

1 ms ビジュアルフィードバックシステムのための 高速対象追跡アルゴリズム

石 井 抱* 石 川 正 俊*

High Speed Target Tracking Algorithm for 1 ms Visual Feedback System

Idaku Ishii* and Masatoshi Ishikawa*

Most conventional visual feedback systems using CCD camera are restricted by video rates and therefore cannot be adapted to the changing environment sufficiently quickly. To solve this problem we developed a 1 ms visual feedback system using a general purpose massively parallel processing vision system in which photo-detectors and processing elements are directly connected. For high speed visual feedback fast image processing algorithms are also required. In particular, the difference of images between frames is very small in our system because of its high speed frame rate. Using this feature, we can realize several image processing techniques by simpler algorithms.

In this paper we propose a simple algorithm for target tracking using the feature of high speed vision, and realize target tracking on the 1 ms visual feedback system.

Key Words: Massively Parallel Processing, High Speed Frame Rate, Image Processing Algorithm, Vision Chip

1. は じ め に

近年、未知環境下でのロボット制御方法として、外界を認識しながら、同時に外界の変化に対応できるビジュアルフィードバックが注目されており[1]、その高速化を実現するためにビジョンシステムの高速リアルタイム化が求められている[2]。しかし、従来のビジョンシステムの多くは、ビデオフレームレート(30 [Hz] or 60 [Hz])で代表される視覚センサと処理装置間の入出力ボトルネックが存在し、高速化に限界があった。この問題に対して、石川らはセンサと処理要素を画素ごとに直結した汎用超並列・超高速ビジョンチップアーキテクチャを提案し[3]、実際にそのアーキテクチャを用いてロボット制御を実現した[4]。

高速なビジュアルフィードバックを実現するためには、このようなハードウェアの高速化と同時に、画像情報から環境を理解し、その情報を抽出するための画像処理アルゴリズムの高速化が重要となる。特に高速な画像入力をもつ超並列・超高速ビジョンでは、アルゴリズムの高速化が直接フィードバックレートの向上につながり、高速なフィードバックレート内に実行可能なアルゴリズムであることが求められ、その高速化の要求は従来に比べて厳しいものとなる。しかし、アルゴリズムの高速化の多くは、画像入力としてビデオフレームレートに従ったものを暗に前提し、アーキテクチャの並列性に着目した並列画像

処理に関するものであり[5]、高速な画像入力に対して必ずしも効率的とはいえない。

このような超並列・超高速ビジョンチップでは、フレームレートが非常に高速であるために、従来考えられた動画像シーケンスに比べ、フレーム間の画像変化が微小であり、画像情報の冗長性が特に時間方向について非常に大きい。そのため、従来の枠組みで提案されたアルゴリズムの多くは、冗長な画像情報に対応して、無駄な処理を行うと考えられる。言い換えれば、高速な画像入力を前提にすることにより、従来の画像処理技術が簡単なアルゴリズムで実現可能である[6][7]。このようなアルゴリズムは、ロボット制御に必要とされる ms オーダーの制御を直接視覚情報を用いて制御可能とし、実際に高速な障害物回避などを行う際に用いられており[8]、その有効性が示されている。

そこで本論文では、超並列・超高速ビジョンにおける高速な画像入力に着目し、これを超高速ビジョン系の特徴として捉え、その特徴を用いた超高速ビジョン画像処理アルゴリズムの可能性および有効性を、対象追跡を例にして議論を行う。具体的には、2 値画像の場合について超高速ビジョン系における画像の特徴を記述し、特に高速ビジュアルフィードバックを実現する際に対象と環境を分離するために重要となる、動画像における対応づけを高速に行うアルゴリズムとして、Self Windowing を提案し、この方法を用いた対象追跡アルゴリズムを 1 ms ビジュアルフィードバックシステム上で実際に動作させることによりその有効性を示す。

原稿受付 1997 年 8 月 11 日

*東京大学大学院工学系研究科

*Graduated School of Engineering, Univ. of Tokyo

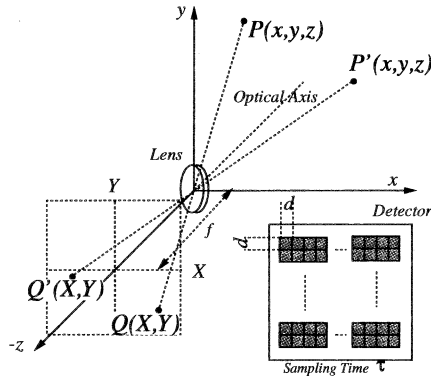


Fig. 1 Pin-hole camera model

2. 超高速ビジョン系における2値画像の特徴

最初に、対象の動きとフレーム間の画像の変化の関係について論じ、高速なフレームレートをもつビジョンシステムがどのような特徴をもつかを考察する。その議論の上で、超高速ビジョン系における2値画像の特徴について論じる。

