

## ABC アルゴリズムを用いた物体追跡手法の検討

## A study of object tracking method using artificial bee colony algorithm

○藤田 敦匡, 萩原 義裕, 萩原 由香里, アデルジャン イミティ

○Nobumasa FUJITA, Yoshihiro HAGIHARA, Yukari HAGIHARA, Adiljan Yimit

岩手大学

Iwate University

キーワード: ABC アルゴリズム (artificial bee colony algorithm), テンプレートマッチング (template matching), 物体追跡 (moving objects tracking)

## 1. 緒言

物体追跡とは、指定した対象が動画像上をどのように移動しているか推定する技術のことであり、デジタルカメラのオートフォーカス機能やパノラマ写真の作成、防犯カメラの人物追跡など幅広い分野で利用されている。物体追跡には大きく分けて2つの種類があり、指定した領域内のカラー情報をもとにマッチングを行う領域ベースの追跡と、画像上の特徴点を用いて画像間のマッチングを行う特徴点ベースの追跡に大別できる<sup>1)</sup>。本論では、領域ベースの追跡方法であるテンプレートマッチングを利用する。しかし、画素単位でテンプレート画像と入力画像の照合を行うのに膨大な計算が必要であるため、専用のハードウェアによる処理、前処理による画像の低解像度化、GA アルゴリズムのような最適化手法などにより高速化が図られている。

そこで、本論では GA アルゴリズムや PSO アルゴリズム等に比べ、様々な優位性が示されている群知能アルゴリズムの一種である ABC アルゴリズム<sup>2), 3)</sup>を用いた物体追跡手法の提案と、その性能を評価した結果について述べる。

## 2. ABC アルゴリズム

## 2-1 ABC アルゴリズムの概要

ABC アルゴリズムは D. Karaboga らにより提案された群知能アルゴリズムであり、働き蜂、傍観蜂、斥候蜂の3種類の人工蜂群と

餌場を基本要素として最適化問題の大局解を求める探索手法である<sup>4)</sup>。以下に ABC アルゴリズムの手順を示す。

Step1: 餌場の数  $N_f$  と同じ数の働き蜂を探索空間 ( $D$  次元の超立方体) においてランダムに配置しながら式(1)により適合度  $f$  の計算を行う。配置された餌場の位置を  $u_i$  ( $i=1, \dots, N_f$ )、この位置における適合度の非更新回数を  $t_i$  とする。本ステップでは  $t_i=0$  と設定する。

$$f_i = \begin{cases} \frac{1}{1+d(u_i)}, & \text{if } d(u_i) \geq 0, \\ 1+|d(u_i)|, & \text{otherwise.} \end{cases} \quad (1)$$

ここで  $d(u_i)$  は適応関数である。

Step2: ステップ 1 における各餌場の適合度を比較することで、もっとも優れた餌場を最良餌場としてその適合度  $f_{best}$  とともに記録する。

Step3: 働き蜂の現在地の周囲から式(2)により新しい餌場  $v_i$  を探索し、その新しい餌場の適合度を計算する。

$$v_i = u_i + \phi(u_i - u_k) \quad (2)$$

ここで  $u_k$  は  $u_i$  以外の場所であり、 $\phi$  は  $[-1, 1]$  内のランダム実数である。

Step4: 新しい餌場  $v_i$  の適合度が  $u_i$  の適合度より大きい場合、この  $v_i$  を記録し、適合度を書き換える。そうではない場合、新しい餌場  $v_i$  を棄却し、非更新回数  $t_i$  の値を 1 増す処理を行う。

Step5: すべての餌場における適合度から相対確率  $p_i$  を式(3)より求める。

$$p_i = f_i / \sum_{i=1}^{N_f} f_i \quad (3)$$

Step6: 傍観蜂を相対確率に基づくルーレッ

トにより  $N_f$  回選択し、毎回選択された傍観蜂に対してステップ 3 と 4 を適用する。

- Step7: 各餌場の適合度  $f_i$  を最良餌場の適合度  $f_{best}$  と比較しながら、より優れた餌場を最良餌場として保持し、その適合度を  $f_{best}$  として記録する。
- Step8: 非更新回数  $t_i$  が設定した限界  $limit$  を超えた餌場を棄却し、そこにいる働き蜂をステップ 1 により再配置する。
- Step9: ステップ 3 からステップ 8 を設定した最大ループ回数  $R_{max}$  まで反復させる。

