

目次

第1章 はじめに	5
第2章 既存の物体認識技術	8
2.1 モデルベース法	8
2.2 ビューアベース法	9
2.2.1 SIFTアルゴリズム	10
第3章 提案システムの概要	15
第4章 システムの設計	17
4.1 物体の管理	17
4.1.1 物体登録	17
4.1.2 物体の変更	18
4.2 物体の探索	18
4.2.1 カメラの動画をキャプチャ	19
4.2.2 画像解析とマッチング	20
4.2.3 ユーザに通知	20
4.3 インターフェースの設計	20
第5章 実装と結果	22
5.1 実装の概要	22
5.2 システムのインターフェース	22
5.2.1 システムを起動した画面	22
5.2.2 物体の追加	23
5.2.3 物体の変更・削除	24
5.2.4 物体の探索	25
5.3 物体の管理	27

5.4	物体の探索.....	32
5.4.1	カメラから動画の取得.....	32
5.4.2	物体の探索	32
5.5	結果	35
5.6	まとめ.....	38
第6章	関連研究.....	39
第7章	終わりに.....	42

目次

図 1-1 「どこいっ太郎」の外観.....	6
図 2-1 画像の特徴点検出方法.....	10
図 2-2 ヒストグラム作成の流れ.....	13
図 2-3 ブロックごとの特徴量記述	14
図 3-1 システムの概要	16
図 4-1 探索機能の流れ	19
図 4-2 システムの画面の設計.....	21
図 5-1 システムの起動画面	23
図 5-2 物体の追加画面	24
図 5-3 物体情報の変更画面	25
図 5-4 物体削除の確認フォーム.....	25
図 5-5 探し物の画面.....	26
図 5-6 探し結果の画面	27
図 5-7 物体のクラス	28
図 5-8 物体追加の流れ	30
図 5-9 物体の変更・削除の流れ.....	31
図 5-10 Webカメラの取得.....	32
図 5-11 物体の探索の流れ.....	33
図 5-12 マッチングの結果.....	35
図 5-13 マッチングの結果.....	36
図 5-14 マッチングの結果を表示.....	37
図 6-1 Autostitch で生成したモザイク画像.....	39

図 6-2	合成画像認識.....	40
図 6-3	道路交通標識の認識例.....	41

第1章

はじめに

人間の感覚は機械と違って時間と共に性能が変化する。たとえば、知っている曲であっても周りの雑音に埋てしまったときには耳で認識できないことがありえるが、機械により音声を分析すれば簡単に認識できる。

一日コンピュータを使い目が疲れた時や多忙の時には、物体発見能力が衰えてしまう。特に散らかった状態での探し物では、物体の一部が隠れてしまっていると目の前にあっても発見できないことも多い。このような状況下では探し物支援システムが有益だと考える。

探し物支援システムについてはいくつかの提案や商品がある。最近、「どこいっ太郎」[1]という商品が発売された(図1-1)。鍵や携帯、財布など大事なものに子機を取り付けておき、親機のボタンを押すだけで、その番号に対応した子機が「ピピ..」と鳴り、大事なモノの居場所を知らせるものである。しかし、子機のサイズが大きく、取り付けられる数も少ないため使いにくい。

Raymondらは微小な無線チップのタグを管理したい物につけるFLO (Frequently Lost Objects)システムを提案した[2]。そのシステムはユビキタス技術を使うため、設置コストが高くなる。

高井らはユーザが光または音を認識できる範囲外に居た場合であっても、その存在位置を知らせることのできる遺失物検知システムを提案した[3]。



[[1]の参考文献から引用]

図 1-1 「どこいっ太郎」の外観

現在までの探し物支援システムは、すべて無線技術や Bluetooth 等の近距離無線通信を利用し、管理したい物に子機やタグを付けている。子機のサイズが大きくて取り付けられる数が少ない問題や、設置コストが高いという問題が存在している。

本研究では Web カメラ等の低価格カメラを使い、画像認識技術で、散らかった状態から物の探索を支援するシステムを提案する。物を探したいとき、カメラを回して周辺画像を入力し、その画像とあらかじめ登録してあった写真と比較することで、指定した物を見つける。

提案システムではパソコンで実装することを前提としているが、将来、カメラ付きの携帯電話でも紛失したものを探索できる可能性がある。本研究では乱雑な状態で一部が見えているような目では発見しにくい状況でも利用可能な探索支援システムを提案する。ただし、全部隠れた物体はシステム

の探索支援対象にはならない。

本論文では、第 2 章で既存の物体認識技術について述べる。特に、物体認識によく使われている SIFT アルゴリズムを説明する。第 3 章はシステムの概要を述べる。第 4 章でシステムの設計、第 5 章で実装と結果を示す。第 6 章は関連技術を述べる。最後の 7 章では、本論文をまとめる。

第2章

既存の物体認識技術

本研究は画像認識技術を用いてシステムを作成する。そこで、まず本章では、既存の物体認識方法について概観する。

物体認識方法についてモデルベース法とビューアベース（またはアピランスベース）法に大別できる。前者はまず画像から検出された特徴点の2次元座標から3次元座標を幾何学計算する。そしてこれにより3次元モデルを推定する。特徴点が正確に検出できれば、精度的には後者に比べて優れているが、実環境では必要な特徴点が十分に求まらない場合が多い。それに対して、後者は物体の見え方に基づいて、画像パターン認識技術で物体の有無を判断する。まず、いろいろ視点や照明条件における物体の見え方を記憶しておく。そして、未知パターンとこれを照合することで3次元物体認識を行う。次に、それぞれの方法について述べる。

2.1 モデルベース法

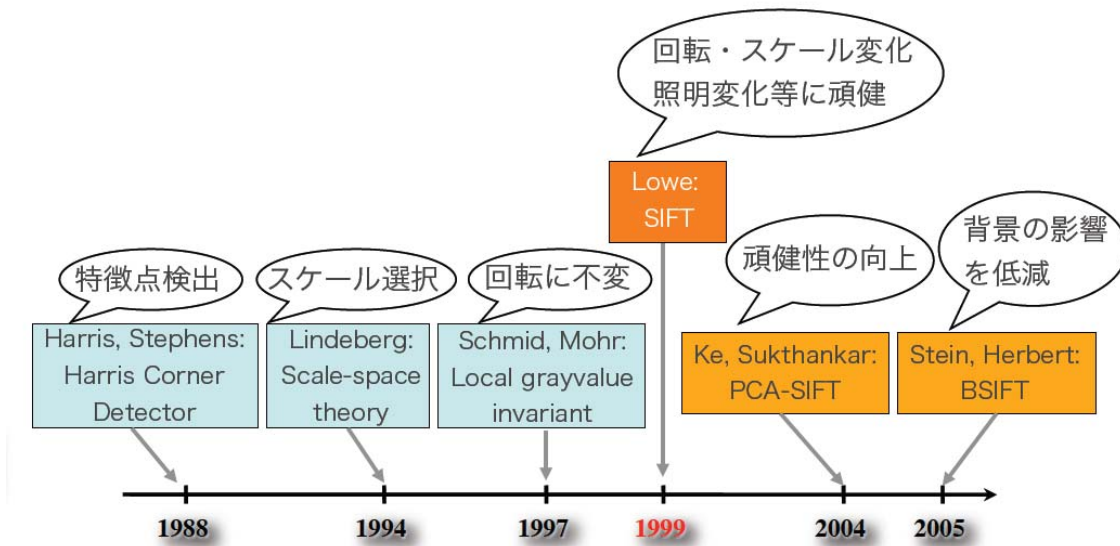
モデルベース物体認識は、知能ロボットの視覚システムの構築などにおいて重要視され、これまでに数多くの研究が行なわれてきた。その中ではステレオ視覚に基づく方法や、**Shape from Motion** に基づく方法が代表である。しかし、物体を外側から見るので物体内部の情報はそれほど重要ではなく、表面の情報のみが意味を持つ。また、表面の情報に限定しても、3次元のモデルベースを作成するのは計算コストのかかる処理である。

2.2 ビューアベース法

ビューアベースの認識法の本質は画像パターン認識そのものである。それは3次元の対象物を複数の方向から観察画像集合で認識する。ひとつの画像から特徴量を抽出し、それを記述することで物体の有無を判断する。特徴量を抽出する方法はいろいろ考えられる。

エッジ抽出に基づく方法は90年代に広く研究され、最近 K. Mikolajczyk[5]らがスケール変更に対処できる方法を提案した。他には、グラジエントベース画像の局所特徴量を抽出する方法もある。それには対象画像から特徴点を検出する必要がある。Harris らは、1988年に特徴点としてコーナーを検出する手法(Harris CornerDetector)を提案した[6]。Lindeberg は1994年にスケールスペースを用い、画像の構造を解析し、blobの検出と自動スケール選択を行う手法を提案した[7]。また、Schmid らは Harris corner detector によって検出された特徴点に対し、その点の画素値や微分値から算出した値を特徴量とし、画像の回転に頑健な局所特徴量を記述した[8]。これにより、回転変化が生じても画像間のマッチングや認識を行うことが可能となった。しかし、Schmid らの手法に用いられている Harris corner detector は、画像のスケール変化に敏感であるため、拡大・縮小等の異なるスケールの画像間ではマッチングが困難である。Lowe は Schmid らの局所領域の特徴量記述という考えを拡張し、スケールスペースを用いることで画像のスケール変化や回転に不変な特徴量を記述する Scale-Invariant Feature Transform(SIFT)を提案した[9]。最近、PCA(Principal Component Analysis)を用いて勾配情報を部分空間へ射影しマッチング精度を向上させる PCA-SIFT や、SIFT の特徴量記述時における背景の影響を軽減する BSIFT 等の SIFT を拡張した手法が提案され

