Hilbert R-treeにおける隣接要素間の距離を用いた検索高速化に関する研究

那須 洋平[†1,a)]　岸川 直樹[†2]　村上 義明[†2]　篠原 武[†2]

概要：画像や音楽などの多次元データの高速な近似検索を実現するために，一般的にR-treeなどの空間索引を用いる．R-treeの亜種であるHilbert R-treeは，マルチメディアデータ検索のための効率的な空間索引として知られている．本論文では，Hilbert R-treeの検索効率向上のために，葉ノード内の隣接する要素間の距離を用いた距離計算削減の手法を提案し，その有効性を検証する．実験より，画像データに対しては，平均距離計算回数を34%，平均検索時間を18%削減することが可能になったが，音データに対しては，いずれもわずかしか削減できなかった．

1. はじめに

近年，計算機の性能向上やインターネットの普及により，個人でも動画や音楽といった大量のマルチメディアデータを扱うようになってきた．そのため，大量のデータの中からユーザーが必要とするデータを取得する情報検索の技術の需要が高まっており，数多くの検索手法が提案されている．動画や音楽等のマルチメディアデータの検索では，あるデータと似ているデータを取得する近似検索の有用性が高い．それはマルチメディアデータはデータ圧縮などの処理により質の劣化がおきるため，完全に一致するものを検索する完全一致検索では，ユーザーが目的とするデータを取得できない場合があるからである．我々はマルチメディアデータから特徴を取り出し多次元データとみなしている．我々の研究の主題は，それら多次元データにおける近似検索の高速化である．

多次元データを高速に近似検索する手法として，階層的空間索引を用いて空間を索引付けする手法がある．代表的な階層的空間索引としてはR-tree [1] が知られており，その亜種となる空間索引も過去に様々なものが提案されている．それら亜種の中でも，Hilbert R-tree [2] は最も検索効率が良いことが確認されている．

Hilbert R-treeはHilbert曲線 [3] と呼ばれる空間充填曲線によって，多次元空間上の要素を一次元順序付けすることによって構成される．

本論文では，Hilbert曲線順で近いものは空間的にも近い可能性が高いという点に着目し，Hilbert R-treeにおける葉ノード内の隣接要素間の距離を用いた距離計算削減法を提案する．隣接要素間距離を用いた距離計算削減法は，質問点との距離計算を終えた要素が検索範囲外ならば，その次に続く要素に対しても検索範囲外である可能性が高いことを利用した距離計算削減法である．距離計算の削減の判定は，隣接要素間距離を用いた三角不等式で行う．

動画から切り出された画像データと楽曲から切り出された音データの二つのデータを用いて提案手法の検証を行った．その結果，画像データでは，隣接要素間距離による距離計算削減法は，距離計算回数を34%削減し，検索時間を18%削減した．しかし，音データに関しては，適応しなかった検索と比べて距離計算回数を1.5%，検索時間を1.2%しか削減できなかった．

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R-treeを説明し，4章でHilbert R-treeに関して説明する．そして5章で隣接要素間距離による距離計算削減法についてついて説明する．6章で実験について説明し，7章で実験結果と考察を述べる．そして8章でまとめと今後の課題を述べる．

2. 近似検索

2.1 距離空間

データ間に非近似度 (距離) を定義することで，質問点からの距離の順番で要素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．近似検索とは，質問点と近似する

†1 現在，九州工業大学院情報工学部知能情報工学科
Presently with Kyushu Institute of Technology Department of Artificial Intelligence
†2 現在，九州工業大学院情報工学府情報科学専攻
Presently with Kyushu Institute of Technology Department of Artificial Intelligence
a) i231061y@iizuka.isc.kyutech.ac.jp

データをデータベースから取り出すことである．

近似検索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 点間の要素間の非近似度の指標を示す距離関数を $d : \mathcal{U} \times \mathcal{U} \to \mathbb{R}^+$ とし，$\mathcal{D} = (\mathcal{S}, d)$ を距離空間とする．距離関数 d は距離の公理を満たすものとする．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．

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要素を x とする．x の特徴は n 組の実数 $(x_{(1)}, x_{(2)}, \ldots, x_{(n)})$ で表される．以下の二つの距離計測関数は距離の公理を満たすものである．

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

$$L_2\text{距離} : D(x,y) = \sqrt{\sum_{i=1}^{n} (x_{(i)} - y_{(i)})^2} \qquad (2)$$

本論文の実験では，距離計算を L_1距離 で計測する．

2.2 質問方法

近似検索は，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 以内の要素を取得する質問である．すなわち，

