CH-001

見失いに頑健な検出混合型 PTZ 追跡システム
A Detection-mixed PTZ Tracking System Robust to Target Loss

小川 拓也† 比嘉 恭太† 谷内田 尚司† 柴田 剛志†
Takuya Ogawa Kyota Higa Shoji Yachida Takashi Shibata

1. はじめに

不審者等の監視において、従来固定カメラが用いられてきた。しかし、固定カメラは画角が固定されてしまうため、監視するターゲットが画角外に出た場合は、監視を継続することができなかった。そのため、監視には固定カメラに加え、PTZ カメラの導入が進んでいる。PTZ カメラは、カメラの向きやズームを制御して画角を変更できるため、固定カメラに比べて広範囲かつ高精細にターゲットを監視することが可能である。しかしながら、PTZ カメラを制御してターゲットを画角中心に捉え続けるためには、ターゲットの移動に合わせてカメラの旋回方向・旋回量を調整する必要があり、マニュアル操作で継続するには卓越した技能が必要となる。そこで、自動で PTZ カメラを制御しターゲットを捉え続ける PTZ 追跡システムが必要とされている。PTZ 追跡システムについては、すでにいくつか提案されているが、実用性における課題が２つ存在している。１つ目は、事前にターゲットが画角中心に映っている必要があること、２つ目は、追跡中に発生する遮蔽や急激な見えの変化に対応しておらず容易にターゲットを見失うことである。

そこで、本稿では物体検出を活用することで画角全体からのターゲット選択を可能にし、かつ遮蔽や急激な見えの変化による見失いを軽減する、検出混合型の PTZ 追跡システムを提案する。

2. 関連研究

PTZ 追跡システムは、これまでもいくつか提案されている[1][2][3][4][5]。これらは、基本的には図 1 左のようにカメラ向き(水平方向 Pan と垂直方向 Tilt)を変更できる Pan-Tilt 雲台(以降、単に雲台という)と、Zoom 倍率を変更できるカメラ、それらを制御する PC から構成され、図 1 右のようにターゲットとして設定した画像特徴が常に画角中心となるように制御することで追跡する。例えば、[4]では画角中心に映るターゲットの画像領域から HSV 特徴量を抽出し、次のターゲットの位置を mean shift 法によって推定、ターゲットが画角中心となるように雲台を制御する。また、[5]では画角中心に映るターゲットの画像領域から RGB 特徴量を抽出し、Cam Shift 法によってターゲットの位置を追跡、ターゲットが画角中心になるように雲台を制御しつつ、ターゲットが一定の大きさで映るようにカメラの Zoom 値を制御する。

ただ、これら従来の PTZ 追跡システムは、事前にターゲットを画角中心に捉えておく必要があった。しかし、ターゲットが移動している場合、ターゲットの移動に合わせてマニュアル操作でカメラの旋回方向・旋回量を制御し画角中心にターゲットを捉えておくことは、卓越した技能が必要であるため、実運用上、追跡対象の選択は容易ではなかった。また、追跡中に、ターゲットの見えの急激な変化や他のものとのすれ違い等による遮蔽が発生すると、ターゲットを見失う場合がある。これらは実運用上頻繁に発生するが、明確な対策が従来の PTZ 追跡システムにおいては考えられておらず、見失った場合は再度マニュアルでターゲットを探し出して設定するしかなかった。

†NEC データサイエンス研究所, Data Science Research Labs, NEC Corporation

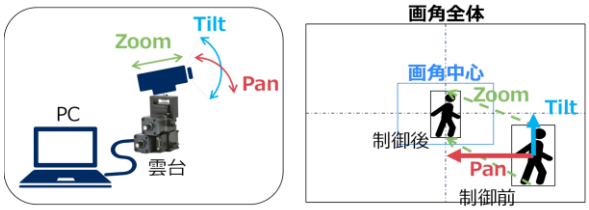

図 1　PTZ 追跡システムの構成(左)と画像上での動き(右)

