時空間最適化フレームワークと粒子群最適化法による インタラクティブ物体追跡システム

坂本　眞人[a)]・濵田　卓[b)]

Interactive Object Tracking System with Space-time Optimization Framework and Particle Swarm Optimization

Makoto SAKAMOTO, Takashi HAMADA

Abstract

We present an object tracking system based on space-time optimization and particle swarm optimization. In a space-time optimization framework, we formulate the object tracking problem as continuous trajectory optimization problem. This enables us to obtain an objective function to track an object. In this paper, we optimize the objective function with Particle Swarm Optimization (PSO) algorithm which is a type of evolutionary computation. We discuss the efficiency and the issues of applying the PSO Algorithm to object tracking problem.

Keywords: Object trackning, Particle swarm optimization, Space-time optimization

1. はじめに

近年、3DCG アニメーションの制作方法として、単眼のビデオカメラで撮られた映像から人間の動きのデータを測定するビデオモーションキャプチャー(Video Motion Capture ： VideoMocap)に注目が集まっている。この手法は、現存する大量のビデオソースをモーションキャプチャのために利用可能で、新たなデータを収集する際にビデオカメラ以外の特殊な機材を一切必要としないという点で優れている。

我々のグループでは、これまでフラ衣装用アパレル CAD に関する研究開発を行なってきており、今回はデザインした衣服のシミュレーションを行うためのアニメーションの生成方法として、VideoMocap に着目した。

その VideoMocap の手法の一つに、X. Wei らによる、時空間最適化(Spacetime Optimization)フレームワークでの物体追跡(Object Tracking) [2]をベースとした 3D アニメーションの生成が挙げられる[3]。この提案手法は非常に効果的なものだったが、物体追跡のための最適化に MATLAB や MATCH MOVER といった高価なソフトウェアを前提としており、多くの人が所有するコンピュータ上で動作するシステムであるとは言い難かった。

この問題を解決するため、本研究では X. Wei らが提唱する物体追跡のための時空間最適化フレームワーク[2)]に、群知能の一種である粒子群最適化法[1)]を組み合わせることで、多くのコンピュータ上で利用可能な物体追跡システムの試作とその有用性の検証を試みた。また、直感的で簡易な方法で操作でき、視覚的にもわかりやすいシステムになるよう、ユーザインターフェースにも十分に配慮した[4-8)]。

a) 情報システム工学科准教授

b) 情報システム工学科学部生

2. 対象領域の表現

この節では時空間最適化による物体追跡のための対象領域のモデリング方法について述べる。

2.1. 追跡対象となる物体の近似

本研究において、追跡の対象となるあらゆる物体は楕円領域で近似される。そしてフレーム$\boldsymbol{t}$における対象領域の状態は次のように 5 つのパラメータで表される。

$$Z_t = \left(x_t, y_t, a_t, b_t, \theta_t\right)^T \tag{1}$$

ここで$\boldsymbol{x_t}$、$\boldsymbol{y_t}$は近似する楕円の中心座標を表し、$\boldsymbol{a_t}$、$\boldsymbol{b_t}$はそれぞれ楕円の長半径、短半径の長さを表す。$\boldsymbol{\theta_t}$は楕円の長軸と$\boldsymbol{x}$軸がなす角度である。

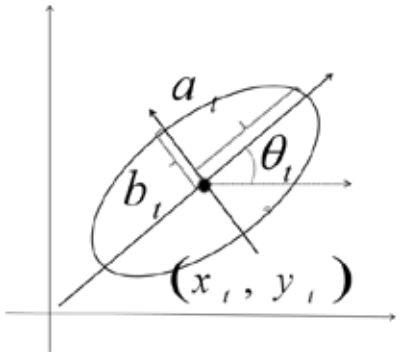

図 1. パラメータ化された対象領域.

