

# ジェスチャインタフェースを用いた連想的画像検索システム

南雲尚子<sup>†</sup> 川田弘明<sup>‡</sup> 関子泰三<sup>††</sup> 清木康<sup>‡</sup>

<sup>†</sup>慶應義塾大学総合政策学部 〒252-8520 神奈川県藤沢市遠藤 5322

<sup>‡</sup>慶應義塾大学環境情報学部 〒252-8520 神奈川県藤沢市遠藤 5322

<sup>††</sup>慶應義塾大学政策・メディア研究科 〒252-8520 神奈川県藤沢市遠藤 5322

E-mail: <sup>†</sup> <sup>‡</sup> {chaco, hiroaki}@mdbl.sfc.keio.ac.jp, <sup>††</sup> <sup>‡</sup> {tz, kiyoki}@sfc.keio.ac.jp

**あらまし** 本論文では、個人のコンピュータ上に大量に存在する画像データを対象とした連想的画像検索システムの実現方式を示す。本方式は、ユーザがある場面を漠然と視覚的に記憶しているが、その内容（登場人物の詳細など）を言葉で表現できない状況が対象である。本方式は、検索の対象となる分野の代表的サンプル画像と、それに対応するジェスチャ（ユーザのポインティングデバイスのストローク）を用いることにより、ユーザが問い合わせとして与えたジェスチャに相関の強い画像を検索することを可能とする。本論文では、検索者が問い合わせとして与えたジェスチャにより代表的サンプル画像を選択するアルゴリズム（以下ストロークインタプリタとする）の実現方式を示す。選択された代表的サンプル画像と画像データベース中の画像群との類似度計算は従来のCBIRにより実現可能である。本実現方式により、ユーザが曖昧に記憶した画像とそれに関連する事象をより直感的に、効率よく検索することが可能となる。

**キーワード** CBIR, デスクトップサーチエンジン

## Associative Image Retrieval System Using Gestural Interface

Hisako NAGUMO,<sup>†</sup> Hiroaki KAWATA,<sup>‡</sup> Taizo ZUSHI<sup>††</sup> and Yasushi KIYOKI<sup>‡</sup>

<sup>†</sup> Faculty of Policy Management, Keio University 5322 Endo, Fujisawa-shi, Kanagawa, 252-8520 Japan

<sup>‡</sup> Faculty of Environmental Information, Keio University 5322 Endo, Fujisawa-shi, Kanagawa, 252-8520 Japan

<sup>††</sup> Graduate School of Media and Governance, Keio University 5322 Endo, Fujisawa-shi, Kanagawa, 252-8520 Japan

E-mail: <sup>†</sup> <sup>‡</sup> {chaco, hiroaki}@mdbl.sfc.keio.ac.jp, <sup>††</sup> <sup>‡</sup> {tz, kiyoki}@sfc.keio.ac.jp

**Abstract** In this paper, we present an associative image retrieval method for a vast personal image database. This method is intended for the situation in which the user has a visual memory of the photo s/he wants to search, but s/he can't express the query by texts. This method will allow the user to get the image correlated to the gesture stroke (stroke created with stylus) s/he inputs. We present an algorithm to get the image from sample database by the gesture stroke input by the user. We use CBIR to calculate the resemblance between the query generated with the sample images by using this algorithm and the images in a database. Users can recall the image and the information of the things more intuitively and efficiently.

