高速･高精度な近似最近傍探索の実現

Realization of Fast and Accurate Approximate Nearest Neighbor Search

岩村 雅一（ Masakazu IWAMURA, Ph.D ）
大阪府立大学大学院工学研究科
知能情報工学分野 准教授
(Deptment of Computer Science and Intelligent Systems, Graduate School of Engineering at Osaka Prefecture University, Associate Professor)

電子情報通信学会、情報処理学会、IEEE、ACM 会員

受賞 : 画像の認識・理解シンポジウム インタラクティブセッション優秀賞 (MIRU2005), 画像の認識・理解シンポジウム デモセッション優秀賞 (MIRU2006), 電子情報通信学会 論文賞（第 63 回（平成 2006 年度）), 画像の認識・理解シンポジウム デモセッション賞 (MIRU2007), IAPR/ICDAR Best Paper Award (ICDAR2007), 第 1 回データ工学と情報マネジメントに関するフォーラム最優秀インタラクティブ賞(DEIM2009), IAPR DAS Nakano Award (Best Paper Award) (DAS2010), ICFHR Best Paper Award (ICFHR2010), IAPR/ICDAR Young Investigator Award (ICDAR2011), 大阪府立大学 学長顕彰 (2006, 2007, 2008, 2009, 2011, 2012 年度)

研究専門分野 : 統計的パターン認識, カメラベース文書画像検索, カメラベース文字認識, 物体認識, 近似最近傍探索

あらまし コンピュータや携帯型デバイスの普及と性能向上、インターネットの普及などにより、我々が利用できるテキスト、画像、音楽、動画などのデータは日々増加している。これらの情報は、うまく活用できれば我々の生活を豊かにすると考えられるが、そのためには膨大な情報の中から所望の情報を効率よく発見できる技術が必要不可欠である。本稿では、情報処理において基本的な処理である「最も似ているデータをみつける」最近傍探索と呼ばれる技術に注目する。最近傍探索に近似を導入した近似最近傍探索は、大規模データに対する探索精度と計算時間において望ましい性質を持っているため、大量のデータを活用する方法として近年よく用いられている。これらの最近の手法を紹介した後、我々が提案している現在世界で最も高速な近似最近傍探索手法を紹介する。

１．はじめに

計算機の発達により、我々はかつて無いほど手軽にテキスト、画像、音楽、動画といった様々な形式のデータを作成できるようになった。また、それらをインターネットの共有サイト(例えば､画像であれば Flickr (http://www.flickr.com/)、動画であれば YouTube (http://www.youtube.com/) やニコニコ動画 (http://www.nicovideo.jp/)など）を通じて共有できるようになった。画像共有サイトの Flickr には 2011 年までに 60 億枚の画像がアップロードされており、1 年間に数千万枚以上の画像がアップロードされて続けている。仮に 1 枚の画像を見るのに 1 秒かかるとすれば、60 億枚の画像を見るにはおよそ 200 年かかる計算になる。これは画像だけに限った話ではなく、動画や音楽においても同様である。したがって、膨大な情報から必要なものを取り出す技術が必要とされている。

大量データを扱う際の課題は処理速度と精度である。通常の方法ではデータが増えれば増えるほど計算時間が増えるので、大量データの全てを処理しない方法が求められる。また、データが増えれば「他人のそら似」的な類似データの存在確率が増加する。それらをいかに区別するかが重要になる。

本稿では、この問題を解決する基本技術とその応用について述べる。基本技術は近似最近傍探索と呼ばれ、情報処理において基本的な「最も似ているデータをみつける」処理を実現する。

