に，物体認識シミュレーションに用いる野口らの手法^[62] と，物体モデルが作成されるまでに行われる各処理について具体的に説明する．

2.3.2 野口らの手法

本節では，物体認識シミュレーションに用いる野口らの手法^[62] について詳しく説明する．この手法は，ハッシュ表を用いた近似最近傍探索に基づく特定物体認識手法である．この手法では，局所特徴量として PCA-SIFT^[21] で求めた 36 次元の特徴ベクトルを

用いる。

まず，野口らの手法のデータベース作成処理について述べる．まず，PCA-SIFT で求めた特徴ベクトル p の第 1 次元から第 d 次元 ($d \leq 36$) までをとり， $p' = (p_1, p_2, \dots, p_d)$ を作成する．次にこのベクトル p' を用い，

$$u_j = \begin{cases} 1 & \text{if } p_j - \theta_j \geq 0 \\ 0 & \text{otherwise} \end{cases} \quad (2.1)$$

によって各次元を 2 値化したビットベクトル $u = (u_1, u_2, \dots, u_d)$ を作成する．ここで θ_j は，認識対象となる画像全ての p_j の中央値である．そして，

$$H_{\text{index}} = \left(\sum_{j=1}^d u_j 2^{(j-1)} \right) \bmod H_{\text{size}} \quad (2.2)$$

というハッシュ関数より，ハッシュ表に登録するためのインデックスを求める．ここで， H_{size} は，ハッシュ表のサイズであり，文献 [62] では $H_{\text{size}} = 2^d$ としている．求めたインデックスに，特徴ベクトル p とこれに対する物体 ID を登録する．

野口らの手法では，特徴ベクトルの登録時に衝突が生じた場合，図 2.4 のようにリストとして特徴ベクトルを登録していく．但し，計算コストの削減のために，リスト長に閾値 c を設け，リスト長が c よりも長くなる場合，リスト全体をハッシュ表から削除し，以後の登録を禁止する．これは，同じハッシュ値を持つ特徴ベクトルは互いに類似しているため，物体認識にあまり寄与しないという考えに基づく．以上の処理を全ての特徴ベクトルに対して行い，データベースを作成する．

次に，野口らの手法の認識処理について述べる．野口らの手法では，まず検索質問画像から PCA-SIFT で求めた特徴ベクトルを用い，データベース作成処理と同様に，ハッシュ表のインデックスを求める．次に，求めたハッシュインデックスに登録されている特徴ベクトルとのみ距離計算し，近似的に最近傍の特徴ベクトルを求める．そして，この特徴ベクトルの物体 ID に投票する．この処理を，検索質問画像から得られる特徴ベクトル全てに対して行い，最終的に最も得票数が高い物体を認識結果とする．以下に具体的な処理について説明する．

まず，検索質問画像から得られる特徴ベクトルから，式 (2.1) よりハッシュ表のインデックスを求める．このとき，ハッシュ関数の性質より，次のような問題が発生する．検索質問画像とデータベースの画像が全く同じになることはほとんどないため，検索質問画像が

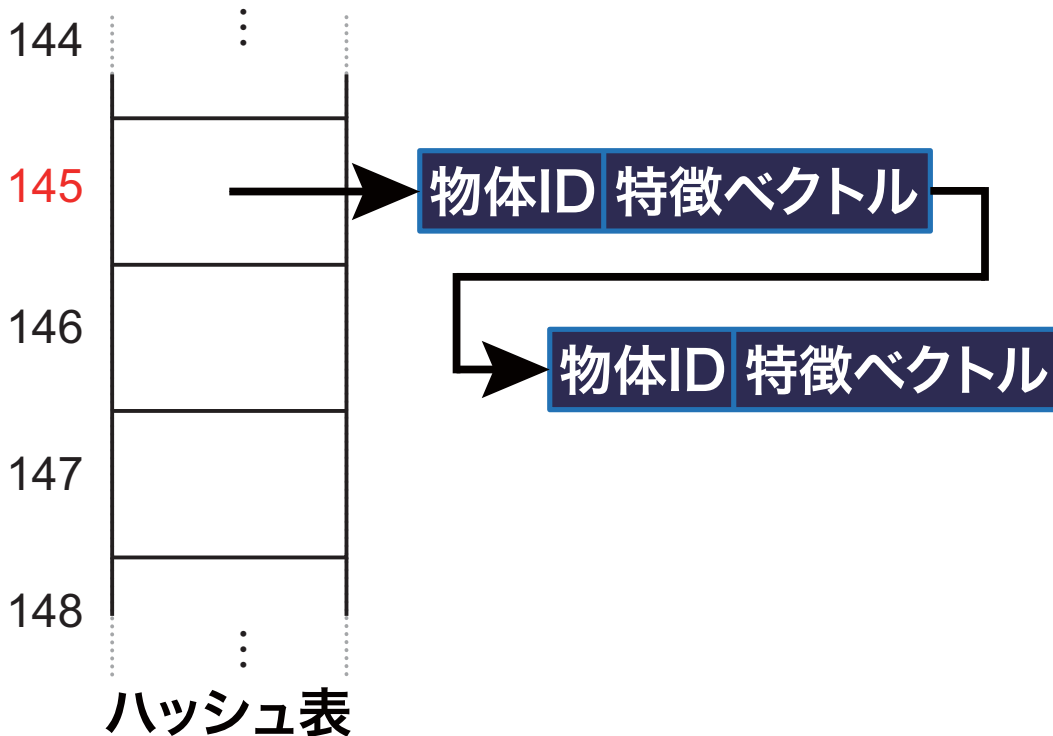
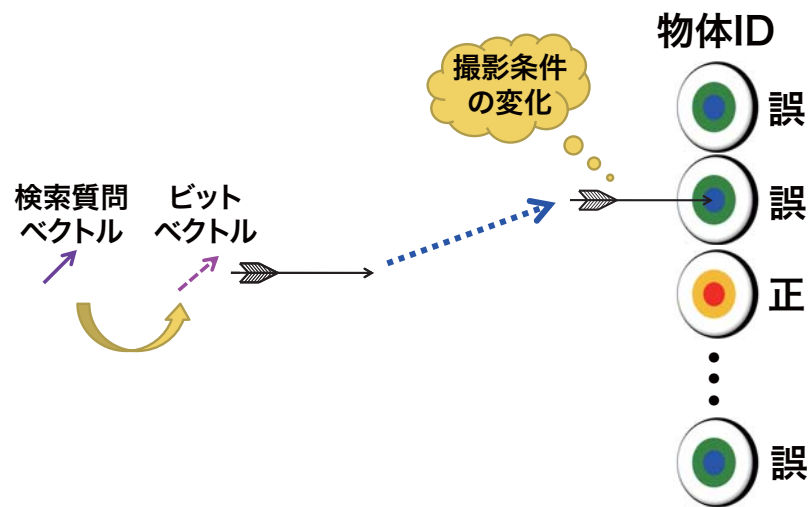
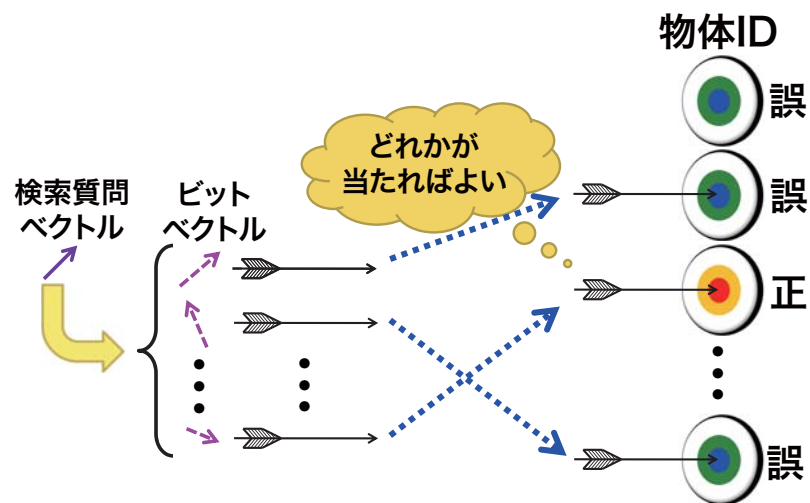


図 2.4: リスト構造による物体 ID の登録

ら得られる特徴ベクトルは、各次元の値がデータベース作成に用いる特徴ベクトルと異なったものとなる。これが原因で、異なったビットベクトルに変換されてしまうと、真に対応する物体 ID をハッシュ表より検索することができなくなる。この問題は、図 2.5 のように、ビットベクトルという矢を用いて、物体 ID という的に当てることに例えることができる。このとき、図 2.5 の (a) のように、一本の矢では正し的に当てるのが困難な場合が生じる。この問題に対処するため、野口らの手法では、図 2.5 の (b) のように「下手な鉄砲も数打てば当たる」という考えに基づき、次のような変動を考慮した複数のビットベクトルを生成し、認識処理に用いる。具体的には、各次元の値に許容変動幅 e を設けて対処する。検索質問となる特徴ベクトル $q = (q_1, q_2, \dots, q_d)$ において、 $\epsilon_j = |q_j - \theta_j| \leq e$ を満たす次元 j に対して、 u_j だけではなく $u'_j = 1 - u_j$ (0 ならば 1, 1 ならば 0) も用いて特徴ベクトルを検索する。しかし、この処理を全ての次元に対して行くと、処理時間が膨大になってしまうため、処理の対象とする次元数に閾値 b を設ける。これにより、最も多い場合で、検索に用いるビットベクトル u'_j の数を 2^b 個に抑える。 $\epsilon_j \leq e$ となる次元数が b 個より多くなる場合、 ϵ_j の小さい次元から順に b 個採用する。



(a)



(b)

図 2.5: 特徴ベクトルの変動への対処

次に、上記の処理で得られたビットベクトルを用いてハッシュ表を検索し、登録されている特徴ベクトルとの距離を計算する。そして、最もユークリッド距離の近い特徴ベクトルを求め、この特徴ベクトルの物体 ID に投票する。以上の処理を検索質問画像から得られる全特徴ベクトルに対して行うことで、最終的に最も得票数の高い物体を認識結果とする。

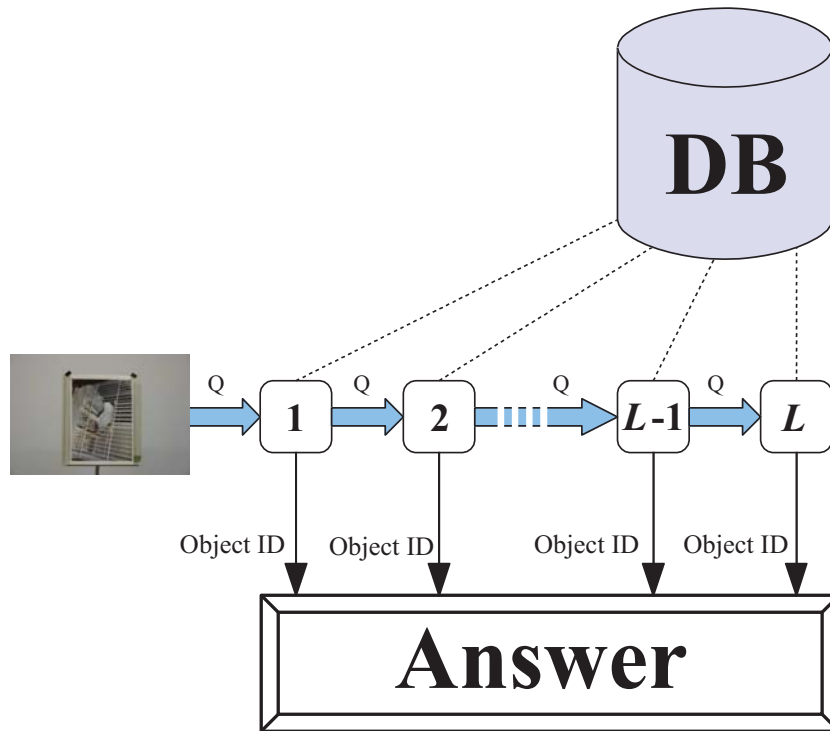


図 2.6: 多段階化による認識

また，野口らの手法では，“認識に必要な特徴ベクトルの照合精度は検索質問画像によって異なる”という観点に基づき，認識処理を効率化するために，図 2.6 に示すような複数の識別器を直列に並べた構成とする．第 l 段では， $b = l - 1$ とした識別器を用いる．検索質問画像をこの識別器にかけた場合の処理について具体的に説明する．まず第一段では， $b = 0$ としたときのビットベクトルを用いて認識する．このとき，得票数が一位の物体と他の物体との得票数の差が十分に大きければ，この時点で最大得票数を持つ物体を認識結果として出力する．これにより，後段の処理を打ち切ることができるため，認識処理を効率的に行うことができる．もし十分に差が開いていなければ，第二段の識別器に移り， $b = 1$ としたときのビットベクトルを用いて認識する．この処理を繰り返すことにより検索質問画像を認識する．この識別器の多段階化による認識手法の特徴は，識別器の段が移ったとき，前の段で検索に用いられたビットベクトル以外のビットベクトルを用いて認識を行うという，差分探索性があることである．このため，最終段の識別器まで移ったとしても，最初から 2^b 個のビットベクトルを用いて認識するのと同ほぼ同じ処理時間で認識できる．

2.3.3 物体認識シミュレーション

提案手法では、全フレーム画像を用いて物体認識シミュレーションを行い、各局所特徴量がどのフレーム画像の局所特徴量と対応付いたかを求める。これにより、各局所特徴量が物体認識に対してプラス関与するかどうかを調べる。本節では、この物体認識シミュレーションについて詳しく説明する。

本研究では、物体認識シミュレーションに前節で説明した野口らの手法を用いる。物体認識シミュレーションでは、ある一枚のフレーム画像から得られる局所特徴量をクエリとし、残りのフレーム画像から得られる局所特徴量を全て物体モデルに登録して認識を行う。このとき、クエリと対応付いた局所特徴量がクエリと同じ物体から得られる場合、この対応付いた局所特徴量は、物体認識に対してプラス関与する局所特徴量とする。物体認識シミュレーションでは、各局所特徴量がどの画像の局所特徴量と対応付いたかを記憶させ、有効性を吟味するために必要な情報を得る。以上の処理を全フレーム画像に対して行い、全局所特徴量のプラス関与性について調べる。

2.3.4 プラス関与性とその性質

本節では、プラス関与性を持つ局所特徴量の性質について説明する。物体認識シミュレーションの結果より、プラス関与性を持つ局所特徴量の中には、図 2.7 に示すように、特徴空間で複数の局所特徴量の最近傍となるものが存在する。ここで、図 2.7 の A_a 等の添え字は、大文字がその局所特徴量の物体 ID を、小文字がその局所特徴量の画像 ID を示している。このように局所特徴量の中でも、より多くの画像の認識にプラス関与できるものを物体モデルに登録する方が、同じ枚数の画像の認識に必要な局所特徴量の数が少なくて済むと考えられる。つまり、 n 枚の画像を認識するのに、一枚にしかプラス関与できない局所特徴量を n 個物体モデルに登録するよりも、 n 枚の画像にプラス関与する局所特徴量を一個登録する方が、登録する局所特徴量の数が少なくて済む。また、このような局所特徴量を近傍の局所特徴量の代表特徴量として物体モデルに登録すれば、メモリ使用量を削減できると考えられる。

以上の考えから本研究では、物体認識シミュレーションにおいて、より多くの画像の認識に対してプラス関与できると判断される局所特徴量ほど、目的達成のために有効な局所

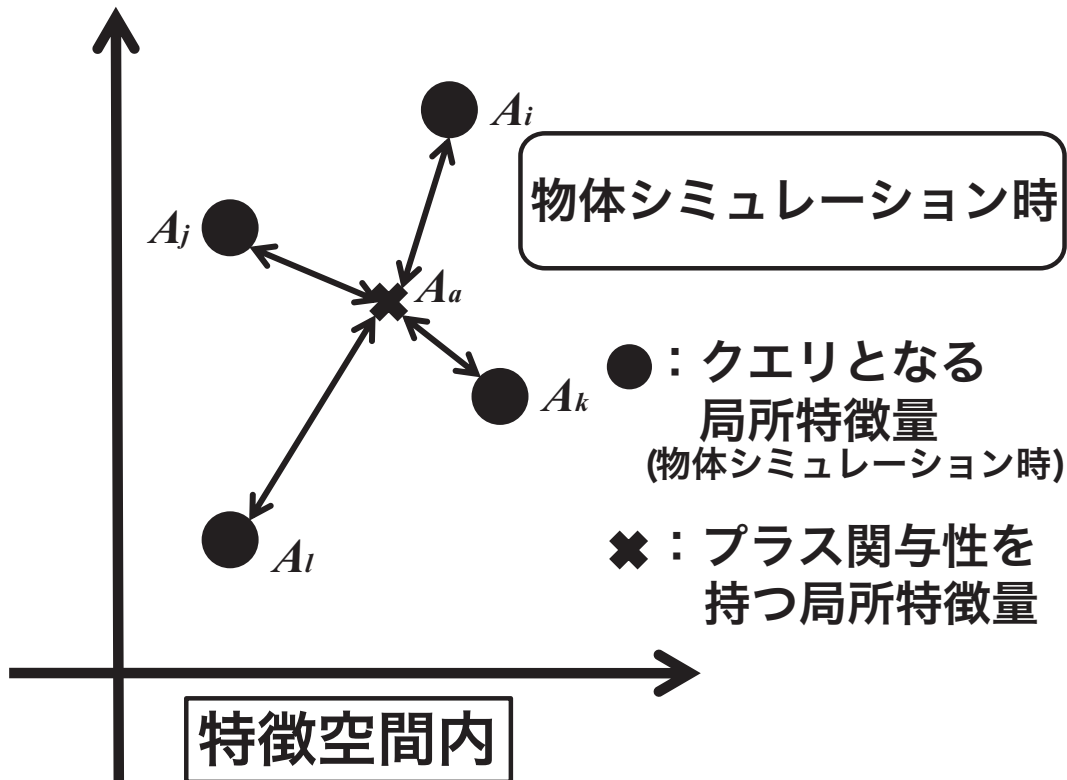


図 2.7: 複数の局所特徴量とプラス関与する局所特徴量

特徴量とする．そこで提案手法では，プラス関与性を持つ局所特徴量の有効性を判断する基準として，何枚の画像に対してプラス関与できるかを判断基準とし，この枚数を評価値 E_1 とする．この枚数計算では，着目している局所特徴量が抽出された画像も計算対象に含める．本研究では， E_1 の大きい局所特徴量ほど，物体認識シミュレーションにおいて多くの画像に対してプラス関与性を持つことから，目的達成に有効な局所特徴量と考える．

2.3.5 登録処理の流れと前提条件

高精度の物体認識を実現するメモリ使用量の少ない物体モデルを作成するには，目的達成のために有効なプラス関与性を持つ局所特徴量の組合せを探す必要がある．投票処理に基づく物体認識システムでは，他の物体より一票でも多く得票した物体が認識結果となるため，この局所特徴量の組合せとは，できるだけ少ない局所特徴量で，正解物体の得票数が最も多くなるものを意味する．さらに，物体をどのような角度から撮影した場合にも，

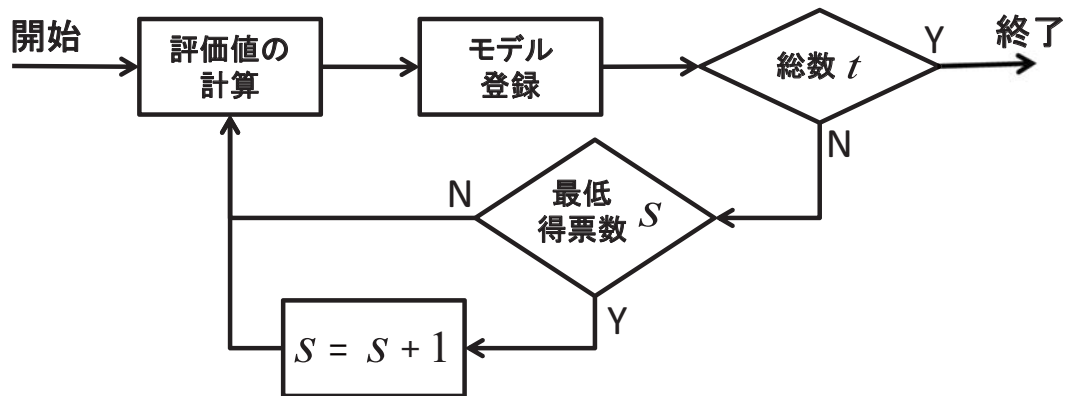


図 2.8: 物体モデル登録処理の流れ

安定して認識できる物体モデルを作成するには、あらゆる視点から得られる局所特徴量を満遍なく物体モデルに登録する必要がある。このため、最適な局所特徴量の組合せを求めることは困難である。そこで本節では、物体認識シミュレーションの結果から得られる局所特徴量のプラス関与性を基に、Greedy な方法で目的を達成する局所特徴量の組合せを求める。以下で提案手法の登録処理について詳しく説明する。

物体モデル登録処理の流れを図 2.8 に示す。まず、最低得票数 s を 1 に初期化する。最低得票数 s については、後で詳しく説明する。そして、物体認識シミュレーションの結果を基に、最低得票数を満たすように物体モデルに登録するための登録評価値を計算する。次に、登録評価値の最も大きな局所特徴量を物体モデルに登録する。提案手法では、物体モデルに登録される局所特徴量の総数を一物体 t 個とし、全ての物体が最低得票数を満たしたとき、物体モデルに登録されている局所特徴量の数が t 個未満であれば、最低得票数 s を 1 ずつ増やす。そして、再び全ての物体が最低得票数を満たすように、局所特徴量を物体モデルに登録していく。以上の処理を、全ての物体に対して登録される局所特徴量の総数が t 個になるまで繰り返す。但し、提案手法は、以下に示す前提条件が成り立つと仮定して物体モデルを作成する。物体モデルに登録されている局所特徴量間の実際の距離関係は、登録されている局所特徴量の組合せによって変動する。このため、ある局所特徴量が登録される前の物体モデルと後の物体モデルを用いて物体認識シミュレーションを行うと、対応する局所特徴量が変わる可能性がある。つまり、いくつかの局所特徴量では、ある局所特徴量の登録前の物体モデルと登録後の物体モデルで、評価値 E_1 の値が変動する可能性がある。従って、高精度な物体認識を実現する物体モデルを作成するには、局所特徴量が登録されるごとに物体認識シミュレーションを行い、評価値 E_1 を求め直す必要

がある．しかし，莫大な数の局所特徴量に対して，この処理を行うと，物体モデル作成に膨大な時間がかかる．そこで本研究では，この問題に対処するために，評価値 E_1 は登録処理によって変動しないと仮定する．つまり，評価値 E_1 は，物体認識シミュレーションにおいて，検索質問画像以外のフレーム画像から得られる局所特徴量を全て物体モデルに登録した場合の局所特徴量間の距離関係にのみ依存して決定されるものとする．

2.3.6 登録のための評価基準 1

本節では，物体認識シミュレーションの結果から得られる局所特徴量のプラス関与性を基に，物体モデルに登録するための具体的な登録基準について述べる．

前述の通り，評価値 E_1 の値が大きい局所特徴量ほど，物体認識シミュレーションにおいて多くの画像の認識に対してプラス関与性を持ち，目的を達成するために有効な局所特徴量である．そこで提案手法では，基本的に評価値 E_1 の大きいものから順に物体モデルへ登録する．しかし，単純に評価値 E_1 の大きいものから順に登録すると，局所特徴量が密に抽出される部分のものばかり登録され，疎な部分のものが登録されにくくなる．このため，局所特徴量が抽出されにくい方向から物体を撮影した画像が検索質問画像として与えられると，このような部分の局所特徴量が物体モデルに登録されておらず，正しく物体を認識できないという問題が発生する．そこで提案手法では，この問題を解決するために，登録のための評価基準 E を以下のように設定する．

話を簡略化するために，いくつかの局所特徴量が既に物体モデルに登録されているとして話を進める．ここで，この物体モデルをカレントモデルと呼ぶ．まず，このカレントモデルを用い，物体認識シミュレーションにおいて，各フレーム画像が何票正しい投票を得ることができるのかを調べる．これは，局所特徴量のプラス関与性から求めることができる．このとき，各フレーム画像の得票数が図 2.9 の左のようになったとしよう．次に，このカレントモデルに対して，新たに局所特徴量を加えることを考える．物体モデル登録候補として，二つの局所特徴量 α と β が存在すると仮定する． α を加えたとき，正しい投票数が図 2.9 の右上段のように， β を加えたとき，正しい投票数が図 2.9 の右下段のようになったとする．ここで， α と β のどちらを物体モデルに加えると良いか考える．新たに加わる票数は α の方が多いものの，全ての画像を認識するという観点から考えると， β は，カレントモデルで正しい投票がない画像に票を多く入れることができる．このため，本研