2.2. 特徴空間

本研究では、色相(Hue)、彩度(Saturation)、Value(明度)の3つの成分からなるHSV色空間を用いて、先で述べた対象領域全体を特徴づける。そうして得られた対象領域内のHSVヒストグラムを特徴空間(feature space)と言う。特徴空間はフレーム$\boldsymbol{t}$において次のように表される。

$$\boldsymbol{h}(\boldsymbol{Z}_t) = \left(h_1(\boldsymbol{Z}_t), \dots, h_M(\boldsymbol{Z}_t)\right)^T, \quad \sum_{\mathrm{m}=1}^{\mathrm{M}} \mathrm{h_m}(\mathbf{Z_t}) = 1 \tag{2}$$

ただし、Mは HSV 色空間におけるビンの総数、$\mathrm{h_m}(\mathbf{Z_t})$はm番目に対応するビンの値である。ここで、フレーム$\boldsymbol{t}$で対象領域内に存在するピクセルの総数を$\mathrm{N_t}$、その中のi番目のピクセルの座標を$\boldsymbol{P_t^i}$とすると，$\mathrm{h_m}(\mathbf{Z_t})$ は次のように表現できる。

$$h_m(\boldsymbol{Z}_t) = \sum_{i=1}^{N_t} \delta\left(f(I(P_t^i)) - m\right) \tag{3}$$

ただし、$\delta(\mathrm{x})$はクロネッカーのデルタ関数で、

$$\delta(\mathrm{x}) = \begin{cases} 1 & (\mathrm{x} = 0) \\ 0 & (\mathrm{x} \neq 0) \end{cases} \tag{4}$$

となり、$I(P_t^i)$は$\boldsymbol{P_t^i}$がもつ色情報を表している。fは$I(P_t^i)$に対応するヒストグラムのビンのインデックスを与える。次に、対象領域の中心から離れたピクセルが特徴空間に与える影響を小さくするため、カーネル関数$k(r)$を用いてヒストグラムに重み付けを行う。今回の実装では次のようなカーネル関数を用いた。

$$\mathrm{k(r)} = \begin{cases} 1 - \mathrm{r} & 0 \leq \mathrm{r} \leq 1 \\ 0 & \mathrm{r} > 1 \end{cases} \tag{5}$$

このカーネル関数により$\mathrm{h_m}(\mathbf{Z_t})$はフレーム$\boldsymbol{t}$において次のように計算できる。

$$h_m(\boldsymbol{Z}_t) = \frac{\sum_{i=1}^{N_t} k\left(\left(\frac{u^i}{a}\right)^2 + \left(\frac{v^i}{b}\right)^2\right)\delta(f(I(\boldsymbol{P_t^i})) - m)}{\sum_{i=1}^{N_t} k\left(\left(\frac{u^i}{a}\right)^2 + \left(\frac{v^i}{b}\right)^2\right)} \tag{6}$$

ここで、u^iとv^iは$\boldsymbol{P_t^i}$の長軸、短軸への正射影であり、これらは$\boldsymbol{P_t^i}$と楕円の中心座標$\boldsymbol{C_t}$を用いて次のように計算される。

$$\begin{pmatrix} u^i \\ v^i \end{pmatrix} = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} (\boldsymbol{P_t^i} - \boldsymbol{C_t}) \tag{7}$$

2.3. テンプレートモデル

キーフレームとして，1フレーム目と$\boldsymbol{T}$フレーム目の対象領域の状態$\boldsymbol{Z_1}$，$\boldsymbol{Z_T}$とその特徴空間$\boldsymbol{h}(\boldsymbol{Z_1})$，$\boldsymbol{h}(\boldsymbol{Z_T})$が与えられた時、その間のフレームの未知の特徴空間を$\boldsymbol{Z_1}$と$\boldsymbol{Z_T}$の線形補間で表現したものを、テンプレートモデルと呼び、0から1の範囲の実数値を取るパラメータβ_tを用いて次のように表す。

$$H_{\boldsymbol{m}}(\beta_t) = \beta_t\, \mathrm{h_m}(\boldsymbol{Z_1}) + (1 - \beta_t)\, \mathrm{h_m}(\boldsymbol{Z_T}) \quad 0 \leq \beta_t \leq 1\ ,\ 1 < t < T \tag{8}$$