## 2-2 修正 ABC アルゴリズム

上述した ABC アルゴリズムは時不変関数の大局解探索を目的としているため、目的関数の時間変化には対応していない。そこで、西田は D. Karaboga らの ABC アルゴリズムに対して修正を施すことで、時間変化に適応できる ABC アルゴリズムの修正版を提案した<sup>3)</sup>。その修正手順について以下に述べる。

- (1) ステップ 3 と 4 間に次のステップを追加する。

Step3-1: 餌場  $u_i$  の適合度を再計算する。

このステップの追加により目的関数の変化に伴う適合度の変化をステップ 4 における働き蜂の更新に反映させることができる。

- (2) ステップ 7 を次のように修正する。

Step7: 各餌場の中からより優れた餌場を最良餌場として保持し、その適合度を  $f_{best}$  として記録する。

この修正により、全時刻ではなく、各時刻の最大適合度を求めることができ、目的関数の変化によって生じる適合度の変化に適応可能となる。

## 3. 修正 ABC アルゴリズムを用いたテンプレートマッチング

### 3-1 修正 ABC アルゴリズムの利用

本論では上記の修正 ABC アルゴリズムを用い、動画に対してテンプレートマッチングを行う。また、ステップ 7 の修正は全ループに対してではなく 1 フレームごとに適用する。テンプレートマッチングはテンプレート画像と探索位置における重み付きヒストグラムを作成し、その差の二乗をとることにより行い、その座標は働き蜂の座標  $v_i$  にあたる。また、マッチング結果は緑枠で囲んで表示する。

## 3-2 重み付きヒストグラムの作成

今回物体追跡に用いるテンプレートマッチングはカラー画像で行う。そのため RGB 各チャンネルの輝度値をそれぞれ  $C\_BIN$  に量子化し、カラー空間を  $C\_BIN^3$  ビンに分割する。

対象画像  $\{x_i\}(i=1, \dots, N)$  は  $y$  を中心座標に画素位置が正規化されているとした時、重み付きヒストグラム  $h = \{h_z\}(z=1, \dots, C\_BIN^3 - 1)$  は式(4)より求められる<sup>(5)</sup>。

$$h_z = C \sum_{i=1}^N k \left( \left\| \frac{y - x_i}{r^*} \right\|^2 \right) \delta[u, b_c(x_i)] \quad (4)$$

ここで、画素  $x_i$  の色に対応するビン番号は  $b_c(x_i)$  で表される。 $k(\cdot)$  はバンド幅を  $r^*$  とする重み関数であり、 $k(x)$  は式(5)より計算される。また、 $\delta[\cdot]$  は Kronecker のデルタ関数、 $C$  は正規化定数であり、それぞれ次のように表される。

$$k(x) = \begin{cases} 1-x, & \text{if } 0 \leq x \leq 1, \\ 0, & \text{otherwise,} \end{cases} \quad (5)$$

$$\delta[a, b] = \begin{cases} 1, & \text{if } a = b, \\ 0, & \text{otherwise,} \end{cases} \quad (6)$$

$$C = 1 / \sum_{i=1}^n k \left( \left\| \frac{y - x_i}{r^*} \right\|^2 \right) \quad (7)$$

## 4. 実験

### 4-1 実験方法

追跡対象が移動している条件の異なる動画 A~E を作成し、各動画に対して 5 回ずつマッチングを行いマッチング率と処理時間を求める。マッチング結果の成否はオーバーラップ 50%以上つまり、マッチング結果が 24 ピクセル以内の誤差であれば成功と判定する。マッチング率はマッチングが成功したフレーム数を総フレーム数で除することで求める。

使用するテンプレートは  $49 \times 49$  ピクセルのものを、動画は  $320 \times 240$  ピクセルで 30 フレーム/秒のものを 300 フレーム分(10 秒)用いた。各フレームでマッチングを行う際の最大ループ回数を  $R_{max}$ 、用いる蜂の総数を  $NP$  として実験を行う。