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

である．それに対して，質問点 Q から距離が最も小さい要素を取得する最近傍質問 $NN(\mathcal{D}, Q)$ がある．すなわち，

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

本実験では，上記二つのうちの最近傍質問を用いて近似検索を行う．最近傍質問では初めに検索範囲 r を無限大に初期化する．そこから検索をはじめて，検索範囲内の要素を暫定解として登録して行き，検索範囲をその要素と質問点との距離に収縮していきながら検索を行う．最近傍質問では要素が検索範囲内にあるかどうかが重要であり，検索範囲外の要素と質問点の距離を正確に求める必要はない．

3. R-tree

Guttman が提案した R-tree [1] は代表的な空間索引の一つであり，要素の追加・削除などの動的な操作を可能とする多岐平衡木である．R-tree は，対象とする空間の座標軸に平行な最小包囲矩形 MBR（Minimum Bounding Rectangle）で分割する．R-tree の内部ノードはその全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ち，葉ノードはデータベース内の要素へのポインタを持つ．

R-tree の検索は根ノードから始まり，深さ優先探索でノードを探索していく．検索が葉ノードのとき，その葉ノードが保持している要素との距離を計算し，検索範囲よりも小さければ検索範囲を更新する．検索が内部ノードのとき，質問点とその子ノードの MBR との距離を計算し，検索範囲と交差する MBR を持つノードのみ訪問する．この時，訪問する必要のあるノードは Active Branch List（ABL）と呼ばれる優先度付き待ち行列に挿入する．

ABL 内のノードは，質問点と MBR との距離の昇順でソートされる．ABL の先頭から順にノードを訪問することで，質問点との距離が小さい MBR を持つノードから訪問することが可能になる．このようにノードの訪問順を制御することで，質問点の検索範囲の収縮を早めることができる．

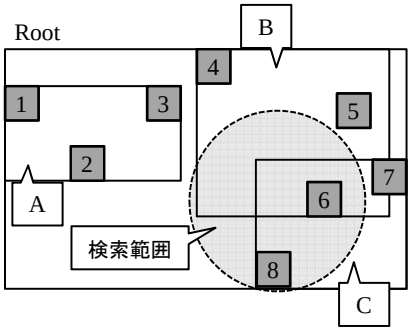

図 1 R-tree の構造
Fig. 1 Structure of R-tree

4. Hilbert R-tree

多次元空間内の全ての格子点をただ一度だけ通る曲線を，空間充填曲線と呼ぶ．多次元データを多次元空間上の点とみなし，空間内でこの曲線が通った順番に要素を順序付けすることで，多次元空間上の要素を一次元順序付けすることが可能となる．空間充填曲線の中でも Hilbert 曲線は，多次元データを空間的に近い順に一次元順序付けできることが知られている．Hilber R-tree は空間充填曲線である Hilbert 曲線を用いて構成されるものである．過去の研究で，R-tree を Hilbert 曲線を用いて構築した Hilbert R-tree は，R-tree に比べ検索効率が大幅に向上することが確認されている．

5. 葉ノード内隣接要素間距離を用いた距離計算削減法

本論文で提案する，葉ノード内隣接要素間距離を用いた距離計算削減法は，Hilbert 曲線順で近いものは空間的にも近い可能性が高く，隣接する要素同士はある程度近似しているという点を利用する．質問点との距離計算を終えた要素が検索範囲外ならば，その要素と Hilbert 曲線順に近

い要素に対しても検索範囲外である可能性が高いことに着目している．

検索が葉ノードに達したとき，通常では質問点と葉ノードが持つすべての要素と距離計算を行う必要がある．しかし，葉ノード内の要素には検索範囲外の要素も含まれている可能性がある．その場合，距離計算が必要のない無駄な要素まで距離計算を行われてしまう．そこで，本手法を用いて，そのような無駄な要素との距離計算を削減する．

まず，質問点との距離計算を行う必要がない要素について説明する．質問点を q，検索範囲を r_q，質問点との距離計算をすでに終えた要素を o_0 とし，葉ノード内のまだ距離計算行なっていない要素を o_n とする．三角不等式より以下の不等式が成り立つ．

$$d(q, o_n) \geq d(q, o_0) - d(o_0, o_n) \tag{3}$$

このとき，

$$d(q, o_0) - d(o_0, o_n) > r_q \tag{4}$$

が成り立つ場合は，式 3，4 より

$$d(q, o_n) > r_{q_i} \tag{5}$$

が成り立つ．この式 5 が成り立つ場合，要素 o_n は検索範囲 r_q に存在しないので，質問点 q と要素 o_n の距離計算は無駄なものとなる．図 2 に計算が省略できる例を示す．

