類似度と隣接度に基づく画像検索

趙　　夢† 大島　裕明† 田中　克己†

† 京都大学大学院情報学研究科社会情報学専攻　〒 606–8510 京都府京都市左京区吉田本町
E-mail: †{zhao,ohshima,tanaka}@dl.kuis.kyoto-u.ac.jp

あらまし　本研究では，場所名とその場所を代表する写真を与えた際に，それと類似する場所の写真集合を検索する手法を提案する．ある場所と環境や情景が類似する場所を発見するのは困難である．たとえば，現在の Web 画像検索エンジンでは，場所を表すキーワードをクエリとしてそのキーワードに適合する画像を検索することや，クエリとして画像を与え，その画像と類似する画像を検索することなどが可能である．ユーザがある場所とその写真を保有している場合でも，現在の検索エンジンではそれらを基に，類似する場所を検索することは困難である．そのような検索を実現するため，本研究では画像の類似度と画像の隣接度に基づき，クエリとして与えられた場所と類似する場所の写真集合を検索する手法を提案する．

キーワード　場所検索, 画像検索

1. はじめに

様々な種類のコンテンツが Web 上で提供されており、テキスト情報ばかりでなく、画像、音声、映像などのマルチメディアコンテンツも非常に増えてきている。コンテンツを発見するためには、検索エンジンが利用されることが多く、マルチメディアコンテンツのためにも多くの Web 検索システムが提供されている。

たとえば、現在、画像のために提供されている検索エンジンとしては、画像のファイル名や画像の周辺テキストなどのテキスト情報を基にして、キーワード検索するというものが一般的である。これらのテキスト情報は必ずしも画像の重要な特徴を説明しているわけではない。そこで、画像そのものをクエリとして与え、それと類似する画像を検索するシステムも提供されつつある。TinEye(注1)は、画像を与えることで、それとほぼ一致する画像を検索することができるサービスである。また、Google 画像検索では、結果の一部の画像に対して、画像として類似している画像を検索することが可能になっている。

ユーザは、これらを含む、様々な検索エンジンを利用して、求める情報を発見しようとする。そのようなユーザの検索意図の一つとして、我々は、ある場所と類似する場所を検索するということが存在すると考えている。たとえば、「金閣寺における鏡湖池とそれを前にした舎利殿」や、「京都大学における正門からの空間を前にした時計台」というような情景をユーザが思い描き、それと似たような場所を探したいというものである。この場合、検索したい場所は、ユーザがクエリとして思い浮かべている場所とは別の場所である。このような情報検索意図で、思い描いている場所名を表すキーワードや、実際にその情景を表す写真がクエリとして与えられたとしても、既存の検索エンジンを利用して、このような場所検索を実現することは困難である。そこで、本研究では、場所名とその場所を代表する画像を与えた際に、それと類似する場所の画像集合を検索する手法を提案する。

我々が取り組む問題は、ある場所の画像を与え、それと類似する場所の画像を検索するという一種の画像検索である。しかし、ある場所の環境や情景は、必ずしも一つの画像で示されるものではなく、周辺の状況まで含めて、はじめて説明可能だと考える。つまり、クエリとして与えられた画像と隣接する画像も、その場所の情景を表すのに必要であると考える。そして、そのような隣接性を考慮した画像集合と、別の場所を表す画像集合においても同様に隣接性を考慮して、それらの画像の類似性を考慮することで、場所の類似性を計算する手法を提案する。本研究において、ある画像と隣接する画像とは地理的に近い場所で撮影され、撮影画像の一部に重なりがあるような画像のことであり、典型的にはパノラマ合成可能であるような画像のことである。実際、ある 1 枚の画像を見ている際に、その周辺の情景を思い浮かべることは困難である事が多く、その画像だけしか考慮しない場合、周辺まで含めた情景について思い違いをしてしまう可能性もある。例えば、ある素晴らしい情景の画像を見て、時間とお金を掛けてそこに実際に行ったとしても、その周辺を含む情景にがっかりさせられることも十分に考えられる。また、ある情景の場所が非常に遠いところにあるなど、行きづらい場合に、その場所と類似した情景を持つ、より近い場所に行ってみたいということも考えられる。このような場合に、我々が提案する技術である、ある画像の実際の情景と類似する場所を検索するサービスが役立つと考えられる。