したがって、上記の線形補間によってフレームt $(1 < t < T$)における対象領域のHSVヒストグラムは次のようにβ_tの関数として表現することが出来る。

$$\mathbf{H}(\boldsymbol{Z_t}) = \left(H_1(\beta_t), \dots, H_M(\beta_t)\right)^T \tag{9}$$

3. 時空間最適化

3.1. 最適化の概要

物体追跡の問題を追跡対象領域の軌道最適化問題と考え、すべてのフレームの対象領域の状態を同時に計算する手法を時空間最適化と呼ぶ。この追跡方法は、跡を行う範囲の最初と最後のフレームでユーザによる対象領域の注釈(Annotation)が必要となるが、事前学習などの前処理が一切不要であるという利点がある。また、物体の急激な動きや、対象物体の隠れ、明度の変化にも強いことが知られている[2)]。

3.2. 目的関数

フレーム$t(1 < t < T)$において、追跡の候補領域の特徴空間$\boldsymbol{h}(\boldsymbol{Z}_t)$と前節で述べた、$\boldsymbol{Z}_1$と$\boldsymbol{Z}_T$の線形補間で表現されるテンプレートモデル$\mathbf{H}(\boldsymbol{Z}_t)$との類似度関数$\boldsymbol{d}$を次のように定義する。

$$\mathrm{d}(\mathbf{Z}_\mathrm{t},\ \beta_t) = \sqrt{1 - \sum_{\mathrm{m}=1}^{\mathrm{M}} \sqrt{\mathrm{h_m}(\boldsymbol{Z}_1) H_m(\beta_t)}} \tag{10}$$

すると、2 から T-1 フレームまでの追跡候補領域の特徴空間$\boldsymbol{h}(\boldsymbol{Z}_t)$とテンプレートモデル$\mathbf{H}(\boldsymbol{Z}_t)$の全体の類似度は次のように評価される。

$$\mathrm{E_I} = \sum_{\mathrm{t}=2}^{\mathrm{T}-1} \rho\left(\mathrm{d}(\mathbf{Z}_\mathrm{t}, \beta_t)\right) \tag{11}$$

ここで、ρはローレンツ推定器であり、

$$\rho(\mathrm{r}) = \log(1 + 1/2(\mathrm{r}/\sigma)^2) \tag{12}$$

である。推定器は$\mathrm{d}(\mathbf{Z}_\mathrm{t},\ \beta_t)$の統計上の異常値を 0 に近づけるために用いられる。

次に、2 から T-1 フレームまでの対象領域の状態$\mathbf{Z}$の変化量と、補間パラメータ$\boldsymbol{\beta}$の変化量の総和をそれぞれ$\mathrm{E_P}$、$\mathrm{E_L}$とし次のように表す。

$$E_P = \sum_{t=2}^{T-1} \|\boldsymbol{Z}_t - \boldsymbol{Z}_{t-1}\| \tag{13}$$

$$E_L = \sum_{t=2}^{T-1} \left(\beta_t - \beta_{t-1}\right)^2 \tag{14}$$

すると、時空間最適化に基づく物体追跡は次のような制約付き非線形最適化問題に帰着される。

$$arg \min_{\boldsymbol{Z},\beta} E_I(\boldsymbol{Z},\beta) + \lambda_P E_P(\boldsymbol{Z}) + \lambda_L E_L(\beta)$$

$$0 \le \beta_t \le 1 \ ,\ \mathrm{t} = 1, \ldots, \mathrm{T} \tag{15}$$

本研究では、次の節で述べる粒子群最適化法を用いて上記の最適化問題を解く。

4. 粒子群最適化法

粒子群最適化法[1)]（Particle Swarm ptimization）とは、生物の群れをモデルとしたアルゴリズムである。生物の群れでは、群れの中の一匹がよさそうな経路を発見すると、群れの残りはどこにいてもそれに倣うことができる。粒子群最適化法では一つの粒子（個体）を設定された問題の解とみなし、この粒子の群れを多次元空間において、位置と速度を持つ粒子群でモデル化する。位置の評価は適応度関数で行い、群れの各粒子同士は常に、より良い位置についての情報を交換する。この情報を基にして、各粒子が自身の位置と速度を調整することで、群れ全体としての最適解を導くことができる。各粒子の位置$\boldsymbol{x}$と速度$\boldsymbol{v}$は粒子毎の最良値$\boldsymbol{x}_{pbest}$と群れ全体の最良値である$\boldsymbol{x}_{gbest}$に従って以下のように更新される。