ている。図 2-1 は以上の流れを示している。



[[11]の参考文献から引用]

図 2-1 画像の特徴点検出方法

その中でも SIFT は回転・スケール変化、照合変化に対処でき、計算コストも高くないため、本システムは SIFT を使うことにした。次節では、SIFT アルゴリズムについて簡単に説明する。具体的には、参考文献の[10]と[11]を参照のことができる。

2.2.1 SIFTアルゴリズム

SIFT アルゴリズムは特徴点(またはキーポイントと呼ぶ)の検出 (detection)と特徴量の記述(description)の 2 段階がある。検出処理は①と②のステップで、特徴量の記述段階は③と④のステップで構成される。

① スケールとキーポイント検出

このステップでは特徴点の発生する可能性がある場所とスケールを確定する。特徴点のスケール探索をするため、ガウス関数が有効であること

は Koenderink (1984) と Lindeberg (1994) が報告していた。しかし、Lindeberg が利用した LoG (Laplacian of Gaussian) は計算コストが高いという問題があった。それを改良するために、Lowe は DoG (Difference of Gaussian) という関数を用いる方法を提案した。

ガウス関数と平滑化画像は次に示す

$$G(u, v, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(u^2+v^2)/2\sigma^2}$$

$$L(u, v, \sigma) = G(u, v, \sigma) * I(u, v)$$

その中に、 σ はスケールのガウス関数と $I(u, v)$ は入力画像である。

DoG は次のように表される。

$$D_i(u, v, \sigma) = L(u, v, \sigma_{i+1}) - L(u, v, \sigma_i)$$

画像の注目点が近傍より DoG の値が大きい小さいならば、その位置はキーポイントと考える。

② キーポイントのローカライズ

①で検出されたキーポイント候補点の中には、DoG 出力値が小さい点 (low contrast) やエッジ上の点が含まれている。これらの点はノイズや開口問題に影響を受け易いという問題がある。そこで、キーポイント候補点の中から、主曲率とコントラストにより安定したキーポイントに絞り込む。さらに、キーポイントのサブピクセル推定により位置とスケールを算出する。

③ オリエンテーションの算出

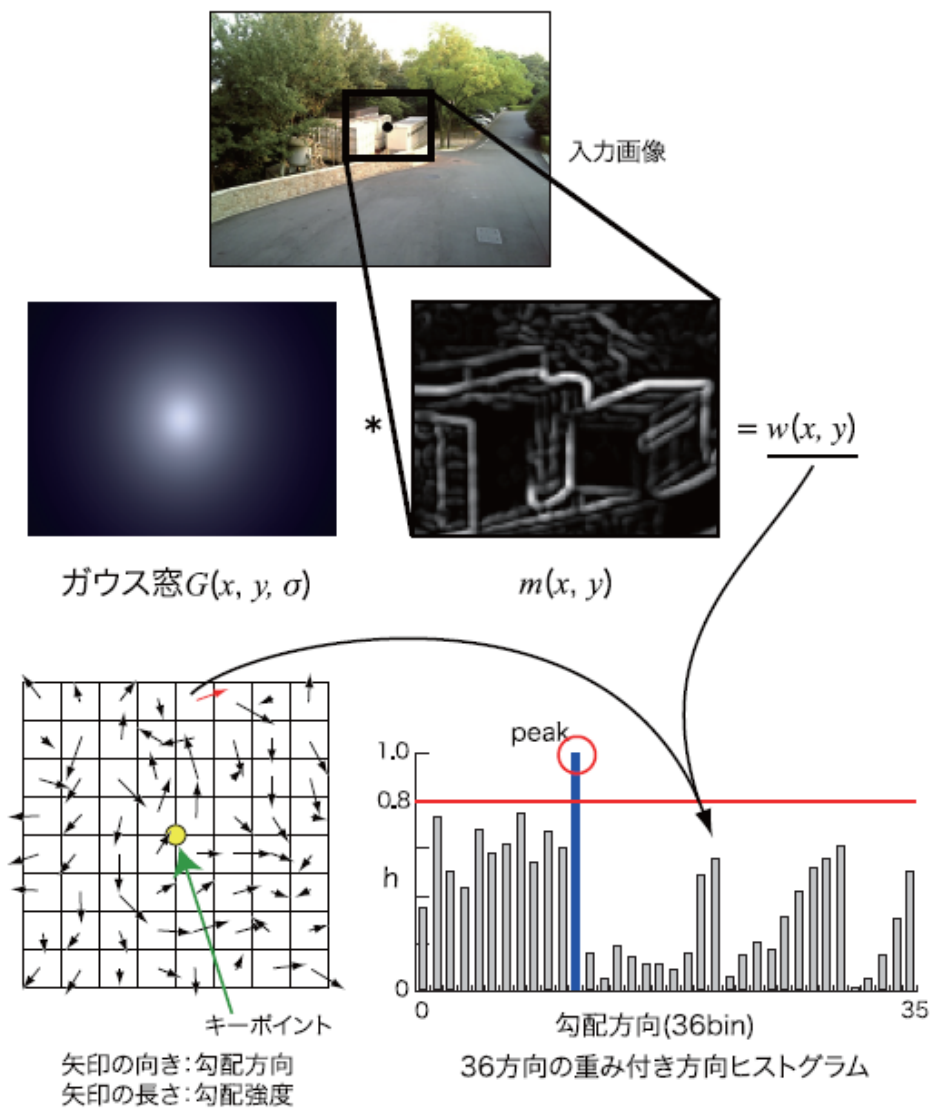
検出したキーポイントに対して、次に第 2 段階の処理である特徴量の記述を行う。まず、検出された各キーポイントのオリエンテーションを求める。オリエンテーションはキーポイントにおける方向を表す。特徴量記述の際にオリエンテーションにより向きの正規化を行うことで、回転に不

変となる。キーポイントのオリエンテーションを求めるには、まずキーポイントが検出された平滑化画像の勾配 $L(u,v)$ 、強度 $m(u,v)$ と勾配方向 $\theta(u,v)$ を以下の式により求める。

$$m(u,v) = \sqrt{(L(u+1,v) - L(u-1,v))^2 + (L(u,v+1) - L(u,v-1))^2}$$

$$\theta(u,v) = \tan^{-1}((L(u,v+1) - L(u,v-1)) / (L(u+1,v) - L(u-1,v)))$$

次は勾配情報から勾配方向ヒストグラムを作成する。全方向の360度を36方向に離散化しヒストグラムを計算する。この36方向のヒストグラムの最大値から80%以上となるピークをキーポイントのオリエンテーションとして割り当てる(図2-2)。



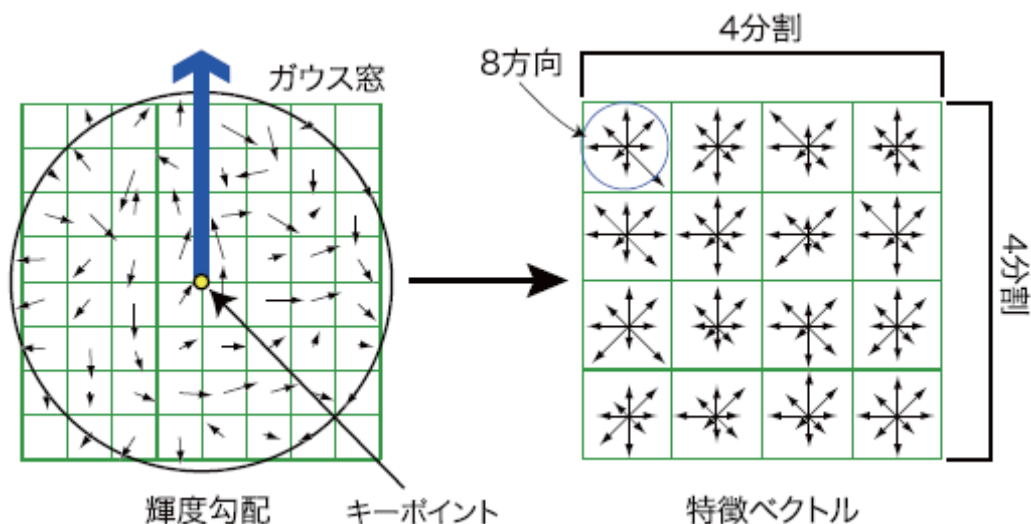
[[11]の参考文献から引用]

図 2-2 ヒストグラム作成の流れ

④ 特徴量の記述

次に③で求めたオリエンテーションに基づいてキーポイントの特徴量を記述することを行う。特徴量の記述にはキーポイント周辺領域の持つ勾配情報を用いる。使用する勾配情報は、キーポイントを中心とし、そのキーポイントが持つスケールを半径とした円領域内から求める。周辺領域の一边を 4 ブロックの計 16 ブロックに分割し、図 2-3 に示すようにブロッ

クごとに 8 方向(45 度ずつ)の勾配方向ヒストグラムを作成する。この勾配方向ヒストグラムは、キーポイントのオリエンテーションを算出したときに作成したヒストグラムと同様の手法で求める。



[[11]の参考文献から引用]

図 2-3 ブロックごとの特徴量記述

以上で SIFT アルゴリズムを説明した。SIFT では物体の色が認識できないが、スケール変化、回轉變化や輝度変化に対処できるため、SIFT を提案システムに利用することにした。次に、提案システムの概要について述べる。

第3章

提案システムの概要

本研究では低価格のカメラで探し物支援システムを提案する。最近人々がコンピュータを持っていることは普通になってきた。それと安いカメラ (Web カメラなど) があると本システムを使うことができる。