以後、2. 節では関連研究について紹介し、3. 節では本研究で取り組む問題について詳しく説明する。4. 節では提案手法について述べ、5. で実験について述べる。最後に、6. 節でまとめと今後の課題について述べる。

(注1)：http://www.tineye.com/

2. 関連研究

局所特徴量を用いた画像の一致性や類似性を求める研究は、すでに 30 年以上行われている。近年、情報検索や自然言語処理分野において文書の特徴をそこに出現する語で表すことと同様に、画像を Visual Words の集合で表現し、画像の類似度を計算する手法が提案されてきている。そこでは、TF-IDF のような概念も取り入れられている。

画像の局所領域から特徴点を抽出する手法としては、Lowe [4] による Scale-Invarient Feature Transform (SIFT) や Bay ら [1] による Speeded Up Robust Features (SURF) がよく知られており、広く利用されている。これらは、スケールや傾きなどの違いによらず抽出される特徴点であり、同一の物体がスケールや傾きを変えて撮影された写真からは、非常に類似性の高い特徴点が抽出される。SIFT では一つの特徴点は 128 次元のベクトルで、SURF では一つの特徴点は 64 次元のベクトルで、それぞれ表現される。

画像の局所特徴を階層的にクラスタリングし、Visual Words の階層構造を構築することで、ロバストな画像類似計算を行う手法が Nister らによって提案されている [2]。まず、対象となる画像群から抽出された局所特徴の抽出を行う。局所特徴量としては、SIFT や SURF が利用可能である。抽出された全ての局所特徴を K-means によってクラスタリングし、各クラスタをさらに K-means によってクラスタリングするということを繰り返し、階層構造を構築する。例えば、7 階層にわたって K-means クラスタリングを行った場合、1,111,111 個のノードを持つツリーが構築されることになる。このツリーは、ボキャブラリツリーと呼ばれる。ある画像が与えられたときに、その画像から取り出された全ての局所特徴を、階層的 K-means クラスタリングのトップノードから最も類似するクラスタを次々とたどっていくように流し、通ったノードを記録する。そのようにして、ノードの数と同じ次元を持つベクトルを生成し、それを、画像の特徴とする。各階層には、下層に行くほど大きくなるように重みが与えられる。画像の類似度は、ベクトルの L1 ノルムや L2 ノルムなどで計算される。局所特徴は、画像の一致性を検出するためには非常に強力だが、類似性は扱いづらい。しかし、Visual Words の階層構造を利用することで、ある程度の画像の類似性を扱うことができる。

複数の写真に重なっている領域があるときに、それらの部分を結合することで写真のパノラマ合成が可能である。Brown らによって提案されたアプローチでは、複数の画像において同一の局所特徴を検出する手法がとられている。彼らは、SIFT 特徴を全ての候補画像から抽出し、kd ツリーを利用したりすることで、k 近傍にある局所特徴を発見する。そして、共通する特徴の数が多く、パノラマ合成の可能性が高い m 枚の候補画像（$m = 6$ などが使われる）を取得する。さらに、RANSAC という手法を用いて、対応する特徴点が幾何学的に整合性を持っているかどうかを検出する。最終的に、確立モデルをもちいて、画像の一致性を検証する。

本研究では、このような画像解析技術を用いながら、目的を達成するための手法を提案する。

3. 画像の類似性と隣接性に基づく場所の類似性

本節では、本研究で取り組む問題について説明する。まず、最終的な入力と出力は以下のように定義される。

入力: 場所の名前とその場所の 1 枚以上の画像

出力: 画像集合

我々の目的は、クエリとして与えられた場所と類似する別の場所を見つけることである。類似する場所というのは、情景、環境、雰囲気などが類似している場所のことである。ユーザは、場所を表す名前をキーワードで与える。さらに、その場所のどのような情景を思い描いているかを端的に表すいくつかの画像を与える。与えられた入力に対して、出力されるのは類似している場所の画像集合であり、それにより、結果的に、ユーザは類似する場所を得ることができる。

入力として、場所名といくつかの画像を用いる理由は、画像がユーザが思い浮かべている情景を説明するのに、キーワードだけでは難しく、画像を用いれば容易に説明できると考えたためである。例えば、雑誌に掲載されているある場所の感動的な風景の写真があり、その風景と類似した場所を表現するのには、言葉よりもその画像そのものを用いた方が良いだろう。

