情報処理学会論文誌　データベース　Vol. 3　No. 1　18–28 (Mar. 2010)

近似 k 最近傍グラフによる距離空間の近傍検索

岩崎　雅二郎[†1]

大量の多種多様なデータを高速に検索するには空間インデックスが不可欠となる．空間インデックスの 1 つである k 最近傍グラフによるインデックス（kNNG）では最近傍検索の高速性が確認されているが，高コストなインデックス生成および非連結グラフに起因する検索精度の低下といった問題がある．本稿ではこの kNNG に着目し，インデックス生成時に連結グラフを保証しながら逐次ノードを追加し，かつ，生成途中のインデックスを用いて k 最近傍検索を行うことで kNNG を近似する ANNG（Approximate k-Nearest Neighbor Graph）を高速に生成する方法を提案する．さらに，一様分布データや実際の画像特徴量を用いて ANNG の評価を行い，kNNG の問題に対して ANNG が優位であることを確認した．

Proximity Search in Metric Spaces Using Approximate K Nearest Neighbor Graph

Masajiro Iwasaki[†1]

Spatial indexing technology is indispensable for efficient retrieval of large amounts of various data. K-nearest neighbor graph (kNNG), which is one of the graph-based index structures, is known to significantly reduce the search time, however, the cost of index construction is very high, while the disconnected graph components lead to decrease in the search accuracy. In this paper we address these issues, and propose a method to efficiently construct an Approximate k-Nearest Neighbor Graph (ANNG). The proposed method preserves the graph connectivity by adding nodes incrementally, and uses the partially constructed index to perform the k-nearest neighbor search during index construction. We experimentally show that the proposed method (ANNG) outperforms the existing one (kNNG) on both the uniformly distributed data and the actual image feature data.

1. はじめに

インターネットの普及につれて画像，動画，音声などの多様なデータを日常的に利用するようになった現在，大量に存在するデータを高速に検索する技術が注目されてきている．条件に適合するデータを検索するには対象となる全データを評価すればよいが，データ量の増加とともに検索時間が増加し大規模なデータに対しては現実的な時間で処理ができなくなる．そこで，大規模なデータには検索用のインデックスが不可欠となる．画像検索を例にすると画像から抽出した多次元の特徴量を高速に検索するには，特徴量空間を検索する空間インデックスが必要になる．空間インデックスには大きく分けて多次元空間インデックスと距離空間インデックスの 2 種類がある．多次元空間インデックスは次元を意識したインデックスであり，空間で定義された距離関数に対してインデックスのアルゴリズムが定義される．したがって，一般に距離関数が変わるとインデックスのアルゴリズム自体を変えなくてはならない．一方，距離空間インデックスは次元を意識せず，どのような距離関数でも適用できるので応用範囲が広い．また，距離空間インデックスの検索処理の時間は主に，記憶装置からデータをロードする時間，および，距離空間内での距離計算の時間，からなる．近年，メモリが安価で大規模になり大量のデータを高速なメモリに配置することが可能になった反面，画像検索のように超多次元で，かつ，空間上での距離計算のコストが高い応用が増加している．したがって，記憶装置からデータをロードする時間よりも，距離計算の回数を削減することが課題となってきている．また，扱うデータ量が総じて大規模になってきており，その結果，インデックスの生成コストを削減することも課題である．以上のことから，本稿では応用範囲の広い距離空間インデックスを対象とし，インデックスの生成コストを抑えつつ距離計算回数を削減することで高速に多次元データの検索を可能とするインデックスを提案する．

2. 関連研究

空間インデックスは次元数が少ない場合には効果的に機能するが，次元数が増えれば増えるほどインデックスの効果がなくなる「次元の呪い」という現象が生じる．この問題を克服すべく様々なインデックスが提案されている．

†1 Yahoo! JAPAN 研究所
Yahoo! JAPAN Research

多次元空間インデックスでは，木構造型の R-tree [1]，R*-tree [2]，kd-tree [3]，quadtree [4]，SS-tree [5]，SS+-tree [6]，X-tree [7] や次元ごとに圧縮して線形探索を高速化する VA-File [8] が提案されている．また，検索結果に漏れが生じることを許容して高速に検索する近似検索の研究がある．kd-tree に近似検索を適用した ANN [9] や，ハッシュを用いた LSH [10] といった近似検索のインデックスが提案されている．

これらの多次元空間インデックスは様々なデータに対して応用されているが，画像検索に応用する場合には，画像から抽出される特徴量が多様であるために問題が生じる．検索精度を重視すると，カラーヒストグラムの特徴量の距離には 2 次形式（quadratic form）距離[11] を，色の特徴量には色差式に基づく距離を，そしてテクスチャの特徴量には L_1 距離（市街地距離）を利用する，といったように様々な距離の組合せが必要になる．多次元空間インデックスは一般に L_p 距離を対象としているので，このような多様な特徴量に対して単純には多次元空間インデックスを適応できない．そこで，平均色による多次元空間インデックスによって検索した後に誤検索を除去する方法[12] や 2 段階の処理により 2 次形式距離に対応したインデックスを生成する方法[13] などがある．

画像から抽出した特徴量の多様な距離関数に対して多次元インデックスは容易には適用できないが，距離空間インデックスはどのような距離関数にもインデックスのアルゴリズムを変更せずにそのまま適用できる．しかし，多次元空間インデックスと同様に距離空間インデックスにも「次元の呪い」が存在する．距離空間インデックスの場合には次元を考慮しないが，一般に次元数が多くなればなるほど近接する点の個数が減少する，つまり，空間上の点が距離的に疎に分布することになり，距離空間インデックスも多次元空間インデックスと同様に次元数が多くなると高速に検索することが困難になる．

多次元空間インデックス同様，距離空間インデックスの研究も以前から継続的に行われ，様々なインデックスが提案されている．多次空間インデックスと同様に木構造型の GH-tree [14]，GNAT [15]，vp-tree [16]，dvp-tree [17]，mvp-tree [18]，M-tree [19] といった方法が提案されている．また，あらかじめ各点間の距離を計算し，その距離データを用いて距離計算せずに対象となる点の絞り込みを行う AESA [20]，LAESA [21] といった方法が提案されている．

その一方，グラフ構造型のインデックスの有効性が確認されている．グラフ構造型のインデックスの場合には，エッジをたどりながら条件に適合するノードを探索して検索結果を得る．各ノードからのエッジ数が多い場合にはグラフの生成コストが大きくなり，かつ，探索コストも増えるが，検索精度は高くなる．逆にエッジが少ない場合にはグラフの生成コストが減り，かつ，探索コストも減るが，検索精度は低くなる傾向がある．ボロノイ図において

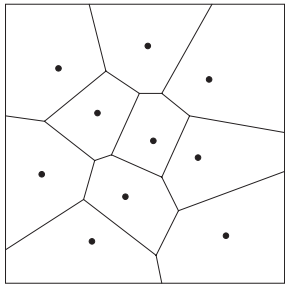

図 1　ボロノイ図
Fig. 1　A voronoi diagram.

