

## Chamfer Matching を利用した有向 NFTG とその応用

岡田 大輝<sup>†</sup> 和田 俊和<sup>†</sup> 坂垣内 洵也<sup>†</sup>

<sup>†</sup>和歌山大学 システム工学部 情報通信システム学科

〒 640-8510 和歌山県 和歌山市 栄谷 930

E-mail :†{d.okada,twada,sakajun}@vrl.sys.wakayama-u.ac.jp

あ ら ま し 我々は、近さ優先探索グラフ (NFTG) を用いた擬似的な最近傍探索により、画像上の対象の追跡と認識を同時に行う方法を提案している。この手法では、姿勢パラメータをあえてずらした画像を登録してそこに埋め込まれたテンプレート修正パラメータを利用して対象枠を変形する姿勢パラメータ埋め込み (PPE) を行っている。このため、大量の画像データをデータベースに登録する必要がある、対象とテンプレート枠がある程度重なっていなければならない、などの問題点が生じる。また、画像そのものを利用して、照明条件の変化の影響を受けやすいという問題もある。これらの問題を解決するために、Chamfer Matching を用いたエッジ画像のマッチングを NFTG に組み込む方法を提案する。Chamfer Matching で用いられる相違度は距離の公理のうちの対称性を満足しないため、NFTG は有向グラフとなるが、そのような場合にも画像上での探索と同時にデータベース内の探索が同時に実行できることを示す。キーワード チャンファーマッチング、近さ優先探索、近さ優先探索グラフ、対象追跡・認識

### Directed NFTG incorporating Chamfer Matching and Its applicaction

Daiki Okada<sup>†</sup>, Toshikazu Wada<sup>†</sup>, Junya Sakagaito<sup>†</sup>,

<sup>†</sup>Department of Computer and Communication Science, Wakayama University,

930 Sakaedani, Wakayama, 640-8510, Japan

E-mail :†{d.okada,twada,sakajun}@vrl.sys.wakayama-u.ac.jp

#### Abstract

We have proposed an image retrieval based object tracking and recognition method. This method consists of two techniques: Pose Parameter Embedding (PPE) and approximate nearest neighbor search based on Nearest First Traversing Graph (NFTG). PPE is a technique for image retrieval based object tracking, which stores misaligned object image with pose correction parameters for updating object position, scale, and orientation. PPE requires many misaligned images stored in the image database, and it reduces the scalability of the method. In this paper, we propose an extended method incorporating Chamfer matching with the NFTG based NN search. Since our method does not use PPE, it drastically reduces the number of images in the database. Also, it improves the stability against the illumination change, because Chamfer matching uses edge images. The asymmetric property of the Chamfer metric makes NFTG directed graph. Through the experiments, we confirmed the effectiveness of our method.

**Keywords** Chamfer Matching, Nearest First Traversing , NFTG , Object Tracking and Recognition

## 1 はじめに

従来コンピュータビジョンでは、画像空間内での対象の位置を探索する「追跡」と、パターン空間内で対象がどのクラスに属するかを調べる「認識」

の問題は、別々の問題として取り扱われてきた。しかし、現実には対象が何であるのかわからなければ、「追跡」を行うことはできず、逆に対象がどこにあるのかわからなければ、「認識」を行うことはできない。このように「追跡」と「認識」を同時に取

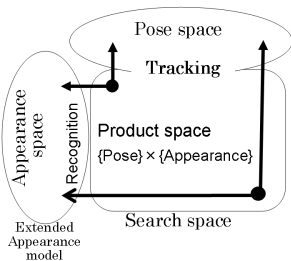


図 1: 直積空間

り扱うためには、図 1 に示すように、「対象の見えの空間」と「対象の位置・姿勢の空間」の直積空間内で探索を行うことが必要になる。しかし、この直積空間は極めて広大であり、対象の位置・姿勢と見えの全ての組み合わせの中から、与えられた画像に最もよくマッチする組み合わせを行うには処理コストが膨大になってしまう問題がある。