システムの概要は図 3-1 に示した。まず、財布や時計などの大切なものを写真で登録しておく。物体を認識するために、一つの物体に対して写真を何枚か登録する必要がある。

物体を探したいとき、その物体を選択する。また、コンピュータに接続したカメラを周辺に回すことで指定された物体を探す。物体が見つかったなら、ユーザに知らせる。指定物体の候補が見つかった場合、システムはその候補を示す。ユーザはそれが指定物体か確認することで探索を行う。さらに、物体の追加や指定された物体の情報を変更することができる。

本システムは人間の支援をすることが目的である。そのため、探している物体の特徴に一致する物を見つけるとユーザに報告する。その後、ユーザの判断のより、探索をやめるか、続けるか選ぶことができる。物体を登録するときには、物体の一部分の写真を登録することもできる。現在、登録物体の写真は 6 枚以下である。物体の一部を発見できるため、固体の物体だけではなく、形が変化する物体や柔らかい物体も発見できる可能性がある。

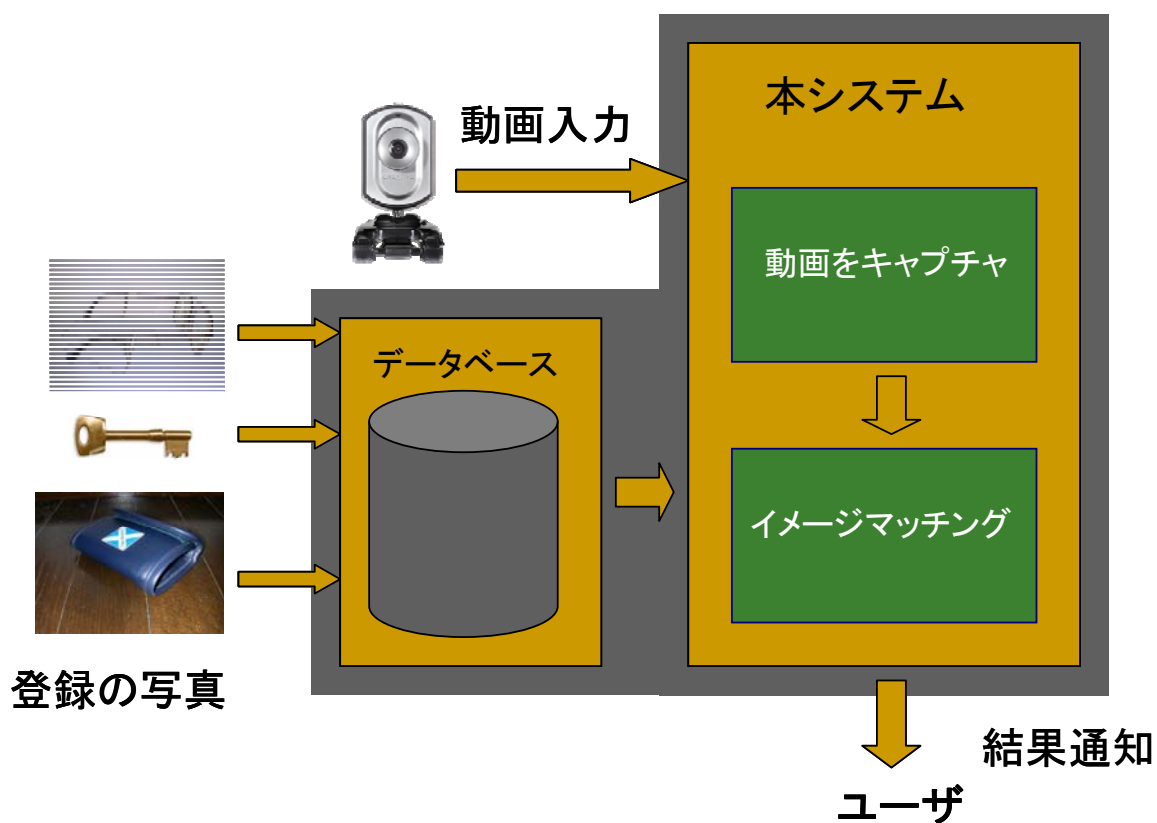


図 3-1 システムの概要

第4章

システムの設計

この章では、システムの設計について述べる。

本システムは物体の追加、物体の変更、また物体の探索の3機能がある。物体の追加と変更については4.1節で述べる。4.2節では、物体の探索について説明する。最後にシステムのインターフェースを設計する。

4.1 物体の管理

この節では、物体を操作することや物体の情報を管理することについて述べる。システムを使う前に、物体登録する必要がある。システムを使っているとき、ユーザが物体の情報を変更したりや物体を追加したりすることができる。そのため、物体の情報を管理する方法も考える必要がある。

4.1.1 物体登録

物体を登録する時、物体の名前と各写真がユーザに提供される。物体登録の目的は物体の特徴を記憶することである。物体によって登録写真の枚数が違う。実際、物体の違う方向からの撮影写真は多ければ多いほど認識率が高くなるが認識時間もかかる。通常、一つの物体に対して、6方向から撮影した6枚の写真により物体の特徴がすべて撮られる。実験速度の関係で、本システムは6枚を上限にした。また、対称性のある物体や薄い物体の場合、写真を6枚まで登録する必要がない。さらに、物体のある面の特徴が少ない場合、物体の一部(その面の特徴部分だけ)の写真を登録することもできる。

そうすると、画像の特徴量が減少し、特徴マッチングのステップが速くなる。システムは登録する物体に対して以下の情報を持つ。

- `id` : その物体の入力番号。
- `name` : 物体の名前
- `imgNum` : 物体のイメージの数
- `imgList` : 物体のイメージのパスのリスト

物体の名前と写真のリストがユーザに提供される。それによって、物体の `id` と `imgNum`、`imgList` がシステムに作成される。

全ての物体情報はファイルに書かれる。システムが起動されると、データがファイルから配列に読み込まれ、操作される。システムを終了する時や、物体を変更、追加した時、データが行列からファイルに書かれ保存される。本システムは数千個の物体を管理することができる。

4.1.2 物体の変更

システムを実行したとき、ユーザは物体の情報、あるいはシステムのデータベースを変更することができる。データベースの操作は新たな物体を追加、存在している物体を削除、また存在している物体の情報を変更することである。このとき、物体の情報を持っている配列を変更するだけでなく、物体の情報を持っているファイルも変更する。そうすることによって、突然の電源切断やシステムダウンからデータベースを守ることができる。

そうすると、万が一システムが動いているとき、電源がきったりするとなっても、データベースが安全である。

4.2 物体の探索

物体の探索機能の流れは図 4-1 に示した。まず、ユーザは探したい物を

指定する。ユーザがカメラの撮影の動画を取得し、それをキャプチャし、指定された物体の写真とマッチングする。

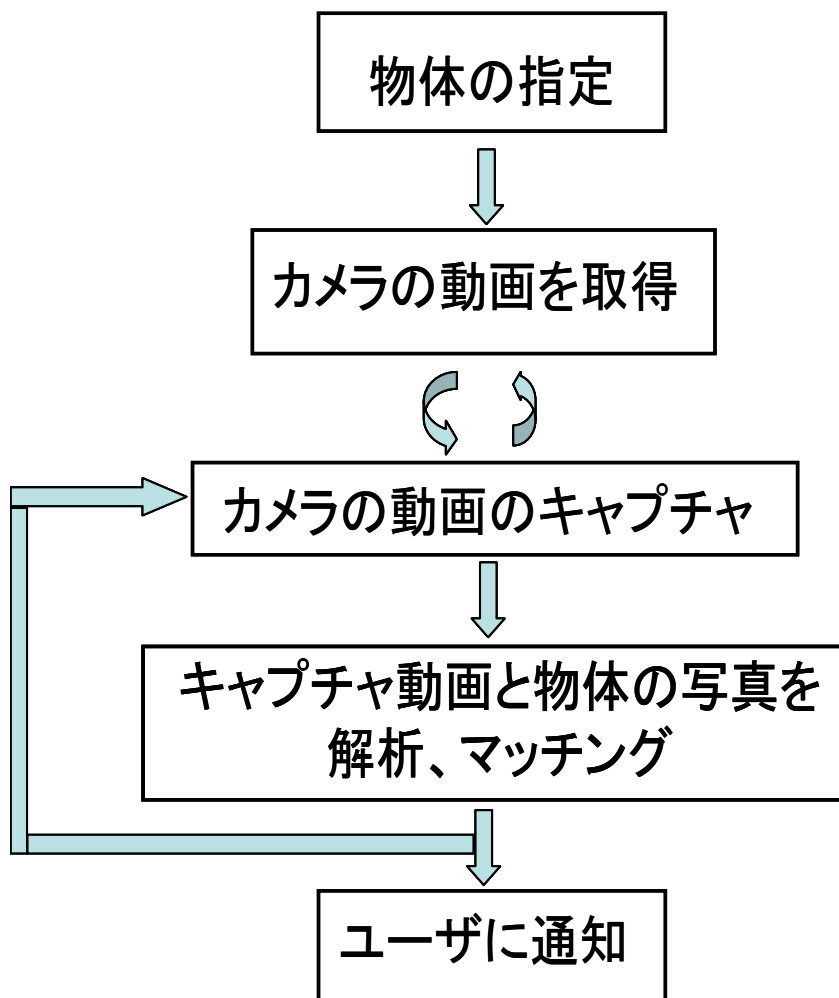


図 4-1 探索機能の流れ

4.2.1 カメラの動画をキャプチャ

カメラの動画をキャプチャする方法として、時間単位でキャプチャするか、前にキャプチャした画像を処理してから次の画像をキャプチャするかという選択がある。物体によって画像処理時間が違うので、キャプチャした画

