

階層的空間索引を用いた検索高速化のための 局所次元縮小射影に関する研究

村上 義明¹ 岸川 直樹¹ 安藤 友佳^{1,a)} 篠原 武¹

概要：

R-tree や M-tree などの階層的空間索引は、高速な近似検索手法として知られている。我々はこれまでに R-tree と M-tree の特徴をあわせ持つ MR-tree の提案を行なった。本論文では、MR-tree における葉ノードの中心点選択法を提案する。画像データと音データを用いて実験を行なった結果、射影距離の総和を最大にする MAXPJT がもっとも良い中心点選択法であることがわかった。また、MR-tree は、画像データでは R-tree よりも高速な検索を実現することができたが、音データに関しては R-tree よりも検索が遅くなった。この原因として、中心点を追加することによるノードサイズの増加が考えられる。この結果を改善するために、Local R-tree(LR-tree) を提案する。LR-tree は、ノードサイズの増加を抑えつつ、R-tree の葉ノード内のデータの局所性を考慮した射影を用いている。LR-tree は R-tree よりも画像データでは約 5%、音データにおいても約 4%高速な検索を実現した。

1. はじめに

大量の多次元データを高速に近似検索する手法として、R-tree[2] や M-tree[1] などの階層的な空間索引を用いて、空間を索引付けする手法がある。R-tree は、最小包囲超矩形 (Minimum Bounding Rectangle, 以下 MBR) を用いてオブジェクトを包括するノードを用いた階層的な空間索引であり、M-tree は、最小包囲超球 (Minimum Bounding Sphere, 以下 MBS) を用いている。我々の研究室では、R-tree と M-tree の構造をあわせ持つ新しい階層的空間索引、Metric Rectangle tree(MR-tree) [7] の提案を行なった。その後の実験 [9] で、MR-tree を用いた検索では、内部ノードにおける MBS の枝刈り効果がないことが確認できた。そのため MR-tree は、内部ノードの MBS を削除し、結果として R-tree の葉ノードに、実距離計算削減に用いる M-tree の MBS の中心点を追加したものとなっている。追加した中心点を用いた実距離計算の削減では、中心点の選択方法が重要となる。

本論文では、まず、MR-tree の葉ノードに追加する中心点の選択法を提案し、それぞれの方法について比較検証を行う。画像データと音データを対象に実験を行った結果、どちらのデータにおいても、射影距離の総和を最大にする MAXPJT が最もよい中心点選択法であることが確認でき

た。また、MAXPJT を用いた MR-tree と R-tree の比較を行った結果、画像データにおいては MR-tree は R-tree よりも検索時間を約 4.9%削減できた。しかし、音データに関しては、MR-tree は R-tree よりも検索時間が若干遅くなり、検索性能として劣る結果となった [9]。

音データにおける MR-tree の検索性能の悪化の原因の一つとして、各葉ノードに新しい中心点を追加することによるノードサイズの増加が考えられる。そこで、音データにおける検索性能悪化を改善するため、局所次元縮小射影による R-tree(Local R-tree) を提案する。Local R-tree(LR-tree) は、MR-tree のように各葉ノードごとに新しい中心点を追加することなく、局所を考慮した次元縮小を行う階層的空間索引である。LR-tree の有効性を検証するため、R-tree との比較実験を行った結果、MR-tree では高速化できなかった音データにおいて、R-tree よりも約 3.9%高速な検索が実現できた。

本論文の構成は以下の通りである。2 章では近似検索、3 章では次元縮小法である S-Map について述べる。R-tree と MR-tree について 4 章で紹介し、5 章で検証を行う。6 章で Local R-tree を提案し 7 章で検証を行い、最後に全体のまとめを 8 章で記述する。

2. 近似検索

近似検索とは、データ間の非近似度 (距離) を用いて、質問点と近似するデータを取り出すことである [4]。近似検

¹ 九州工業大学大学院情報工学府情報科学専攻知能情報工学分野

^{a)} E-mail:m673002y@ai.kyutech.ac.jp

素の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 点のオブジェクト間の非近似度の指標を示す距離関数を $d: \mathcal{U} \times \mathcal{U} \rightarrow \mathbb{R}^+$ とし， $\mathcal{D} = (\mathcal{S}, d)$ を距離空間とする．本論文で扱う近似検索において，距離関数 d は距離の公理 [4] を満たすものを用いる．距離の公理の中で最も重要な条件は，以下の三角不等式である．

