階層的空間索引R-teeにおける ノード訪問順制御による検索高速化に関する研究

那須 洋平[†1,a)]　児玉 晋[†1]　篠原 武[†1]

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 の検索において，検索効率を上げるためには，ノードの訪問数を減らし，検索範囲を素早く収縮させるかが重要となる．ノード訪問順制御の従来の手法として，質問点と MBR の最短距離の順番でノードを訪問する手法があるが，ノード訪問の空振りにより，必ずしも最適な訪問順になるとは限らない．そこで本論文では，R-tree の検索効率向上のために，二つのノード訪問順制御の手法を提案し，その有効性を検証する．実験より，従来手法より，画像データに対して，平均ノード訪問数を 0.2%，平均検索時間を 2.1%削減し，音データに対しては，平均ノード訪問数を 0.4%，平均検索時間を 1.8%削減できた．

キーワード：近似検索，R-tree，ノード訪問順制御

Control of Node Visit Order in Hierarchical Spatial Indexe R-tree

Yohei Nasu[†1,a)]　Shin Kodama[†1]　Takeshi Shinohara[†1]

Abstract:
We usually use index structures such as R-tree for accelerating retrieval process of high dimensional multimedia data. In search of R-tree, in order to improve the search efficiency, it is important to reduce the number of visit nodes, and to shrink search range quickly. There is a method as conventional approaches, to visit the nodes in order of shortest distance of node and query points. But by wasted visit node, it is not optimal visit order necessarily. In this paper, to improve search efficiency of R-tree, we propose two methods of node visit order control, to verify the effectiveness of them. From experiments, we confirmed that the proposed method can reduce the number of visit nodes by 0.2% and search time by 2.1% in picture data. And music data, it can reduce the number of visit nodes by 0.4% and search time by 1.8%.

Keywords: Approximate search，R-tree，Control of node visit order

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．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

[†1] 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)] n673011y@ai.kyutech.ac.jp

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] やその亜種の Hilbert R-tree [2] が知られている．

本論文の構成は以下の通りである．2 章では近似検索，3 章では次元縮小を説明し，4 章で R-tree に関して説明する．そして 5 章で提案手法であるノード訪問順制御について説明する．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．近似検索とは，質問点と近似する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 $d : \mathcal{U} \times \mathcal{U} \to \mathbb{R}^+$ 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 [3] 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オブジェクトを 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．

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

$$L_\infty\text{距離} : D(x, y) = \max_{i=1}^{n} |x_{(i)} - y_{(i)}| \qquad (2)$$

本論文の実験では，オブジェクト間の距離計算は L_1距離で計測する．

2.2 範囲質問と近傍検索

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 以内のオブジェクト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オブジェクト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，検索範囲内のオブジェクトを暫定解として登録して行き，検索範囲をそのオブジェクトと質問点との距離に収縮して行きながら検索を行う．最近傍質問ではオブジェクトが検索範囲内にあるかどうかが重要であり，検索範囲外のオブジェクトと質問点の距離を正確に求める必要はない．

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．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) [4] 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

$$\varphi_p(O) = d(p, O) \qquad (3)$$

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．

$$d'(X, Y) = |\varphi_p(X) - \varphi_p(Y)| \leq d(X, Y) \qquad (4)$$

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 $P = \{p_1, p_2, \ldots, p_{n'}\}$ とすると，射影関数は

$$f_P(X) = (\varphi_{p_1}(X), \varphi_{p_2}(X), \ldots, \varphi_{p_{n'}}(X)) \qquad (5)$$

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，

$$d'(X, Y) = \max_{p \in P} |\varphi_p(X) - \varphi_p(Y)| \leq d(X, Y) \qquad (6)$$

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．中心点の数が多いほど，射影空間上で距離の情報を多く保存できる．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．

4. 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，オブジェクトの追加・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オブジェクトへのポインタを持つ．

R-tree の検索は根ノードから始まり，はじめは検索範囲を無限大にし，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オブジェクトとの距離を計算し，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 との最短距離 mindist [5] 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．このとき，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．

