近さの多段階表現に基づく近似最近傍探索の一般的な分布への拡張

多田　匡志† 　武藤　大志† 　岩村　雅一† 　黄瀬　浩一†

† 大阪府立大学大学院工学研究科　〒 599-8531 堺市中区学園町 1-1
E-mail: †{tada,mutoh}@m.cs.osakafu-u.ac.jp, †{masa,kise}@cs.osakafu-u.ac.jp

あらまし　近似最近傍探索は，クエリと最も距離が近い点を探索する最近傍探索問題において，探索精度を犠牲にすることで計算時間，メモリ使用量を大幅に削減する手法である．我々は文献 [1] で 2 つの近似最近傍探索手法 MVH1 と MVH2 を提案した．これらの手法は距離尺度が L_1 ノルムであり，データの分布が一様分布の場合で有効性を確認した．本稿では，これらの手法の制約を排除し，L_p ノルムならびに一般的な分布に適用可能な手法を提案する．距離尺度が L_2 ノルムである時，改良した 1 つ目の提案手法は，計算時間を LSH の約 70%に削減した．改良した 2 つ目の提案手法は，LSH に対してメモリを約 50%削減した．

キーワード　近似最近傍探索, Locality Sensitive Hashing, Multi-Valued Hashing, Principal Component Hashing, 計算時間，メモリ使用量, 探索精度

Extensions of Approximate Nearest Neighbor Search Based on a Multi-Valued Expression of Closeness to General Distributions

Masashi TADA†, Tomoyuki MUTO†, Masakazu IWAMURA†, and Koichi KISE†

† Graduate School of Engineering, Osaka Prefecture University
1-1 Gakuencho, Naka, Sakai, 599-8531 Japan
E-mail: †{tada,mutoh}@m.cs.osakafu-u.ac.jp, †{masa,kise}@cs.osakafu-u.ac.jp

Abstract　Approximate nearest neighbor search is a technique which greatly reduces processing time and required amount of memory for nearest neighbor search. In [1], we proposed two approximate nearest neighbor methods: MVH1 and MVH2. We comfirmed the effectiveness of these methods for uniform distribution with L_1 norm. In this report, we get rid of the restrictions of these methods and propose efficient methods for L_p norm and general distributions. For L_2 norm, improved MVH1 achived about 75% of processing time of LSH. Improved MVH2 achived about 50% of required amount of memory of LSH.

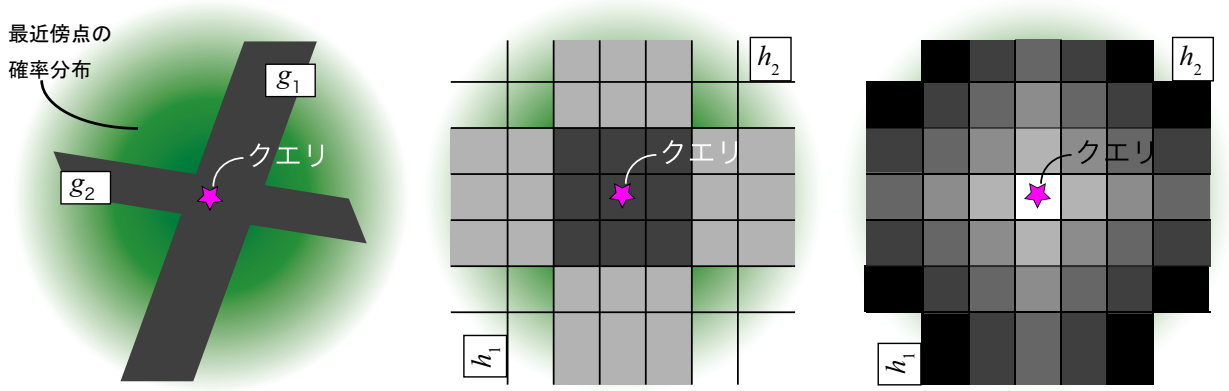
Key words　Approximate nearest neighbor search, Locality sensitive hashing, Multi-valued hashing, Principal component hashing, Processing time, Required mount of memory, Retrieval accuracy

1. 前　書　き

近年の計算機性能の向上と画像照合技術の発展によって，大規模データベースを用いた物体認識研究が盛んに行われている．例えば，Hervé ら [2] や野口ら [3] は，約 100 万枚の画像の中からクエリ画像と一致する画像を高速に検索する手法を提案している．これらを実現するためには，いずれの場合も画像から SIFT [4] や PCA-SIFT [5] などの特徴量ベクトルを抽出し，データベースに登録されている特徴量ベクトルと類似のものを探索する必要がある．この探索において，検索精度が重視されるのか，それとも高速性や省メモリが重視されるのかはアプリケーションに依るが，検索精度が同一であれば，計算時間とメモリ使用量が小さいほうが良いのは言うに及ばない．

