縮小構造Sketchと質問点との距離下限値を利用した近似検索の高性能化に関する研究

樋口 直哉[†1,a)]　今村 安伸[†1]　篠原 武[†1]

概要：多次元データの近似検索のための縮小構造 Sketch について考察する．Sketch は局所性鋭敏型ハッシュ (locality sensitive hash) の一種である．Sketch は基礎分割関数を用いて作成され，実空間での類似性をある程度保持するようにオブジェクトをバイナリ文字列で表したものである．基礎分割関数はデータを均等に分割することで検索精度が良くなることが知られている．本論文では，ランダムに選んだオブジェクトをデータ中央値によって 2 値量子化した点を中心点とし，中心点と中央値座標との距離を半径とする基礎分割関数を提案し，量子化を用いることの有効性を検証する．また，Sketch の最適化のために焼きなまし法が有効であることや，Sketch と質問点との距離下限を用いた検索優先順序付けが大幅に精度を向上できることも示す．

キーワード：近似検索，局所性鋭敏型ハッシュ，球面分割，2 値量子化，焼きなまし法，距離下限値，検索優先順序付け

A study on high-performance of similarity search using distance lower bounds between compressed sketches and queries

Naoya Higuchi[†1,a)]　Yasunobu Imamura[†1]　Takeshi Shinohara[†1]

Abstract: In this paper, we consider compressed sketches for approximate similarity search of multimedia data. Sketches are bit strings constructed by basic partitioning functions, which associate 0 or 1 with objects. Sketch functions are considered as a kind of LSH (locality sensitive hash). We adopt ball partitioning functions which separate objects by whether or not inside of balls. We propose a binary quantized object, which is a point with max or min coordinate values depending on whether original values greater than the median of the coordinate as a center of the ball, and the distance between the center and the median of objects as the radius. Since equally separating partitioning functions provide similarity search of higher precision, proposed sketches have advantages. We report the performance of proposed method by experiments. Furthermore, we show that simulated annealing works well in optimizing sketches and the visiting order of objects using lower bound of distance between sketches and queries improves search precision.

Keywords: approximate similarity search of multimedia data, locality sensitive hash, ball partitioning functions, binary quantization, simulated annealing, order of precedence in searching

1. はじめに

計算機の演算能力や記憶容量の向上により，大量のマルチメディアデータを用いたシステムが多く作られている．そのため，膨大なデータの中から必要なデータだけを高速に探し出す情報検索技術が重要である．マルチメディアデータは多くの場合でかなり高次元であり，また劣化や加工も多く，完全一致検索を行うことは難しい．そのため，それらのデータの高速な検索を実現するために，近似検索を用いることが一般的である．近似検索の一般的な検索手法として，R-tree[1], [2] や M-tree[3] などの索引構造を用

[†1] 現在，九州工業大学
Presently with Department of Artificial Intelligence, Kyushu Institute of Technology
[a)] p676018n@mail.kyutech.jp

いるものがある．それらの空間索引は，空間の次元が大きくなると次元の呪いと呼ばれる現象により，検索効率が悪化してしまうことが知られている．そのため，高次元のデータに索引構造を用いる場合，次元縮小を行うことで次元の呪いを緩和する．しかし，空間索引を用いた検索では，データベース内に近似データが存在する質問に対しては非常に高速に検索できる反面，近似データが存在しない質問に対しては検索速度が遅くなる．

過去に次元縮小射影 Simple-Map(S-Map)[4] を用いた R-tree による近似検索が本研究室では提案されているが，本論文では近年考案された検索手法として，高速かつ一定時間で検索を行うことが可能である Sketch[5], [6], [7], [8] を用いる．Sketch は一般化超平面分割 (GHP)[7] 等の基礎分割関数を用いることで，オブジェクトの類似度をある程度保持したまま，オブジェクトをバイナリ文字列で表現する．Sketch 間の距離にはハミング距離を用いるため，ビット演算による高速な検索が可能である．Sketch は実空間上での類似性を完全には保持しないため，Sketch 上での最近傍解が実空間上での最近傍解と等しいとは限らない．そこで Sketch 用いた検索では，まず Sketch をフィルタリングとして用いることで K 個の解候補を取り出す．次にその K 個の解候補に対して実距離計算を行うことで近傍解を得る．基礎分割関数はデータを均等に分割することで検索精度が良くなることが知られている．本論文では，基礎分割関数としてピボットと半径を与えて分割する，球面分割をについて考察する．ランダムに選んだオブジェクトをデータ中央値によって 2 値量子化した点をピボットとし，ピボットと中央値座標との距離を半径とする手法を提案し，量子化を用いることの有効性を検証する．また，ピボット探索における最適化や検索時の検索優先順位付けに関する検証も行う．