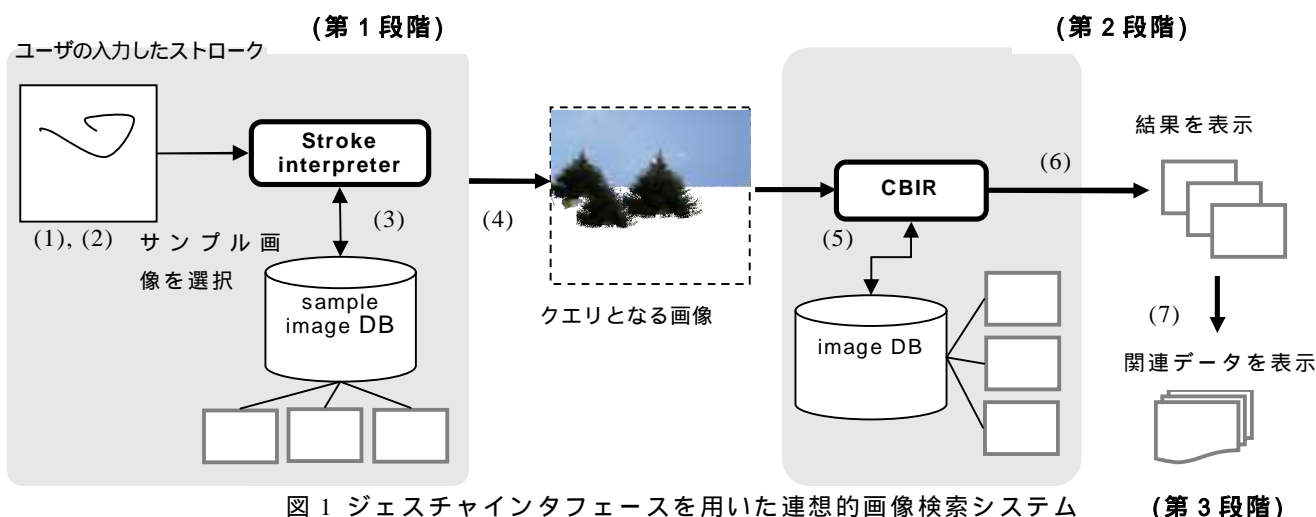
**Keyword** CBIR(Content-Based Image Retrieval), desktop search engine

### 1. はじめに

近年、デジタルカメラやカメラ付き携帯電話の普及、および、記憶デバイスの容量増加により、個人のコンピュータ上に大量の画像データが存在している。また、膨大な画像データに加え、音楽、動画、文章、メールという膨大なファイルを背景に、デスクトップサーチエンジン[9, 10]とよばれる個人のコンピュータ上のファイル群の検索ソフトウェアが実現されている。これ

らのソフトウェアを用いることにより、画像に付与されている Exif データなどのメタデータを対象にファイルの検索を行うことが可能である。そのため、あらかじめ与えられたデータのみで、効率よく画像の検索を行うことができる。

しかし、そのようなソフトウェアを使った場合においても、莫大な画像群に対しての検索は、撮影された時間に関するメタデータなどの既知のメタデータを対



象としてのみ可能であり、画像の内容に関する検索は困難である。さらに、日付などの、ファイルに関する情報を曖昧に記憶している場合においては、ユーザはさらに大量の画像を参照する必要があり、負担が大きい作業となる。また、撮影した画像データを保存する際に、大量の画像群に対してタイトルやキーワードなどのメタデータを入力していく作業自体もユーザに対する負担が大きい。

そこで、本稿では、テキストを用いない画像検索方式として用いられている Content-Based Image Retrieval(以下、CBIR)のアルゴリズムを応用し、ユーザがメタデータを曖昧に記憶している画像を検索することを可能とする連想的画像検索システムの実現方式を示す。

本稿の基本アイデアは、入力されたジェスチャによって、ジェスチャと関連した画像の選択を行い、CBIRのクエリとなる画像を、ユーザに負担をかけることなく生成する点にある。また、ジェスチャはストロークインタプリタと呼ばれる画像の選択を行うアルゴリズムにより関連した画像を選択するために利用される。このアルゴリズムは、ユーザがポインティングデバイスによって入力したジェスチャ形状と類似するデータをもつサンプル画像を取り出すものである。

本方式は次の3段階により、連想的画像検索システムを実現する。

第1段階では、検索の対象となる分野の代表的サンプル画像と、それに対応するジェスチャ(ユーザのポインティングデバイスのストローク)を用いることにより、ユーザが問い合わせとして与えたジェスチャに相関の強い画像を抽出する。このプロセスを繰り返すことにより、画像クエリを生成する(図1において(1),(2),(3)の処理)。

本方式の第2段階では、CBIRを用いて、ストロークインタプリタにより生成した画像クエリとデータベース中の画像群の色の類似度を計量する(図1において(5)の処理)。