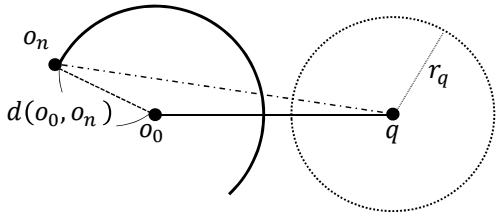

図 2 距離計算が省略できる例
Fig. 2 Example of reducible distance calculations

この計算が省略できる要素との距離計算を削減するために，本手法を用いる．本手法ではあらかじめ，Hilbert R-tree を構築する際に，Hilbert 曲線順により順序付けした葉ノード内の隣接する要素間の距離を求め，Hilbert R-tree に持たせる．葉ノード内の隣接する要素間の距離の総和によって省略可能な要素との距離を見積もり，削減可能か判断する．以下の条件式を基に要素 o_0 と要素 $o_n (n \geq 2)$ の距離を見積もる．要素 o_0，要素 o_n と o_0 から o_n までの隣接要素間距離の総和は以下の式が成り立つ．

$$d(o_0, o_n) \leq \sum_{i=1}^{n} d(o_{i-1}, o_i) \tag{6}$$

これより

$$d(q, o_0) - d(o_0, o_n) \geq d(q, o_0) - \sum_{i=1}^{n} d(o_{i-1}, o_i) \tag{7}$$

が成り立つ．よって

$$d(q, o_0) - \sum_{i=1}^{n} d(o_{i-1}, o_i) > r_q \tag{8}$$

が成り立つならば，必ず式 5 が成り立つので，質問点 q と要素 o_n との距離計算を行う必要がなくなる．$i = 2$ において距離計算が削減できる例を図 3 に示している．葉ノードで，質問点がある要素と距離を計算し終えると，その要素の後に続く要素に対し上記の条件式 8 を適用していく．この手法により，検索における距離計算回数の削減が期待できる．

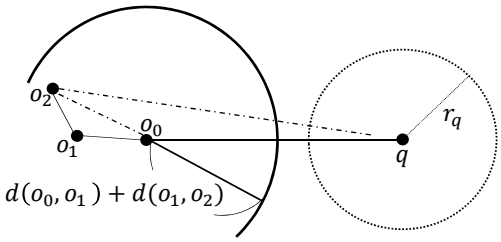

図 3 隣接要素間距離を用いた距離計算削減
Fig. 3 Reduction of distance calculation using distance between adjacent elements

6. 実験

実験に用いた PC の性能を表 1 に示す．

表 1 実験に用いた PC の性能
Table 1 The SPEC of the PC used in Experiment

CPU	Intel(R) Core(TM) i7 CPU 975 3.33GHz
メモリ	16GBytes

実験には画像データ [4] と音データ [5] が登録されてる 2 つのデータベースを使用する．画像データは約 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 次元に特徴抽出し登録したデータベースである．質問データとして約 100 本の動画から切り出した 9 万件の画像フレームを特徴抽出したものを用いる．

また，音データは約 1,500 曲の楽曲から切り出した約 700 万フレームを 96 次元に特徴抽出し登録したデータベースである．音データは，曲の頭からフレームの長さの 4 分の 1 ずつずらしながら，曲の終端まで切り出したものである．質問データとして約 30 曲の楽曲から切り出した 9 万フレームを特徴抽出したものを用いる．それぞれの質問データは，似たデータが見つかる近質問，やや似たデータが見つかる準近質問，似たデータが見つからない遠質問の各 3 万件で構成されている．質問方法は最近傍質問を用いる．

実験に用いる空間索引は，Hilbert R-tree である．Hilbert

R-tree に対し画像データと音データを用いて隣接要素間距離による距離計算削減法の効果を検証する．比較する項目は，質問に対する距離計算回数，検索時間の平均である

7. 実験結果と考察

7.1 実験結果

画像データの平均距離計算回数と検索時間を表 2 に，最近傍解との距離に対する平均距離計算回数と検索時間を図 4 と図 5 に，最近傍解との距離に対する距離計算回数と検索時間の削減率を図 6 示す．また，音データの平均距離計算回数と検索時間を表 3 に，最近傍解との距離に対する平均距離計算回数と検索時間を図 7 と図 8 に，最近傍解との距離に対する距離計算回数と検索時間の削減率を図 9 示す．