我々はこの問題を解決するために、近さ優先探索グラフ (NFTG) を用いた擬似的な最近傍探索 [1] により、画像上の対象の追跡と認識を同時に行う方法を提案している。この手法では、姿勢パラメータをあえてずらした画像を登録して、そこに埋め込まれたテンプレート修正パラメータを利用して対象枠を変形する姿勢パラメータ埋め込み (PPE) を行っている。

このため、

- 大量の画像データをデータベースに登録する必要がある。
- 対象とテンプレート枠がある程度重なっていないなければならない。
- 照明条件の変化に対して弱い。

などの問題点が生じる。そこで、これらの問題を解決するために Chamfer Matching を用いたエッジ画像のマッチングを NFTG に組み込む方法を提案する。

## 2 従来研究

### 2.1 姿勢パラメータ埋め込み

入力画像上で探索を行うことなく、対象の追跡を行う方法として、「姿勢パラメータ埋め込み」 (Pose Parameter Embedding:PPE) がある。PPE は、対象の「見え」に対象の「位置・姿勢の変化情報」を埋め込む方法である。具体的には、図 2(a) に示すように、対象のプロトタイプに加えて、あらかじめ平行移動、スケールなどを变化した画像を生成し、

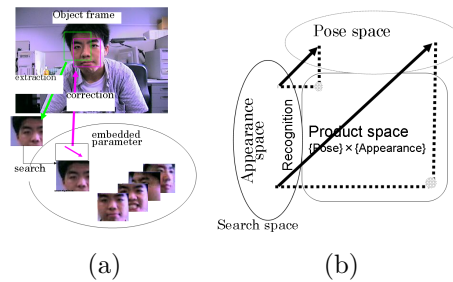


図 2: Pose Parameter Embedding の概要

それらもプロトタイプとして登録して、同時に変化パラメータも登録する方法である。

入力画像の部分画像から最近傍パターンを探索したときに、探索されたパターンが平行移動やスケール変化しているパターンであれば、その分だけ部分画像を切り出すフレームを移動させて、次のフレームから部分画像を切り出すことで、追跡を行うことができる。これにより、探索空間は同図 (b) のように対象の見えの空間に縮約されるため、効率よく探索を行うことが可能になる。

### 2.2 近さ優先探索グラフ

高次元でも構成可能であり、かつ効率的に擬似的な最近傍探索が可能なグラフとして、近さ優先探索グラフ (Nearest First Traversing グラフ:NFTG) が存在する。

NFTG が持つべき性質は、入力プロトタイプの一つであるという仮定のもとで、任意の頂点からスタートして、入力との距離が近づくようにグラフの頂点を辿っていくことで、入力と同じ学習データに到達できることである。このことより、入力がグラフの頂点の一つである場合は、正確な最近傍探索ができるが、それ以外の場合にもほぼ正しい最近傍探索が行える。また、直前の最近傍探索結果を次の初期位置として利用できるため、追跡のタスクに向けた方法であると言える。

## 3 NFTG を用いた対象追跡・認識の問題点

我々は追跡と認識の問題を同時に取り扱うために、2.1、2.2 で述べたような方法を提案しているが、Pose Parameter Embedding を利用すると、一つの対象に対して、複数の画像を生成する必要があるために、多数のクラスにおいて識別を行う場合には、大量の画像をデータベースに登録する必要が出てくる。このことは NFTG の構成時間、メモリの消費量の増加を引き起こし、登録できるプロトタイプの数にも制限を受ける。より多数のク

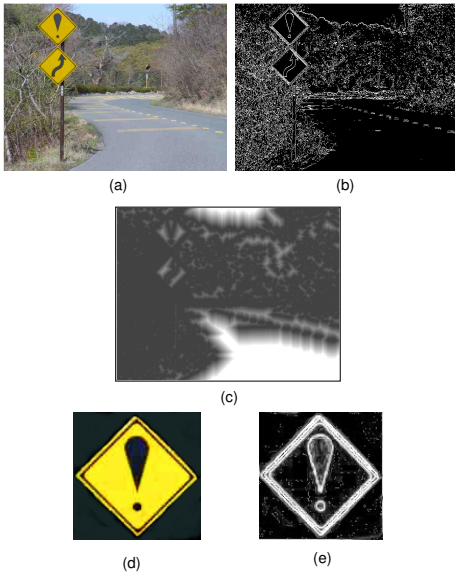


図 3: (a)original image(d)edge image (c)DT image (d)original template image(e)edge template image