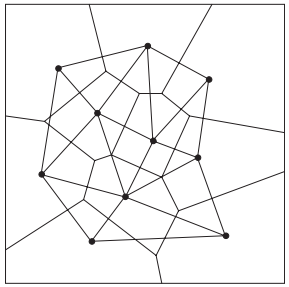

図 2　ドロネーグラフ
Fig. 2　A delaunay graph.

隣接するノードをエッジで結合したグラフであるドロネーグラフ[22] ではエッジ数が比較的少なく，探索コストを低減でき，かつ，高い検索精度を実現できる．ボロノイ図の例を**図 1** に，そのドロネーグラフを**図 2** に示す．

ユークリッド距離の場合にはドロネーグラフを生成することができるので検索用のインデックスとして利用できるが，多次元の場合にはドロネーグラフの生成コストは高く，さらに，ユークリッド距離ではない場合にはドロネーグラフを生成することが困難とされている．また，次元数が増えるほどエッジ数は増加するので，多次元データの場合にドロネーグラフが生成できたとしても，そのエッジ量が膨大となるので実用的とはいい難い．

そこで，ドロネーグラフではないが，その一部を構成する木構造型のインデックスである sa-tree [23] や，擬似的にドローネーグラフを生成する最近傍検索の方法[24] が提案されている．また，k 最近傍グラフにより最近傍のノードを検索する kNNG [25] も提案されている．

kNNG は各ノードから k 個の最近傍のノードへの有向エッジで構成されたグラフである．kNNG は簡便なグラフ構造でありながら，最近傍検索において少ない距離計算回数で高い検索精度を実現している反面，kNNG の生成コストがきわめて高い．また，kNNG は連結グラフ（グラフ内の任意の 2 点間の経路が必ず存在する無向グラフ）ではない，つまり，分離したグラフが存在するので検索できないノードが存在する．さらに，検索において重要である範囲検索（指定された範囲内の点の検索）や k 最近傍検索（指定された個数 k の最近傍の点の検索）に関して言及されていない．そこで，本稿では kNNG のグラフ構造を近似する ANNG（Approximate k-Nearest Neighbor Graph）を提案し，評価実験により kNNG の問題点に関して ANNG が優位であることを確認する．

3. グラフ構造型インデックスによる検索

ユークリッド空間においてグラフが以下の**条件 1** を満たすならば，エッジで結合した近傍ノードを探索すること（局所探索）によって最近傍検索が可能である[23] とされている．

条件 1　G をグラフ，p を G のノード，$\mathrm{N}(G,p)$ を G における p の近傍ノードの集合，q を検索点，$\mathrm{d}(p,q)$ を p，q 間の距離とすると，ノード p が $\mathrm{N}(G,p)$ に属する任意のノード x に対して $\mathrm{d}(p,q) < \mathrm{d}(x,q)$ が成り立つならば，G に属する任意のノード x に対して $\mathrm{d}(p,q) < \mathrm{d}(x,q)$ が成り立つ．

条件 1 を満足する場合，**Algorithm 1** に示す最良優先探索により最近傍検索が可能である．Algorithm 1 の G をグラフ，q を検索点，s を探索開始ノード（グラフの探索を開始するノード）とする．各ノードに全ノードへのエッジが付与されている完全グラフも条件 1 を満たすが，あるノードの近傍ノードは全ノードとなるので，全件に対して線形探索することと同等となる．ドロネーグラフも条件 1 を満足する[23] とされており，かつ，比較的少ないエッジで構成される．k 最近傍グラフ（kNNG）の場合には条件 1 を満足しないので，Algorithm 1 では最近傍のノードが見つからない場合が存在する．そこで，異なる探索開始ノードにより何回か最近傍検索を繰り返し，最も近いノードを最近傍のノードとする必要がある．探索開始ノードの決定方法は様々考えられるが，本稿では kNNG，ANNG ともに全ノードからランダムに選択する方法を採用した．

Algorithm 1 NearestNeighborSearch(G,q,s)

```
p ← s
P ← N(G,p)
c ← argmin_{x∈P} d(x,q)
if d(c,q) < d(p,q) then
  p ← NearestNeighborSearch(G,q,c)
end if
return p
```

範囲検索の場合には，まず最近傍検索（Algorithm 1）を行い，次にエッジをたどって検索範囲内のすべてのノードを探索する．条件 1 を満足する場合にはこの方法で十分であるが，kNNG の場合には，最近傍検索において範囲内の近傍ノードが見つからない場合がある．そこで，異なる探索開始ノードにより範囲内のノードが見つかるまで繰り返し最近傍検索を行う．また，範囲内のノードの探索においても，最近傍のノードへの経路が 1 度範囲外に出た後に再度範囲内に戻る場合を想定しなければならない．そこで，検索範囲より広い範囲を探索する必要があるので，探索する範囲（超球）の探索半径 r_s を式 (1) のように決定する．

$$r_s = (1+\epsilon)r \tag{1}$$

ただし，r は検索範囲を示す検索半径，ϵ は探索半径係数とする．この探索半径 r_s を用いた範囲検索のアルゴリズムを **Algorithm 2** に示す．r を検索半径，R を検索結果集合とし，C は同じノードを何度も探索しないように判定するための集合である．R，C の初期値には