本稿では，上記のような探索問題のうち，最近傍探索と呼ばれるものを扱う．改めて問題を定義しておくと，最近傍探索とは，ベクトルで表現されるデータ（点）の中から，クエリと最も類似している（距離が小さい）データ（以後，最近傍点と呼ぶ）を探し出す問題である．大規模データに対する最近傍探索問題において，最近傍点を正確かつ高速に求めるためには膨大な計算時間やメモリ使用量を必要とするため，最近傍探索の誤りを許容することで，計算時間とメモリ使用量を大幅に削減できる近似最近傍探索が近年注目されている．

本研究では，近似最近傍探索手法のうち，特にハッシュを使用する方法に注目し，探索精度を維持しつつ，計算時間とメモリ使用量を削減する方法を検討する．探索精度としては，探索によって求めた近似最近傍点が真の最近傍点と一致する確率を



(a) LSH　(b) PCH　(c) 提案手法

図 1: LSH，PCH，MVH（提案手法）の距離計算対象の選定の様子．各図の背景は最近傍点の確率分布であり，クエリに近い領域は最近傍点が存在する確率が比較的高いことを表している．

用いる．

ハッシュを用いる近似最近傍探索の代表的な手法として，Locality Sensitive Hashing(LSH) [6]〜[8] がある LSH は，複数のハッシュ関数を利用して，クエリの近傍にあると考えられる特徴点を選出し，それらの特徴点に対してのみ距離計算を行う．LSH の計算時間は，距離計算に要する時間が支配的であるため，距離計算の候補を絞れば絞る程計算時間を削減することができる．しかし，絞り過ぎると真の最近傍点が距離計算の対象に含まれないため，正しく探索できない可能性が高くなる．つまり，最近傍点である可能性の高い点をうまく絞り込むことが求められる．

距離計算の対象になる点を効率良く選定する手法として，Principal Component Hashing(PCH) [9] や Adaptive PCH(A-PCH) [10] が提案されている．LSH では，複数のハッシュを調べて，1 回でもクエリと同じビンに入った点を全て距離計算の候補としたが，PCH と A-PCH では，クエリと同じビンやその近傍のビンに入った回数を数え，回数が多い点のみを距離計算の候補とする．これにより，LSH より効率の良い探索が実現できる．

我々は文献 [1] において，距離計算の対象をさらに効率良く選定するために，クエリとの近さに応じた重みを導入する 2 種類の手法 (MVH1 と MVH2) を提案した．1 つ目は，各ハッシュにおいて，クエリに近い点に大きい重みを，遠い点に小さい重みを与え，重みの総和が大きい点のみを距離計算の候補とする手法である．この重みの総和はクエリの近傍では L_1 距離を概ね反映しているため，距離計算の候補を PCH よりも正確に求めることができる．しかも，重みを導入しても PCH より計算量が増加しないという優れた性質を持つ．2 つ目は，1 つ目の手法の距離計算を省く手法である．すなわち，重みの総和が大きい点を最近傍点と決定する手法である．このようにすれば，距離計算のために保持しておく必要があった特徴ベクトルを一切使用しなくても良いので，メモリ使用量の大幅な軽減が実現できる．

前述のように文献 [1] では，距離尺度が L_1 ノルムで，一様分布なデータに対応した手法を提案した．本稿では，この手法を改良し，距離尺度が L_p ノルムで，一般的な分布に対応した手法を提案する．新たに提案する MVH の有効性を確認する実験も行う．

2. ハッシュを用いる近似最近傍探索手法

本研究に関連するハッシュを用いた既存の近似最近傍探索を概説する．

2.1 Locality Sensitive Hashing(LSH)

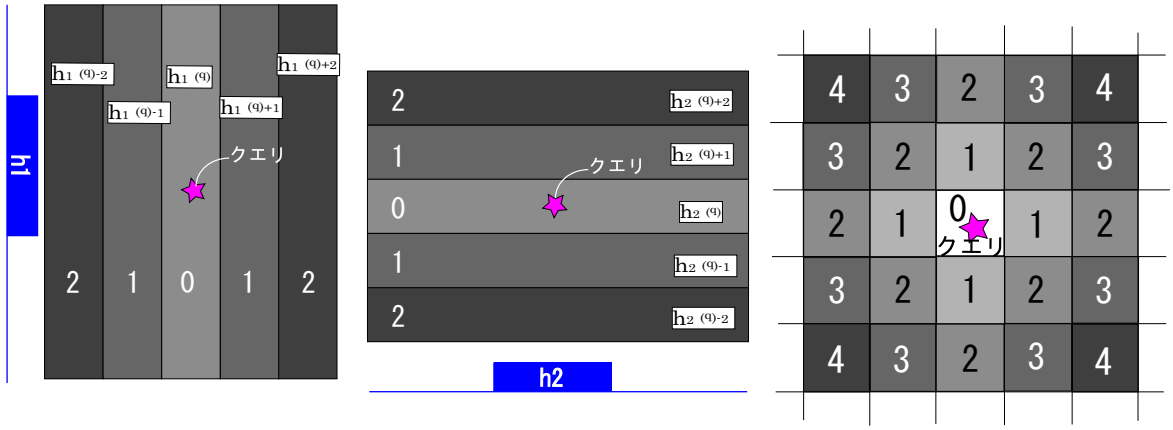
Locality Sensitive Hashing(LSH) [6]〜[8] はハッシュを用いた近似最近傍探索の代表的手法である．ここでは，LSH の中でもベクトル空間で用いることが出来る文献 [7] の LSH について，本研究と関連のある部分のみを述べる．LSH は，複数のハッシュを用いて距離を計算する対象になる点を求め，それらの点とクエリとの距離計算を行うことにより近似最近傍点を求める．