しかし，質問点との mindist が同じ MBR が二つ以上存在すると ABL に登録した順にノードを訪問するため無駄なコストが出てしまう可能性がある．

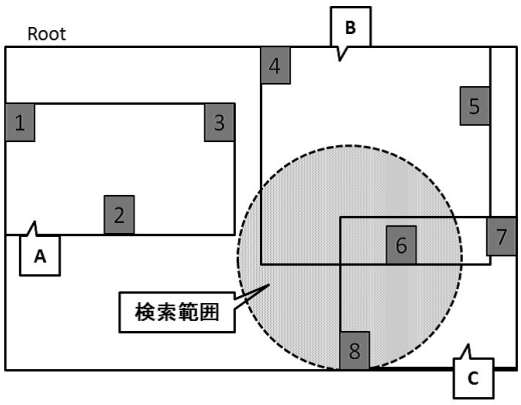

図 1 R-tree の構造
Fig. 1 Structure of R-tree

4.1 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オブジェクトを順序付けすることで，多次元空間上のオブジェクト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 [6] 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．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．今回の実験では，この Hilbert R-tree を用いる．

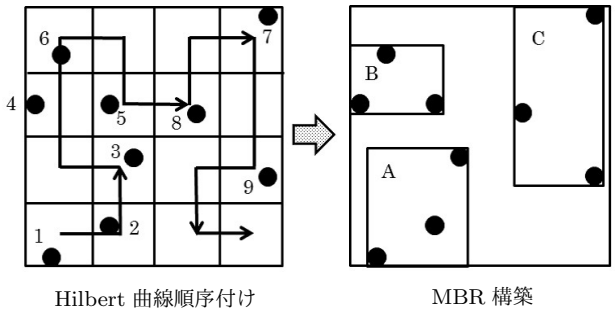

図 2 Hilbert R-tree の構築手順
Fig. 2 Construction of Hilbert R-tree

4.2 ノード訪問の空振り

R-tree を用いた検索時に無駄なコストとなり得る要因の一つに，図 3 に示されるようなオブジェクト分布が不均一な MBR の存在が挙げられる．ここで質問点を q，検索範囲を r_q とし，不均一な MBR を MBR A とする．r_q と MBR A は交差しているため，訪問し各オブジェクトとの距離を測らなければならないが，検索範囲 r_q 内にオブジェクトが存在せず，検索範囲が縮まらず，MBR A への訪問は無駄な訪問となる．結果として検索の際に無駄なコストが生じてしまう可能性がある．また，図 4 のように mindist が同じ値をとる MBR が存在する場合，ABL に登録した順にノードを訪問するため，仮に MBR A から ABL に登録すると A からノードを訪問してしまう．このような無駄なコストの増加を減らすために，本論文では二つのノード訪問順制御の手法を示す．

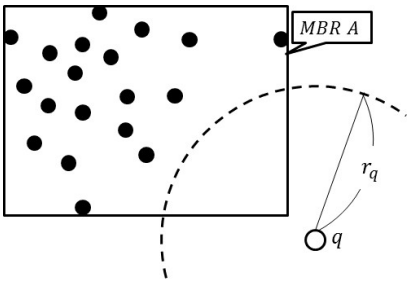

図 3 オブジェクトが不均一な MBR
Fig. 3 MBR with nonuiniform objects

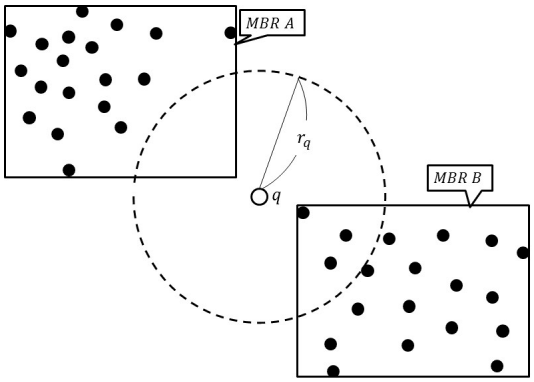

図 4　mindist が同じ値を取る場合
Fig. 4　Two MBRs with same mindist

5. ノード訪問順制御

ここでは，本論文で提案するノードの訪問順制御について示す．しかし，先述した質問点と MBR の最短距離 mindist も R-tree の検索においては重要である．そこで，基本的なノード訪問の順番は mindist で行い，mindist が同じ場合は以下に示す手法で求める距離順で訪問するノードを決定する．