2.1 サンプル時間と画像の変化の関係

ビジョンシステムとしてピンホールカメラ系を仮定する。カメラの焦点距離を f とし、Fig. 1 のようにレンズの光軸が z 軸、レンズの中心が原点となる xyz 座標系を考え、画像面において、 X, Y 軸がそれぞれ x, y 軸に平行であり、原点がレンズの光軸上にある XY 座標系を考える。画像面上にはフォトディテクタ (PD) が二次元アレイ状にあり、そのPDピッチを d 、画像のサンプル時間を τ とする。

対象の動きをレンズ系からみた対象の画角の変化とし、サンプル時間あたりの画像面上での対象の像 $Q(X, Y)$ の動きとの関係を考察する。

対象の画角 $\theta_x = \tan^{-1}(x/z)$, $\theta_y = \tan^{-1}(y/z)$ と画像面での対象の像 $Q(X, Y)$ は、透視変換の式より次の関係が成立する。

$$X = -f \tan \theta_x \quad Y = -f \tan \theta_y \quad (1)$$

式 (1) のそれぞれの両辺を時間微分することにより、サンプル時間あたりの画像面上の対象に対応する像の動き $D_X, D_Y [\text{pixel}]$ と対象の画角の変化 $(\omega_x, \omega_y) = (\dot{\theta}_x, \dot{\theta}_y)$ との関係は次の式で与えられる。

$$D_X = \frac{f\tau}{d} (1 + \tan^2 \theta_x) |\omega_x| \quad (2)$$

$$D_Y = \frac{f\tau}{d} (1 + \tan^2 \theta_y) |\omega_y| \quad (3)$$

ここで、対象の画角およびその変化に次のような制限

$$\max\{|\theta_x|, |\theta_y|\} \leq \theta_m \quad (4)$$

$$\max\{|\omega_x|, |\omega_y|\} \leq \omega_m \quad (5)$$

が与えられた場合を考える。ただし θ_m はカメラがとりうる画角の最大値、 ω_m は対象の画角変化がとりうる最大値である。

Table 1 Displacement of image per frame

	CCD camera (30 [Hz]) $\omega_K = 0.015 [\text{rad/s}]$	Vision Chip (10 [kHz]) $\omega_K = 5.0 [\text{rad/s}]$
1.0 [m/s] $\omega_m = 0.10 [\text{rad/s}]$	7 [pixels]	1 [pixel]
40 [m/s] $\omega_m = 4.0 [\text{rad/s}]$	260 [pixels]	1 [pixel]

このとき、 D_X, D_Y のとりうる上限 D_m は、次のようになる。

$$D_m = \frac{f\tau}{d} (1 + \tan^2 \theta_m) |\omega_m| = \frac{|\omega_m|}{\omega_K} [\text{pixel}] \quad (6)$$

ここで、

$$\omega_K = \frac{d}{f\tau} (1 + \tan^2 \theta_m) \quad (7)$$

は、ビジョンシステムにより決定される定数であり、対象の動きがビジョンシステムの画像面における最小単位、つまり1画素以内になる最大の角速度である。特に、観測する対象の画角の変化の絶対値がこの定数よりも小さい場合は、実際に観測される対象の動きの大きさの変化は0または1[pixel]のみとなる。これはサンプル時間の高速化により、フレーム間の画像の変化において慣性力が支配的になることに関連する特徴である。

例として、ビジョンシステムのパラメータとして標準のCCDカメラをもつものを考え、焦点距離 $f = 20 [\text{mm}]$ 、PDピッチ $d = 2.0 \times 10^{-2} [\text{mm}]$ のシステムから $z = 10 [\text{m}]$ 先で運動する対象を観測する場合を考える (画角は最高 45° としている)。このとき、人の歩く速度 (1.0 [m/s])、野球の投手の速球の速度 (40 [m/s]) の二つの例において、ビデオフレームレート ($1/\tau = 30 [\text{Hz}]$)、超並列・超高速ビジョンチップ上で実現可能なフレームレート ($1/\tau = 10 [\text{kHz}]$) の場合の D_m を Table 1 に記す。この10 [kHz] というフレームレートについては、後述する1 ms ビジュアルフィードバックシステム [4] 上で実際に動作することを確認している。

この例からも分かる通り、フレームレートを高速化することにより、高速に運動する多くの対象に対しても、画像面での対象の動きが微小になり、1画素以下に抑えることができる。

2.2 超高速ビジョン系の定義

次に2値画像における超高速性を数学的に定義し、その定理を導出する。なお取り扱う画像は、時間、空間方向にすでに離散化されたものであるとする。

ここで、離散化された時刻 $k \in I$ (以降では、単に時刻と呼ぶ) における2値画像集合を C_k とし、特に点 (画素) を p_k とおく。点 p_k の近傍領域を $N(p_k)$ とする。近傍には、4近傍、8近傍などが考えられる。このような近傍領域を用いて、次のような画像集合を定義する。

$$C_k \text{ の dilation 画像 } :D_1(C_k) = \bigcup_{p_k \in C_k} N(p_k) \quad (8)$$

$$C_k \text{ の erosion 画像 } :E_1(C_k) = \bigcap_{p_k \in C_k} N(p_k) \quad (9)$$

これらの操作を N 回繰り返したものを、それぞれ C_k の N -dilation 画像, N -erosion 画像とし, $\mathcal{D}_N(C_k), \mathcal{E}_N(C_k)$ と記述する. これらの議論の詳細については, 数学的モルフォロジの本に譲る [9].