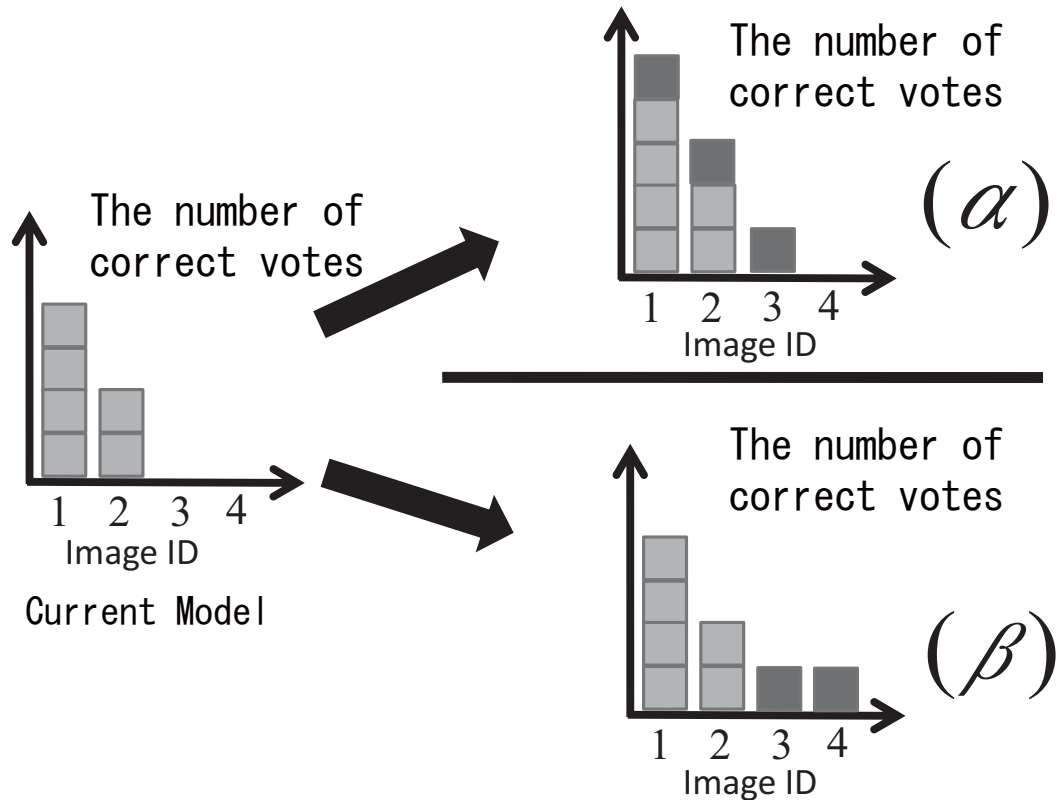


図 2.9: カレントモデルを用いた場合の正しい投票数

究では β の方が目的達成のために有効であると考える。

そこで提案手法では、このような効果を評価するために、正しい投票に対して最低得票数 s を設定し、全てのフレーム画像が s 票以上の票が得られるように、評価値 E を以下のように設定する。

$$E = E_1 - E_2 \quad (2.3)$$

ここで、 E_2 は、ある局所特徴量がプラス関与する画像のうち、現時点で獲得している票数が最低得票数 s 以上の画像の枚数を意味する。提案手法では、この登録評価基準 E の最も大きい局所特徴量から順に Greedy に物体モデルへ登録する。図 2.9 の場合、 $s = 1$ とすると α の評価値 E の値は $E = 1 (E_1 = 3, E_2 = 2)$ 、 β は $E = 2 (E_1 = 2, E_2 = 0)$ である。提案手法では、物体毎に物体モデルへの登録総数に閾値 t を設定し、全ての画像が s 票を満したとき、物体モデルに登録されている局所特徴量が t 個未満であれば、 s を 1 ずつ増やす。そして、再び全ての画像が s 票得られるように局所特徴量を物体モデルへ登録する。

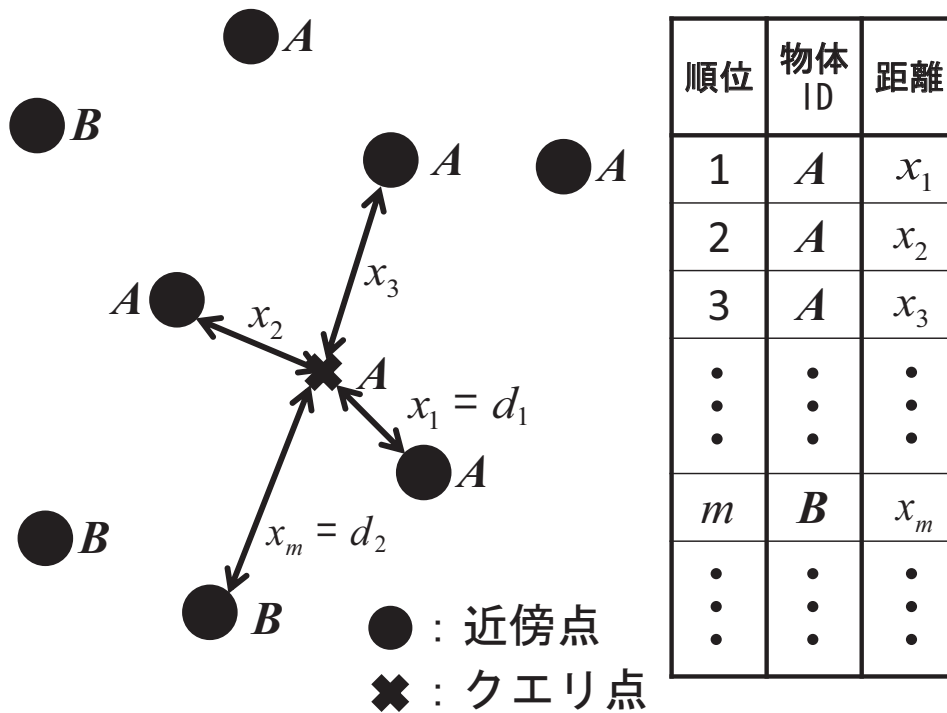


図 2.10: 近傍点との距離計算

2.3.7 登録のための評価基準 2

提案手法では、 E の値を評価基準として局所特徴量を物体モデルへ登録する。このとき、 E の値が等しい局所特徴量が存在する場合がある。本節では、このような場合に局所特徴量を物体モデルへ登録する順番を決める評価基準について説明する。

まず、評価基準設定の考え方から述べる。提案手法では、物体認識シミュレーション時に、近傍点との距離を計算している。そこで、各局所特徴量と近傍の局所特徴量との距離を保存しておき、これをもとに評価基準を設定する。この距離計算により、クエリの局所特徴量に対して、特徴空間で同じ物体 ID を持つ局所特徴量との距離が近ければ、クエリの局所特徴量が正しく投票する可能性が高くなると考えられる。また、特徴空間で異なる物体 ID を持つ局所特徴量との距離が遠ければ、誤投票を行う可能性が低くなると考えられる。そこで提案手法では、上記の考えのもとで登録する局所特徴量の順番を決める。以下で具体的に説明する。

提案手法では、図 2.10 に示すように、物体認識シミュレーションにおいて、クエリとなる局所特徴量とその近傍点との距離を計算する。このとき、クエリとなる局所特徴量の物体 ID と、最近傍となる局所特徴量の物体 ID が等しいとき、この局所特徴量間の距離を d_1 とする。図 2.10 では、 x_1 を d_1 とする。この d_1 の値が小さいものほど、 E が等しい局所特徴量の中でも、よりクエリとなる局所特徴量の近傍に同じ物体 ID を持つ局所特徴量が存在することを意味する。次に、クエリとなる局所特徴量と異なる物体 ID を持つ局所特徴量の中で、最も距離の小さいものとの距離を d_2 とする。図 2.10 では、 x_m が d_2 となる。この d_2 の値が大きいものほど、クエリとなる局所特徴量の近傍に他の物体の物体 ID を持つものが存在しないことを意味する。提案手法では、この d_1 と d_2 を用い、

$$D = \frac{d_2}{d_1} \quad (2.4)$$

を、局所特徴量の登録評価基準として設定し、 E の値が等しいとき、評価値 D の値が大きいものから順に物体モデルへ登録する。但し、 D の値を求める際に以下の三つの点を考慮する。

第一に、第 2.3.2 節で述べたように、物体認識シミュレーションで用いる野口らの手法では、ハッシュ表を用いた近似最近傍探索によってクエリとなる局所特徴量の近傍点を選定している。このため、近傍に他の物体 ID を持つ局所特徴量が存在しない場合がある。このような局所特徴量は、物体モデルへ登録しても誤認識を引き起こす可能性が低いと考えられる。そこでこのような場合、 $d_2 = \infty$ とする。第二に、クエリとなる局所特徴量と最近傍の局所特徴量の物体 ID が異なる場合がある。このような局所特徴量は、物体モデルへ登録すると誤認識を引き起こす可能性が高いと考えられる。そこで提案手法では、 $D = 0$ とする。第三に、クエリとなる局所特徴量の近傍点が全く求まらない場合がある。このような局所特徴量は、大量にあるフレーム画像のどの画像の認識にも関与していないことを意味する。このため、このような局所特徴量は、認識率に影響を及ぼす可能性が低いと考えられる。そこで提案手法では、このような場合 $D = 0$ とする。

2.4 実験

本実験では、我々が独自に用意した 55 個の物体と COIL-100 を用いて、物体モデルに登録される局所特徴量の数と認識率、メモリ使用量の関係について検証した。



図 2.11: 55 物体の例

2.4.1 実験条件

まず，55 物体のデータセットについて説明する．図 2.11 に本実験で用いた物体の例を，付録の図 A.1，A.2 に全物体を示す．本実験では，物体に対して，真正面，真正面から上へ 15° ， 30° の角度から，ハイビジョンビデオカメラ¹で， 5° ごとに物体を回転させて撮影した画像を用いた．このうち，フレーム画像として，回転角度が 0° ， 10° ， \dots ， 350° の画像 $36 \text{ 枚} \times 3 \text{ 方向}$ の，一物体当たり 108 枚を用い，残りの画像を検索質問画像とした．フレーム画像から得られる全局所特徴量数は約 150 万個であり，これらを全て物体モデルへ登録した場合の物体モデルのメモリ使用量は 115.7 [MB] である．物体モデルのメモリ使用量は，局所特徴量そのものを記憶するためのメモリ使用量と，その局所特徴量の物体 ID を記憶するためのメモリ使用量の合計とする．本実験では， I_t を全検索質問画像の枚数， I_c を正しく認識できた検索質問画像の枚数とおくと，物体の認識率 \mathcal{R} を以下で定義する．

$$\mathcal{R} = \frac{I_c}{I_t} \quad (2.5)$$

ここで，正しく認識できるとは，認識結果の物体と正解物体が一致することを意味する．全局所特徴量を登録した物体モデルを用いた場合の物体の認識率は 99.6% である．

次に，COIL-100^[68] のデータセットについて説明する．COIL とは，100 物体を 5° ごとに回転させて撮影した画像データベースである．解像度は 128×128 である．本実験では，角度が 0° ， 10° ， \dots ， 350° の画像，一物体当たり 36 枚をフレーム画像に用い，残りの画像を検索質問画像とした．全フレーム画像から得られた全局所特徴量数は約 18 万個で，その

¹SONY HDR-HC1，解像度:720×480



図 2.12: COIL-100 のフレーム画像の例

ときの物体モデルのメモリ使用量は 13.6 [MB] である．この物体モデルを用いた場合の認識率は 98.9% である．

両実験において三次元物体認識手法には，野口らの手法を用いた．ここで，野口らの手法のパラメータについてまとめておく．ビットベクトルの変換に用いる次元数 d ，ビットベクトルで反転操作を行う次元数 b ，値の変動幅 e ，重複するビットベクトルの上限数 c の四つである．提案手法で行う物体認識シミュレーションと本実験では，用いたパラメータは同じで， $d = 24, b = 10, e = 200, c = 10$ ，ハッシュ表のサイズを $H_{\text{size}} = 2^d$ とした．また比較手法として，最も単純で現実的な時間で処理が終了するランダム手法を用いた．ランダム手法では，物体ごとに登録する局所特徴量数が一定となるように，局所特徴量をランダムに物体モデルへ登録する．両実験において，ランダム手法では登録総数 t ごとに物体モデルを 100 種類作成した．登録総数 t は，55 物体を用いた実験では， $t = 200, 300, \dots, 1000$ とし，COIL-100 を用いた実験では， $t = 100, 150, \dots, 300$ とした．

2.4.2 実験結果

55 物体を用いて行った実験の結果を図 2.13 に示す．但し，ランダム手法は 100 個の物体モデルの平均認識率を示している．実験結果より，提案手法はランダム手法より，メモ

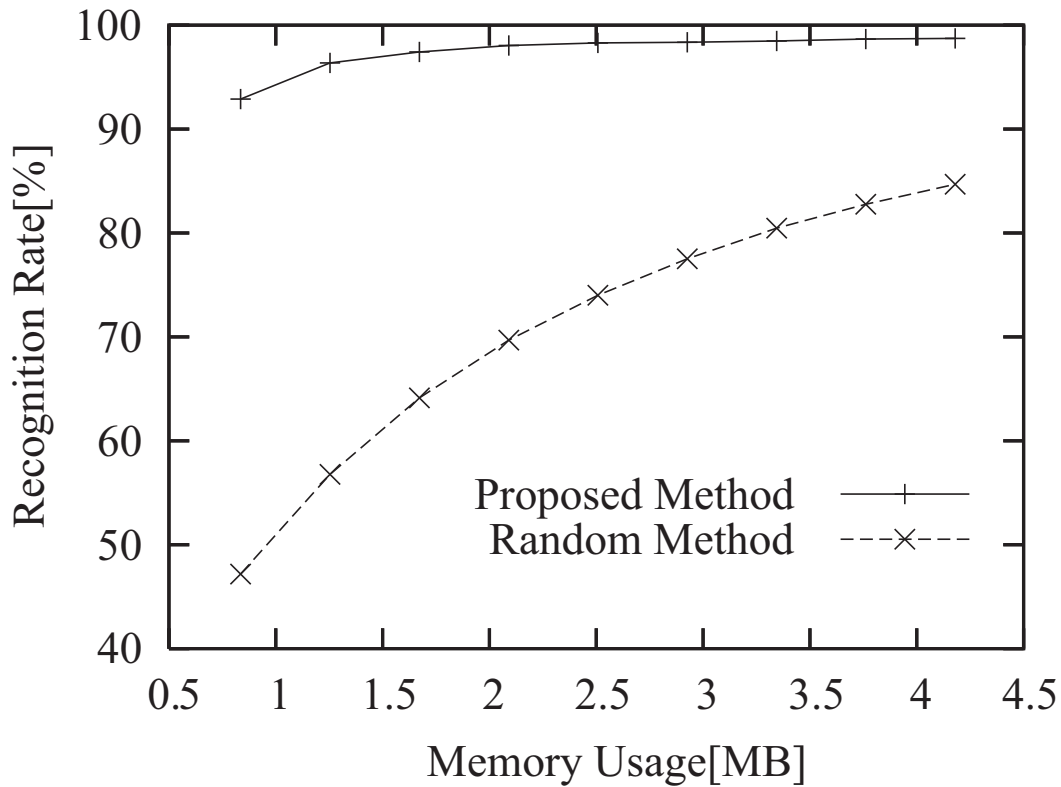


図 2.13: 実験結果 (55 物体)

り使用量が少なくなればなるほど有効であることが分かる．ここで，提案手法で作成した物体モデルのうち，メモリ使用量が 0.84[MB] と最も小さい $t = 200$ の物体モデルに着目する．この物体モデルでは，誤認識を起こした画像のうち約 40% が，図 2.11 の下段に示す物体と誤認識していた．この原因として，物体認識シミュレーション時に同一物体に正しい投票をする局所特徴量しか評価せず，他の物体の局所特徴量への誤投票のしやすさを評価していないということが考えられる．この問題を解決するには，誤認識を引き起こしやすい局所特徴量を評価する新しい評価基準を設ける必要があるだろう．また提案手法では，第 2.3.5 節で述べたように，物体モデルの作成処理中において，評価値 E_1 は局所特徴量の組合せによって変動しないものと仮定し，物体認識シミュレーション時の評価を用いている．しかし，実際は登録されている局所特徴量の組合せによって，評価値 E_1 の値は変動するため，この仮定が認識率に影響を与えていると考えられる．このことから，認識率を向上させるには，この仮定を見直す必要があると考えられる．しかし，単純に局所特徴量が登録されるごとに物体認識シミュレーションを行い評価値 E_1 を計算し直すには非常

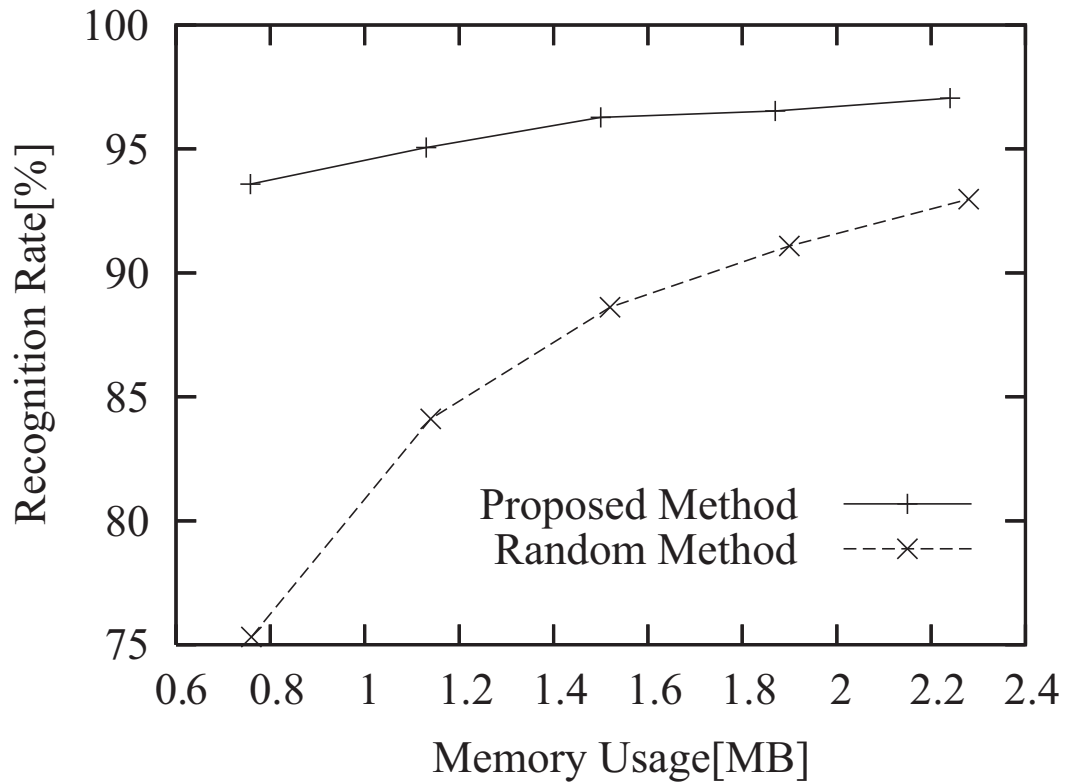


図 2.14: 実験結果 (COIL-100)

に時間がかかる．このため，効率よく評価値を再計算する手法も提案する必要がある．

次に，COIL-100 を用いて行った実験の結果を図 2.14 に示す．55 物体の場合と同様，提案手法はランダム手法より，メモリ使用量が少なくなるほど有効であることが分かった．ここで，提案手法で作成した物体モデルの内，容量が 0.76[MB] と最も小さい $t = 100$ の物体モデルに着目する．この物体モデルを用いた場合，認識に失敗した画像のうち約 50% が，全く得票することができないことが分かった．これは，得られる局所特徴量の数が物体によって違うにも関わらず，局所特徴量の登録数を一定にしていることが原因である．このため，物体ごとにモデルに登録する局所特徴量の数を調整する必要があると考えられる．

2.4.3 考察

55 物体の実験結果と COIL-100 の実験結果について比較すると，ランダム手法に対する提案手法の有効性が異なることが分かる．この原因として，COIL-100 の物体の方が 55

物体と比べ、画像の解像度が低く、得られる局所特徴量の数が少ないため、提案手法とランダム手法で物体モデルに登録される局所特徴量が重複しやすいことが考えられる。このため物体モデル構築用画像には、ある程度の解像度が必要であると考えられる。

以上のように、いくつか改善の必要性があるものの、提案手法のような単純な手法でもランダム手法の認識率を大幅に上回ることができた。

次に、ベクトル量子化を用いた従来手法との関係について述べる。実験に用いたデータのように、三次元物体を少しずつ違う角度から撮影した画像群は冗長性が高いため、クラスタリングによるベクトル量子化でも高い削減効果が得られると思われるかもしれない。具体的に見てみよう。55 物体の一物体当たりの画像数は、108 枚であり、COIL-100 の一物体当たりの画像枚数は、36 枚である。これに対して、それぞれの物体モデルのメモリ使用量を $1/140$ 、 $1/18$ に削減すると、元の画像群から得た多数の局所特徴量を 0.8 枚、2 枚の画像から得た局所特徴量の数になるまで、クラスタリングでまとめることになる。その結果、得られた代表ベクトルは元のものからかけ離れ、認識には有効でなくなってしまうと考えられる。提案手法では、プラス関与という概念に基づく大胆な取捨選択により、このような削減を可能としている。

2.5 結言

本章では、局所特徴量を用いた特定物体認識において、局所特徴量を取捨選択することで、高精度な認識を行えるメモリ使用量の少ない物体モデルを作成する手法を提案した。55 物体を対象とした実験の結果、メモリ使用量を約 140 分の 1 に削減した物体モデルで 92.9% の認識率を得た。また、COIL-100 を対象とした実験の結果、メモリ使用量を約 18 分の 1 に削減した物体モデルで 93.6% の認識率を得た。実験結果より、局所特徴量のプラス関与性という考えのもと、物体認識に対して貢献度の高いものから Greedy に物体モデルへ登録することが、無作為に取捨選択するよりも有効であることが分かった。しかし実験結果より、得られる局所特徴量の数が少ない物体は、投票処理に基づく認識手法では、得票しにくいことが分かった。このことから、物体毎に登録する局所特徴量の数を調整することや、得票しにくい物体が得票しやすくなるような評価基準を設定することが今後の課題である。他の課題として、誤投票のしやすさを登録のための評価の導入することや、局所特徴量間の距離関係の変動を考慮した評価基準の選定などが挙げられる。

第3章 Bloomier Filterを用いたメモリ 使用量削減法

3.1 緒言

本章では，局所特徴量の照合に一致検索を用いて物体認識に必要なメモリ使用量を削減する手法を提案する．局所特徴量の照合に，距離のような量的な概念に基づく類似検索を用いる物体認識手法では，距離計算に必要な特徴ベクトルの情報を全て保存しておく必要があり，これが物体認識に必要なメモリ使用量の大半を占めている．この問題を解決する一つの考え方は，照合に際して距離計算を放棄することである．これを実現する一つのアプローチは，局所特徴量の一致検索である．このアプローチでは，各局所特徴量にIDを割り振り，このIDが一致するかどうかで照合する．一致検索では，距離計算に必要な局所特徴量自体の記録は必要なく，あるIDを持った局所特徴量の登録の有無のみを記録すれば良いため，大幅にメモリ使用量を削減できる．

実際これまでに，この考えのもとハッシュ表を用いたメモリ使用量削減法^[62]が提案されている．文献[62]では，局所特徴量の照合を，ハッシュ関数の値(ハッシュ値)が同じかどうかを調べることで実現している．しかし，この手法にもまだ問題がある．局所特徴量の識別性を高めるには，類似する局所特徴量は同じハッシュ値を持つようにし，類似しない局所特徴量は異なるハッシュ値を持つようにしなければならない．つまり，類似する局所特徴量は同じハッシュ表のインデックスに登録し，類似しない局所特徴量は異なるハッシュ表のインデックスに登録される必要がある．これを達成するには，上記のような類似性を反映したハッシュ関数を作成しなければならない．文献[62]で用いられているハッシュ関数は，ハッシュ値の取り得る値の絶対数を，登録される局所特徴量数よりもかなり大きくすることで，局所特徴量の識別性を高めている．しかしその結果，局所特徴量に割り当てられるハッシュ値に偏りが生じ，ハッシュ表の大部分のインデックスに一つの局所特徴量も登録されていないという問題が生じる．このため，文献[62]で用いられてい

るハッシュ表には、空間効率が悪いという問題点がある [76],[77] .

本章では、この問題を解決するために、文献 [62] で用いられているハッシュ表の代わりに Bloomier Filter [63] と呼ばれる空間効率の良い確率的データ構造を用いる手法を提案する [64]-[67] . Bloomier Filter とは、ある要素がデータ集合のメンバであるかどうかを記録する Bloom Filter [78] と呼ばれるデータ構造を複数組み合わせることで、登録される要素に関連する値を返す連想配列である . 提案手法では、この Bloomier Filter を用い、局所特徴量を符号化し、この符号が一致するかどうかで局所特徴量を照合する . そして、この符号から物体 ID を連想させることで物体を認識する . また、提案手法では、上記のハッシュ表を用いる手法と同様に、局所特徴量の距離計算に必要な情報ではなく、Bloom Filter に局所特徴量の有無のみを記録するため、大幅にメモリ使用量を削減できる . さらに、提案手法では、局所特徴量の保存に用いるデータ構造を、ハッシュ表から空間効率の良い Bloomier Filter に変更することで、物体認識システム全体に必要なメモリ使用量も削減できる .

文献 [62] の手法との大きな違いは、局所特徴量の照合精度である . 提案手法では、Bloomier Filter を用いて局所特徴量を符号化する際に、符号化に失敗し、本来とは異なる符号に変換される場合がある . このとき、正しく局所特徴量を照合できず、局所特徴量から本来とは異なる物体 ID が連想される . 提案手法では、ハッシュ表を用いる手法よりも空間効率を良くすることと引き替えに、この符号化誤りの影響で、誤った物体 ID を連想してしまうことを容認する . このため、ハッシュ表を用いる手法よりも認識精度が低下する問題がある . この問題に対処するために、本研究では誤り検出符号であるパリティビットの概念を導入する . 本研究では、この概念に基づき、符号化誤りが認識精度に与える影響を少なくするために、パリティビットを連想させる Bloomier Filter も作成し、認識処理に用いる [79] .