LSH が近似最近傍点を求めることができるのは局所性に鋭敏な (Locality Sensitive) ハッシュ関数を用いているためである．局所性に鋭敏なハッシュ関数とは，距離が近い点同士は同じハッシュ値を取る確率が高く，距離が遠い点同士は同じハッシュ値を取る確率が低いハッシュ関数である．文献 [7] の LSH では次式を用いて局所性に鋭敏なハッシュ関数を実現している．

$$h_{ji}(\boldsymbol{p}) = \left\lfloor \frac{\boldsymbol{a}_{ji} \cdot \boldsymbol{p} + b_{ji}}{w} \right\rfloor \tag{1}$$

ただし，$\boldsymbol{a}_{ji}$ は各次元の要素の値を平均 0，分散 1 のガウス分布から独立にとったベクトル，$\boldsymbol{p}$ はデータ点，b_{ji} は区間 $[0, w]$ からランダムに選ばれた実数であり，w はハッシュ幅である．

LSH では，ハッシュ関数 h_{ji} を k 個組み合わせてハッシュ関数群 g_j を作る．図 1(a) は LSH を用いて濃着色の距離計算の対象領域にある点を求める様子である．図 1(a) の g_1 は $k = 2$ のとき，h_{j1} と h_{j2} の 2 つのハッシュ関数で求めた距離計算の対象領域の共通部分である．このようなハッシュ関数群 g_j を L 個作り，L 個の領域を組み合わせた領域を距離計算の対象領域とする．図 1(a) は $L = 2$ の場合であり，図 1(a) の濃着色部



(a) $h_1(q)$ の距離情報　(b) $h_2(q)$ の距離情報　(c) $h_1(q)$ と $h_2(q)$ の距離情報の和

図 2: 提案手法の距離情報

分は，g_1 と g_2 の 2 つのハッシュ関数群を用いて得られる距離計算の対象領域を図示したものである．LSH は，上記の手順で距離計算の対象を削減して処理を高速化する．

2.2 Principal Component Hashing

Principal Component Hashing(PCH) [9] は，LSH と同様に距離計算を必要とするが，データの分布を正規分布と仮定し，距離計算の候補を効率的に求めることで性能の向上を図っている．

PCH は，まずデータの主成分分析を行い，主軸 φ_i を求める．各主軸にデータ点を射影し，正規分布の累積分布関数 $P_g(p)$ を用い，各バケットにデータ点が格納される確率を一定値にしている．PCH で用いられるハッシュ関数は

$$h_i(\boldsymbol{p}) = \left\lfloor \frac{P_g(\boldsymbol{\varphi}_i \cdot \boldsymbol{p})}{w} \right\rfloor \tag{2}$$

である．$\boldsymbol{\varphi}_i$ は主成分分析で求めた主軸（第 i 固有ベクトル），$\boldsymbol{p}$ はデータ点，w はハッシュ幅である．PCH では，各ハッシュ関数 $h_i(\boldsymbol{p})$ について，各点がクエリと同じビンに入った回数を数え上げ，これを重複度と呼ぶ．そして，重複度が高い点のみを距離計算の対象とする．距離計算では，距離計算の打ち切り効率を上げるために，寄与度の高い主軸 $\boldsymbol{\varphi}_i$ に射影を行ってから距離計算を行う．また，LSH ではクエリ点とハッシュ値が同じビンだけを探索していたのを，距離計算の候補に最近傍点が含まれる確率を高めるため，PCH ではその両脇 δ 個までの隣接するビンを参照する．

図 1(b) は，$\delta = 1$ の時，ハッシュ関数 h_1，h_2 を用いて得られる距離計算の対象領域を図示したものである．h_1 と h_2 の両方に参照される領域，つまり重複度が高い領域は濃い着色部分で示した．その領域にある点に対してのみ距離計算を行い，近似最近傍点を得る．