第 2 章では，Sketch について紹介する．第 3 章では Sketch 作成のためのピボット探索の最適化手法を紹介し，第 4 章で提案を行う．第 5 章で実験を行い，第 6 章でまとめる．

2. Sketch

2.1 近似検索システム

近似検索システムとは，質問点に近似するデータをデータベースから取り出すシステムのことである．データ間に非近似度 (距離) を定義し，質問点からの距離の順番でオブジェクトを取り出すことにより，近似検索を実現することができる．

近似検索システムにおいて，データベースが対象とする特徴空間全体を $\mathcal{U} = \mathbb{R}^n$ とする．ここで，n は特徴データの次元数である．任意の 2 点間のオブジェクト間の非近似度を示す距離関数を $d : \mathcal{U} \times \mathcal{U} \to \mathbb{R}^+$ とし，$\mathcal{D} = (\mathcal{U}, d)$ を距離空間と呼ぶことにする．近似検索システムでは，距離関数 d は距離の公理と呼ばれる条件を満たすものと仮定する．最も重要な条件は三角不等式と呼ばれる以下の条件である．

$$d(X, Y) \leq d(X, Z) + d(Z, Y)$$

ここで，$X, Y, Z \in \mathcal{U}$ である．

近似検索には，主に範囲質問および近傍質問の 2 種類の方法が用いられている．質問点 q と半径 $r \in \mathbb{R}^+$ を質問のパラメータとする**範囲質問** $Range(\mathcal{D}, q, r)$ は，q から距離 r 以内のオブジェクトを取得する質問である．ここで，データベース内のオブジェクトの集合を S とする．すなわち，

$$Range(\mathcal{D}, q, r) = \{o \in \mathcal{S} | d(o, q) \leq r\}$$

である．

また，質問点 q から距離が小さいものから k 個のオブジェクトを取得する質問 $NN(\mathcal{D}, q, k)$ を k **近傍質問**という．さらに，上記の二つの質問方法を組み合わせた**範囲限定** k **近傍質問** $k\text{-}NNRange(\mathcal{D}, q, r, k)$ がある．これは，質問点 q から距離が r 以内のものの中で，距離が近い順番にオブジェクトを k 個取得する．つまり，

$$NNRange(\mathcal{D}, q, r, k) = \mathrm{NN}(\mathcal{D}, q, k) \cap \mathrm{Range}(\mathcal{D}, q, r)$$

である．範囲限定最近傍質問は，一定の距離以上の解が不必要であるときに用いると効果的である．

2.2 Sketch

Sketch を用いた検索は，検索の精度を犠牲にしつつ，検索を高速化する近似アルゴリズム [9], [10], [11], [12] の一種である．Sketch は，実空間での類似性をある程度保持するように基礎分割関数を用いてオブジェクトをバイナリ文字列で表現したものである．オブジェクト x の Sketch を $\sigma(x)$，その第 i ビットを $\sigma_i(x)$ で表す．Sketch 間の距離はハミング距離を用いて計測する．下の式は長さ m の Sketch 間の距離である．

$$d_\sigma(x, y) = \sum_{i=1}^{m} |\sigma_i(x) - \sigma_i(y)| \qquad (1)$$

Sketch はバイナリ文字列で表現されているため，(1) の式はビット演算で高速に計算できる．

2.3 検索法

Sketch を用いた k 近傍検索は 2 段階で行う．

(1) 質問データから作成した Sketch とデータベース内に登録されているデータの Sketch との距離を (1) の式を用いて計算し，実距離計算する候補データを取りだす．
(2) 取り出した候補データと質問データとの実空間上の距離を計算する．