第3段階ではCBIRの結果の画像を受け、デスクトップサーチを応用し関連事象情報を獲得する。

本方式により実現する連想的画像検索システムにおいては、クエリ生成(サンプル画像の抽出)時には、ジェスチャの形状によって画像を取り出し、その画像で生成されたクエリをCBIRによる画像の類似度計算において用いている。この理由は、ユーザは画像を形状や配置により記憶していることが多いが、形状のみでユーザの意図する画像を検索することは困難であるためである。第1段階と第2段階に分け、ジェスチャとそれに対応する画像、および、画像とデータベース群中の画像の類似度計算を行うことにより、ユーザの曖昧な視覚的記憶をクエリにし、そのクエリに関連の深い画像群をCBIRによって抽出することが可能である。

## 2. 関連研究

CBIRを用いた画像検索の過去の研究として、クエリとしてあるイメージ例をユーザに入力させる、もしくは色などの詳細について選ばせる方式(query by visual example),ユーザにスケッチをさせる方式(query by visual sketch)がある。

query by visual exampleを用いたCBIRの研究としては文献[1],[2],[3]がある。query by visual sketchを用いた研究としては領域を色で描く文献[4]や、文献[5]がある。文献[6]はデータベースのオーバービューを見て、ユーザが検索したい画像を選択し、その画像と類似する画像のランキングを行い、ランキング結果を表示する方式である。文献[7]の研究は、クエリとしては”people”, ”sailing&surfing”などの言葉を提出し、

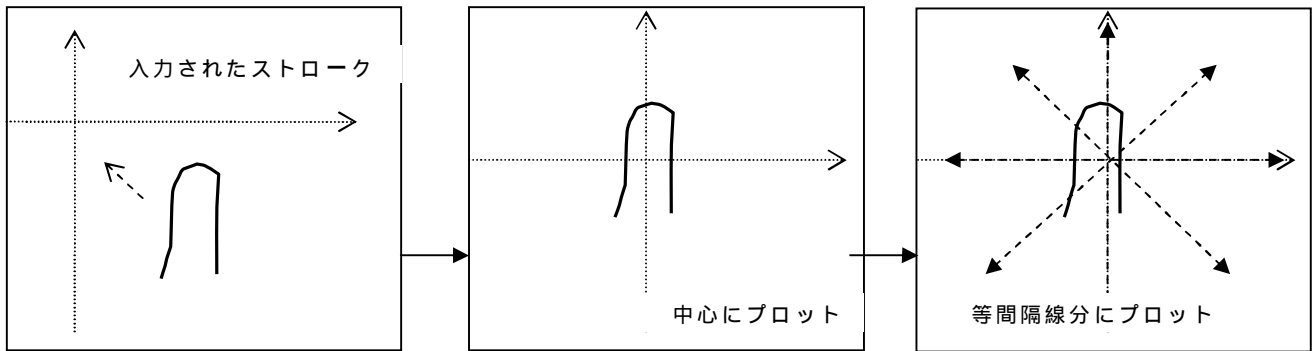


図2 ストロークインタプリタの特徴量の抽出の処理

people ならばオレンジ色の多く使われた画像を取り出し, sailing&surfing ならば青みがかった画像を取り出すというシステムを提案している.

文献[8]では, スケッチによってダイレクトに検索するツールと, 色を指定することにより検索結果の画像を表示するツール, それぞれにおける有用性を検証している. スケッチによる検索ではクエリとなる画像のコンテンツの明確な場所を記憶している場合は有効である. しかし, 記憶が曖昧である場合は色を指定することによる画像のブラウジング方式の方が有効であることが示されている. しかし, 色を指定してブラウジングするツールでは指定する色によって出てくる画像群の変化が激しくユーザに負担をかけることにもなることから, 用途とクエリ生成のツールの選択が重要であることが述べられている.

### 3. システム概要

図1に本システムの概要を示す. また, 図3にシステムの動作状態を示す.

本システムは, 次の3段階により実現される. 第1段階はユーザが漠然とした記憶をストロークとして表現(ジェスチャ)するとその記憶を具現化するサンプル画像を選択し, 抽出するプロセスである. 第2段階は第1段階で生成したクエリ画像をCBIRに入力し, そのクエリ画像に対応する画像を検索, 抽出するプロセスである. 第3段階では, 第2段階で抽出した画像に付加されたExifデータを抽出し, それをキーワードとしてデスクトップサーチに発行することにより, 画像に対応する関連事象情報を獲得する.