3. 提 案 手 法

本節ではハッシュを用いた近似最近傍探索を改良した提案手法（Multi-Valued Hashing; MVH）を 2 種類示す．これらは文献 [1] の手法をさらに発展させたものである．

3.1 ハッシュを用いた既存の近似最近傍探索の問題点

ハッシュを用いた近似最近傍探索の処理は大きく分けると 2 段階ある．1 段階目の処理において，ハッシュを用いて距離計算の対象となる点を選ぶ．2 段階目の処理では，1 段階目の処理によって得られた点とクエリの距離を計算して，近似最近傍点を得る．

1 つ目の提案手法の MVH1 では，1 段目の距離計算対象の選定能力を改良する．MVH1 は，LSH や PCH のように近さの情報を「近い」と「近くない」の 2 値情報で表現するのではなく，多値情報で表現することで，図 1(c) のようにクエリの近傍では L_p 距離の概算値を反映した情報を得ることができる．

2 つ目の提案手法の MVH2 では，MVH1 から距離計算の処理を省いた手法を提案する．このようにするのは，MVH1 で L_p 距離の概算値を反映した情報を得ることができるならば，2 段目の処理は不要と考えられるからである．

3.2 提案手法 1：MVH1

手法の詳細を以下で述べる．まず，距離尺度 L_1 ノルムに対応した MVH を説明する．提案手法では，

$$h_i(\boldsymbol{p}) = \left\lfloor \frac{\boldsymbol{\Psi}_i \cdot \boldsymbol{p}}{w} \right\rfloor \tag{3}$$

で示されるハッシュ関数を用いる．ただし，$\boldsymbol{\Psi}_i$ は正規直交基底，$\boldsymbol{p}$ はデータ点，w はハッシュ幅である．図 2(a) は h_1 の処理に関するものである．クエリと同じビンに入った点には 0 という距離の概算値を付加する．そして，その両隣のビン（$h_1(\boldsymbol{x}) = h_1(\boldsymbol{q}) \pm 1$ を満たすビン）に入った点には 1 という距離の概算値を付加する．つまり，s 個隣のビンについて考えると，クエリの s 個隣のビン（$h_1(\boldsymbol{x}) = h_1(\boldsymbol{q}) \pm s$ を満たすビン）には s という距離の概算値を付加する．これを，距離の概算値が t となる，t 個隣のビン ($h_1(\boldsymbol{x}) = h_1(q) \pm t$ を満たすビン) まで行い，$t+1$ 個以降の隣のビンには距離の概算値 t を与える．図 2(b) は h_2 の処理に関するものであり，こちらも h_1 と同様に処理を行う．このような処理を全ての h_i に対して行う．

次に h_i で付加された距離の概算値を全て加算する．図 2(c) は，h_1, h_2 で付加された距離の概算値を全て加算した結果を示したものである．例えば，h_1 で u，h_2 で u の距離の概算値を

付加された点はそれぞれの値を足して，$2u$ という距離の概算値を保持する．これを全ての点に対して行う．その結果を基にして，距離の概算値のヒストグラムを作る．このヒストグラムはクエリの近傍では概ね L_1 距離を反映する．故に，このヒストグラムの中で，任意の閾値 v よりも小さな点のみを距離計算の対象とすることで，距離計算のコストの削減が期待できる．このような処理により，PCH の重複度よりも距離を反映した情報を利用することができ，1 段階目の処理では同じ計算量で高い精度が得られる．

上記の説明では距離尺度に L_1 ノルムを用いた．これは文献 [1] の処理と概ね同じである．本稿で新たに追加する L_p ノルムに拡張させるための処理を以下に述べる．距離尺度 L_p ノルムにおける 2 点 $(x_1, x_2 \cdots)$，$(y_1, y_2, \cdots)$ の距離は $\sqrt[p]{\sum_i (x_i - y_i)^p}$ となる．故に，s 個隣のビンには s^p の距離の概算値を付加する．これにより，最終的に得られる距離の概算値のヒストグラムはクエリの近傍では概ね L_p 距離を反映する．

3.3 提案手法２：MVH2

上述の多値化処理を施した提案手法では近さの多値情報を付与して距離計算の対象を効率的に絞った．しかし，距離の概算値が距離を反映しているのであれば，そもそも距離計算を行う必要がない．故に，距離の概算値のヒストグラムの中で最も値が小さい点を近似最近傍点とすれば十分のはずである．そこで，２つ目の提案手法として，距離計算をせずに近似最近傍点を得る手法を提案する．このような処理では，精度が低下することが予想されるが，距離計算が不要になると距離計算を行うための元のデータを保持する必要がなくなる利点も存在する．そのため，精度をある程度犠牲にする代わりに，計算時間とメモリ使用量の大幅な削減が期待できる．