図 1, 2 にテンプレート画像とマッチング



結果とその軌跡の一例を示す。



Fig. 1 Template image



Fig. 2 Snapshot of tracking result

#### 4-2 実験用動画

実験には図 1 で示した画像が類度の高い背景を移動している動画を使用する。動画 A は追跡対象と背景のみのもの、動画 B はオクルージョンを生じさせるために黒い太線を引いたものを、動画 C は追跡対象とは別に以下の図 3 に示す画像を追加したものを、動画 D には B,C に用いた 2 つの画像を追加したものとなっている。図 4 は動画 D においてオクルージョンが発生した時の画像である。



Fig. 3 Similar object with high similarity to the template

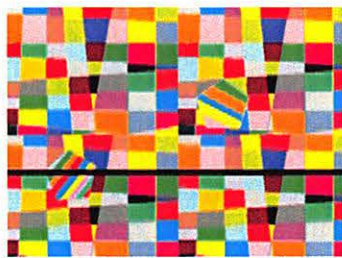


Fig. 4 Snapshot from video-D sequence

#### 4-3 事前実験

事前実験として ABC アルゴリズムのパラメータの組み合わせによるトラッキング率への影響を調査する。今回は  $R_{max} \times NP$  が同数である場合のトラッキング率の関係を調べるために、異なる蜂の数において比較する。動画 A, D を用い各  $R_{max}$  の値において  $NP$  の値を変え 10 回ずつ追跡を行う。表 1, 3 はトラッキング率の平均を、表 2, 4 は標準偏差を示す。

Table 1 Tracking results for video- A using different colony size  $NP$  and cycle number  $R_{max}$ . [%]

$R_{max} \backslash NP \times R_{max}$	120	160	200	300	400
1	90.4	95.3	95.4	97.6	98.4
2	79.8	90.5	95.7	98.3	99.1
4	61.8	72.6	79.2	94.0	98.1
6	54.1	62.7	69.0	85.4	93.1
8	52.8	58.4	63.3	79.2	87.7
10	50.0	57.2	60.6	76.8	86.1

Table 2 Standard deviation of tracking results for video- A using different colony size  $NP$  and cycle number  $R_{max}$ .

$R_{max} \backslash NP \times R_{max}$	120	160	200	300	400
1	6.84	5.26	3.04	1.94	1.59
2	5.86	4.60	2.04	1.41	1.26
4	4.01	3.84	3.14	2.11	1.36
6	3.80	3.07	2.98	2.15	1.56
8	3.59	3.42	2.84	2.13	1.91
10	2.81	2.77	2.48	2.02	1.18

Table 3 Tracking results for video- B using different colony size  $NP$  and cycle number  $R_{max}$ . [%]

$R_{max} \backslash NP \times R_{max}$	120	160	200	300	400
1	84.4	87.3	88.3	92.7	92.9
2	74.7	82.7	91.2	95.3	95.4
4	56.8	66.4	75.5	88.0	93.6
6	49.6	56.0	64.4	77.8	85.5
8	43.8	54.9	56.1	73.6	80.6
10	43.3	51.6	58.2	71.8	77.1

Table 4 Standard deviation of tracking results for video- B using different colony size  $NP$  and cycle number  $R_{max}$ .

$R_{max} \backslash NP \times R_{max}$	120	160	200	300	400
1	5.51	5.01	4.83	2.05	1.87
2	6.52	3.36	1.99	1.22	0.75
4	5.56	4.93	3.19	2.25	1.71
6	3.26	3.14	3.85	2.21	2.33
8	3.82	2.98	3.22	2.89	1.26
10	2.92	3.42	3.46	3.50	3.13

これらの結果より、 $R_{max} \times NP$  の値が増加するにつれトラッキング率も上昇し、標準偏差は小さくなる傾向にある。また、 $R_{max}$  の値が小さいほどにトラッキング率が高く標準偏差も大きくなる傾向にある。この結果により実験に用いる  $R_{max}$  の値は、多い計算回数においてトラッキングが高く標準偏差が小さい 2 を使用するものとする。