Algorithm 2 RangeSearch(G,q,s,r,R,C)

```
r_s ← (1+ε)r
for all p ∈ N(G,s) do
  if p ∉ C then
    C ← C ∪ {p}
    if d(p,q) ≤ r then
      R ← R ∪ {p}
    end if
    if d(p,q) ≤ r_s then
      RangeSearch(G,q,p,r,R,C)
    end if
  end if
end for
```

空集合を設定する．

本稿では，検索半径の初期値を無限大とし，検索結果の最遠のノードの半径により検索半径を狭めていく一般的な k 最近傍検索の方法を範囲検索（Algorithm 2）に適用して k 最近傍検索を実現する．範囲検索と同様に，まず最近傍検索を行うが，範囲検索とは異なり検索半径の初期値が無限大であることから，最近傍検索により必ず範囲内（無限大）に入るので最近傍検索は 1 回で十分となる．k 最近傍検索のアルゴリズムを **Algorithm 3** に示す．k_s には検索結果の最近傍のノードの数を，検索半径 r の初期値には無限大を，R，C の初期値には空集合を設定する．

このようにして最近傍検索，範囲検索，k 最近傍検索が可能であるが，条件 1 を満たさないので，いずれも近似検索となり検索漏れが発生する可能性がある．しかし，エッジ数を増やすことで条件 1 を満たすノードが増加し，その結果，検索精度が向上する．したがって，検索精度をエッジ数により制御が可能であると考える．

Algorithm 3 KnnSearch(G, q, s, r, k_s, R, C)

```
for all p ∈ N(G, s) do
  if p ∉ C then
    C ← C ∪ {p}
    if d(p, q) ≤ r then
      R ← R ∪ {p}
      if |R| > k_s then
        R ← R − {argmax_{x∈R} d(x, q)}
      end if
      if |R| = k_s then
        r ← max_{x∈R} d(x, q)
      end if
    end if
    r_s ← (1 + ε)r
    if d(p, q) ≤ r_s then
      KnnSearch(G, q, p, r, k_s, R, C)
    end if
  end if
end for
```

4. ANNG インデックス

kNNG では次に示す問題点が存在する．kNNG のインデックスを生成するには全ノード間の距離計算を必要とすることから距離計算回数は以下の式で求められる．

$$\text{距離計算回数} = \frac{1}{2}N(N-1) \tag{2}$$

したがって，kNNG の計算オーダは $O(n^2)$ となり，データ量が多くなると現実的な時間ではインデックスが生成できなくなるという問題がある．また，連結グラフではないためにグラフをたどっても到達できないノードが存在することになり，結果的に検索精度の低下を招くという問題もある．つまり，kNNG では**図 3** のようにある程度離れたクラスタが存在すると分離したグラフが生成されてしまい，グラフをたどっても到達することができないノードが存在することになる．

これらの問題点を解消するために，kNNG の以下の点に着目した．kNNG は条件 1 を満たしていないので検索漏れが発生するが，それを許容している．漏れが発生することを許容するならば，k 最近傍グラフを正確に構成する必要はなく，k 最近傍グラフが近似されていれば同等の検索精度が実現可能であろうと想定できる．そこで，以下に示す ANNG を提案する．

既存のグラフに 1 ノードずつ無向エッジにより逐一連結することで連結グラフを保証する．また，追加するときに k 個の最近傍のノードを求める必要があるが，全ノードとの距

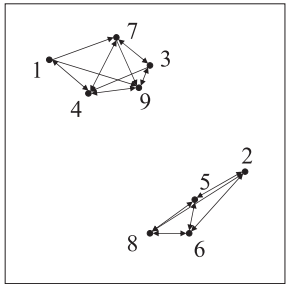

図 3　kNNG
Fig. 3　kNNG.

Algorithm 4 Create(G, P, k)

```
for all q ∈ P do
  if G = ∅ then
    G ← {q}
  else
    s ← Random(G)
    s ← NearestNeighborSearch(G, q, s)
    r ← ∞
    R ← ∅
    C ← ∅
    KnnSearch(G, q, s, r, k, R, C)
    G ← G ∪ {q}
    N(G, q) ← R
    for all p ∈ R do
      N(G, p) ← N(G, p) ∪ {q}
    end for
  end if
end for
```

離を計算したうえで k 個の最近傍のノードを求める処理はきわめてコストの高い処理である．そこで，生成過程にあるインデックスを利用して前述の k 最近傍検索（Algorithm 3）を用いて検索することにより低コストで k 個の最近傍のノードを求める．ANNG 生成の手順を **Algorithm 4** に示す．ただし，P を登録点の集合，Random(G) は G からランダムに 1 ノードを返す関数，k は各ノードからエッジで結合される近傍のノードの数である．

図 4 に ANNG の例を示す．図中の数字は追加順序を示す．図 3 のように kNNG では非連結グラフが生成されるのに対して ANNG では連結グラフが生成される．ANNG は無向エッジを逐一追加するので，たとえば 2 つのノードのみが追加された時点では，第 1 のノードには 1 本のエッジしか存在しないが，順次ノードを追加していくうえで，第 1 のノードの近傍にノードが追加されると無向エッジが第 1 のノードにも追加されることになり，結果的に各ノードは近傍のノードへのエッジが付与されることになる．実際には 1 つのノードから k 本以上のエッジが生成されることになり，各ノードのエッジ数は k とはならないので，k 本のエッジを有する kNNG とはエッジ数では異なるが，kNNG と同様に各ノードから近傍のエッジが生成されることになる．なお，初期に追加したノードにはエッジが多く付与されることになるが，全体のエッジ数は最大 kn に抑えられる．このように生成された ANNG は kNNG とは同一のグラフ構造にはならないが，検索時の漏れを許容するインデックスとしては十分であり，かつ，kNNG の問題点を解消することが可能である．

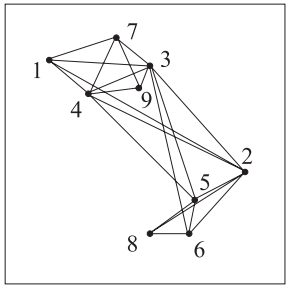

図 4　ANNG
Fig. 4　ANNG.