本章の構成は以下の通りである . まず、第 3.2 節では、従来のメモリ使用量削減法のうち、局所特徴量の照合に距離計算を利用するものについて述べた後、本実験で比較手法として用いる従来のメモリ使用量削減法について詳しく説明する . 次に、第 3.3 節では、提案手法に用いる Bloomier Filter とその基礎となる Bloom Filter を説明しつつ、これら二つの Filter を用いたメモリ使用量削減法を提案する . そして、提案手法の有効性を確認するために、1 万個の二次元物体データセットと 55 個の三次元物体のデータセットを用いて行った実験とその結果、考察について、それぞれ第 3.4 節と第 3.5 節で述べ、最後に、

第 3.6 節を結言とする。

3.2 従来手法

本節では、特定物体認識における従来のメモリ使用量削減法のうち、局所特徴量の照合に距離計算を利用するものについて紹介する。また、本実験で比較手法として用いる木構造に基づくメモリ使用量削減法^[52]と、ハッシュ表に基づくメモリ使用量削減法^[62]についてそれぞれ詳細を示す。

3.2.1 従来研究の主な取り組み

特定物体認識における従来のメモリ使用量削減法には、第 2.2 節で述べた局所特徴量をベクトル量子化するものに加え、スカラ量子化^{[43],[55]}するものがある。局所特徴量をスカラ量子化する手法では、局所特徴量の各次元の値を量子化することでメモリ使用量を削減している。この手法では、ある程度大規模な物体認識でも高い認識精度を保ったままメモリ使用量を削減できる。例えば、野口らの研究では、局所特徴量の各次元の値を 2 bit にまで量子化しても、高精度に認識できることを報告している^{[43],[55]}。しかし、例え局所特徴量の各次元の値を 1 bit で表現しても、局所特徴量の数に比例してメモリ使用量が増加するため、更なる大規模な物体認識を行うには限度がある。このことから、スカラ量子化といった局所特徴量そのもののメモリ使用量を削減する手法には限界がある。

局所特徴量を量子化する以外のアプローチには、主記憶ではなく補助記憶を利用する手法^{[57],[58]}や部分空間法^{[80]–[82]}を用いるものがある。前者のアプローチでは、局所特徴量の情報を補助記憶に保存しておき、記録している場所へのポイントのみを主記憶に保存することでメモリ使用量を削減している。この手法では、かなり大規模な物体認識を行えるが、補助記憶へのアクセス時間がボトルネックとなり、認識処理に時間がかかるという問題が生じる。また、部分空間法を用いる手法では、類似する局所特徴量を主成分分析することにより、これらの部分空間を作成し、データベースに登録する。この手法では、複数の局所特徴量を一つの部分空間で表現するため、全ての局所特徴量を保存するよりもメモリ使用量を削減することができる。しかし、この手法も部分空間の類似度計算に時間がかかるという問題がある。

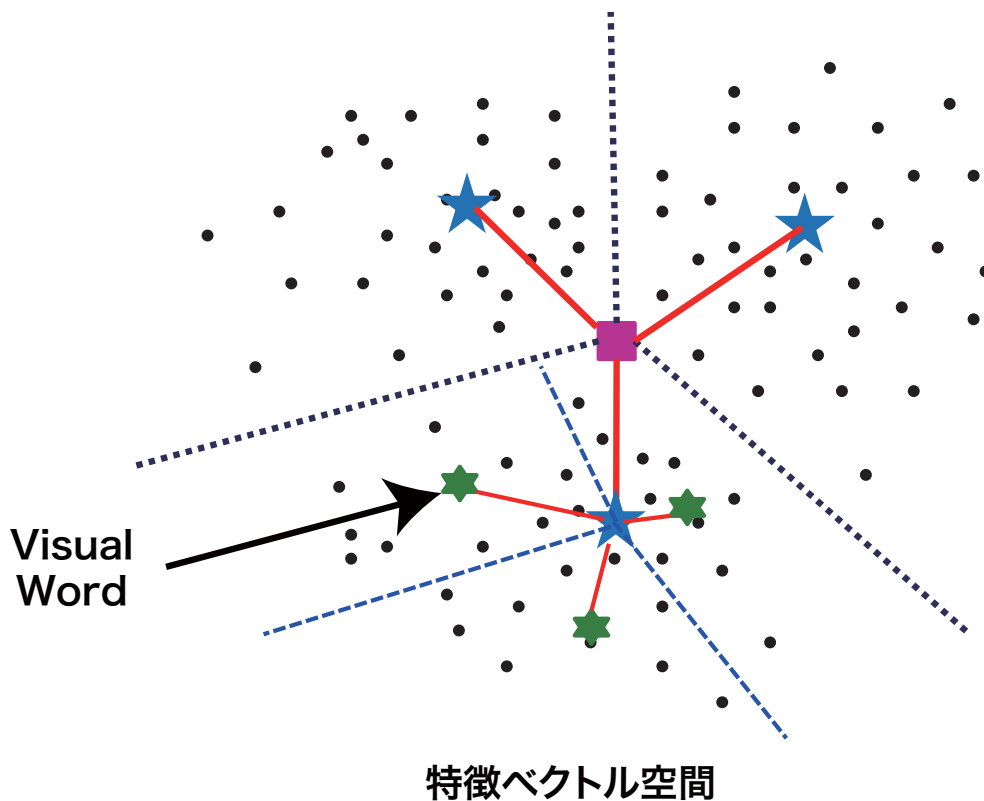


図 3.1: 階層的 k -means クラスタリングによる Vocabulary Tree の作成

上に述べたような手法では，局所特徴量間の距離を計算するため，どの手法でも個々の局所特徴量自体を記録する必要があり，認識に必要なメモリ使用量を削減するには限界がある．この問題を解決するための一つのアプローチとして，局所特徴量の距離計算を放棄することが考えられる．本章で提案する手法は，この考えに基づくものである．

3.2.2 木構造に基づくメモリ使用量削減法

本節では，従来のメモリ使用量削減法として，木構造を用いた Vocabulary Tree ^[52] を取り上げる．この手法は，物体認識の分野で近年最も広く用いられている “Bag-of-Features” ^{[17],[83]} と呼ばれる，文書検索法の “Bag-of-Words” に着想を得た手法の一種である．本研究では，提案手法の性能を評価するために，この手法を比較手法として用いる．

まず，Vocabulary Tree のデータベース作成処理について述べる．この手法は，図 3.1 のように，階層的 k -means クラスタリングを行うことにより，Vocabulary Tree と呼ばれ

る木を作成し，葉ノードを Visual Word とする．具体的には，全ての物体から得られる局所特微量に k -means クラスタリングを施して k 個のクラスタを作成し，各クラスタ重心を Vocabulary Tree のノードとする．そして，クラスタごとに対応している局所特微量のみを用いて再び k -means クラスタリングを行い，クラスタを形成し，木を拡張していく．この処理を繰り返すことにより作成された Vocabulary Tree の葉ノードを Visual Word とする．この手法では，Vocabulary Tree の枝の分岐数 x と，Vocabulary Tree の深さ y を調節することにより，Visual Word 数を変えることができる．例えば， $x = 10$ ， $y = 6$ のとき，Visual Word の数は 10^6 となる．Vocabulary Tree では，求めた Visual Word を用い，データベースに登録する物体ごとに，Visual Word の出現頻度を記録しておく．

認識処理では，木構造を根から葉に向かって調べることにより，検索質問画像から得られる局所特微量と最も類似する Visual Word を求める．この処理を繰り返し，tf-idf 重み^[84]を用いて，検索質問画像の Visual Word の出現頻度を計算する．そして，検索質問画像の Visual Word の出現頻度と，データベースに登録されている物体から得られる Visual Word の出現頻度を比較し，最も類似度の高い物体を認識結果とする．

Vocabulary Tree では，葉ノードの数を減らしたり，Vocabulary Tree の深さを浅くすることで容易にメモリ使用量を削減できる．しかし，これは，多数の局所特微量を一つの Visual Word に対応付けることを意味するため，得られた Visual Word は元の局所特微量からかけ離れたものとなることがある．このため，局所特微量の識別能力が低下するという問題が生じる．

3.2.3 ハッシュ表に基づくメモリ使用量削減法

本節では，従来のメモリ使用量削減法として，ハッシュ表を用いた手法^[62]について詳しく述べる．この手法は，第 2.3.2 節で説明した野口らの手法を改良したものであり，特徴ベクトルの照合に一致検索を用いる．これは，提案手法の基礎となる手法である．本研究では，この手法も比較手法として用いる．ここで説明のために，第 2.3.2 節で説明した野口らの手法を「ハッシュ(距離計算あり)」，本節で説明する野口らの手法を改良したものを「ハッシュ(距離計算なし)」と呼ぶ．ハッシュ(距離計算あり)からの改良点は，以下の三点であり，その他の処理は両者とも同じである．

一点目の違いは，ハッシュ表に登録するデータである．ハッシュ(距離計算あり)では，

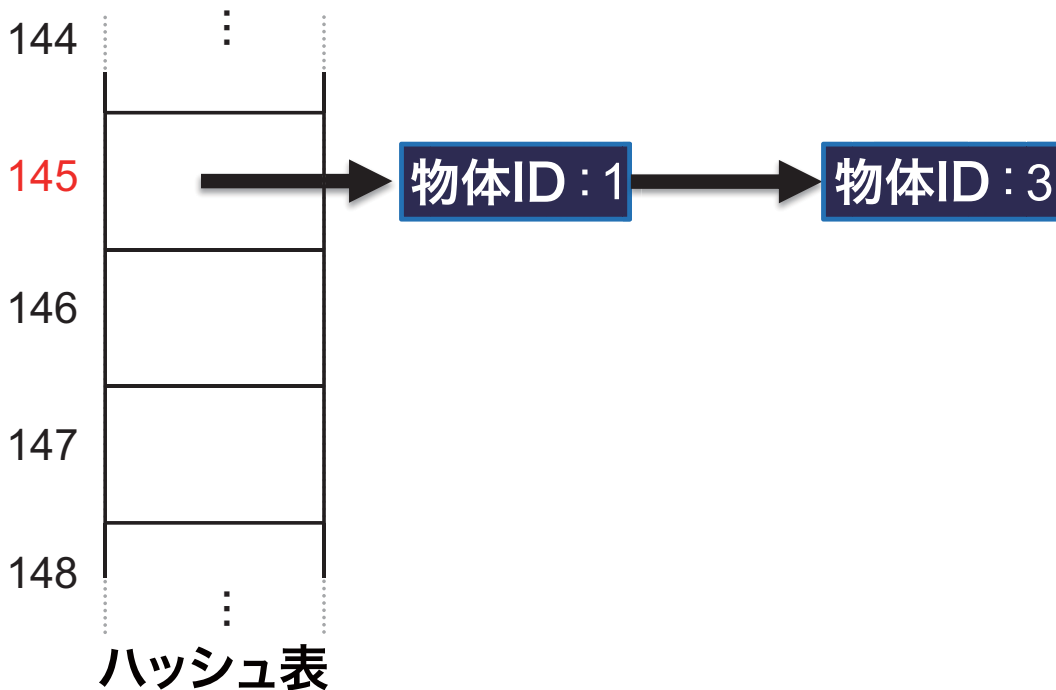


図 3.2: ハッシュ表へのデータ登録 (ハッシュ(距離計算なし))

特徴ベクトルから式 (2.2) で計算されたハッシュ表のインデックスに、物体 ID と特徴ベクトルを登録するが、ハッシュ(距離計算なし) では、図 3.2 に示すように物体 ID のみ登録する。これにより、特徴ベクトルの登録に必要なメモリ使用量を削減している。

二点目の違いは、特徴ベクトルの照合方法である。ハッシュ(距離計算あり) では、特徴ベクトルからハッシュ表のインデックスを求め、このインデックスに登録されている特徴ベクトルとの距離を計算することで、特徴ベクトルを照合する。そして、これらの中でクエリとなる特徴ベクトルの最近傍となるものを求め、この特徴ベクトルに対する物体 ID を照合結果として得る。一方ハッシュ(距離計算なし) では、式 (2.2) で求まるハッシュ表のインデックスが一致するかで特徴ベクトルを照合する。つまりこの手法では、求まったハッシュ表のインデックスに登録されている全ての物体 ID を照合結果として得る。

三点目の違いは、特徴ベクトルの照合結果の統合方法である。ハッシュ(距離計算あり) では、第 2.3.2 節で述べたように特徴ベクトルの照合結果を投票処理によって統合する。ハッシュ(距離計算なし) でも、統合の基本的な考え方は投票処理である。ハッシュ(距離計算あり) との違いは、照合結果によって得られた全ての物体 ID を投票処理に用いることである。しかし、単純な投票処理では、得られる特徴ベクトルの数が少ない物体が認識

できないという問題が生じる．これは，データベースに登録される特徴ベクトルの数は物体ごとに異なるため，物体によってハッシュ表から物体 ID が検索される確率も異なることによって生じる．そこで，ハッシュ(距離計算なし)では，この問題に対処するために次のようにスコアを更新することで，照合結果を統合する．

$$s_i \leftarrow s_i + \frac{1}{\sqrt{\eta_i}} \quad (3.1)$$

ここで， s_i は物体 ID i の物体のスコア， η_i は物体 ID i の物体から得られる特徴ベクトル数である．式 (3.1) により， L_2 ノルムで正規化した値に近似できる^[85]．以上の処理を検索質問画像から得られる全ての特徴ベクトルに対して行い，最終的に最もスコアの高い物体を認識結果とする．

ハッシュ(距離計算なし)では，認識率を確保するために，ビットベクトルの次元数 d の値を大きくする必要がある．しかし， d の値を大きくすれば，ハッシュ表の大きさも指数的に増加し，ハッシュ表に必要なメモリ使用量が大きくなるという問題がある．また，5千個の二次元物体データセットから得られる約1千万個の PCA-SIFT 特徴ベクトルを，ハッシュ(距離計算なし)で $H_{\text{size}} = 2^d$ のハッシュ表にマッピングした予備実験より，91%以上の二次元物体の認識率を得るのに $d = 24$ ，約95%の認識率を得るのに $d = 28$ のハッシュ表が必要であることが分かった．このとき， $d = 24$ で65%以上， $d = 28$ で96%以上のハッシュインデックスが，一つの特徴ベクトルも登録されていない，またはリストが削除された状態となっていた．このことから，ハッシュ(距離計算なし)には，ハッシュ表の空間効率が悪いという問題点がある^{[76],[77]}．

一方， d を小さくすると，メモリ使用量を削減できるものの，異なる特徴ベクトルが同じビットベクトルに変換される確率が高くなるため，特徴ベクトルの識別能力が失われるという問題が生じる．また，ビットベクトル作成に用いる次元数を $d = 28$ ，ハッシュ表の大きさを $H_{\text{size}} = 2^{24} - 1$ とした予備実験より， $d = 24$ ， $H_{\text{size}} = 2^{24}$ としたものよりも，認識精度が低下することも分かった^[76]．このことから，ハッシュ(距離計算なし)では効率よく特徴ベクトルの情報を保存することが難しいといえる．

以上の処理で識別能力が問題となる一つの原因は，ハッシュ表から検索された物体 ID 全てに対して式 (3.1) でスコアを更新する点にある．正しい対応は一つであるため，良いものを選択する仕組みを導入すれば識別性が向上する．ハッシュ(距離計算あり)では，元の特徴ベクトルの情報を保持し，ハッシュ表から検索質問となる特徴ベクトルの最近傍となるものを一意に求めることができる．ハッシュ(距離計算あり)では，特徴ベクトルを全

て保持するためメモリ使用量は大きいですが、精度が高まるという利点が得られる。本研究では、この手法も比較対象とする。

3.3 提案手法

本節では、Bloomier Filter を用いた特定物体認識手法について述べる [64]–[67]。提案手法でも、文献 [62] の手法と同様に、画像から PCA-SIFT [21] によって求めた特徴ベクトルを用いる。以下では、まず、提案手法に用いる Bloomier Filter [63] と、その基礎となる Bloom Filter [78] について説明する。そして、提案手法の具体的な処理について説明する。

3.3.1 Bloom Filter

Bloom Filter [78] は、平衡 2 分探索木やハッシュ表と比べて空間効率の良い確率的データ構造であり、あるデータ集合と要素が与えられたとき、この要素がデータ集合のメンバーであるか否かを調べるために用いられる。この手法には、ある要素がデータ集合のメンバーではないにもかかわらず、集合のメンバーであると判断される偽陽性 (False Positive) が生じるという問題点や、データ集合から元の要素を取り出すことができないという問題点がある。一方、この手法には、メモリ使用量を増やすことなくデータ集合にいくらかでも要素を追加することができるという利点がある。もちろん要素を追加すればするほど偽陽性の可能性は高くなるが、同じメモリ使用量で保存できる要素数は、平衡 2 分探索木やハッシュ表と比べると圧倒的に多い。本研究では、この空間効率の良さを利用し、Bloom Filter を特徴ベクトルの記録に用いる。本論文では、以後データ集合の要素を特徴ベクトルとして話を進める。以下に、Bloom Filter への特徴ベクトル登録方法と特徴ベクトルの認識方法の具体的な処理について説明する。

図 3.3 に特徴ベクトルの登録処理の流れを示す。まず、空の Bloom Filter として m bit の配列を用意し、全ての bit を 0 で初期化する。以下では、 m を Bloom Filter の “TableSize” と呼ぶこととする。次に、 k 個のハッシュ関数を用意し、それぞれのハッシュ関数に対して、特徴ベクトルをキーとしてハッシュ値を計算する。ここで、得られるハッシュ値は、 $1 \sim m$ の整数値とする。そして、得られた k 個のハッシュ値 h_1, h_2, \dots, h_k を基に、Bloom Filter の $h_i (i = 1, 2, \dots, k)$ 番目の bit を 1 にすることで特徴ベクトルを登録する。特徴ベ

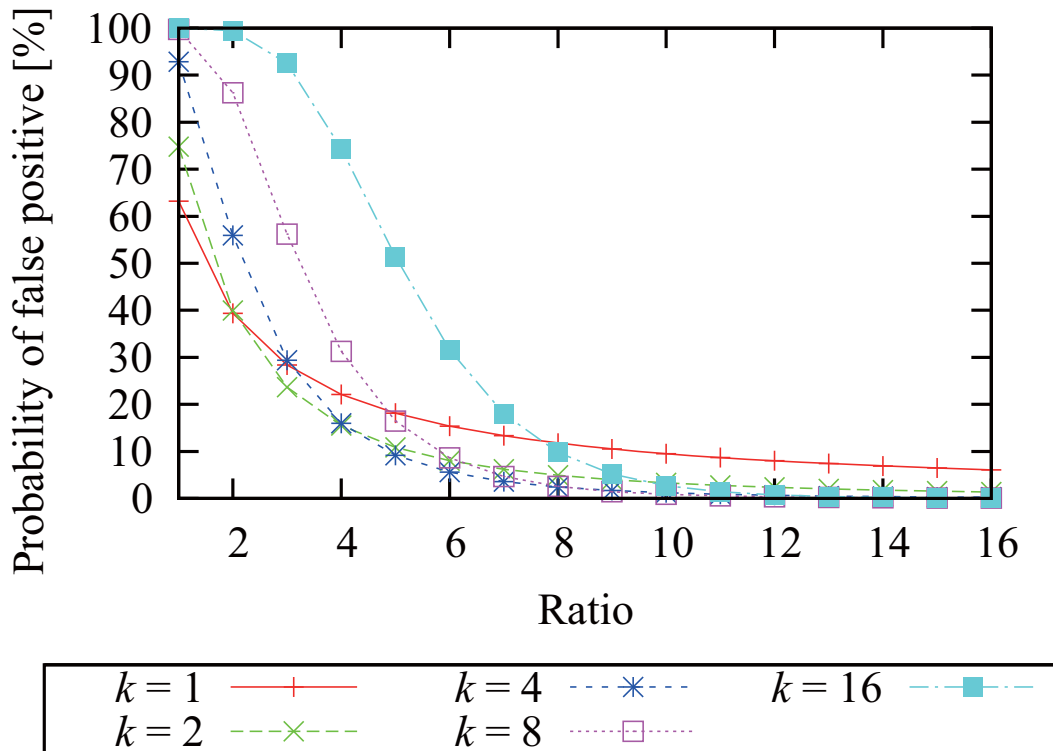


図 3.4: 偽陽性の発生確率 $P_{fp}(k, m, n)$

3.3.2 Bloomier Filter

Bloomier Filter^[63] は、複数の Bloom Filter を用いることで、登録されている特徴ベクトルを符号化し、符号に関連する値を連想させるこのとのできるデータ構造である。この手法にも、偽陽性の可能性があるという問題点や、元の特徴ベクトルをデータ集合から取り出すことができないという問題点があるものの、ハッシュ表などと比べると、圧倒的に空間効率が良いという利点がある。以下に、Bloomier Filter の動作について説明する。

簡単のために、Bloomier Filter で連想させる値が“0”と“1”の二種類のみについて説明する。まず、二つの Bloom Filter $B^{(0)}$ と $B^{(1)}$ を用意する。次に、特徴ベクトルに連想させたい値が“0”である場合、Bloom Filter $B^{(0)}$ に特徴ベクトルを登録しておく、連想させたい値が“1”である場合、Bloom Filter $B^{(1)}$ に特徴ベクトルを登録しておく。これにより、ある特徴ベクトルを認識させたとき、 $B^{(0)}$ に登録されていた場合、その特徴ベクトルから連想させる値は“0”である可能性が高いと言え、また、 $B^{(1)}$ に登録されていた場合、連想される値は“1”である可能性が高いと言える。

Bloomier Filter に任意の数字を連想させる場合，連想させる数字を二進数表現し，桁毎に二つの Bloom Filter を用意すれば良い．これにより， N 種類の数字を識別するために連想させたい数字ごとに Bloom Filter を用意すると， N 個の Bloom Filter が必要となるが，Bloomier Filter を用いることで， $2 \log_2 N$ 個の Bloom Filter でこれらの数字を表現することができる．

3.3.3 符号の定義

提案手法では，Bloomier Filter を用いて，特徴ベクトルを符号化し，この符号を基に物体 ID を連想させることを考える．これを実現するには，まず物体 ID と符号との対応関係を定義する必要がある．提案手法では，物体 ID を二進数表現したものを符号とする．以後，この二進数表現した物体 ID を “*bit ID*” と呼ぶ．また， β_i^j を物体 ID i の “*bit ID*” における j 番目の桁の値とする．

3.3.4 符号化関数の作成

まず，文献 [62] の手法と同様に PCA-SIFT で求めた特徴ベクトルに対して，式 (2.1) よりビットベクトル u を求める．そして，物体 ID を ν 桁の *bit ID* に変換し，桁ごとに二つの Bloom Filter を用意する．符号 “0” を記録する Bloom Filter を $B_1^{(0)}, B_2^{(0)}, \dots, B_\nu^{(0)}$ ，符号 “1” を記録する Bloom Filter を $B_1^{(1)}, B_2^{(1)}, \dots, B_\nu^{(1)}$ とする．また， n_κ^τ ($\kappa \in \{1, 2, \dots, \nu\}, \tau \in \{0, 1\}$) を “*bit ID*” の第 κ 桁が τ となる物体から得られる特徴ベクトルの総数とすると，

$$n_\kappa^\tau = \sum_{i=0}^{N-1} \gamma \eta_i, \quad \gamma = \begin{cases} 1 & \text{if } \beta_i^\kappa = \tau \\ 0 & \text{otherwise} \end{cases} \quad (3.3)$$

となる． η_i は，物体 ID i の物体から得られる特徴ベクトル数を意味する．ここで，式 (3.2) より，Bloom Filter に登録される特徴ベクトル数に対して TableSize と偽陽性の間にはトレードオフの関係があることに着目する．この関係性より，偽陽性の発生を抑制するために，式 (3.3) を用いて各 Bloom Filter の TableSize m_κ^τ を以下の式で与える．

$$m_\kappa^\tau = a \times n_\kappa^\tau + \alpha_\kappa^\tau \quad [\text{bit}] \quad (3.4)$$

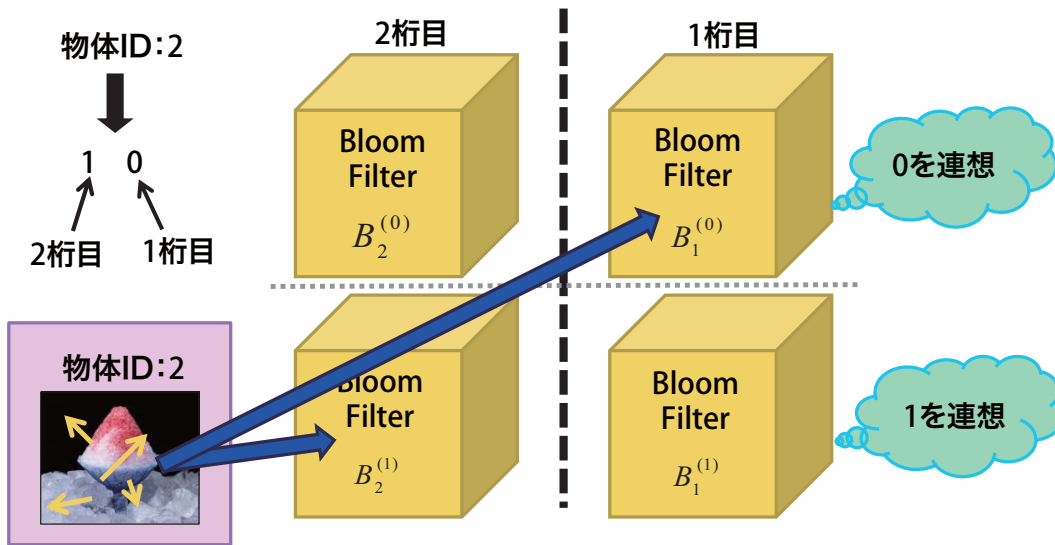


図 3.5: Bloomier Filter への登録方法

ここで, a は, TableSize の大きさ及び, 偽陽性の発生確率 (式 (3.2)) を左右するパラメータである. また, α_k^τ は, TableSize m_k^τ が $a \times n_k^\tau$ より大きな最小の素数となるための定数である.

