

ユーザインタフェースのための オブジェクト追跡手法に関する研究

宮平 陽栄[†] 熊澤 逸夫[†]

[†] 東京工業大学 像情報工学研究施設 〒226 8503 神奈川県横浜市緑区長津田町 4259 R2 棟 330
E-mail: †{miya,kumazawa}@isl.titech.ac.jp

あらまし きたるユビキタス社会に向けて情報端末の小型化、高精度化が進む一方、情報技術を使いこなせる人と使いこなせない人との格差であるデジタルデバイドも深刻な問題となっている。その原因の一つとして人間と情報機器間の仲介をするユーザインタフェースの扱いにくさが上げられる。本研究では、人間の指の動きや運動している特定オブジェクトの画像情報を USB カメラ取得・処理し、人間が情報機器を直感的に扱えるようなユーザインタフェースを提案する。エッジベースの追跡手法を基とし、安定なエッジ領域を抽出した上で、オブジェクトの形状や移動量の変化情報をリアルタイムに得る手法を構築する。その情報を使うことによって、簡単な直感的なユーザインタフェースの構築を行う。

キーワード ユーザインタフェース, エッジ, パーティクルフィルタ, アフィン変換

Object Tracking for User Interface

Youei MIYAHIRA[†] and Itsuo KUMAZAWA[†]

[†] Imaging Science and Engineering Laboratory, Tokyo Institute of Technology
R2-330, 4259 Nagatsuta-tyo, Midori-ku, Yokohama-shi, Kanagawa, 226 8503 Japan
E-mail: †{miya,kumazawa}@isl.titech.ac.jp

Abstract Facing upcoming ubiquitous community, in terms of device operation, the gap between experienced users and novices has been a serious problem. One of the causes of problem is the difficulty in controlling the user-interface which intermediates human and computer communication. In this paper, we proposed a user-interface which focus on obtaining and processing image motion of intuitive finger movement or a specified object using USB camera. We constructed a method based combination of edge-based tracking method which can extract a stable edge area, and the information of object's shape and its changes in the movement quantity. By using such information, simple and intuitive user-interface was created.

Key words User Interface, Edge, Particle Filter, Affine Transform

1. はじめに

きたるユビキタス社会に向けて情報端末の小型化、高精度化が進む一方、情報技術を使いこなせる人と使いこなせない人との格差であるデジタルデバイドも深刻な問題となっている。その原因の一つとして人間と情報機器間の仲介をするユーザインタフェースの扱いにくさが上げられる。現状の多くのデバイスは、ブラインド入力に多くの労力を要するものや、入力の度に手元のデバイスと手の位置関係を目で確認しなければならない等の煩わしさを生じるものが多い。代表的なものを上げれば、文

字入力等で使われるキーボードや運転中のカーナビの操作やデジタルテレビのリモコン操作などを考えればわかりやすいだろう。そのため、扱う人間がより直感的に操作できるようなユーザインタフェースが必要が高まってきている。

直感的なユーザインタフェースを実現するために最も重要な技術の一つとなるのが、人間の行動や動作を認識する技術である。そのための方法として、モーションキャプチャ等を被認識者に装着する方法やカメラ等で撮影した画像を解析して認識する方法がある。前者は動作認識

には比較的容易だが、被認識者に専用器具等をつけるなどの拘束条件が生じるのに対し、後者では取得した画像を解析する必要があるが、被認識者に余計な拘束条件はほとんどなくなる。また、USBカメラや携帯電話に付属するカメラ等の急速に普及や、画像処理技術の進歩により顔認識や視線入力[1]やジェスチャー認識[2]等様々な研究が盛んに行われている。近年では、Wiiリモコンや視線入力装置など実用化されているものも多い。

本研究では、作者の指やマーカなどの特定オブジェクトの画像情報をUSBカメラ取得・処理し、人間が情報機器を直感的に扱えるようなユーザインタフェースを提案する。具体的にはカメラから取得した画像の輝度値情報から指やオブジェクトのエッジ情報を抽出し、その情報を元に対象の形状や移動量の変化に対応できる形状モデルを構築し、リアルタイムに検出することを目指す。その検出量をもとに直感的なユーザインタフェースの構築を行った。