前節の議論より, 点の運動についてフレームレートが高速化するにつれて, 移動画素量 D_m は小さくなり, ビジョンシステムの最小単位である 1 以下となる. そこで本論文では超高速ビジョン系の定義を, 2 値画像に関して次のように行う.

[超高速ビジョン系の定義]

任意の時刻 $k \in I$ において, 画像集合 C_k に属するすべての点 p_k について移動画素量 $D_m(p_k)$ が常に 1 より小さいとき, 画像集合 C_k は超高速ビジョン系に属するとする.

$$D_m(p_k) \leq 1 \quad (p_k \in C_k) \quad (10)$$

次に超高速ビジョン系のいくつかの定理を示す. ただし, 画像集合 C_k は超高速ビジョン系に属するとする.

[定理 1]

任意の時刻 $k \in I$ において, 画像集合 C_k の dilation 画像 $\mathcal{D}_1(C_k)$ は, 時刻 $k+1, k-1$ の画像 C_{k+1}, C_{k-1} を含む.

$$C_{k+1}, C_{k-1} \subset \mathcal{D}_1(C_k) \quad (11)$$

この定理は, 定義を p_k の近傍集合 $N(p_k)$ による記述, つまり, すべての $p_k \in C_k$ について $p_{k+1}, p_{k-1} \in N(p_k)$ と置き換え, この両辺を C_{k+1}, C_{k-1} でまとめることにより導出される.

[定理 2]

任意の時刻 $k \in I$ において, 画像集合 C_k の erosion 画像 $\mathcal{E}_1(C_k)$ は, 時刻 $k+1, k-1$ の画像 C_{k+1}, C_{k-1} に含まれる.

$$C_{k+1}, C_{k-1} \subset \mathcal{E}_1(C_k) \quad (12)$$

この定理は C_k の補画像集合 $\overline{C_k}$ が超高速性ビジョン系に属することにより導出される.

[画像集合の最大速度]

以降の議論で超高速ビジョン系に属する場合とそうでない場合を明確にするために, 画像集合のフレームあたりの最大画素移動量を画像集合の最大速度とし, 任意の時刻 k において, ある整数 $M \in I$ に対し,

$$C_{k+1}, C_{k-1} \subset \mathcal{D}_M(C_k) \quad (13)$$

$$C_{k+1}, C_{k-1} \subset \mathcal{E}_M(C_k) \quad (14)$$

を満たすとき, この条件を満たす画像集合 C_k の最大速度は M [pixel/frame] 以下であると定義する. 超高速ビジョン系に属するときは, この最大速度が 1 [pixel/frame] である.

3. Self Windowing

次に, このような画像の超高速性を考慮した画像処理アルゴリズムの例として, 複数の対象情報を抽出するための動画像対応付けを高速に実現可能とする Self Windowing を提案する.

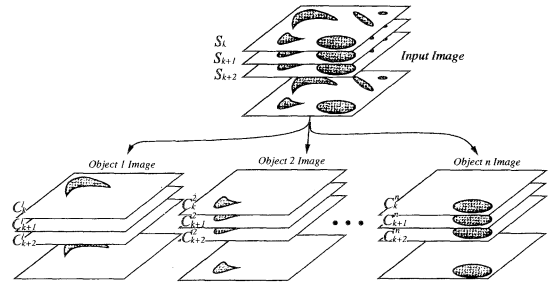


Fig. 2 Corresponding problem in this paper

3.1 実現すべき動画像対応付け問題

時刻 k における入力画像 S_k は, 2 値画像とし, 対象が n 個写っているものを考え, 時刻 k における $i (i = 1, \dots, n)$ 番目の対象画像を C_k^i とおく. 対象同士には重なりが生じていないとする.

$$S_k = C_k^1 \cup C_k^2 \cup \dots \cup C_k^n = \bigcup_{i=1}^n C_k^i \quad (15)$$

$$C_k^i \cap C_k^j = \emptyset \quad (i \neq j) \quad (16)$$

本論文で論じる動画像対応付け問題は, Fig. 2 のように, このような入力画像 S_k が与えられたとき, すべての時刻 k に対して, n 個の対象画像 $C_k^i (i = 1, \dots, n)$ に分解することである.

このような動画像の対応づけは, 環境と制御対象を分離し, それらの相互関係をもとにビジュアルフィードバックを行う場合に不可欠な処理である. また, この処理は, フィードバックレートと同程度の速度で実現される必要があり, アルゴリズムの高速化が必要となるものである.