次に, 物体 ID を連想させるために, ビットベクトルを ν 個の Bloom Filter に登録する. このとき, 物体 ID i の bit ID の j 番目の桁が $\beta_i^j = 0$ の場合 $B_j^{(0)}$ に, $\beta_i^j = 1$ の場合 $B_j^{(1)}$ にビットベクトルを登録する. 例えば, 認識対象物体が四つの場合, すなわち, 物体 ID を 2 bit で表現する場合は以下のように登録する. 図 3.5 に示すように, 物体 ID の 2 を “10” で表現したとすると, bit 列の一桁目が “0”, 二桁目が “1” であるため, $B_1^{(0)}$ と $B_2^{(1)}$ にそれぞれ物体 ID 2 の物体から得られる特徴ベクトルを登録する. 提案手法では, 登録の際に文献 [62] の手法と同様に, PCA-SIFT で求めた特徴ベクトル p の第 1 次元から第 d 次元を用いて, 式 (2.1) により, ビットベクトル $u = (u_1, u_2, \dots, u_d)$ に変換し, このビットベクトル u を各 Bloom Filter に登録する. 但し, データベースに登録する特徴ベクトル全てをビットベクトルで表現したとき, 同じビットベクトルが閾値 c 個以上あれば, 提案手法でも文献 [62] の手法と同様に, そのビットベクトルに変換される特徴ベクトルは, 認識にはあまり有効でないと考え, データベースに登録しない. それ以外の場合は, 求めたビットベクトル u をキーとし, k 個のハッシュ関数を用いて, Bloom Filter のどのビットを 1 にするかを決める. 提案手法では, Bloom Filter への登録処理に用いるハッシュ関数を以下のように構成する. まず, ビットベクトルから MD5 値を計算し, 次に, 得られた 128bit

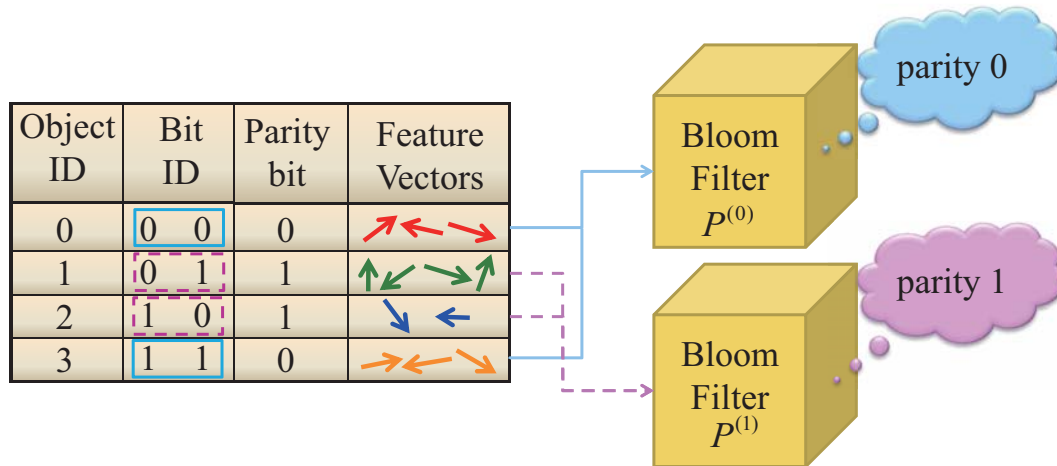


図 3.6: 誤り検出用 Bloomier Filter の作成方法の略図

のビット列を 4 分割することで $k = 4$ 個のハッシュ関数を用意する．つまり，各ハッシュ値 $h_i (i = 1, 2, 3, 4)$ は，32bit の値をとる．このとき，ハッシュ値が TableSize より大きくなる場合は，TableSize を法とする剰余算により，値を TableSize の範囲に収める．

3.3.5 パリティビットに基づく誤り検出用 Bloomier Filter の作成

提案手法では，文献 [62] の手法と異なり，特徴ベクトルを保存するのに Bloom Filter を用いているため，偽陽性が発生する可能性がある．提案手法では，この偽陽性の影響で，誤った物体 ID を連想してしまう可能性があり，ハッシュ表を用いる手法よりも認識精度が低下する問題がある．この問題に対処するために，本研究では誤り検出符号であるパリティビットの概念を導入する [79]．提案手法では，この概念に基づき，偽陽性が認識精度に与える影響を少なくするために，パリティビットを連想させる Bloomier Filter を作成し，認識処理に用いる．これは，認識処理で求めた物体 ID が正しい ID であるかを調べるための簡易的な誤り検出器である．この誤り検出用 Bloomier Filter の作成方法は以下の通りである．

図 3.6 に誤り検出用 Bloomier Filter の作成方法の略図を示す．まず，二つの Bloom Filter $P^{(0)}$, $P^{(1)}$ を用意する．そして，物体 ID を二進数表現したとき，1 の数が偶数個の物体 ID を持つ物体から得られる特徴ベクトルを $P^{(0)}$ に登録し，1 の数が奇数個の物体 ID を持つものから得られる特徴ベクトルを $P^{(1)}$ に登録する．それぞれの Bloom Filter の TableSize

は，式 (3.4) と同様に与えられる．

3.3.6 認識処理

まず，提案する物体認識手法の処理の流れを説明する．この手法では，検索質問画像から得られる特徴ベクトル q を用い，どの Bloom Filter に q が含まれているかを調べ，物体 ID を求める．そして求めた物体 ID の物体に対して式 (3.1) によりスコアを増加させる．但し，誤り検出 Bloomier Filter で正しいと判断された物体 ID に対してのみスコアを増加させる．この処理を検索質問画像から得られる特徴ベクトル全てに対して行い，最大スコアの物体を認識結果とする．以下に，具体的な処理について説明する．

検索質問画像から得られる特徴ベクトルは，一般に各次元の値がデータベース作成に用いる特徴ベクトルのものと異なったものとなる．このため提案手法でも，文献 [62] の手法と同様に，各次元の値の許容変動幅 e を設け， $\epsilon_j = |q_j - \theta_j| \leq e$ を満たす次元 j に対して，変動に対処する．また，この処理を適用する次元数を閾値 b 個以下に制限し，次元数が b を上回る時には， ϵ_j の小さな次元から b 個採用する．そしてこれらのビットベクトルを用いて， $bit ID$ の i ($i = 1, 2, \dots, n$) 番目の桁が 0 か 1 かを求める．このとき，Bloom Filter $B_i^{(0)}$ に登録されていれば，物体 ID の i 番目の桁を 0 とし，Bloom Filter $B_i^{(1)}$ に登録されていれば， $bit ID$ の i 番目の桁を 1 とする． $B_i^{(0)}$ と $B_i^{(1)}$ のいずれにも登録されていない場合，この特徴ベクトルはデータベースに登録されていないとし， i 番目以降の処理を打ち切る．この処理では， $B_i^{(0)}$ と $B_i^{(1)}$ の両方に登録されていた場合に問題が起こる．これは少なくともどちらかの Bloom Filter が偽陽性を引き起こし，実際には登録されていないにもかかわらず，登録されていると判断されるためである．提案手法では，このような場合両方の可能性を試すことにより対処する．具体的には， i 番目の桁が 0 となる $bit ID$ と 1 となる $bit ID$ の両方を認識処理に用いる．但し，全ての桁に対して上記の処理を適用すると，誤って作成される $bit ID$ が増加するため，提案手法では，偽陽性の発生を容認する $bit ID$ の桁数を閾値 t 個に限定する．偽陽性の発生する桁数が t 個以上の場合，その特徴ベクトルは認識には寄与しないと考え，処理を打ち切る．

次に，上記の処理により求めた $bit ID$ の物体に対して，誤り検出 Bloomier Filter を用い，実際にスコアを増加させるかどうかを決める．まず， P_0 と P_1 にクエリベクトルが登録されているか調べる．このとき，図 3.7(a) と (b) に示すように， $bit ID$ とパリティビッ

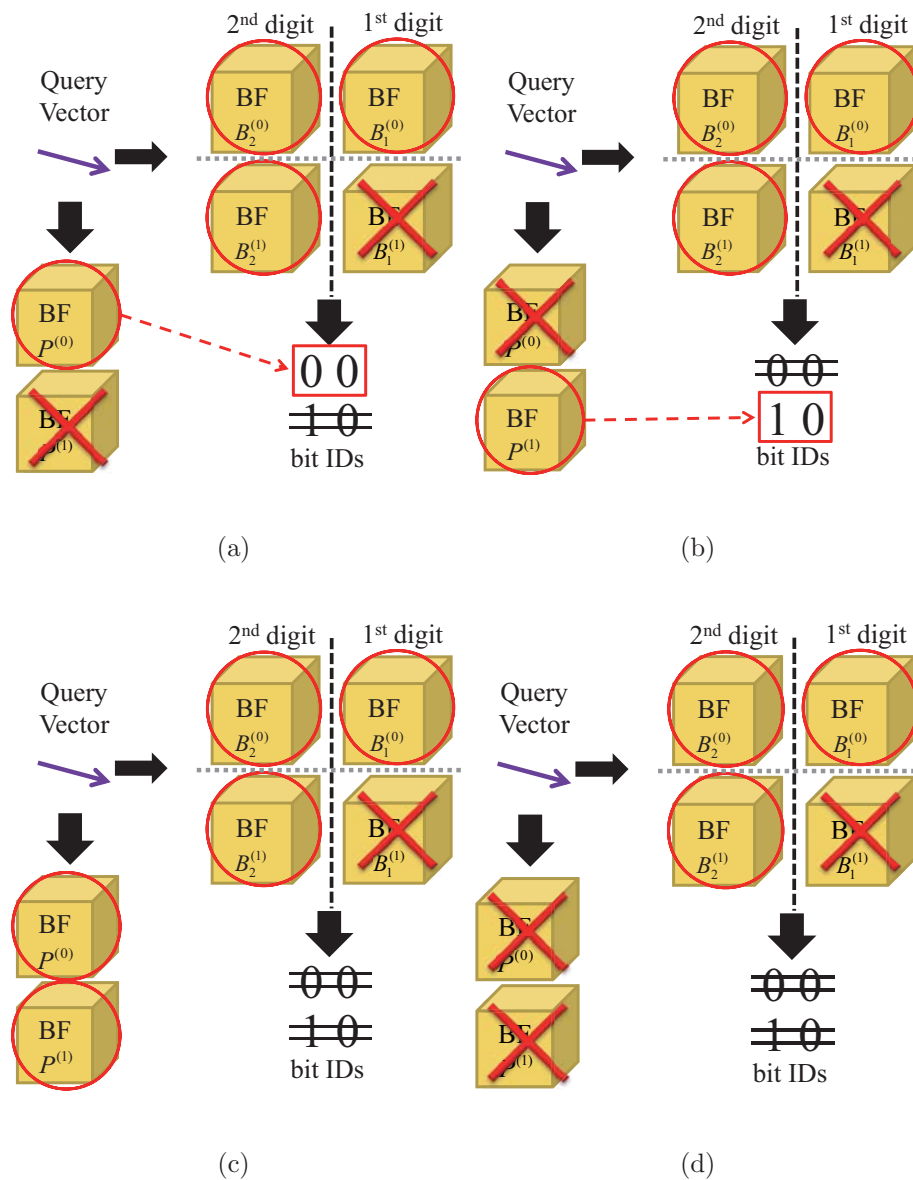


図 3.7: 誤り検出用 Bloomier Filter によるスコア更新 ID の選定． \circ はその Bloom Filter にクエリベクトルが登録されていることを意味し， \times は登録されていないことを意味する．

トの組合せが正しいもののみ，式 (3.1) によりスコアを更新する．また，図 3.7(c) のように P_0 と P_1 の両方に登録されていると判断された場合や，図 3.7(d) のように， P_0 と P_1 の両方に登録されていないと判断された場合では，求めた $bit ID$ は誤りと判断し，破棄する．この処理を検索質問画像から得られる特徴ベクトル全てに対して行い，最終的に最もスコアの高い物体を認識結果とする．

また提案手法では，認識処理の効率化を図るために，文献 [62] の手法と同様に，第 l 段が $b = l - 1$ とした識別器を用いて認識する．これによって，他の物体とのスコアの差が十分に開いていれば，途中の段で処理を打ち切ることができるため，認識処理を効率的に行うことができる．

3.4 1万個の二次元物体を用いた実験

本節では，提案手法の有効性を確かめるために，1万個の二次元物体のデータセットに対して行った実験の結果について述べる．

3.4.1 実験条件

1万個の二次元物体のデータセットについて説明する．図 3.8 に，用いた二次元物体の例を示す．本実験では，Google イメージ検索において，“雑誌”，“ポスター”，“表紙”等の検索キーワードで収集した 3100 枚の画像 (図 3.8 の (a)) と，PCA-SIFT のサイトで公開されていた 3450 枚の画像 (図 3.8 の (b)) と，写真共有サイトの Flickr [86] において，“animal”，“birthday”，“food”，“japan”等のタグにより収集した 3450 枚の画像 (図 3.8 の (c)) をデータベース作成用画像とした．なお，収集の際には， 600×600 pixel 以下のサイズの画像は除外し，画像の長辺が 640pixel 以下になるように縮小した．データベース作成用画像から得られる特徴ベクトルの数は，約 2 千万個である．検索質問画像としては，次のように用意した．まず，データベースに含まれる画像の内，Google イメージ検索で収集した画像から 100 枚，PCA-SIFT のサイトから得た画像から 200 枚，Flickr から収集した画像から 200 枚の合計 500 枚を無作為に選択した．次に，これらの画像を A4 の用紙にカラーレーザープリンタ¹を用いて印刷し，カメラ²で撮影した．撮影した画像の例を図 3.9 に示す．図 3.9 に示す通り，紙面全体が写る配置で，紙面に対するカメラの光軸の角度 θ を 90° ， 75° ， 60° に変化させた．また，角度を 90° として紙面の一部を撮影した．その結果，一枚の紙面に対して，合計四通りの画像を得た．さらに，撮影した画像を 512×341 pixel に縮小し，PCA-SIFT により特徴ベクトルを求めた．その結果，画像一枚あたり平均で

¹OKI C5200n

²CANON EOS Kiss Digital，630 万画素，付属レンズ：EF-S 18-55mm USM



(a)



(b)



(c)

図 3.8: 登録画像の例

約 630 個の特徴ベクトルが得られた．以下で述べる実験に用いた計算機は，CPU が AMD Opteron8378 2.4GHz，メモリが 128GB RAM のものである．また，実験結果で示す物体の認識率は，式 (2.5) で定義される．

3.4.2 従来手法との性能比較

まず，提案手法と従来手法の性能を比較する．本実験では，以下の三手法を比較に用いる．すなわち，文献 [52] で提案されている木構造を用いた Vocabulary Tree (以後 V-Tree と表記する)，文献 [62] で提案されている特徴ベクトルの距離計算をしない手法 (ハッシュ

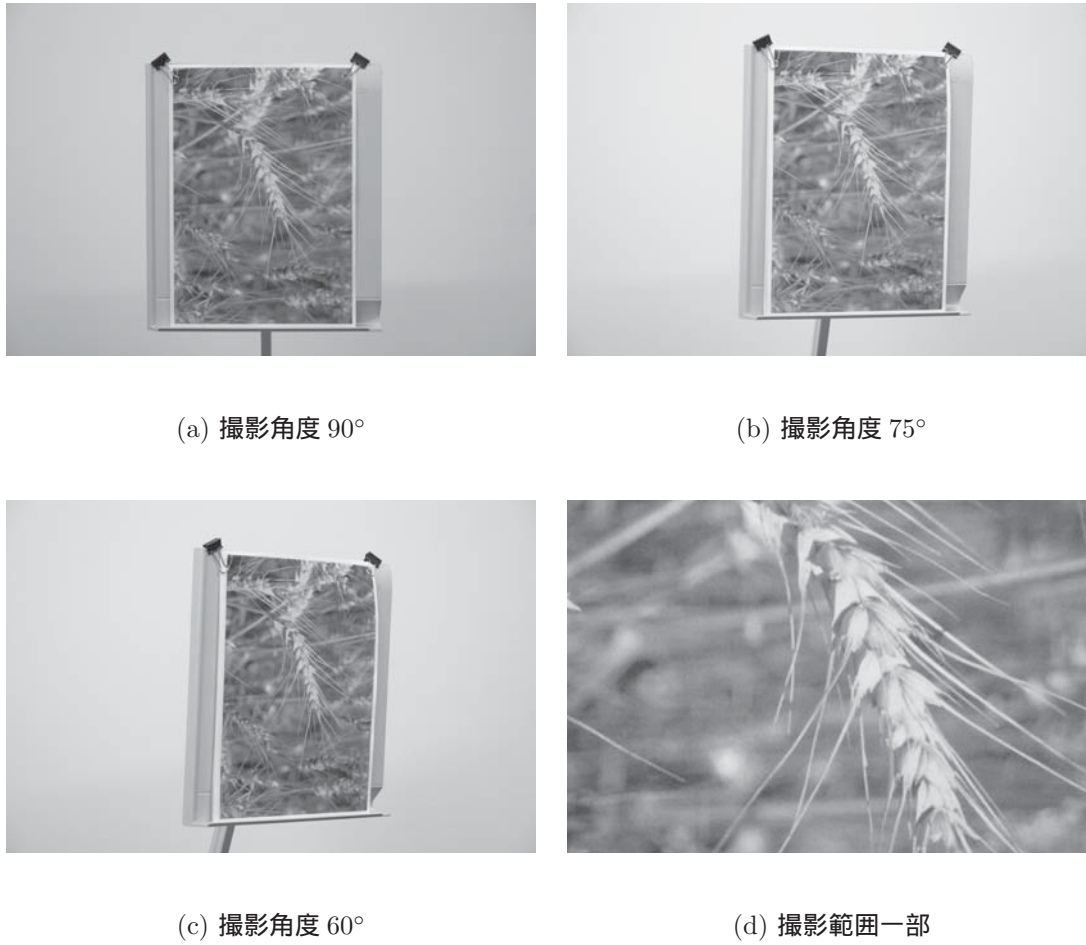


図 3.9: 質問質問の例

(距離計算なし)), ならびに距離計算を行う手法 (ハッシュ(距離計算あり)) である. 本実験では, V-Tree で用いる正規化には, L_2 -ノルムを用い, ハッシュ(距離計算あり・なし) で用いるハッシュ表のサイズを $H_{\text{size}} = 2^d$ とした. ここで, 以下の実験結果で示す処理時間は, 一枚の検索質問画像の認識にかかった平均の処理時間であり, 特徴ベクトルの抽出時間等は含まない. ハッシュに基づく比較手法および提案手法で用いるパラメータ d, b, e, c を確認しておく. ビットベクトル変換に用いる次元数 d , ビットベクトルで反転の操作を行う次元数 b , 値の変動幅 e , 重複するビットベクトルの上限数 c の四つである. 本実験では, $d = 22, 24, 26, 28, 30, b = 0, 1, \dots, 10, e = 200, c = 1, 2, \dots, 10$ とした. また, 提案手法ではこれらに加えて偽陽性の発生を容認する *bit ID* の桁数 t を用いており, $t = 0, 1, \dots, 5$ とした. V-Tree で用いたパラメータは表 3.1 に示すものである.

表 3.1: 実験に用いた V-Tree(Vocabulary Tree) のパラメータ (1万個の二次元物体)

	分岐数 x	深さ y	分岐数 x	深さ y
10		4	11	6
		5	12	
		6	13	
		7	14	

表 3.2: 各手法のパラメータ (1万個の二次元物体)

手法	パラメータ
提案手法	$b = 7, c = 7, d = 30, e = 200, t = 1$
V-Tree(同認識率)	$x = 14, y = 6$
V-Tree(最大認識率)	同上
ハッシュ(距離計算なし)(同認識率)	$b = 4, c = 3, d = 26, e = 200$
ハッシュ(距離計算なし)(最大認識率)	$b = 5, c = 4, d = 26, e = 200$
ハッシュ(距離計算あり)(同認識率)	$b = 6, c = 5, d = 22, e = 200$
ハッシュ(距離計算あり)(最大認識率)	$b = 8, c = 8, d = 24, e = 200$

提案手法で最も認識率の高かった結果と、三手法の結果のうち、それぞれ提案手法と同程度の認識率になるものと、最も認識率の高かったものを表 3.2 と表 3.3 にまとめる。ここで、表 3.2 に各手法のパラメータ、表 3.3 に各手法の結果についてそれぞれ示す。また、これらの表では、提案手法と同程度の認識率となる結果を“同認識率”、最も認識率の高かった結果を“最大認識率”と表記する。

結果より、認識率が同程度のとき、提案手法が最も少ないメモリ使用量で物体を認識できることが分かった。特に、V-Tree と比較すると、提案手法の方が処理時間がかかるものの、認識率、メモリ使用量の点で大きく有効性があると言える。V-Tree で誤認識した物体の特徴として、得られる特徴ベクトルの数が少ないことが挙げられる。この特徴より、得られる特徴ベクトルの数が少ないにもかかわらず、複数の特徴ベクトルをクラスタリングなどによって一つの Visual Word にまとめてしまうことが、認識率低下の原因と考えられる。このことから、特定物体の認識には、ある程度元の特徴ベクトルを保存しておく必要があると言える。

表 3.3: 各手法の性能評価 (1万個の二次元物体)

手法	認識率 [%]	処理時間 [ms]	メモリ使用量 [MB]
提案手法	95.15	16.37	415
V-Tree(同認識率)	88.95	8.75	3017
V-Tree(最大認識率)	同上	同上	同上
ハッシュ(距離計算なし)(同認識率)	95.15	0.56	1174
ハッシュ(距離計算なし)(最大認識率)	95.75	0.66	1200
ハッシュ(距離計算あり)(同認識率)	95.15	1.22	1480
ハッシュ(距離計算あり)(最大認識率)	98.45	1.14	2210

次に、ハッシュ(距離計算なし)およびハッシュ(距離計算あり)と比較すると、認識率が同程度のとき処理時間では大きく劣るものの、メモリ使用量をそれぞれ約 $1/3$ 、約 $2/7$ に削減することができた。処理時間が大きくかかる原因として、提案手法では物体 ID を求めるために 2ν 個の Bloom Filter を調べなければならないことが挙げられる。このことから、Bloom Filter の検索処理を並列化するなどといった高速化処理を導入することが、今後の課題の一つとして挙げられる。また、ハッシュを用いる手法と比較すると、最大認識率では若干認識精度が低下することが分かった。これは、Bloom Filter で発生する偽陽性が原因として考えられる。

以上の結果をまとめると、若干の認識率の低下が許容できるのであれば、提案手法を用いるとメモリ使用量を抑えることができると言える。また、処理時間で比較すると、提案手法は従来手法と比べると大きく劣るものの、 $16[\text{ms}]$ で 95% 以上の認識率を得ることができた。このことから、緒論で述べたような携帯端末で物体を撮影し、認識することで、その物体に関する情報を手に入れるようなサービスでは、提案手法でも十分リアルタイム処理が可能であると考えられる。また、このようなサービスでは、より多くの物体を認識することが望まれるため、従来手法よりも認識に必要なメモリ使用量が少ない提案手法を用いる方が、より多くの物体を認識できると考える。

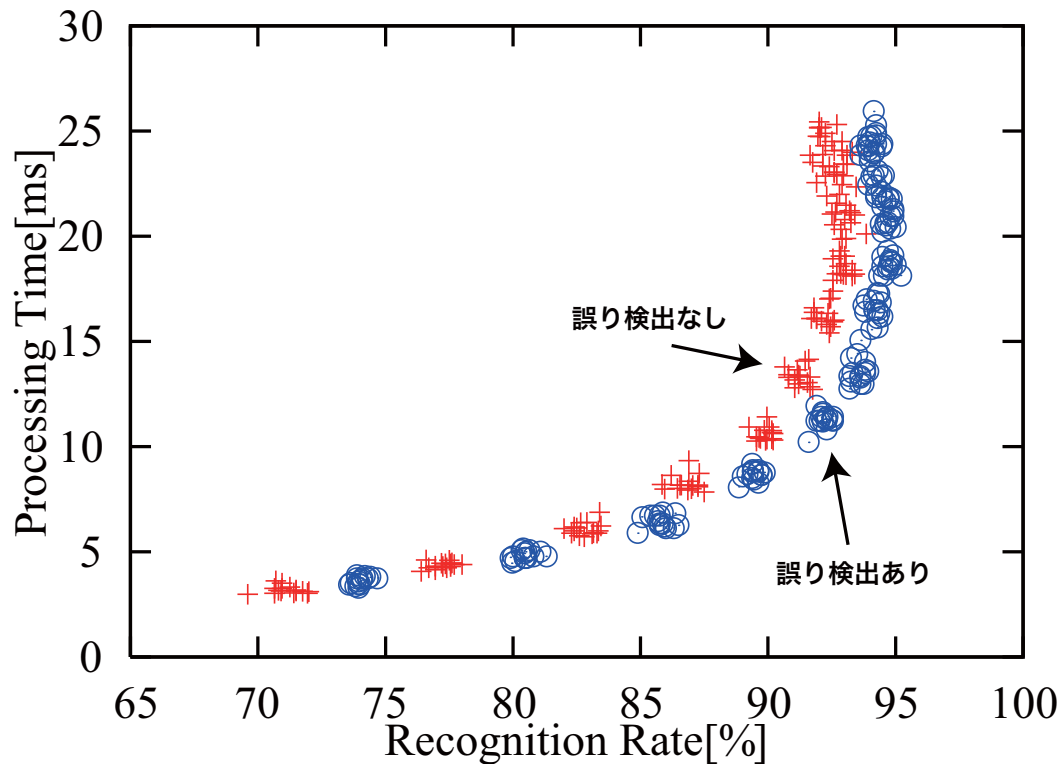


図 3.10: 誤り検出用 Bloomier Filter の有無による性能比較

3.4.3 誤り検出用 Bloomier Filter の有効性に関する実験

次に、誤り検出用 Bloomier Filter の有効性を調べるために、提案手法と誤り検出用 Bloomier Filter を用いない手法の性能を比較する。誤り検出用 Bloomier Filter を用いない手法では、Bloomier Filter より求めた物体 ID の物体全てに対してスコアを加算する。本実験では、提案手法のパラメータを $b = 0, 1, \dots, 10$, $c = 1, 2, \dots, 15$, $d = 30$, $e = 200$, $t = 0, 1, \dots, 5$ と変化させた。また本実験では、誤り検出用 Bloomier Filter を用いない手法と同程度のメモリ使用量となるように、提案手法のパラメータ a を調整した。図 3.10 に、提案手法で最も認識精度の高かった $t = 1$ と、誤り検出用 Bloomier Filter を用いない手法で最も認識精度の高かった $t = 0$ の結果を示す。