ラスにおいての識別を行うには、この問題を解決する必要がある。

また、ずれの許容範囲が限られているために、対象とテンプレートがある程度重なっていないという問題がある。このことは、対象とテンプレートが一定距離以上離れると、追跡を再開することができないことを意味している。

さらに、入力画像とプロトタイプが色情報を基にした実画像であるために、照明条件の変化に弱いという問題もある。同じクラスに属している画像であっても、照明が変化することで、違うクラスであると認識される場合がある。

### 3.1 Chamfer Matching と有向近さ優先探索グラフの導入

これらの問題を解決するためには、ずれの影響による制限を受けることなく、効率的に対象の位置を探索する方法が必要となる。本論文では、距離変換画像と、エッジ画像のテンプレートをを用いる Chamfer Matching [2] を利用する。これは距離変換画像の画素値が最も近いエッジポイントまでの距離値になっているという性質と、このことから生じるサイドローブを利用することで、画像上を全探索することなく、効率的に対象の位置を探索する方法である。また、一度対象を見失った場合でも、再度マッチングを行うことで、追跡可能である。

さらに、この Chamfer Matching を NFTG に組み込む。このようにすることで PPE を用いる必要がなくなり、必然的にデータベースに登録する画像の量は減少し、多数のクラスにおいての識別も

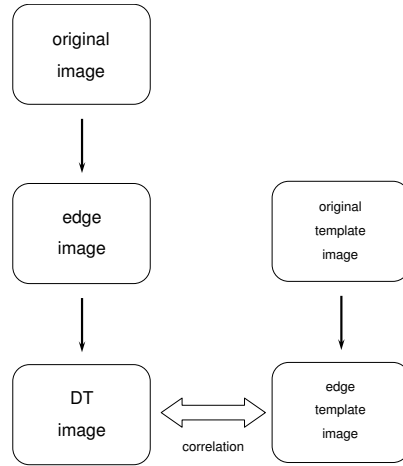


図 4: Chamfer Matching の概要

可能になる。またエッジ画像であるテンプレート画像を、NFTG のプロトタイプに設定することで、背景雑音や、照明変化に頑健な識別を行うことが可能になる。それに加え、NFTG の識別結果に伴い、随時 Chamfer Matching のテンプレートを更新することで、対象の追跡と認識を同時に行うことが可能になる。

## 4 Chamfer Matching

ここでは、Chamfer Matching の概要を述べる。

まず、逐次的距離変換アルゴリズム (sequential Distance Transformation algorithm) をベースにした既存手法 [3] を用いて、入力画像に距離変換を施す。その入力画像 (図 3(b))  $I(x, y)$  中の点  $(u, v)$  にエッジ画像のテンプレート (図 3(e))  $T_{w,h}(i, j) = \{(i, j) | 0 \leq i \leq w, 0 \leq j \leq h\}$  が重なっているとき、点  $(u, v)$  におけるテンプレートと入力画像の相違度  $D(u, v)$  は以下ようになる。

$$D(u, v) = \sum_{i=0}^w \sum_{j=0}^h [T_{w,h}(i, j) I(u+i, v+j)] \quad (1)$$

但し、式 (1) で求めた相違度は、距離変換画像とエッジ画像間の距離 (以下、Chamfer Distance) であるが、この値は距離の公理の一つである対称性を満足していない。詳細については 5 章で述べる。