2. エッジ情報の安定領域の抽出法

エッジベース追跡法において有効とされるエッジ情報には、任意の相似変換に対して、

- 空間的に正しい位置で検出ができる
- 局所的な変化に影響されにくい（大域的な変化のほうを優先する）

という条件が必要となる。これを満たさないエッジ情報はマッチングにおいて不要な処理を生じることになるので、前処理の段階で除外すべきである。

一方、カメラから入力される画像は主に図1のような特徴を持つ4つの領域に大別することができる。

- Area1 大域的な形状の輪郭を含む領域
- Area2 局所的な形状の輪郭を含む領域
- Area3 輝度変化が小さく微細なテクスチャを含む領域
- Area4 大域的な形状の輪郭と輝度変化が小さく微細なテクスチャを含む領域

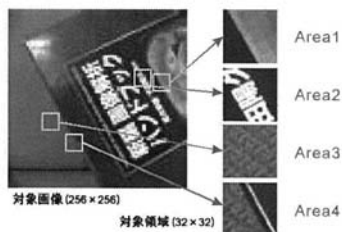


図1 輪郭の特徴が異なる領域

Area2では、領域内に存在する形状は実空間では明確な輪郭として存在するが、撮影画像においては輪郭が不安定となっている。すなわち、撮影装置の解像度が撮影対象の輪郭を表現する上で十分でない。Area3では、領域内の輝度の変化が少ないため、量子化による影響を大

きく受ける。すなわち、任意の相似変換において量子化の影響が大きく受ける可能性がある。

以上のことより、Area2、Area3に類する領域を除外し、Area1、Area4に類する領域のみを抽出することができれば安定なエッジ情報を抽出することができる。Area2では本来あるべき輪郭が不安定すなわち画像内の輝度の勾配ベクトルも不安定であること、Area3では領域内の輝度の分布のばらつきが少ないことに着目し、安定領域を抽出する。

2.1 輝度分布を利用した安定領域の抽出

(1) 図2に示すように、入力画像をブロック分けを行う。ブロックの境界にエッジが存在する可能性もあるため、ブロック幅の半分ずつ重複を許す。



図2 画像のブロック分け

例えば、 256×256 の画像を 32×32 のブロックで分ける場合、以下の式により225種類のブロックが生成される。

$$(256/32 \times 2 - 1)^2 = 225[\text{blocks}] \quad (1)$$

(2) 各ブロック内の領域を切り出したものを1つの画像 B_i ($0 \leq i < N$)とみなす。

(3) B_i における輝度の分布を求め、その標準偏差 $\sigma(i)$ を求める。

(4) 閾値 T_σ (注1)を定義したとき、ブロック B_i が以下の条件を満たせば、エッジ情報が安定する領域 S_1 に含まれるとする。

$$\sigma(i) > T_\sigma \Rightarrow B_i \in S_1 \quad (2)$$

2.2 輝度勾配ベクトルを利用した安定領域の抽出

(1) S_1 に含まれる領域に対して、2.1で用いたブロックよりもさらに細かいブロックを利用して、重複を許したブロック分けを行う。ただし、 S_1 を取得した時点で領域は絞り込まれているので、今回は1画素ずつブロックをずらしながら重複した領域を取得するブロック分けを行う。

(2) 各ブロック内の領域を切り出したものを1つの画像 B'_i ($0 \leq i < N - 1$)とみなす。

(3) 領域 B'_i での平均勾配ベクトル \bar{w} を求める。

$$\bar{w} = \frac{1}{N} \sum_{(i,j) \in B'_i} w(i,j) \quad (3)$$

ただし、 N は B'_i における画素数とする。

(注1): 本論文では $T_\sigma = \frac{1}{N} \sum_{i=0}^{N-1} \sigma(i)$ ($\sigma(i)$ の平均値)を用いた。

(4) \bar{w} の示す勾配方向 $\text{dir}(\bar{w})$ を B'_i における輝度の勾配方向の平均 $\bar{\theta}$ とする。