結果より、提案手法は、全てのパラメータ設定において認識精度を向上できることが分かった。このことから、提案手法は、誤ったスコア加算を抑制することがある程度できていると言える。また、表 3.4 に、両手法で最も高い認識精度を得ることのできるパラメータ設定において、正しく認識できた検索質問画像の枚数と、誤って認識された検索質問

表 3.4: 誤り検出用 Bloomier Filter の有無による検索質問画像の正誤枚数

		誤り検出あり	
		correct	false
誤り検 出なし	correct	1867	10
	false	37	86

画像の枚数を示す。結果より、提案手法で認識できなくなった検索質問画像が 10 枚あることが分かった。これは、誤り検出用 Bloomier Filter でも偽陽性が発生し、誤った物体 ID だけでなく正しい物体 ID も認識処理から除外してしまっていることが原因であると考えられる。このことから、さらに認識精度を向上させるためには、誤り検出用 Bloomier Filter で連想するパリティビットの数を増加させたり、誤り訂正符号を用いるなどで対処する必要があると考える。

3.4.4 偽陽性の発生を容認する *bit ID* の桁数 t に関する実験

提案手法では、特徴ベクトルの照合に距離といった量的な概念に基づく類似検索ではなく、登録されている Bloom Filter が同じかどうかという一致検索を用いている。このような単純な照合にもかかわらず、95%以上の認識率を得ることができる。提案手法では、特徴ベクトルの照合結果より求めた物体に対して式 (3.1) のようにスコアを増加させ、最終的に最大スコアを持つ物体を認識結果としている。そこで本研究では、提案手法で用いるスコアに着目し、提案手法が高い物体認識率を得る理由について調べる。提案手法では、偽陽性の発生を容認する *bit ID* の桁数 t や、ビットベクトルの反転操作を行う次元数 b によって、検索質問ベクトル q ごとに複数の物体 ID を求め、対応する物体のスコアを増加させる。しかし、ある q に対する正解物体は本来一つである。そこで本実験では、まず偽陽性に対処する桁数 t に着目し、複数の物体に対してスコアを増加させることが物体の認識率に与える影響について述べる。ここで、実際は物体毎に認識に必要な特徴ベクトルの照合精度は異なるが、本実験では、 t が認識精度に与える影響のみを調べるために、認識に必要な特徴ベクトルの照合精度は一定であるとして話を進める。すなわち、認識器の多段階化を用いず、最初から 2^b 個のビットベクトルを用いて認識する。

図 3.11 に前節までの結果で最も認識率の高かったパラメータ $b = 7, c = 7, d = 30,$

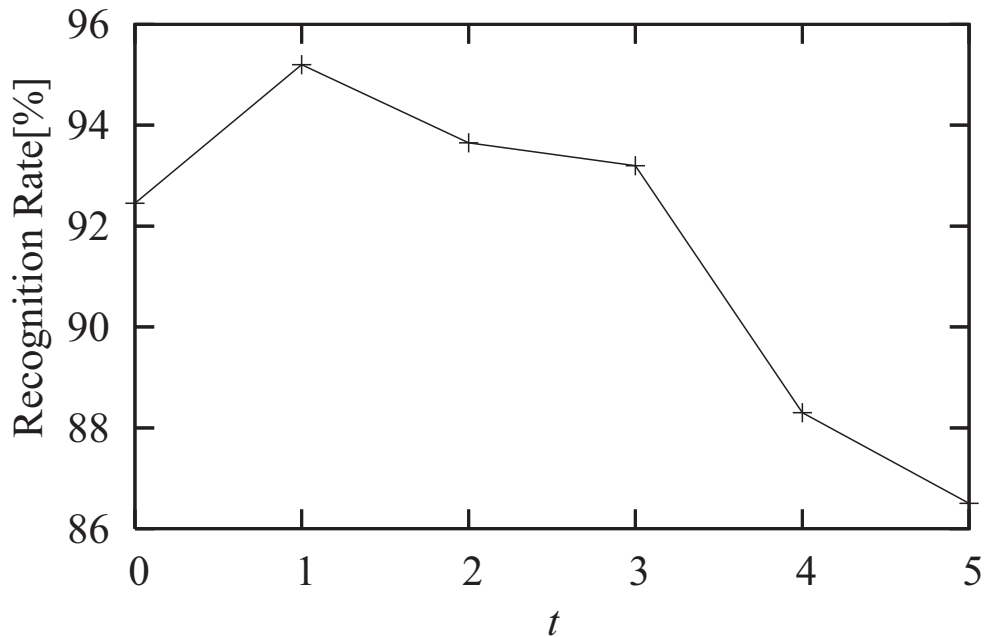


図 3.11: 偽陽性の発生を容認する $bit ID$ の桁数 t と認識率の関係

$e = 200$ を用い, t を $t = 0, 1, \dots, 5$ と変化させた場合の認識率を示す. 図 3.11 より, t を大きくすれば, 認識率が下がるということが分かる. これは, 正解物体のスコアを増加させる可能性は高くなるものの, それ以上に正解物体以外のスコアを増加させる可能性も高くなるためである. また図 3.11 より, $t = 1$ の時, 最も認識率が高いことから, 提案手法ではほとんど偽陽性に対処する必要がないと言える.

3.4.5 ビット反転の操作を行う次元数 b に関する実験

次に, ビット反転の操作を行う次元数 b に着目し, 複数の物体に対してスコアを増加させることが物体の認識率に与える影響について調べる. 本研究では, 提案手法を物体 ID 検索器に見立て, 物体 ID 検索の再現率 P_R (Recall) と適合率 P_P (Precision) の 2 つの指標を用いて b の影響を解析する. ここで再現率 P_R は, Ω を全検索質問ベクトル $\{q\}$ の総数, ω を提案手法によって検索された物体 ID に, 正しい物体 ID が含まれている q の数とすると,

$$P_R = \frac{\omega}{\Omega} \quad (3.5)$$

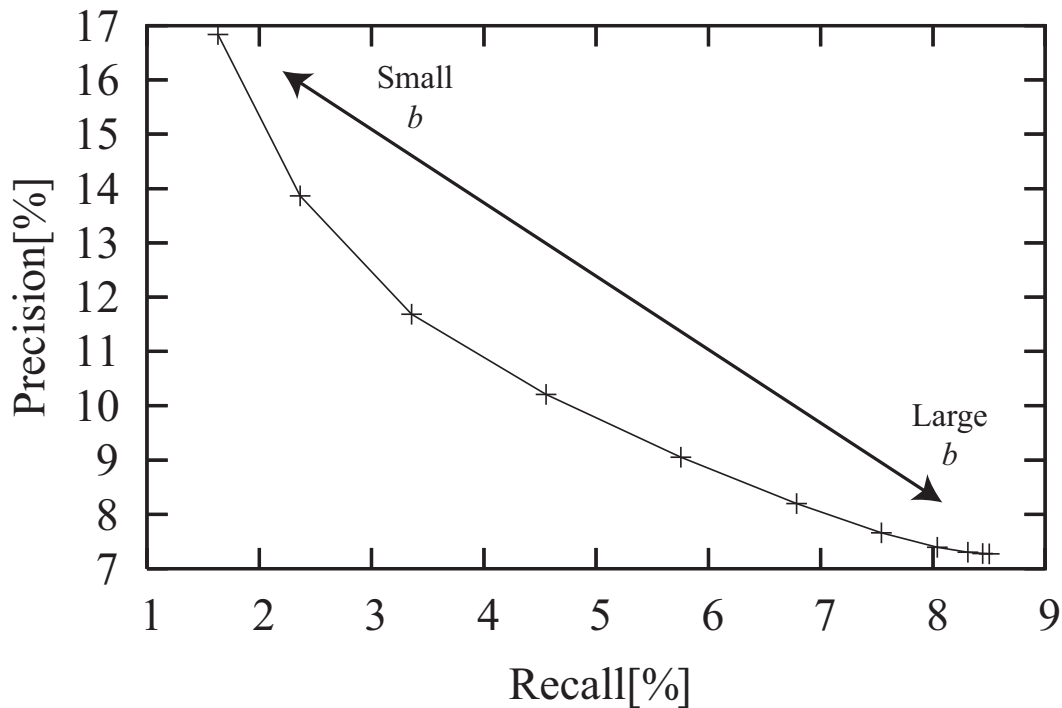


図 3.12: 再現率と適合率の関係

で定義される．また適合率 P_P は， Φ を q に対して検索された物体 ID の数， ϕ を検索された物体 ID 中に含まれている正しい物体 ID の数とすると，

$$P_P = \frac{\phi}{\Phi} \quad (3.6)$$

で定義される．以下の実験結果では，全検索質問ベクトル $\{q\}$ に対する適合率の平均を示す．本実験でも，簡単のため認識器の多段階化を用いないこととする．

図 3.12～図 3.14 に最も認識率の高かったパラメータ $c = 7$ ， $d = 30$ ， $t = 1$ を用い， b を変化させた場合の再現率，適合率，認識率の関係を示す．図 3.12 より， b を大きくすると再現率が高くなる，つまり正解物体の物体 ID が検索される可能性が高くなることが分かる．また，再現率が高くなると正解物体のスコアを増加させる回数が増えるため正解物体のスコアが最大スコアとなる可能性も高くなる．このため，図 3.13 から分かるように， b を大きくすると認識率が高くなると言える．図 3.15 に， b を $b = 0$ から最も認識率の高かった $b = 7$ へ増加させることで認識できるようになった検索質問画像の例を示す．しかし， b を大きくし過ぎると，正解物体以外のスコアの方が高くなり，若干認識率が下がる．これより， b を大きくするには限界があると言える．図 3.16 に， b を $b = 7$ から $b = 10$ へ

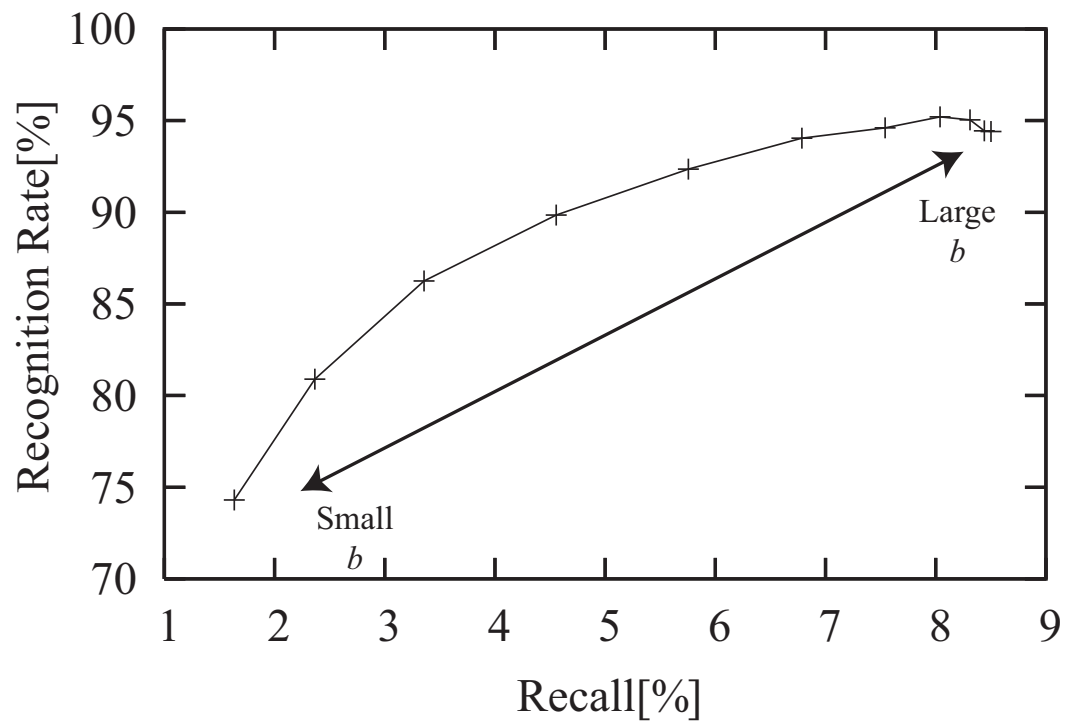


図 3.13: 再現率と認識率の関係

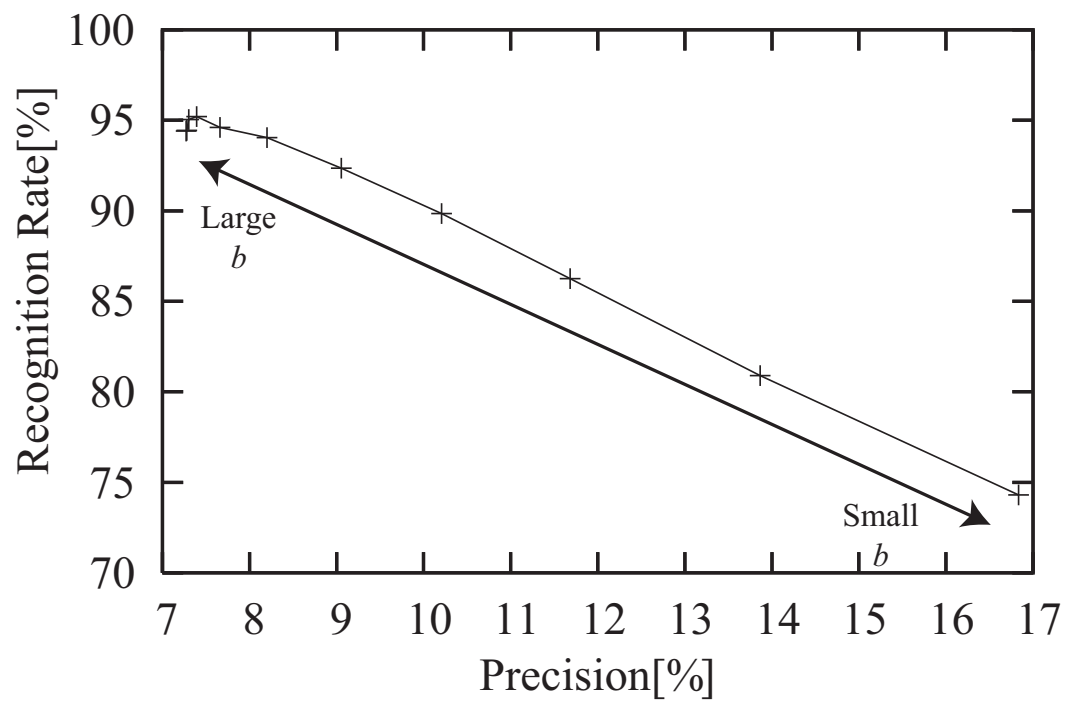


図 3.14: 適合率と認識率の関係

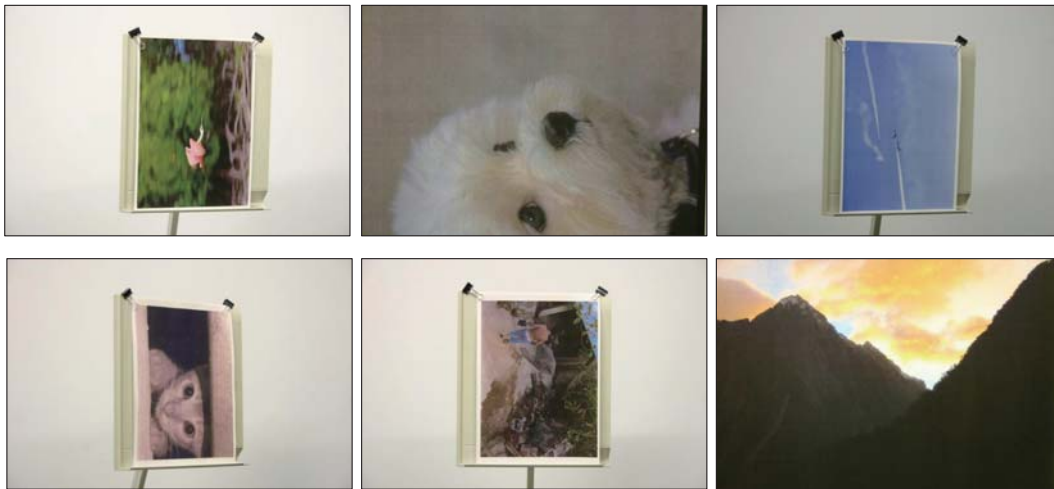


図 3.15: b を増加させることで認識できるようになった検索質問画像の例

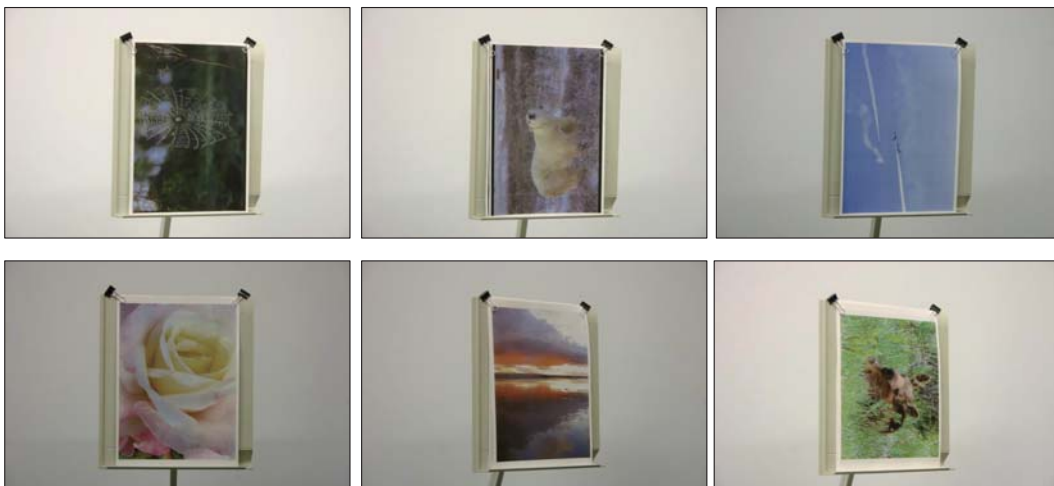


図 3.16: b を増加させることで認識できなくなった検索質問画像の例

増加させることで認識できなくなった検索質問画像の例を示す．また，図 3.17 に， b を増加させても全く認識できなかった物体と検索質問画像の例を示す．図 3.16 や図 3.17 に示す検索質問画像の特徴としては，得られる特徴ベクトルの数が少ないということが挙げられる．さらに図 3.12 より， b を大きくすると再現率は高くなるものの，正解物体以外の物体 ID が検索される可能性も高くなるため，適合率は低下しているのが分かる．

次に図 3.14 に着目すると， b を大きくすると適合率は低下するが，認識率は高くなることがわかる．これは，正解物体以外の物体 ID が多く検索されても，物体の認識率にあまり影響を与えないことを意味する．実際どのようにスコアが加算されているか具体的に

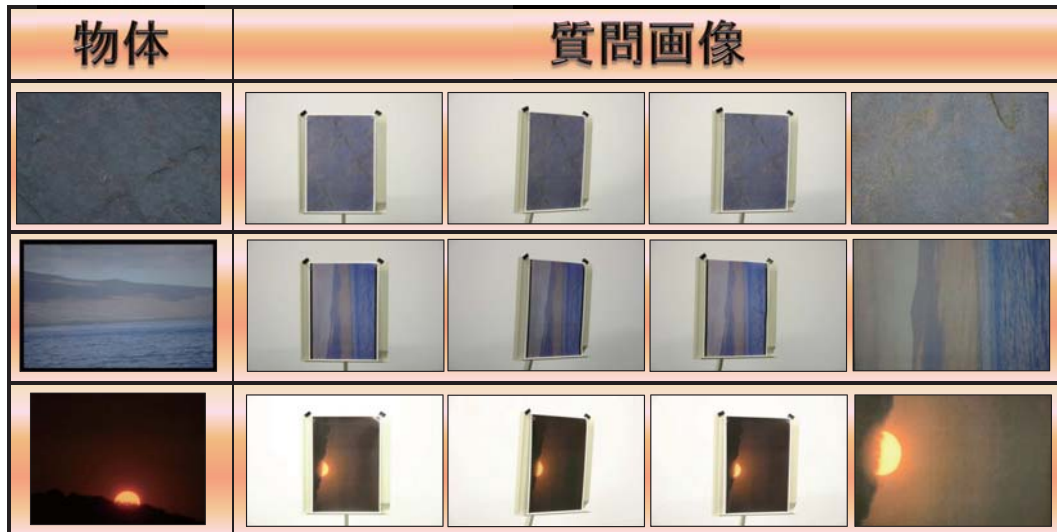


図 3.17: b を増加させても全く認識できなかった検索質問画像の例

表 3.5: 物体 ID の平均検索回数

	$b = 0$	$b = 7$
正解物体	11	55
正解物体以外	1	1

見ていく．まず，検索質問画像当たり正解物体以外の物体 ID が何個検索されるか調べたところ， $b = 0$ の時平均 43 個， $b = 7$ の時平均 1148 個検索されることが分かった．次に，検索された物体 ID のみに着目する．このとき，検索質問画像当たり各物体 ID が平均で何回検索されているか調べたところ，表 3.5 のような結果を得た．表 3.5 より，正解物体以外の物体 ID が多数検索されても，ある一つの物体 ID が集中して検索されず，正解物体の物体 ID は集中して検索されることが分かった．これより，提案手法のようなスコアを用いると高い認識率を得ることができると言える．

以上の結果より，提案手法を物体 ID 検索器と見立てた場合，適合率よりも再現率を重視した方が，高い物体の認識率を得ることができると言える．



図 3.18: 55 個の三次元物体の例

3.5 55 個の三次元物体を用いた実験

本節では，提案手法の有効性を確かめるために，55 個の三次元物体のデータセットに対して行った実験結果について述べる．

3.5.1 実験条件

55 個の三次元物体のデータセットについて説明する．本実験では，物体に対して，真正面，真正面から上へ 15° ， 30° の角度から，ウェブカメラ³で， 5° ごとに物体を回転させて撮影した画像を用いた．このうち，データベース作成用画像として，回転角度が 0° ， 10° ， \dots ， 350° の画像 36 枚 \times 3 方向の，一物体当たり 108 枚を用い，残りの画像を検索質問画像とした．図 3.18 に本実験で用いた 55 個の三次元物体⁴の例を，付録の図 B.1，B.2 に全物体を示す．データベース作成用画像から得られる特徴ベクトルの数は，約 120 万個である．また，検索質問画像から得られる平均特徴ベクトル数は，約 200 個である．三次元物体の実験も二次元物体の実験と同様に，実験に用いた計算機は，CPU が AMD Opteron8378 2.4GHz，メモリが 128GB RAM のものである．実験結果で示す物体の認識率は，式 (2.5)

³Logicool Qcum Pro 9000，解像度：640 \times 480．第 2.4 節で述べた実験では，ハイビジョンビデオカメラを用いて物体を撮影していたが，結論で述べたようなサービスを考慮すると，より手軽な機器を用いる方が良いと判断し，本実験ではウェブカメラを用いた．

⁴撮影機器変更によって，第 2.4 節の実験で用いた三次元物体を再度撮影し直したが，一部の物体は，消耗や紛失等の理由で，別の物体と置き換えた．

表 3.6: 実験に用いた V-Tree(Vocabulary Tree) のパラメータ (55 個の三次元物体)

分岐数 x	深さ y	分岐数 x	深さ y
10	4	7	7
10	5	7	6
11		8	
12		9	
13		10	

で定義される .

3.5.2 従来手法との性能比較

本実験では, 第 3.4.2 節と同様に V-Tree, ハッシュ(距離計算なし) およびハッシュ(距離計算あり) の三手法と性能を比較する. ハッシュに基づく比較手法および提案手法で用いるパラメータ d, b, e, c を再度確認しておく. ビットベクトル変換に用いる次元数 d , ビットベクトルで反転の操作を行う次元数 b , 値の変動幅 e , 重複するビットベクトルの上限数 c の四つである. 本実験では, $d = 22, 24, 26, 28, 30$, $b = 0, 1, \dots, 10$, $e = 200$, $c = 1, 2, \dots, 10$ とした. また, 提案手法ではこれらに加えて偽陽性の発生を容認する *bit ID* の桁数 t を, $t = 0, 1, \dots, 5$ とした. V-Tree で用いたパラメータは表 3.6 に示すものである. その他の設定として, 本実験でも V-Tree で用いる正規化には, L_2 -ノルムを用い, ハッシュ(距離計算あり・なし) で用いるハッシュ表のサイズを $H_{\text{size}} = 2^d$ とした. ここで, 以下の実験結果で示す処理時間は, 一枚の画像の認識にかかった平均の処理時間であり, 特徴ベクトルの抽出時間等は含まない.

提案手法で最も認識率の高かった結果と, 三手法の結果のうち, それぞれ提案手法と同程度の認識率になるものと, 最も認識率の高かったものを表 3.7 と表 3.8 にまとめる. ここで, 表 3.7 に各手法のパラメータ, 表 3.8 に各手法の結果についてそれぞれ示す. また, 1 万個の平面物体の実験と同様にこれらの表でも, 提案手法と同程度の認識率となる結果を “同認識率”, 最も認識率の高かった結果を “最大認識率” と表記する.