3.2 Self Windowing

このような対応付けを簡単に実現する方法として, Self Windowing を提案する. このアルゴリズムは, 時刻 $k+1$ における $i (i = 1, \dots, n)$ 番目の対象画像 C_{k+1}^i を, 時刻 k における i 番目の対象画像 C_k^i をもとにしたウィンドウ画像 $W(C_k^i)$ と入力画像 S_{k+1} の AND を式 (17) のようにとることにより計算するものであり, ウィンドウ画像が 1 フレーム前の対象画像自体から生成するため, Self Windowing と名前を付けている.

$$C_{k+1}^i = S_{k+1} \cap W(C_k^i) \quad (17)$$

このアルゴリズムの特徴は, 従来のアルゴリズムで行われた (1) フレーム内の連結成分抽出のための反復演算, (2) フレーム間における対象の対応付けを行うための反復演算における冗長な部分を一体化し, 空間方向の領域分割と時間方向の対応付けを統一化することにより, 至って単純な処理で実現されている点にある.

Self Windowing でのアルゴリズムの単純化は, 動作可能であるための条件が厳しくなること, 特にフレームレートの高速化に対する要求とトレードオフの関係にある. そこで, 次に Self Windowing 可能な条件として, (1) 対象画像 $C_k^i (i = 1, \dots, n)$ の条件, (2) ウィンドウ画像 $W(C_k^i) (i = 1, \dots, n)$ の設定条件について論じ, Self Windowing が超高速ビジョンに適したアルゴリズムであることを述べる.

3.3 Self Windowing 可能な条件

Self Windowing 可能であるとは、任意の時刻 k において、すべての対象画像 i について式 (17) が成立することであり、式 (15) を代入し整理した次式が任意の時刻 k で成立することである。

$$C_{k+1}^i = \bigcup_{j=1}^n (C_{k+1}^j \cap W(C_k^i)) \quad (18)$$

ここで式 (16) より、次式が成立する。

$$(C_{k+1}^j \cap W(C_k^i)) \cap C_{k+1}^i = \emptyset \quad (j \neq i) \quad (19)$$

これにより式 (18) は i, j に関する直交性により、次の二つの条件に分解することができる。

$$\text{自己包含条件} \quad C_{k+1}^i \cap W(C_k^i) = C_{k+1}^i \quad (20)$$

$$\text{排他的条件} \quad C_{k+1}^j \cap W(C_k^i) = \emptyset \quad (j \neq i) \quad (21)$$

が任意の i, j, k について成立することがセルフウィンドウイングが可能である条件である。

3.3.1 自己包含条件

式 (20) で表される自己包含条件は、ウィンドウ画像の設定に関する条件である。この条件はその名前の通り、次の時刻の対象画像を完全に包含するための条件であり、次のように変形される。

$$C_{k+1}^i \subset W(C_k^i) \quad (22)$$

ここで、画像におけるフレームあたりの動きとの関係をみるために、すべての対象 $C_k^i (i=1, \dots, n)$ の最大速度を M とする。このとき、式 (14) を満たすので、自己包含条件を満たすウィンドウ画像の十分条件は次のように与えられる。

$$W(C_k^i) \supset \mathcal{D}_M(C_k^i) \quad (23)$$

自己包含条件を満たすウィンドウ画像 $W(C_k^i)$ は、最低でも $\mathcal{D}_M(C_k^i)$ 以上の大きさをもつ必要があり、 M が小さいほど、ウィンドウ画像を生成するための処理量が小さくなるといえ、特に M が最小値の 1 をとる超高速ビジョン系では近傍演算レベルの処理でウィンドウ画像を生成することが可能となる。

3.3.2 排他的条件

式 (21) で示される排他的条件は、対象画像 $C_k^i (i=1, \dots, n)$ が満たす条件である。これは、Self Windowing を行うときに対象を誤認しないための条件であり、任意の $i(\neq j)(=1, \dots, n)$ について次式が成立することである。

$$C_{k+1}^j \subset \overline{W(C_k^i)} \quad (24)$$

自己包含条件同様、すべての対象 $C_k^i (i=1, \dots, n)$ の最大速度を M としたとき、式 (14) および自己包含条件の式 (23) より、ウィンドウ画像 $W(C_k^i)$ について次の式が成立する。

$$W(C_k^i) \subset \mathcal{D}_{m+M}(C_{k+1}^i) \subset \mathcal{D}_{2M}(C_{k+1}^i) \quad (25)$$

ここで、 $m = \max_{k \in I} \{m : W(C_k^i) \subset \mathcal{D}_M(C_k^i)\}$ である。

これより、排他的条件を満たすための十分条件は、対象間の距離 $d(C_k^i, C_k^j)$ について次式が成立することといえる。

$$d(C_k^i, C_k^j) \geq m + M \geq 2M \quad (26)$$

このように、常に排他的条件を満たすためには、対象間の距離は $2M$ より大きい必要があるので、 M が大きくなるにつれて対応づけにおける空間的な精度が落ちるといえる。これに対し、 M が最小値の 1 となる超高速ビジョン系では、この対象画像同士間の距離を 2 まで抑えることができる。

3.4 最小 Self Windowing

前述した Self Windowing を可能とするための二つの条件において、いずれも M が小さいほどよく、Self Windowing は超高速ビジョン系に適したアルゴリズムであると考えられることができる。