第1段階のストロークインタプリタは, 入力としてユーザの入力するジェスチャ(ポインティングデバイスによるストローク)があり, 出力として画像クエリがある. 第2段階のCBIRは, ストロークインタプリ

タを用いて生成された画像のクエリが入力であり, 出力として画像データベース中の画像がある. 第3段階のデスクトップサーチエンジンでは, 入力として第2段階で抽出された画像のExifデータがあり, 出力としては様々な関連事象のファイルがある.

検索過程は以下の通りである.

#### 図1における第1段階:

#### ユーザの曖昧な記憶に基づくストローク入力, および, イメージ想起プロセス

- (1) ユーザがポインティングデバイスによって, 取得したい画像の輪郭をジェスチャとして入力する.
- (2) 入力されたストローク  $P$  をベクトル  $P = \{P_1, P_2, P_3, \dots, P_n\}$  として保存する.
- (3) 保存された入力ストロークとサンプルデータのストロークデータ  $Q = \{Q_1, Q_2, Q_3, \dots, Q_m\}$  の形状の類似度を計算し, ストロークに近いサンプル画像を抽出する.
- (4) 1~3 を繰り返してクエリとなる画像を生成する.

#### 図1における第2段階:

#### 想起イメージに対応する画像データ抽出プロセス

- (5) 第1段階で生成された画像クエリ  $ImageQ$  に対し, 4.2節で後述する4ステップによる処理を行う. データベース中の画像  $ImageT$  群に対しては, 4.2節で後述する4ステップのうち, step1とstep2をあらかじめ行っておく.
- (6) CBIRにより類似度計算を行った結果の画像をランキング形式で表示する.

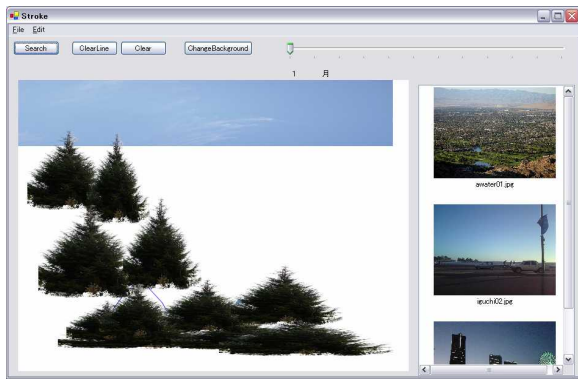


図 3 システム動作状態

図 1 における第 3 段階:

抽出された画像データ入力による関連事象獲得プロセス

- (7) 応用として、ユーザが結果の一覧から選んだ画像の Exif データからデスクトップサーチを用いて、関連するファイルを獲得する。

第 2 段階の検索結果として抽出された画像群からユーザが意図する画像が抽出されたと仮定し、その画像を選んだと仮定する。その場合に、検索結果中の選んだ画像をクリックすることにより Exif データを抽出し、MSN のデスクトップサーチに入力する。実行結果を図 4 に示す。

例えば、選んだ画像から、本システムが Exif データの 12 月 20 日を抽出し、デスクトップサーチエンジン左上のクエリ入力欄に入力する。その結果、エンジン左側の一覧に 12 月 20 日に関連するファイルが表示され、右側には選択されているインスタントメッセージのファイルの内容が表示される。