出力としては、画像集合を返すが、実際には、順位付けされた画像列を返す。例えば、その上位 10 件のみを返すことで出力とする。ユーザがさらに画像を求めた場合には、さらに下位の画像集合を与えることができる。

さらに、類似する場所の検索を行うという目的のため、その結果を加工して出力することを検討する。

(a) 地理的に近い場所で撮影された画像は同一の場所の画像集合としてひとまとまりにする。

(b) 画像集合で表現された場所の名前と元キーワードである場所の名前とは同義語である。

(c) 各画像集合内の各画像と、入力である場所を表す画像とはいずれの関係性 (後述) がある。

先述したとおり、我々の目的は、ある情景や環境にある場所を求めることである。この目的を達成するため、我々は以下のような仮定をおいた。

仮定 ある画像が写している場所の情景と、その画像と隣接する画像が写している場所の情景は同一である。また、ある画像と別の画像の類似性が高いとき、それらの場所が同一であっても異なっていても、情景は類似しているといえ、すなわち、場所が異なっていた場合には、それらの場所は類似しているといえる。

この仮定が表すことの 1 つは、ある場所の情景は 1 つの画像では表現しきれるものではないということである。そして、類似する場所を求めるために、本研究では画像の間にある以下の 2 つの関係性について考慮すれば良いと考え、手法を提案する。

- 画像の類似性
- 画像の隣接性

以下で、それぞれについて説明する。

3.1 画像の類似性

画像の類似性とは、ある画像全体の特徴と、別のある画像全体の特徴が類似しているということである。ある画像集合 I_n=$\{img_1, img_2, ..., img_i, ..., img_n\}$ があるとする。ここで、n は画像集合の画像の数である。そのうちの 1 つの画像 img_i が入力として与えられたとき、出力として、その画像と類似する他の画像集合が得られるとする。この機能は、2 つの画像の類似性を計算する以下のような関数があれば実現される。

$$Sim(img_i, img_j)$$

$1 \leqq i \leqq n$、$1 \leqq j \leqq n$、かつ $i \neq j$。

この関数は、実数を返し、img_j と img_i の類似性が高いほど大きな値を返すものとする。

3.2 画像の隣接性

本節では、画像検索における、画像の隣接性について説明する。

例を示しながら説明を行う。図 1 は金閣寺の画像である。キーワードクエリとして「金閣寺」が与えられると、画像検索システムからは、図 1 に示したような画像と類似する画像を数多く検索結果が返される。実際に、金閣寺においてこの写真に写る湖を周ると、図 1(b) のような光景も見ることができる。さらに歩くと、図 1(c) のような光景も目にすることになる。図 1(c) には、まだ、少しではあるが、金閣が見えている。これは確かに金閣寺の光景であるのだが、人によっては分からないかもしれない。さらに、図 1(d) や図 1(e) まで行くと、ここがどこか分かる人はあまりいないと考えられる。これらは全て金閣寺において撮影された写真であるのだが、これらのどれも 1 枚では金閣寺の実際の情景や環境を表現できておらず、これらを総体としたときによりよく金閣寺の実情が表現できると考えられる。また、昼と夜、季節の変化などによって、情景は変化することが考えられる。金閣寺の場合、図 1(f) や図 1(g) のような変化を見せる。これにより、金閣寺の実際の情景といえば、季節や時間による変化も含めて考慮する必要があると考えている。

画像の隣接性とは、ある画像と、別のある画像が部分的に一致しており、実世界で隣接することを指す。ある画像集合 I_n=$\{img_1, img_2, ..., img_i, ..., img_n\}$ があるとする。ここで、n は画像集合の画像の数である。そのうちの 1 つの画像 img_i が入力として与えられたとき、出力として、その画像と隣接する他の画像集合が得られるとする。この機能は、2 つの画像の隣接性を計算する以下のような関数があれば実現される。