像を比較してから、次の画像をキャプチャする方法にした。

4.2.2 画像解析とマッチング

この段階はキャプチャした画像と指定された物体の写真とを比較する。写真マッチング技術について、それぞれの画像の特徴量抽出方法や、エッジ抽出に基づく方法[5]、領域分割に基づく方法[6]などいろいろが存在する。第2章で述べた SIFT は複数の特徴領域を検出することで、物体の視点変化、輝度変化や、スケール変化または隠れに対処できるため、シーンや物体認識に有効な方法といわれている。そのため、本研究では SIFT を利用することにした。

カメラ動画からキャプチャした画像と指定された物体の各写真を SIFT 特徴抽出してから、マッチングする。

4.2.3 ユーザに通知

物体が見つかった場合、その辺を強調し、ユーザに通知する。

ユーザは確認してから、探索を終える。探している物体ではない場合はさらに探索を続ける。

4.3 インターフェースの設計

システムのインターフェースは図 4-2 のように設計した。

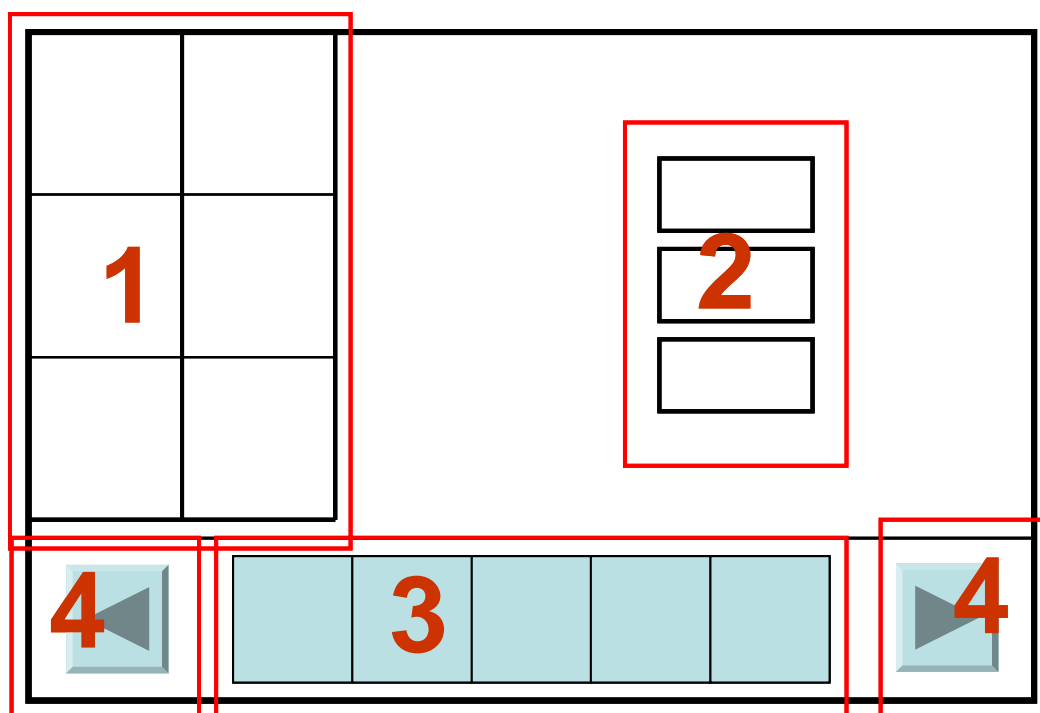


図 4-2 システムの画面の設計

その中、**1**部は指定した物体の各写真(現在 6 枚まで)、**2**部は物体追加、**1**に示している物体の変更・削除と物体探索の 3 機能のボタンを設計した。**3**部は格物体の一番目の写真、**4**部は物体の一覧の変更ボタンである。物体を指定したいとき、**3**部での物体の代表写真を選択する。

第5章

実装と結果

この章では、システムの実装について述べる。また、実験の結果からシステムのまとめを行う。

5.1 実装の概要

本システムは Window XP 上で、Visual C++ 2005 を用いて実装された。使っているカメラは Web カメラ 3 万画素で、GIF と VGA 画面に対して 30 frame/sec の速度がある。

カメラの動画を取得するために、OpenCV 1.0 のライブラリを使っている。SIFT のアルゴリズムはオープンソースバージョン 1.1.1 を利用している。本システム構築のために新たに書いたプログラムは 2000 行弱である。

5.2 システムのインターフェース

5.2.1 システムを起動した画面

システムを起動すると、図 5-1 のような画面（これからメイン画面と呼ぶ）が出てくる。一番目の物体の写真は左の方に描画される。ほかの物体を選択すると、それに応じて、その物体の写真が描画される。

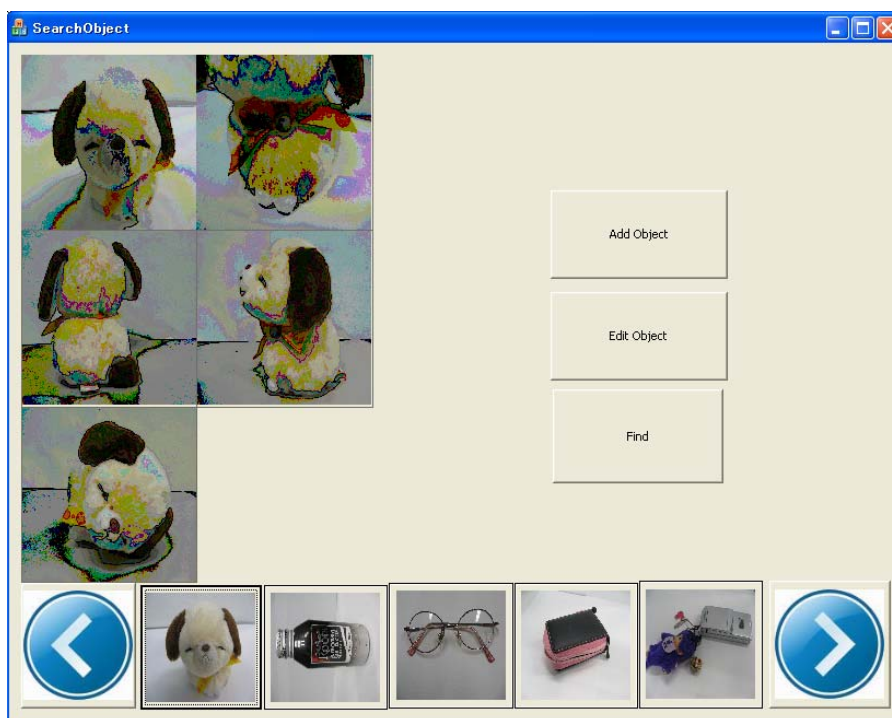
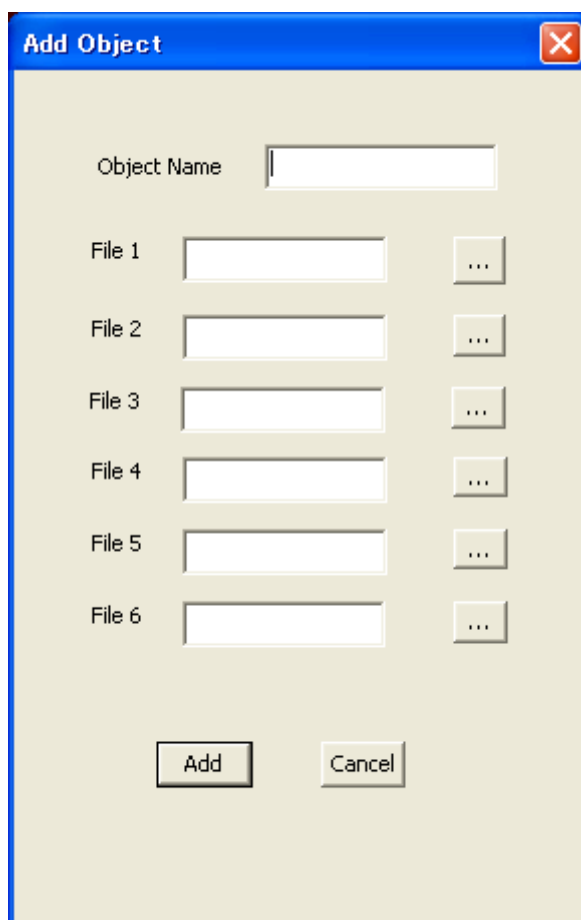


図 5-1 システムの起動画面

5.2.2 物体の追加

ユーザが追加のボタン (Add Object) を押すと、物体の情報の提出要求フォームが出てくる(図 5-2)。



The image shows a software dialog box titled "Add Object". It features a blue header bar with a close button (X) in the top right corner. The main area is light beige and contains the following elements:

- An "Object Name" label followed by a text input field.
- Six rows, each labeled "File 1" through "File 6". Each row consists of a text input field and a button with three dots (ellipsis) to its right.
- At the bottom, there are two buttons: "Add" and "Cancel".

