

高解像スポーツ映像における リアルタイムボール追跡手法の一検討

鶴崎 裕貴^{1,a)} 渡邊 良亮¹ 野中 敬介¹ 内藤 整¹

概要：審判による判定に異議がある場合にビデオ判定を導入しているスポーツがある。球技においては、カメラ映像からボールを追跡することで判定に活用している。また、サッカーのような会場が広い競技においては、フィールドから離れた位置から撮影するため、ボールを捉えるためには、4K や 8K といった高解像度のカメラで撮影する必要がある。そこで本稿は、高解像度のスポーツ映像におけるリアルタイムボール追跡手法を提案する。ボール追跡手法はテンプレートマッチングを用いているが、高解像映像にそのまま適用すると、処理する画素数が膨大であるため、リアルタイムでの処理が困難となる。上記の問題は、処理が軽量のテンプレートマッチングを用いることと、テンプレートマッチングを適用する範囲を限定することで解決している。また、フレーム間の情報を用いることで追跡精度を高める手法を提案する。

キーワード：ボール追跡，テンプレートマッチング，4K/8K

A Study of Real Time Ball Tracking Method for High-Resolution Sport Video

HIROKI TSURUSAKI^{1,a)} RYOSUKE WATANABE¹ KEISUKE NONAKA¹ SEI NAITO¹

1. はじめに

スポーツにおける判定は審判による目視が基本であるが、現在では、様々な競技において、映像技術を用いた判定が行われている。陸上競技の写真判定や、テニスのホークアイ [1] などが代表的な例である。テニスや野球などの球技では、カメラ映像からボールを追跡し、判定に用いている。映像技術を用いた判定は試合中に利用され、即時性が求められており、球技におけるボール追跡は、判定までの前段階の処理となるため、素早い処理が求められる。また、ボール追跡の利用シーンとして、ボール追跡単体で利用されることは考えにくく、例えば、ボールの軌跡の表示に使われる [2]。そのため、計算リソースを節約する観点でも、処理は可能な限り軽量であり、Intel i9 シリーズと同程度の CPU でもリアルタイムに動作する手法を目指す。ま

た本稿は、サッカーを対象とした実験結果を報告するものである。

サッカーのような広い会場で行われる球技に関しては、競技が行われているフィールドから離れた位置で撮影するため、一般的なカメラの解像度である FHD、つまり 1920 × 1080 の解像度では、映像上のボールの画素数が不十分となり、ボールの追跡が困難となる問題がある。一方、4K や 8K などの高解像度の撮像機器が普及しているため、これらを使用することで上述の問題を解決できるが、解像度が高くなることで、処理する画素数が増え、処理時間が大幅に増加する問題がある。

ボールを追跡する手法はいくつか提案されている。予測と分類を用いた手法 [3]、機械学習を用いた手法 [4]、Convolutional Neural Network (CNN) を用いた手法 [5] がある。これらの手法は、非常に高い追跡精度であるが、処理が重く、CPU では処理時間が膨大になってしまうため、リアルタイム処理が困難となる問題がある。

¹ 株式会社 KDDI 総合研究所
KDDI Research, Inc.

^{a)} hr-tsurusaki@kddi-research.jp

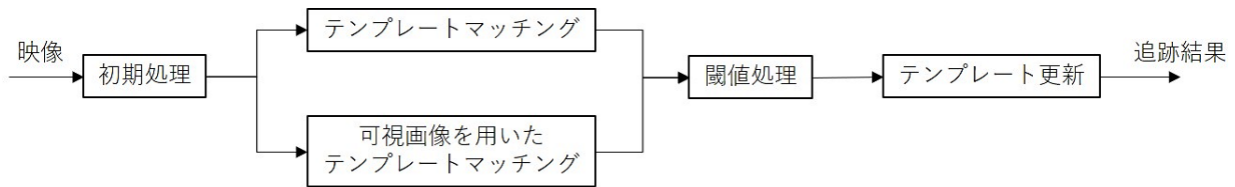


図 1 ボール追跡手法の処理手順

Fig. 1 Processing flow of proposed method.