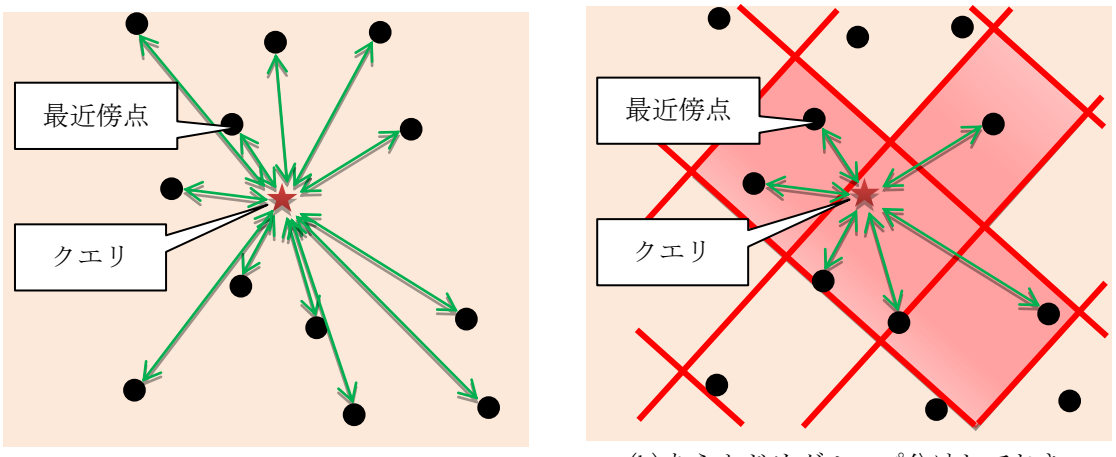
２．最近傍探索と近似最近傍探索

最近傍探索とは、検索質問データ（クエリ）が与えられたとき、それに最も近いデータ（最近傍点）をデータベースから探し出す問題である。単純であるが故に応用範囲が広く、基本的かつ重要な問題である。データとしては、一般にベクトルデータを想定する。具体的な応用例としては、図 1 に示すように、物体認識[1]、文書画像の検索[2]、カメラを用いた文字認識[3]、顔認識[4]などがあり、いずれも大規模なデータに対して高速で高精度な検索・認識が実現可能である。この問題の最も単純な実現方法は、図 2(a)に示すようなクエリとデータベース中の全てのデータとの距離を計算して、最も近いデータを探すものである。

高速・高精度な近似最近傍探索の実現

Realization of Fast and Accurate Approximate Nearest Neighbor Search

図１　最近棟探索の応用



(a) 全てのデータとの距離を計算する単純な方法

(b) あらかじめグループ分けしておき、近いグループのデータとのみ距離計算する方法

図２　最近棟探索の実現方法

高速・高精度な近似最近傍探索の実現

Realization of Fast and Accurate Approximate Nearest Neighbor Search

この問題は時間さえ掛ければ必ず正しい答えが見つかる反面、データ数が多く（例えば、数千万、数億のオーダーに）なると膨大な計算時間を必要とする。そのため、高速化のために様々な工夫が提案されてきた。

最近傍探索を高速化するためには、図 2(b)に示すようにクエリが与えられる前にあらかじめデータを構造化しておくことが有効である。構造化とは、データをあらかじめいくつかのグループに分けておくことである。探索の際には、クエリと近いグループをいくつか選び出して、それらに属するデータのみを最近傍点の候補としてクエリとの距離計算に用いる。これにより、クエリとデータの距離を計算する前に最近傍点の可能性が高いデータのみを選択することができ、計算時間を大幅に削減できる。この仕組みは特にデータ数が多いときには効果を発揮する。

このようにデータの構造化と選択を導入すると、以下のように検索処理は大きく 2 段階に分けられる。

(1) クエリに近いデータのグループを見つける。

(2) 選ばれたグループに属しているデータとクエリとの距離を計算し、一番クエリに近いデータを選択する。

これらのうち、処理(2)は単にクエリとデータの距離を計算するだけなので、特段工夫の余地はない。近似最近傍探索の性能の善し悪しを決めるのは処理(1)であり、いかにクエリに近いデータを選択し、クエリから遠い距離を選択しないかが課題となる。

注意しなければいけないのは、最近傍探索では処理(1)でクエリに近いグループを選択する際には、真の最近傍点が含まれるグループが必ず選ばれるように工夫しておかなければならないことである。もしそのグループが距離計算の対象として選ばれなければ、真の最近傍点は距離計算に用いられないので、真の最近傍点が正しく探索されることはない。具体的な工夫としては、少しでも真の最近傍点が含まれる可能性のあるグループは距離計算の対象とすることが考えられる。しかし、この工夫を施せば、折角削減できた計算時間を多少なりとも再び増加させることになる。特にデータの次元数が高いときには、データの構造化をしたために却って全探索（クエリとデータベース中の全てのデータとの距離を計算する場合）に比べてよりも計算量が増加してしまう場合がある。