5. 評価実験

本稿では，kNNG および ANNG に適用できる範囲検索および k 最近傍検索のアルゴリズムを定義したうえで ANNG を提案した．そこで，一様分布データを用いて，定義した範囲検索および k 最近傍検索の有効性の評価，インデックス生成および検索精度の比較評価，そして，実際の画像特徴量による検索精度の比較評価を行った．

評価に使用した一様分布データと画像特徴量は，いずれもデータ数は 10 万件である．一様分布データは範囲 [0.0,1.0] の浮動小数点の要素からなる次元数 10 から 500 の多次元ベクトルデータであり，L_2 距離を距離関数として用いた．画像特徴量は，画像データから抽出したカラーヒストグラム，色やエッジの分布，テクスチャで構成される特徴量である．特徴量の総次元数は 1,228 であり，各次元は 1 バイトの整数型である．式 (3) のように各特徴量の距離（色差や L_1 距離など）の加重平均を距離関数として用いた．

$$\mathrm{d}(x, y) = \frac{1}{n} \sum_{i}^{n} w_i \mathrm{d}_i(x_i, y_i) \tag{3}$$

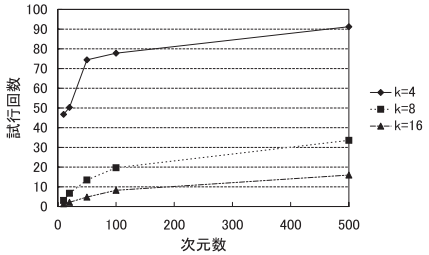

図 5　kNNG におけるエッジ数 k に対する次元数と最近傍検索試行回数の関係
Fig. 5　Number of dimensions versus number of nearest neighbor search trials for number of edges k using kNNG.

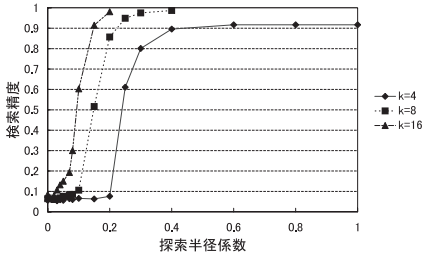

図 6　kNNG におけるエッジ数 k に対する探索半径係数と検索精度の関係
Fig. 6　Exploring radius factor versus recall for number of edges k using kNNG.

なお，x_i，y_i は i 番目の単一特徴量，x，y はそれぞれ x_i，y_i で構成される複合特徴量，$d_i(x_i, y_i)$ は i 番目の単一特徴量の距離関数，w_i は i 番目の単一特徴量の重み付け，n は単一特徴量の総数である

範囲検索および k 最近傍検索の検索精度の測定では，各評価ごとに検索を 50 回試行して各指標の平均値を求めて評価した．範囲検索では検索数が 20 件となる検索半径をあらかじめ決定し検索を行った．これは，同一の検索数を保証することによって，次元数を変動させた場合の比較評価を可能とするためである．また，k 最近傍検索の検索結果のノード数 k_s は 20 とした．

5.1　一様分布データによる範囲検索と k 最近傍検索の有効性の評価

kNNG を用いて一様分布データに対する範囲検索（Algorithm 2）と k 最近傍検索（Algorithm 3）のアルゴリズムの有効性を評価した．

前述のように範囲検索では，まず検索範囲内のノードが見つかるまでランダムに選択した探索開始ノードを用いて最近傍検索（Algorithm 1）を繰り返す．検索範囲内のノードが見つかったなら，そのノードを探索開始ノードとして範囲検索を行う．エッジ数 k ごとに探索半径係数を 0.0 から 0.1 刻みで 0.5 まで変化させて範囲検索したときの最近傍検索の試行回数の平均を**図 5** に示す．エッジが少ない場合には条件 1 を満たすノードが減り，その結果，範囲内のノードが見つからず，最近傍検索の試行回数が増加している．また，より多くのノードが条件 1 を満たすには次元数が増えるに従いエッジ数を多くしなければならない．次元数が増えてもエッジ数が同等なら検索精度は低下する．したがって，同一エッジ数では次元数が増えるに従い最近傍検索の試行回数が増加することになる．

範囲検索の評価のために，50 次元のデータにおいて探索半径係数を 0.0 から 1.0 へ徐々に増やし各検索における検索精度と距離計算回数を求めて，検索精度が 0.98 を超えた時点で測定を終了した．以降の検索精度の測定は同様の方法で行った．以下に検索精度の指標である再現率と適合率の算出式を示す．なお，正答数は検索されるべき検索結果の数である．

$$\text{再現率} = \frac{\text{検索結果内の正答数}}{\text{全正答数}} \tag{4}$$

$$\text{適合率} = \frac{\text{検索結果内の正答数}}{\text{検索結果数}} \tag{5}$$

アルゴリズム上，検索結果に誤検索は含まれず，適合率はつねに 1.0 となるので，検索精度の指標として再現率を用いた．各エッジ数 k に対する探索半径係数と検索精度の関係を**図 6** に示す．

探索範囲が広がれば検索精度が向上ことは自明であるが，エッジ数が増加しても検索精度が向上する．つまり，エッジ数が増えることにより，近傍のノードを網羅的に探索することになるので検索精度が向上する．いい換えれば，エッジ数が増加することにより条件 1 を満足するノードが増加するので検索精度が向上する，ともいえる．エッジ数が 4 の場合には探索半径係数が増加しても検索精度が頭打ち状態となっている．これは kNNG ではエッジの連結性を保証していないので，エッジ数が少ない場合に検索されるべきノードへの経路自体がなくて到達できない状況が発生することに起因すると考えられる．

図 6 から判断すると検索精度を上げるには単純に探索半径係数を大きくすればよいことになるが，検索コストを考慮する必要がある．高速なメモリ上に配置された多次元ベクトルデータにおける検索コストは，ほぼ距離計算コストとなる．そこで，検索性能を評価するために，図 6 の測定結果の距離計算回数と検索精度の関係を**図 7** に示す．自明ではあるが距離計算回数が増加するに従い，検索精度が向上している．また，エッジ数の多い方が距離計

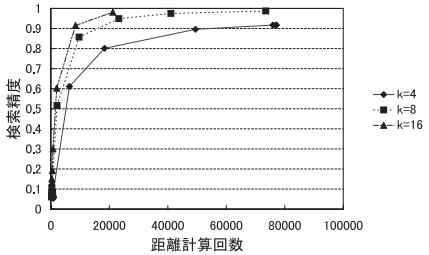

図 7　kNNG におけるエッジ数 k に対する距離計算回数と検索精度の関係
Fig. 7　Number of distance computations versus recall for number of edges k using kNNG.