5.1 MBR の重心を用いたノード訪問順制御

まず，MBR center dist 法を提案する．この手法は，R-tree に MBR の重心の座標を新たに持たせ，検索時に質問点から MBR の重心点までの距離 *cdist* を用いてノード訪問順を制御する手法である．質問点から MBR の重心までの距離が大きいほど，検索の空振りが起きやすく，逆に質問点から MBR の重心点までの距離が小さい物ほど，検索の空振りが起きにくいと考えられるので，ノードの訪問順は質問点と MBR の重心までの距離が小さい順に訪問する．従来手法である，mindist 法に比べて，オブジェクトの分布が質問点と近い MBR から訪問し，よりコンパクトな MBR から訪問する期待値が高くなるので，検索の高速化が期待できる．

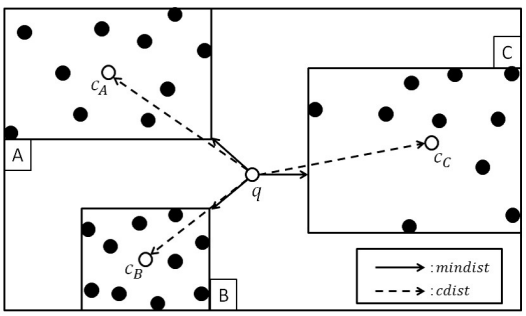

図 5　MBR の重心を用いたノード訪問順制御
Fig. 5　Node visit order control using MBR centroid

5.2 SDR を用いたノード訪問順制御

次に Standard deviation rectangle（SDR）を提案する．SDR は MBR に含まれるオブジェクトの座標の散らばり具合を元に作成される．具体的には各オブジェクトの座標値の平均を m，標準偏差を σ とするとき，平均値 m に $\pm k\sigma$ を加えたもので SDR の各辺の大きさを決める．2 次元上の SDR の大きさを決めるときの例を図 6 に示す．

SDR は，MBR 内のオブジェクトの平均と標準偏差から作られるので，MBR 内のオブジェクトが均一ならば図 7 の左の様に SDR の形は元の MBR の形に近くなり，逆に不均一なときは図 7 の右の様に MBR の形から離れた物となると考えられる．このように SDR を作成し，検索時に質問点から SDR までの距離 sdist を使用することで，検索の空振りを減らすことが期待できる．

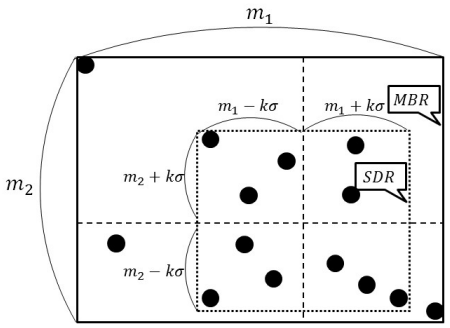

図 6　2 次元上の SDR
Fig. 6　SDR on two-dimensional

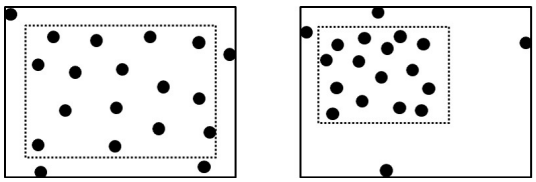

図 7　SDR の例
Fig. 7　Examples of SDR

6. 実験

それぞれのノード訪問順制御手法の検索性能の比較を行う．実験には画像データ [7] と音データ [8] が登録されている二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．質問データとして約 100 本の動画から切り出した画像フレームを特徴抽出したものを用いる．

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．音データの

フレームの切り出し幅は，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，曲の終端まで行う．質問データとして約 30 曲の楽曲から切り出した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，データベース内に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 万件の計 9 万件で構成されている．質問方法は最近傍質問を行う．

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，今回の提案手法の効果を検証する．比較する項目は，1 つの質問に対するノード訪問数と距離計算回数，検索時間の平均である．

実験に用いた PC の性能を以下に示す．

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) i7-3770 CPU @ 3.40GHz
メモリ	16GBytes