結果より, 従来手法と比較すると, 1 万個の平面物体の実験と同様に処理時間では劣るものの, 同程度の認識率を得るのにメモリ使用量を大幅に削減できることが分かった. 1

表 3.7: 各手法のパラメータ (55 個の三次元物体)

手法	パラメータ
提案手法	$b = 5, c = 7, d = 26, e = 200, t = 1$
V-Tree(同認識率)	$x = 9, y = 6$
V-Tree(最大認識率)	$x = 10, y = 6$
ハッシュ(距離計算なし)(同認識率)	$b = 3, c = 2, d = 24, e = 200$
ハッシュ(距離計算なし)(最大認識率)	$b = 3, c = 4, d = 24, e = 200$
ハッシュ(距離計算あり)(同認識率)	$b = 8, c = 1, d = 22, e = 200$
ハッシュ(距離計算あり)(最大認識率)	$b = 5, c = 5, d = 22, e = 200$

表 3.8: 各手法の性能評価 (55 個の三次元物体)

手法	認識率 [%]	処理時間 [ms]	メモリ使用量 [MB]
提案手法	99.85	3.18	20
V-Tree(同認識率)	99.87	1.64	391
V-Tree(最大認識率)	99.98	1.79	557
ハッシュ(距離計算なし)(同認識率)	99.85	0.33	167
ハッシュ(距離計算なし)(最大認識率)	99.88	0.31	168
ハッシュ(距離計算あり)(同認識率)	99.83	0.28	251
ハッシュ(距離計算あり)(最大認識率)	100	0.19	275

万個の二次元物体の実験との違いは、処理時間が減少したことである。これは、対象物体が減少したことにより、単純に Bloom Filter の検索回数も減少したことと、検索質問画像から得られる特徴ベクトルの数が、1万物体の実験の時より少ないことが原因として挙げられる。図 3.19 に、提案手法で最も認識精度の高いパラメータを用いた場合において正しく認識できなかった検索質問画像の例を示す。図 3.19 に示す検索質問画像の特徴としては、二次元物体の実験と同様に得られる特徴ベクトルの数が少ないということが挙げられる。このことから、さらに認識精度を上げるには、PCA-SIFT 特徴量以外の特徴量も認識処理に用いる必要があると考える。

以上の結果より、三次元物体を対象とした実験でも、二次元物体の実験と同様に、認識率や処理時間が許容できるのであれば、提案手法を用いるとメモリ使用量を抑えることが



図 3.19: 提案手法で認識できなかった検索質問画像

できると言える。

3.6 結言

本章では，Bloomier Filter を用いた特徴ベクトルの符号化による照合方法について述べ，特定物体認識に必要なメモリ使用量を削減する手法を提案した．また，誤り検出符号であるパリティビットの概念を基に，Bloomier Filter で発生する偽陽性が認識精度に与える影響を少なくする手法を提案した．

1万個の二次元物体を用いた実験では，木構造およびハッシュ表を用いた従来手法と比較すると，同程度の認識率を得るのに必要なメモリ使用量を，それぞれ約 $1/7$ ，約 $1/3$ に削減できることが分かった．また，55個の三次元物体を用いた実験では，従来手法と比較すると，同程度の認識率を得るのに必要なメモリ使用量を約 $1/20$ ，約 $1/8$ に削減できることが分かった．両実験において処理時間で比較すると，提案手法は従来手法と比べると大きく劣るものの，二次元物体の実験では， $16[\text{ms}]$ で 95% 以上の認識率を，三次元物体の実験では， $3[\text{ms}]$ で 99.8% の認識率を得ることができた．今後の課題として，並列処理などを導入することで，認識処理の効率化を図ることが挙げられる．

誤り検出 Bloomier Filter の有効性に関する実験では，全てのパラメータ設定において認識精度を向上できることができた．さらに認識精度を向上させるために，誤り訂正符号の概念を導入することも今後の課題の一つである．

上述した課題以外に今後解決すべき課題として，更なる大規模な物体データベースを用い，提案手法のスケーラビリティを調べることや，Bloomier Filter と認識率，メモリ使用量の関係について理論的に解析することなどが挙げられる．

第4章 データセットの違いが物体認識手法に与える影響の解析

4.1 緒言

本章では、データセットの違いが物体認識手法に与える影響を解析する。認識対象物体が増加すると、特に三次元物体では、それらの中に互いの見た目が酷似するものが含まれる可能性が高くなる。これらの個体差を区別するには、データベースに登録される局所特徴量の数は多い方がよい。しかし、局所特徴量の登録数が増加すると、処理時間やメモリ使用量に影響する。このため、これらにできるだけ影響を与えずに多くの局所特徴量の情報を扱う必要がある。

この問題を解決する方法として、第2章で局所特徴量の取捨選択によるものを、また、第3章で Bloomier Filter を用いたものを提案した。特に、第3章で提案した局所特徴量の一致検索に基づくもの^{[64]-[67]}は、局所特徴量間の距離を計算する必要がなく、局所特徴量の存在のみを記録すれば良いため、少ないメモリ使用量でリアルタイム認識が可能である。但し、第3.5節で述べた実験を含め、特に三次元物体を対象とした従来の研究では、検証に用いられている三次元物体データセットは小規模なものが多い。例えば、COIL-100^[68]等の一般的な三次元物体データセットには、100物体程度しか含まれておらず、物体数の違いが手法に与える影響を検証するには十分であるとは言い難い。このため、上述の影響を検証するには、より大規模な三次元物体データセットが必要である。そこで、本研究では独自に1002個の三次元物体データセットを作成し、物体数の違いが手法に与える影響について検証する^{[69]-[71]}。また、COIL-100に含まれている物体の構成に着目すると、COIL-100には物体構造の単純なものが多く含まれている。しかし、一般的に物体数が増加すると、データセットを構成する物体が構造の単純なものに偏るとは限らない。そこで、本研究では作成する1002個の三次元物体データセットの構成をCOIL-100と異なるものとし、データセットの構成の違いが手法に与える影響についても検証する^[69]。



図 4.1: 三次元物体の画像例．三次元物体は，撮影される角度が変化すると，物体の見え方が大きく変化する．

本章の構成は以下の通りである．まず，第 4.2 節にて，一般的に公開されているデータセットに言及した後，独自に作成するデータセットを物体の性質ごとに詳しく説明する．次に，COIL-100 と 1002 個の三次元物体データセットを用いた実験より，第 4.3 節にて，物体データセットの構成の違いが一致検索に基づく手法に与える影響について検証する．また，第 4.4 節にて，物体数の違いが手法に与える影響について検証する．最後に，第 4.5 節を結言とする．

4.2 物体データセットと物体に関する性質

緒論で述べたような携帯端末を用いた物体認識サービスを実現するには，撮影された物体が何であることを認識する特定物体認識技術が必須である．ここで，特定物体認識について再度まとめる．特定物体認識^{[16]–[18]}は，物体のインスタンス名をコンピュータに認識させることである．対象となる物体は，大きく平面物体と三次元物体に分けられる．本章で扱う物体認識タスクは，後者を対象としている．図 4.1 に示すように，三次元物体は平面物体と異なり，撮影位置が変化すると，物体の見えも大きく変化する．本研究の三次元物体認識タスクは，撮影位置によらず特定の三次元物体を高精度に認識することである．

このような認識タスクでは，その性質より，物体データベース作成には，三次元物体ごとに様々な方向から撮影された画像から，物体の見えの情報を登録する必要がある．また，手法の有効性を検証するには，検索質問画像も様々な方向から撮影されたものを用意する必要がある．しかし，一般的に公開されている物体データベースには，上記のような条件の揃ったものがほとんどない．以下では，一般的な物体データベースについて概説するとともに，本研究の物体認識タスクに対する問題点について説明する．物体認識の分野



図 4.2: Caltech-256 の画像例

で用いられている一般的なデータベースは、大きく一般物体認識手法の有効性を検証するものと、特定物体認識手法の有効性を検証するものに分類することができ、それぞれについて説明する。また、本研究で我々が独自に用意した物体データについて説明する。

4.2.1 一般物体認識のための物体データベース

一般物体認識のための物体データベースでは、ある物体クラスに対して、複数の物体インスタンス画像が用意されている。このような物体データベースとして、Caltech-101^[87]、Caltech-256^[88]、PASCAL^[89]等が有名である。上記データベースでは、図 4.2 に示すように、物体インスタンスに重複はなく、各物体インスタンスに対して一枚の画像しか用意されていない。このため、ある物体インスタンスに対して、撮影位置を変更した検索質問画像を用意することが難しい。上記の理由より、Caltech-256 等は、特定物体認識手法の



図 4.3: COIL-100 の画像例

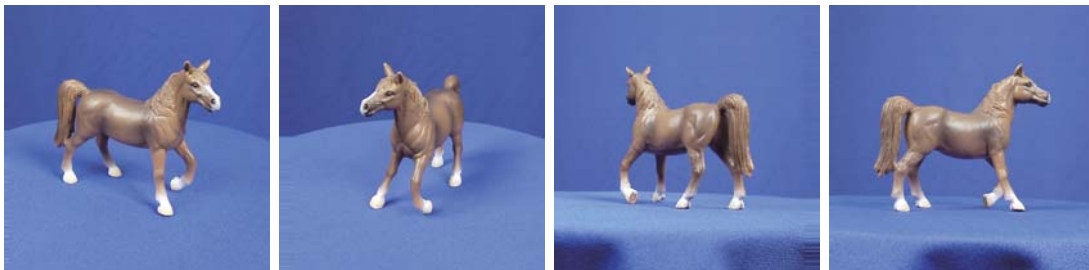


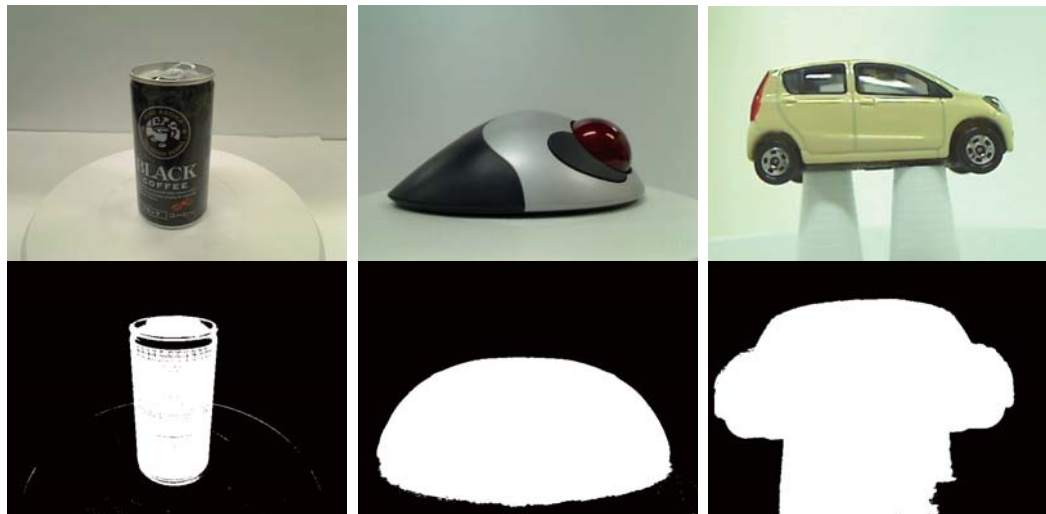
図 4.4: ETH-80 の画像例

検証に用いられることは少ない。

4.2.2 特定物体認識のための物体データベース

特定物体認識のための物体データベースでは、物体インスタンスごとに、様々な方向から撮影された複数枚の画像が用意されている。このような物体データベースとして、図 4.3 に示すような COIL-100 や図 4.4 に示すような ETH-80^[90] 等が有名である。特に、COIL-100 のデータセットは、100 個の物体インスタンスを 5° ごとに回転させて撮影された画像が用意されているため、本研究の物体認識タスクの検証に適していると言える。しかし、物体認識タスクの性質上、形状の異なる物体の収集やそれらの撮影に手間がかかるため、COIL-100 では 100 物体、ETH-80 では 80 物体と、登録されている物体数が少ないという問題点がある。このため、物体数の違いが物体認識手法与える影響を検証するのに十分であるとは言い難い。

また、COIL-100 には物体の構成に偏りがあるという問題点がある。一般的に物体数が増加すると、作成されるデータセットの構成が、必ずしも COIL-100 と同じものになるとは限らないこのため、物体データセットの構成の違いが特定物体認識手法に与える影響を



(a) 円柱型

(b) ドーム型

(c) 鏡餅型

図 4.5: 物体構造の複雑さの判断基準．上段に物体の例，下段にその物体をターンテーブルで回転させた際にできるシルエットを示す．これらの形に近似できる物体を物体構造の単純な物体とする．

検証するには，COIL-100のみでは不十分である．ここで，物体データセットを構成する物体の性質についてより詳しく言及しておく．本研究では，物体の見えに関する性質に基づいて物体を分類し，以後の議論を進める．物体の見えに関する性質には，大きく物体構造の複雑さとテクスチャの豊富さの二つがある．ここで本研究では，物体構造の複雑さを，物体をターンテーブルで回転させた際にできるシルエットが，円柱型，ドーム型，鏡餅型に近似できるかで判断する．これらの形に近似できる物体を物体構造の単純な物体とし，それ以外を複雑な物体とする．物体構造の複雑さは，撮影位置の変化による物体の見えに大きな影響を与える．特に，構造の複雑な物体は，撮影方向が少しでも変化すると，物体の部位による隠れや，影の位置が変化するという影響を受けやすい．このため，物体の見えに基づいた特定物体認識では，このような物体は認識するのが難しい部類に属する．一方，本研究では，物体のテクスチャの豊富さを，一物体から得られる特徴ベクトル数の平均と標準偏差を用いて判断する．特徴ベクトル抽出手法にはPCA-SIFTを用いる．このとき，平均から標準偏差を引いた値よりも得られる特徴ベクトル数が少ない物体をテクスチャの乏しい物体，それ以外を物体をテクスチャの豊富な物体とする．テクスチャ

表 4.1: 各物体属性に属する物体の数 (COIL-100)

	テクスチャが豊富	テクスチャが少ない
物体構造が単純	64	10
物体構造が複雑	22	4

表 4.2: 各物体属性に属する物体の数 (1002 個の三次元物体)

	テクスチャが豊富	テクスチャが少ない
物体構造が単純	209	37
物体構造が複雑	701	55

の豊富さは、撮影画像から抽出される特徴ベクトルの数に大きな影響を与える。特にテクスチャが少ない物体は、撮影画像より得られる特徴ベクトルの数が少ない傾向がある。このため、このような物体を認識するには、高い識別性を持った特徴ベクトルを用いる必要がある。上記性質をもとに、COIL-100 に含まれる物体を分類すると、表 4.1 のようになる。表 4.1 より、COIL-100 には物体構造の単純な物体が多く含まれている。

本研究では、一般的なデータベースにおける上記問題に対して、物体データセットの構成が COIL-100 とは異なる 1002 個の三次元物体データベースを独自に作成する。本論文では、COIL-100 に加え 1002 個の三次元物体データベースも用いて、データセットの違いを物体数及び物体データセットの構成の違いと定義し、これらが物体認識手法に与える影響について議論する。

4.2.3 1002 個の三次元物体

前節で述べたように、特定物体認識のための物体データベースでは、物体を様々な方向から撮影した複数枚の画像を必要とする。このような構成の一般的なデータベースは、いずれも登録されている物体数が少ないという問題がある。そこで本研究では、独自に 1002 個の三次元物体を用意する。付録の図 C.1 に全物体を示す。本論文では、これらを用いて以後の議論を進める。本節では、まず我々が用意した物体について、前節で述べた属性ごとに紹介する。各属性の物体数は、表 4.2 の通りである。

図 4.6(a) に、テクスチャが豊富で、構造が単純な物体の例を示す。このような物体は、

撮影位置の変化による物体の見えの違いが小さく、撮影画像から大量の特徴ベクトルを得ることができるという特徴がある。次に、テクスチャが豊富で、構造が複雑な物体の例を図 4.6(b) に示す。このような物体は、図 4.6(a) と比較すると、撮影位置の変化による物体の見えの違いが大きい。また、図 4.6(a) の物体と同様に、撮影画像からは大量の特徴ベクトルを得ることができる。さらに、テクスチャが少なく、構造が複雑な物体の例を図 4.6(c) に、テクスチャが少なく、構造が単純な物体の例を図 4.6(d) に示す。このような物体は、撮影画像から得られる特徴ベクトルの数が少ないため、いかにこれらの照合を正しく行うかが大きな課題となる。

次に、物体の類似性について説明する。データベースに登録される物体数を増加させると、見た目が類似するものが含まれる可能性が高くなる。具体例を図 4.7 に示す。このように、物体の大まかな見た目が類似し、細部で異なるものが世の中には多数存在する。この問題を物体認識の観点で議論すると、小規模な物体データベースでは、識別空間内において各物体における最近傍物体との距離が遠いため比較的識別も容易である。しかし、物体数が増加すると、見た目の類似する物体からは、似た特徴が得られるため、最近傍物体との距離が近くなる。このような見た目の類似する物体の個体差を正しく識別するためには、データベースに多くの特徴ベクトルを登録し、より精度の高い識別境界を求める必要がある^[52]。しかし、登録される特徴ベクトル数が増加すると、メモリ使用量が莫大になる。このため、大規模物体認識を実現するには、物体の個体差の情報をできるだけ失うことなくメモリ使用量を削減することが求められる。本研究では、大規模物体データベースにおける類似物体の認識精度についても言及するため、見た目の酷似する物体も用意する。

本研究では、上述のような観点の下で収集した 1002 個の三次元物体を用いて、物体データを作成する。本研究では、物体に対して、真正面、真正面から上へ 15° 、 30° の角度から、ウェブカメラ¹ で、 5° ごとに物体を回転させて撮影した画像を物体データとする。

¹Logicool Webcam Pro 9000，解像度：640×480



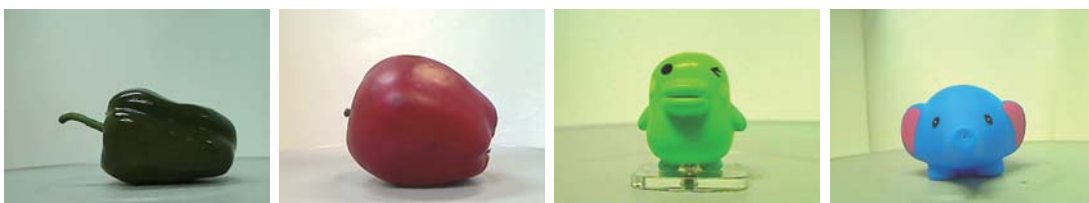
(a) テクスチャが豊富で、構造が単純な物体の例



(b) テクスチャが豊富で、構造が複雑な物体の例



(c) テクスチャが少なく、構造が複雑な物体の例



(d) テクスチャが少なく、構造が単純な物体の例

図 4.6: 1002 個の三次元物体の例



図 4.7: 見た目が類似する三次元物体の例

4.3 物体データセットの構成の違いに関する実験

本研究では，データセットの違いが特徴ベクトルの一致検索手法に与える影響を調べるために，一般的なデータセットである COIL-100 と，独自に用意した 1002 個の三次元物体データベースを用いて実験を行った．本節では，まず各データベースを用い，物体データセットの構成の違いが一致検索手法に与える影響について言及する．

4.3.1 実験条件

まず，COIL-100 [68] を対象とした実験の条件について説明する．本実験でも第 2.4.1 節と同様に， 5° ごとに撮影されている画像の内，角度が $0^\circ, 10^\circ, \dots, 350^\circ$ の画像 1 物体あたり 36 枚をデータベース作成用画像とし，残りを検索質問画像とした．一枚の検索質問画像から得られる平均特徴ベクトル数は約 50 個である．

次に，1002 個の三次元物体を対象とした実験の条件について述べる．本研究では，物体に対して，真正面，真正面から上へ $15^\circ, 30^\circ$ の角度から， 5° ごとに物体を回転させて撮影した画像を用いる．本実験では，このうちデータベース作成に，回転角度が $0^\circ, 10^\circ, \dots, 350^\circ$ の画像 36 枚 \times 3 方向の，一物体あたり 108 枚を用いた．また，残りの画像から仰角ごとに四枚ずつランダムに選択し検索質問画像とした．検索質問画像は，一物体あたり 12 枚であり，一枚の検索質問画像から得られる平均特徴ベクトル数は約 500 個である．

本実験では，特徴ベクトルの一致検索に基づく認識手法として，ハッシュ表を用いた手法 [62] (以後 Hash 法) と第 3 章で提案した Bloomier Filter を用いた手法 (以後 BF 法) の二手法を用いた．本実験に用いた計算機は，CPU AMD Opteron8378 2.4GHz，128GB RAM のものである．両手法で共通するパラメータについてまとめておく．両手法では，ビットベクトルの次元数 d ，ビットベクトルで反転操作を行う次元数 b ，及び値の許容変動幅 e が共通する．本実験では，それぞれ $d = 22, 24, 26, 28, 30$ ， $b = 0, 1, \dots, 10$ ， $e = 200$ を採用した．Hash 法においてハッシュ表のサイズは，文献 [62] と同様に $H_{\text{size}} = 2^d$ とした．これは，ビットベクトルの次元数を d ， $H_{\text{size}} = 2^d$ とする方が，次元数 $d'(d' > d)$ ， $H_{\text{size}} = 2^d$ とするよりも認識精度が高いためである [76]．また BF 法において，TableSize を左右するパラメータ a を， $a = 8$ とした．このとき，式 (3.2) より，各 Bloom Filter の偽陽性の発生確率は約 2% である．さらに BF 法において，偽陽性の発生を容認する *bit*

表 4.3: 両手法の最大認識率 (COIL-100)

手法	認識率 [%]
BF 法	98.77
Hash 法	98.83

表 4.4: 物体属性ごとの誤認識率 (COIL-100)

物体属性	BF 法 [%]	Hash 法 [%]
	(誤認識した検索質問画像の枚数 / 検索質問画像の合計枚数)	(誤認識した検索質問画像の枚数 / 検索質問画像の合計枚数)
物体構造が単純, テクスチャが豊富	0.95 (22 / 2304)	0.91 (21 / 2304)
物体構造が複雑, テクスチャが豊富	1.26 (10 / 792)	1.26 (10 / 792)
物体構造が複雑, テクスチャが少ない	4.17 (6 / 144)	1.39 (2 / 144)
物体構造が単純, テクスチャが少ない	1.67 (6 / 360)	2.50 (9 / 360)

ID の桁数 t を, $t = 0, 1, \dots, 5$ とした. 以下の実験結果で示す処理時間は, 一枚の検索質問画像の認識に必要な平均の処理時間で, 特徴ベクトルの抽出時間等は含まれていない. また, 実験結果で示す物体の認識率は, 式 (2.5) で定義される.

4.3.2 COIL-100 を用いた実験

まず, 一般的に公開されているデータベースである COIL-100 を用いて, 物体データセットの構成が一致検索手法に与える影響について述べる.