最初の段階における Sketch データベースに対する検索

では，全探索を用いて K 近傍質問 $(K \geq k \geq 1)$ を行う．通常，全探索は索引構造を用いた場合と比較して，非常に検索に時間がかかるが，Sketch における全検索では，Sketch は元のデータに比べて小さく圧縮されており，距離を高速に計算することができるため，非常に高速に解を得ることが可能である．次に，得られた解を用いて，実際の距離に基づいた解を生成する [13]． Sketch は，実空間上の類似性を完全に保持できるわけではないため，Sketch を用いた検索では精度が悪化する．したがって，より良い類似性を保持した Sketch を作成するため基礎分割関数の選択が重要である．

2.4 球面分割 (BP)

本章では，任意の距離空間における Sketch を生成するための基礎分割関数を紹介する．基礎分割関数は，球面分割 (BP)[14] を用いる．基礎分割関数を m 回適用することで長さ m の Sketch が作成できる．BP では，ピボットと半径を用いて空間を二分割する．

BP は，球を用いた空間分割手法であり，以下のように定義される．オブジェクト集合 $S \in \mathcal{U}$ とピボット $(p) \in \mathcal{U}$, ピボットとそれぞれのオブジェクト $o \in S$ との距離の中央値である半径 R が与えられたとすると，S は以下のように部分空間 $S_{p_{in}}, S_{p_{out}}$ に分割される．

$$S_{p_{in}} = \{o \in S | d(p, o) \leq R\}$$
$$S_{p_{out}} = \{o \in S | d(p, o) > R\}$$

図 1 は，L_1 空間において集合 $S = \{A, B, C, D, E, F, G, H, I, J\}$ を分割した例である．L_1 データを用いているため，半径も L_1 で求めている．

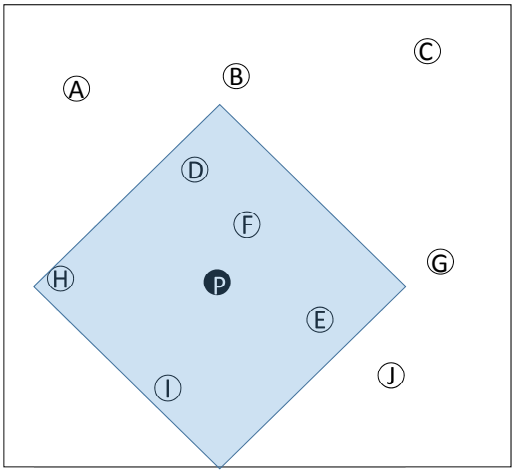

図 1 L_1 空間における BP

図 1 では，ピボット P により，集合 S は，$S_{P_{in}} = \{D, E, F, H, I\}$ と $S_{P_{out}} = \{A, B, C, G, J\}$ に分割される．オブジェクト x から基礎分割関数 σ を用いて生成される長さ m の Sketch を $\sigma(x) \in \{1, 0\}^m$ とすると，BP では Sketch の各ビット $\sigma_i(x)$ はピボット (p_i) を用いて以下のように定義される．

$$\sigma_i(x) = \begin{cases} 0, \text{if } d(p_i, x) \leq R \\ 1, \text{if } d(p_i, x) > R \end{cases} (i = 1, 2, ..., m) \qquad (2)$$

3. 最適化手法

検索精度向上ために，ピボットの選択における最適化が重要である．従来の最適化では，ピボット候補としてデータベースからランダムにオブジェクトを取り出し，そのピボット候補を評価関数によって評価する．同様の処理を任意の回数行い，最もスコアの良かったピボット候補をピボットとして決定することで行っていた．ピボットの評価には最小衝突法を用いている．本来異なるオブジェクトが Sketch 上において同じバイナリ文字列になることを衝突と呼び，この衝突が少ないほど良いピボットと見なす評価手法である．

3.1 焼きなまし法

少数再サンプリングによる焼きなまし法 [15] を用いる．従来の焼きなまし法では，温度を用いて高温時にランダムウォーク，徐々に温度が下がるにつれて局所探索を行うようにする．少数再サンプリングによる焼きなまし法は温度の代わりにサンプル数を変化させることで焼きなましを行う．高温時には少ないサンプルを用いることで誤評価を発生させ，従来の焼きなましにおけるランダムウォークを実現しており，低温時には多くのサンプルに対して局所探索を行うことになる．また，試行の度にサンプルの取り換えを行う．この手法は高温時には少ないサンプルに対してのみ評価を行うため，従来の焼きなましより高速に焼きなますことが可能である．