$$Adj(img_i, img_j)$$

$1 \leqq i \leqq n$、$1 \leqq j \leqq n$、かつ $i \neq j$。

この関数は、実数を返し、img_j と img_i の隣接性が高いほど大きな値を返すものとする。

3.3 画像の類似性と隣接性の同時利用

本節では、画像の類似性と隣接性を同時に利用した場合にどのようなことが可能になるかについて説明する。

図 2 は、それら 2 つの関係性を考慮することによって、どのようなことが可能になるかを表している。既存の画像検索技術

(a) Image1　(b) Image2

(c) Image3　(d) Image4

(e) Image5　(f) Image6

(g) Image7

図 1　(a) 典型的な金閣寺の画像。(b)-(e) あまり典型的ではない金閣の画像。(f) 秋の紅葉の金閣寺の画像。(g) 冬の雪の金閣寺の画像。(a) の画像だけ見ても金閣寺の実際の情景が分かるわけではない。そのため、我々の仮定の基では、(b)-(e) の画像も (a) の画像とまとめて扱うべきであるとしている。さらに、(f) や (g) のように金閣寺が珍しい情景を見せていることもあり、これらも含めて、金閣寺の実際の情景を表していると考えるべきである。

を用いれば、図中の画像 i_1 がクエリとして与えられたときに、それとパノラマ合成可能であるような画像 i_2 が存在したとすると、画像の隣接性を考慮して、画像 i_2 を発見することは可能である。さらに、画像 i_2 と画像 i_3 の隣接性が高いことを考慮して、画像 i_3 まで発見することも可能である。一方、類似性については、この場合、残念ながら画像 i_1 と類似する画像がなかった。つまり、クエリが画像 i_1 の場合、いかなる類似画像も発見することはできない。ここまでが、画像の隣接性と類似性を別個に利用した場合である。

画像の類似性と隣接性を同時に考慮した場合には、クエリが画像 i_1 の場合、まず、隣接性を考慮して、画像 i_2 が発見され、さらに、画像 i_3 も発見することができる。ここで、画像 i_2 には類似する画像 j_2 と画像 j_3 がある。我々の仮定が正しいとす

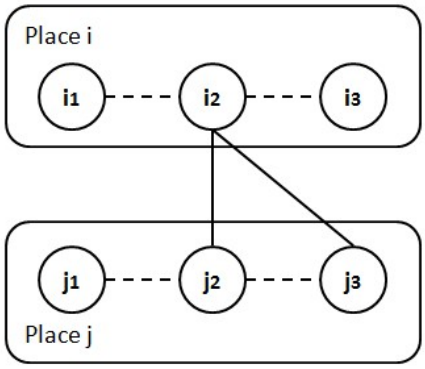

図 2　地理的に同一の場所や近い場所で撮影された画像の隣接性は点線で表現されている。また、画像の類似性は実線で表現されている。それぞれのノードは画像を表しており、枠で囲われた部分は同じ場所や近い場所の画像をまとめている。

ると、画像 i_1 と画像 i_2 の情景は類似しており、さらに、画像 i_2 と画像 j_2 や画像 j_3 の情景は類似していることから、画像 i_1 と画像 j_2 や画像 j_3 の情景が類似していると考えられる。すなわち、我々が本研究で取り組む問題とは、このように、i_1 のような画像がクエリとして与えられたときに、画像 j_2 や画像 j_3 を発見するというものである。

4. 類似場所画像のランキング

与えられた画像と直接的に類似していなくても、情景が似ていると考えられる場所の画像を発見するため、我々は、画像の類似性と隣接性を考慮した画像のランキング手法を提案する。提案する手法は、TextRank に基づくものであり、TextRank において画像の類似性を用いて構築したエッジに、画像の類似性とともに隣接性を用いる。さらに、類似する画像に付与されているタグは類似しているという仮定に基づき、タグの類似度をもエッジの重みにおいて考慮する。

4.1 隣接性を考慮した TextRank に基づく画像ランキング手法

4.1.1 画像の類似性の計算

画像の類似性を計算する手法については、従来手法を用いた。用いた手法は、Nister らによる局所特徴の階層クラスタリングに基づく手法である [2]。Nister らの手法において、画像は 1 つのベクトルで表現される。ボキャブラリツリーの各ノード i に、重み w_i が設定されているとき、クエリとして与えられた画像のベクトル q_i と全体画像集合中の画像のベクトル d_i はそれぞれ以下のように定義される。

$$q_i = n_i w_i$$

$$d_i = m_i w_i$$

ここで、n_i はクエリ画像の特徴量をボキャブラリツリーの最上位より流し込んだ際にノード i を通った回数であり、m_i は現在対象となっている全体画像集合中の画像のそれである。