(5) $\bar{\theta}$ を用いて、以下の式で B'_i における輝度の勾配方向のばらつき度 σ_θ を以下の式で定義する。この σ_θ は、 $\bar{\theta}$ を勾配方向の平均と捉えたとき、勾配強度を重みとして得られる勾配方向の標準偏差となっている。

$$\sigma_\theta = \sqrt{\frac{\sum_{(i,j) \in B'_i} |w(i,j)| \{w(\text{dir}(w(i,j)) - \bar{\theta})\}^2}{\sum_{(i,j) \in B'_i} |w(i,j)|}} \quad (4)$$

ただし、関数 $w(x)$ は角度 x を $-\pi \leq x < \pi$ となるように正規化する関数を示す。

(6) B'_i における輝度の勾配方向のばらつき度 σ_θ を求める。

(7) 閾値 T_{σ_θ} (注2) を定義したとき、 B'_i が以下の条件を満たせば、エッジ情報が安定する領域、かつ不十分な解像度で表現されている形状が存在しない領域 S_2 に含まれるとする。

$$\sigma_\theta(i) > T_{\sigma_\theta} \Rightarrow B'_i \subset S_2 \quad (5)$$

以上の処理をエッジ情報の安定領域抽出法 [3] とする。エッジ情報の安定領域抽出結果を図 3 に示す。



図 3 安定領域抽出結果

3. 形状モデルの構築

本論文では、特徴ベースモデルを基盤とした初期画像より形状モデルを以下の流れで構築した。流れ図を図 4 に示す。

(1) 追跡対象の輝度情報に Canny フィルタを適用し、エッジ情報を取得する。

(2) 追跡対象のエッジ情報の外形輪郭部分を端から連結して、グループ化する。

(3) グループ化したエッジ情報を両端を始点、終点と決め等間隔に N 個の代表点を取得する。

(4) 取得した代表点間をスプライン補間を用いて、補間し代表点での法線ベクトルの角度 $\{\theta_i, i = 1, \dots, N\}$ を求める。

(5) 取得した代表点の状態ベクトルを $\{s_i, i = 1, \dots, N\}$ とし、対象の形状モデルとする。

(6) 対象の輪郭に関してはスプライン補間を用いることで補間する。

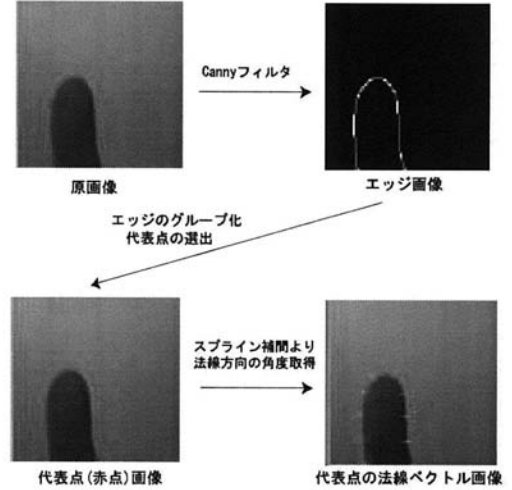


図 4 初期画像より形状モデル構築の流れ

状態ベクトル s_i は以下の情報を持つ

- 入力画像の右上を原点とする 2 次元座標 (x_i, y_i)
- 各代表点の法線方向の角度 θ_i
- 重み w_i

重み w_i に関しては 4. で述べる。また、状態ベクトル $s_i, i = 1, \dots, N$ の集合を $\mathbf{S} = \{s_1, \dots, s_N\}$ とし、重心位置 (G_x, G_y) を以下のように定義する。

$$G_x = \frac{1}{N} \sum_{i=1}^N x_i \quad (6)$$

$$G_y = \frac{1}{N} \sum_{i=1}^N y_i \quad (7)$$

また、時間 t において状態ベクトルの集合を $\mathbf{S}(t)$ とすると時間 $t+1$ における対象の拡大・縮小・回転・平行移動等の形状変化を考慮した $\mathbf{S}(t+1)$ はアフィン変換を適用して以下のように表される。 $i = 1, \dots, N$ に関して、