具体的には，本研究では 64 次元に特徴抽出した各次元 0 から 255 の座標値を持つデータを用い，32bit の Sketch を作成することを考える．データベース内からピボットをランダムに 32 個選び，焼きなましを開始する．32 個のピボットから一つを選んで，そのピボットの 64 次元から一つを選び，その座標値を最小から最大まで変化させてスコアが最良のものを選ぶ．最良のものが複数あるときは，ランダムに選ぶようにした．スコア計算では，一部の値しか変化しないことを利用して，256 の候補のスコアを効率よく計算できるように工夫した（計算途中結果の再利用，部分スコアの利用など）．これにより，より多くの候補から遷移先を選択できるようにして，焼きなまし法がより効果的に動作するようになった（とくに，高温時に相当する小数のサンプルでの評価を用いる場合のランダムウォークで多くの点を走査できるようになったと考えている）．

4. 提案手法

4.1 量子化球面分割 (QBP)

2 章で紹介したように Sketch を用いた検索では Sketch を作成する基礎分割関数によって，検索性能が大きく変化する．本章では，既存の基礎分割関数として知られている球面分割 (BP) において新たな中心点探索方法を提案する．BP は中央値付近のデータの多くが球の内側となってしまうため，衝突が多くなってしまうことが考えられる．そこでこの問題を解決するために，Simple-Map において良い中心点選択法として知られている 2 値量子化法 [16] を BP に適用した，量子化球面分割 (QBP) を提案する．QBP は、ランダムに選んだオブジェクトをデータ中央値によって 2 値量子化した点をピボットとし，ピボットと中央値座標との距離を半径とする手法である．

オブジェクト集合 $S \in \mathcal{U}$ とピボット $(p) \in \mathcal{U}$, ピボットと中央値座標との距離である半径 R が与えられたとすると，S は以下のように部分空間 $S_{p_{in}}, S_{p_{out}}$ に分割される．

$$S_{p_{in}} = \{o \in S | d(p,o) \leq R\}$$
$$S_{p_{out}} = \{o \in S | d(p,o) > R\}$$

図 2 は，L_1 空間おいて集合 $S = \{A,B,C,D,E,F,G,H,I,J\}$ を QBP によって分割した例である．ここで，P はランダムに選ばれたオブジェクト，P' は P を 2 値量子化したピボット，M はデータ中央値座標を示している．

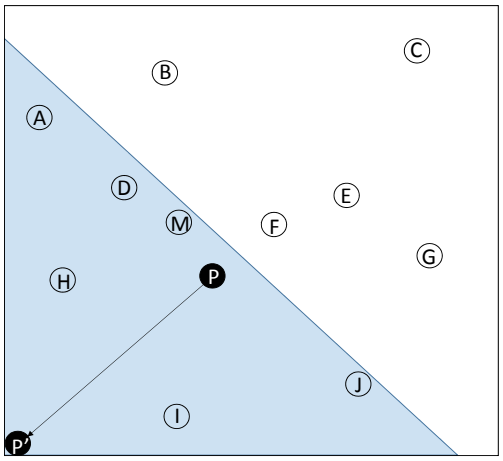

図 2 L_1 空間における QBP

図 2 では, ピボット P' により，集合 S は，$S_{P'_{in}} = \{A,D,H,I,J\}$ と $S_{P'_{out}} = \{B,C,E,F,G\}$ に分割される．オブジェクト x から基礎分割関数 σ を用いて生成される長さ m の Sketch を $\sigma(x) \in \{1,0\}^m$ とすると，BP では Sketch の各ビット $\sigma_i(x)$ はピボット (p_i) を用いて以下のように定義される．

$$\sigma_i(x) = \begin{cases} 0, \text{if } d(p_i,x) \leq R \\ 1, \text{if } d(p_i,x) > R \end{cases} (i = 1,2,...,m) \quad (3)$$

4.2 距離下限を用いた検索優先順位付け

Sketch はハミング距離で近いもの K 個に対して実距離計算を行うことで近傍解を得る．ここでは，ハミング距離ではなく距離の下限値を用いたスコア付けにより K 個を選ぶ手法を提案する．オブジェクト x，質問点 q，第 i ビットのためのピボット p_i，半径 r_i を用いると，距離の下限値 e_i は

$$e_i = \begin{cases} |d(p_i,q) - r_i| & (\sigma_i(q) \neq \sigma_i(x)) \\ 0 & (\sigma_i(q) = \sigma_i(x)) \end{cases} (i = 1,2,...,t)$$