次に、この相違度に対して、最急降下法を用いてテンプレートの移動先を決定する。

具体的には、点  $(u, v)$  における相違度  $D(u, v)$  に対して勾配  $grad(u, v)$  を求め、テンプレートの移動先である点  $(u', v')$  を以下のように決定する。

$$(u', v') = (u, v) - \alpha grad(u, v) \quad (2)$$

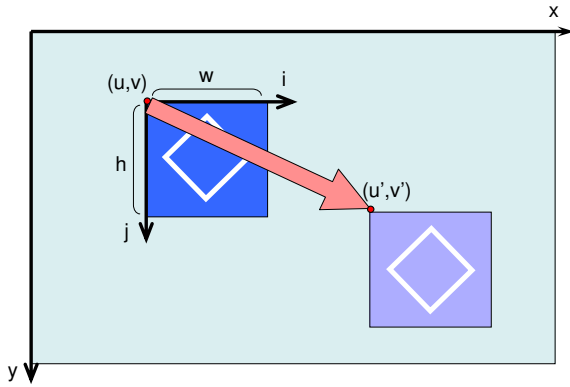


図 5: 入力画像とテンプレートの位置関係

また、式 (2) において、 $\alpha$  は不定係数であり、テンプレート  $T_{w,h}(i,j)$  内でのエッジポイント  $(i_e, j_e) (e = 0, 1, \dots, N)$  における勾配  $grad(i_e, j_e)$  の平均値を用いて、テンプレートの移動量にあたる  $\alpha grad(u, v)$  を以下の関係を満足するように設定する。

$$\alpha grad(u, v) = \frac{1}{N} \sum_{e=0}^N grad(i_e, j_e) \quad (3)$$

距離変換画像の画素値が最も近いエッジポイントまでの距離値になっているという性質から、式 (3) の条件を加えることにより、より効率的にテンプレートの移動を行うことが可能となる。

そして、テンプレートの移動先の点  $(u', v')$  が決定した際に、入力画像  $I(x, y)$  上にテンプレートと重なっている領域  $R_{w,h}(u', v') = \{(u', v') | 0 \leq u' \leq w, 0 \leq v' \leq h\}$  を新たに設定する。

この  $R_{w,h}(u', v')$  を NFTG の入力とし、プロトタイプ画像集合内で最近傍探索を行い、 $R_{w,h}(u', v')$  に最も近いとされたプロトタイプを新たなテンプレートとし、テンプレートの移動、相違度計算を繰り返し行う。

もし、テンプレートの移動量  $\alpha grad(u, v)$  の値が 0 になった場合、その地点における相違度  $D(u, v)$  を測り、その値が閾値以下ならば、画像上での対象の位置とする。もし、閾値よりも大きな値を示す場合は、テンプレートの移動先を新たに設定した後に、これらの処理を繰り返し行ってより相違度が小さくなる位置を探索する。

なお、本手法では式 (2) によってテンプレートの移動先が決定するたびに、テンプレートの更新を行っているが、このように頻繁にテンプレートの更新を行うことによって、

- 1 フレーム目でも確実に対象の検出と認識を行うことが可能。
- 識別回数を増やすことで認識制度が向上。

といった効果が現れる。

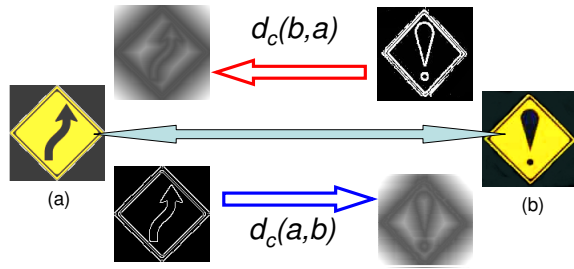


図 6: Chamfer Distance の概要

## 5 Directed NFTG

### 5.1 Chamfer Distance

まず、Chamfer Matching を NFTG に組み込むにあたって、Chamfer Distance の特異性を明らかにする。

一般に、ある集合  $S$  の要素  $p, q, r$  に対して、次の性質 (距離の公理と呼ばれる) を満足する関数  $d$  を距離 (distance) という。

1. 同一性:  $d(p, q) = 0 \Leftrightarrow p = q$
2. 対称性:  $d(p, q) = d(q, p)$
3. 三角不等式:  $d(p, r) \leq d(p, q) + d(q, r)$
4. 非負性:  $d(p, q) \geq 0$