図 5-2 物体の追加画面

物体の情報を付けてから、Add を押すと、物体の追加を行う。

5.2.3 物体の変更・削除

ユーザが物体の変更ボタン (Edit Object) を押すと、指定された物体の情報が示される(図 5-3)。

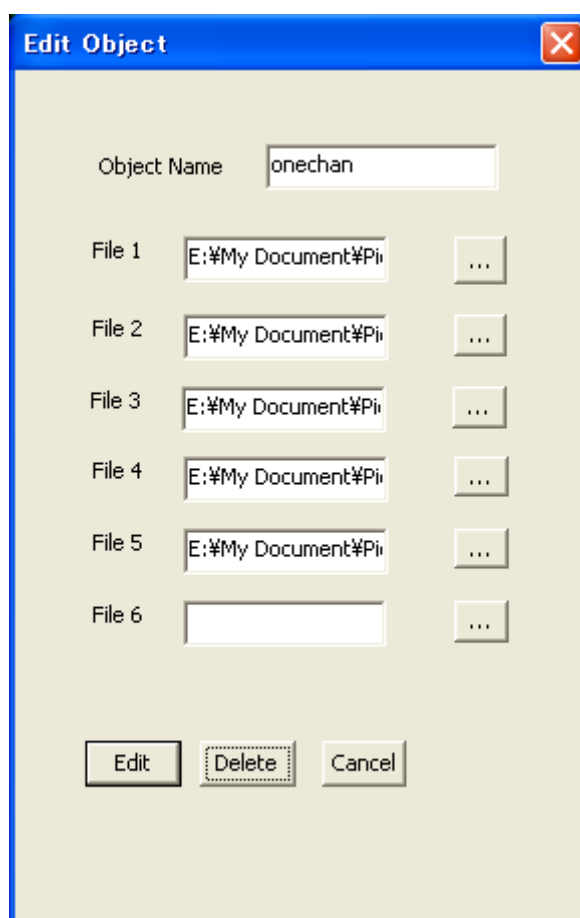


図 5-3 物体情報の変更画面

物体を変更・削除する前に、ユーザにもう一回確認がなされる(図 5-4)。

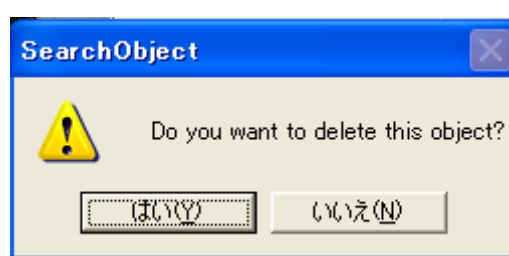


図 5-4 物体削除の確認フォーム

5.2.4 物体の探索

物体の探索ボタンを押すと、別のウィンドウを生成し、Web カメラの動画を示す(図 5-5)。探したい物を発見すれば、さらに別のウィンドウを生

成し、物体の部分を赤い四角で強調する。同時にシステムが一時的に停止し、ユーザの確認を待つ。探している物体である場合は探すことを終了する。また、そうではないときに、さらに探索を続ける(図 5-6)。

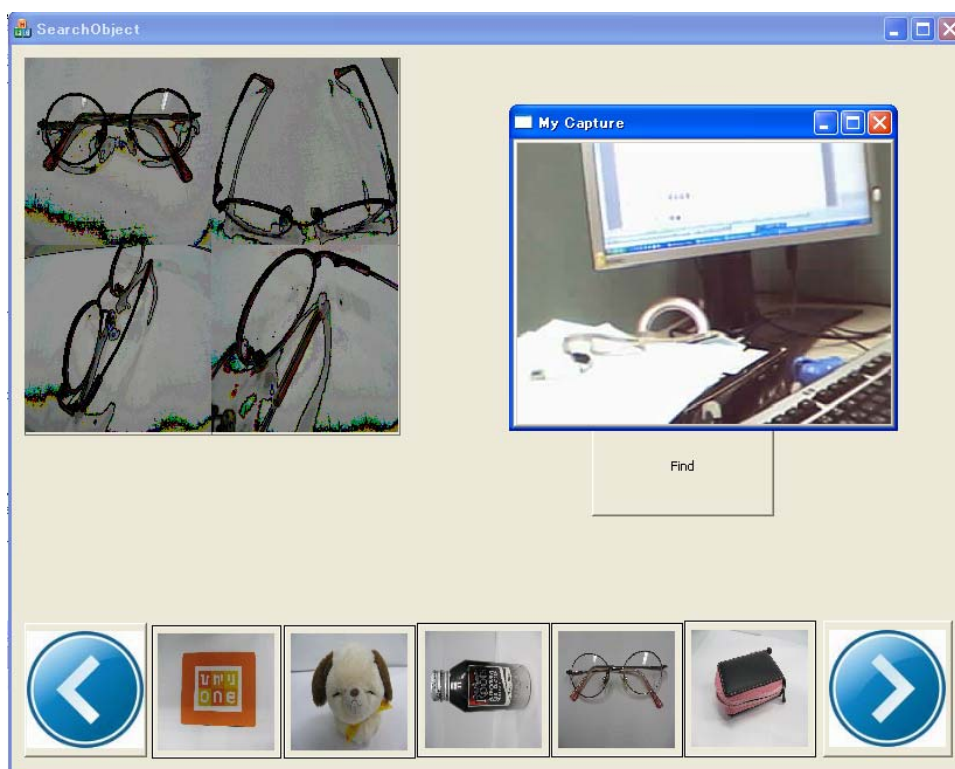


図 5-5 探し物の画面

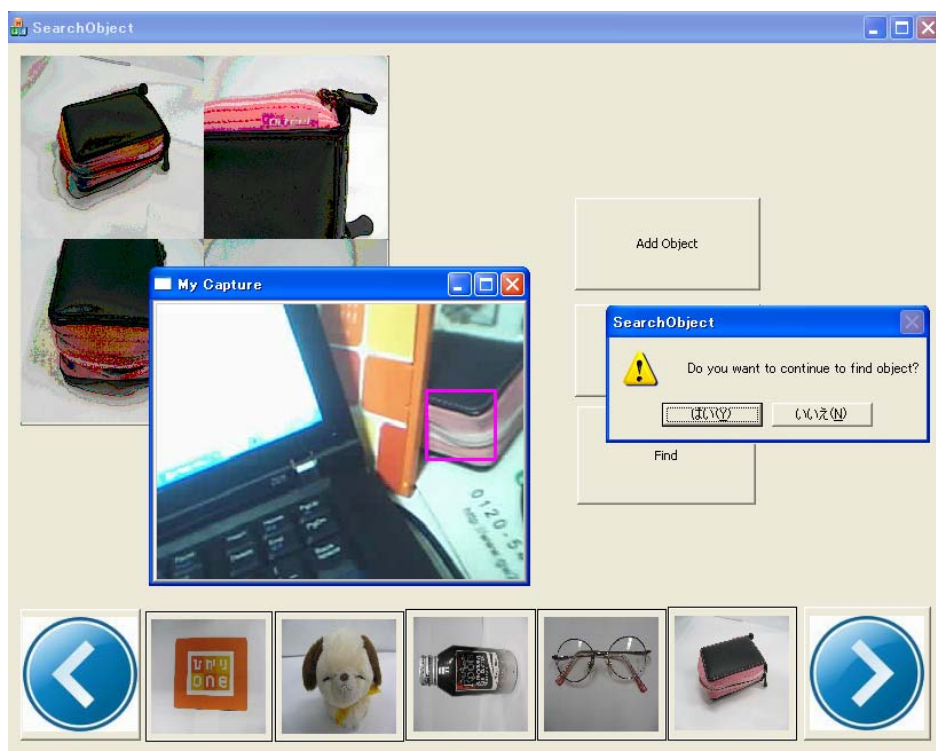


図 5-6 探し結果の画面

5.3 物体の管理

物体の情報を管理するために、図 5-7 に示したクラスを構成した。

C言語やC++で文字集合を管理するには通常、ASCIIのサブセットの文字を使用する。しかし、漢字のような拡張文字集合を扱う場合には、マルチバイト文字を使わなければならない。本システムでは物体の名前と物体の写真のパスのリストを管理するのにマルチバイト文字に対処できるTCHAR型を使っている。

```
class MyObject
{
public:
    int id;
    TCHAR name[50];
    int img_num;
    TCHAR* img_list[6];
    MyObject(void);
    int ReadFromFile (FILE *f);
    int WriteToFile (FILE *f);
    MyObject(int id, TCHAR* name, int _img_num, TCHAR* _img_list[]);
    int objNum;
public:
    ~MyObject(void);
};
```

図 5-7 物体のクラス

ReadFromFile() と WriteToFile() 関数は物体の情報をファイルから読む関数とファイルに書く関数である。

各物体の情報を管理するとき、ポインタを使うリストと配列の2つ方法がある。リストを使う利点は配列より要素が多く対処でき、データの削除が便利だが、ある要素に飛ぶ操作が後者より遅くなる。また、リストの方は実装の手間がかかる。それに対して、配列の方は要素の追加とある要素の位置に飛ぶことがリストより速いが、要素の削除のとき、削除された物体の各要素を直前に移動する必要があるので、遅くなる。

本システムでは物体の削除より、物体情報を変更する機能をよく使うと考えた。要素の数は、配列を使っても数千個まで管理できるから、配列で物体情報を管理することにした。

登録された物体は、object.dat というファイルに書かれている。システムが実行されると、このファイルから ObjList 配列に読み込まれる。

システムを終了した時や、物体を変更、追加した時は ObjList がファイルに書かれ保存される。

メイン画面の物体表示の管理には、presentObjIndex と firstObjIndex という変数を使っている。presentObjIndex は指定された物体、あるいは現在の物体の id を持つ。firstObjIndex は示している物体リストの一番目の物体 id を持つ。そのとき、ObjList[presentObjIndex] は指定した物体と ObjList[firstObjIndex] は一番目の示している物体のリストである。