となり距離の下限を見積もることができる．この下限を求めるために，データベース内のオブジェクト x については Sketch しか用いないが，質問点 q については，ピボットとの実距離を用いることができるので，三角不等式により，$d(q,x)$ の下限を求めることができる．$\sigma_i(x) = 0, \sigma_i(q) = 1$ のとき，つまり，

$$d(p_i,x) \leq r_i < d(p_i,q) \quad (4)$$

のときを，図 3 に示す．

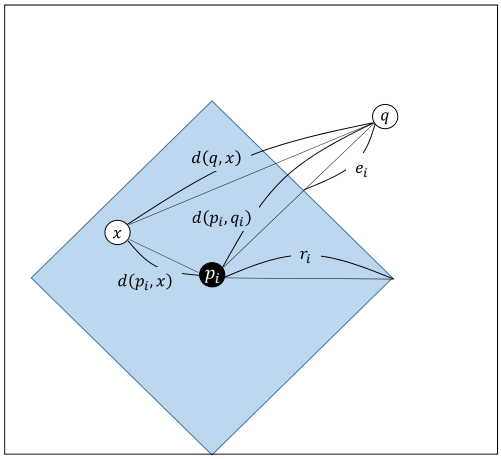

図 3 L_1 空間における距離下限値

三角不等式より，

$$d(q,x) \geq d(p_i,q) - d(p_i,x) \quad (5)$$

(4), (5) より，

$$d(q,x) \geq d(p_i,q) - r_i \quad (6)$$

逆に，$\sigma_i(x) = 1$, $\sigma_i(q) = 0$ のときは，

$$d(q,x) \geq r_i - d(p_i,q) \quad (7)$$

以上により，x と q の距離の下限 e_i が次式で与えらるれる

ことがわかる．

$$e_i = |d(p_i, q) - r_i| \quad (\sigma_i(x) \neq \sigma_i(q)) \qquad (8)$$

実際には Sketch 全体の距離の下限値を用いるので，スコア e は

$$e = max_{i=1}^{t} e_i,$$

のように L_∞ で見積もる．また，つぎのように L_1 や L_2 として下限値を総合すると，理論的には下限ではあることは保証できなくなる．

$$e = \begin{cases} \sum_{i=1}^{t} e_i & (L_1) \\ \sqrt{\sum_{i=1}^{t} e_i^2} & (L_2) \end{cases}$$

ここで，次節の実験で用いるデータを用いて，従来のハミング距離，L_∞，L_1，L_2 それぞれについて BP による射影前後の距離の相関係数を求めた結果を表 1 に示す．

表 1 距離空間ごとの相関係数

ハミング	L_∞	L_1	L_2
0.398	0.511	0.586	0.602

このように，距離の相関を調べてみると，L_∞ より L_1，L_2 で見積もると相関が高くなることから，L_1，L_2 を用いる方が検索の精度が向上することが期待できる．これらのスコアが小さい順に検索を行う手法を提案する．この距離下限 e_i を用いたスコア（L_∞，L_1，L_2）の計算は，ビットパターンに対する部分スコアを表（配列を用いる）関数とすることで高速に計算することができるので，ハミング距離の距離計算と比べてもほとんど変わらない検索速度を達成できる．実距離計算では，検索範囲を与えて K 個に対してスコアの小さい順に実距離計算を行う．このとき検索範囲内であれば検索範囲をその距離まで収縮し，計算途中で検索範囲外であるとわかったオブジェクトに関してはその時点で計算を打ち切る．距離下限を用いたスコアで順序付けを行うと，特に近質問に関してこの検索範囲の収縮が速くなり，検索全体（スコア計算 + 実距離計算）では従来よりも距離下限を用いる方が速くなる．

5. 実験

実験データとして，約 2,900 本の動画から特徴抽出した約 700 万件の 64 次元画像フレームデータを用いる．質問データとしては，データベース内によく似たデータがある近質問，やや似たデータがある準近質問，似たデータがない遠質問，各約 3 万件の計約 9 万件のデータを用いる．このデータに対して，比較対象となる従来手法の BP と提案手法 QBP を用いた 32bit の Sketch の解精度を検証する．質問方法として，範囲限定最近傍質問を採用し，近質問と遠質問をうまく分類する $r = 500$ を採用する．実距離計算回数として，$K = 1000$ を採用する．