表 4.3 に両手法において最も認識率が高かった結果を示す. 結果より, 両手法は COIL-100 において高精度に物体を認識できることが分かった. ここで, 物体を第 4.2.2 節で述べた物体形状の複雑さとテクスチャの豊富さという二つの属性で分類し, 属性ごとに認識に失敗した検索質問画像の割合を調べた. 結果を表 4.4 に示す. 結果より, テクスチャの



図 4.8: 両手法で誤認識した検索質問画像の例 (COIL-100)

乏しい物体の方が誤認識の割合が高いことが分かった。また、両手法において正解物体以外のものに誤認識した検索質問画像に着目すると、同様の誤り傾向が二つあった。以下で詳しく述べる。

一つ目は、図 4.8 に示すように、認識手法の違いに関係なく、テクスチャの乏しい方向から撮影された検索質問画像は誤認識しやすいという点である。具体的に見て行こう。図 4.8 の左側二つに示すような物体は、撮影方向によらず物体の見えがあまり変化せず、テクスチャも乏しい。このような物体では、撮影方向によらず複数の検索質問画像で誤認識し、両手法の認識精度を最も低下させていることが分かった。特に、図 4.8 の左側二つの物体において誤認識した検索質問画像は、両手法とも全誤り検索質問画像のうち 1/4 以上を占めることが分かった。次に、図 4.8 の右から二つ目に示すような物体に着目すると、その物体においてテクスチャの乏しい方向から撮影されていることがわかる。最後に、図 4.8 の右から一つ目に示すような物体に着目すると、物体構造の複雑な物体において、最も構造が単純に見える方向と、その物体においてテクスチャの乏しい方向が一致する場合に誤認識しやすいことが分かった。以上のことから、特徴ベクトルの一致検索に基づく物体認識手法では、テクスチャの豊富さが、物体構造の複雑さよりも認識精度に大きく影響を与えていると言える。このため、これらの物体を正しく認識するには、テクスチャ情報に基づかない特徴抽出法を用いる必要があると考えられる。

二つ目の誤りの傾向は、物体のインスタンスに違いはあるものの、認識手法の違いに関係なく物体形状の類似する物体で誤認識しやすいという点である。図 4.9 において左側二つが BF 法で、右側二つが Hash 法で認識できなかった検索質問画像をそれぞれ示す。図 4.9 に示す物体は、図 4.8 に示す物体と比べると、いずれもテクスチャが豊富で、比較的認識しやすいものに分類される。しかし、図 4.9 に示す物体は、物体形状がお互いに類似するため、似た特徴ベクトルが得られやすくなり、正解物体以外のものに誤認識したと



図 4.9: COIL-100 において、見た目が類似する物体の検索質問画像．左側二枚と右側二枚の検索質問画像は、それぞれ BF 法と Hash 法で誤認識した検索質問画像である．

表 4.5: 両手法の最大認識率 (DB1002)

手法	認識率 [%]
BF 法	99.20
Hash 法	99.56

考えられる．このことから、テクスチャの豊富さに加え、物体形状の類似性も一致検索に基づく物体認識手法の認識精度に大きく影響を与えていると言える．形状の類似する物体の個体差を正しく識別するには、より識別能力の高い特徴ベクトルを用いることや、他の特徴と組み合わせる必要がある．

4.3.3 1002 個の三次元物体を用いた実験

次に、1002 個の三次元物体を用いて、物体データセットの構成が一致検索に基づく物体認識手法に与える影響について述べる．以後、1002 個の三次元物体データセットを DB1002 と表記する．

表 4.5 に両手法において最も認識率が高かった $d = 30$ の結果を示す．表 4.5 より、両手法は COIL-100 と同様に DB1002 において高精度に物体を認識できることが分かった．ここで、前節と同様に物体形状の複雑さとテクスチャの豊富さをもとに、各属性で認識に失敗した検索質問画像の割合を調べた．結果を表 4.6 に示す．表 4.6 より、COIL-100 とは異なり、構造が単純でテクスチャの豊富な物体が誤認識されやすいということが分かった．ここで、本節でも誤認識した検索質問画像に着目し議論を進める．

両手法において誤認識した検索質問画像例を図 4.10 に示す．図 4.10 より COIL-100 と

表 4.6: 物体属性ごとの誤認識率 (DB1002)

物体属性	BF 法 [%] (誤認識した検索質問画像の枚数 / 検索質問画像の合計枚数)	Hash 法 [%] (誤認識した検索質問画像の枚数 / 検索質問画像の合計枚数)
物体構造が単純, テクスチャが豊富	2.39 (60 / 2508)	1.36 (34 / 2508)
物体構造が複雑, テクスチャが豊富	0.30 (25 / 8412)	0.15 (13 / 8412)
物体構造が複雑, テクスチャが少ない	0.45 (3 / 660)	0.15 (1 / 660)
物体構造が単純, テクスチャが少ない	1.80 (8 / 444)	0.90 (4 / 444)



図 4.10: 両手法で誤認識した検索質問画像の例 (DB1002)

同様に、物体構造の複雑さによらずテクスチャの乏しい方向から撮影された検索質問画像は、誤認識されやすいことが分かった。このことから、データセットの違いによらず物体に対してテクスチャの乏しい方向から撮影すると、一致検索に基づく物体認識手法では、誤認識しやすいことが分かった。物体構造の複雑さの影響が少なかった原因として、データベース作成に用いた画像の枚数が多かったことが挙げられる。このため、今後の課題として、データベース作成に用いる画像の枚数を削減した場合に、物体構造の複雑さが物体認識手法にどのような影響を与えるか調べる必要がある。

また、DB1002 においても COIL-100 と同様に、形状の類似する物体で誤認識しやすいことが分かった。図 4.11 において左側二つが BF 法で、右側二つが Hash 法で認識できなかった検索質問画像をそれぞれ示す。両手法とも、COIL-100 と同様に認識に失敗しているインスタンスに違いはあるが、そのほとんどが、車というクラスに属しており、他の車



図 4.11: DB1002 において、見た目が類似する物体の検索質問画像．左側二枚と右側二枚の検索質問画像は、それぞれ BF 法と Hash 法で誤認識した検索質問画像である．

と誤認識しやすいことが分かった．COIL-100 と異なる点は、上記の誤りが両手法における誤りに占める割合が非常に高いことである．具体的に見ると、データセット内における車クラスの割合は、COIL-100 では 10%、DB1002 では 12% である．これに対し、全誤認識画像の内、車クラスに対するものが、両手法とも COIL-100 では約 1 割、DB1002 では約 6 割占めることが分かった．また本研究では、車というクラスは物体形状が単純でテクスチャの豊富な物体に分類されている．一般的にこれに分類される物体は、比較的認識しやすいことが知られている．しかし、形状の類似する物体の場合、データセット内に含まれる絶対数が増加すると、類似物体同士で誤認識し、手法の認識精度に大きく影響を与える可能性が高くなることが分かった．さらに、両手法における物体のスコアに着目すると、車に属する物体のスコアは、他の物体のスコアと比較すると、最大スコアと二番目のスコアの差が小さかった．このことから、形状の類似する物体は、物体数が増加すると一致検索に基づく物体認識手法に対して大きな影響を及ぼす可能性が高くなると言える．よって、物体形状の類似する物体を正しく認識するには、より識別能力の高い特徴ベクトルを用いる必要がある．

4.4 物体データセットの物体数の違いに関する実験

次に、DB1002 を用いて行った実験より、物体数の違いが特徴ベクトルの一致検索手法に与える影響について言及する．

表 4.7: 物体データベースの規模に対する認識率・処理時間・メモリ使用量の関係．両手法において最も認識率の高かった結果を示す．

手法 (データベースの規模)	認識率 [%]	処理時間 [ms]	メモリ使用量 [MB]
BF 法 (55 物体)	99.73	1.26	19
BF 法 (400 物体)	99.57	8.63	202
BF 法 (1002 物体)	99.37	18.16	457
Hash 法 (55 物体)	99.80	0.39	8236
Hash 法 (400 物体)	99.71	0.40	8724
Hash 法 (1002 物体)	99.74	0.38	9316

4.4.1 実験条件

本実験では，DB1002 を用いて，物体データベースの規模を 55 物体，400 物体，1002 物体と変化させた．また，本実験では検索質問画像として，55 個の三次元物体を真正面，真正面から上へ 15° ， 30° の角度から，回転角度が 5° ， 15° ， \dots ， 355° の画像 36 枚 \times 3 方向の，一物体あたり 108 枚を用いた．また，検索質問物体を 5 グループに分け，5-fold CV により求めた平均認識率・平均処理時間・平均メモリ使用量により，物体数が両手法に与える影響を調べた．

本実験では，ビットベクトルの次元数を前節の実験で最も認識精度の高かった $d = 30$ とした．他のパラメータは，第 4.3.1 節で述べたものを用いた．以下の実験結果で示す処理時間も，一枚の検索質問画像の認識に必要な平均の処理時間で，特徴ベクトルの抽出時間等は含まれていない．

4.4.2 実験結果

表 4.7 に，物体データベースの規模を 55 物体，400 物体，1002 物体と変化させた場合の結果の内，最も認識率の高かった結果をそれぞれ示す．結果より，両手法とも物体データベースの規模を拡大させると，認識精度が低下することが分かった．ここで，両手法において，誤認識した検索質問画像に着目する．まず，図 4.12 に物体データベースの規模によらず，両手法において誤認識した検索質問画像を示す．図 4.12 より，物体に対してテクスチャの乏しい方向から撮影されていることが分かった．このことから，物体データベースの構成だけでなく，物体データベースの規模にもよらず，物体に対してテクスチャ



図 4.12: 物体データベースの規模よらず両手法で誤認識した検索質問画像の例

の乏しい方向から物体を撮影すると、一致検索に基づく物体認識手法では、誤認識しやすいことが分かった。また、登録物体数が増加することで、誤認識できなくなった検索質問画像に着目すると、そのほとんどが、図 4.12 に示す物体を図 4.12 と異なる方向から撮影した検索質問画像だった。ここで、両手法における物体のスコアに着目すると、物体数が増加することで正解物体のスコアが減少し、他の物体のスコアと大小が入れ替わっていることが分かった。これは、データベースに登録される特徴ベクトルの数が増加するため、両手法の特徴ベクトルの検索アルゴリズムにおいて、正解物体以外から得られる特徴ベクトルの方が、正解物体から得られる特徴ベクトルよりも先に検索される可能性が高くなるためである。一方、物体数が増加することで、正しく認識できるようになった検索質問画像に着目すると、こちらも上記のように、検索される特徴ベクトルの変化により正しく認識できることが分かった。具体的に述べると、小規模なデータベースでは、両手法において検索される特徴ベクトルが、正解物体以外のある物体のものに偏り、この物体に誤認識していた。しかし、物体数が増加することで、検索される特徴ベクトルが変化し、この偏りが軽減されることが分かった。これにより、正解物体のスコアと正解物体以外のスコアの大小の入れ替わりが生じたと考えられる。また、上記のスコアの変動より、物体数が増加すると特徴ベクトル空間内において、各物体の特徴ベクトルの分布が互いに重なり合いやすくなっていると言える。つまり、データベースに登録される特徴ベクトルは、特徴ベクトル空間において均一ではなく偏りを持って増加することが分かった。このことから、さらに大規模な物体認識を実現するには、より識別能力の高い特徴ベクトルを用いて、特徴ベクトル空間内における物体間の距離を空ける必要がある。

次に、物体数増加が一致検索に基づく物体認識手法の処理時間に与える影響について述べる。表 4.7 より、BF 法は物体数が増加するにつれて徐々に処理時間も増加し、Hash 法はあまり処理時間に変化が無いことが分かった。これは、BF 法では特徴ベクトルの検索処理に用いられる Bloom Filter の数が物体数によって増加するが、Hash 法では、物体

表 4.8: BF 法において、最高認識率を達成するパラメータ b の平均値

データベースの規模	パラメータ b
55 物体	3.8
400 物体	6.0
1002 物体	6.2

数によらず一つのハッシュ表しか検索処理に用いないからである。また、BF 法の処理時間が増加するもう一つの原因として、Bloom Filter 数の増加に伴う偽陽性発生確率の増加が挙げられる。偽陽性発生確率が増加すると、特徴ベクトルの識別能力が低下する。ここで、表 4.8 に BF 法におけるビットベクトル反転操作を行う次元数 b の各データベースに対する最適値の平均を示す。表 4.8 より、BF 法では物体数が増加するに伴って b を大きくし、偽陽性による識別能力低下を、検索処理に用いる特徴ベクトルの数で補っていることが分かった。以上より、Hash 法は処理時間に関して物体数増加の影響をあまり受けませんが、BF 法は物体数に大きく影響を受けることが分かった。

最後に、物体数増加が両手法のメモリ使用量に与える影響について述べる。表 4.7 より物体数が増加するにつれて、両手法とも徐々にメモリ使用量が増加することが分かった。ここで、両手法のメモリ使用量のオーダーについて考える。 η_{ave} を一物体から得られる平均特徴ベクトル数とすると、BF 法のメモリ使用量のオーダーは $O(N\eta_{\text{ave}} \log N)$ 、Hash 法は $O(2^d + N\eta_{\text{ave}})$ となる。ここで、ハッシュ表のメモリ使用量を $H_{\text{size}} = 2^d, d = 30$ と固定すると、Hash 法のメモリ使用量のオーダーは $O(N\eta_{\text{ave}})$ となる。このことから、BF 法は Hash 法と比較して、メモリ使用量に対する物体数増加の影響を受けやすいと言える。また、実験結果および上記オーダーより、物体数をさらに増加させた場合のメモリ使用量の推定値を図 4.13 に示す。図 4.13 より、両手法のメモリ使用量に対する有効性が変わるのは、約 420 万物体と分かった。これは、一般的な博物館に収蔵されている展示物の数よりも多い^[91]。しかし図 4.13 より、両手法とも 10 万物体でおよそ 100GB 程度のメモリ容量を持つ計算機が必要であることが分かった。このため、実用的な物体認識サービスを提供するには、一致検索手法だけでなく、第 2 章で提案したような認識に有効な特徴ベクトルを取捨選択する手法と組み合わせることで、物体数増加の影響を軽減する必要がある。

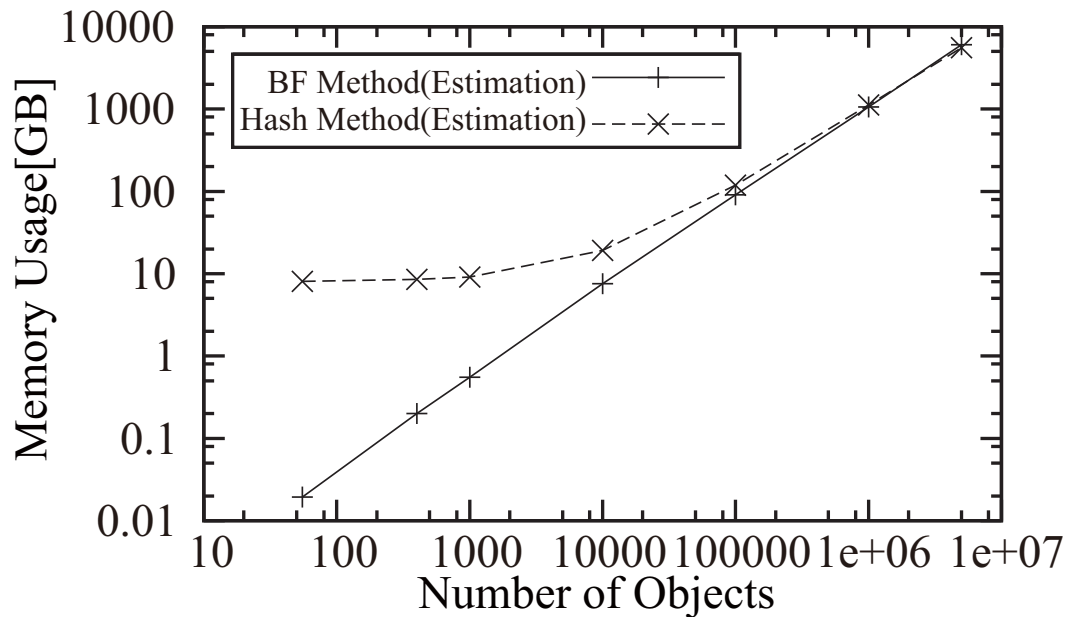


図 4.13: 物体数に対するメモリ使用量の推定値

4.5 結言

本章では、特徴ベクトルの一致検索に基づく物体認識手法として、Bloomier Filter に基づくものとハッシュ表に基づくものを用い、データセットの構成や登録物体数の違いが、手法にどのような影響を与えるかを検証した。構成の異なる二つのデータセットを用いた実験より、物体形状の複雑さよりもテクスチャの豊富さが、認識精度に大きく影響を与えていることが分かった。また、データセットに含まれる形状の類似する物体の数が、両手法の認識精度に大きな影響を及ぼすことが分かった。さらに、登録物体数を変化させた実験より、登録される特徴ベクトルが、特徴ベクトル空間において均一ではなく偏りを持って増加することが分かった。このため、上記問題を解決するためには、さらに識別能力の高い特徴ベクトルを用いる必要がある。

今後の課題として、さらなる物体データベースの大規模化やデータベース作成用画像の枚数削減、一致検索手法の理論的解析などが挙げられる。

第5章 結論

局所特徴量を用いた大規模特定物体認識を実現するためには、処理時間・メモリ使用量・認識率といった三つの大きな課題を解決する必要がある。これらの問題に対処するために、本論文では、特に後者二つの課題に焦点を当て、メモリ使用量の課題について、局所特徴量の取舍選択によるアプローチと Bloomier Filter を用いるアプローチの二つを提案した。また、認識率の問題に対して、物体数の違いや物体構成の違いが物体認識手法に与える影響について検証した。以下に、本論文の成果をまとめる。

第2章では、局所特徴量の取舍選択によるアプローチにより、物体モデルに必要なメモリ使用量を削減する手法を提案した。提案手法では、プラス関与性という物体認識に対して有効な性質を持った局所特徴量を、物体モデル構築用画像を用いた物体認識シミュレーションにより求め、より貢献度の高いものから Greedy に物体モデルへ登録する。また、あらゆる角度から撮影された場合にも安定して認識できるようにするために、あらゆる視点から得られる局所特徴量を満遍なく登録する。このような大胆な取舍選択により、高い認識精度を維持しつつ大幅なメモリ使用量の削減を実現した。55 物体を対象とした実験より、物体モデルのメモリ使用量を約 140 分の 1 に削減し、92.9% の認識率を得た。また、COIL-100 を対象とした実験結果より、メモリ使用量を約 18 分の 1 に削減した物体モデルで約 93.6% の認識率を得た。

第3章では、Bloomier Filter を用いた局所特徴量の符号化による照合方法について述べ、特定物体認識に必要なメモリ使用量を削減する手法を提案した。この手法では、局所特徴量の照合方法を従来の距離計算に基づくものから、局所特徴量を符号化し、この符号が一致するかどうかという一致検索に基づくものへ変更した。これにより、距離計算に必要な局所特徴量の情報がなくなり、大幅にメモリ使用量を削減した。また、Bloomier Filter で発生する偽陽性に対処するために、提案手法では誤り検出符号であるパリティビットの概念を導入し、認識精度に与える影響を抑えた。1 万個の二次元物体を用いた実験より、木構造及びハッシュ表を用いた従来のメモリ使用量削減法と比較して、同程度の認識

率を得るのに必要なメモリ使用量を、それぞれ約 $1/7$ 、約 $1/3$ に削減することができた。また、55 個の三次元物体を用いた実験では、従来手法と比較し、同程度の認識精度を得るのに必要なメモリ使用量を約 $1/20$ 、約 $1/8$ にできることが分かった。さらに、処理時間に関する実験結果より、提案手法は従来手法よりも大きく劣るものの、二次元物体の実験では、16[ms] で 95% 以上の認識率を、三次元物体の実験では、3[ms] で 99.8% 以上の認識率を達成できることが分かった。

第 4 章では、データセットを変更することにより、物体数の違いや物体構成の違いが、第 3 章で述べたような局所特徴量の一致検索に基づく物体認識手法にどのような影響を与えるのかを検証した。従来の研究において、手法の有効性の検証に用いられている物体データセットは、小規模なものが多かった。このため、物体数を大規模化すると、手法にどのような影響が出るのか検証が不十分であった。一般的に、データベースに登録される物体数が増加すると、これを構成する物体には、物体形状の複雑なものや単純なもの、テクスチャの豊富なものや乏しいもの、物体形状の類似するものの数に偏りが生じる。小規模なデータセットと異なる問題が、物体認識手法に発生すると考えられる。そこで本論文では、物体構成の異なる二つのデータセットを用いて、物体数を大規模化すると生じる影響について検証した。まず、本論文では物体の性質として、物体形状の複雑さとテクスチャの豊富さという二つの性質を定義し、これらの性質を用いて、手法に与える影響について議論した。独自に収集した 1002 個の三次元物体データセットと COIL-100 を用いた実験の結果、物体形状の複雑さよりもテクスチャの豊富さの方が、局所特徴量の一致検索に基づく物体認識手法の認識精度に大きく影響を与えることが分かった。また、物体数を変化させた実験より、物体数が増加しても、形状の大きく異なる物体は正しく認識できるものの、形状の類似する物体は、互いに誤認識しやすいことが分かった。このことから、登録物体数が増加すると、データベースに登録される局所特徴量が、特徴空間において均一ではなく偏りを持って増加することが分かった。このため、局所特徴量の分布が密な領域に存在する局所特徴量を高精度に識別する手法が必要であることが分かった。

今後の課題として、以下の事柄が挙げられる。まず、本論文で述べた提案手法について、理論的に解析することにより、与えられたデータセットに対して、認識率・処理時間・メモリ使用量の最適値を求めることである。また、本研究で扱ってきた物体データセットは、背景が単純なものばかりである。しかし、携帯端末を用いた実用的な物体認識サービスを提供するためには、背景が複雑でも、画像中に存在する物体を正しく認識する必要が

ある．このため，提案手法を背景の複雑な検索質問画像に対しても高精度な認識を行えるように提案手法を改良していくことも今後の課題である．上記以外の課題として，物体認識サービスを実現するために，更なる大規模な物体データセットの大規模化や，更なるメモリ使用量削減手法を提案することなどが挙げられる．

謝辞

本研究を進めるにあたり，大阪府立大学大学院 工学研究科 黄瀬 浩一 教授には，懇切丁寧な御指導，御鞭撻を賜り，心から感謝するとともに，厚く御礼申し上げます。また，現在よりも稚拙な文章を書いていた学部生の時代から今日まで，貴重なお時間を割いてまで根気強く原稿などの訂正にお付き合い頂いたことに，深い感謝の意を表します。現在でもまだまだ至らぬ所は多々ありますが，御指導頂いた内容を胸に刻んで，これからも日々精進していきたいと思いをします。

本論文をまとめるにあたり，大阪府立大学大学院 工学研究科 汐崎 陽 教授，吉岡 理文 教授には，有益な御指導，御鞭撻を賜るとともに，副査として本論文を査読して頂き，深く感謝致します。

本研究の遂行に際して，多数の御助言を賜りました，大阪府立大学大学院 工学研究科 岩村 雅一 准教授に心から感謝の意を表します。また，岩村准教授には，常日頃から研究に限らず私の進路等の相談に対して，親身な御意見，御助言を頂き，重ねて御礼申し上げます。

大阪府立大学 高等教育推進機構 小島 篤博 准教授には，本研究の遂行に際して学部生時代より多数の御意見，御助言を賜り，深く感謝致します。

大阪府立大学大学院 工学研究科 内海 ゆづ子 助教には，私の研究遂行や研究生生活に関する内容から些細な内容までと多岐にわたる相談に対して，親身な御意見，御助言を賜り，深く感謝致します。

大阪府立産業技術総合研究所 南 努 所長には，博士後期課程に進学する際に，多大なる御高配を賜り，ここに深い感謝の意を表すとともに，厚く御礼申し上げます。また，南所長には，研究に望む姿勢や，成果発表等の発表準備の大切さについて，御指導頂きました。重ねて感謝の意を述べさせていただきます。

本道 貴行 氏（現在，パナソニック株式会社），岩田 和将 氏（現在，MHI エアロスペースシステムズ株式会社），住岡 浩之 氏（現在，株式会社村田製作所），多田 匡志 氏（現

在，株式会社日立製作所），辻 智彦 氏（現在，アイテック阪急阪神株式会社），三木 博史 氏（現在，株式会社日立製作所），外山 託海 氏（現在，ドイツ人工知能研究センター），泉野 和弘 氏（現在，株式会社浅野歯車工作所），クリンキグト マルティン 氏，孫 維瀚 氏をはじめとする大阪府立大学大学院 工学研究科 知能メディア処理研究室の皆様には，日頃から研究に限らず多岐に渡り大変お世話になりました．心より感謝を表します．

和田 遼介 氏（現在，日本精工株式会社），夏目 博法 氏（現在，近畿容器株式会社）を始めとする友人一同には，日頃より多大なるご支援を頂き，また励ましの声を掛けて頂きました．ここに深く感謝致します．

最後に，研究生生活を健康と精神面で支え，時には温かく見守り続け，時には叱咤激励してくれた家族，親戚に最大限感謝致します．

参考文献

- [1] P. Viola and M. Jones: “Rapid Object Detection using a Boosted Cascade of Simple Features”, In Proceedings of the 14th IEEE Conference on Computer Vision and Pattern Recognition, **1**, pp. 511–518 (2001).
- [2] C. Castillo and D. Jacobs: “Face Recognition with Large Pose Variation”, In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 537–544 (2011).
- [3] G. Tzimiropoulos, S. Zafeiriou and M. Pantic: “Robust and Efficient Parametric Face Alignment”, In Proceedings of 13th International Conference on Computer Vision, pp. 1847–1854 (2011).
- [4] N. Surendran and S. Xie: “Automated facial expression recognition – an integrated approach with optical flow analysis and Support Vector Machines”, International Journal of Intelligent Systems Technologies and Applications, **7**, 3, pp. 316–346 (2009).
- [5] “<http://www.google.com/mobile/goggles/>”.
- [6] Y. Zheng, M. Zhao, Y. Song, H. Adam, U. Buddemeier, A. Bissacco, F. Brucher, T.-S. Chua and H. Neven: “Tour the World: building a web-scale landmark recognition engine”, In Proceedings of the IEEE Computer Vision and Pattern Recognition, pp. 1085–1092 (2009).
- [7] D. J. Crandall, L. Backstrom, D. Huttenlocher and J. Kleinberg: “Mapping the World’s Photos”, In Proceedings of the 18th international conference on World wide web, WWW '09, pp. 761–770 (2009).