クエリ画像とある画像の類似性における適合度はこれらの 2 つのベクトルの相違から計算される。Nister らによると、正規化されたベクトルの L1 ノルムを用いた場合、が L2 ノルムを用いる場合よりも良いとされている。そこで、我々は、画像の類似性の計算には L1 ノルムを利用することにした。以下が、Nister らの手法を用いた画像の類似性の程度を計算する式である。

$$NisterScore(img_i, img_j) = s(q,d) = \| \frac{img_i}{\|img_i\|} - \frac{img_j}{\|img_j\|} \|$$

Nister らの手法を用いて計算される類似性は、画像をグレースケールにする処理が含まれているため、色彩などは考慮されない類似性である。そこで、さらに、Color Coherence Vectors (CCV) という色特徴量を用いることとした。これは、Pass らによって提案された色ヒストグラムを基にする手法であり、通常の色ヒストグラムでは考慮されない、各色がどの程度隣接した空間に存在するかを考慮している [5]。色ヒストグラムよりも多少なり情報量が多いため、CCV を用いる事にした。

CCV では、各色（ビン）がどの程度隣接しながら、もしくは隣接せずに存在しているかを表す。CCV によるベクトルは以下のように表現される。

$$< (\alpha_1, \beta_1), ..., (\alpha_n, \beta_n) >$$

CCV を作成するにあたっては、ある閾値が設定される。その閾値よりも多いピクセル数が隣接して存在している場合、α_j にカウントされ、閾値よりも少ないピクセル数の隣接で存在している場合 β_j にカウントされる。

2 つの画像の CCV によるベクトルの違いを基に、類似性が計算される。以下が、その式である。

$$CCVScore(img_i, img_j) = \Delta G = \sum_{j=1}^{n} |(\alpha_j - \alpha'_j)| + |(\beta_j - \beta'_j)|$$

以上の説明した 2 つの類似性計算手法を用いて、我々は、それらを統合した画像の類似計算手法を用いる。すなわち、画像 img_i と画像 img_j の類似性を以下のように表す。

$$Sim(img_i, img_j) =$$

$$w_1 NisterScore(img_i, img_j) + w_2 CCVScore(img_i, img_j)$$

w_1 と w_2 は、それぞれの類似度計算手法のどちらを重視するかの重みであり、$w_1 + w_2 = 1$ とする。

4.1.2 画像の隣接性の計算

画像の隣接性を計算する手法についても既存の技術を用いた。我々が用いたのは、パノラマ構成アルゴリズム [3] である。このアルゴリズムでは、2 つの画像において、パノラマを構成するにあたって、対応する点を発見する。

2 つの画像が与えられたとき、まず、SURF による特徴量をそれぞれから抽出する。そして、2 つの画像において対応する点を発見する。最終的には、RANSAC によって対応点の整合性を考慮する。ここで、隣接性を考慮するにあたって、2 つの画像における対応点の数を用いる事とする。すなわち、2 つの画像 img_i と img_j の隣接性は以下のように計算される。

$$Adj(img_i, img_j) = \frac{N_{cp}}{Min(N_{img_i}, N_{img_j})}$$

ここで、N_{cp} は、対応点の数であり、N_{img_i} は、画像 img_i から抽出された SURF 特徴の数である。

4.1.3 TextRank を基にしたランキング手法

テキスト処理の分野において、典型的な語や文章を求める手法として、グラフベースのモデルが提案されている [6]。そこでは、語や文章がノードとなり、その間のエッジには重みが与えられる。我々のランキング手法は、その手法に基づくものである。グラフにおけるノードは画像である。あるノードのスコアは以下のように計算される。

$$S(V_i) = (1-d) * v_i + d * \sum_{V_j \in In(V_i)} \frac{w_{ji}}{\sum_{V_k \in Out(V_j)} w_{jk}} S(V_j)$$

ここで、d はダンピングファクターであり、0 から 1 の値をとる。通常は、0.85 が用いられる。$In(V_i)$ はノード V_i に向けてエッジをもつノード集合であり、$Out(V_j)$ は V_j からエッジを持つノード集合である。

先述したとおり、本研究では、ノードは画像であり、画像 img_i のノードを V_i とする。また、v_i を V_i のスコアの初期値とする。

$$v_i = \begin{cases} \frac{1}{|SI|}, img_i \in SI \\ 0, img_i \notin SI \end{cases}$$