そのため、計算時間の削減を目的として近年主流になっているのは、「真の最近傍点が必ず得られる」という最近傍探索の制約を緩和することである。このような問題を近似最近傍探索と呼ぶ。近似最近傍探索では、必ずしも距離計算対象として選ばれるグループに、真の最近傍点が必ずしも含まれる必要は無い。そのため、最近傍探索に比べると距離計算対象を選択する前述の処理(1)の処理が簡潔になり、さらに結果的に処理(2)の処理の計算時間も削減される。近似最近傍探索は、最近傍点が正しく求まらない場合があるため、応用を選ぶものの、計算量の削減という意味では魅力的である。

近似最近傍探索では一般に、図 3 に示すように近似の強さを強くするにつれて、計算時間は削減できるが、探索精度も低下する。最近傍探索では真の最近傍点が必ず（100%の確率で）探索されるため、探索に要する計算時間が手法の善し悪しを決める評価尺度として用いられる*1。近似最近傍探索では最近傍点が求まらない場合があるため、検索精度も重要な評価指標となる。実際には、アプリケーションによって必要な検索精度、許容できる計算時間が決まることが多いため、これらのバランスをいかに追求するかが重要となる。

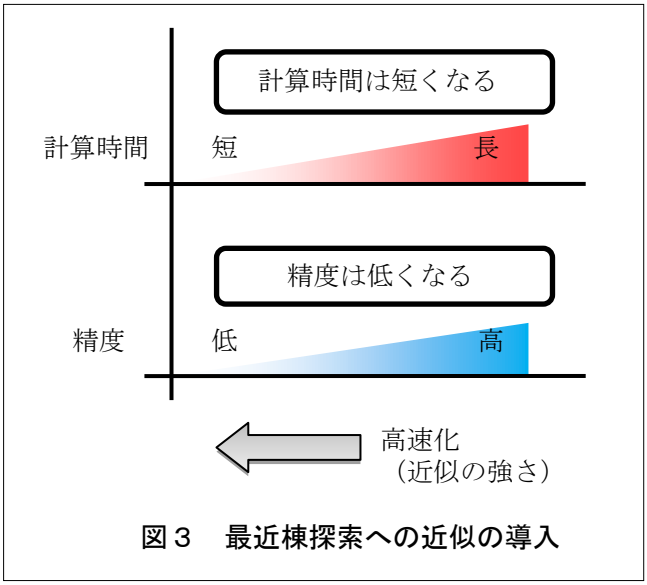


図３　最近棟探索への近似の導入

*1 探索に要する計算時間の他にも、学習に要する計算時間やメモリ使用量が評価尺度として考えられる。

３．代表的な近似最近傍探索手法

近似最近傍探索の代表的な既存手法を簡単に紹介する。

既存の近似最近傍探索の手法は、木構造を用いるものとハッシュ構造を用いるものに分けることができる。木構造を用いるものには、ANN[5]、Randomized kd-tree (RKD)[6]、階層的 k-means (HKM) [7]に加えて、RKD と HKM を組み合わせた FLANN[8]などがある。ハッシュ構造を用いるものには、Locality Sensitive Hashing (LSH)[9]、Spectral Hashing (SH) [10]、Inverted File with Asymmetric Distance Calculation (IVFADC)[11]、Inverted Multi-Index (IMI)[12]などがある。これらを実際のデータで試してみた結果、Randomized kd-tree、階層的 k-means、IVFADC、IMI が同じ検索精度を短時間で実現することができた。以下で述べる我々の提案手法はハッシュ構造を用いる手法であるため、これらの中からハッシュ構造を用いる手法を概観する。

図 4 は IVFADC における距離計算対象の選択方法（前述の処理(1)）を図示したものである。近似最近傍探索の準備として、IVFADC はデータをクラスタリングして、データをあらかじめ複数の代表点で表す。そして、クエリが与えられたとき、クエリに近い代表点を幾つか選択し、その代表点に属する点を距離計算の候補とする。IVFADC で用いるクラスタリング手法はベクトル量子化と呼ばれ、クラスタ数が同じであれば、データを代表点で表現することによって生じる誤差が最小であることが知られている。この誤差が大きければ、距離計算対象の選択において真の最近傍点を発見できず、探索精度の低下に繋がる。したがって、誤差が最小であるベクトル量子化を使用している IVFADC はメモリ効率という意味で優れていると考えられる。IVFADC を提案した文献[11]では、さらにデータの省メモリな表現方法を合わせて提案しているが、これについて本稿では言及しない。