そこで本論文では、特に超高速ビジョン系において、これらの条件を満たす最小なウィンドウ $W(C_k^i) = \mathcal{D}_1(C_k^i)$ による Self Windowing を最小 Self Windowing と呼ぶ (Fig. 3)。

$$C_{k+1}^i = S_{k+1} \cap \mathcal{D}_1(C_k^i) \quad (27)$$

ただし、任意の $i, j (j \neq i)$ について次の関係が成立するとする。

$$d(C_k^i, C_k^j) \geq 2 \quad (28)$$

この処理は、AND 演算 1 回と、dilation 操作 1 回という極めて単純な処理で実現されており、超並列・超高速ビジョンでは、画像のサイズに依存しない、オーダー $O(1)$ での高速な処理が実現することができる [10]。

以降、最小 Self Windowing を単に Self Windowing と呼ぶ。

3.5 例外処理について

これまでの議論では S_k, C_k^i が自己包含条件や排他的条件を満たされるとしてきたが、実際にはいくつかの例外がある。これらの例外も考慮して、対応付けを実現するためには、Self Windowing による対象画像の抽出だけではなく、例外検出アルゴリズムの高速化が重要となる。そこで次に、考えられる例外を示した上で、その例外検出を、超高速ビジョン系の条件に対する破綻を逆に利用することにより、簡単化されたアルゴリズムについて述べる。

3.5.1 入力画像 S_k が自己包含条件を満たさない場合

(1) 画面内に新たに対象が出現する場合

ビジョン系の画角内に対象が出現する場合や、照明や画素間隔の関係で出現する場合が考えられる。いずれの場合も、次の

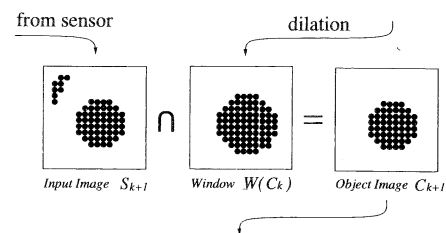


Fig. 3 Self windowing

ような処理を行えば、新たに出現した対象 C_{k+1}^{appear} の出現を検出できる。

$$S_{k+1} \cap \overline{D_1(S_k)} = \begin{cases} C_{k+1}^{appear} & (\text{対象出現時}) \\ \emptyset & (\text{それ以外}) \end{cases} \quad (29)$$

(2) 画面内から対象が消失する場合

ビジョン系の画角から対象が外れる場合や、照明や画素間隔の関係で消失する場合が考えられる。いずれの場合も、次のような処理を行えば、対象の消失を検出できる。この状態は、対象が出現する場合と時間を逆向きに考えた場合と同じなので、次のように対象 $C_k^{disappear}$ の消失を検出することができる。

$$S_k \cap \overline{D_1(S_{k+1})} = \begin{cases} C_k^{disappear} & (\text{対象消失時}) \\ \emptyset & (\text{それ以外}) \end{cases} \quad (30)$$

3.5.2 対象画像 C_k^i が排他的条件を満たさない場合

(1) 対象同士が衝突する場合

Self Windowing の条件であった排他的条件は、常に満たされる訳ではなく、実際には対象同士が重なるなどして、誤認される場合がある。本論文では、このような誤認が起こる状態を対象同士が衝突するとし、以下にこの衝突を検出する方法を述べる。

時刻 $k+1$ において、対象 i に対象 $j (\neq i)$ が衝突した場合、Self Windowing を繰り返して行くと、時刻 k での Self Windowing により得られた対象 $i (i=1, \dots, n)$ の画像を T_k^i として、ある時刻後 $k+a$ 後には、本来抽出したい対象画像 C_{k+a}^i ではなく、対象 j も含んだ画像 $T_{k+a}^i = C_{k+a}^i \cup C_{k+a}^j$ が得られる。この過程で、Self Windowing により得られた画像の面積は急激に変化するが、このような変化は、排他的条件が破綻していない限り生じない。

そこで、この現象を用いて衝突を次の式が成立した場合として検出することができる。

衝突した対象を取り込むために十分な時間を a 、 \sum を画像の総和の計算記号、 c をしきい値としている。

$$\sum T_{k+a}^i - \sum T_k^i > c \quad (31)$$

この処理については、後述する 1 ms ビジュアルフィードバックシステム上での対象追跡実験で実装している。

(2) 一つの対象が複数の対象に分離する場合

一つの対象から複数の対象が分離した場合、Self Windowing により抽出される画像には分離したすべての対象が含まれる。このとき分離前と、分離後では対象の数に相当する連結成分数が増加するので、この変化をみることであれば、追跡対象が分離したかどうかを検出できる。

連結成分の数は、画像の連結成分数から孔数を引いたものであるオイラー数から計算可能である [11]。オイラー数は近傍演算と総和演算により簡単に計算されるものであり、例えば画像集合 C_k の 4-連結性のオイラー数 e_k の計算は次のように行われる。

$$e_k = s_k - t_k + u_k \quad s_k = \sum C_k(i, j) \\ t_k = \sum (C_k(i, j) \cap C_k(i+1, j))$$

$$u_k = \sum (C_k(i, j) \cap C_k(i+1, j) \\ \cap C_k(i, j+1) \cap C_k(i+1, j+1))$$