ここで、SI は、ユーザがクエリとして与えた画像集合を表す。重み w_{jk} は、以下のように定義する。

$$w_{jk} = \lambda_1 Sim(img_j, img_k) + \lambda_2 Adj(img_j, img_k)$$

λ_1 と λ_2 は画像の類似性と隣接性のどちらを重視するかの重みであり、$\lambda_1 + \lambda_2 = 1$ を満たすものとし、何らかの手法によって決定されるものとする。

4.2 タグ類似度によるランキング手法

画像を Web 上で共有することができる Web サイトとして、Flickr(注2)が存在する。そこでは、ユーザは画像をアップロードし、それらにテキストでタグを付与することができる。自分がアップロードした画像だけでなく、他の人の画像にもタグ付けすることが可能である。このような Flickr の特徴は、画像処理において有用な情報源となる。そのため、今回我々は、Flickr より様々な場所の画像を収集してデータベースを構築することにした。特に、近年、カメラが自動的に地理情報を付与していることがあり、地理情報が付与された画像のみを収集した。

先述した TextRank を基にした手法を適用することにより、我々は各画像のランキングのスコアを得ることができる。そこでの上位 N 件の画像が、次のグループ化処理に利用される。

グループ化処理において、画像は地理的に近い画像集合にグループ化される。その際には、画像に付与された緯度経度情報を利用する。画像集合における画像の数が一定数 m（例えば、$m = 10$ などとする）以上になった場合、その画像集合を最終的な出力の候補とする。図 3 は、グループ化の概念図である。それぞれの画像集合において、頻出タグが求められる。各タグはその画像集合で何回出現したかがカウントされる。そして、その上位 k 個の頻出タグが取得される。これらのタグをシードタグと呼ぶ。シードタグと、その他のタグの共起を基に、各タグには重みを与える。そのようにしてタグを次元とするベクトルが以下のように作成される。$\boldsymbol{v_{img_i}}$=(w_1v_1, w_2v_2, ..., w_jv_j, ..., w_mv_m) ここで、$\boldsymbol{v_{img_i}}$は画像 img_i のタグの特徴を表すベクトルであり、w_j はタグ v_j に対する重みである。m は全ての画像におけるタグの総数であり、$1 \leqq i \leqq m$ である。

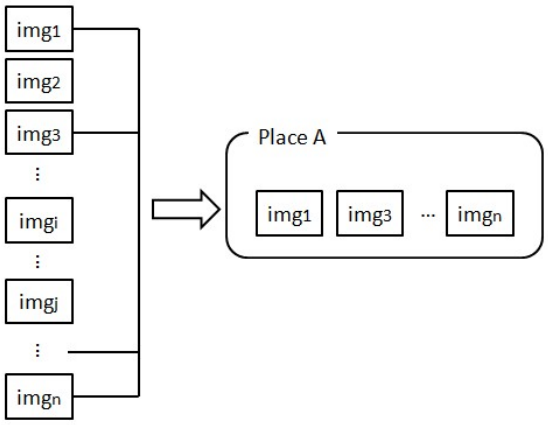

図 3　これらの N 個の画像は、TextRank に基づく手法によって得られた上位 N 件の画像である。例えば、画像 img_1、画像 img_3、...、画像 img_n が地理的に近い場所の画像であったとすると、それらはグループ化されて 1 つの画像集合が形成される。画像集合に含まれる画像の数がある一定数を越えた場合、その画像集合をその画像集合の場所を代表するものとして提示する。

$$v_j = \begin{cases} 1, \text{when it tagged to } img_i \\ 0, \text{otherwise} \end{cases}$$

以下は、タグ v_j とあるシードタグ v_{seed_i} の共起度は以下のように定義する。

$$co(v_j, v_{seed_i}) = \frac{|I_{v_j} \cap I_{v_{seed_i}}|}{|I_{v_j}| + |I_{v_{seed_i}}|}$$