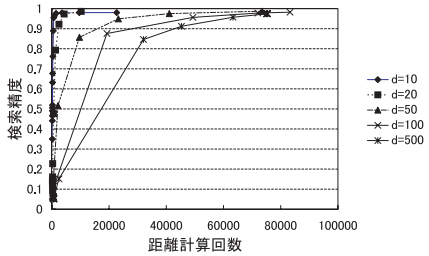

図 8　kNNG の範囲検索における次元数 d に対する距離計算回数と検索精度の関係
Fig. 8　Number of distance computations versus recall for number of dimensions d for range search using kNNG.

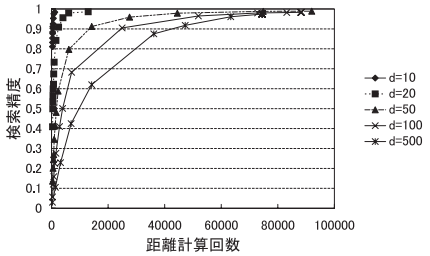

図 9　kNNG の k 最近傍検索における次元数 d に対する距離計算回数と検索精度の関係
Fig. 9　Number of distance computations versus recall for number of dimensions d for knn search using kNNG.

算回数は少なく抑えられている．

エッジ数が 8 における各次元数 d に対する距離計算回数と検索精度の関係を**図 8** に示す．次元数が増えるに従い距離計算回数が増えている．次に k 最近傍検索でのエッジ数 8 における各次元数 d に対する距離計算回数と検索精度の関係を**図 9** に示す．k 最近傍検索は範囲検索とほぼ同様の傾向を示している．

これらの結果から kNNG では次元数が大きい場合でも本稿の範囲検索や k 最近傍検索のアルゴリズムを用いれば距離計算回数を抑えつつ検索可能であると判断できる．また，本評価で用いた次元数より，さらに大きい次元数の場合にもエッジ数を増やすことで距離計算回数を抑えつつ検索が可能であると予想される．

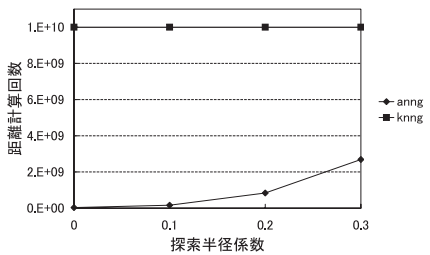

図 10　インデックス生成時の探索半径係数と距離計算回数の関係
Fig. 10　Exploring radius factor and number of distance computations during index creation.

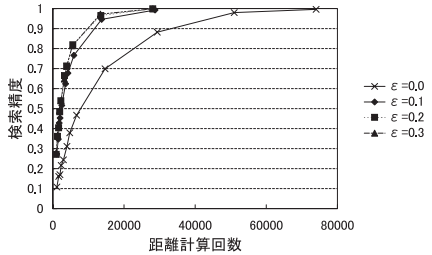

図 11　ANNG におけるインデックス生成時の探索半径係数に対する距離計算回数と検索精度の関係
Fig. 11　Number of distance computations versus recall for exploring radius factor of index creation using ANNG.

5.2　一様分布データによる kNNG と ANNG のインデックス生成の比較評価

前述のように ANNG の生成ではインデックスに逐一ノードを追加する．追加時に近傍のノードへのエッジを生成するためにインデックスを用いて k 最近傍検索を行うので，検索コストが低ければ低いほど，インデックスの生成コストも低くなるという特徴を ANNG は有する．そこで，kNNG と ANNG のインデックスの生成コストを比較する．

ANNG と kNNG の生成の比較を行うにあたりエッジ数の対応を明確にする必要がある．kNNG では k 個の最近傍のノードへの有向エッジを生成するので，n 個のノードを登録するとインデックス全体では kn 本の有向エッジが生成される．ANNG は生成時に k 最近傍検索により k 本の無向エッジが生成される．つまり，有向エッジで表現すると $2k$ 本の有向エッジが生成され，インデックス全体では $2nk$ 本の有向エッジが生成されることになる．比較評価するためにインデックス全体での有向エッジの数を同等とする必要がある．インデックス全体の有向エッジ数を同等とするには，kNNG の k 本に対して ANNG では $k/2$ 本のエッジを生成することにより，全体では，ほぼ同等数のエッジのインデックスが生成されることになる．したがって，本評価ではたとえば k を 4 とする ANNG とは，生成時に k を 2 とする k 最近傍検索を行って生成した ANNG を意味する．

kNNG と ANNG の次元数 50，エッジ数 8 における一様分布データに対するインデックス生成時の探索半径係数と距離計算回数の関係を**図 10** に示す．kNNG の距離計算回数は式 (2) により求められるので，探索半径係数の値にかかわらず一定値となる．ANNG では探索半径係数が増加すれば距離計算回数も増加しているが，kNNG に比較するときわめて

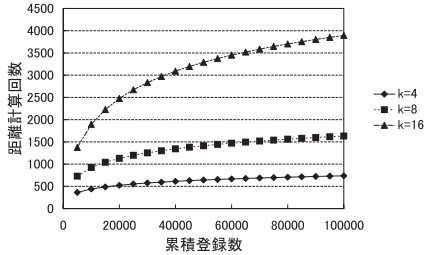

図 12　ANNG におけるエッジ数に対する累積登録数と 1 ノード登録時の距離計算回数の関係
Fig. 12　Total number of nodes versus number of distance computations of each node insertion for number of edges using ANNG.