図 6 において、画像 (a), (b) 間の Chamfer Distance ( $d_c(a, b)$ ) を考えた場合、 $d_c(a, b) \neq d_c(b, a)$  となり、Chamfer Distance は距離の公理の対称性を満足していないということになる。これは Chamfer Distance が、エッジ画像と距離変換画像のような異なった性質を持つ 2 種類の画像間の距離を示していることによる影響であると考えられる。

このことから、Chamfer Matching に NFTG を組み込むには、従来の NFTG において最近傍探索を行う際に、ユークリッド距離を用いて距離比較が行われていること、またユークリッド距離は言うまでもなく、対称性を満足していることから無向グラフで構成されている、という 2 点に対して改良を施す必要性が出てくる。

そこで、以下のことを定義する。

「Chamfer Distance はエッジ画像から距離変換画像への距離とする。」すなわち、画像 (a), (b) 間の Chamfer Distance ( $d_c(a, b)$ ) というのは、図 6 において、 $d_c(a, b)$  の方向 (エッジ画像から距離変換画像の向き) のみで成立するものとする。

またこの定義の下で、NFTG を有向グラフとすることを提案する。NFTG を有向グラフとすることで、距離比較に Chamfer Distance を用いることが可能になる。



## 5.2 Directed NFTG の構成法

以下、有向 NFTG(Directed NFTG : DNFTG) の構成アルゴリズムについて述べる。

辺の集合を  $E$ 、頂点集合を  $V$  とし、構成する DNFTG を  $G = (V, E)$  とする。また、 $x_i$  から  $x_j$  に近づくとき、 $x_i$  がスタックノードになることを  $Stuck(x_i, x_j)$ 、2 つのパターン  $x_i, x_j \in V$  間の Chamfer Distance を  $d_c(x_i, x_j)$  とする。

**Initialization:**  $E = \phi$

**Step1:** すべての  $x_s \in V$  に対して

$$C = \{x \in V | Stuck(x_s, x)\}$$

$$x_d = \arg \min_{x \in C} d(x_s, x)$$

$$E := E \cup (x_s, x_d) \text{ を実行する。}$$

**Step2:**  $G = (V, E)$  が単調性を満たすかどうか判定する。

満たすならば、グラフ構成終了。  
そうでなければ、Step1 へ。

スタックノードとは、任意の頂点から目的の頂点との距離が近づくようにグラフの辺を辿る途中で、より近いパターンへの辺が存在しなくなり、止まってしまう頂点のことをいう。また、NFTG が満足すべき条件として、以下の性質が挙げられる。

尚、これらの性質は NFTG を有向グラフとした場合でも満足させなければならない。

**連結性:** すべての頂点は、連結でなければならない。

**単調性:** 任意の頂点からスタートして頂点  $x_j$  を移動させるたびに、入力  $q$  と  $x_j$  との距離は、単調に減少しなければならない。

**極小性:** グラフのどの一辺でも取り除くと上記の二つの性質が満足されなくなる。すなわち、冗長性のないグラフでなければならない。

これらの性質のうち、単調性が満足されれば、連結性は満足される。なぜならば、連結でなければ単調性は満足されないからである。また、単調性を満たさないグラフにある辺を追加することによって単調性（と連結性）が満足されたとすると、少なくとも冗長なグラフにはなっていないと言える。グラフ構築時に追加する辺がある頂点に関する単調性を成立させるために必要不可欠な辺であれば、このようにして出来上がるグラフは極小性も満足すると言える。

したがって、任意の頂点に関して単調性を満足させることが本質であり、順に辺を追加していき、この性質が成立した時点で辺の挿入をやめれば、上記の 3 つの性質を満足するグラフが得られる。

以上のことから、Step2 において  $G = (V, E)$  が単調性を満たすかどうかを判定することで、NFTG が満たすべき条件を全て満たしているかどうかを判定することができる。また、DNFTG を構成するにあたって、Chamfer Distance を距離比較に用いるが、先に定義したように Chamfer Distance はエッジ画像から距離変換画像への画像間の距離であるので、構成時には 1 つの頂点に対してエッジ画像と、距離変換画像の 2 種類の画像を登録した上で、定義に基づいて方向に注意しながら、画像間の距離比較を行う必要がある。ここが、無向 NFTG の構成法との大きな違いである。