$$\begin{bmatrix} x_i(t+1) \\ y_i(t+1) \end{bmatrix} = A \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x_i(t) - G_x \\ y_i(t) - G_y \end{bmatrix} + \begin{bmatrix} dx \\ dy \end{bmatrix} + \begin{bmatrix} G_x \\ G_y \end{bmatrix} \quad (8)$$

$$\theta_i(t+1) = \theta_i(t) + \theta \quad (9)$$

ようになる。ここで、 A は拡大縮小、 θ は回転、 dx は

(注2) : 本論文では $T_{\sigma_\theta} = \frac{1}{N} \sum_{i=0}^{N-1} \sigma_\theta(i)$ ($\sigma_\theta(i)$ の平均値) を用いた。

x 方向の移動量, dy は y 方向の移動量である. すなわち, 各時間毎の対象の形状変化はアフィン変換のパラメータ (A, θ, dx, dy) を決定することができれば対応することができる.

4. 追跡手法の構築

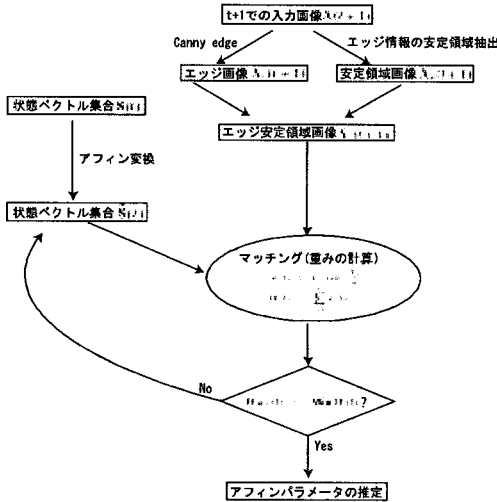


図5 アフィンパラメータ推定の流れ

時間 $S(t) \rightarrow S(t+1)$ に変化する際のアフィンパラメータ (A, θ, dx, dy) の推定を以下のように求める.

時間 $t+1$ で得た入力画像を $X(t+1)$ とすると,

- (1) $X(t+1)$ に対して, Canny フィルタを適用し 2 値エッジ画像 $X_e(t+1)$ を得る.
- (2) $X(t+1)$ に対して, エッジ情報の安定領域抽出法を適用し, 安定領域画像 $X_s(t+1)$ を得る.
- (3) $X_e(t+1)$ と $X_s(t+1)$ より, 安定な領域のみの 2 値エッジ画像 $\tilde{X}_e(t+1)$ を得る.
- (4) 状態ベクトル集合 $S(t)$ に一定の範囲を動くパラメータ $(\tilde{A}, \tilde{\theta}, \tilde{dx}, \tilde{dy})$ を適用し, アフィン変換された状態ベクトル集合 $\tilde{S}(t)$ を得る
- (5) $\tilde{X}_e(t+1)$ 中で \tilde{s}_i の法線方向においての最も近いエッジを探索し, その距離を l_i とし, 重み $\tilde{w}_i(t)$ を以下の式で求める. σ は定数.

$$\tilde{w}_i(t) = 1 - \exp\left(-\frac{l_i^2}{\sigma}\right) \quad (10)$$

(6) 状態ベクトルの全ての重みを $\tilde{W}(t)$ とおくと, 以下の式で定義する.

$$\tilde{W}(t) = \sum_{i=1}^N \tilde{w}_i(t) \quad (11)$$

(7) 全てのパラメータの範囲内の計算を終えるまで, パラメータの値を変え (4) に戻る.

(8) 以下の式で求まる $\tilde{W}_{Min}(t)$ のときのアフィンパラメータを形状変化パラメータとみなす.

$$\tilde{W}_{Min}(t) = \text{Min } \tilde{W}(t) \quad (12)$$

(9) $\text{Min } \tilde{W}(t)$ がある値 W_{thres} よりも大きければ, $\tilde{X}_e(t+1)$ で最も近かったエッジを新たに状態ベクトルとして更新する.

図5に以上によるパラメータ推定の流れを示す.