図 2　検出結果の出力遅延による枠ズレ

3. 提案方式

そこで、本稿では、前章で述べた従来の PTZ 追跡システムにおける課題を改善する方式を提案する。事前に画角中心にターゲットを捉えておく必要がある課題については、画角全体で物体検出を行い、検出枠によりターゲット選択を容易にすることで、ターゲットが画角中心に映っていない場合においても自動追跡を可能にする。このターゲット設定についての詳細は 3.1 節で述べる。また、見えの急激な変化や遮蔽による見失いに対する課題については、追跡と並行して適応的に検出を行うことで、時空間的連続性を重視する追跡の弱点を補完し、見失いへの改善を試みた。この検出混合型追跡についての詳細は 3.2 節で述べる。

3.1 ターゲット設定

提案方式では、画角全体における検出結果の中からターゲットを選択し、選択したターゲットが画角中心に選択時の大きさで映るようカメラと雲台を制御する。ターゲットの選択には、物体検出結果の枠を用い、物体検出方式には、yolo9000[6]を用いる。yolo9000 は物体検出結果として、物体の検出領域とその物体のカテゴリを同時に出力する。今回は、出力された検出領域の枠内をクリックすることで、ターゲットの設定を実現する。

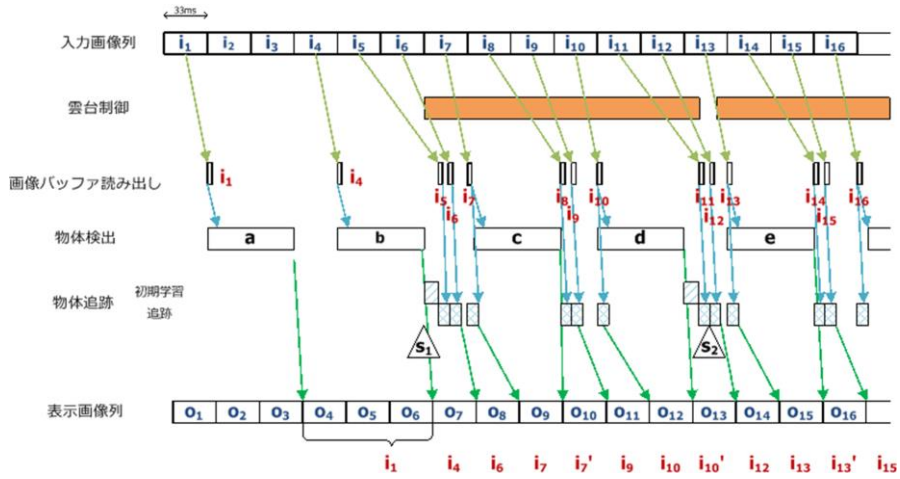

図 3　提案システムのスケジューリング

しかしながら、yolo9000 は深層学習を用いた方式であり、他の深層学習方式同様、処理速度が課題である。本稿で用いたノート PC のスペック(Windows 7 64bit, Core i7-6700, 2.6GHz, 32GB RAM, GeForce GTX-970M GPU)では、yolo9000 による検出は HD 画像 1 フレームあたり平均 67ms かかる。これは、一般的な映像(30fps, フレーム間隔に換算すると 33ms)をリアルタイムに扱う場合、およそ 3 フレーム遅れて検出結果が表示されることを意味する。そのため、図 2 のようにターゲットと検出枠の間にズレが生じる可能性がある。そこで、図 3 の物体検出 a に関するフローに示すように、リアルタイムに伝送されてくる画像フレーム列を一度入力画像列$\{i_1, i_2, \ldots\}$としてバッファリングする。バッファにあるi_1の検出を終えてから、画像とともに検出結果枠を表示することで、枠のズレを解消し、ターゲットを正しく設定可能にする。こうして設定したターゲットをA_tとする。