そこで本研究は、CPU で高速に処理させるために、上述の先行研究の手法と比べて処理が軽量のテンプレートマッチングを用いたリアルタイムボール追跡手法を提案する。しかしながら、テンプレートマッチングをそのまま画像に適用すると、ボールの追跡精度が高くないため、ボール追跡手法では、フレーム間の差分を計算した結果を画像とすることで、移動した物体を可視化しているが、照明変動などでも差分値が現れてしまうため、差分値単体では使用せずに、通常のテンプレートマッチングと組み合わせて使用している。また、差分値を用いることに加えて、テンプレートを更新させることで、追跡精度を向上させている。さらに、高解像度の映像に対してもリアルタイムに動作させるために、テンプレートマッチングを適用する範囲を限定することで処理を高速化させている。

2. テンプレートマッチング

テンプレートマッチングは、検出したい物体のテンプレートをあらかじめ決めておき、画像の中からテンプレートと最も類似した位置を求める手法である。類似度の算出方法は複数考えられるが、ボール追跡手法では、式 (1) より類似度を計算している。

$$R(x, y) = \frac{\sum_{x', y'} (T(x', y') - I(x + x', y + y'))^2}{\sqrt{\sum_{x', y'} T(x', y')^2 \cdot \sum_{x', y'} I(x + x', y + y')^2}} \quad (1)$$

I を $W \times H$ 画素の入力画像、 T を $w \times h$ 画素のテンプレートとすると、テンプレートマッチングでは、テンプレートを入力画像全体に対してスライドさせ、テンプレートと重なる領域と式 (1) で求められた結果を R に保存する。画面外で処理を行わないため、 R は $(W - w + 1) \times (H - h + 1)$ 画素の配列となる。 x と y は入力画像の水平と垂直の画素位置を、 x' と y' はテンプレートの水平と垂直の画素位置を示す。類似度計算に式 (1) を用いた場合は、 R の最小値がテンプレートと最も類似した位置となる。

3. ボール追跡手法

ボール追跡手法の処理手順を図 1 に示す。図 1 の初期処理は、テンプレートの手動設定と、入力画像上のテンプレートの位置情報を保存している。テンプレートの位置情報を保存する理由は、3.4 で述べる処理の高速化に必要なためである。処理は 1 フレーム毎に行われ、初期処理は入

力映像のボールが映り込んでいる最初のフレーム (以降、初期フレームと呼ぶ) に限定して適用される。初期フレーム以降のフレームには初期処理は適用されない。

3.1 可視画像を用いたテンプレートマッチング

図 1 より、可視画像とは、1 フレーム前から現フレームで移動した物体を可視化した画像である。可視画像の作成方法は、現フレームと 1 フレーム前で差分を計算して得られた差分画像を 0 と 255 に二値化することで得られる。二値化方法を式 (2) に示す。

$$B(i, j) = F(I(i, j) - \hat{I}(i, j)) \quad (2)$$

F は閾値処理の関数であり、

$$F(a) = \begin{cases} 255 & (if \ a \geq 5) \\ 0 & (otherwise) \end{cases} \quad (3)$$

と定義する。 B は可視画像、 \hat{I} は I の 1 フレーム前の画像、 i と j は水平と垂直の画素位置、 a は現フレームと 1 フレーム前の画像の差分値を示す。可視化画像を用いたテンプレートマッチングでは、可視画像と画素値が全て 255 (白色画素) のテンプレート (以降、白テンプレートと呼ぶ) を用いて、式 (4) によりテンプレートマッチングを行う。

$$R_M(x, y) = \frac{\sum_{x', y'} (T_W(x', y') - B(x + x', y + y'))^2}{\sqrt{\sum_{x', y'} T_W(x', y')^2 \cdot \sum_{x', y'} B(x + x', y + y')^2}} \quad (4)$$

ここで、式 (4) の T_W は白テンプレートを示す。

3.2 閾値処理

ボール追跡手法は、通常のテンプレートマッチングと可視画像を用いたテンプレートマッチングの両方を現フレームに適用し、どちらかの結果を採用する。通常のテンプレートマッチングと可視画像を用いたテンプレートマッチングのどちらの結果を採用するかは、閾値処理にて決定する。閾値は 2 つあり、1 つは十分な動きがあるか、もう 1 つは最も動きがある画素範囲が現在のテンプレートとどれだけ似ているかである。この 2 つの閾値を TH_1 と TH_2 とする。

