

縮小型構造データ Sketch を用いた空間検索法に関する研究 ～GHP を用いた Sketch 作成関数のためのピボット選択法～

岩崎 瑤平 Arnoldo Jose Muller-Molina 篠原 武
(九州工業大学大学院情報工学府情報科学専攻知能情報分野)

1 はじめに

近年、空間索引に代わる近似検索の手法として Sketch[2,4] を用いた検索法が注目を浴びている。Sketch は、オブジェクトの類似性のある程度保持したまま、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り、ある種の Locality Sensitive Hashing (LSH) である。本論文で用いる Sketch には、基礎分割関数として一般化超平面分割 (GHP)[3] を用いる。GHP は点对を用いて定義され、どちらの点に近いかによって空間を二つの部分空間に分割する手法である。Sketch ではオブジェクト間の距離関係を完全に保持しているわけではないため、検索結果にある程度の誤りが存在する。そのため、より高精度の Sketch を実現するためには点对の選択が重要である。本論文で提案する手法を用いることで、多次元索引構造である R-tree[1] を用いた場合とほぼ同じ検索速度を維持したまま、約 99% の検索精度を実現できる。

2 縮小型構造データ

2.1 一般化超平面分割 (GHP)

GHP は 1 組の点对を用いた空間分割法であり、任意の距離空間に適用可能である。任意の距離空間において、集合 $X \in \mathcal{D}$ と 1 組のピボット $(p_0, p_1) \in \mathcal{D}^2$ において、GHP による分割は以下のように定義される。

$$S_{p_0} = \{o \in X | d(p_0, o) \leq d(p_1, o)\}$$

$$S_{p_1} = \{o \in X | d(p_0, o) > d(p_1, o)\}$$

ここで S_{p_0}, S_{p_1} は GHP によって分割された集合 X の部分空間を示す。

2.2 Sketch

m 組の点对を用いて、GHP を m 回適用することによって、長さ m の Sketch を作成する。GHP 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、その各ビット $\sigma_i(x)$ を以下のように定義する。

$$\sigma_i(x) = \begin{cases} 0, & \text{if } d(p_{0i}, x) \leq d(p_{1i}, x) \\ 1, & \text{if } d(p_{0i}, x) > d(p_{1i}, x) \end{cases}$$

また、任意の 2 点 x, y 間における Sketch の距離を次のように定義する。

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

これは、Sketch におけるハミング距離であり、Sketch がバイナリ文字列で表現されたものであり、ビット演算のみで計算が行えるため、高速な演算が可能である。

2.3 検索法

Sketch を用いた検索では、以下の手順で検索を行う。

1. Sketch データベースに対する K ($K \geq k \geq 1$) 近傍質問
2. K 個の解から実空間距離に基づき k 個の解を返す

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 近傍質問を行う。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。

3 点对の評価法

GHP を用いた空間分割を行った場合、分割後の部分空間同士のバランスは保証されない。そのため、分割に用いる点对の選択は Sketch を用いた検索の性能を上げるために非常に重要である。本章では、4 種類の点对評価法について提案を行う。点对の高速な検索のために、データベース内の M 個のサンプルに対し、点对の評価を用いる。

◇ バランス法

サンプルを用いて試験的に Sketch を作成し、各ビット毎のバランスを調べることで評価を行う。バランスが取れるように分割することで、Sketch の各 bit が持つ情報量を多くすることが可能である。

◇ 距離法

分割に用いる点对同士の距離を用いて評価を行う。具体的には、分割に用いる点对同士の距離を小さくなるものを良い点对であるとする。これによって、多方向の軸を用いて空間を分割することが可能なため、Sketch における距離の伸びを小さくできると考える。

◇ 最小衝突法

試験的に Sketch を作成し、作成した Sketch の衝突の個数を用いて評価を行う。具体的には、Sketch の衝突が少なくなるものを良い点对であるとする。衝突が少なくなるように点对を作成することで、オブジェクトの持つ情報量の多くを保持することが可能である。

◇ 複合法

距離法と最小衝突法を複合した評価法である。点对間の距離と衝突の個数を考慮することで、オブジェクトの持つ情報量の多くを保持したまま、距離の伸びの小さい Sketch を作成できると考える。

4 実験

データベースとして約 2,800 本の動画から切り出した、約 700 万件の画像フレームデータ [5] を登録し、提案した評価法の有効性について検証を行う。質問方法には、ある一定範囲内で最も距離が小さいオブジェクトを解とする範囲限定最近傍質問を用いる。質問用のデータとして、約 25000 件の画像フレームデータを用いる。実験では、Sketch データベースに対する K 近傍質問における K の値を 700, 7000, 70000 の 3 種類に設定し、検索精度と検索速度について検証を行った。以下、全ての図において、(1) がバランス法、(2) が距離法、(3) が最小衝突法、(4) が複合法、(5) がデータベースからランダムに点対を選択するランダム法の実験結果を示す。