ここで、A_tの追跡処理に移行する前に、検出を活用して遅延の改善を施す(図 3 の物体検出 b のフロー参照)。表示画像o_6でA_tがクリックされ設定された場合、そのままではi_2から追跡開始となり、遅延は 5 フレーム存在する。そこで、ターゲット選択時に別スレッドで動作している検出 b の検出結果を用いてA_tと同一のターゲットA_{t+1}を、検出した入力画像i_4から探索する。各検出結果の領域を$R_{A_t}, R_{A_{t+1}}$、各検出結果のカテゴリを$C_{A_t}, C_{A_{t+1}}$、A_tとA_{t+1}の重複率αを$(R_{A_t} \cap R_{A_{t+1}})/(R_{A_t} \cup R_{A_{t+1}})$とすると、

$$\alpha > 0.0, \quad (1)$$

$$C_{A_t} = C_{A_{t+1}} \quad (2)$$

を満たすA_{t+1}があれば、A_{t+1}をターゲットとして更新する。満たすものが複数ある場合は、αが最大のA_{t+1}を採用する。これによって、i_2ではなくi_5からの追跡処理開始となり、遅延は 2 フレームに改善される。残る遅延については後段の処理で解消する。詳細は 3.2.3 節で説明する。

次に、画角中心に選択したターゲットが映るよう、雲台を制御する(選択時の大きさにズーム制御するためここではズームは行わない)。ターゲットの画像中の重心座標を(x_0, y_0)、画像中心の座標を(x_c, y_c)とすると、画角上の水平方向移動量X、垂直方向移動量Yの初期値X_0, Y_0は次のようになる。

$$X_0 = x_c - x_0, \quad (3)$$

$$Y_0 = y_c - y_0. \quad (4)$$

これに基づき雲台に制御信号を出力する(図 3 のS_1)と同時に、設定したターゲット情報をもとに次節の追跡処理に移行する。このとき、雲台の動作完了は待たずに追跡処理に移行することで、雲台動作による遅延を解消する。

3.2 検出混合型 PTZ 追跡

本稿の追跡処理では、見失いの課題に対処するため、いわゆる追跡に加えて検出も並行して行い、よりターゲットらしい方を適宜判断しながら、ターゲットが画角中心に一定の大きさで映るようにカメラ、雲台を制御する。追跡方式、検出の混合、それを実現するための遅延吸収処理について、以降で詳しく説明する。

3.2.1 物体追跡方式 DMM

我々は、映像上で指定された物体を高速・高精度に追跡する Object Tracking with Dynamic Mask and Model (DMM)[7]という方式を既に提案しており、本稿における追跡にも活用する。DMM は、ターゲット領域中に混在する背景を動的に排除しながら Median Shift という特徴点ベースの追跡を行う、背景変化やターゲットの形状変化に頑健で、かつ演算量を抑えた追跡方式である。処理速度に関しては、HD 画像 1 フレームあたり 9ms(追跡開始フレームでは初期学習も行うため 20ms)であり 30fps の映像において遅延なく動作する。

3.2.2 検出の混合

DMM のみの追跡システム構築も可能であるが、本稿では、より見失いに対して頑健にするため、DMM と並行して適宜検出を行い、よりターゲットらしい方を選び追跡していく。検出方式は、3.1 節のターゲット設定時同様、yolo9000 を活用する。検出を行う詳細なタイミングについては 3.2.3 節で説明する。DMM では、追跡結果のターゲット領域R_Tに加え、追跡開始から取得しているターゲットの見え特徴との類似度であるターゲット尤度$L_T(0.00 \leq L_T \leq 1.00$、値が大きいほどターゲットらしい)が得られる。一方、yolo9000 では、検出結果において、ターゲット領域に加えカテゴリC_D、物体尤度$L_D(0.00 \leq L_D \leq 1.00$、値が大きいほ

ど物体らしい)が得られる。これらを用いて、そのフレームのターゲット領域R_tを決定する。R_tに関する条件式を以下のように定める。

$$R_t = \begin{cases} R_T\ (if\ L_T \geq 0.4) \\ R_D\ (if\ L_T < 0.4\ and\ L_D \geq 0.3\ and\ C_D = C_0\ and\ d < r) \\ R_{t-1}\ (otherwise). \end{cases}$$