式 (5) より、 R_M が最小となる座標を求める。 R_M が最小となる座標を x_{min} と y_{min} とする。

$$(x_{min}, y_{min}) = \arg \min_{x, y} (R_M(x, y)) \quad (5)$$

x_{min} と y_{min} を用いて、式 (6) と式 (7) の両方の条件を満たした場合は可視画像を用いたテンプレートマッチングの結果を採用し、それ以外であった場合は通常のテンプレートマッチングの結果を採用する。

$$R_M(x_{min}, y_{min}) < TH_1 \quad (6)$$

$$R(x_{min}, y_{min}) < TH_2 \quad (7)$$

3.3 テンプレート更新

テンプレートの更新は、式 (8) を用いる。

$$T(x', y') = \frac{O(x', y') + \hat{O}(x', y')}{2} \quad (8)$$

O は現フレームの追跡結果の座標から $w \times h$ 画素を格納した配列、 \hat{O} は 1 フレーム前の追跡結果の座標から $w \times h$ 画素を格納した配列である。テンプレートを更新することによって、追跡対象の明るさやわずかな形状の変化に耐えられるような追跡が可能となる。また、 O と \hat{O} の平均を求めている理由は、 O に追跡対象以外の画素が含まれた場合、テンプレートを更新する毎に追跡対象以外の画素の影響が伝搬し誤追跡することを防ぐ狙いがある。

3.4 処理の高速化

Intel(R) Core(TM) i9-9900K 3.5GHz を用いて、4K のカラー映像にテンプレートマッチングをそのまま適用すると 1 フレームあたり 1 秒以上の処理時間がかかるため、提案手法では、処理の高速化のために、テンプレートマッチングを適用する範囲を限定する。30fps や 60fps の球技の映像において、1 フレーム間でボールが数百画素単位で移動することは考えにくい。解像度が 4K (4096 × 2160)、フレームレートが 30fps のサッカーの試合において、最大で 47 画素、平均 10 画素程度の移動量であった。そこで、一度ボールを検出後、次フレームからは、ボールを検出した位置から周囲数十画素のみをテンプレートマッチングを適用する範囲とすることで処理の高速化を実現する。また、入力映像がカラーの場合は、処理される範囲をグレースケールにしてからボール追跡手法を適用することで、更に処理時間を短縮している。

4. 実験結果と考察

解像度は 4K (4096 × 2160)、フレームレートは 30fps で、全 538 フレームのサッカーの映像を実験で使用している。CPU は Intel(R) Core(TM) i9-9900K 3.5GHz を用いている。閾値は $TH_1 = 0.5$ 、 $TH_2 = 0.05$ に設定している。また、高速化におけるテンプレートマッチングを適用する範囲は、1 フレーム前の追跡結果の画素位置を中心に、 101×101 画素である。テンプレートのサイズは 12×12 画素である。本稿における追跡成功率とは、全フレームにおける、ボールを正しく追跡できたフレームの割合を示している。

表 1 テンプレート更新と可視画像ありとなしの比較 (追跡成功率及び 1 フレームあたりの処理時間)

Table 1 Comparison of results between ON and OFF of template update and visualized image. (rate of tracking succeed and processing time per frame)

		可視画像	
		なし	あり
テンプレート更新	なし	15.4% (0.9ms)	15.4% (1.4ms)
	あり	15.6% (0.9ms)	94.6% (1.4ms)

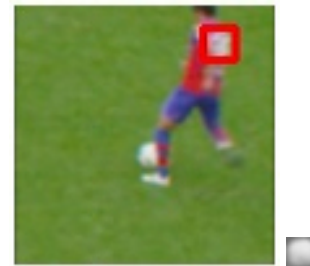


図 2 可視画像ありテンプレート更新なし (第 83 フレーム)

Fig. 2 Visualized image ON, Template update OFF (frame no.83).



図 3 可視画像なしテンプレート更新あり (第 84 フレーム)

Fig. 3 Visualized image OFF, Template update ON (frame no.84).