4. 実験および考察

提案手法の有効性を確認するために実験を行った．用いた計算機は，CPU が Opteron8439SE(2.8GHz)，メモリは 128GB である．距離尺度は L_2 ノルムを用いた．

4.1 実　験　1

一様分布に従うデータと正規分布に従うデータに対してあるクエリを与えた時，LSH と PCH と提案手法の，距離計算対象の選定結果と距離の関係を確認する実験を行った．なお，PCH は正規分布に特化しているため，一様分布において PCH は本来の性能を発揮できない．そこで，一様分布については PCH そのものではなく，提案手法において 2 値情報を用いたものに相当する手法を Binary-Valued Hashing(BVH) と名付けて用いた．BVH の 2 値情報では，各ハッシュ関数 h_i で参照された点には 1 という重みを与え，参照されなかった点には 0 という重みを与える．各手法の比較を表 1 にまとめておく．一様分布なデータは，各要素の値が $[0, 10000]$ の範囲で一様に分布する 100 次元，10 万点の人工データを作成した．正規分布に従うデータは，平均 μ，分散 σ の正規分布に従う 100 次元，10 万点の人工データである．ただし，μ は 0〜100，σ は 10〜110 の範囲からランダムに選ぶ．実験で用いたパラメータは表 2 にまとめた．

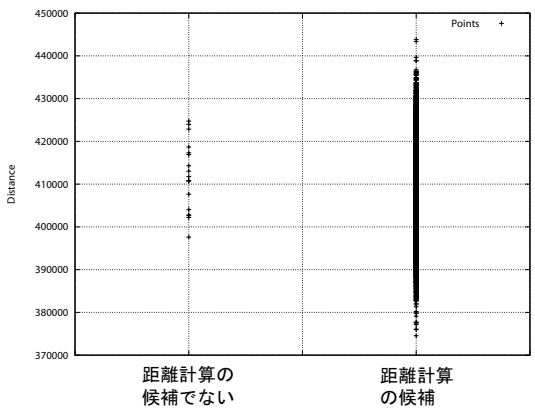
図 3(a)，図 3(b) は，LSH の場合である．横軸は，距離計算の候補か距離計算の候補でないかを表し，縦軸は L_2 距離である．図 3(a)，図 3(b) より，LSH は，一様分布と正規分布の両データに対して，距離計算の候補になった点の中にもクエリから近いものも遠いものもあることが分かる．つまり，LSH では，距離計算の候補を効率的に選定できていないといえる．図 4(a)，図 4(b) は，BVH と PCH にクエリを与えた時の重複度と実際の距離の関係を示している．横軸は各点に対する重複度，縦軸は L_2 距離である．図 4(a) より，BVH は，一様分布に対して，重複度が大きい点はクエリに近いという傾向にある．しかし，図 4(b) より，正規分布に対しては，重複度が高くてもクエリに近くない傾向にあることがわかる．つまり，BVH の距離計算候補の選定能力は一様分布に対しては高かったが，PCH の距離計算候補の選定能力は正規分布に対してはあまり高くなかった．PCH の選定能力が高くなかった理由としては，正規分布の累積分布関数 P_g によってハッシュの各バケットにデータ点が格納される確率を一定になるようにしているが，この P_g による誤差が影響して，一様にならなかったことが考えられる．図 5(a)，図 5(b) は，MVH にクエリを与えた時の距離の概算値と実際の距離の関係を表している．横軸は各点に対する距離の概算値，縦軸は L_2 距離である．図 5(a)，図 5(b) より，MVH は一様分布と正規分布の両データに対して，距離の概算値の値が小さいとクエリと近いという傾向にある．つまり，MVH の距離計算候補の選定能力は，一様分布と正規分布の両データに対してある程度有効であると言える．

4.2 実　験　2

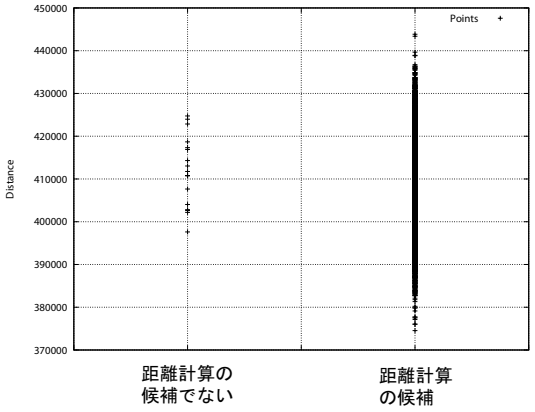
一様分布に従うデータと正規分布に従うデータに対して，LSH と BVH と PCH と提案手法の性能比較実験を行った．用いたデータは実験 1 と同じである．実験で用いたパラメータは表 3 の範囲から選び，その組み合わせをいくつか試した．実験結果のグラフは，横軸は計算時間 (ms)，縦軸はメモリ使用量 (MB) で出力し，計算時間は 1 クエリ当たりの平均時間とした．