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

ここで， $X, Y, Z \in \mathcal{D}$ である．

次に，本実験で用いる距離計測について説明する．特徴空間内の任意のオブジェクトを x とする． x の特徴は n 個の実数の組 $(x_{(1)}, x_{(2)}, \dots, x_{(n)})$ で表される．以下の二つの距離関数は距離の公理を満たすものである．

- L_1 距離: $D(x, y) = \sum_{i=1}^n |x_{(i)} - y_{(i)}|$
- L_∞ 距離: $D(x, y) = \max_{i=1}^n |x_{(i)} - y_{(i)}|$

本論文では元の空間での距離計算 (実距離計算) を L_1 距離，3.1 章で説明する射影後の空間での距離計算 (射影距離計算) を L_∞ 距離で行う．

2.1 質問方法

近似検索では，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．本実験では，近傍質問の中でも，質問点 Q から距離が最も小さいオブジェクトを取得する最近傍質問 $NN(\mathcal{D}, Q)$ を採用する．

$$NN(\mathcal{D}, Q) = \{O_i \in \mathcal{S} \mid O_i \text{ は } d(O_i, Q) \text{ が最小のもの}\}$$

最近傍質問では初期検索範囲を無限大に設定する．そこから検索をはじめて，検索範囲内のオブジェクトを見つけた場合，そのオブジェクトを暫定解として登録し，検索範囲を暫定解と質問点との距離に収縮させる．以上の操作を検索範囲内に新たなオブジェクトが見つからなくなるまで繰り返す．

3. 次元縮小

高次元の特徴空間に対して R-tree などの階層的空間索引を用いた場合，次元の呪いの影響で，その検索性能が悪化することが知られている [8]．次元の呪いを緩和する手法として，特徴空間をより低次元空間に射影する次元縮小法がある．本論文では次元縮小法として，Simple-Map(S-Map) [3] を用いる．

3.1 Simple-Map

S-Map は射影関数として，中心点とオブジェクトの実距離を用いる．また，S-Map は距離の公理を満たすすべての距離空間に対して適用可能である．射影後の空間 (射影空間) 上では任意のオブジェクトの 2 点間の距離が，射影前

の空間 (元空間) 上の距離よりも伸びることはないが縮まる可能性がある．このことを距離の縮みと呼ぶ．

射影に用いる中心点を p とすると，任意のオブジェクト O の射影空間上での座標値は

$$\varphi_p(O) = d(p, O)$$

と定義される．三角不等式より，射影空間上の任意の 2 点のオブジェクト X, Y 間の距離を $d'(X, Y)$ とすると，射影距離 $d'(X, Y)$ は中心点を媒介することで距離の縮みが生じる可能性がある．

$$d'(X, Y) = |\varphi_p(X) - \varphi_p(Y)| \leq d(X, Y)$$

中心点を複数用いた場合に拡張する．S-Map による射影では，中心点の数が射影空間での次元数となる．中心点の集合を $P = \{p_1, p_2, \dots, p_{n'}\}$ とすると，射影関数は

$$f_P(X) = (\varphi_{p_1}(X), \varphi_{p_2}(X), \dots, \varphi_{p_{n'}}(X))$$

となる．ここで n' は射影空間での次元数である．また，複数の中心点を用いて射影された 2 点のオブジェクト間の距離は，

$$d'(X, Y) = \max_{p \in P} |\varphi_p(X) - \varphi_p(Y)| \leq d(X, Y)$$

となる．このように，射影空間上での距離は L_∞ 距離で計測する．中心点の数が多ほど，射影空間上で距離の情報をもっと保存できる．中心点を多くとることで距離の縮みを小さくできるが，射影空間が高次元になるため，射影距離計算のコストが増加し次元の呪いの影響を強く受ける．低次元空間へ射影を行うことで，高次元な元空間での距離計算よりも低いコストで距離計算を行うことができる．

3.2 中心点選択法

S-Map による射影では，中心点の選び方によって射影の効率に影響を与える．そのため，どの中心点を選択するかが重要となる．S-Map では，以下の手順で中心点を選択する．