孔数の変化が生じないと考えた場合、オイラー数の変化が連結成分の数の変化と等しいといえるので、時刻 k におけるオイラー数 e_k について、

$$e_k - e_{k-1} > 0 \quad (32)$$

であれば、分離が生じたことを検出できる。

分離後の新たな対象の再認識については、連結成分抽出アルゴリズムを利用したラベリングを行い、衝突時の対象画像の特徴パラメータとラベリングされた画像のそれが一番近いものを、同一対象として対応づけを行い、もし対応がとれない場合には、新たな対象として考える [7]。

これらのオイラー数の計算や、対象の再認識における連結成分抽出については、後述する 1 ms ビジュアルフィードバックシステム上での対象追跡実験で実際に実装しており、超並列・超高速ビジョンチップ上で実現可能なものである。

4. 対象追跡実験

最後にこの Self Windowing を用いて、超並列・超高速ビジョンシステムである 1 ms ビジュアルフィードバックシステム上で、実時間対象追跡を実現し、その有効性を示す。

4.1 対象追跡アルゴリズム

提案するアルゴリズムは、複数の対象の中から一つの対象を抽出し、その追跡対象の画像の重心の位置をセンサの中心に制御するもので、たとえ、対象に衝突・分離が起こっても対応できるものである。このアルゴリズムは、Fig. 4 のように、追跡対象の抽出 (Self Windowing)、追跡対象と他の対象との衝突の検出、追跡対象の分離の検出、分離が生じたときの追跡対象の認識の四つの部分により構成されている。なお、追跡対象の認識における特徴パラメータとして画像の面積を用いた。これらの部分のアルゴリズムについてはすでに説明しているので、詳細は省略する。

4.2 1 ms ビジュアルフィードバックシステム

1 ms ビジュアルフィードバックシステムは、超並列・超高速ビジョンチップアーキテクチャ SPE (Sensory Processing Element) [3] を採用したロボット制御を実現する応用システムである [4]。16×16 の PD アレイと画素ごとに直結した処理要素が完全並列で処理した結果により、チルト・パンの 2 自由度を

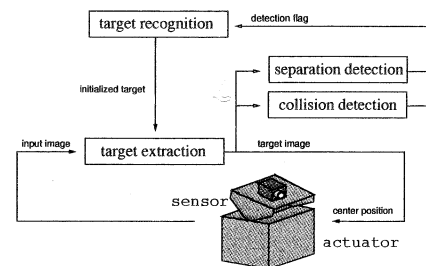


Fig. 4 Target tracking algorithm

もつアクティブビジョンやマニピュレータを、1 [ms] 以下のサイクルタイムで制御可能である。

処理部分である SPE は集積化を念頭において、汎用性と高速性を維持したまま PE の回路規模を小さくするために、SIMD 型制御、4 近傍接続、マイクロインストラクション方式、ビットシリアル演算を採用している。PE には 24 bit レジスタ、AND, OR, XOR, ADD が計算できるビットシリアル ALU 及び乗算器があり、様々な初期視覚処理の高速な動作を確認している。

インストラクションは、開発の簡便さなどから並列処理 DSP (TMS320C40) から送出速度に制限がある Parallel I/O を介して SPE-256 へ送られているため、現在 13 [μ s] の速度で動作しているが、ビジョンチップ専用コントローラ [12] を用いることにより、最高 100 [ns] の速度で実現可能なものである。

4.3 アルゴリズムの速度評価

提案した対象追跡アルゴリズムを 1 ms ビジュアルフィードバックシステム上で動作させたときの実行ステップ数および実行時間 () 内は実現可能な最高速度) を述べる。

追跡対象を抽出する Self Windowing は 23 [steps] = 300 [μ s] (2.3 [μ s])。衝突の検出は総和演算 1 回 = 480 [μ s] (0.9 [μ s])、分離の検出は総和演算 3 回 + 20 [steps] = 1.7 [ms] (4.7 [μ s]) で実現され、これらを組み合わせたものが総和演算 4 回 + 43 [steps] = 2.5 [ms] (7.9 [μ s]) で実現できた。

現在、総和演算は出力をダンプし DSP 上で行っているため、演算時間が大きいものとなり、アルゴリズムの実行時間の多くを占めている。このような問題に対し、坂口らは超並列・超高速ビジョンチップの並列出力に対応した総和演算回路 [13] を設計・開発しており、上記の最高速度はこの総和演算回路を SPE と合わせて実装した場合に実現可能な数字として記している。この回路による総和演算は、 $2^N \times 2^N$ の 2 値画像について計算するとき、SPE の処理ステップに対し $(2N + 1)$ [steps] かかる。

追跡対象の分離が検出されたときに行う追跡対象の認識については、反復計算を用いたものであり、他の部分に比べ高速なアルゴリズムとはいえないが、常に実行される必要があるものではなく、極端な高速性は必要とされない。