図 6(a) は，一様分布なデータに対するもので，精度が 98%〜99%の時の計算時間とメモリ使用量を示している．図 6(b) は，正規分布に従うデータに対するもので，精度が 98%〜99%の時の計算時間とメモリ使用量を示している．図 6(c) は，一様分布なデータに対するもので，精度が 50%〜55%の時の計算時間とメモリ使用量を示している．図 6(d) は，正規分布に従うデータに対するもので，精度が 50%〜55%の時の計算時間とメモリ使用量を示している．

図 6(a)，図 6(c) より，一様分布のデータに対して，MVH1 は LSH に比べて計算時間を約 50%削減できたが，BVH とは同程度であることが分かる．MVH2 は LSH や BVH に比べてメモリ使用量を約 40%削減できたことが分かる．図 6(b)，図 6(d) より，正規分布のデータに対して，MVH1 は LSH や PCH に比べて計算時間を約 30%削減できたことが分かる．MVH2 は LSH に比べてメモリ使用量を約 40%削減でき，PCH に比べて約 50%削減できたことが分かる．以上より，MVH1 は既存手法よりも計算時間で概ね優位性があり，MVH2 は既存手法よりもメモリ使用量で優位性があるといえる．ただし，提案手法の

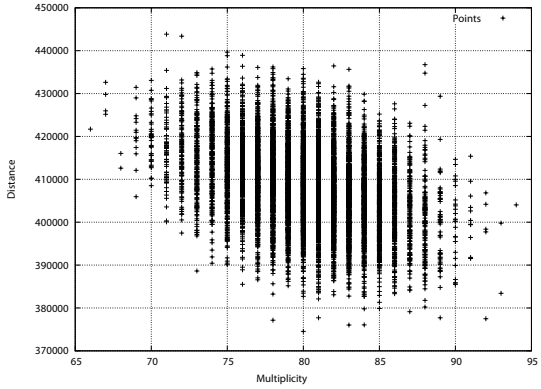


(a) 一様分布，$k = 3$，$L = 3$，$w = 20000$

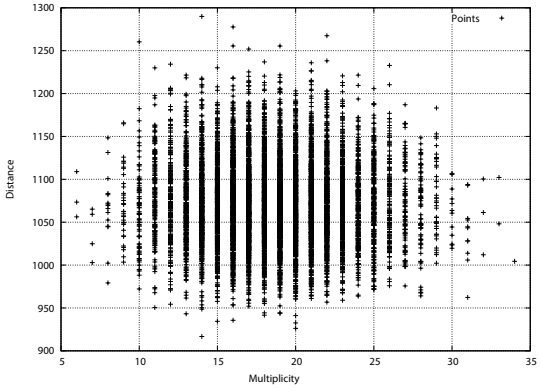


(b) 正規分布，$k = 3$，$L = 3$，$w = 20000$

図 3: LSH の距離計算候補と距離

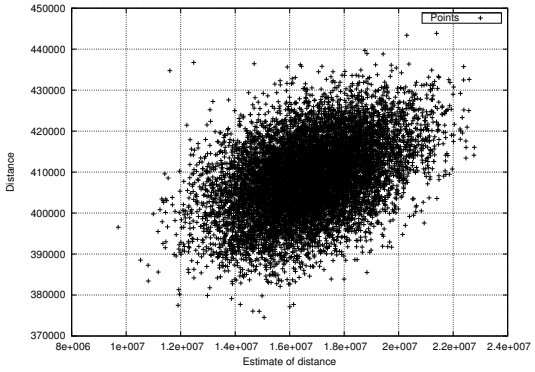


(a) BVH：一様分布，$k = 100$，$w = 1000$，$\delta = 5$

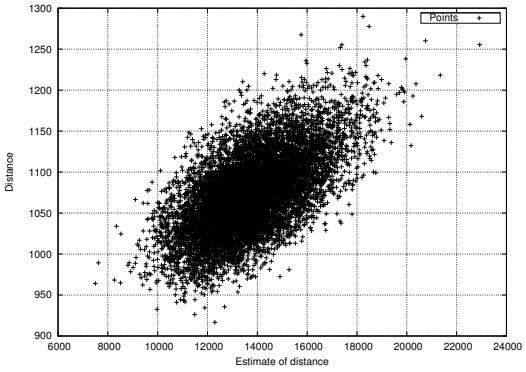


(b) PCH：正規分布，$k = 100$，$w = 0.01$，$\delta = 10$

図 4: BVH と PCH の重複度と距離



(a) 一様分布，$k = 100$，$w = 10$，$s = 1000$



(b) 正規分布，$k = 100$，$w = 1$，$s = 1000$

図 5: MVH の距離の概算値と距離

表 1: 各手法の比較

	距離計算	射影する軸	ハッシュで用いる情報	隣接するハッシュのビンの参照
LSH	あり	斜交基底	2 値情報	しない
BVH,PCH	あり	正規直交基底	2 値情報	する
MVH1	あり	正規直交基底	多値情報	する
MVH2	なし	正規直交基底	多値情報	する