- 対象となるデータベースから評価関数を用いて中心点を選択
- 選択した中心点に対して局所探索法?を適用しより良い中心点を選択

まず，データベースからオブジェクトをいくつか取得する．取得したオブジェクトは中心点候補となり，中心点候補を用いてデータベース内を射影する．作成された射影データに対して評価関数を適用し中心点の評価を行い，最も評価のよいものを中心点として採用する．評価関数に関しては，各オブジェクト間の射影距離の総和が最大となる中心点を選択する，MAXPJT(MAXimum ProjecTed distortion) を用いる．

選択された中心点は、データベース内のオブジェクトであり、空間全体に対してオブジェクトが存在している点は限られているため、より良い中心点を選択できていない可能性がある。そこで、選択された中心点に対し、局所探索法を適用し、中心点の各座標軸の値を変化させながら、再度評価関数を適用することで、より良い中心点を探索する。

4. R-tree

Guttman が提案した R-tree [2] は代表的な階層的空間索引の一つであり、対象とする空間を MBR (Minimum Bounding Rectangle) と呼ばれる n 次元超直方体で分割する。ここで n は特徴の次元数である。各ノードはそのすべての子ノードを包括する MBR と子ノードへのポインタを持つ。ただし、葉ノードはデータベース中のオブジェクトへのポインタを持つ。検索の際には、検索範囲と交差する MBR を持つノードのみを訪問することで、無駄なノードへの訪問を省き、高速な検索を実現する。R-tree は、高次元空間では次元の呪いにより、MBR の重なりが大きくなったり、形状が正規形から離れたりしてしまうため、検索性能が悪化することが知られている。そこで、3.1 章で述べた S-Map を用いて次元縮小した射影空間上のオブジェクトを対象に索引付けを行うことで、次元の呪いを緩和することができる。

4.1 Simple-Map を用いた R-tree

S-Map を用いた R-tree は、射影空間を対象とするため、MBR は n' 次元超直方体となる。ここで、 n' は射影空間の次元数である。従来の R-tree と同様に、各ノードはそのすべての子ノードを包括する MBR と子ノードへのポインタを持つ。葉ノードに関しては、データベース中のオブジェクトへのポインタと、オブジェクトの射影データを持つ。

S-Map による射影では、射影距離は実距離に対して縮むことはあるが、伸びることはない。つまり、元空間で近いものは必ず射影空間でも近くなり、射影空間で遠いものは必ず元空間でも遠くなる。これを利用して S-Map を用いた R-tree では効率的な検索を実現する。具体的には、検索を行う際、射影空間上で検索範囲内の候補となるオブジェクトを探す。見つかった候補に対してのみ、元空間でも検索範囲内であるかを確認するために実距離計算を行う。S-Map の性質から、射影空間上で検索範囲外のオブジェクトは、元空間でも検索範囲外となることは保証されるので、元の R-tree に対して、検索精度が落ちることはなく、低コストの射影距離計算で高コストの実距離計算を省くことができるため、高速な検索を実現することができる。以下、本論文における R-tree は、本章で紹介した S-Map を用いた R-tree とする。

4.2 MR-tree

MR-tree[7] は、M-tree の索引付けの単位である MBS と、R-tree の MBR の二つを用いる階層的空間索引である。MBS は中心点と半径で表される超球であり、MBR は座標軸に平行な超矩形で表される。元空間を MBS、射影空間を MBR で索引付けを行う。

MR-tree を用いた検索では、R-tree と M-tree の検索方法を併用するが、その後の実験 [9] から内部ノードにおいて M-tree の枝刈りに効果がないことが確認された。そのため、内部ノードの MBS を削除し、葉ノードにのみ M-tree の中心点を追加している。R-tree と同様に、内部ノードにおける検索では MBR と質問範囲が交差しているノードのみを訪問する。検索が葉ノードに到達した場合、まず、新しく追加した中心点を用いた三角不等式による枝刈りを行い、不必要な実距離計算を削減する。質問点を q 、検索範囲を r 、追加した中心点を p^M とすると、以下の条件を満たすオブジェクト o の実距離計算を削減することができる。

$$\bullet \quad |d(q, p^M) - d(o, p^M)| > r$$

それでも省くことのできないオブジェクトに対しては、R-tree と同様に、S-Map の射影データを用いた実距離計算の削減を行う。このように、葉ノードごとに新しく追加した M-tree の中心点を用いて実距離計算を削減することで、R-tree よりも多く実距離計算を削減することができ、検索の高速化が期待できる。また、M-tree の中心点を用いた実距離計算の削減は、S-Map の射影データを用いた実距離計算削減方法と同じである。結果として、本論文での MR-tree は、過去に岩本が提案した局所射影を用いた R-tree[5] と同じものとなった。