通常のテンプレートマッチングのみを用いた場合は、追跡成功率が 25.8% であった。これは、追跡途中で映像に映り込んでいるボールの形状や明るさが変化したことにより、ボールの追跡ができなかったためである。また、処理時間は 1 フレームあたり平均で 1268ms かかっている。

次に、テンプレート更新と可視画像がありとなしの場合の比較を表 1 に示す。表 1 の可視画像なしテンプレート更新なしは高速化された通常のテンプレートマッチングのみの結果である。表 1 より、可視画像もテンプレート更新も単体では追跡率成功向上にはほとんど寄与していない。一方、可視画像とテンプレート更新ありの場合、追跡成功率が大幅に向上している。また、処理時間に関しては、高速化したことにより、1400 倍以上向上している。これらについて考察を行う。

図 2 は、可視画像ありテンプレート更新なしの最初に誤追跡が発生したフレームの画像である。併せて、その時のテンプレートも示す。図 2 より、可視画像を用いた場合

でも、テンプレートが更新されていないため、図2のようにボールがわずかにかけている場合に誤追跡が発生してしまう。

図3は、可視画像なしテンプレート更新ありの最初に誤追跡が発生したフレームの画像である。図3より、テンプレートは更新されているものの、追跡途中でテンプレートにボール以外の画素の割合が増えていったため、第84フレームで誤追跡が発生してしまった。

図4に、可視画像ありテンプレート更新ありの最初に誤追跡が発生したフレームの画像を示す。図4より、ボールが選手の脚に隠れてしまっていたため、誤追跡が発生している。これは、ボール追跡手法においてボールがフレーム内に映り込んでいない場合が考慮されていないためである。また、可視画像ありテンプレート更新ありの場合、第83及び84フレームにおいても正しくボールを追跡できている。

これらの結果から、テンプレートは更新しつつも、更新時にはボール以外の余計な画素が含まれないように可視画像を用いて、精度の高い追跡を行う必要があると言える。

最後に、2つの閾値について考察を行う。 TH_1 は、値を大きくすると可視画像を用いたテンプレートマッチングが採用されやすくなる。 TH_2 は、値を大きくすると TH_1 と同じく可視画像を用いたテンプレートマッチングが採用されやすくなるが、追跡結果が現フレームのテンプレートと似ていなくても採用されるようになるため、誤追跡となる可能性が高くなる。逆に値を小さくすると通常のテンプレートマッチングが採用されやすくなるが、通常のテンプレートマッチングでは、十分な追跡精度が得られないため、値の設定は慎重に行う必要がある。

5. まとめ

本稿は、高解像スポーツ映像におけるリアルタイムボール追跡手法を提案した。ボール追跡手法は、現フレームと1フレーム前との差分を計算することで、物体の動きを可視化した画像を作成し、テンプレートマッチングに用いることと、テンプレートを更新することで追跡成功率を向上させている。また、テンプレートマッチングを適用する範囲を限定することで処理時間を大幅に短縮させた。しかしながら、誤追跡が発生し、ボールが処理範囲外に出てしまった場合に、追跡できなくなる問題とボールが人などで隠れてしまった場合の処理がないため誤追跡となる問題があるが、これらは今後の課題とする。

参考文献

- [1] N. Owens, C. Harris and C. Stennett, "Hawk-eye tennis system", International Conference on Visual Information Engineering VIE 2003, pp.182-185, 2003.
- [2] M. Takahashi, K. Ikeya, M. Kano, H. Ookubo and T. Mishina, "Robust Volleyball TrackingSystem using

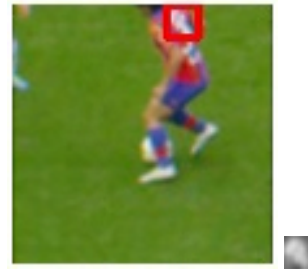


図4 動き情報ありテンプレート更新あり (第458フレーム)
Fig. 4 Visualized image ON, Template update ON. (frame no.458)

- [3] Z. Deng, Y. Hou, X. Cheng and T. Ikenaga, "Multi-Peak Estimation for Real-Time 3D Ping-Pong Ball Tracking with Double-Queue Based GPU Acceleration", IEICE Trans. on Information and Systems, vol.E101, no.12, pp.1251-1259, 2018.
- [4] S. Sarkar, A. Chakrabarti and D. P. Mukherjee, "Generation of Ball Possession Statistics in Soccer Using Minimum-Cost Flow Network", Proc. of IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2019.
- [5] V. Renó, N. Mosca, R. Marani M. Nitti, T. D'Orazio and E. Stella, "Convolutional Neural Networks based ball detection in tennis games", Proc. of IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2018.