上記に示す通り、一時的に見失ったと判断した場合は、一つ前のフレームのターゲット領域R_{t-1}をR_tとする。ここで、C_0はターゲット設定したときの検出結果のカテゴリである。d は式(5)で導出される、見失った位置から当該検出結果までの距離である。r は式(6)で導出される検出半径であり、R_{t-1}の重心からの半径rに存在する検出結果に絞っている。

$$c = \begin{cases} 0 & (if\ R_t = R_T,\ R_D) \\ c+1 & (otherwise\ \ (R_t = R_{t-1})), \end{cases}$$

$$d = \sqrt{(x_{t-1} - x_D)^2 - (y_{t-1} - y_D)^2}, \quad (5)$$

$$r = r_{R_{t-1}} + c * \overline{v}. \quad (6)$$

ここで、x_{t-1}, y_{t-1}はR_{t-1}の重心座標、x_D, y_DはR_Dの重心座標とする。また、$r_{R_{t-1}}$をR_{t-1}の外接円の半径、$\overline{v}$ をターゲットの直近 3 フレームでの平均移動速度(pixel/frame)とする。

検出で出力されたターゲット領域R_Dがそのフレームのターゲット領域R_tに選ばれた場合、DMM は再度初期学習を行う。これは、DMM の想定を超える変化をターゲットが起こしたと解釈し、変化後のターゲットに適応した学習をするためである。

3.2.3 バッファスケジューリングによる遅延の吸収

本節では、検出・雲台制御などによる遅延を、システム全体として吸収する方法とその詳細を説明する。入力される映像は一般的な 30fps とする。システムは 1 秒に 30 フレーム画像を取得するので、入力フレーム間隔はおよそ 33ms となる。この間隔より早く検出・追跡・雲台制御のすべての処理が実行・完了できれば、毎フレームリアルタイム処理が可能である。しかし、1 フレームあたりの処理時間は検出の場合 67ms、追跡は初期学習時が 20ms でそれ以降が 9ms、そして雲台制御可能間隔は 100ms(雲台制御完了のまで時間は制御量や制御速度による)のため、単純に直列でスケジューリングすると 1 フレームあたりの処理に最大で 187ms 以上かかる。この問題について、処理するフレームを間引く対策を講じると、処理フレーム間でのターゲットの位置や見えの変化が大きくなり、追跡が難しくなってしまう。そこで、図 3 に示すように、画像バッファ読み出し・検出・追跡・雲台制御のスレッドに分けスケジューリングを行うことで各処理がシステムに起こしうる遅延を吸収し、システムのリアルタイム動作を可能にする。ターゲット選択におけるスケジューリングについては 3.1 節で述べたため、ここでは、それ以降のスケジューリングについて図 3 をもとに説明する。まず、バッファに溜まった入力画像列$\{i_5, i_6, i_7\}$に対して順次 DMM による追跡を行い、その高速性から入力映像に対する遅延を吸収していく。バッファが残り 1 フレームになり遅延を解消する目処がたつと、DMM と同時に Yolo9000 による検出を行い(図 3 の検出 c)、見失いの防止を狙う。ここで、追跡中の雲台制御について述べる。あるフレームtの追跡処理が終了しターゲット移動量(X, Y)が決定された段階で、雲台が制御可能かチェックする。ただし、雲台の動作間隔は最低でも 100ms