4.2.1 MR-tree の葉ノードにおける中心点選択法

MR-tree の葉ノードにおける中心点を用いた実距離計算の削減では、中心点の選択方法が重要となる。本論文では、三つの中心点の選択法を提案する。

CENT: CENTroid. ノード内のすべてのオブジェクトに対する L_1 距離の重心を中心点とする。中心点とオブジェクト間の距離が小さくなり、枝刈りの条件式から、質問点と中心点の距離が大きい質問に対して効果が期待できる。

MINSUM: MINimum Sum of Distance. 他のオブジェクトとの L_1 距離の総和が最小となるオブジェクトを中心点とする方法である。重心と同様の効果が期待できる。

MAXPJT: MAXimum ProjecTed distortion. ノード内のオブジェクトを中心点として射影を行う場合、射影距離の総和が最大となるようなオブジェクトを選択する。次元縮小法 S-Map で最もよいとされている中心点選択法であり、質問点とオブジェクトの距離の縮みの緩和に期待ができる。

5. 実験 1

それぞれの中心点選択法を用いて構築した MR-tree と

R-tree で検索性能の比較を行う．実験に用いるデータとしては画像データ [6] と音データ [10] を採用した．

画像データ [6]: 約 2,800 本の動画から切り出した, 約 700 万件の画像フレームを 64 次元に特徴抽出したものをデータベースに登録する．特徴データに対して S-Map を用いて 8 次元に射影を行った．

音データ [10]: 楽曲約 1,400 曲から切り出した, 約 700 万件の音楽フレームを 96 次元に特徴抽出したものを登録する．特徴データに対して S-Map を用いて 12 次元に射影を行った．

質問用データとして, データベース内に非常に似たデータが存在する近質問, 比較的似たデータが存在する準近質問, データベース内に似たデータが存在しない遠質問の各約 3 万件ずつを用意した．また, 三つの質問を混ぜた 9 万件の質問データを全質問と呼ぶ．比較項目としては, 平均の距離計算回数, 検索時間を比較する．訪問ノード数に関しては R-tree, MR-tree で違いはないため比較は行わない．

実験に用いた PC の性能を以下に示す．

- CPU: Intel(R) Core(TM) i7-2600 @ 3.40GHz
- メモリ: 8GBBytes

5.1 実験結果 1

5.1.1 画像データ

表 1 の結果から, 近質問, 準近質問, 遠質問, すべての質問において, MAXPJT が最も距離計算回数を削減できることが確認できた．MAXPJT を用いて構築した MR-tree は全質問に対して, R-tree に比べて距離計算回数を約 21%削減することができた．距離計算回数を多く削減したことで, 検索時間についても, MAXPJT を用いて構築した MR-tree は R-tree よりも平均で 4.9%高速化することが確認できた．

表 1 画像データ: 各質問に対する平均距離計算回数 (回)

質問	距離計算回数			
	R-tree	CENT	MINSO	MAXPJT
近	9.22×10^2	9.52×10^2	9.21×10^2	8.29×10^2
準近	1.46×10^5	1.26×10^5	1.28×10^5	1.11×10^5
遠	1.67×10^6	1.46×10^6	1.52×10^6	1.32×10^6

表 2 画像データ: 平均検索時間 (ms)

	検索時間			
	R-tree	CENT	MINSO	MAXPJT
全質問	60.4	59.7	61.0	57.5

5.1.2 音データ

表 3 の結果から画像データと同様に, すべての質問において MAXPJT が最も距離計算回数を削減できた．全質問に対しては, R-tree に比べて, 平均で距離計算回数は 2.1%減少した．しかし, 表 4 から検索時間については, MR-tree の中で最も性能のよい MAXPJT でも, R-tree より若干増加してしまう結果となり, MR-tree は R-tree より

も劣る性能となった．

表 3 音データ: 各質問に対する平均距離計算回数 (回)