表 2 画像データに対する結果
Table 2 Results for image data

削減法	距離計算回数	検索時間 (ms)
適用無し	6.71×10^5	76.0
適用有り	4.44×10^5	62.1

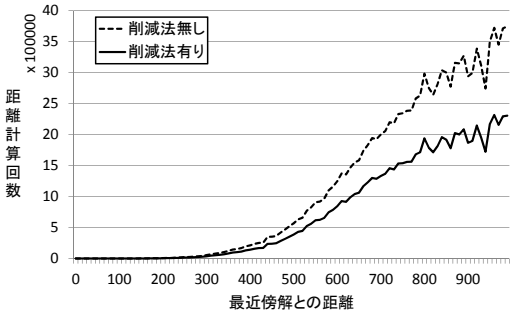

図 4 画像データに対する距離計算回数
Fig. 4 Number of distance calculations for image data

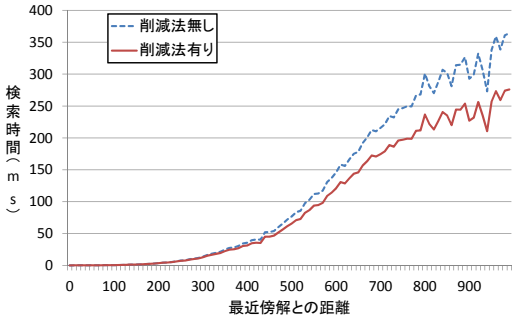

図 5 画像データに対する検索時間
Fig. 5 Search time for image data

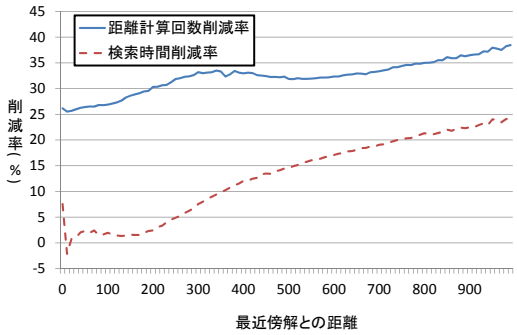

図 6 画像データに対する削減率
Fig. 6 Reduction ratio for image data

表 3 音データに対する結果
Table 3 Results for music data

削減法	距離計算回数	検索時間 (ms)
適用無し	5.69×10^5	116
適用有り	5.61×10^5	115

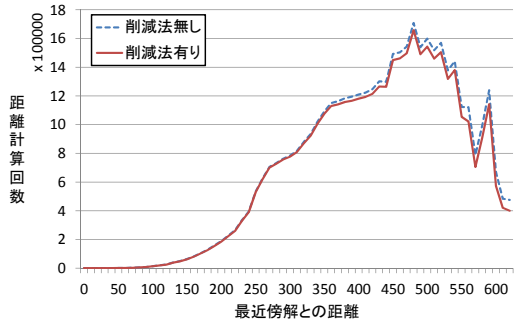

図 7 音データに対する距離計算時間
Fig. 7 Number of distance calculations for music data

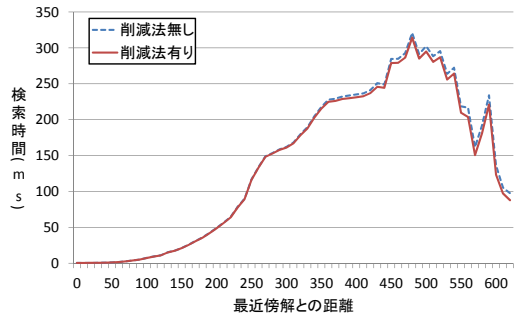

図 8 音データ対する検索時間
Fig. 8 Search time for music data

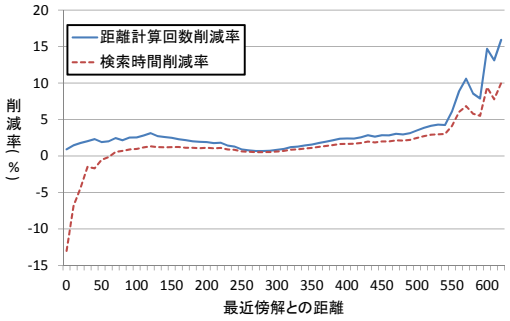

図 9 音データに対する削減率
Fig. 9 Reduction ratio for music data

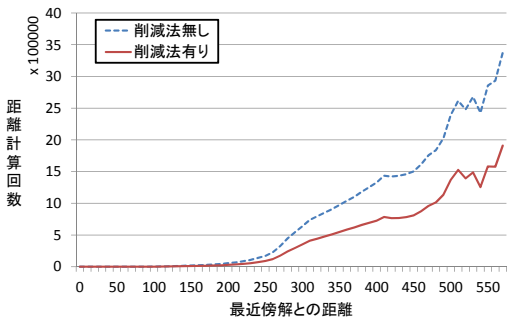

図 10 切り出し幅変更音データに対する距離計算回数
Fig. 10 Number of distance calculations for music data with shorter cut-out width