このように、ストロークインタプリタ、CBIR、デスクトップサーチエンジンの 3 つのプロセスを踏むことにより、ユーザの曖昧な情報から、意図する画像を抽

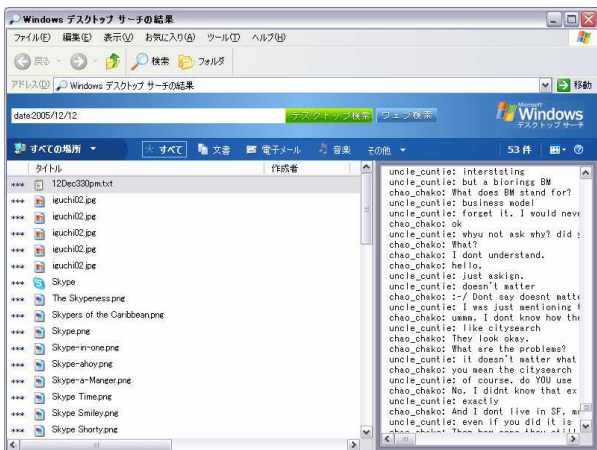


図 4 デスクトップサーチ実行結果

出し、選択された画像から関連する事象データを獲得

することが可能である。

## 4. 実現方式

本節では、ジェスチャインタフェースを用いた連想的画像検索システムの実現方式について説明する。

### 4.1 ストロークインタプリタ

3 章における第 1 段階、ユーザによって入力されたストロークデータ  $P = \{P_1, P_2, P_3, \dots, P_n\}$  と、サンプル画像データ群中の画像のストロークデータ  $Q = \{Q_1, Q_2, Q_3, \dots, Q_m\}$  との類似度計算の詳細について説明する。

サンプル画像データ群中の画像のストロークデータ  $Q = \{Q_1, Q_2, Q_3, \dots, Q_m\}$  に対しては、以下に示す 5 ステップのうち、step1 と step2 にあたる処理を、前処理としてあらかじめ行っておく。

**Step1:** ユーザが入力したストロークデータ  $P = \{P_1, P_2, P_3, \dots, P_n\}$  を座標上に等間隔に引かれた線分  $M = \{M_1, M_2, \dots, M_n\}$  の上にプロットする。

**Step2:**  $P = \{P_1, P_2, P_3, \dots, P_n\}$  と  $M = \{M_1, M_2, M_3, \dots, M_n\}$  の交差する点を  $I_g = \{I_{g1}, I_{g2}, I_{g3}, \dots, I_{gn}\}$  として保存する。

**Step3:** あらかじめ step1 と step2 の処理により保存しておいた  $Q = \{Q_1, Q_2, Q_3, \dots, Q_m\}$  と  $M = \{M_1, M_2, M_3, \dots, M_n\}$  の交点である  $I_s = \{I_{s1}, I_{s2}, I_{s3}, \dots, I_{sn}\}$  と  $I_g = \{I_{g1}, I_{g2}, I_{g3}, \dots, I_{gn}\}$  の距離 similarity を計算する。次式によって表される。

$$\text{similarity}(a, b) = \sqrt{(b_x - a_x)^2 + (b_y - a_y)^2}$$

**Step4:** step3 で計算された similarity を足し合わせたものを distance とする。distance は次式によって表される。

$$\text{distance}(n) = \sum_{i=1}^n \text{similarity}(I_{s_i}, I_{g_i})$$

**step5:** 最小の distance を値としてもつサンプル画像が、ユーザの入力ストロークに類似する画像として採用される。

### 4.2 CBIR

3 章における第 2 段階、入力されるクエリ画像  $ImageQ$  とデータベース中の画像群のそれぞれの画像  $ImageT$  に対する処理について説明する。この処理は一般的に用いられている画像のテンプレートマッチング方式である。一般的な CBIR としては、画像の色、形状、模様という様々な要素を利用して類似度計算をする方式がある。

本稿では、 $ImageQ$  と  $ImageT$  群間の、RGB の色のユークリッド距離を求めることで、画像の類似度を計量する方式を応用する。 $ImageQ$  において、ユーザに何も

描かれなかった座標は白ではなく、何も描かれなかった場所として認識される。

*ImageT*群に対しては、以下に示す4ステップのうち、step1 と step2 を前処理としてあらかじめ行っておく。*ImageQ* に対しては、ユーザが検索を行う際に、step1、step2 を含めて全行程の処理を行う。

以下に4ステップを示す。

**Step1:** *ImageT(Q)*をグリッドにより  $M \times N$  分割する(図5)。分割された各グリッドはRGBの要素を持つ。

**Step2:** *ImageT*における、各グリッドのRGB要素の平均値を計算し、*AverageGrid<sub>TMN</sub>*として保存しておく。

**Step3:** 入力された *ImageQ* に対して、step1、step2 で行ったように、各グリッドのRGB要素の平均値を計算し、*AverageGrid<sub>QMN</sub>*とする。

**Step4:** *AverageGrid<sub>TMN</sub>*と *AverageGrid<sub>QMN</sub>*の距離を  $M \times N$  個分計算し、 $M \times N$  個分足し合わせたものを *R* とする。*R*の最小の値をもつ画像を類似する画像とする。