4.1 検索精度

3 節で紹介した 4 つの点対評価法を用いた場合の性能を 64bit の Sketch を用いて比較したものを図 1 に示す。

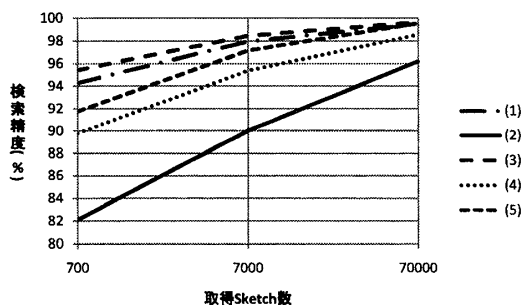


図 1: 64bit-Sketch の検索精度のグラフ

図 1 から、最小衝突法が最も検索精度が良いことが分かる。

4.2 検索時間

Sketch の検索時間として、4.1 節の検証で精度の良かった、バランス法、最小衝突法、ランダム法を用いて、多次元索引構造である R-tree と比較を行った。R-tree は高次元空間において、検索効率が悪化してしまうため、次元縮小法を用いて特徴空間を低次元空間に射影し、索引付けを行ったものを使用している。図 2 は、各評価法を用いて作成した SKetch と、R-tree との検索時間の比較を行ったグラフである。

評価尺度には、 IE を用いる。 IE は検索コストが評価対象と比較して、どの程度改善されているかを示す指標である。本論文では、検索コストとして検索時間を採用し、 IE は検索時間が何倍高速化されたかを示す。

図 2 から、 $K = 70000$ に設定した場合でも、R-tree とほぼ同程度の検索速度であることが分かる。

5 まとめ

本論文では、Sketch の基礎分割関数である GHP に用いる点対の評価法について提案を行い、その妥当性について評価を行った。GHP の点対評価関数として、

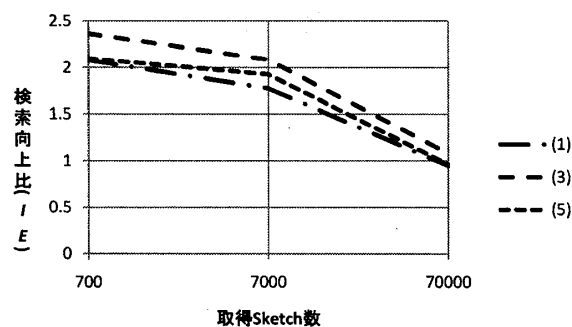


図 2: 64bit-Sketch と R-tree の比較

Sketch 作成時の衝突を考慮した評価関数を用い、 $K = 70000$ (全データの約 100 分の 1) に設定することで、約 99% の検索精度を維持することが可能であった。これは、Sketch での衝突が少ない点対を用いることで、実空間上のオブジェクトが持つ情報の多くを保持したまま、Sketch を作成することができたからであると考えられる。検索時間に関しても、Sketch の衝突を考慮した評価関数を用いることで、99% 以上の検索精度を維持したまま、R-tree とほぼ同じ検索速度を実現できた。しかし、索引構造を用いた検索では、質問データに近いオブジェクトがデータベース内に存在する場合 (近質問) は非常に高速に検索でき、逆に、近いオブジェクトがデータベース内に存在しない場合 (遠質問) は検索効率が悪化するという性質がある。近質問によって Sketch と R-tree を用いた検索の比較を行った場合、R-tree の方が約 1.5~5 倍高速であることが分かった。逆に、遠質問で比較を行った場合、Sketch の方が約 3 倍高速であることがわかった。

今後の課題として、GHP 以外の基礎分割関数を用いた Sketch 作成関数に関する研究が挙げられる。また、今回適応した動画の画像フレームデータ以外のマルチメディアデータに対して検証を行うことも今後の課題である。

参考文献

- [1] Guttman, A.: R-trees: A Dynamic index structure for spatial searching, Proc. ACM SIGMOD, pp.47-57, (1984).
- [2] Arnoldo Jose Muller-Molina, Takeshi Shinohara: Efficient similarity search by reducing I/O with compressed sketches In SISAP. IEEE, (2009).
- [3] Pavel Zezula, Giuseppe Amato, Vlastislav Dohnal, and Michal Batko: Similarity Search: The Metric Space Approach. Springer-Verlag, Secaucus, NJ, USA, (2005).
- [4] Qin Lv, Moses Charikar, and Kai Li: Image similarity search with compact data structures. In CIKM '04, pp.208-217, New York, NY, USA, (2004).
- [5] 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, 火の国シンポジウム (2009)