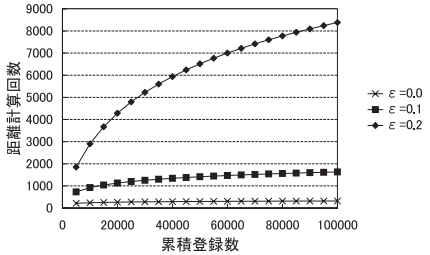

図 13　ANNG における探索半径係数に対する累積登録数と 1 ノード登録時の距離計算回数の関係
Fig. 13　Total number of nodes versus number of distance computations of each node insertion for exploring radius factor using ANNG.

少ない距離計算回数でインデックスが生成できる．

ANNG では前述のようにインデックス生成時に生成途中のインデックスを用いて k 最近傍検索を行い，その結果をそのインデックスに追加する．したがって，インデックス生成時の k 最近傍検索の探索半径係数の違いはインデックスの生成精度に影響し，最終的に生成されたインデックスを用いた検索精度にも影響する．そこで，インデックス生成時の k 最近傍検索の探索半径係数 ϵ が異なるインデックスに対する k 最近傍検索時の距離計算回数と検索精度の関係を**図 11** に示す．探索半径係数が 0.0 と 0.1 では検索精度に大きな違いが生じているが 0.1 と 0.2 では大差はなく 0.2 と 0.3 ではほぼ同値になっていることから，探索半径係数は 0.1 が適当と判断する．また，図 10 において探索半径係数が 0.1 の場合にはインデックス生成時の距離計算回数は kNNG に対して ANNG は約 1.6%でしかなく，ANNG の生成コストはきわめて少ない．

次元数 50，探索半径係数 0.1 においてエッジ数 k に対するインデックス生成途中における 1 ノード登録時の距離計算回数を**図 12** に示す．また，次元数 50，エッジ数 8 において探索半径係数 ϵ に対する登録途中の 1 ノード登録時の距離計算回数を**図 13** に示す．これらの図から累積登録数が多いほど 1 ノードの登録時における距離計算回数はそれほど増えないことが示されている．式 (2) より kNNG の計算オーダは $O(n^2)$ に対して，累積登録数が多い場合には ANNG は $O(n)$ に近づくと判断できる．したがって，大量のデータに対してインデックスを生成する場合にも距離計算回数を低く抑えることが可能であると考えられる．

表 1　10 次元での距離計算回数と検索精度
Table 1　Number of distance computations and accuracy of nearest neighbor search for 10 dimensions.

	kNNG		ANNG	
k	距離計算回数	検索精度	距離計算回数	検索精度
4	12.9	0.03	20.9	0.00
8	39.1	0.07	61.4	0.08
16	92.4	0.33	151.7	0.53

表 2　20 次元での最近傍検索の距離計算回数と検索精度
Table 2　Number of distance computations and accuracy of nearest neighbor search for 20 dimensions.

	kNNG		ANNG	
k	距離計算回数	検索精度	距離計算回数	検索精度
4	12.7	0.00	35.5	0.00
8	33.8	0.01	74.9	0.08
16	71.1	0.05	154.8	0.13

5.3　一様分布データによる kNNG と ANNG の検索の比較評価

次元数 10 と 20 における近傍ノードへのエッジ数 k ごとの kNNG と ANNG の最近傍検索（Algorithm 1）の検索精度を**表 1**，**表 2** に示す．検索精度は最近傍検索を 100 回試行した結果のうちの正答数の割合である．距離計算回数は kNNG より ANNG の方が多いものの検索精度は ANNG の方が高くなっている．kNNG は非連結グラフあることから，最近傍のノードまで到達できずに途中で処理を終了する場合が多いので，距離計算回数が少なく，かつ，検索精度が低く抑えられていると考えられる．一方，ANNG は連結グラフであることから最近傍のノードへ到達できる場合が多くなり，その分，距離計算回数が増えて検索精度も高くなっていると考えられる．また，表 1 の k が 4 の場合には kNNG の検索精度が ANNG より上回っている．これはエッジ数が少ない場合には，近傍のノードにエッジが正確に付与されている kNNG の方が効果的に探索が可能であることに起因すると考えられる．

kNNG と ANNG の範囲検索において次元数 50 での各エッジ数に対する距離計算回数と検索精度の関係を**図 14** に示す．図中の knng および anng に続く数値はエッジ数を意味する．kNNG は連結性が保証されていないのでエッジ数が少ない場合には検索精度が低くなるが，エッジ数が多い場合には連結性は保証されないが，孤立したグラフが少なくなり，その結果，検索精度の差異がほぼなくなると考えられる．同一の条件で k 最近傍検索時の kNNG と ANNG の距離計算回数と検索精度の関係を**図 15** に示す．k 最近傍検索は範囲検

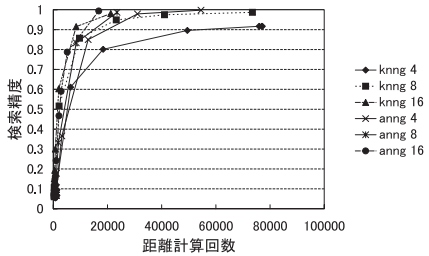

図 14　範囲検索時の距離計算回数と検索精度の関係
Fig. 14　Number of distance computations versus recall for range search.

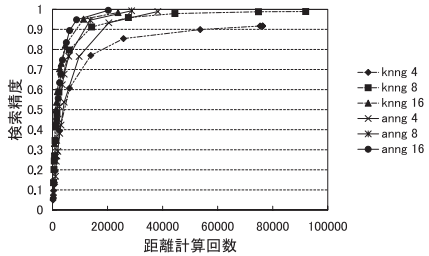

図 15　k 最近傍検索時の距離計算回数と検索精度の関係
Fig. 15　Number of distance computations versus recall for k nearest neighbor search.

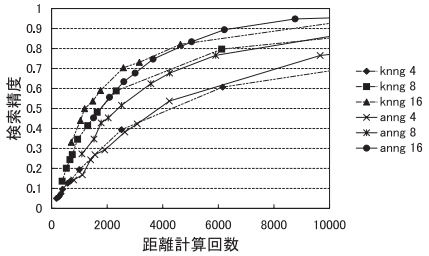

図 16　k 最近傍検索時の距離計算回数と検索精度の関係
Fig. 16　Number of distance computations versus recall for k nearest neighbor search.