5.1 BP と QBP の比較

BP，QBP においてランダムに選んだ 1 万個の評価用サンプルに対して最小衝突法による評価で Sketch を作成し，その時の平均衝突確率と検索時の平均正答率をそれぞれ表 2 に示す．ここでの正答率とは，Sketch 上での K 近傍解に実空間上での最近傍解が含まれる確率である．表には，手法 X について $E(X) \pm \sigma(X)$ で表記する．このとき二つの手法の差を明白にするために，最適化は行わないものとする．

表 2 BP と QBP における衝突確率と正答率

基礎分割関数	衝突確率 [$\times 10^{-5}$]	正答率 [%]
BP	67.8 ± 4.03	72.8 ± 0.411
QBP	1.53 ± 0.933	83.5 ± 1.79

表 2 より，QBP は BP に対して衝突確率が低く，また検索精度が高いことがわかる．前述したように，BP は中央値付近のデータの多くが球の内側となってしまうため，衝突が多くなってしまうことが考えられる．そこで，全てのピボット選択が終わったあと，評価用のサンプルを中央値から近いオブジェクト 1 万個に取り換えて，再度評価し直す実験を行い，その結果をそれぞれ表 3，表 4 に示す．このときの評価には最小衝突法と Sketch 間のハミング距離の総和の 2 種類で行った．ハミング距離の評価では 1bit 辺りの平均距離を示している．

表 3 BP における中央値付近のデータの衝突確率とハミング距離

衝突確率 [$\times 10^{-1}$]	ハミング距離 [$\times 10^{-2}$]
3.20 ± 1.57	3.68 ± 1.30

表 4 QBP における中央値付近のデータの衝突確率とハミング距離

衝突確率 [$\times 10^{-5}$]	ハミング距離 [$\times 10^{-2}$]
51.3 ± 7.66	39.6 ± 2.20

表 3，表 4 より，中央値付近のデータに関しては BP は衝突がかなり多いことがわかる．ハミング距離に関しては QBP の方が大きく，これらの結果より中央値付近に関しては QBP の方がしっかりと分離できている．

5.2 焼きなまし法の有効性

次に BP に焼きなまし法を適用し，表 5 に示す．比較のため，ピボット探索にかかる時間が同程度となるように以下のように設定する．

- 従来：各次元試行回数 1000 回
- 焼きなまし法：初期サンプル 100，総試行回数 1 万回，局所探索 2028 回

焼きなまし法における局所探索の回数は各ピボットの各次元 1 回ずつ変更を行える程度，2028 回 (ピボット数 × ピボットの特徴次元数) にしている．焼きなまし法は初期値に依存しないため，基礎分割関数は BP で実験を行う．

表 5　焼きなまし法の適用

最適化	衝突確率 [$\times 10^{-7}$]	正答率 [%]
従来 (BP)	25.9 ± 6.20	88.5 ± 1.05
従来 (QBP)	22.0 ± 4.95	90.9 ± 0.620
焼きなまし法	10.2 ± 4.33	87.8 ± 0.855

焼きなまし法を適用することにより，従来の最適化手法よりも上手く最適化できていることがわかる．しかし検索時の正答率はよくなっていないため，評価関数の見直しが必要である．

5.3　検索優先順位付けの比較

次に距離下限による検索優先順位付けを行う．解候補数 K を変化させ，それぞれの場合の正答率を表 6 に示す．

表 6　検索優先順位付けの適用

最適化	優先順位	$K = 1000$	$K = 7000$
従来 (BP)	ハミング	88.5 ± 1.05	95.2 ± 0.675
従来 (BP)	L_1	95.0 ± 0.610	98.5 ± 0.390
従来 (BP)	L_2	94.9 ± 0.641	98.6 ± 0.402
従来 (QBP)	ハミング	90.9 ± 0.620	96.6 ± 0.626
従来 (QBP)	L_1	96.9 ± 0.421	99.4 ± 0.199
従来 (QBP)	L_2	96.8 ± 0.519	99.4 ± 0.235
焼きなまし法	ハミング	87.8 ± 0.855	94.5 ± 0.664
焼きなまし法	L_1	95.0 ± 0.465	98.5 ± 0.308
焼きなまし法	L_2	94.9 ± 0.423	98.3 ± 0.171