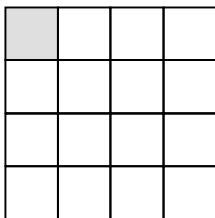


図5  $M \times N$  分割した *Image*

## 5 実験

本実験では、本方式によって実現した実験システムを用いて、ストロークデータからサンプル画像を用いてクエリを生成し、実際に類似する対象画像データを検索することにより、連想的画像検索システムの実現可能性を検証する。

実験は次の5つのステップで行われる。

**Step1:** ストロークインタプリタで使用するサンプル画像を設定する。

ここでは、「木」、「空」、「ビル」、「芝生」の四種類のサンプル画像を設定した。

**Step2:** サンプル画像に対して、ストロークデータをあらかじめ付与しておく。

ここでは、木には先のとがった山のような輪郭のデータ、空には大きな横長の長方形のデータ、ビルは小さな四角形のデータ、芝生は小さな山を繰り返したような形状のデータを付与した。

**Step3:** ユーザが頭に思い描いている曖昧なイメージを設定する。

イメージ 1: 「背景が青空で、左の方に木がある。」

イメージ 2: 「背景が青空で、ビルや建物がたくさんある。」

イメージ 3: 「背景が青空で、木や緑がたくさんある。」

イメージ 4: 「左の方に木がある。」

イメージ 5: 「ビルや建物がたくさんある。」

イメージ 6: 「木や緑がたくさんある。」

**Step4:** 各ユーザはそれぞれのイメージを頭に思い描いていると設定し、ストロークインタプリタを用いてクエリ生成を行う。

**Step5:** 生成されたクエリをCBIRに発行し、類似度計算を行い、検索結果画像を出力する。

### 5.1 実験方法

この節で示す実験方法は、Step3の曖昧なイメージ1を例としている。同様の方法によって、イメージ2、イメージ3、イメージ4、イメージ5、イメージ6に対してもクエリ生成を行い、実験を行うものとする。

本実験では、ストロークインタプリタにおいて等間

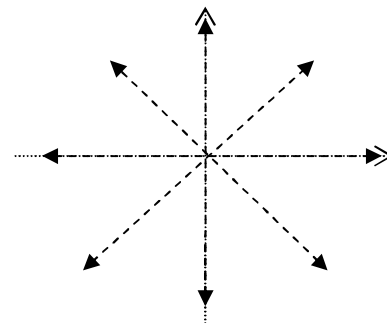


図6 8方向等間隔線

隔線分を、放射線状に8方向に引く。ストロークを入力する座標の端から端まで引く。図6に示す。

実験方法を次に示す。

- (1) マウスの入力により、座標上の上部に大きな長方形を描く。するとストロークインタプリタにより、サンプル画像の空が抽出される。描画の大きさや位置などが、ユーザの思い描くイメージに合致するように何度か繰り返す。
- (2) 同様にマウスの入力により、座標の左側に木のサンプルイメージを抽出する。

- (3) 検索ボタンを押すことにより，生成したクエリ 1(query1)を，CBIR に発行する．
  - (4) query1 と画像データベース群との類似度計算を行い，抽出された画像データのうち，類似度上位 3 位までを表示する．
- Step3 で設定した曖昧なイメージ 1, 2, 3 から生成されたクエリ群 (query1, query2, query3) を図 7 に示す．

### 5.1.2 実験結果

実験結果として得られた検索結果を表に示す． query1, query2, query3, を表 1 に, query4, query5, query6 を表 2 に示す．

表 1 実験結果画像



















	1位	2位	3位
Query1	I01.jpg 	I04.jpg 	A01.jpg 
Query2	I04.jpg 	I01.jpg 	I09.jpg 
Query3	A01.jpg 	A04.jpg 	A06.jpg 

表 2 実験結果画像

	1位	2位	3位
Query4	DSC00368.jpg 	A05.jpg 	A02.jpg 
Query5	I00.jpg 	A05.jpg 	A02.jpg 
Query6	DSC00382.jpg 	A02.jpg 	I08.jpg 

曖昧なイメージ 1: 「背景が青空で，左の方に木がある .」 から生成された query1 に対する検索結果を見ると，1 位(I01.jpg)，2 位(I04.jpg)，3 位(A01.jpg)ともに，木がある画像ではない．生成されたクエリ画像の左の木のボリュームが少なく，類似時計量を行った結果，空に対応する部分が多かったためと考えられる．