物体の情報を変更したら、物体の情報を持っている ObjectList 配列を変更し、さらに、object.dat ファイルも書き直す。

物体の追加操作は図 5-8 のように処理する。最初の物体の場合、その物体を指定物体としてメイン画面に表示する。

追加された物体の id は**最後の物体の id + 1**である。

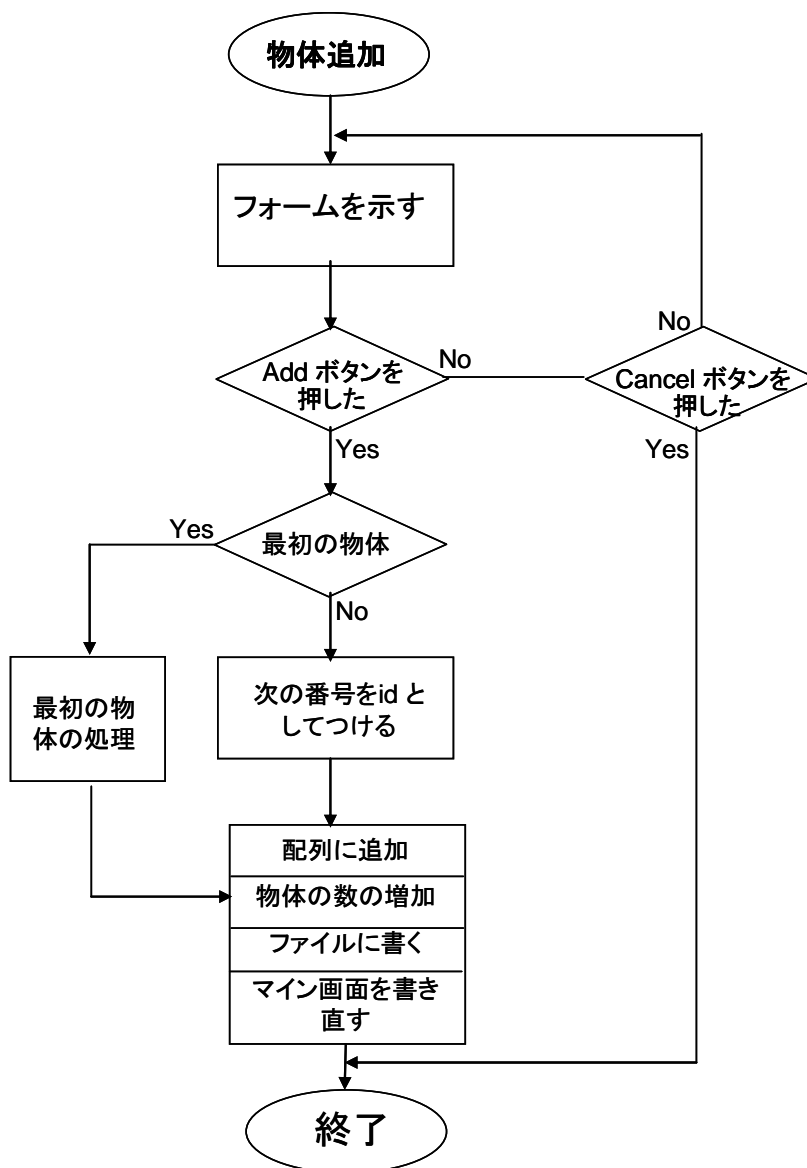


図 5-8 物体追加の流れ

物体の情報の変更や物体の削除という操作も同じように行っている。削除・変更の操作を行う手順は図 5-9 に示した。また、これらの操作の時は念のために、ユーザにもう一回確認してもらう。削除された物体が示している物体リストの最後であれば、`presentObjIndex` と `firstObjIndex` を変更する。さらに、メイン画面も描き直す。

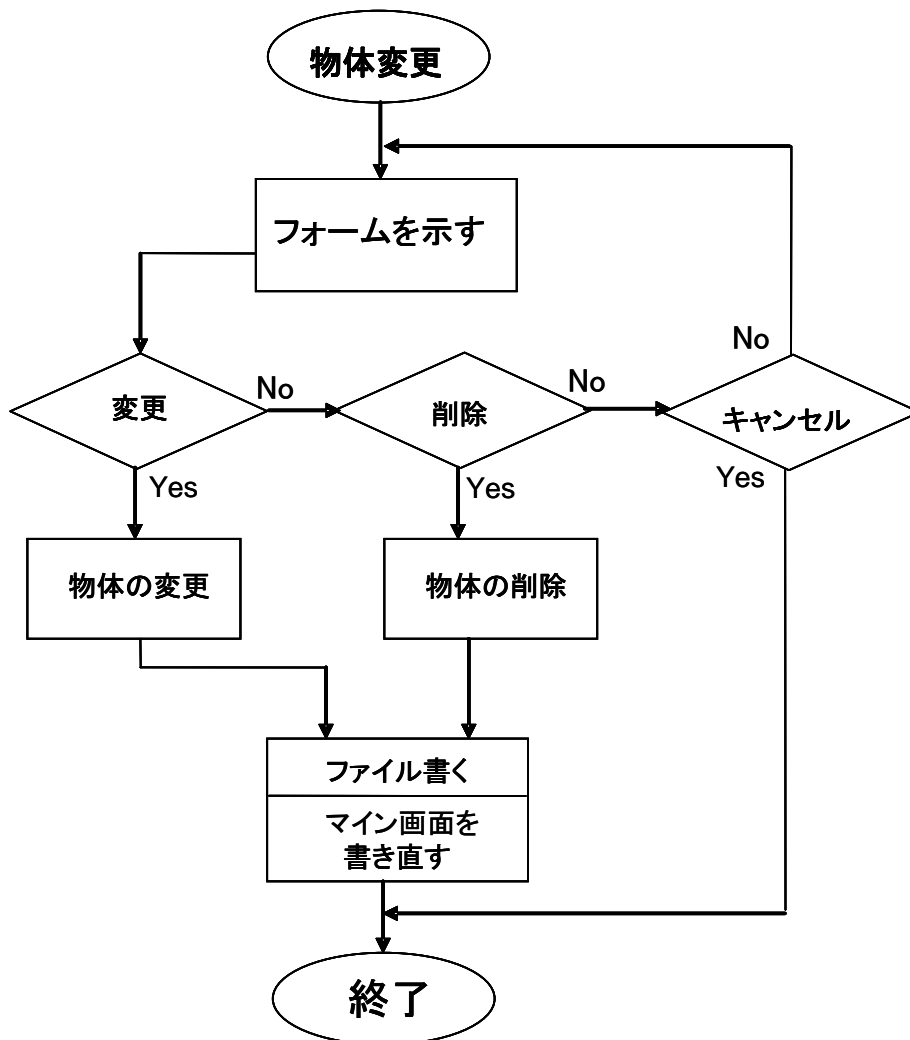


図 5-9 物体の変更・削除の流れ

5.4 物体の探索

5.4.1 カメラから動画の取得

Web カメラから動画を取得する方法は[12]を参考した。

OpenCV 1.0 ライブラリを使うと、Web カメラの動画を受け、それをキャプチャする操作は図 5-10 のように行った。

```
char* captureWindow = "Capture";  
capture = cvCaptureFromCAM(-1) //Webカメラの動画を取得  
while (1){  
    captureImage = cvQueryFrame(capture); //動画をキャプチャ  
    cvShowImage(captureWindow, captureImage); //動画を示す  
    キャプチャした画像を処理;  
    処理した画像を示す;  
}
```