5. 高速化の検討

4. で, 追跡手法を構築したが, 実際に追跡処理を行うためには, 適切なアフィンパラメータの範囲を指定しなければいけない. パラメータの範囲が広いほど計算コストが高くなるため, 適切な範囲を調整しなければいけない. そこで, 指定したパラメータ範囲全てをアフィン変換するという無駄なくすことや, 時系列データにおいて過去情報よりある程度の形状や運動を予測する概念を導入することで計算コストを削減することを考える.

5.1 評価関数の検討

4. で説明したアフィンパラメータの推定法では, 定めたパラメータ範囲全てをアフィン変換して評価関数を適用した. しかし, この方法では無駄なアフィン変換があるため計算コストが高くリアルタイムには向かない. そこで, 領域ベース追跡法のテンプレートマッチングでよく使われる高速化のための手法である残差逐次検定法を応用する. すなわち, あらかじめ決まったアフィンパラメータ全てを変換してから評価関数でパラメータを推定するのではなく, 逐次パラメータ変換後評価を行い推定を行う. その際に, 適切な閾値を定め無駄なパラメータによる変換を削除していく. すなわち, 以下のステップにより高速化を実現する

- (1) ある値のアフィン変換を適用する.
- (2) 式 (10), 式 (11) を計算する.
- (3) その際に, 以前までに求めた評価関数式 $\tilde{W}(t)$ に関して, $\tilde{W}(t) < W(t)$ となった時点で計算を打ち切る.
- (4) (1) に戻る.

5.2 パーティクルフィルタの導入

時系列データにおいてベイズ理論に基づき, 対象の状態を推定する方法にパーティクルフィルタ [5] [6] [7] がある. 本論文の提案手法におけるパーティクルフィルタの導入をパーティクルフィルタの操作である, サンプリング, 重み付け出力, 再サンプリングごとに説明する.

このパーティクルは以下のような情報を持つ.

- 形状モデルの重心位置座標 $\mathbf{s} = (G_x, G_y)$
- 重み w
- サンプリング数 n

時間 k における i 番目のパーティクルを $\mathbf{x}_k^{(i)}$, それを持つ情報を並べて $\{\mathbf{x}_k^{(i)} = (\mathbf{s}_k^{(i)}, w_k^{(i)}, n_k^{(i)})\}$ とする. またパーティクルの総数を N とする.

- サンプリング

平均 0, 分散 \mathbf{V} の 2 次元ガウス分布に従う $n_{k-1}^{(i)}$ 個の乱数を発生させる。この乱数を使い、各パーティクルのある位置 $\mathbf{s}_k^{(i)}$ の周りに、 $n_{k-1}^{(i)}$ 個の点をうつ。この点を慣性の法則により動かし、移動した 1 つ 1 つの点をパーティクル $\mathbf{x}_k^{(i)}$ とする。

- 重み付け

パーティクルの持つ重み $w_k^{(i)}$ は、4. で説明した評価関数を適用し、 $\tilde{w}_k^{(i)}$ とすると以下の式により正規化した値とする。

$$w_k^{(i)} = \frac{\tilde{w}_k^{(i)}}{\sum_{j=1}^N \tilde{w}_k^{(j)}} \quad (13)$$

ここでは、二次元座標すなわち d_x, d_y のパラメータの範囲をパーティクルフィルタにゆだねることになるため、それ以外のパラメータである拡大縮小量や回転量のパラメータ範囲は従来どおりに行う。

- 出力

もっとも重みの重い位置を出力とする。

- 再サンプリング

各パーティクルの位置における追跡対象の確率密度の割合、すなわち $w_k^{(i)}$ の値に応じて、各パーティクルに割り振るパーティクル数 $n_k^{(i)}$ を決める。

パーティクルフィルタでは、以上のステップを用いることにアフィンパラメータの移動量である値 d_x, d_y の範囲を定める。そのため、移動量の探索範囲はパーティクル数 n に依存した計算コストとなる。また、パーティクル自身がまったく同じ位置に来たりすることもあるため、常に計算コストが同じになるわけではない。