曖昧なイメージ 2: 「背景が青空で，ビルや建物がたくさんある .」 から生成された query2 に対する検索結果を見ると，1 位に雪の画像(I04.jpg)が出てきてしま

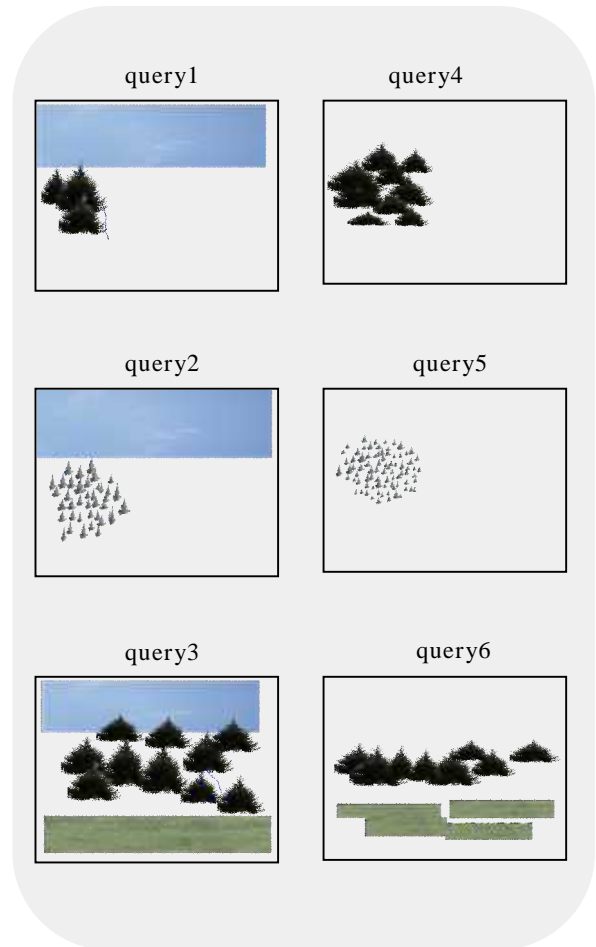


図 7 生成したクエリ

うものの，検索結果 2 位(I01.jpg)はユーザの思い描く曖昧なイメージに忠実な結果になったといえる．ランキング形式の結果表示であるため，結果からクエリに最も近い，2 番目の画像を探し出すことができる．検索結果 3 位(I09.jpg)は湖の画像ではあるが，query2 に示すように，湖の場所はクエリ生成時には何も描かれていない部分であるため，このような結果になった．

曖昧なイメージ 3: 「背景が青空で，木や緑がたくさんある .」 から生成された query3 に対する検索結果を見ると，1 位(A01.jpg)は木の画像ではないが，緑の部分において類似度が高いという結果となっている．2 位(A04.jpg)が，ユーザのイメージに最も近い画像であるといえる．

曖昧なイメージ 4: 「左の方に木がある .」 から生成された query4 に対する結果を見ると，1 位(DSC00368)は，木が結果画像における黒っぽい部分に対応してしまっていることがわかる．2 位(A05.jpg)は左に大きな木がある画像となっており，ユーザのイメージに一番近い画像であるといえる．

曖昧なイメージ 5: 「ビルや建物がたくさんある .」 から生成された query5 に対する結果を見ると，1 位

(I00.jpg)はコンクリートの色がビルと類似しているためであると考えられる。2位(A05.jpg), 3位(A02.jpg)は, ユーザの思い描いたイメージが異なるにも関わらず, query4の結果と同じである。

曖昧なイメージ6: 「木や緑がたくさんある。」から生成されたquery6の結果を見ると, 1位(DSC00382)は, 木の色こそ違うものの芝生に木がある風景はユーザのイメージに近いといえる。また, 2位(A02.jpg)も草と木がある画像である。さらに, 3位(08.jpg)は木の種類が違うものの, 緑の土地に木がある画像である。

## 6 考察

5.1.2節に示したように, query1, query2のようにクエリ画像に描かれるボリュームが少ないことにより, 類似度計算をおこなう要素が少なくなることで, ユーザのイメージに見合う結果が出ないこともある。