#### 4-4 実験結果

$R_{max} = 2$  の時の各蜂数におけるトラッキング率と処理時間を表 5,6 に示す。

Table 5 Tracking results for each video when using different colony size  $NP$ . [%]

Video \ $NP$	40	60	80	100	200
A	61.5	80.5	93.1	96.7	99.1
B	60.5	80.1	91.4	96.9	99.5
C	55.3	79.4	87.7	93.0	98.3
D	53.7	73.0	81.4	89.9	95.8

Table 6 Average processing time for per video when using different colony size  $NP$ . [Sec]

$NP$	40	60	80	100	200
average	21.3	30.5	35.7	40.9	73.3

#### 4-5 考察

事前実験より,  $R_{max} \times NP$  の値が同じである際,  $R_{max}$  の値が少ない時トラッキング率が高くなるのは, 一つのフレームに配置する働き蜂の数が  $R_{max}$  が多い時と比べ増えるためであると考えられる。つまり, 探索空間内今回は  $320 \times 240$  ピクセルの動画内において, 一度に配置される働き蜂の数が多くなることにより探索範囲が広がり, 正解座標を探索しやすくなるためであると考えられる。また, 表 2 において標準偏差は概ね  $R_{max}$  の値が少ない時に大きくなる傾向にある。これは, 働き蜂の配置に乱数を用いているため, 正解座標付近に配置されなかった場合, 次に再配置され正解座標に近づく機会が少なくなるためであると考えられる。

表 5 において蜂の数の増加に伴いトラッキング率も上昇しているのが分かる。これは先に述べた通りの理由であると考えられる。また, 各動画でトラッキング率の差がある。動画 A, B はほぼ変わらない値であるがこれは, B に追加した図 3 がトラッキングに与える影響が少なかったためである。動画 C はオクルージョンが発生した際に背景の適合度が高い箇所にトラッキングするため動画 A よりトラッキング率が低いものとなった。動画 D は動画 B の時に影響が少なかった図 3 を用いているが動画 C よりトラッキング率が低い。これは黒棒によるオクルージョンが発生している際に, 図 3 と背景に用いた色により動画中で最も高い適合度となっていた

ためである。しかし, 用いる蜂の数を増加させることによりほぼ追跡は可能である。

本実験に用いた動画は約 10 秒のであるが, 表 6 より結果はどの数値も 10 秒以上であるため, 現状リアルタイムでの処理ができない。そこで, 今後高いトラッキング率を実現している蜂の数の場合においてもリアルタイムでの処理が可能になるよう処理速度を向上させる必要がある。

#### 5. 結言

本論では, 修正 ABC アルゴリズムを用いた動画に対する物体追跡の実装を行った。また用いるパラメータの組み合わせによるトラッキングへの影響を調査した。今後の方針として, 他のアルゴリズムとの比較や, 他のテンプレートマッチングの評価関数との比較, 処理速度の向上が挙げられる。

#### 参考文献

- 1) 藤吉弘巨, "4-2-1 テンプレートマッチング (ブロックマッチング)". 電子情報通信学会. (オンライン), 入手先 <[http://www.ieice-hbkb.org/files/02/02gun\\_02hen\\_04.pdf#page=8](http://www.ieice-hbkb.org/files/02/02gun_02hen_04.pdf#page=8)>, (参照 2015-05-20).
- 2) D.Karaboga and B.Akay "A comparative study of Artificial Bee Colony algorithm" "Applied Mathematics and Computation, 214, pp.108-132, 2009
- 3) 西田 健, "時変関数に適応するための ABC アルゴリズムの修正," 電気学会論文誌 C, Vol.132, No4, pp.584-591, 2011
- 4) D.Karaboga and B.Basturk, "A powerful and efficient algorithm for numerical function optimization:Artificial bee colony (ABC) algorithm," J.Global Optimization, Vol.39, pp.459-471, 2007
- 5) D. Comaniciu, V. Ramesh and P. Meer: Kernel-Based Object Tracking, IEEE Trans. of Pattern Analysis and Machine Intelligence, vol.25, no.5, 2003.