質問	距離計算回数			
	R-tree	CENT	MINSO	MAXPJT
近	1.75×10^4	1.71×10^4	1.72×10^4	1.68×10^4
準近	8.52×10^5	8.55×10^5	8.51×10^5	8.42×10^5
遠	1.34×10^6	1.30×10^6	1.33×10^6	1.30×10^6

表 4 音データ: 平均検索時間 (ms)

	検索時間			
	R-tree	CENT	MINSO	MAXPJT
全質問	65.9	66.2	66.8	66.2

5.2 ノードサイズについて

実験に用いた R-tree と MR-tree の葉ノードのサイズについて, 音データの例を以下に示す．ノードの格納数は 100 である．

- R-tree(葉ノード)
 - オブジェクト: 96bytes $\times$ 100
 - 射影データ: 24bytes $\times$ 100
 - オブジェクトインデックス: 4bytes $\times$ 100
- MR-tree(葉ノード)
 - オブジェクト: 96bytes $\times$ 100
 - 射影データ: 24bytes $\times$ 100
 - オブジェクトインデックス: 4bytes $\times$ 100
 - 中心点: 96bytes $\times$ 1
 - 中心点とオブジェクトの距離: 4bytes $\times$ 100

R-tree の葉ノード 1 つのサイズは 12400bytes, MR-tree は 12896bytes である．このように, MR-tree は R-tree に比べノードサイズが増加している．

5.3 R-tree と MR-tree

MR-tree の葉ノードでの実距離計算削減に用いる中心点に関しては, 画像データと音データ, どちらのデータに関しても, 質問点とオブジェクトの距離の縮みを緩和する MAXPJT で選択する中心点が最もよい結果となった．また, MAXPJT で構築した MR-tree に関して, 画像データでは従来の R-tree よりも大幅に距離計算回数を削減することができ, 検索の高速化へとつながった．しかし, 音データに関しては, MAXPJT で選択した中心点を付け加えても, 画像データほど距離計算回数を削減することはできず, 新しい中心点を追加することによるノードサイズの増加などから, 検索時間は R-tree よりも劣る結果となった．

6. 局所次元縮小射影による R-tree

R-tree の葉ノードでの S-Map による実距離計算の削減においては, 効果のある中心点がノードごとに違う可能性が高い．しかし, MR-tree のように各葉ノードに新しい中

心点を追加すると、ノードサイズの増加などから、実験 1 のように検索性能に悪影響を与えてしまう。そこで、各葉ノードに新しい中心点を追加することなく、葉ノードの局所を考慮することのできる、Local R-tree(LR-tree)を提案する。

LR-tree はまず、S-Map で取得する中心点候補を、R-tree の構築次元数より多く取得する。次に、葉ノードにおける実距離計算削減に用いる中心点を、中心点候補の中から、葉ノードごとに R-tree の構築次元数と同じ数選択する。R-tree の構築次元数と同じ数選択することで、ノードサイズの増加を抑えることができる。また、葉ノードごとに効果のある中心点を選択することで、それぞれの葉ノード内での距離の縮みを緩和することができ、より多くの実距離計算の削減が期待できる。

6.1 LR-tree の葉ノードにおける中心点選択法

LR-tree の葉ノードに用いる中心点の選択法として以下の三つを提案する。

MAXPJT : MAXimum ProJecTed distortion. MR-tree の時と同じ手法であるが、葉ノード内のオブジェクトではなく、S-Map の中心点候補で葉ノード内のオブジェクトを射影した際に、射影距離の総和の降順に中心点を選択する。

LONGMBR : LONGest side of MBR. 中心点候補でノード内のオブジェクトを射影した場合、射影データの最大値と最小値の差の降順に中心点を選択する。つまり、葉ノードの MBR の辺の長さの降順に中心点を選択する。

MAXVAR : MAXimum VARiance. 中心点候補でノード内のオブジェクトを射影した場合、分散の降順に中心点を選択する。分散の大きい軸を選ぶことで、実距離計算削減の効果が大きくなると期待できる。

7. 実験 2

それぞれの中心点選択法を用いて構築した LR-tree と R-tree の検索性能の比較を行う。実験 1 と同様の特徴データを使用し、LR-tree は、S-Map で取得した 16 個の中心点候補から葉ノードごとに中心点を選択し射影を行う。画像データについては 8 次元、音データについては 12 次元に射影を行い、それぞれの葉ノードに書き込む。