最適化手法によらず，距離下限による検索優先順位付けによって正答率が大きく上昇することがわかる．距離下限による検索優先順位付けを行う場合は，$K = 1000$ の時点でハミング距離の場合の $K = 7000$ の正答率と同等になる．つまり同等の正答率を出すために必要な実距離計算回数を少なくすることが可能なため，高速化することができる．

6.　まとめと今後の課題

QBP の BP に対する優位性を示すことができた．また，焼きなまし法が良い最適化手法であることがわかった．しかし，現在の評価関数では，評価が高いからといって必ずしも検索時の正答率が高くなるわけではないという問題がある．よって，射影前後の距離の相関を用いるなど，別の評価関数の検討が必要である．また，距離下限による検索優先順位付けは非常に大きな効果を発揮することがわかった．本論文では球面分割による Sketch しか検証できていないため，一般化超平面分割 (GHP) などの基礎分割関数においてもその効果を確かめる必要がある．射影前後の Sketch の相関が高いことがわかっている座標分割法 [17] の適用も考えられる．本実験では画像データでのみ検証を行っているため，音データやランダムデータでの検証が必要である．

参考文献

[1] G. Navarro: Searching in metric spaces by spatial approximation, The VLDB Journal, pp. 28–46, (2002).

[2] A.Guttman: R-trees: A dynamic index structure for spatial searching, Proc. ACM SIGMOD, International Conference on Management of Data, pp. 47–57, (1984).

[3] P. Ciaccia, M. Patella, P. Zezula: M-tree: An efficient access method for similarity search in metric spaces, Proc. 23rd Int. Conf. on Very Large Data Bases, pp. 426–435, (1997).

[4] T. Shinohara, H. Ishizaka: On dimension reduction mappings for approximate retrieval of multi-dimensional data, Progress in Discovery Science, pp. 224–231, (2002).

[5] Q. Lv and M. Charikar and K. Li: Image similarity search with compact data structures, Conference on Information and Knowledge Management archive, Proceedings of the thirteenth ACM international conference on Information and knowledge management", pp. 208–217, (2004).

[6] Q. Lv and W. Josephson and Z. Wang and M. Charikar and K. Li: Efficient filtering with sketches in the ferret toolkit, International Multimedia Conference archive, Proceedings of the 8th ACM international workshop on Multimedia information retrieval, pp. 279–288, (2006).

[7] A. Jose Muller-Molina and T. Shinohara: Efficient Similarity Search by Reducing I/O with Compressed Sketches, International Workshop on Similarity Search and Applications archive, Proceedings of the 2009 Second International Workshop on Similarity Search and Applications, pp. 30–38, (2009).

[8] 岩崎 瑶平，篠原 武，A.J.Mulle : Sketch による大容量記憶データに対する高速な空間検索法に関する研究，第 74 回 SIG-FPAI，(2009)．

[9] P. Zezula and G. Amato and V. Dohnal and M. Batko: Similarity Search: The Metric Space Approach, Springer Verlag, Advances in Database Systems, (2006).

[10] P. Zezula and P. Savino and G. Amato and F. Rabitti: Approximate similarity retrieval with M-trees, Springer-Verlag, The VLDB Journal - The International Journal on Very Large Data Bases, (1998).

[11] G. Amato: Approximate Similarity Search in Metric Spaces, University of Dortmund, (2002).

[12] G. Amato and F. Rabitti and P. Savino and P. Zezula: Region proximity in metric spaces and its use for approximate similarity search, ACM Press, AACM Transactions on Information Systems, (2003).

[13] 岩崎 瑶平，篠原 武 : Sketch を用いた空間検索法の精度向上に関する研究，第 76 回 SIG-FPAI，(2010)．

[14] 大野真吾，岩崎瑶平，篠原武 : 空間検索のための Sketch の基礎分割関数に関する研究，火の国情報シンポジウム，(2010)．

[15] 今村安伸，篠原武 : 焼きなまし法を用いた次元縮小射影

Simple-Map の中心点探索，電機関係学会第 69 回九州支部連合大会 (第 69 回連合大会) ，(2016).

[16] 中島正八：データベース内オブジェクトの離散化を利用した次元縮小射影 Simple-Map の中心点探索に関する研究，九州工業大学 卒業論文，(2014).

[17] 今村安伸：空間索引のための射影法 - 座標分割と成分分解について，九州工業大学大学院 修士論文，(2007).