4.4 追跡実験結果

提案した対象追跡アルゴリズムを 1 ms ビジュアルフィードバックシステム上で動かした結果を次に示す。

実験としては、固定された障害となる対象の上方を、水平方向に動く対象を追跡することを考え、(1) 追跡対象と障害となる対象が衝突しない場合 (Fig. 5 (a))、(2) 途中で追跡対象と障害となる対象が衝突および分離が生じた場合 (Fig. 5 (b)) の二つのケースについて実験を行った。なお、どちらのケースにおいても、画像のサンプリング時間あたりの移動画素量は 1 [pixel] 以下である。

4.5 衝突・分離のない場合の追跡

最初に、提案したアルゴリズムのうち、Self Windowing による追跡対象の抽出のみを実装して、対象追跡が実現した。なお制御サイクルタイムは 1 [ms] である。

実験結果として Fig. 6 に、入力画像、Self Windowing による抽出された対象画像、対象追跡時のアクチュエータの軌跡 (1 ms ビジュアルフィードバックシステムのアクティブビジョ

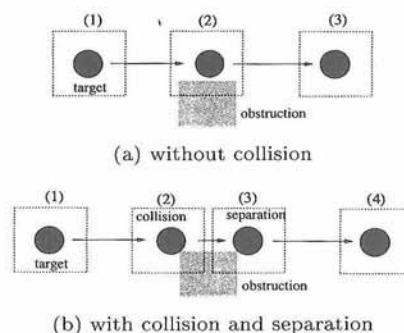


Fig. 5 Experiments of target tracking

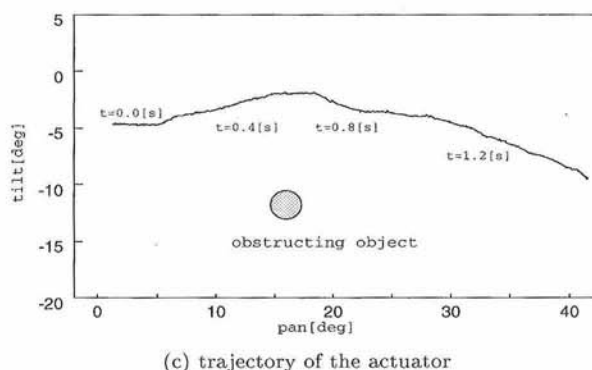
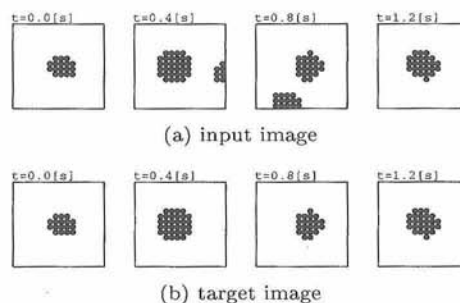


Fig. 6 Tracking results without collision

ンの視線方向に相当する) を示す。なお、アクチュエータの軌跡のグラフには、参考のため、固定された障害となる対象の位置を記している。

この結果を見ても分かる通り、画面内に追跡対象以外に障害となる対象が見えるときも (図では時刻 $t = 0.4$ [s], $t = 0.8$ [s] のとき)、入力画像と対象画像の比較から分かる通り、追跡対象のみが抽出され、Self Windowing が正しく動作することが分かる。またアクチュエータの軌跡についても、対象の水平方向の動きに対応したものであり、障害となる対象に妨害されることなく追跡対象を追跡していることが分かる。

4.6 衝突・分離がある場合の追跡

次に、Fig. 4 に示したすべてのアルゴリズムを実装し、追跡対象に対して衝突や分離が生じた場合について対象追跡を実現した。制御サイクルタイムは 4 [ms] で実現した。なお、衝突検出時の総和の変化のしきい値は $c = 7$ [pixels] と設定した。

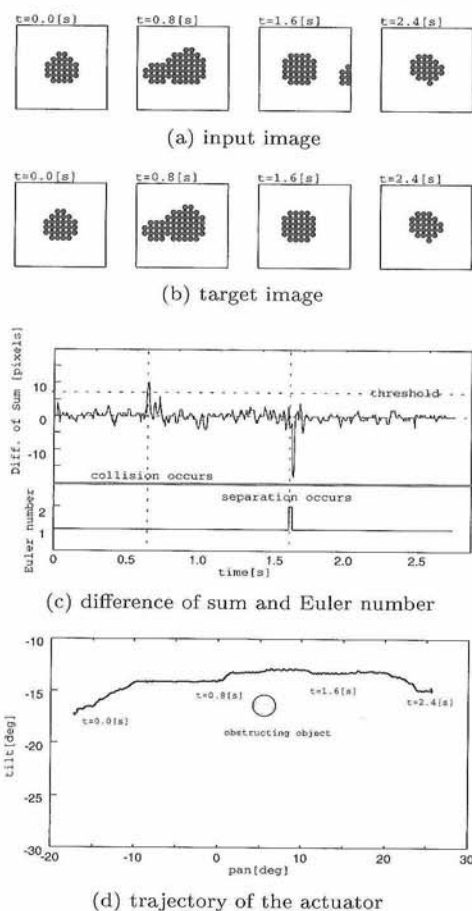


Fig. 7 Tracking results with collision and separation

実験結果として Fig. 7 に、入力画像、対象画像、衝突・分離検出に必要な対象画像の総和の変化、オイラー数、アクチュエータの軌跡を示す。

Fig. 7 (c) について、追跡対象の衝突時 ($t = 0.6$ [s]) で処理画像の総和が急激に変化しており、しきい値を越えた変化が起き、障害となる対象との衝突が検出されている。追跡対象の分離時 ($t = 1.5$ [s]) では、オイラー数が 1 から 2 へ変動し、追跡対象と障害となる対象の分離が検出されている。