また、LR-tree の葉ノードのサイズについて、音データの例を以下に示す。ノードの格納数は 100 である。

- LR-tree(葉ノード)
 - オブジェクト : 96bytes × 100
 - 射影データ : 24bytes × 100
 - オブジェクトインデックス : 4bytes × 100
 - 中心点のインデックス : 1bytes × 12

LR-tree の葉ノード 1 つのサイズは 12412bytes である。5.2 節から MR-tree に比べノードサイズの増加を大きく抑えていることがわかる。

7.1 実験結果 2

7.1.1 画像データ

表 5 の結果から、近質問、準近質問、遠質問、すべての質問において LONGMBR が最も距離計算回数を削減できることが確認できた。LONGMBR を用いた LR-tree は全質問に対して平均で、R-tree に比べて距離計算回数を約 16%削減することができた。検索時間についても、LONGMBR が最も高速に検索を行うことができ、R-tree よりも平均で約 4.7%高速化することができた。

表 5 画像データ：各質問に対する平均距離計算回数 (回)

質問	距離計算回数			
	R-tree	MAXPJT	LONGMBR	MAXVAR
近	9.22×10^2	6.46×10^2	6.45×10^2	6.45×10^2
準近	1.46×10^5	9.67×10^4	9.62×10^4	9.67×10^4
遠	1.67×10^6	1.44×10^6	1.43×10^6	1.44×10^6

表 6 画像データ：平均検索時間 (ms)

	検索時間			
	R-tree	MAXPJT	LONGMBR	MAXVAR
全質問	60.4	58.9	57.6	58.4

7.1.2 音データ

表 7 の結果から、LR-tree については、三つの中心点選択法でほとんど差はなかったが、MAXVAR が最も距離計算回数を削減することができ、全質問に対して平均で R-tree よりも距離計算回数を約 4.3%ほど削減することができた。表 8 から検索時間についても、MAXVAR を用いた LR-tree が最も高速に検索を行うことができ、R-tree よりも約 3.9%高速に検索を行うことができた。

表 7 音データ：各質問に対する平均距離計算回数 (回)

質問	距離計算回数			
	R-tree	MAXPJT	LONGMBR	MAXVAR
近	1.75×10^4	1.56×10^4	1.57×10^4	1.56×10^4
準近	8.52×10^5	8.15×10^5	8.15×10^5	8.14×10^5
遠	1.34×10^6	1.29×10^6	1.29×10^6	1.29×10^6

表 8 音データ：平均検索時間 (ms)

	検索時間			
	R-tree	MAXPJT	LONGMBR	MAXVAR
全質問	65.9	63.3	63.3	63.3

7.2 中心点候補数

LR-tree では、中心点候補数を増やしても、階層的空間索引のサイズの増加を抑えつつ、LR-tree を構築することができる。また、中心点候補数は多いほど葉ノードごとによりよい中心点を選択できる可能性が上がり、より多く距離計算回数を削減できるはずである。そこで、最も効果のあった中心点選択法に関して、中心点候補数を増加させて、平均の距離計算回数の比較実験を行った。画像データに関しては LONGMBR、音データに関しては MAXVAR で実験を行う。実験の結果を表 9 に示す。

表 9 中心点候補数による距離計算回数の比較

LR-tree	中心点候補数		
	16	32	64
画像データ	5.54×10^5	5.82×10^5	5.89×10^5
音データ	5.48×10^5	5.66×10^5	6.10×10^5

表 9 からわかるように、どちらのデータにおいても、中心点候補数を増やしても距離計算回数は減少することなく、むしろ悪化してしまう結果となった。

7.3 R-tree と LR-tree

LR-tree の葉ノードにおける中心点選択法に関しては、画像データでは LONGMBR が最もよい中心点選択法となった。音データに関しては MAXVAR で選択する中心点 が最もよい結果となったが、三つの中心点選択法で差はほとんどなかった。また、LR-tree に関して、画像データでは LONGMBR を用いて LR-tree を構築することで、従来の R-tree よりも大幅に距離計算回数を削減することができ、検索の高速化へとつながった。音データについては、MAXVAR を用いて構築した LR-tree で、従来の R-tree よりも画像データほどではないが、距離計算回数を削減することができ、ノードサイズの増加を抑えつつ距離計算回数を削減することで、R-tree よりも高速な検索を実現することができた。しかし、それぞれの中心点選択法でほとんど性能に差がないため、音データについても、最も簡単に中心点を選択できる LONGMBR を用いることで十分であると考えられる。