7. 実験結果と考察

7.1 実験結果

画像データの平均の訪問ノード数，距離計算回数，検索時間を表 2 に示し，音データの平均の訪問ノード数，距離計算回数，検索時間を表 3 に示す．

7.2 考察

実験結果より，画像データにおいて mindist に cdist を追加したものは，近，準近，遠質問のノード訪問数，検索時間共に mindist のみに比べ微々たるものであるが減少している．ノード訪問数は，近質問 0.9%，準近質問 0.3%，遠質問 0.1%，検索時間は，近質問 1.6%，準近質問 2.3%，遠質問 2.21%削減できた．sdist に関しては，mindist のみのものに比べて cdist ほどではないが，訪問ノード数，検索時間を削減し，距離計算回数を近質問 0.5%，準近質問 0.02%，遠質問 0.002%削減と cdist よりも削減する結果となった．音データにおいては，画像データと似た結果となり，cdist を用いたものが，ノード訪問数が近質問 0.7%，準近質問 0.5%，遠質問 0.3%削減し，sdist を用いたものが距離計算回数を近質問 0.1%，準近質問 0.01%，遠質問 0.004%削減，検索時間は近質問 1.5%，準近質問 1.3%，遠質問 1.3%削減することができた．sdist よりも cdist の方がノード訪問数を削減できた理由としては，今回の実験では Hilbert R-tree を用いているため，特に画像データでは Hilbert R-tree を構築する際に Hilbert 曲線順に並べて近いオブジェクトで MBR を構築するため，MBR 内のオブジェクトの分布が極端に不均一になるものが少なく，動画から 1 フレームずつ切り出しているため，互いに似たデータが存在するため，SDR を用いるメリットが少なかったことが考えられる．

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

画像データ	質問	平均ノード訪問回数	平均距離計算回数	平均検索時間 [msec]
mindist のみ	全	13259	770940	63.98
	近	572	8887	1.83
	準	8978	410151	38.21
	遠	31307	1959265	157.22
mindist 優先 cdist 順	全	**13239**	**770872**	**62.64**
	近	**567**	8843	**1.80**
	準	**8951**	**410062**	**37.35**
	遠	**31275**	**1959192**	**153.88**
mindist 優先 sdist 順	全	13251	**770872**	63.58
	近	570	**8798**	1.81
	準	8967	410083	37.97
	遠	31294	1959219	156.09

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

画像データ	質問	平均ノード訪問回数	平均距離計算回数	平均検索時間 [msec]
mindist のみ	全	9864	573039	62.63
	近	1959	104024	11.63
	準	12974	744788	81.31
	遠	14438	858401	92.76
mindist 優先 cdist 順	全	**9823**	573019	61.64
	近	**1945**	104001	**11.45**
	準	**12913**	744738	80.42
	遠	**14392**	**858348**	93.09
mindist 優先 sdist 順	全	9851	**572961**	**61.52**
	近	1955	**103898**	11.46
	準	12954	**744714**	**80.29**
	遠	14423	858364	**91.51**

8. まとめと今後の課題

本論文では，R-tree におけるノードの訪問順の新しい制御方法を提案した．今回提案した，cdist，sdist を用いたノード訪問順制御は実験より効果的ではあるが，大きな改善とはならなかった．これは，mindist の値が同じになるような MBR がさほど多くないため大きな改善に至らなかったものと思われる．また，今回の実験に用いたデータは動画から切り出した画像データと，楽曲から切り出された音データを扱っており，R-tree を構築する際に Hilbert 曲線順に並べて近いオブジェクトで MBR を構築するため，MBR 内のオブジェクトの分布が極端に不均一になるものが少なく，SDR を用いるメリットが少なかったと考える．SDR の効果を検証する際には，もっと MBR 内のオブジェクトがばらけるようなデータを用いて検証する必要があると考える．

今後の課題としては，ノード訪問順制御のさらなる最適化が挙げられる．今回は SDR 作成時に MBR 内のオブジェクトの各次元の標準偏差を用いたが，この標準偏差を用いて，質問点から MBR の重心までの距離にデータがどのくらい存在しているかを測ることができれば，検索範囲内にデータが多いノード順に訪問することができるため，検索範囲の縮小を早め，さらにノード訪問回数を減らすことができるのではないかと期待される．

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] P. Zezula, G. Amato, V. Dohnal, M. Batko: Similarity Search, The Metric Space Approach, Secaucus, NJ, USA, Springer–Verlag, 2005.

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

[5] N. Roussopoulos, S. Kelley, F. Vincent: Nearest Neighbor Queries, Proc. ACM-SIGMOD Intl. Conf. on Management of Data, 1995.

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

[7] 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.

[8] 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.