また、アクチュエータの軌跡をみても分かる通り、分離前だけでなく、分離後についても、障害となる対象と誤認するこ

となく、衝突前に追跡していた対象のみが再び追跡されており、分離時の追跡対象の再認識が実現されていることが分かる。

5. おわりに

本論文では、超並列・超高速ビジョンの特徴である高速な画像入力に着目し、2 値画像の超高速ビジョン系の特徴を定義した上で、超高速ビジョン画像処理アルゴリズムの例として動画像対応付けを実現する Self Windowing を提案した。また、超並列・超高速ビジョンシステムである 1 ms ビジュアルフィードバックシステムにそのアルゴリズムを実装し、対象追跡を実現することにより、対象追跡を例に実際の動作を確認した。

参考文献

- [1] 橋本: “視覚フィードバック制御—静から動へ”, システム/制御/情報, vol.38, no.12, pp.659–665, 1994.
- [2] 井上: “リアルタイムビジョンとその応用”, 日本ロボット学会誌, vol.13, no.3, pp.306–310, 1995.
- [3] M. Ishikawa, A. Morita and N. Takayanagi: “High speed vision system using massively parallel processing,” Proc. IEEE Int. Conf. on Intelligent Robotics and Systems, pp.373–377, 1992.
- [4] 中坊, 石井, 石川: “超並列・超高速ビジョンを用いた 1 ms ターゲットトラッキングシステム”, 日本ロボット学会誌, vol.15, no.4, pp.417–421, 1997.
- [5] R. Cypher and J. Sanz: “SIMD architectures and algorithms for image processing and computer vision,” IEEE Trans. on Acoustics, Speech and Signal Processing, vol.37, no.12, pp.2158–2174, 1989.
- [6] 石井, 石川: “超並列ビジョンのためのマッチングアルゴリズム”, 電子情報通信学会技術報告, pp.121–126, 1995.
- [7] I. Ishii, Y. Nakabo and M. Ishikawa: “Target tracking algorithm for 1 ms visual feedback system using massively parallel processing,” Proc. IEEE Int. Conf. on Robotics and Automation, pp.2309–2314, 1996.
- [8] 中坊, 石井, 石川: “ビジュアルインピーダンスを用いたロボットの制御”, 日本機械学会 ROBOMECH 講演会論文集, pp.999–1002, 1996.
- [9] J. Serra: Image Analysis and Mathematical Morphology. Academic Press, London, 1982.
- [10] T. Komuro, I. Ishii and M. Ishikawa: “Architecture using general purpose processing elements for 1 ms vision system,” Proc. CAMP'97, pp.276–279, 1997.
- [11] A. Rosenfeld and A.C. Kak: Digital Picture Processing. Academic Press, London, 1982.
- [12] 村田, 松内, 石井, 石川: “ビジョンチップシステムのための制御アーキテクチャ”, 日本機械学会 ROBOMECH 講演会論文集, pp.1089–1092, 1997.
- [13] 坂口, 小室, 石井, 石川: “ビジョンチップのためのモーメント出力回路”, 第 35 回計測自動制御学会学術講演会予稿集, pp.829–830, 1996.

石井 抱 (Idaku Ishii)



1970 年 2 月 14 日生。1994 年、東京大学大学院修士課程修了。1996 年同大学院工学系研究科博士課程中退。同年同大学院工学系研究科計数工学専攻助手、現在に至る、ビジョンチップ、センサ並列情報処理、ロボット制御システムの研究に従事。

1997 年高度自動化技術振興賞 (本賞)、第 3 回ロボティクスシンポジウム最優秀論文賞、1998 年日本ロボット学会論文賞受賞。(日本ロボット学会正会員)



石川正俊 (Masatoshi Ishikawa)

1954 年 8 月 22 日生。1979 年、東京大学大学院修士課程修了。同年工技院製品科学研究所 (現応用生命工学研究所)、1989 年、東京大学工学部計数工学科助教授。現在、東京大学大学院工学系研究科計数工学専攻助教授。触覚センサ、超並列超高速ビジョン、光コンピューティング、センサフュージョンの研究に従事。工学博士。1984 年計測自動制御学会論文賞、1988 年度工業技術院長賞、1989 年応用物理学会光学論文賞、1997 年高度自動化技術振興賞 (本賞)、1998 年日本ロボット学会論文賞受賞。(日本ロボット学会正会員)