7.2 考察

表 2 と図 4，5，6 より，画像データに距離計算削減法を適用した場合，距離計算を 34%，検索時間を 18%を削減でき，削減法による効果がみられた．しかし，表 3 と図 7，8，9 より，音データの場合は距離計算を 1.5%，検索時間を 1.2%ほどしか削減できず，画像データに比べて大きな効果が得られなかった．隣接要素間距離による削減法は，要素間の距離が大きくなると式 8 が成り立ちにくく，音データは画像データに比べ，切り出すフレーム間の類似度が低く，葉ノード内の隣接要素間距離が大きくなるため，うまく削減法が適用できなかったと考えられる．

これを検証するために考察実験を行う．前の実験では，音データをフレームの長さの 4 分の 1 ずつずらしながら切り出した．この実験では，楽曲からの切り出し幅をフレームの長さの 128 分の 1 ずつずらしながら切り出した音データを用いて，比較実験を行い，検証した．平均の距離計算回数と検索時間を表 4 に，最近傍解との距離に対する平均の距離計算回数と検索時間を図 10 と図 11 に示す．

表 4 と図 10 と図 11 より，距離計算削減法を適応した検索は，適応しなかった検索と比べて距離計算回数が 45%，検索時間が 28%減少しており，フレームのずらし幅を小さくした場合，距離計算削減法の効果が出ていることが確認できた．以上のことより，音データに対して距離計算削減法が大きな効果を持たなかった原因は，音データのフレーム間の類似度が低く，葉ノード内の隣接要素間距離が大きくなるため，うまく削減法が適用できなかったと確認できる．

表 4 切り出し幅変更音データに対する結果
Table 4 Results for music data with shorter cut-out width

削減法	距離計算回数	検索時間 (ms)
適用無し	6.78×10^5	136
適用有り	3.75×10^5	97.6

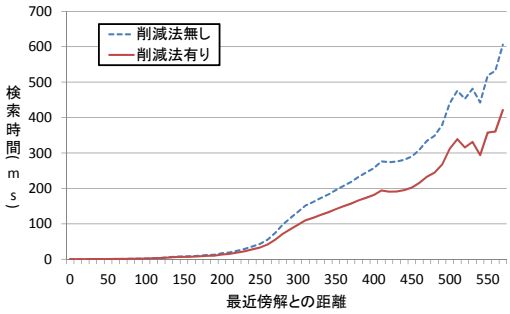

図 11 切り出し幅変更音データに対する検索時間
Fig. 11 Search time for music data with shorter cut-out width

8. まとめと今後の課題

Hilbert R-tree において葉ノード内の隣接要素間距離を用いた距離計算削減法を適応した場合，音データは大きな距離計算の削減には至らなかったが．画像データにおいては距離計算を大きく削減することができ検索時間を減少できた．しかし，画像データと音データの両方とも距離計算回数の減少に比べて検索時間がさほど減少しなかった点がある．これは，距離計算削減法を行う際に，隣接要素間距離を Hilbert R-tree に保持させるため，検索の際の I/O コストが大きくなることが原因と考えられる．また，削減法を適応する際に葉ノード内の全ての要素に対して判定を行うので，それに対するオーバーヘッドも原因とみられる．以上のことから，今後の課題として，図 6 と図 9 より，最近傍解との距離の近い質問に対しては距離計算回数，検索時間共に削減率が小さいという点に着目して，最近傍解の距離によって削減法の適応を判断するシステムの提案が考えられる．

参考文献

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

[2] I. Kamel, C. Faloutsos: Hilbert R-tree: An improved R-tree using fractals, Proc. 20th Int. Conf. on Very Large Data Bases, pp. 500–509, 1994.

[3] D. Hilbert: Uber die stetige Abbildung einer Linie auf ein Flachenstuck, Math. Ann. Vol. 38, pp. 459–460, 1891.

[4] 浦郷祐希，田島圭，青木隆明，岩崎瑶平，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 2009，2009.

[5] 篠原武，山元智大，中西義和，鞆谷拓真，高木俊和，川久保幸，三浦文彦: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 2004，2004.