図 5-10 Web カメラの取得

5.4.2 物体の探索

物体探索の操作は図 5-11 のように実装した。

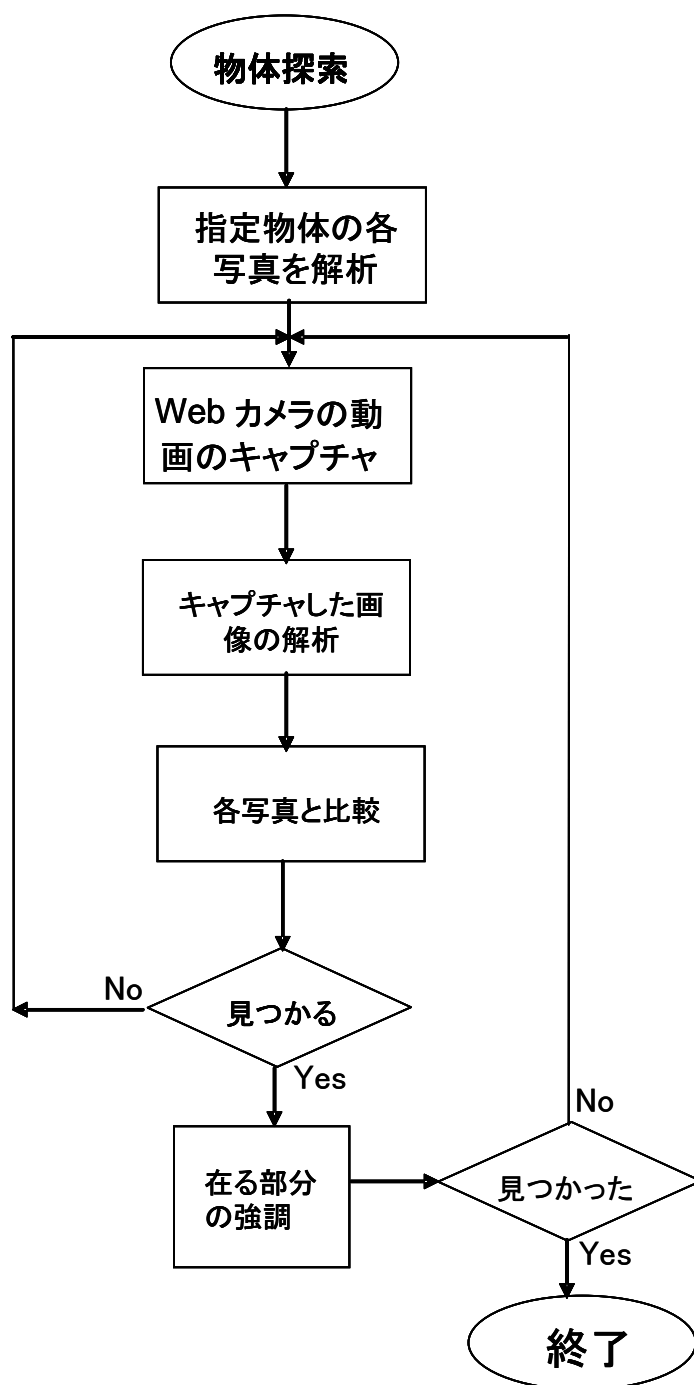


図 5-11 物体の探索の流れ

物体の写真とキャプチャした画像の解析は第2章で述べた SIFT アルゴリズムを利用し、それぞれの SIFT キーポイントを抽出する。

次に、それぞれのキーポイント間のユークリッド距離を計算することで、

それが最小となる点をキーポイントの対応点とする。それで、写真に対して、キーポイントの対応配列を計算できる。

しかし、ある物体のビューのキーポイントと他の物体のビューのキーポイントとかなり似ている場合が多いので SIFT キーポイントを使うと誤認識することも多い。物体がないけれども対応点が多く、物体の有無の判断が難しくなった。

それに対しては RANSAC [16] を利用し、以上の対応配列から、画像からマッチング対象の画像に変更する可能性のある配列を計算した。その配列を計算できる場合、物体があると判断する。物体がないと判断した場合は、次のカメラ画像のキャプチャを繰り返す。そうすると、誤認識数が減少できた。

SIFT アルゴリズムを実現する段階では特徴を抽出するために写真をグレイにしていた。そのため、違う色の物体を分別できないという問題があり、誤認識される場合がある。そのとき、ユーザの確認が必要である。

物体を発見した場合、その部分を強調し、ユーザの確認をもらう。ユーザが探している物体が見つかったと確認したら、探索を終了する。探している物体ではない場合、次の画像をまたキャプチャし、探索を行う。

強調部分は 60×60 ピクセルの赤い四角(カメラの画像は 320×240 ピクセル)で示している。各特徴点の位置を中心として、 60×60 ピクセルの四角を描く。特徴点が一番多く含まれている四角を強調部分として表示する。

5.5 結果

実際の物体について携帯電話、本、財布などのいろいろな形の物体を実験した。図 5-1 2 と図 5-1 3 はマッチング結果の例を示している。

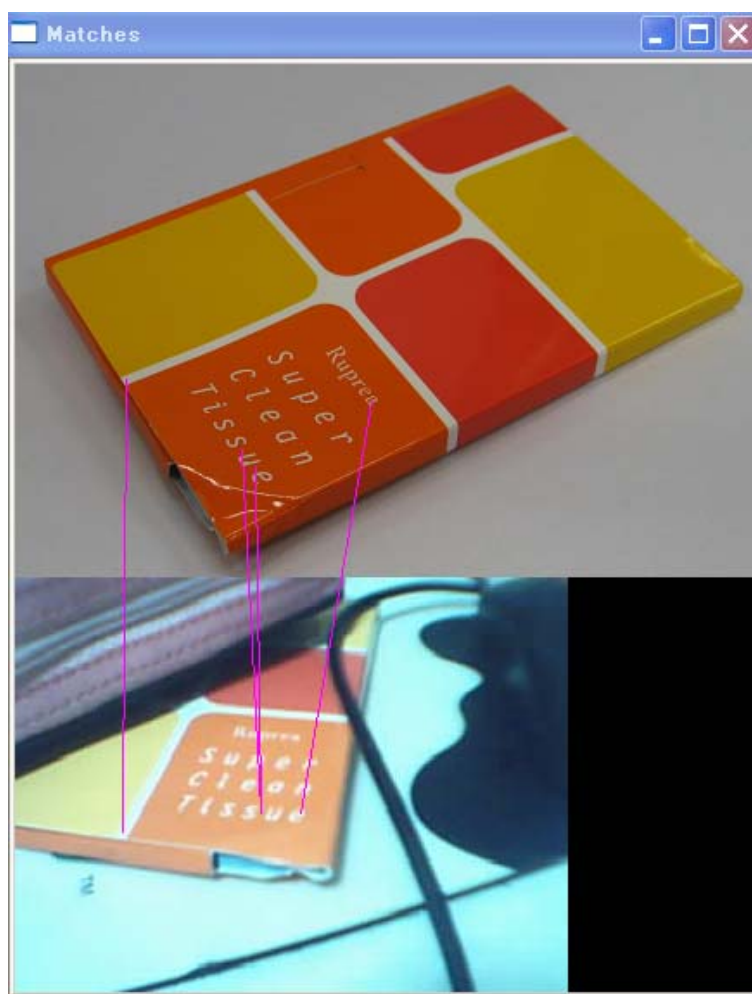


図 5-12 マッチングの結果



図 5-13 マッチングの結果

それによって、ユーザに図 5-14 のように通知する。

物体が登録した画面の 30 度以下回転すれば認識できた。また、輝度変更とスケール変更にも対処できた。写真のマッチング時間はその写真の SIFT キーポイントの数にも関係するが 648×486 ピクセルの写真 1 枚と Web カメラキャプチャした画像(320×240 ピクセル)をマッチングする時間は平均で 0.38 秒であった。6 枚の写真とマッチングすると、約 2.2 秒が必要となる。

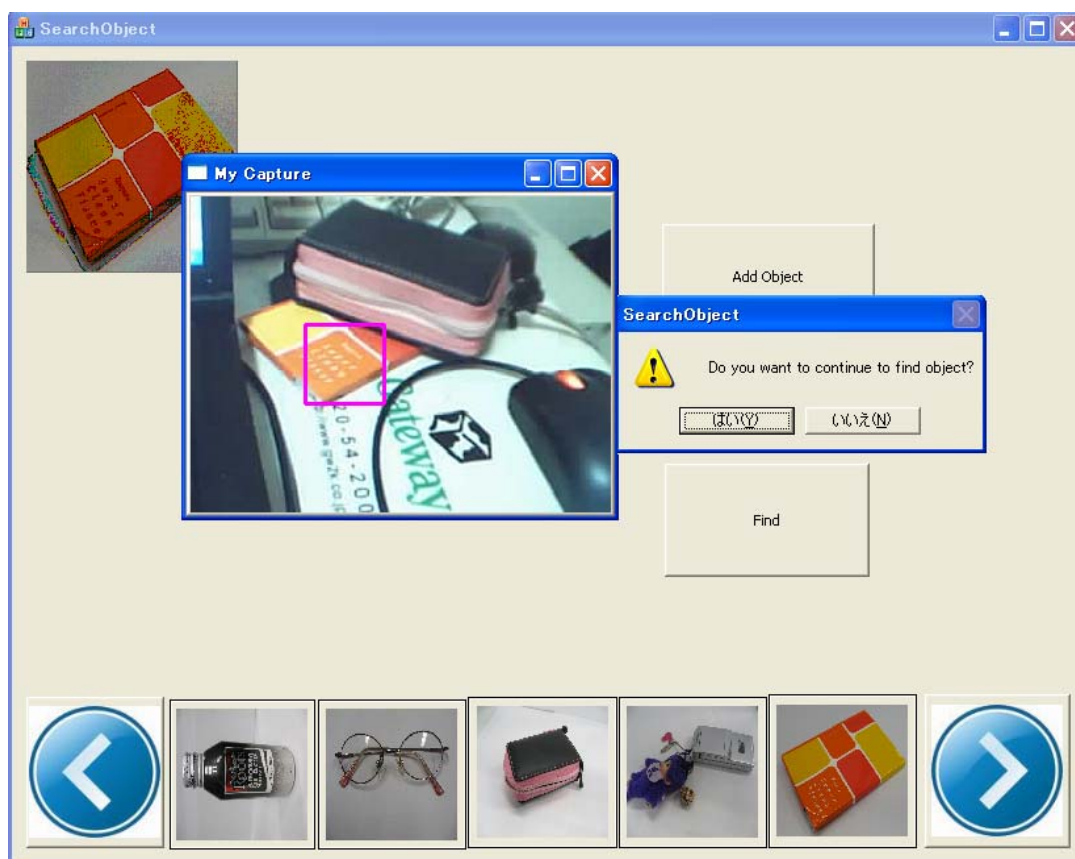


図 5-14 マッチングの結果を表示

探している物体が見つかったときは処理を一時的に中止し、以上のように示してユーザに報告する。ユーザが「はい」のボタンを押すと探索を再開し、「いいえ」のボタンを押すと探索を終了する。

5.6 まとめ

実際の実験から次の結果が示された。

本システムは 2 次元の写真から 3 次元の物体を探索することができる。物体の一部だけでも報告を出すことができるので、形が変わる物体を探索できる。本やティッシュパックのように平面があるものは他のものより、発見しやすかった。携帯電話やめがねは見つけにくかったがこれは光の反射が強いためと考えられる。しかし、携帯電話につけている物体をすぐ発見できるので、ユーザに報告することができた。また、本やティッシュパックは物体の 3 分の 2 が隠れていても見つけることができた。