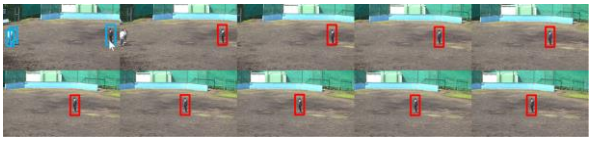

図 4　画角中心外ターゲットの設定結果

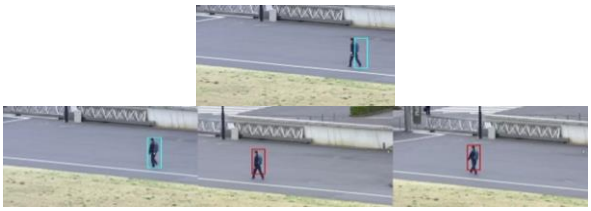

図 5　移動状態であるターゲットの設定結果
上（従来方式）、下（提案方式）

のため、制御を毎フレーム行うことはできない。そこで、雲台が動作中のため新たな制御ができない場合は、その起点フレームt_1からのフレーム間のターゲット移動量(X, Y)を逐次積算し、(X_{sum}, Y_{sum})として保持しておく(式(7)参照)。また、ズーム値 Z は、R_iの面積を$S(R_i)$で表すとすると、式(8)で算出される。そして、追跡処理が終了した後の雲台動作可否判断時において雲台動作が可能だった場合(図 3 におけるS_2)、積算した移動量とズーム値をもとに雲台制御を開始し、各値をリセットする($(X_{sum}, Y_{sum}, Z) = (0,0,1.0)$)。こうして雲台動作によるシステム全体の遅延を回避する。

$$(X_{sum}, Y_{sum}) = \sum_{t=t_1}^{t_n} (X_t, Y_t) \quad (7)$$

$$Z = S(R_{t_n}) / S(R_{t_1}) \quad (8)$$

4. 実験

監視のターゲットとして主要である、人物を対象に実験を行った。建物の高所に機材を設置し、地上の歩行者を俯瞰する画角で実験を行った。実験機材は次の通り。ノート PC(Windows 7 64bit, Core i7-6700, 2.6GHz, 32GB RAM, GeForce GTX-970M GPU)、雲台(PTU-E46)、可視カメラ(FCB-EV7520)。

4.1 ターゲット設定の有効性検証

3.1 節では、ターゲットが移動中であっても画角全体からターゲット設定が可能な方式を提案した。その有効性の検証のため、まずは画角中心外で静止したターゲットを提案方式で選択する実験を行った。結果を図 4 に示す。図 4 左上では画角検出された枠を示しており、その検出枠をクリックしターゲットを設定する。そうすると、雲台制御により徐々に画角中心にターゲットが映るよう画角が移動し、図 4 右下では、正常にターゲットを画角中心に捕捉し雲台制御が止まったことが確認できる。これにより、画角中心外に映るターゲットの設定が実現できたことを確認した。

また、ターゲットが移動中の場合についても、画角全体から検出した結果を用いてターゲットを設定する実験を行った。その結果を図 5 に示す。図 5 上には、画像バッファを用いずにターゲットをクリックし選択した場合の結果を示した。ターゲットが移動状態であるため、ターゲットと

図 6　柱による遮蔽発生時の追跡結果

図 7　すれ違いによる遮蔽発生時の追跡結果

は少しズレのある枠を出力していることが分かる。こちらをもとに追跡してしまうと、追跡は失敗し見失いやすくなる。図 5 下の提案方式は、画像バッファにより画像と検出枠が同期されているため、検出枠が正しく表示をされていることが分かる。選択後、ターゲットが移動を続けているため、始めは図 5 右中央のように画角左にターゲットが寄って映っているが、やがて落ち着き図 5 右下のようにターゲットが画角中心に捉えられたことが分かる。

4.2 遮蔽・急激な見えの変化への有効性検証

3.2 節では、ターゲットが遮蔽、あるいは急激に見えが変わる場合においても追跡を継続できる方式を提案した。その有効性の検証のため、これらに対する実験を行った。遮蔽への頑健性に対する実験としては、ターゲットが柱に一時隠れて再び現れるシナリオと、ターゲットと別の人物とがすれ違うシナリオを用意し実験した。実験結果をそれぞれ図 6,7 に示す。図 6 では柱による遮蔽の後、向きを変えてターゲットが再び出現している。遮蔽したことにより、ターゲットの位置や特徴も遮蔽前と変わっているが、図 6 に示す通り、提案手法は追跡を継続できた。また、図 7 を参照すると、ターゲットの手前と奥に別の人物がすれ違い一時遮蔽した場合でも、検出によってターゲットを見失うことなく追跡を継続できていることが分かる。