$$\boldsymbol{x} \leftarrow \boldsymbol{x} + \boldsymbol{v} \tag{16}$$

$$\boldsymbol{v} \leftarrow \omega\boldsymbol{v} + c_1 r_1\left(\boldsymbol{x}_{pbest} - \boldsymbol{x}\right) + c_2 r_2\left(\boldsymbol{x}_{gbest} - \boldsymbol{x}\right) \tag{17}$$

ここでr_1、r_2は 0 から 1 の範囲の実数値を取る乱数であり、ωは慣性定数である。c_1、c_2は正加速度計数で、通常は$\mathrm{c_1} = \mathrm{c_2} = 1$を用いる。

5. システムの実装

本研究では、C++とクロスプラットフォーム GUI ライブラリの Qt，そして画像処理用ライブラリである OpenCV を用いてシステムの試作を行った。 このシステムの機能は大きく分けると以下 4 つである。

1. ビデオの再生
2. 追跡物体の注釈
3. 注釈されたフレーム間の物体追跡
4. 追跡結果の保存，読み込み

6. 実験結果

この節では、試作したシステムを利用して実際に物体追跡を行った結果を示す。なお、実験にはアメリカンフットボールの試合の映像を使用し、映像中の選手 1 名を追跡の対象とした。

6.1. キーフレーム 2 枚での結果

図 2 は動画の最初と最後のフレームのみをキーフレームとして追跡を行った結果である。この実行結果では図 2 の#52 のように、対象物体を捉えられていないフレームが目立った。

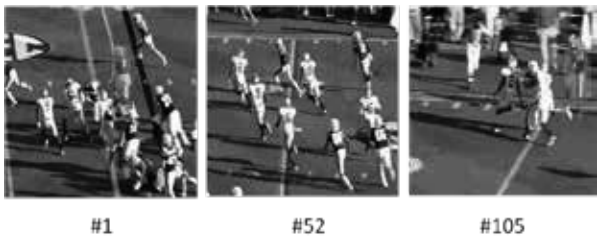

図 2. キーフレーム 2 枚時の追跡結果.

6.2. キーフレーム 4 枚

図 3 は、更に 2 枚のキーフレームを追加して実験を行った結果である。この追加により、ほぼ全てのフレームで対象となる選手が追跡できた。

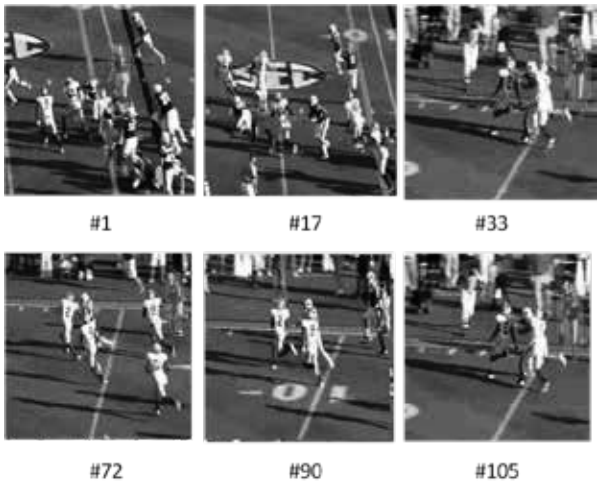

図 3. キーフレームが 4 枚時の追跡結果.