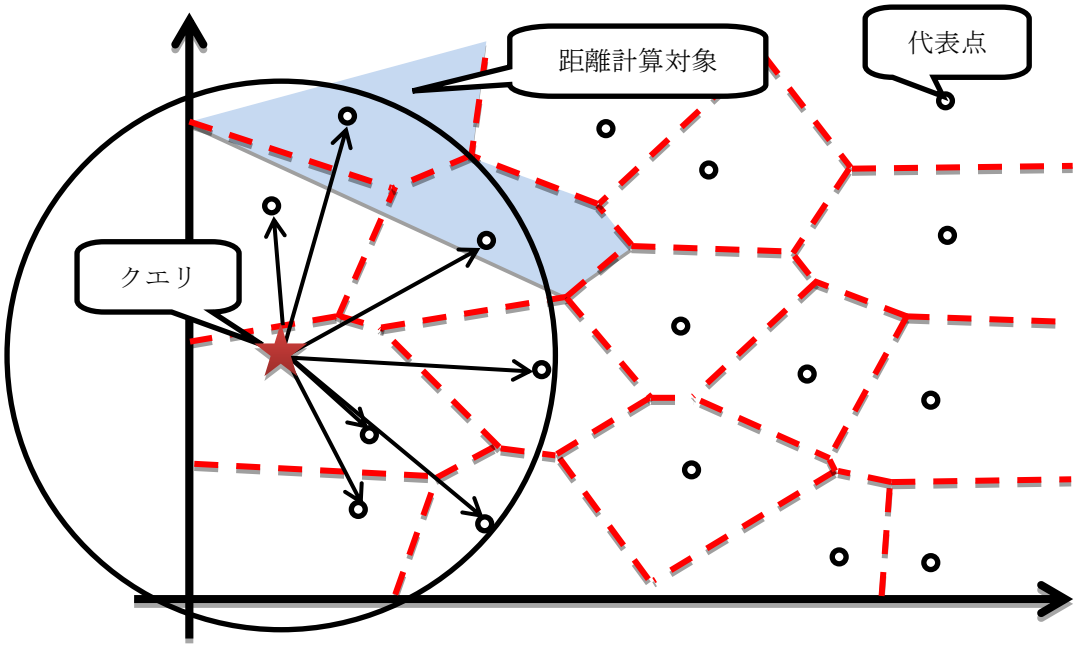


図４　従来手法 Inverted File with Asymmetric Distance (IVFADC)における距離計算対象の選択

IVFADC はメモリ効率に優れた手法ではあるが、探索領域の決定に時間が掛かるという欠点があり、探索速度においては改良の余地がある。そこで提案されたのが IMI である。図 5 は IMI における距離計算対象の選択方法を図示したものである。IMI では、プロダクト量子化と呼ばれる方法でクラスタリングを行っている。これは、データを複数の部分空間に分割してベクトル量子化を行う方法である。図 5 のように各データを表すベクトルを 2 分割したとする。それぞれの空間でクラスタリングを行い、図 5 の場合は 1 番目の部分空間に 4 つの代表点（図中の「1 番目の部分代表点」）、2 番目の部分空間にも 4 つの代表点（図中の「2 番目の部分代表点」）が求まり、それらの直積として元の空間で 16 個の代表点が求まる。この場合に必要な距離計算回数は 4+4=8 回のみである。IVFADC のように全ての代表点と距離計算する場合は 16 回の距離計算が必要になる。

部分空間の代表点数が増えれば、両者の差は大きくなるため、IMI での計算量削減効果は大きくなる。ただし、その過程で新たな計算が必要になる。IMI においても距離計算対象を算出するために必要なのは元の空間でのクエリと代表点の距離であるが、IMI はデータを部分空間に分割してから部分代表点との距離を計算する。元の空間でのクエリと代表点の距離を得るためには、各部分空間でのクエリと部分代表点の距離を足し合わせる必要がある。クエリと代表点の全ての組み合わせ（図 5 では 16 通り）について距離を計算するのは効率が悪いため、IMI では Multi-Sequence Algorithm（MSA)と呼ばれるアルゴリズムを用いて、クエリからの距離が小さい順に代表点を算出している。以上の工夫により、IMI は同一の検索精度を実現するための計算時間を IVFADC よりも減らすことに成功した。