6. 計算機実験

以上の高速化の工夫を踏まえた上で、計算機上で作った動画像に対して提案した追跡法の検証を行う。評価は以下の点を検証する。

- 画像一枚あたりの処理速度
- 追跡対象の拡大・縮小誤差
- 追跡対象の回転誤差
- 追跡対象の平行移動の誤差 (マンハッタン距離)

一番目は計算コストのための検証であり、残りは追跡精度の検証である。計算機上で作成した動画像は以下のような特徴を持つ。

- 対象は白で、背景は黒の二値化画像
- 追跡対象は正方形
- フレーム毎に拡大縮小や回転や平行移動を行う
- 全部で 53frame
- 画像サイズ: 320 × 240
- フレームレート: 15 枚/秒

図 6 に実験動画の画像例を示す。

また、今回の実験動画において設定したパラメータは以下の通りである。

- θ : $0 \leq \theta \leq 3$ (1 度刻み)

- A : $0.8 \leq A \leq 1.2$ (0.1 倍刻み)
- 状態ベクトルの数: $N = 25$
- Canny フィルタの閾値: $T_h = 130, T_l = 30$
- エッジ情報の安定領域抽出の小領域サイズ: 8×8
- パーティクル数 $Num = 100$

図 7 に結果の画像例を示す。図中の点はパーティクルの位置を示す。

各々の手法を比較してみると、カルマンフィルタによる追跡結果はパーティクルフィルタによる追跡結果に比べて拡大・縮小、回転、平行移動のいずれも良い結果を出していることがわかる。しかし、計算コストに関しては、カルマンフィルタよりもパーティクルフィルタのほうが少ないことがわかる。各手法の全フレームでの平均誤差、平均処理速度を表 1 にしめす。

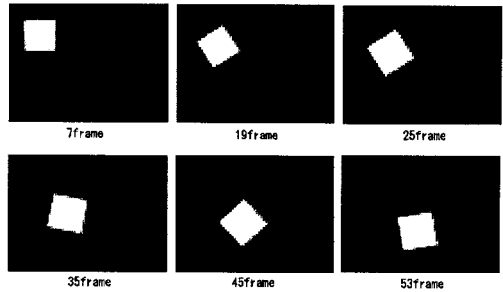


図 6 実験動画の画像例

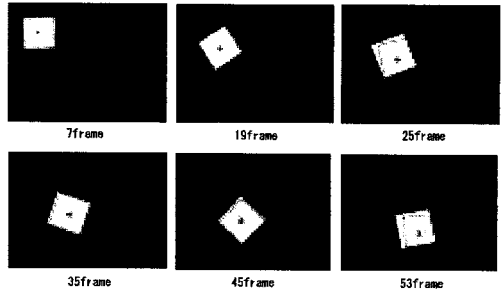


図 7 結果の画像例 (パーティクルフィルタ使用)

表 1 平均精度

処理速度	拡大縮小誤差	回転誤差	平行移動誤差
58.4 [msec]	0.051 [倍]	0.74 [度]	2.81 [pixel]

平均処理速度を見てみると、15fps の速度を出すのに必要一枚あたりの処理速度 66.6msec よりも速い処理速度を実現しているのがわかる。拡大・主供養や回転誤差に関しても変数幅の最小単位以下の精度で追跡できていることがわかるが、平均移動誤差に関しては誤差が大きいのが目立つ。しかし、高速化の手法を組み込むことでリアルタイムに追跡が実現できるような手法の構築を確認することができた。