索とほぼ同様の傾向を示している．

図 15 の距離計算回数が少ない部分のみを**図 16** に示す．距離計算回数が少ない場合には kNNG の方が若干精度が高い．kNNG の方が ANNG より近傍のノードへのエッジが精度良く生成されているので，距離計算回数が少ない場合，つまりは，探索範囲が狭い場合には精度が高くなる傾向がある．しかし，距離計算回数が多い場合，つまりは，探索範囲が広く網羅的にグラフを探索する場合には連結グラフではないことにより検索精度が伸びないと考えられる．

5.4　一様分布データによる既存のインデックスとの比較評価

20 次元と 50 次元の一様分布データに対して既存の距離空間インデックスとの比較評価を行った．比較したインデックスは Metric Spaces Library [⋆1] で提供されている LAESA，

⋆1 http://www.sisap.org/Metric_Space_Library.html

表 3　20 次元での距離計算回数と検索精度
Table 3　Number of distance computations and recall for 20 dimensions.

インデックス	LAESA	sa-tree	mvp 2	mvp 6
距離計算回数	89,706.4	91,532.3	100,000.0	100,000.0
検索精度	1.0	0.937	1.0	1.0

表 4　50 次元での距離計算回数と検索精度
Table 4　Number of distance computations and recall for 50 dimensions.

インデックス	LAESA	sa-tree	mvp 2	mvp 6
距離計算回数	99,977.2	99,999.9	100,000.0	98,922.0
検索精度	1.0	1.0	1.0	1.0

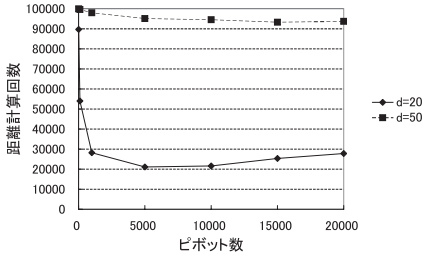

図 17　LAESA におけるピボット数と距離計算回数の関係
Fig. 17　Number of pivots versus number of distance computations for number of dimensions using LAESA.

sa-tree，mvp-tree である．20 次元および 50 次元に対する k 最近傍検索時の各インデックスの距離計算回数を**表 3** および**表 4** に示す．mvp-tree に関しては木構造上のノードの分岐数が 2 と 6 の場合について測定した．木構造上の各ノードのピボット数は 1 である．なお，mvp-tree のピボットとは木構造を構成する各ノードの部分空間を分割するための基準となる点のことである．いずれのインデックスもほぼ全件の距離計算を行っている．なお，sa-tree のみ検索漏れが生じる可能性のある近似検索なので，20 次元のデータでは検索精度が 1.0 に達していない．

LAESA についてはピボットを増やせば距離計算回数が減るので，ピボット数に対する距離計算回数を測定し，その結果を**図 17** に示す．なお，LAESA のピボットは距離計算の基準となるノードのことである．ピボット数が 5,000 ぐらいまではピボット数が増加するにつれ距離計算回数が減少するが，それ以上になると逆に距離計算回数が増加する．ピボット数

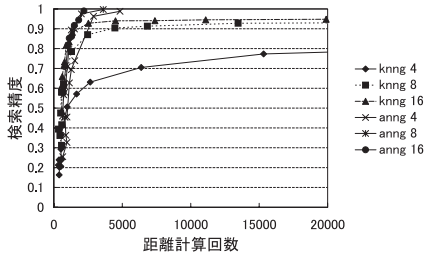

図 18　画像特徴量の範囲検索における距離計算回数と検索精度の関係

Fig. 18　Number of distance computations versus recall for range search using image features.

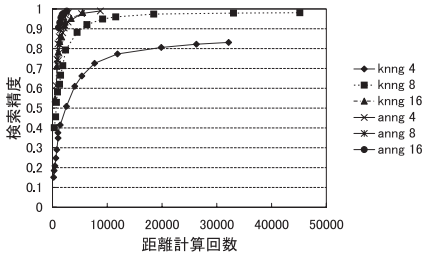

図 19　画像特徴量の k 最近傍検索における距離計算回数と検索精度の関係

Fig. 19　Number of distance computations versus recall for k nearest neighbor search using image features.

が 5,000 のときに距離計算回数は最少である約 20,000 となる．ANNG では図 15 のようにエッジ数が 16 であれば検索精度 0.995，距離計算回数は約 20,000 となり，ほぼ同等の性能となる．しかし，LAESA は約 5 億の距離データをインデックスとして保持しなければならないが，それに対して ANNG は約 160 万の距離データで十分であり，ANNG は LAESA に対してインデックスのサイズが約 0.32%で済む．

以上の結果より，ANNG は近似検索ではあるものの，LAESA 以外の距離空間インデックスよりも検索時の距離計算回数に関して優位であることを確認した．また，LAESA に対しては同等精度の場合には距離計算回数がほぼ同等ではあるが，インデックスのサイズでは ANNG がきわめて優位であることを確認した．

5.5　画像特徴量による kNNG と ANNG の比較評価

実際の画像特徴量による範囲検索の結果を**図 18** に，k 最近傍検索による結果を**図 19** に示す．利用した画像特徴量は 1,000 次元を超えるが一様分布のデータと比較すると距離計算回数がかなり抑えられている．画像特徴量では次元数が多くても特徴量空間上で分布に偏りがあり，インデックスの効果が高いと考えられる．評価に用いたデータの各次元が 1 バイトの整数型に量子化されていることも一因と考えられる．これらの結果から，ANNG は実データの検索においても kNNG に対して優位であることを確認した．

6. おわりに

多次元データの最近傍検索を実現するグラフ構造型のインデックスとして kNNG が提案されている．これは，各ノードに対する k 個の最近傍のノードへのエッジによりグラフが構成されるが，