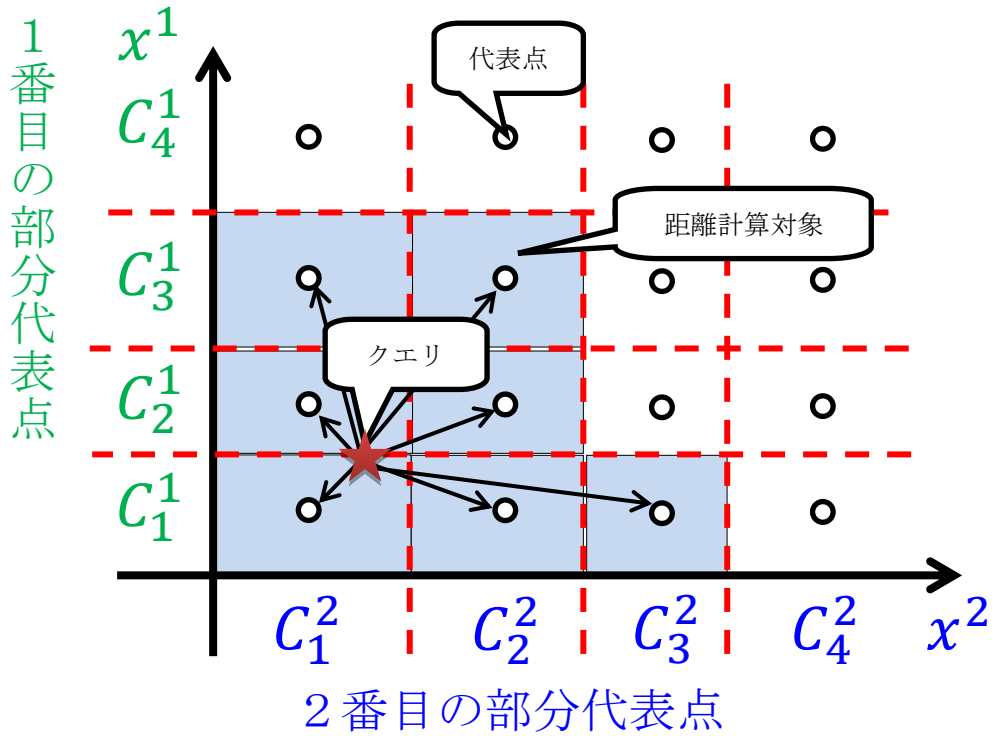


図５　従来手法 Inverted Multi-Index (IMI)における
距離計算対象の選択

高速・高精度な近似最近傍探索の実現

Realization of Fast and Accurate Approximate Nearest Neighbor Search

４．近似最近傍探索のさらなる高速化

我々は前述した既存手法よりもさらに効率的な近似最近傍探索手法を提案している[13]。この手法は、ある検索精度を達成するために必要な計算時間を従来手法である IMI よりも数倍程度高速である。具体的には、検索精度が 90%のときは 2.9 倍、60%のときは 9.4 倍の高速検索が可能である。我々が知る限り、この手法は現在世界で最も高速な近似最近傍探索手法である。

前述の IMI は、IVFADC でクエリに近い代表点を計算するために必要な距離計算回数を削減することで高速化を図った。そして、部分空間でのクエリと部分代表点との距離から元の空間でクエリに近い代表点を効率良く選択する MSA を提案した。ところが、実は MSA の処理には非効率的なところが含まれている。