表 2: 実験 1 で用いたパラメータ

	用いたデータ	k	L	w	δ	s
LSH	一様分布，正規分布	3	3	20000		
BVH	一様分布	100		1000	5	
PCH	正規分布	100		0.01	10	
MVH	一様分布	100		10		1000
MVH	規分布	100		1		1000

表 3: 実験 2 で用いたパラメータ

	用いたデータ	k	L	w	δ	s	v
LSH	一様分布	1〜10	1〜10	1000〜50000			
LSH	正規分布	1〜10	1〜10	1〜50000			
BVH	一様分布	1〜100		10〜10000	0〜5		0〜1
PCH	正規分布	10〜100		0.01〜1	0〜100		0〜1
MVH1	一様分布	1〜100		1〜10000		0〜10000	0〜1
MVH1	正規分布	1〜100		1〜100		0〜10000	0〜1
MVH2	一様分布	1〜100		1〜10000		0〜10000	
MVH2	正規分布	1〜100		0.01〜100		0〜10000	

実装上の問題も明らかになった．

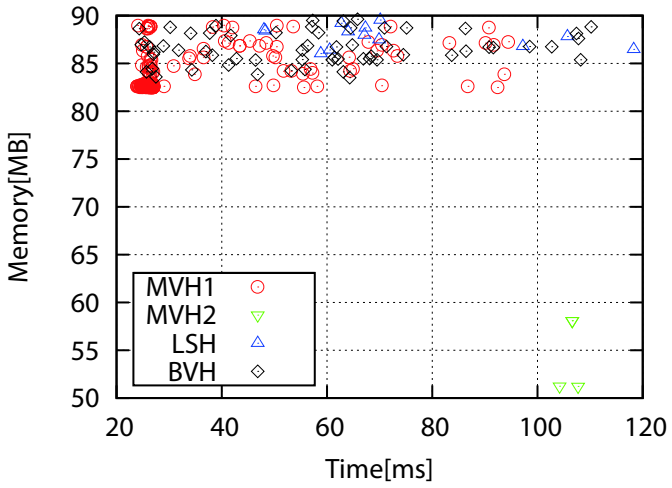
5. 結　　び

本稿では，近似最近傍探索問題において，同一の探索精度を従来手法より小さい計算時間とメモリ使用量で実現する方法を 2 通り検討した．1 つ目の提案手法では，近さの情報を多値化することでクエリの近傍での L_p 距離の概算値を求め，距離計算の対象を効率的に選定できるようにした．2 つ目の提案手法では，クエリの近傍での L_p 距離の概算値を利用して距離計算の処理を省いた．実験より，1 つ目の提案手法では LSH や PCH に比べて計算時間を約 30%〜50%削減ができた．2 つ目の提案手法では LSH や PCH に比べてメモリ使用量を約 40%〜50%削減できた．今後の課題としては，

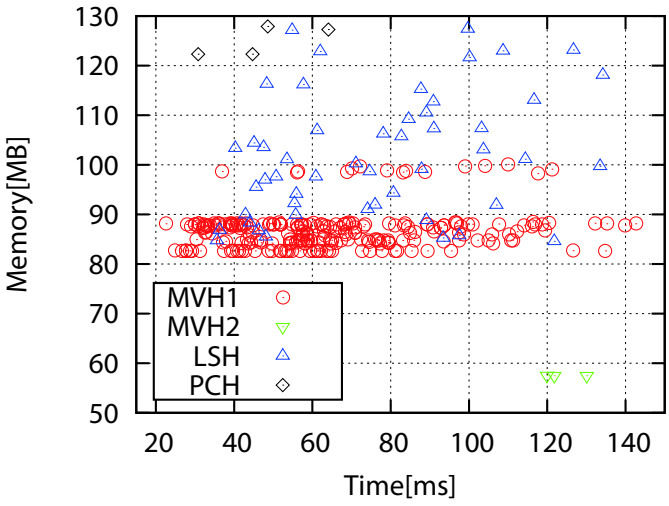
- 人工データ以外のデータに対して実験を行う．
- 提案手法の理論解析
- 木構造の手法である ANN [11]，A-PCH [10] など，ハッシュを用いる手法以外のものと性能比較実験を行う．