中心点候補数を増加させることで、LR-tree でさらに多くの実距離計算回数を削減できると考えたが、どちらのデータにおいても、距離計算回数は増加してしまい、中心点候補数は 16 個が最もよい結果となった。これに関しては、S-Map で取得する中心点候補や、LR-tree の中心点選択法が最適でない可能性が考えられる。

8. まとめと今後の課題

本論文では、MR-tree の葉ノードにおける中心点選択法の提案と、MR-tree では高速化できなかった音データの改善を目標に、LR-tree の提案を行った。実験の結果、MR-tree の葉ノードにおける中心点選択法に関しては、画

像データと音データ、どちらのデータに対しても MAXPJT が最も効果のある中心点選択法であることがわかった。また、MR-tree では改善を行うことができなかった音データに関して、LR-tree を用いることで期待通り R-tree よりも高速な検索を実現することができた。音データにおいては、最も距離計算回数を削減できた MAXPJT を用いた MR-tree よりも、LR-tree を用いることで、距離計算回数を削減することができた。しかし、画像データについては、MAXPJT を用いた MR-tree ほど、LR-tree では距離計算回数を削減することはできなかったが、ノードサイズの増加を MR-tree よりも抑えることで、検索時間についてはほぼ同等の性能を発揮することができた。以上のことから、LR-tree は、画像データと音データのどちらにおいても有効な手法であるといえる。しかし、LR-tree において、中心点候補数を増やした場合、16 個の時よりも距離計算回数が増加してしまった。これは、最適な中心点選択が行えていないということである。

以上のことから、今後の課題として、LR-tree の中心点選択法の追究が挙げられる。また、MR-tree、LR-tree どちらにおいても、画像データの方が音データよりも各手法の効果が大きくあらわれた。この結果については、データ自体の散らばりの違いなどが影響していると予想されるが、詳しく追及する必要があると考えている。

参考文献

- [1] P. Ciaccia, M. Patella, P. Zezula: M-tree: An efficient access method for similarity search in metric spaces, VLBD '97 Proceedings of the 23rd International Conference on Very Large Data Bases, pp. 426–435, 1997.
- [2] A. Guttman: R-trees: A dynamic index structure for spatial seaching, in ACM SIGMOD, pp. 47–57, 1984.
- [3] T. Shinohara, H. Ishizaka: On dimension reduction mapping for approximate retrieval of multi-dimensional data, in Progress Discovery Science, pp. 224–231, 2002.
- [4] P. Zezula, G. Amato, V. Dohnal, M. Batko: Similarity Search, The Metric Space Approach, Secaucus, NJ, USA, Springer-Verlag, 2005.
- [5] 岩本 貴澄, 篠原 武: R-tree による高次元空間索引のための大域的および局所的次元縮小射影に関する研究, 九州工業大学大学院 修士論文, 2004.
- [6] 浦郷 祐希, 田島 圭, 青木 隆明, 岩崎 瑤平, 篠原 武: 空間索引による近似画像の高速検索を用いた動画同定システムの実現, 火の国情報シンポジウム 2009.
- [7] 岸川 直樹, 村上 義明, 田島 圭, 岩崎 瑤平, 篠原 武: 射影データを用いた Metric Rectangle tree(MR-tree) の提案, 人工知能学会 第 81 回 人工知能基本問題研究会, 2011.
- [8] 村上 義明, 岸川 直樹, 岩崎 瑤平, 田島 圭, 篠原 武: 空間索引構造を用いた多次元データ検索における次元縮小の有効性の検証, 火の国情報シンポジウム 2011.
- [9] 村上 義明, 岸川 直樹, 川口 尚志, 篠原 武: 空間索引構造 MR-tree を用いた多次元データの近似検索に関する研究, 第 65 回電気関係学会九州支部連合大会, 2012.
- [10] 山元 智大, 中西 義和, 鞆谷 拓真, 高木 俊和, 川久 保幸, 三浦 文彦, 篠原 武: 空間索引を用いた大量音楽データからの高速類似検索システムの実現について, 火の国情報シンポジウム 2004.