MSA に求められる処理は、距離計算対象のデータを選択する事である。一旦距離計算対象が選択されれば、距離計算によって最もクエリに近いデータが選択されるため、MSA にはデータとクエリの距離計算やデータをクエリから近い順に並べるなどといった処理は求められていない。それにもかかわらず MSA は距離計算対象のデータを選択する過程でこれらの処理をしてしまっている。そのため、MSA を使った IMI では、ベクトルを 2 分割する場合に最も性能が良くなることが報告されている。代表点の個数を K とすると、ベクトルの 2 分割によって代表点の距離計算回数が $\sqrt{K}$に削減できるのであれば、ベクトルを P 分割すれば$\sqrt[P]{K}$になり、（認識率の低下は別として計算時間に関して言えば）より効率の良い計算ができるはずであるが、MSA を使用する IMI では、MSA の効率の悪さが原因で 2 分割のときが最良である。提案手法は分枝限定法を用いた効率の良いアルゴリズムを MSA の代わりに使用する。これにより、距離計算対象のデータの選択が大幅に効率化され、P>2 のときに最良になった。

提案手法の具体的な処理はデータの登録と検索の 2 つに分けられ、大まかな手順は以下の通りである。

● データの登録（ハッシュテーブルの構築）

(S1) データベースに登録するデータを主成分分析し、データの分散が大きい主成分（軸）を u 個選択する。

(S2) 分散が大きい主成分から p 個ずつ選び、p 次元部分空間を m=u/p 個作成する。

(S3) m 個の p 次元部分空間においてクラスタリングを行い、i 番目の部分空間では k_i 個の部分代表点を選び、データをいずれかの代表点に属するように登録する。k_i は部分空間内のデータの分散を反映して決まる値である。これらの積を

$$N_B = \prod_{i=1}^{m} k_k$$

としたとき、N_B はハッシュテーブルのバケット数になる。実験的にこの値はデータ数に近いときに性能が良いことが分かっている。

● 検索

(R1) (S2)で選択した各部分空間において、(S3)で求めた各部分代表点とクエリの部分ベクトルの距離を計算する。

(R2) (R1)で求めた距離を用いて、元の空間の代表点とクエリとの距離を計算する。ただし、効率化のため、距離を全て計算するのではなく、分枝限定法を用いて必要最低限の計算に留める。

５．評価実験

提案手法の性能評価を行い、3 節で述べた代表的な近似最近傍探索手法と比較する。

実験には 128 次元の SIFT 特徴量[14]を 1 億ベクトル用いる。CPU が AMD Opteron 6174 (2.2GHz)の CPU をシングルコアで使用する。クエリとして 1,000 個のデータを探索したときに真の最近傍点が得られる確率（Recall）と近似最近傍探索に要する平均時間を図 6 に示す。図より、いずれの検索精度においても提案手法が他の手法よりも同じ探索精度を少ない計算時間で実現していることが分かる。特に検索精度が 90%のときは IMI の 2.9 倍、60%のときは 9.4 倍の高速であった。