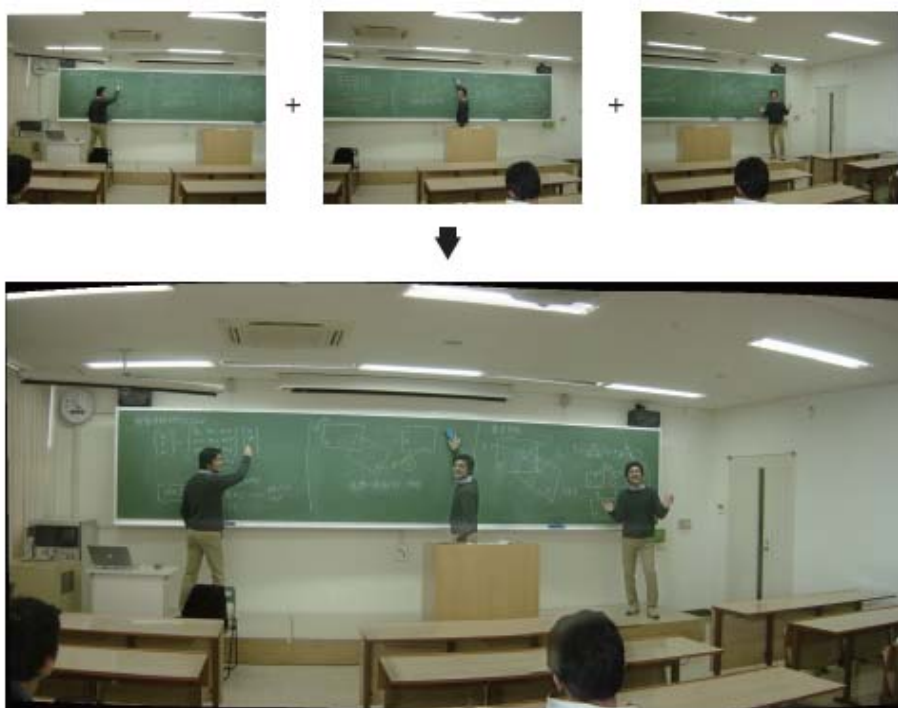
第6章

関連研究

この章では SIFT アルゴリズムの適用に関する研究について述べる。

SIFT アルゴリズムを用いる応用が広く研究されている。特に対応点の探索による画像のマッチングの応用や特定画像を用いた物体認識の応用が多く存在している。

Brown と Lowe は入力写真の集合から写真を分析し、写真特徴をマッチングすることで、合成画像を自動生成する方法を提案した[17] [18]。図 6-1 と図 6-2 は写真の集合から合成画像を作成する例を示す。

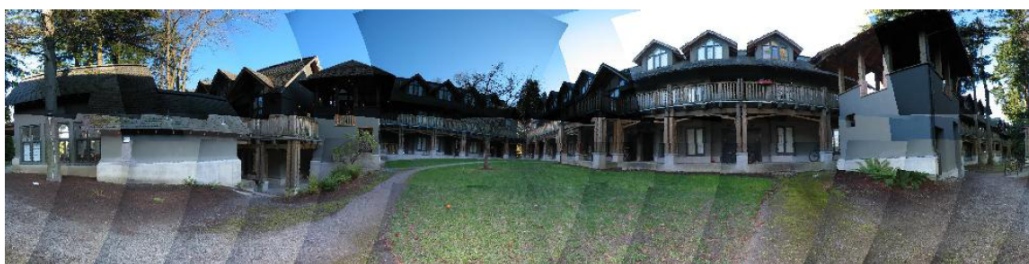


[[17]の参考文献から引用]

図 6-1 Autostitch で生成した合成画像



(a) 40 of 80 images registered



(b) All 80 images registered



(c) Rendered with multi-band blending

[[18]の参考文献から引用]

図 6-2 合成画像認識

また、高木らは、車載カメラによって撮影された前方実環境画像から、SIFT 特徴量を用いるイラストパターンの道路標識と実画像のマッチングを行っている。それにより標識を認識する手法を提案している[20]。得られた対応点のスケール、オリエンテーションから、中心位置座標に投票して標識の認識を実現している。図 6-3 に標識検出・認識結果例を示す。



[[19]の参考文献から引用]

図 6-3 道路交通標識の認識例

第7章

おわりに

本研究は画像認識技術で探し物を支援する新しいアプローチを提供した。このシステムは既存の物体探索支援システムと違って、物体に子機を付ける必要がない。また管理したい物の数を自由に増やすことができる。本システムは既存の物体認識システムと違って、物体の情報を登録するとき、物体の一部の写真で登録できる。

そして、システムの設計と実装も行った。実験から、物体が一部隠れていても探索できることが分かった。

本システムは既存の物体探索支援システムより発見率は落ちるが、低価格カメラを使うことで設置コストを安くできた。また、物体の一部が見えていれば発見できるため、形が変化する物体や柔らかい物体も発見できる可能性がある。

散らかった状態で一部が隠れていても発見できるため、将来、カメラのついていないロボットで地震の現場で物を探すことにも有益である。また、カメラ付きの携帯電話でも紛失した物を探索できる可能性がある。

今後の課題として、まず登録した物体の写真に背景がある場合、誤認識になるという問題がある。その時、エッジ抽出技術を使って写真の前処理すると良いのではないかと考えている。また、低価格カメラを使っているため取得画像の質は高くない。画像処理技術でキャプチャした画像の質を改良すると、もっと良い結果が出る可能性がある。

最後に、システムはキャプチャ画像を処理してから次の画像をキャプチャして

いるが、画像を処理しているときにカメラが物体のある画像を撮影すると、システムが物体を発見できないという問題がある。その解決方法として、処理速度の改良やカメラのキャプチャ仕方を考える必要がある。

謝辞

本研究を行うことにあつて、いろいろ方にお世話になりました。

まず、指導教員の多田好克先生には題目から熱心なご指導をいただきました。さらに、ご多忙中にもかかわらず、論文の草稿を丁寧に読んでいただき、日本語を直し、貴重なご意見をいただきました。

また、研究を進めるにあたり、小宮常康准教授と佐藤喬助教には貴重な助言をいただきました。ここに厚く御礼申し上げます。

最後に、本研究を行っているとき、研究方針や方法論について議論をご意見いただいた多田研、小宮研、村山研、水野研の皆さんに感謝いたします。

参考文献

- [1] 「どこいっ太郎」の商品,
<http://www.kumazaki-aim.co.jp/2007/03/rf315.html>
- [2] Robert J. Orr, Ronald Raymond, Joshua Berman, A. Fleming Seay: “A System for Finding Frequently Lost Objects in the Home,” Georgia Institute of Technology, GVU Technical Report:GIT-GVU-99-24, 1999.
- [3] 遺失物支援システム,
<http://www.j-tokkyo.com/2005/G01S/JP2005-003627.shtml>
- [4] 福井和広, 山口修: “部分空間法の理論拡張と物体認識への応用”、電子情報通信学会, 2004.
- [5] K. Mikolajczyk, A. Zisserman and C. Schmid: “Shape recognition with edge based features,” Proc. British Machine Vis. Conf., pp.384-393, 2003.
- [6] C. Harris and M. Stephens: “A combined corner and edge detector,” Proc. of Fourth Alvey Vision Conference, pp. 147-151, 1988.
- [7] T. Lindeberg: “Scale-space theory: A basic tool for analysing structures at different scales,” Proc. of Journal of Applied Statistics, vol.21, no.2, pp. 224-270, 1994.
- [8] C. Schmid and P. Mohr: “Local gray value invariants for image retrieval,” Proc. of IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI), vol. 19, no. 5, pp.530-534, 1997.
- [9] D. G. Lowe: “Object recognition from local scale invariant features,” Proc. of IEEE International Conference on Computer Vision (ICCV), pp.1150-1157, 1999.
- [10] D.Lowe: “Distinctive Image Feature from Scale-Invariant Keypoints,” Int.Journal of Computer Vision, vol.60, no.2, pp.91-110, 2004.

- [11] 動画像理解技術基礎講座, <http://www.vision.cs.chubu.ac.jp/>
- [12] Webcam Algorithm Input, J.R.V. Javantea,
https://www.altsci.com/concepts/ppm_parse1.html
- [13] J.S. Beis and D.G. Lowe: "Shape Indexing using approximate nearest-neighbour search in high-dimensional spaces," *Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition*, pp 1000-1006, 1997.
- [14] Andrea Selinger and Randal C. Nelson: "Appearance-Based Object Recognition Using Multiple Views," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 905-911, 2001.
- [15] Xavier Armangué and Joaquim Salvi: "Overall view regarding fundamental matrix estimation," *Image and Vision Computing* 21, pp 205-220, 2003.
- [16] Matthew Brown and David Lowe: "Invariant Features from Interest Point Groups," In *British Machine Vision Conference*, Cardiff, Wales, pp. 656-665, 2003.
- [17] Arthur R. Pope and David G. Lowe: "Probabilistic Models of Appearance for 3-D Object Recognition," *International Journal of Computer Vision*, vol.40, no.2, pp 149-167, 2000.
- [18] M. Brown and D. Lowe: "Recognising panoramas," *Proc. Int. Conf. Comp. Vis.*, vol.2, pp1218-1225, 2003.
- [19] Autostitch, <http://www.cs.ubc.ca/~mbrown/autostitch/autostitch.html>
- [20] 高木雅成, 藤吉弘亘: "SIFT 特徴量を用いた交通道路標識認識", 第 13 回 画像センシングシンポジウム SSII07, LD2-06, 2007.