ここで、v_{seed_i} はシードタグの中の 1 つのタグである。I_{v_j} はタグ v_j を付与された画像の集合であり、同様に、$I_{v_{seed_i}}$ はタグ v_{seed_i} が付与された画像の集合である。タグ v_j に対する重みは以下のように定義される。

$$w_j = \sum_{i=1}^{N} co(v_j, v_{seed_i})$$

なお、N はシードタグの総数である。

このようにして、各画像に対してタグの特徴を表すベクトルが生成される。これらのベクトルの類似度はコサインによって計算する。

$$TSim(img_i, img_j) = \frac{v_i \times v_j}{|v_i| \times |v_j|}$$

(注2)：http://www.flickr.com/

この、$TSim(img_i, img_j)$ をグラフにおけるエッジの重み w_{ij} として用いる。各ノードの初期スコアは節 4.1 で説明したランキングのスコアを用いる。そして、TextRank に基づく手法を再適用することによって各画像のあらたなスコアが計算される。

同様に、上位 N 件の画像が次のグループ化処理のために選択される。グループ化処理において、地理的に近い画像がグループ化される。画像数が 10 以上の画像集合が結果候補として取得される。最終的に、画像に与えられたスコアの最大、最小、ないしは平均を用いて、結果の画像集合は順位付けされる。各画像集合の上位 10 件の画像が最終出力とされる。

5. 実　　験

本節では、提案手法を実装して行った実験について説明する。まず、我々は、観光サイトなどから人手で京都にある寺院の名前を 232 個収集した。その中には、金閣寺、銀閣寺、竜安寺などの有名な寺院から、余り知られていない寺院まで含んでいる。それらの各々をクエリとして、Flickr において画像検索を行った。画像検索において得られた結果の中で、画像が撮影された緯度経度情報をタグに含むものだけを収集して、実験において対象とする全画像集合とした。

そのようにして収集された画像は、14,366 枚出会った。タグの類似度を計算するために、全ての画像に付与されているタグ情報を取得し、それらのデータベースを構築した。その中には、画像の緯度経度情報も含まれている。残念ながら、いくつかの寺院においては、緯度経度情報を持つ画像はあまり多く得られなかった。全く画像が得られない寺院もあれば、千枚以上の画像が得られた寺院もあった。

まず、画像の類似性と隣接性を計算する手法がどの程度適切に機能するかを調べる実験を行った。平均適合率の平均（Mean Average Precision: MAP）によって評価したところ、類似性については 0.65、隣接性については 0.41 というスコアが得られた。

次に、提案手法のための実験を行った。収集された画像の数が非常に多く、現在の実装では現実時間で処理を終わらせることが困難であったため、ここでは、小さなデータセットを作成した。まず、14 の寺院(注3)を選択した。そして、各寺院において、10 個の適当な画像を選択し、140 の画像のデータからなる小さなデータセットが作成された。このデータセットに対して我々の提案手法を適用した。

提案手法を適用することで、ランキングされた画像のリストが取得される。先述したとおり、それらの画像は地理的に近いものごとにグループ化されている。図 4 は、「竜安寺」という場所名と、竜安寺のある 3 つの画像がクエリとして与えられたときの実行結果を示している。クエリの画像は、緑の枠で囲われた画像である。画像の間にあるエッジは、類似性と隣接性により、比較的大きな値を持つエッジが存在することを示している。赤いエッジは、2 つの画像の類似性が高いことを示しており、青いエッジは、2 つの画像の隣接性が高いことを示している。

(注3)：永観堂，祇王寺，金閣寺，三千院，神護寺，清水寺，西芳寺，大仙院，天龍寺，東福寺，南禅寺，竜安寺，龍源院，曼殊院

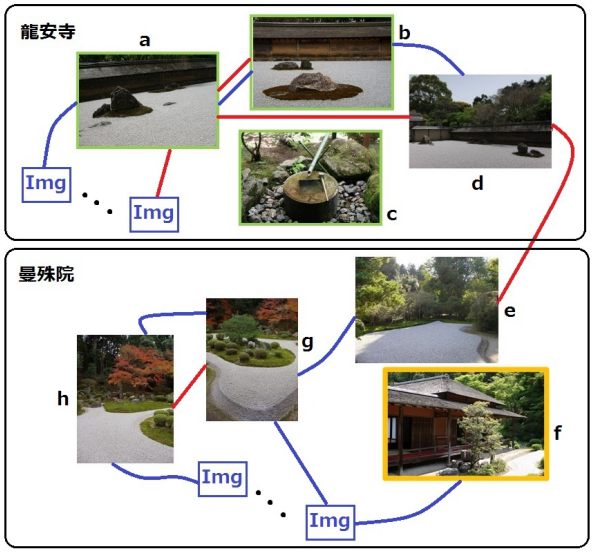

図 4　画像の類似性と隣接性に基づく提案手法の実行結果例。黒い枠で囲われた部分は、同じか近い場所の画像である事を表している。緑の枠で囲われた 3 つの画像はクエリとして与えられたものであり、ユーザは竜安寺の石庭のような場所を検索意図として持っている。赤い線は類似性が高いことを示す。青い線は隣接性が高いことを示す。