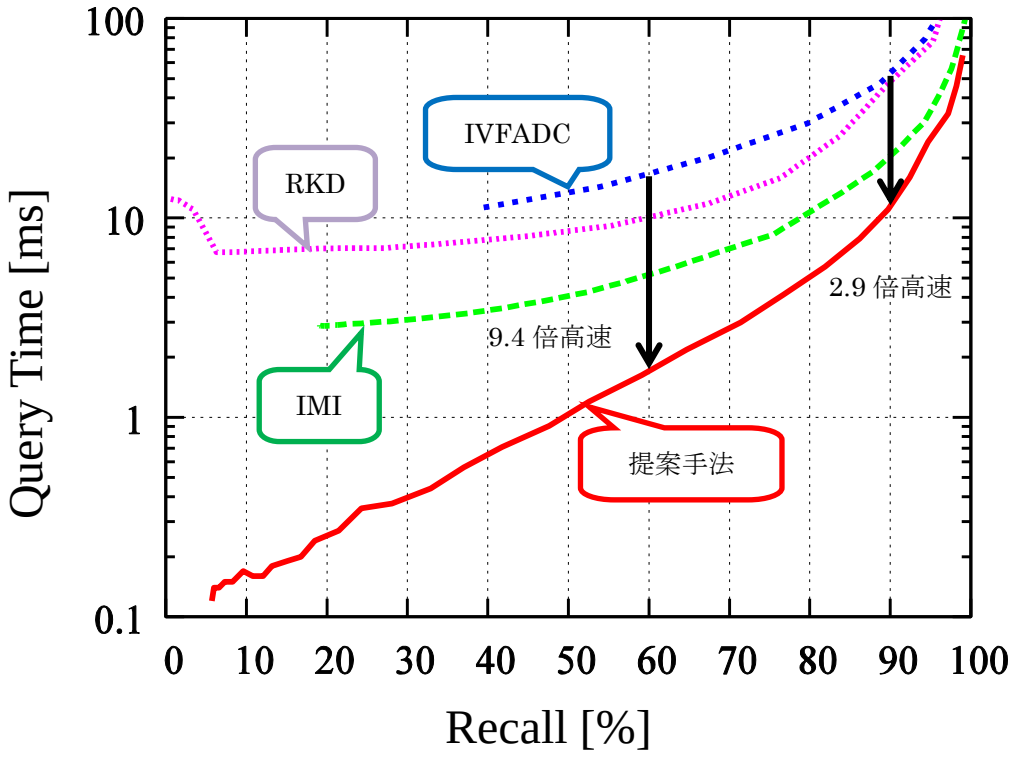


図６　SIFT 特徴量 1 億点を用いた場合の性能比較

６．まとめ

本稿では、膨大な情報の中から所望の情報を効率よく発見できる技術として、近似最近傍探索と呼ばれる技術について述べ、我々が提案した高速で高精度な近似最近傍探索手法を紹介した。メモリ使用量の削減が今後の課題である。

参考文献

[1] 野口 和人, 黄瀬 浩一, 岩村 雅一, "大規模特定物体認識における認識率,処理時間,メモリ量のバランスに関する実験的検討(パターン認識と学習,第12 回画像の認識・理解シンポジウム推薦論文,<特集>画像の認識・理解論文)," 電子情報通信学会論文誌. D, 情報・システム, vol.92, no. 8, pp.1135-1143, 2009.

[2] 中居 友弘, 黄瀬 浩一, 岩村 雅一, "特徴点の局所的配置に基づくディジタルカメラを用いた高速文書画像検索(画像認識,コンピュータビジョン)," 電子情報通信学会論文誌. D, 情報・システム, vol.89, no. 9, pp.2045-2054, 2006.

[3] K. K. Masakazu Iwamura, Tomohiko Tsuji, "Memory-Based Recognition of Camera-Captured Characters," Proc. 9th IAPR International Workshop on Document Analysis Systems (DAS2010), pp.89-96, 2010.

[4] 内海ゆづ子, 坂野悠司, 前川敬介, 岩村雅一, 黄瀬浩一, "局所特徴量と近似最近傍探索を用いた大規模データベースに対する高速顔認識," 情報処理学会研究報告. CVIM, [コンピュータビジョンとイメージメディア], vol.2013, no. 4, pp.1-7, 2013.

[5] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Y. Wu, "An optimal algorithm for approximate nearest neighbor searching fixed dimensions," Journal of the ACM, vol.45, no. 6, pp.891-923, 1998.
[6] C. Silpa-anan and R. Hartley, "Optimised KD -trees for fast image descriptor matching," Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2008.
[7] D. Nist?r and H. Stew?nius, "Scalable Recognition with a Vocabulary Tree," Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp.775-781, 2006.
[8] M. Muja and D. G. Lowe, "Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration.", Proc. International Conference on Computer Vision Theory and Application (VISSAPP'09), 2009.
[9] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni, "Locality-sensitive hashing scheme based on p-stable distributions," Proc. twentieth annual symposium on Computational geometry - SCG '04, p.253, 2004.
[10] Y. Weiss, A. Torralba, and R. Fergus, "Spectral hashing," Advances in Neural Information Processing Systems, no. 1, pp.1-8, 2008.
[11] H. J?gou, M. Douze, and C. Schmid, "Product quantization for nearest neighbor search.," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.33, no. 1, pp.117-128, 2011.
[12] A. Babenko and V. Lempitsky, "The inverted multi-index," Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp.3069-3076, 2012.
[13] 佐藤 智一,岩村 雅一黄瀬 浩一, "空間インデクシングに基づく距離推定を用いた高速かつ省メモリな近似最近傍探索," 電子情報通信学会技術研究報告, vol.112, no. 441, pp.73-78, 2013.
[14] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," International Journal of Computer Vision, pp.1-28, 2004.

この研究は、平成２０年度ＳＣＡＴ研究助成の対象として採用され、平成２１～２２年度に実施されたものです。