なお、構築された DNFTG で最近傍探索を行う際には距離変換画像は、入力画像中の  $R_{w,h}(u', v')$  を入力画像として利用するので、データベース中に距離変換画像を登録しておく必要はない。

## 6 実験結果

### 6.1 DNFTG の構築



図 7: 基本プロトタイプ画像 (1)

実際に、図 7 の道路標識画像を基にエッジ画像、距離変換画像としたものをプロトタイプ画像とし、DNFTG を構成した。すなわち、プロトタイプは  $120 \times 120$  のエッジ画像 10 種類と、 $120 \times 120$  の

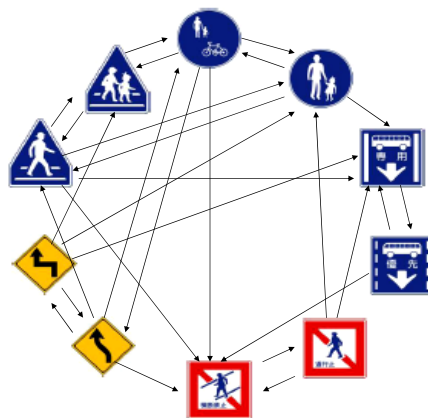


図 8: DNFTG

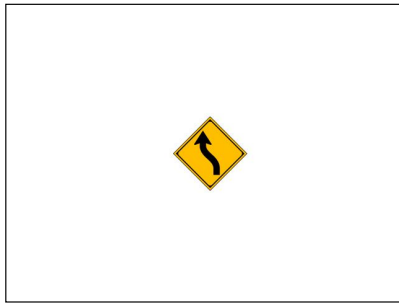


図 9: 入力画像

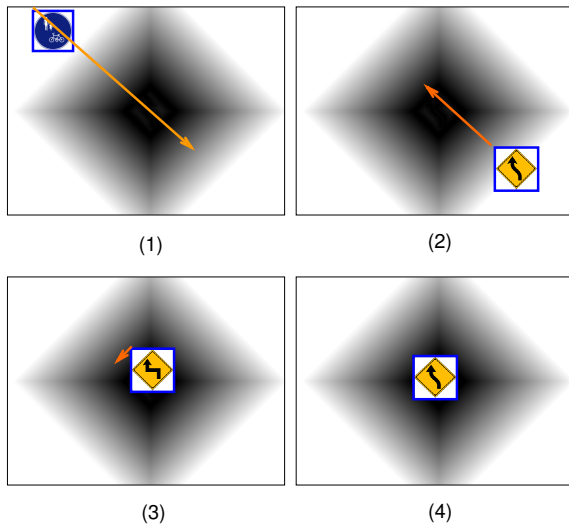


図 10: テンプレートの移動と変化

距離変換画像 10 種類であり、計 20 種類の 14400 次元の画像である。

図 8 に DNFTG の構成結果を示す。図 8 より、形状や内容が似通っているもの同士の間には、双方向グラフが構築されているが、それ以外の部分では単方向グラフが構築されていることが確認できる。

## 6.2 シミュレーション実験

本論文で提案した DNFTG の構成アルゴリズムと、それを用いた Chamfer Matching を評価するために、図 9(640 × 480) のような入力画像を用意し、6.1 で構成した図 8 の DNFTG を用いて、検出のシミュレーション実験を行った。実験は、Linux の動作する 3.2GHz の Pentium4 CPU, 1GB のメモリを搭載したデスクトップ PC 上で行った。

図 10 の (1)~(4) は、この実験における Chamfer Matching によるテンプレートの移動経路と、DNFTG によって探索されたその地点でのテンプレートの状態を示している。(1) の初期状態から Chamfer Matching を開始させ、テンプレートの位

置と状態を更新させながら、わずか 3 ステップでマッチングが完了 ((4) の状態) していることを示している。

(1) の状態ではテンプレートの位置、状態共に、入力画像内の対象とは全く違ったものとなっているが、(2),(3),(4) と探索を進めるにつれて正しいものへ近づくように変化していっている。