また, query2に対する3位の結果(I09.jpg)や, query3に対する3位の結果(A06.jpg)に示したように, サンプルイメージにおける深い緑の木の色が, 湖などの黒っぽい色と類似しているという結果になってしまう点は改良の余地がある。

query4に対する2位の結果(A05.jpg)や, query6に対する1位, 2位, 3位の結果(DSC00382.jpg, A02.jpg, I08.jpg)などは, クエリに描かれた部分に対しては忠実な結果をだしているといえる。描かれていない部分に対しては, 類似度計算を行わないため, 描かないことでクエリを生成する要素の一つになるといえる。

また, 同じようなイメージを思い描きつつクエリ生成を行ったとしても, クエリ生成はユーザの入力したストロークに大きく依存するため, 全く同じクエリを生成することはできない。そのため, サンプル画像の描かれたボリュームや, 場所がCBIRの結果に大幅に影響する。そのため, 同じイメージを思い描きつつ生成されたクエリの, CBIRの類似度計算結果が異なるものになることがある。

全ての結果は, 描かれたクエリに対してのCBIRの結果としては, 妥当な結果であると言える。前提として, ユーザが獲得したい画像はもともと曖昧な情報であるため, 曖昧な情報をもつクエリに対して, 提案し抽出する画像としては十分な結果であるといえる。

5節において実験を行った結果, ストロークインタプリタとCBIRの類似度計算を用いることにより, ユーザの曖昧な情報から, 意図する画像を抽出することが可能である例を示した。

## 7 まとめ

本稿では, 第1段階: ユーザの曖昧な記憶に基づくストローク入力, および, イメージ想起プロセス, 第2段階: 想起イメージに対応する画像データ抽出プロセス, 第3段階: 抽出された画像データ入力による関

連事象獲得プロセスという3つのプロセスを踏むことにより, CBIRのアルゴリズムを応用し, ユーザがメタデータを曖昧に記憶している画像, および関連事象を, テキストを用いずに検索することを可能とする連想的画像検索システムの実現方式を示した。

5節の実験においては, 連想的画像検索システムが機能し, 上記のように3段階により, 少ない曖昧な情報をもつクエリから, それに関連する画像の抽出が可能である例を示した。さらに, 抽出された画像から任意の画像を選択し, 選択された画像の日付を抽出しデスクトップサーチエンジンのクエリとした。その結果, 同日に保存されたインスタントメッセージのファイルを抽出することにより, 連想的に関連事象を獲得することが可能であるケースを示した。

今後は, 様々なCBIRの適用, サンプル画像とそれに対応するジェスチャの自動生成方式を行うことを考えている。具体的には, ユーザの入力するジェスチャとサンプル画像に対応するジェスチャとのマッチングをインクリメンタルに行う方法の導入によって, サンプル画像群のスケラビリティへの対応を行っていくことを考えている。

## 8 謝辞

本稿を執筆するにあたって助言を頂いた, 慶應義塾大学政策・メディア研究科吉田尚史氏, 佐々木史織氏, 鷹野孝典氏に感謝する。

## 文 献

- [1] W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, P. Yanker, C. Faloutsos, G. Taubin, The QBIC Project: Querying Images By Content Using Color, Texture, and Shape,
- [2] K.Hirata, T. Kato, Query by Visual Example-Content based Image Retrieval-
- [3] H. Tamura, S. Mori, T. Yamawaki, Textural Features Corresponding to Visual Perception
- [4] J. Smith, S. Chang, Querying by Color Regions using the VisualSEEK Content-Based Visual Query System
- [5] E. Di Sciascio, G. Mingolla, M. Mongiello, Content-based Image Retrieval over the Web using Query by Sketch and Relevance Feedback
- [6] J. Vendrig, M. Worring, A.W.M. Smeulders, Filter Image Browsing Exploiting Interaction in Image Retrieval
- [7] C. Carson, V. E. Ogle, Storage and Retrieval of Feature Data for a Very Large Online Image Collection, Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, Vol. 19, No. 4, December, 1996
- [8] S. McDonald, J. Tait, Search Strategies in Content-Based Image Retrieval, SIGIR, Toronto, Canada, 2003
- [9] <http://desktop.msn.co.jp>
- [10] <http://desktop.google.com/en/>