急激な見えの変化への頑健性に対する実験としては、ターゲットが前面から側面に向きを変えるときに起こる見えの変化をシナリオとして用意した。提案した検出混合型追跡の効果が分かるよう、DMM による追跡のみでターゲットが向きを変えたケースと、提案する検出混合型追跡でターゲットが向きを変えたケースで実験を行った。それぞれの実験結果を図 8,9 に示す。図 8 では、ターゲットが向きを変え見えが急激に変わったことによってターゲットを見失ったのに対し、図 9 では、追跡を継続できていることが分かる。

これらの実験から、提案手法は移動中のターゲットであっても画角全体から選択が可能で、従来見失ってしまうようなターゲットの遮蔽や急激な見え変化が起こっても頑健に追跡できることを確認した。これらは PTZ 追跡システムの実用性向上に大きく貢献する結果である。

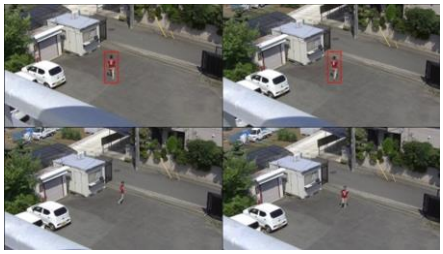

図 8　急激な見えの変化発生時の追跡結果
（検出非混合時）

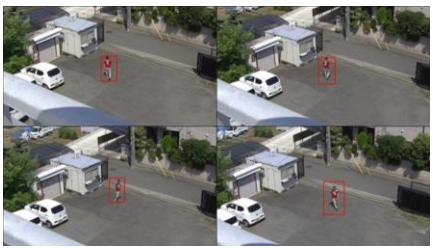

図 9　急激な見えの変化発生時の追跡結果
（提案方式）

5. おわりに

本論文では、不審者等の監視を目的として、見失いに頑健な検出混合型 PTZ 追跡システムを提案した。我々は、システムの実用性を向上させるため、監視ターゲットを画角全体で検出し、その中から追跡ターゲットを選択する方式を採ることで、追跡開始を容易にした。また、ターゲットの見失いを軽減するため、追跡と並行して適応的に物体検出を行い、急激な見えの変化や遮蔽に対応した。実験の結果、移動中のターゲットであっても PTZ 追跡を開始・継続でき、ターゲットに急激な見えの変化や遮蔽があっても PTZ 追跡を継続できたことを確認した。

参考文献

[1] Sanjay Singh, Chandra Shekhar, and Anil Vohra, “Real-Time FPGA-Based Object Tracker with Automatic Pan-Tilt Features for Smart Video Surveillance Systems”, Journal of Imaging, 3(2), 18 (2017).
[2] Sangkyu Kang, Joon-Ki Paik, Andreas Koschan, Besma R. Abidi, Mongi A. Abidi, "Real-time video tracking using PTZ cameras", International Conference on Quality Control by Artificial Vision, Vol. 5132, pp. 103-112 (2003).
[3] C. S. Yang, Ren-Hao Chen, Chao-Yang Lee and Shou-Jen Lin, "PTZ camera based position tracking in IP-surveillance system," International Conference on Sensing Technology, pp. 142-146 (2008).
[4] Faliang Chang, Guoqiang Zhang, Xiaolin Wang, and Zhenxue Chen, “PTZ Camera Target Tracking in Large Complex Scenes”, World Congress on Intelligent Control and Automation (WCICA), pp. 2914-2918 (2010).
[5] Zhang Qigui, and Libo, “Search on Automatic Target Tracking Based on PTZ System”, International Conference on Image Analysis and Signal Processing (IASP), pp. 192-195 (2011).
[6] Joseph Redmon, and Ali Farhadi, “YOLO9000: Better, Faster, Stronger”, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6517-6525 (2017).
[7] Takuya Ogawa, Kyota Higa, Kengo Makino, Shoji Yachida, and Katsuhiko Takahashi, “A Robust and Fast Object Tracking Method Using A Dynamic Mask and An Adaptive Search”, IEEE International Conference on Image Processing (ICIP), (forthcoming in 2018).