また、(1) の初期状態から (2) の状態へのテンプレート移動時に対象を飛び越えている現象が発生しているが、これは (1) の初期状態のテンプレートの位置が、対象から生じるサイドロープの範囲外にあるため、最急降下法を用いてテンプレートの移動先を決定することができず、(2) の状態のテンプレート位置へランダム移動したためである。

この結果より、画像空間内での対象の位置、パターン空間内での対象の見えにおいて、正しい探索が行われていると言える。なお、(1)~(4) 間の処理時間は 41msec である。表 1 にその処理時間の内訳を示す。

距離変換	16msec
マッチング (1 ステップあたり)	24msec (8msec)
計	41msec

表 1: 処理時間の内訳

距離変換に 16msec, Chamfer Matching と DNFTG を組み合わせたマッチングに 24msec がかかっており、この実験では 3 ステップでマッチングが完了していることから、1 ステップあたりのマッチングの処理時間は 8msec となる。

なお、この (1)~(4) の処理は実際の動画を用いた追跡実験において、第一フレームの処理と同等の処理である。

## 6.3 追跡実験

次に、実データを用いた追跡実験結果を示す。基本プロトタイプは図 11 の道路標識画像であり、それを基に 120 × 120 のエッジ画像 89 種類と、120 × 120 の距離変換画像 89 種類をプロトタイプとして、新たな DNFTG を構築した。その DNFTG を利用しカメラで撮影した平行移動を含む動画において追跡・認識の実験を行った。

その追跡・認識結果を図 12, 13 に示す。各画像の青い枠が対象として検出された位置を示している。これらの図から、いずれの場合も追跡・認識ともにほぼ正確に行われている事が確認できる。また、対象枠の中に背景雑音が含まれている場合でも、検出は可能である。なお、画像中には ETC と書かれた標識が 2 箇所存在しているが、これはプロトタイプには入っていない標識であるので、検出されていない。



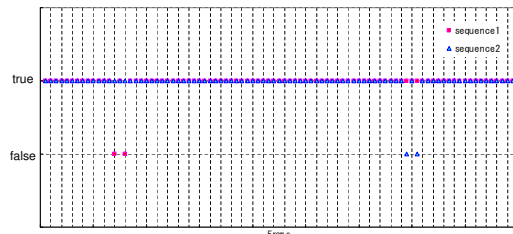
図 11: 基本プロトタイプ画像 (2)



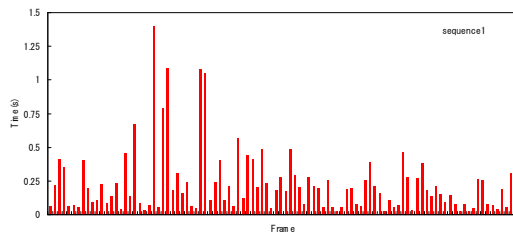
図 12: 追跡結果 (シーケンス 1)



図 13: 追跡結果 (シーケンス 2)



(a)



(b)

図 14: 各フレームに対する認識結果と処理時間

認識結果が正しいことを確認するために、図 14(a) に各シーケンスに対するフレーム毎の識別結果のグラフを示す。このグラフで、true は DNFTG を用いた最近傍探索で入力画像内の対象の標識と同じ標識が識別されていることを示し、false はそれ以外の結果が出ていることを示している。このグラフから、ほとんどのフレームで正しい識別が行われていることがわかる。また、この例の場合、最多識別回数で判断すれば、100% で標識の同定が行えている。

次に、この実験における追跡・認識時の各フレーム当たりの処理時間を図 14(b) に示す。このグラフは、シーケンス 1 の追跡・認識処理時間を表してい

る。このグラフから、フレームごとにかんりの処理時間の差が見受けられる。このシーケンスにおける処理時間の最短時間は 26msec であり、最長時間は 1.39sec である。また平均処理時間は 0.22sec となった。このような結果となった原因として、処理時間が長くなっているフレームでは、その前フレームにおいて DNFTG 内の最近傍探索の結果によって、テンプレートが本来の対象とは違ったものに更新されたために対象を検出するのに時間を要しているものと考えられる。