- 非連結グラフであることに起因する検索精度の低下
- 高コストなインデックス生成

といった問題があり，また，kNNG では最近傍検索の方法しか定義されていない．本稿では範囲検索および k 最近傍検索の方法を定義し，その有効性を確認した．そして，生成中のインデックスを用いて k 最近傍ノードを検索することでインデックスの生成時の距離計算回数を低減させるだけでなく，ノードを無向エッジにより逐一グラフに追加することで連結グラフとなる ANNG を提案した．ANNG では非連結グラフに起因する検索精度の低下を解消しつつ，きわめて低コストでインデックスを生成できることを実際のデータを用いて確認した．今後は，大規模データに対して適用し，さらなる検索速度の向上を目指す．

参　考　文　献

1) Guttman, A.: R-trees: A dynamic index structure for spatial searching, *ACM Sigmod Record*, Vol.14, No.2, pp.47–57 (1984).
2) Beckmann, N., Kriegel, H., Schneider, R. and Seeger, B.: The R*-tree: An efficient and robust access method for points and rectangles, *ACM SIGMOD Record*, Vol.19, No.2, pp.322–331 (1990).
3) Bentley, J.: Multidimensional binary search trees used for associative searching, *Comm. ACM*, Vol.18, pp.509–517 (1975).
4) Samet, H.: The quadtree and related hierarchical data structures, *ACM Computing Surveys* (*CSUR*), Vol.16, No.2, pp.187–260 (1984).
5) White, D. and Jain, R.: Similarity indexing with the SS-tree, *Proc. 12th International Conference on Data Engineering*, pp.516–523 (1996).
6) Kurniawati, R., Jin, J. and Shepherd, J.: The SS+-tree: An improved index structure for similarity searches in a high-dimensional feature space, *Proc. SPIE/IS&T Conf. on Storage and Retrieval for Image and Video Databases V*, Vol.3022, pp.110–120 (1997).
7) Berchtold, S., Keim, D. and Kriegel, H.: The X-tree: An index structure for high-dimensional data, *Readings in multimedia computing and networking*, p.451 (2001).
8) Weber, R., Schek, H. and Blott, S.: A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces, *Proc. International Conference on Very Large Data Bases*, pp.194–205, IEEE (1998).
9) Arya, S., Mount, D., Netanyahu, N., Silverman, R. and Wu, A.: An optimal algorithm for approximate nearest neighbor searching fixed dimensions, *J. ACM*, Vol.45,

No.6, pp.891–923 (1998).
10) Gionis, A., Indyk, P. and Motwani, R.: Similarity search in high dimensions via hashing, *Proc. 25th Internat. Conf. on Very Large Data Bases*, pp.518–528 (1999).
11) Ioka, M.: A method of defining the similarity of images on the basis of color information, IBM Res., Tokyo Res. Lab., Tech. Rep., RT-0030 (1989).
12) Faloutsos, C., Barber, R., Flickner, M., Hafner, J., Niblack, W., Petkovic, D. and Equitz, W.: Efficient and effective querying by image content, *Journal of intelligent information systems*, Vol.3, No.3, pp.231–262 (1994).
13) Seidl, T. and Kriegel, H.: Efficient user-adaptable similarity search in large multimedia databases, *Proc. International Conference on Very Large Data Bases*, pp.506–515, IEEE (1997).
14) Uhlmann, J.: Satisfying general proximity/similarity queries with metric trees, *Information Processing Letters*, Vol.40, No.4, pp.175–179 (1991).
15) Brin, S.: Near neighbor search in large metric spaces, *Proc. International Conference on Very Large Data Bases*, pp.574–584, IEEE (1995).
16) Yianilos, P.: Data structures and algorithms for nearest neighbor search in general metric spaces, *Proc. 4th annual ACM-SIAM Symposium on Discrete algorithms*, Society for Industrial and Applied Mathematics Philadelphia, PA, USA, pp.311–321 (1993).
17) 岩崎雅二郎：類似画像検索を実現する距離空間インデックスの実装及び評価，情報処理学会論文誌：データベース，Vol.40, No.SIG3(TOD1), pp.24–33 (1999).
18) Bozkaya, T. and Ozsoyoglu, M.: Distance-based indexing for high-dimensional metric spaces, *Proc. 1997 ACM SIGMOD international conference on Management of data*, pp.357–368, ACM New York, NY, USA (1997).
19) Ciaccia, P., Patella, M. and Zezula, P.: M-tree: An efficient access method for similarity search in metric spaces, *Proc. International Conference on Very Large Data Bases*, Citeseer, pp.426–435 (1997).
20) Ruiz, E.: An algorithm for finding nearest neighbours in (approximately) constant average time, *Pattern Recognition Letters*, Vol.4, No.3, pp.145–157 (1986).
21) Micó, M., Oncina, J. and Vidal, E.: A new version of the nearest-neighbour approximating and eliminating search algorithm (AESA) with linear preprocessing time and memory requirements, *Pattern Recognition Letters*, Vol.15, No.1, pp.9–17 (1994).
22) Okabe, A., Boots, B. and Sugihara, K.: *Spatial tessellations: concepts and applications of Voronoi diagrams*, John Wiley & Sons, Inc. New York, NY, USA (1992).
23) Navarro, G.: Searching in metric spaces by spatial approximation, *The VLDB Journal*, Vol.11, No.1, pp.28–46 (2002).
24) Sakagaito, J. and Wada, T.: Nearest first traversing graph for simultaneous object tracking and recognition, *IEEE Conference on Computer Vision and Pattern Recognition, 2007, CVPR'07*, pp.1–7 (2007).
25) Sebastian, T. and Kimia, B.: Metric-based shape retrieval in large databases, *Proc. 16th International Conference on Pattern Recognition*, Vol.3, pp.291–296 (2002).

(平成 21 年 9 月 14 日受付)
(平成 22 年 1 月 7 日採録)

(担当編集委員　定兼 邦彦)

岩崎雅二郎（正会員）

1987 年早稲田大学理工学部工業経営学科卒業．1989 年同大学院理工学研究科機械工学専攻修士課程修了．同年日本電気（株）入社．1990 年（株）リコー入社．全文検索，画像検索，文書検索の研究開発に従事．2007 年ヤフー（株）入社．画像検索の研究開発に従事．