本研究の目的は、情景、環境や雰囲気が類似する場所を検索することである。図 4 において、ユーザが意図しているのは、日本の枯山水の庭園が醸し出す情景であるため、クエリとして与えられた画像のうちの 2 つは竜安寺の石庭の写真である。これらの画像に対して、他の場所の画像で、類似すると判定されたものは存在せず、すなわち、既存の検索エンジンでは、図の画像 g のように、人間が見ると類似していると判断できる場所を検索することが難しいことが分かる。竜安寺の画像の中には、石庭を別の角度からや別のフレーミングを用いて撮影したものがあり、クエリの画像はそれらと、隣接性が高いと判定される可能性がある。今回の場合、クエリに含まれる画像 b と、別の画像 d が隣接性があると判定された。画像 d には石庭といくらかの森が写っており、別の場所の画像 e が類似していると判定された。そして、画像 e と画像 g は隣接性が高いと判定された。以上のようなことを全て検討すると、従来発見できなかった、画像 e、画像 g、画像 h のような画像を発見することができる。

結果として、竜安寺の石庭をクエリとして思い浮かべた場合、最終的に、萬殊院の石庭などを発見することができた。

6. まとめと今後の課題

本研究では、ユーザが思い描く場所と情景、環境、雰囲気が類似する別の場所を検索することを目的として、場所の名前といくつかの画像が与えられたときに、別の場所で情景の類似する画像を発見する手法について提案した。

場所の類似性を考えるためには、画像の類似性と隣接性という 2 つについて共に考慮する必要があると考えた。既存の手法

を用いて、画像の類似性と隣接性について計算する手法を実装し、その結果を用いて、TextRank を基にする手法によって画像のランキングを行う手法を提案した。さらに、画像に付与されたタグの類似性を考慮したランキングについても提案した。

Flickr において公開されている、京都の観光地の画像を収集し、画像の類似性を計算する手法と、画像の隣接性を計算する手法がある程度正しく機能することを確認した。

さらに、小さなデータセットを用いて、提案手法が正しく機能することを明らかにした。

今後の課題としては、まず、もっと様々な画像特徴について、我々の手法に取り入れることを考えている。また、今回は、各種パラメータを適当に設定したが、最適な値を機械学習などで学習することを検討する。また、大きなデータセットを用いて提案手法の評価を行わなくてはならない。

7. 謝　　辞

本研究の一部は，グローバル COE 拠点形成プログラム「知識循環社会のための情報学教育研究拠点」（拠点リーダ：田中克己）によるものです．ここに記して謝意を表します．

文　　献

[1] Herbert Bay, Andreas Ess, Tinne Tuytelaars, Luc Van Gool ”SURF: Speeded Up Robust Features”, Computer Vision and Image Understanding (CVIU), Vol. 110, No. 3, pp. 346–359, 2008

[2] D. Nister and H. Stewenius, ”Scalable Recognition with a Vocabulary Tree”, Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 2161–2168, 2006

[3] M. Brown and D. G. Lowe, ”Recognising panoramas”, Proceedings of Ninth IEEE International Conference on Computer Vision, Vol. 2, pp. 1218–1225, 2003

[4] David G. Lowe, ”Distinctive image features from scale-invariant keypoints”, International Journal of Computer Vision, Vol. 60, No. 2, pp. 91–110, 2004

[5] Greg Pass, Ramin Zabih, Justin Miler, ”Comparing images using color coherence vectors”, Proceedings of the Fourth ACM international conference on Multimedia, pp. 65–73, 1996

[6] Rada Mihalcea and Paul Tarau, ”TextRank: Bringing Order into Texts”, Proceedings of EMNLP, pp. 404–411, 2004