-
- [8] Y. Li, D. J. Crandall and D. P. Huttenlocher: “Landmark Classification in Large-scale Image Collections”, In Proceedings of IEEE 12th International Conference on Computer Vision, pp. 1957–1964 (2009).
- [9] G. Takacs, V. Chandrasekhar, N. Gelfand, Y. Xiong, W.-C. Chen, T. Bismpiagiannis, R. Grzeszczuk, K. Pulli and B. Girod: “Outdoors Augmented Reality on Mobile Phone using Loxel-Based Visual Feature Organization”, In Proceeding of the 1st ACM international conference on Multimedia information retrieval, MIR '08, pp. 427–434 (2008).
- [10] A. Kumar, J.-P. Tardif, R. Anati and K. Daniilidis: “Experiments On Visual Loop Closing Using Vocabulary Trees”, 2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, pp. 1–8 (2008).
- [11] J. Hays and A. A. Efros: “IM2GPS: estimating geographic information from a single image”, In Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition, pp. 1–8 (2008).
- [12] D. Joshi and J. Luo: “Inferring generic activities and events from image content and bags of geo-tags”, In Proceedings of the 2008 international conference on Content-based image and video retrieval, CIVR '08, pp. 37–46 (2008).
- [13] 八重樫恵太, 丸山拓馬, 柳井啓司: “ジオタグ画像認識における位置情報の利用法の検討と分析”, 画像の認識・理解シンポジウム (MIRU2010) 論文集, OS6-2, pp. 1340–1347 (2010).
- [14] L. Yang, R. Jin, R. Sukthankar and F. Jurie: “Unifying Discriminative Visual Codebook Generation with Classifier Training for Object Category Recognition”, In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (2008).
- [15] 柳井啓司: “一般物体認識の現状と今後 (テーマ関連/オーガナイズドセッション)”, 情報処理学会研究報告. CVIM, [コンピュータビジョンとイメージメディア], 2006, 93, pp. 121–134 (2006).
-

-
- [16] C. Schmid and R. Mohr: “Local grayvalue invariants for image retrieval”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **19**, pp. 530–535 (1997).
- [17] J. Sivic and A. Zisserman: “Video Google: A Text Retrieval Approach to Object Matching in Videos”, In *Proceedings of the 9th International Conference on Computer Vision*, pp. 1470–1477 (2003).
- [18] J. Philbin, O. Chum, M. Isard, J. Sivic and A. Zisserman: “Object Retrieval with Large Vocabularies and Fast Spatial Matching”, In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8 (2007).
- [19] D. Lowe: “Distinctive Image Features from Scale-Invariant Keypoints”, *International Journal of Computer Vision*, Vol. 60, No. 2, pp. 91–110 (2004).
- [20] 藤吉弘巨: “Gradient ベースの特徴抽出 - SIFT と HOG -”, *情報処理学会 研究報告 CVIM 160*, pp. 211–224 (2007).
- [21] Y. Ke and R. Sukthankar: “PCA-SIFT: A more distinctive representation for local image descriptors”, In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Vol. 2, pp. 506–513 (2004).
- [22] K. Mikolajczyk and C. Schmid: “A performance evaluation of local descriptors”, *IEEE Transactions on Pattern Analysis & Machine Intelligence*, **27**, 10, pp. 1615–1630 (2005).
- [23] 安部満, 吉田悠一: “高速かつメモリ消費量の少ない局所特徴量 CARD: Compact And Real-time Descriptors”, *画像の認識・理解シンポジウム (MIRU2011) 論文集*, IS4-59, pp. 1682–1689 (2011).
- [24] M. Calonder, V. Lepetit, C. Strecha and P. Fua: “BRIEF: Binary Robust Independent Elementary Features”, *Computer Vision – ECCV 2010*, Vol. 6314 of *Lecture Notes in Computer Science*, pp. 778–792 (2010).
- [25] S. Belongie, J. Malik and J. Puzicha: “Shape Matching and Object Recognition Using Shape Contexts”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **24**, pp. 509–522 (2001).
-

-
- [26] S. Lazebnik, C. Schmid and J. Ponce: “A Sparse Texture Representation Using Local Affine Regions”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **27**, 8, pp. 1265–1278 (2005).
- [27] F. Schaffalitzky and A. Zisserman: “Multi-view Matching for Unordered Image Sets”, In *Proceedings of European Conference on Computer Vision*, pp. 414–431 (2002).
- [28] J. Matas, O. Chum, M. Urban and T. Pajdla: “Robust Wide Baseline Stereo from Maximally Stable Extremal Regions”, In *Proceedings of British Machine Vision Conference In Proceedings of British Machine Vision Conference*, pp. 384–393 (2002).
- [29] C. Harris and M. Stephens: “A combined corner and edge detector”, In *Proceedings of the 4th Alvey Vision Conference*, pp. 147–151 (1988).
- [30] K. Mikolajczyk and C. Schmid: “Scale and affine invariant interest point detectors”, *International Journal of Computer Vision*, **60**, 1, pp. 63–86 (2004).
- [31] K. Mikolajczyk and C. Schmid: “Indexing based on scale invariant interest points”, In *Proceedings of the 8th International Conference on Computer Vision*, pp. 525–531 (2001).
- [32] F. Rothganger, S. Lazebnik, C. Schmid and J. Ponce: “3D Object Modeling and Recognition from Photographs and Image Sequences”, Ponce et al., Eds., *Toward Category-Level Object Recognition*, LNCS4170, Springer, pp. 105–126 (2006).
- [33] F. Rothganger, S. Lazebnik, C. Schmid and J. Ponce: “3d object modeling and recognition using local affine-invariant image descriptors and multi-view spatial constraints”, *International Journal of Computer Vision*, Vol. 66, No 3 (2006).
- [34] 金崎朝子, 中山英樹, 原田達也, 國吉康夫: “部分空間法とカラー立体高次局所自己相関特徴を用いた高速三次元物体認識”, *OS4-3*, pp. 103–110 (2009).
- [35] P. Yan, S. M. Khan and M. Shah: “3D Model based Object Class Detection in An Arbitrary View”, In *Proceedings of IEEE 11th International Conference on Computer Vision*, pp. 1–6 (2007).
-

-
- [36] H. Jégou, M. Douze, C. Schmid and P. Pérez: “Aggregating local descriptors into a compact image representation”, In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 3304–3311 (2010).
- [37] Š. Obdržálek and J. Matas: “Sub-linear Indexing for Large Scale Object Recognition”, In Proceedings of British Machine Vision Conference (BMVC), pp. 1–10 (2005).
- [38] T. Kinnunen, J.-K. Kamarainen, L. Lensu and H. Kälviäinen: “Bag-of-Features Codebook Generation by Self-Organisation”, In Proceedings of the 7th International Workshop on Advances in Self-Organizing Maps, WSOM '09, pp. 124–132 (2009).
- [39] M. Stark and B. Schiele: “How Good are Local Features for Classes of Geometric Objects”, In Proceedings of IEEE 11th International Conference on Computer Vision, pp. 1–8 (2007).
- [40] S. A. Nene and S. K. Nayar: “A Simple Algorithm for Nearest Neighbor Search in High Dimensions”, IEEE Transactions on Pattern Analysis and Machine Intelligence, **19**, 9, pp. 989–1003 (1997).
- [41] T. M. Cover and P. E. Hart: “Nearest Neighbor Pattern Classification”, IEEE Transactions on Information Theory, **13**, 1, pp. 21–27 (1967).
- [42] D. Lowe: “Local Feature View Clustering for 3D Object Recognition”, Proc. of CVPR2001, Vol. 1, pp. 682–688 (2001).
- [43] 野口和人, 黄瀬浩一, 岩村雅一: “大規模特定物体認識における認識率, 処理時間, メモリ量のバランスに関する実験的検討”, 電子情報通信学会論文誌 D, pp. 1135–1143 (2009).
- [44] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman and A. Y. Wu: “An optimal algorithm for approximate nearest neighbor searching fixed dimensions”, J. ACM, **45**, 6, pp. 891–923 (1998).
-

-
- [45] M. Özuysal, M. Calonder, V. Lepetit and P. Fua: “Fast Keypoint Recognition Using Random Ferns”, *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 99 No. 1 (2009).
- [46] V. Lepetit and P. Fua: “Keypoint Recognition Using Randomized Trees”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **28**, 9 (2006).
- [47] A. Andoni and P. Indyk: “Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions”, *Commun. ACM*, **51**, 1, pp. 117–122 (2008).
- [48] P. Indyk and R. Motowani: “Approximate nearest neighbors : Towards removing the curse of dimensionality”, In *Proceedings of the 30th ACM Symposium on Theory of Computing(STOC’98)*, pp. 604–613 (1999).
- [49] M. Datar, N. Immorlica, P. Indyk and V. S. Mirrokni: “Locality-Sensitive Hashing Scheme Based on p-Stable”, In *Proceedings of the 20th Annual Symposium on Computational Geometry(SCG2004)* (2004).
- [50] 松下裕輔, 和田俊和 : “Principal Component Hashing -等確率バケット分割による近似最近傍探索法-”, *画像の認識・理解シンポジウム (MIRU2007) 論文集*, pp. 127–134 (2007).
- [51] Y. Weiss, A. Torralba and R. Fergus: “Spectral Hashing”, *Advances in Neural Information Processing Systems*, pp. 1753–1760 (2008).
- [52] D. Nistér and H. Stewénius: “Scalable Recognition with a Vocabulary Tree”, In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 775–781 (2006).
- [53] 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 the IEEE Conference on Computer Vision and Pattern Recognition* (2008).
- [54] H. Jégou, M. Douze and C. Schmid: “Packing Bag-of-features”, *Proc. of ICCV2009*, pp. 2357–2364 (2009).
-

-
- [55] K. Kise, K. Noguchi and M. Iwamura: “Memory Efficient Recognition of Specific Objects with Local Features”, In Proceedings of the 19th International Conference of Pattern Recognition (ICPR2008) WeAT3.1 (2008).
- [56] 本道貴行, 黄瀬浩一: “特定物体認識のためのデータベース容量削減法の検討～局所特徴量の量子化と取舍選択～”, 電子情報通信学会技術研究報告, PRMU2008-265, pp. 171–176 (2009).
- [57] F. Fraundorfer, H. Stewènius and D. Nistèr: “A Binning Scheme for Fast Hard Drive Based Image Search”, In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–6 (2007).
- [58] 姫井教考, 和田俊和: “B+木を用いた補助記憶装置上での近似最近傍探索”, 電子情報通信学会技術研究報告, PRMU2008-273, pp. 223–228 (2009).
- [59] 井上勝文, 三宅弘志, 黄瀬浩一: “局所記述子に基づく3次元物体認識のためのメモリ削減—局所記述子の取舍選択によるアプローチ—”, 画像の認識・理解シンポジウム (MIRU2008) 論文集, OS15-3, pp. 363–370 (2008).
- [60] K. Inoue, H. Miyake and K. Kise: “A Memory Reduction Method for 3D Object Recognition Based on Selection of Local Features”, In Proceedings of Third Korea-Japan Joint Workshop on Pattern Recognition (KJPR2008), pp. 7–8 (2008).
- [61] 井上勝文, 三宅弘志, 黄瀬浩一: “三次元物体認識に用いる局所特徴量の取舍選択によるメモリ使用量削減手法の実験的検討”, 電子情報通信学会論文誌 D, J92-D, 9, pp. 1686–1689 (2009).
- [62] 野口和人, 黄瀬浩一, 岩村雅一: “近似最近傍探索の多段階化による物体の高速認識”, 画像の認識・理解シンポジウム (MIRU2007) 論文集, OS-B2-02, pp. 111–118 (2007).
- [63] B. Chazelle, J. Kilian, R. Rubinfeld and A. Tal: “The Bloomier Filter: An Efficient Data Structure for Static Support Lookup Table”, In Proceedings of 15th Annual ACM-SIAM SODA, pp. 30–39 (2004).
- [64] 井上勝文, 黄瀬浩一: “特定物体認識における Bloomier Filter を用いたメモリ削減法とその実験的評価”, 電子情報通信学会論文誌 D, J93-D, 8, pp. 1407–1416 (2010).
-

-
- [65] K. Inoue and K. Kise: “Compressed Representation of Feature Vectors Using a Bloomier Filter and Its Application to Specific Object Recognition”, In Proceedings of IEEE 1st Workshop on Emergent Issues in Large Amount of Visual Data, pp. 2133–2140 (2009).
- [66] K. Inoue and K. Kise: “A Method of Memory Reduction for Specific Object Recognition with a Bloomier Filter”, Proceedings of the 2009 International Conference on Multimedia, Information Technology and its Applications (MITA2009), pp. 27–28 (2009).
- [67] K. Inoue and K. Kise: “Memory Reduction Method for Specific Object Recognition Based on Lossy Encoding of Feature Vectors with a Bloomier Filter”, In Proceedings of the 2nd China-Japan-Korea Joint Workshop on Pattern Recognition (CJKPR2010), pp. 166–171 (2010).
- [68] “<http://www1.cs.columbia.edu/cave/software/softlib/coil-100.php>”.
- [69] 井上勝文, 黄瀬浩一: “データセットの違いが物体認識に与える影響の解析—特徴ベクトルの一致検索を用いた認識手法の場合—”, 電気学会論文誌 (C), **131**, 11, pp. 1915–1924 (2011).
- [70] 井上勝文, 黄瀬浩一: “大規模特定物体認識における Bloomier Filter を用いたメモリ削減法の有効性の検証”, 平成 22 年電気関係学会関西連合大会 講演論文集, pp. 922–925 (2010).
- [71] 井上勝文, 黄瀬浩一: “登録物体数の違いが Bloomier Filter を用いた特定物体認識手法に与える影響”, 平成 23 年電気関係学会関西連合大会 講演論文集, pp. 618–619 (2011).
- [72] A. Wakulicz-Deja and M. Przybyla-Kasperek: “Application of the Method of Editing and Condensing in the Process of Global Decision-making”, *Fundamental Informatica*, **106**, 1, pp. 93–117 (2011).
- [73] S. Nordbotten: “Evaluating Efficiency of Statistical Data Editing: General Framework”, UN/ECE Conference on European Statisticians – Methodological Material, pp. 1–18 (2000).
-

-
- [74] 加藤丈和, 和田俊和: “近接性グラフに基づく効率的 condensing のアルゴリズムと評価”, 信学技報 PRMU2003-15, Vol. 103, No. 96, pp. 19–24 (2003).
- [75] 野口和人, 黄瀬浩一, 岩村雅一: “物体の認識率と局所記述子の照合精度の関係”, 画像の認識・理解シンポジウム (MIRU2008) 論文集, p. 504 (2008).
- [76] 井上勝文, 黄瀬浩一: “Bloomier Filter を用いた特徴ベクトルの圧縮表現とその特定物体認識への応用”, 画像の認識・理解シンポジウム (MIRU2009) 論文集, OS12-2, pp. 343–350 (2009).
- [77] 井上勝文, 黄瀬浩一: “特定物体認識における Bloomier Filter を用いたメモリ削減”, 電子情報通信学会技術研究報告, PRMU2009-56, pp. 101–106 (2009).
- [78] B. H. Bloom: “Space/Time Trade-offs in Hash Coding with Allowable Errors”, Commun. ACM, Vol. 13, No. 7, pp. 422–426 (1970).
- [79] 井上勝文, 黄瀬浩一: “Bloomier Filter を用いた特定物体認識手法の性能に関する実験的検討”, 電子情報通信学会技術研究報告, PRMU2009-109, pp. 115–120 (2009).
- [80] K. Kise and T. Kashiwagi: “1.5 Million Subspaces of a Local Feature Space for 3D Object Recognition”, In Proceedings of the 1st Asian Conference on Pattern Recognition (2011).
- [81] 柏木隆宏, 黄瀬浩一: “局所特徴量の多数の部分空間を用いた 3 次元物体認識と其の高速化・高精度化”, 電子情報通信学会技術研究報告, 111, 193, pp. 133–138 (2011).
- [82] R. Basri, T. Hassner and L. Zelnik-Manor: “Approximate Nearest Subspace Search”, IEEE Transactions on Pattern Analysis and Machine Intelligence, **33**, 2, pp. 266–278 (2011).
- [83] G. Csurka, C. R. Dance, L. Fan, J. Willamowski and C. Bray: “Visual categorization with bags of keypoints”, In Workshop on Statistical Learning in Computer Vision, ECCV, pp. 59–74 (2004).
- [84] P. Soucy and G. W. Mineau: “Beyond tfidf weighting for text categorization in the vector space model”, IJCAI’05: Proceedings of the 19th international joint conference on Artificial intelligence, pp. 1130–1135 (2005).
-

-
- [85] H. Jegou, M. Douze and C. Schmid: “Hamming embedding and weak geometric consistency for large scale image search”, In Proceedings of the 10th European Conference on Computer Vision, pp. 304–317 (2008).
- [86] “<http://www.flickr.com/>”.
- [87] “http://www.vision.caltech.edu/image_datasets/caltech101/”.
- [88] “http://www.vision.caltech.edu/image_datasets/caltech256/”.
- [89] “<http://pascallin.ecs.soton.ac.uk/challenges/voc/>”.
- [90] “<http://people.csail.mit.edu/jjl/libpmk/samples/eth.html>”.
- [91] “<http://www.tnm.jp/>”.
-

付録

付録 A 第 2.4 節の実験で用いた三次元物体

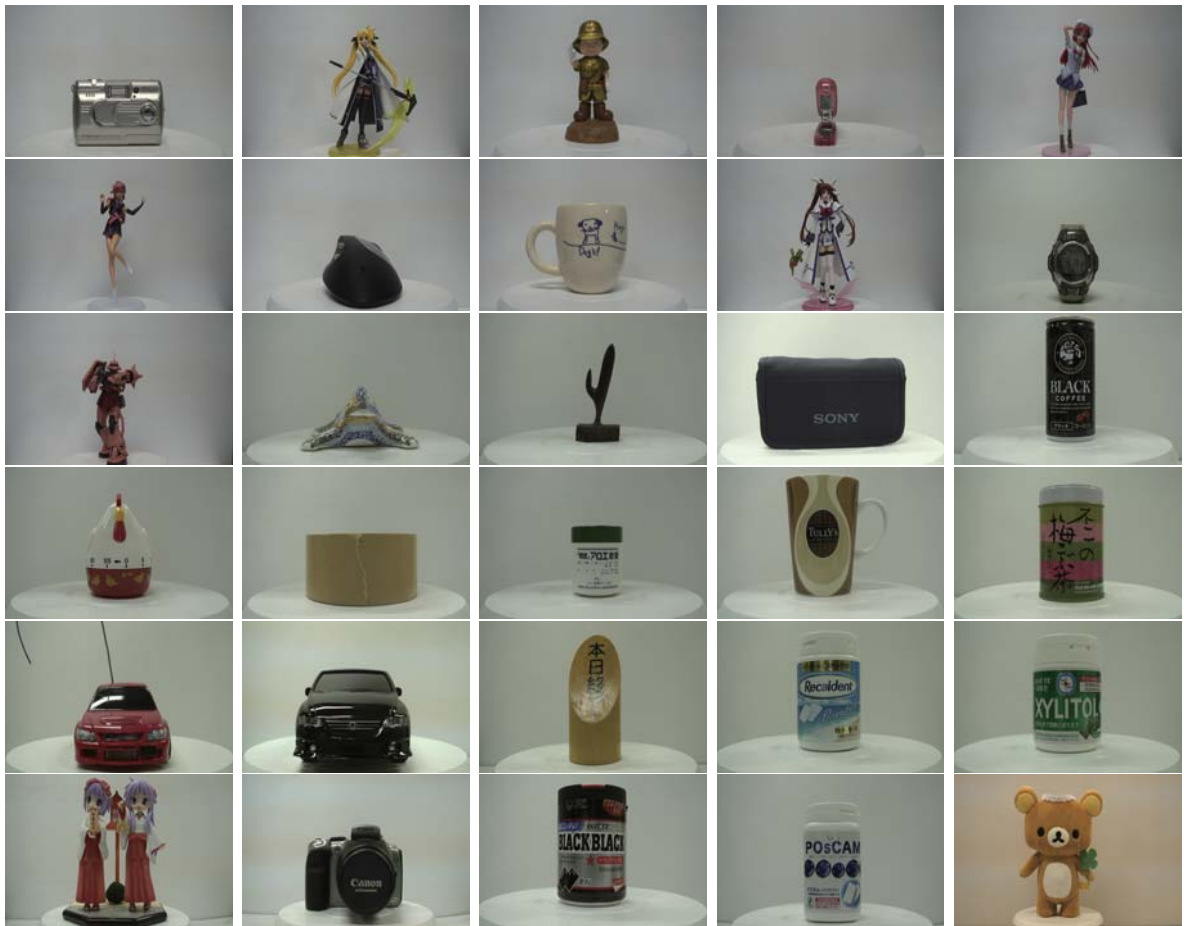


図 A.1: 第 2.4 節の実験で用いた三次元物体 (1 ~ 30)



図 A.2: 第 2.4 節の実験で用いた三次元物体 (31 ~ 55)

付録B 第 3.5 節の実験で用いた三次元物体

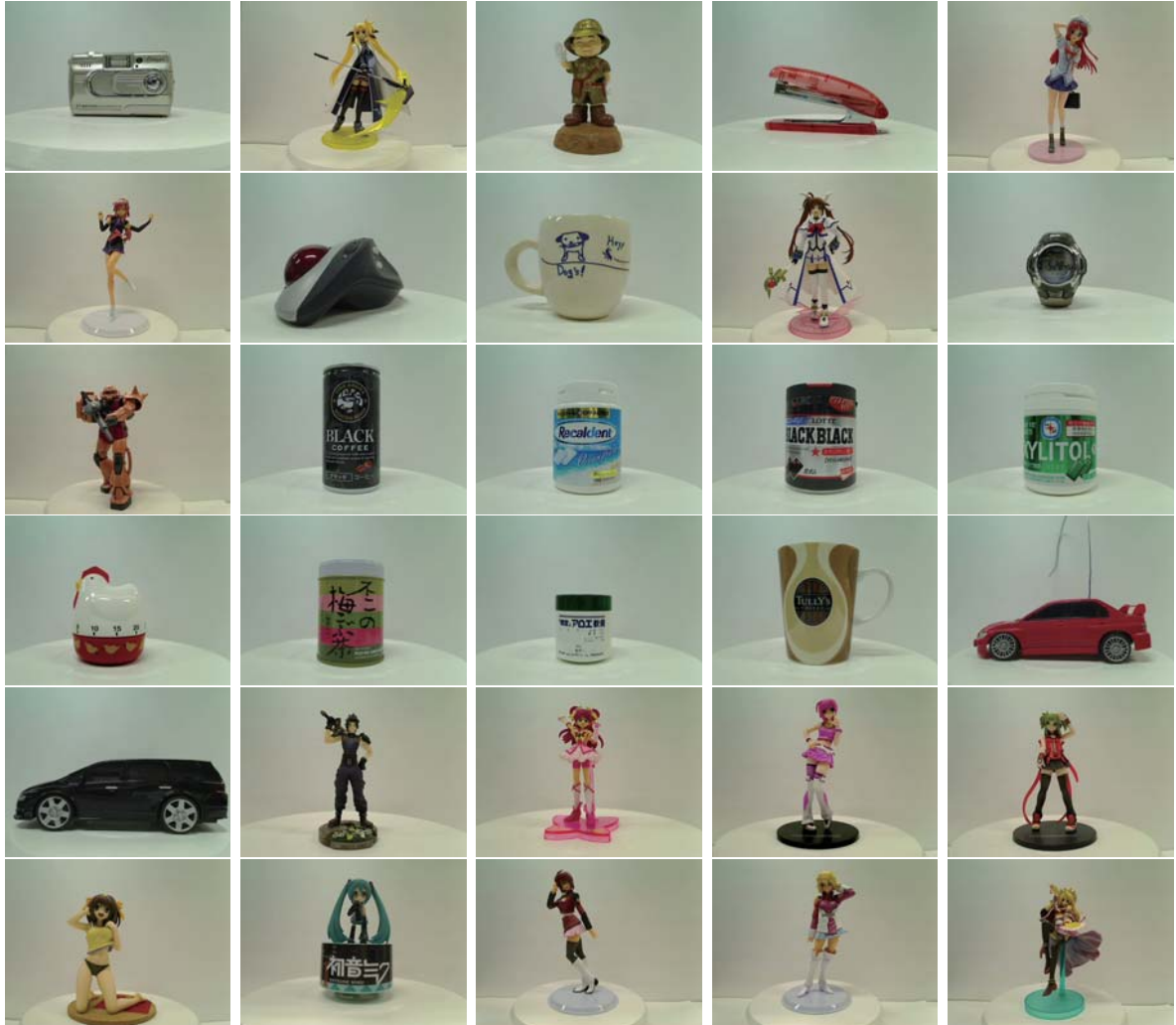


図 B.1: 第 3.5 節の実験で用いた三次元物体 (1 ~ 30)



図 B.2: 第 3.5 節の実験で用いた三次元物体 (31 ~ 55)

付録C 第 4.2.3 節で述べた 1002 個三次元物体データセット

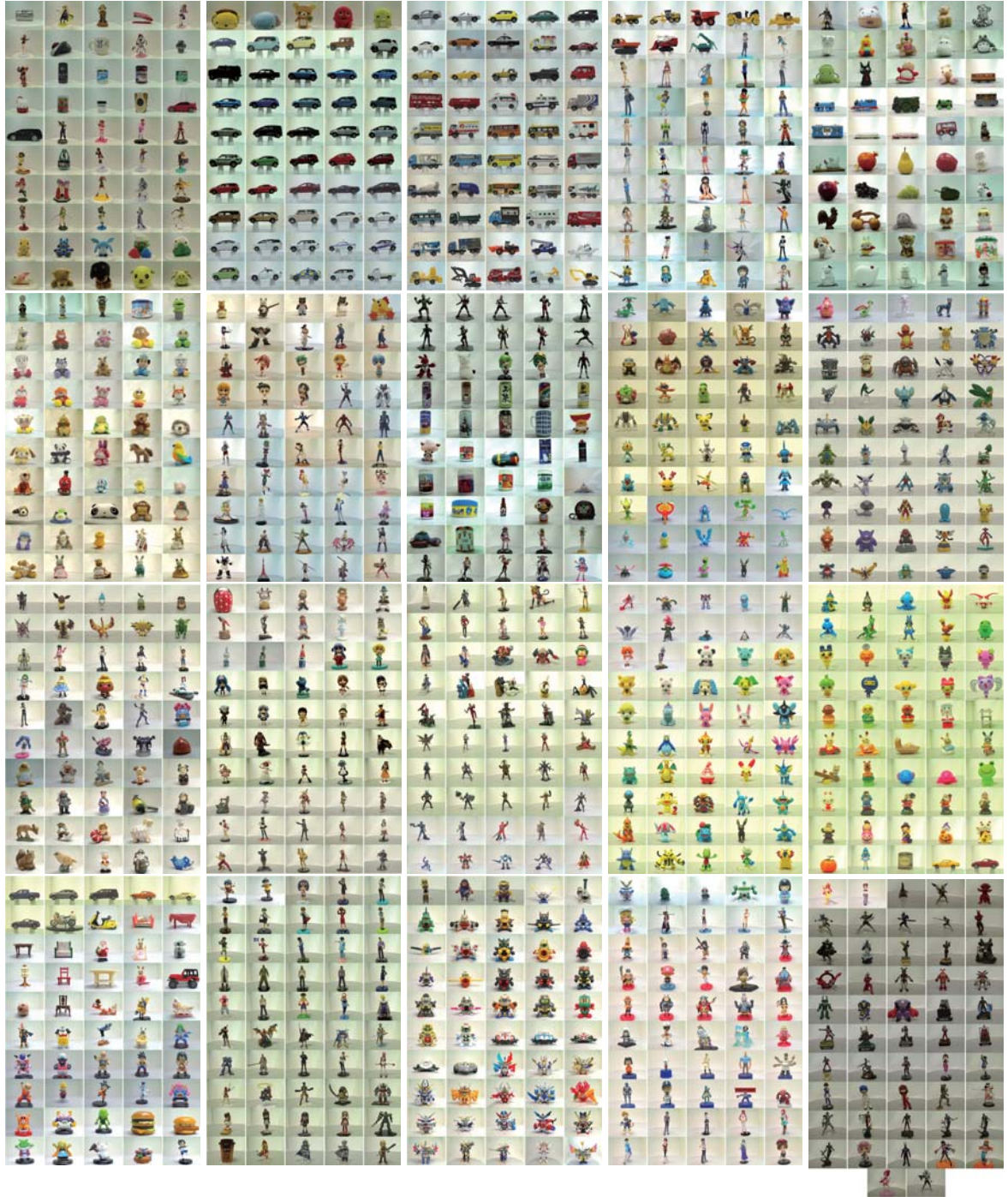


図 C.1: 第 4.2.3 節で述べた 1002 個三次元物体データセット