7. 考察

先の実験で得られた実験結果より、キーフレーム数が十分である場合に、本システムが正しく物体を追跡出来ていることを確認できた。一方で、追跡範囲の総フレーム数に対して注釈されたキーフレーム数が少なすぎる場合、正確な追跡が行えないことがわかった。本システムの核となる粒子群最適化法は比較的新しいアルゴリズムであり、一般に遺伝的アルゴリズムなどの他の進化型計算手法よりも早く解に収束することで知られている。しかしその反面、局所解（極小値，極大値）に陥りやすいこともわかっている。そのため、現在ではそれを回避するための様々な手法が提案されているが、本システムでは、粒子群最適化法の最も基本的な手法で目的関数の最適化を行なっている。したがって、対象物体の追跡が*失敗*している場合、最適化時に目的関数の局所解に陥っている可能性が考えられる。

また、時空間最適化フレームワークによる物体追跡は、追跡を行うフレーム数に比例して目的関数の解空間が大きくなるため、追跡時間が長くなるほどを最適化に要する時間も長くなる。さらに、今回は確率的な最適化アプローチである粒子群最適化法を用いて目的関数の最小化を試みているため、フレーム数が大きい場合の最適化速度の遅さが際立っている。本研究で行った実験では 100 フレーム程度の追跡区間の最適化に 400[s]前後を要しており、十分な速度とは言い難い。

今回の実験で、粒子群最適化を物体追跡に用いた際の一定の有用性は認められたが、前述のとおり速度の面から考えると、この手法による最適化はまだまだ検討が必要であると言える。

ここで、粒子群最適化の高速化の手法として、GPGPU(GPUによる汎目的計算)による粒子の並列化が考えられる。この方法を用いることで、これまではループの中で 1 個体ずつ位置と速度の更新を行なっていたものが、複数の個体の位置と速度の更新が同時並行で行われるようになるため、最適化速度の飛躍的な向上が期待できる。

また、粒子群最適化以外の最適化手法として、反復法などの数値計算に基づいた最適化手法の採用なども有効な候補として考えられる。

8. まとめ

本研究では，粒子群最適化法を用いて一般的なコンピュータ上で動作する物体追跡システムを試作した．試作したシステムはユーザインタフェースの面では当初の目標に達することができたが，物体追跡の性能，計算速度の面では十分ではなかった．これは，本研究で用いた粒子群最適化法の性質に起因すると考えられるため、今後の十分な検討を必要とする。

しかし、広く普及しているコンピュータ上で動作するシステムという意味においては、Windows、 Mac、Linux などのプラットフォームに依存しない環境で動作するシステムが構築できたという点や、MATLAB など、外部のソフトウェアを必要としない点で一定の成果が挙げられたのではないかと考える。

本研究で構築したシステムの今後の課題と展望としては、最適化手法の改善や、今回試作したシステムをベースとした VideoMocap システムの構築等が挙げられる [4-8]。

参考文献

1) C.Dorin and M.Shift: A Robust Approach Toward Feature Space Analysis. s.l.,IEEE . TRANANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, Vol. 24, No.5 , pp. 603 - 618, MAY 2002.
2) J.D.Foley 他:佐藤　義雄　監訳，コンピュータグラフィックス理論と実践，オーム社，2002.

3) K.James and E.Russell: Particle Swarm Optimization, IEEE Service Center, Piscataway, NJ, IV, pp. 1942-1948, 1995.
4) S.Masaki, H.Matsushita and Y.Nishio: Particle Swarm Optimization Containing Plural Swarms Using Shared Velocity, RISP Journal of Signal Processing, Vol. 15, No. 4, pp.255-258, 2011.
5) X.Wei and J.Chai: Interactive Tracking of 2D generic Objects with Spacetime Optimization,European Conference on Computer Vision (ECCV 2008), 2008.
6) X.Wei and J.Chai:VideoMocap: Modeling Phisically Realistic Human Motion from Monocular Video Sequences, ACM Transactions on Graphics, SIGGRAPH2010, Vol.29, No.4, 2010.
7) Qt4.7, Online Reference Documentation. [Online] http://doc.trolltech.com/
8) OpenCV2.3, OpenCV v2.3 documentation. [Online] http://opencv.itseez.com/