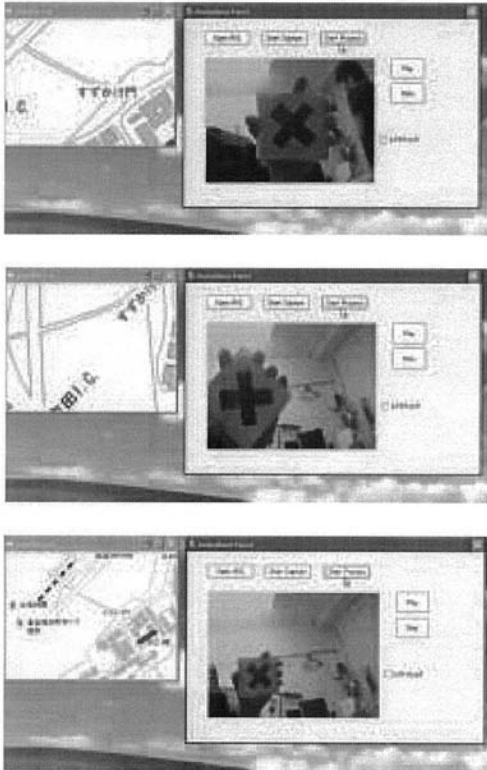


図 8 ユーザの操作による地図操作例

7. 画像認識による地図操作システムの構築

以上に提案した追跡法を使って、実際に対象の動きに対応した地図の操作システムの構築を行う。地図の一部分のみの領域のみがユーザが見える領域とする。これを以下のようなオブジェクトの動きによって地図の表示方法を変更する。

- (1) オブジェクトがカメラの光軸に対して平行移動を行えば地図を移動した方向の表示を行う
- (2) オブジェクトがカメラの光軸に対して回転を行えば地図を回転させ表示を行う。
- (3) オブジェクトがカメラの光軸に向かって近づいたり遠ざかったりすれば地図を拡大したり縮小したりして表示する。

実験環境は以下の通りである。

- 実験動作マシンの計算機：Pentium4 2.4GHz
- 使用USBカメラ：Logitech QuickCam Pro 4000
- 画像サイズ：320 × 240
- フレームレート：15 枚/秒
- プログラム環境：Visual Studio 2005

図 8 に実際に操作した例を示す。

8. ま と め

本研究では、特徴量としてエッジを用い、安定領域抽出法と組み合わせることでよりエッジ情報の安定な領域のみを検出し、その中の代表点を使った形状モデルを構築した。形状モデルにアフィン変換を基とするマッチングの評価関数を定義し、高速化を行うことで、リアルタイム処理に適した追跡法を構築することができた。また、その追跡法を使って、地図操作システムという直感的なユーザインタフェースの一部を試作し、操作できるのを確認した。

今後の課題として、現状ではピクセル単位での検出を行っているため、カメラから取得できるサイズ以上に検出を行うことはできない。これは、ユーザインタフェースへの応用を考えると操作性に問題がでてくる。そのため、検出精度をピクセル以下（サブピクセル精度）で検出することができればより高精度な検出が可能と考えられる。また、今回提案した手法では、外輪郭がとれるオブジェクトであればどんなオブジェクトでもアフィンパラメータを推定することで追跡を行うことができた。しかし、ユーザインタフェースに応用する場合を考えた場合、追跡対象の拡大・縮小、回転、平行移動だけでなくどんな動作を行ったかのようなジェスチャー認識も必要となってくる場合が多い。例えば、指の追跡・認識に関しては指を曲げるなどの動きも認識することができれば直感的なインタフェースとしての利用価値はさらに高まってくることを考えられる。

文 献

- [1] 北島 光太郎, 佐藤洋一, 小池英樹 “机型実世界指向インターフェースのための顔特徴領域実時間追跡” 情報処理学会インタラクティブ 2000 論文集, pp.145-146, February 2000.
- [2] Soren Lenman et al “Computer Vision Based Hand Gesture Interfaces for Human-Computer Interaction” TRITA-NA-D0209.
- [3] 高畑 知也, 宮平 陽栄, 熊澤 逸夫 “パターンマッチングにおけるエッジ情報の安定性の評価および安定したエッジの抽出法に関する研究” 電気学会 情報処理・産業システム情報化合同研究会, August 2006.
- [4] J.Canny, “A Computational Approach to Edge Detection” IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol8, Nov 1986.
- [5] M. Isard and A. Black, “CONDENSATION-conditional density propagation for visual tracking” Int’l J. of Comput.Vision vol.1, pp767-781,1998
- [6] Pihua LI, Tianwen Zhang, Authur E. C. Pece, “Visual contour tracking based on particle filters”, Image and Vision Computing 21, PP.111-123,2003.
- [7] Bando T et al “Switching Particle Filters for Efficient Real-time Visual Tracking”, int ConfPatRecog(ICPR2004),Cambridge,U.K,2004(in press)