などが挙げられる．

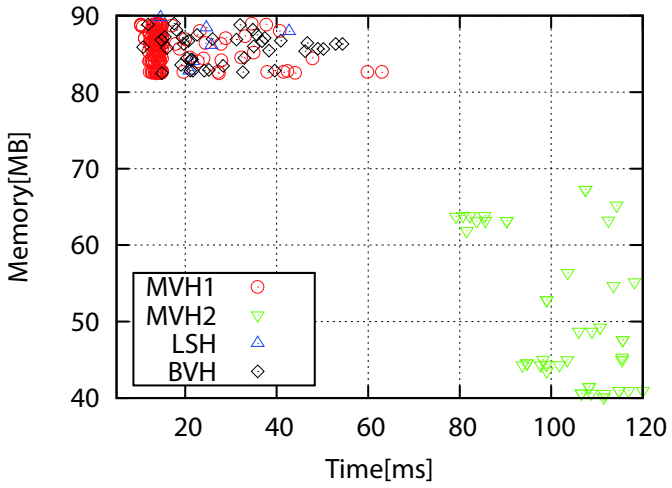
謝辞　本研究の一部は，平成 20 年度 SCAT 研究費助成ならびに科研費補助金基盤研究 (B)(19300062) の補助による．



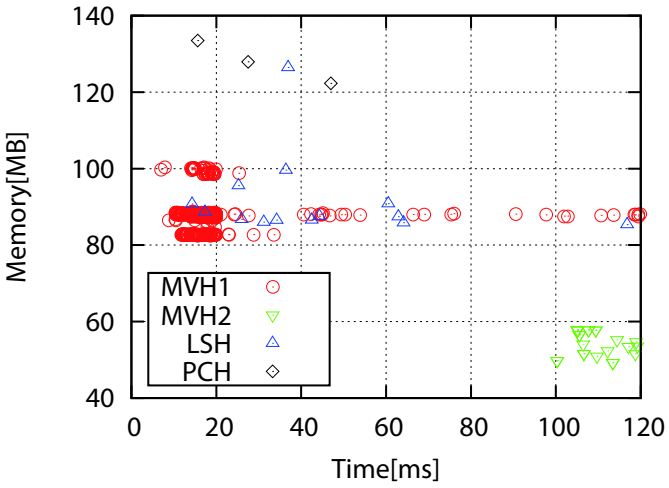
(a) 一様分布，精度:98%～99%の計算時間とメモリ使用量



(b) 正規分布，精度:98%～99%の計算時間とメモリ使用量



(c) 一様分布，精度:50%～55%の計算時間とメモリ使用量



(d) 正規分布，精度:50%～55%の計算時間とメモリ使用量

図 6: 一様分布と正規分布に対する性能評価

文　　献

[1] 多田匡志，武藤大志，岩村雅一，黄瀬浩一，“近さの多段階表現に基づく近似最近傍探索，” 信学技報 PRMU2009-110，Nov. 2009.

[2] H. Jégou, M. Douze, and C. Schmid, “Packing bag-of-features,” Proc. Int’l Conference on Computer Vision 2009, pp.2357–2364, Sept. 2009.

[3] K. Kise, K. Noguchi, and M. Iwamura, “Robust and efficient recognition of low-quality images by cascaded recognizers with massive local features,” Proc. 1st Int’l Workshop on Emergent Issues in Large Amount of Visual Data (WS-LAVD), pp.2125–2132, Oct. 2009.

[4] D. Lowe, “Distinctive image features from scale-invariant,” Int’l Journal of Computer Vision, vol.60, no.2, pp.91–110, 2004.

[5] Y. Ke and R. Sukthankar, “PCA-SIFT: A more distinctive representation for local image descriptors,” Proc. Computer Vision and Pattern Recognition2004, pp.506–513, 2004.

[6] P. Indyk and R. Motwani, “Approximate nearest neighbor: Towards removing the curse of dimensionality,” Proc. 30th Symposium on Theory of Computing, pp.604–613, 1998.

[7] M. Datar, N. Immorlica, P. Indyk, and V.S. Mirrokni, “Locality-sensitive hashing scheme based on p-stable distributions,” Proc. 20th annual symposium on Computational geometry, pp.253–262, 2004.

[8] A. Andoni and P. Indyk, “Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions,” Proc. 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS’06), pp.459–468, Oct. 2006.

[9] 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，July 2007.

[10] 松下裕輔，和田俊和，“一般分布に対する Principal Component Hashing，” 信学技報 PRMU2007-290，March 2008.

[11] S. Arya, D.M. Mount, N.S. Netanyahu, R. Silverman, and A.Y. Wu, “An optimal algorithm for approximate nearest neighbor searching in fixed dimensions,” Journal of the ACM, vol.45, no.6, pp.891–923, Nov. 1998.