また、シミュレーション実験では、背景雑音が無いために、距離変換した際に対象から生じるサイドローブが非常に顕著に現れていたが、今回の実験では、ある程度背景に雑音が生じている影響で、対象から生じるサイドローブが他の雑音部分からのサイドローブによって打ち消されていることが影響しているとも考えられる。しかし、対象枠の内側にプロトタイプには無かった背景雑音がある場合でも、以前の NFTG のように検出できないといったことは無くなっている。

なお、今回の実験では、テンプレートの移動先が決定するたびに、DNFTG 内を最近傍探索し、その結果に応じてテンプレートを更新させながらマッチングを行っている(手法 1)。この点に関して、シーケンス 1 において、テンプレートの移動量が 0 になった場合のみ、DNFTG 内を最近傍探索し、テンプレートを更新する手法(手法 2)を用いて追加実験を行った。手法 1 と手法 2 のフレームごとの処理時間の比較を表 2 に示す。

	手法 1	手法 2
最短時間	26msec	26msec
最長時間	1.39sec	4.87sec
平均時間	0.22sec	0.34sec

表 2: テンプレート更新回数の違いによる処理時間の比較

表 2 から、どちらも最短時間は 26msec であるが、最長時間は 4.875sec と大幅に増加している。また平均処理時間についても増加している。このことから、頻繁に DNFTG 内を最近傍探索し、テンプレートを更新させながらマッチングを行うことが、4 章で述べた利点に加えて、処理時間の向上にも貢献しているといえる。

なお、以前の PPE を用いた手法では、今回行った平行移動を含む検出を行う場合、一つの基本プロトタイプにつき、縦方向の平行移動画像 7 種類、横方向の平行移動画像 7 種類の計 49 種類のプロトタイプの登録が必要となり、この実験の場合、 $49 \times 89 =$  計 4361 種類のプロトタイプの登録が必要となるが、本手法では約 50 分の 1 (DNFTG 構築時に関しては約 25 分の 1) のプロトタイプ数で同等の検出が行える。

## 7 おわりに

本論文では、NFTG と PPE を用いて画像上の対象の追跡と認識を同時に行う方法における問題点を解決するために、以下の方法を提案した。

- PPE を用いることによるずれの制限を受けることなく、効率的に対象の位置を探索するために、エッジ画像と距離変換画像を用いる Chamfer Matching を利用する。
- 照明条件の変化や背景雑音に頑健な追跡と認識を行うために、Chamfer Matching を DNFTG に組み込む。

シミュレーション実験では、DNFTG と Chamfer Matching の組み合わせによって、画像空間内での対象の位置、パターン空間内での対象の見えにおいて、正しい探索が行われていることを確認した。また、実データを用いた実験では、平行移動に対する追跡に関して、追跡と認識を同時に行えることを示した。しかし、処理時間においてはまだまだ実用的なものとなっていない。

今後の課題として、次の二点を挙げる。一つ目は処理時間の向上であり、リアルタイム処理を行えることが目標である。二つ目は、現時点では平行移動のみの追跡だが、スケール変化やアフィン変換への対応をすることで、さまざまな条件に適應させることである。

## 参考文献

- [1] 坂垣内, 和田, 加藤: “近さ優先探索グラフを用いた対象追跡・認識”, 画像の認識・理解シンポジウム (MIRU2005), pp. 16 - 23, 2005.
- [2] Gunilla Borgfors: “Hierarchical Chamfer Matching: A Parametric Edge Matching Algorithm”, IEEE TRANSACTIONS ON PATTERN AND MACHINE INTELLIGENCE, Vol. 10, No. 6, pp. 849 - 865, November 1988.
- [3] D.M. Gavrila: “Multi-feature Hierarchical Template Matching Using Distance Transforms”, IEEE International Conference on Pattern Recognition, Brisbane, Australia, pp. 439 - 444, 1998.