Vol.2018-CVIM-210 No.4
2018/1/18

動画像コーデックにおける動きベクトルを用いたCNN物体検出の負荷緩和

氏家 隆之[1,a)]　廣本 正之[1]　佐藤 高史[1]

概要：畳み込みニューラルネットワーク（CNN）を用いたリアルタイム物体検出手法は高い検出性能を実現できることが知られているが，演算量やパラメータ数が大きく組込み機器などエネルギー制約が厳しい環境への実装が課題である．本稿では，動画像の符号化時に得られる動きベクトルを活用することで，I フレームでの物体検出と P フレームでの補間による追跡を行なってリアルタイム物体検出を効率よく実現する手法を提案する．提案手法を複数物体追跡データセット MOT16 に適用し検出頻度と複数物体追跡の総合評価指標 MOTA のトレードオフを求めた．検出頻度を 1/12 に削減する場合，基準手法に対し約 88%の MOTA が保たれることを確認した．

Load Mitigation of CNN-Based Object Detection Utilizing Motion Vectors in Video Codecs

Takayuki Ujiie[1,a)]　Masayuki Hiromoto[1]　Takashi Sato[1]

1. はじめに

パターン認識技術の発展に伴い，IoT デバイスや車載システムなど様々な組込み機器において画像認識技術の導入が進められている．また近年，深層学習と呼ばれる高精度なパターン認識技術が発展を遂げており，画像や音声，文章などの複雑なデータを対象とした認識タスクで優れた性能を実現できるモデルが報告されている．その中でも，畳み込みニューラルネットワーク (Convolutional Neural Network; CNN) は，特に画像認識において優れた性能を達成できることが知られている [1,2].

近年, 物体検出タスクにおいて高い性能を示す CNN [2–5] が広く利用されるようになり，より発展的な課題として複数物体追跡タスク [6] での性能向上を図る研究が進展している．主にオフラインで追跡性能を重視して動画中の物体追跡を行う手法として，文献 [7,8] 等が挙げられる．これらの手法は追跡物体として人物に特化し，各フレームの検出器のみならず姿勢推定やフレーム間の対象同定にも CNN を利用することで正確な追跡を行うことを目標としている．また，オンラインでリアルタイムに動画像ストリームの物体追跡を行う手法として，文献 [9,10] が挙げられる．これらの手法は，各フレームで CNN や DPM [11] などの検出手法を用いそちらに主要な負荷を割きつつも，時空間的な追跡処理にかかる負荷を抑えることでリアルタイムに比較的正確な追跡を行うことを目標としている．

一方で，エネルギー制約の厳しい組込み環境では，これら複数物体追跡手法の前提となる CNN 等の物体検出ネットワークをリアルタイムに各フレームに適用すること自体が多くの場合課題となる．このため既存手法のように，これに加えて時空間的な特徴を抽出する検出器を追加することは現実的でなかった．そこで本研究では，検出器として CNN を用いることで検出性能を確保しつつ，可能な範囲でエネルギー負荷の小さい複数物体追跡手法をその上で構築することを狙う．そのため，既に組込み環境で専用チップ等の開発が進み十分最適化された動画像の解析手法としての動画像コーデックに着目し，その演算過程で生じる副次的な結果を利用し，物体追跡の演算量を低減する．

なお，動画圧縮に付随する情報を画像認識に用いる研究

[1] 京都大学 大学院情報学研究科
[a)] paper@easter.kuee.kyoto-u.ac.jp

も存在する．特に文献 [12] では，動画より抽出した動きベクトルを用いて効率的にオプティカルフローを構成する手法が提案されている．またこの研究を踏まえ，文献 [13] は本稿で試みるように動きベクトルと CNN を活用した動作認識手法を提案している．文献 [13] の目標は本研究と近いが，本研究では物体追跡タスクを対象として検出結果を補間することで，より大幅に演算量を削減することを目的とする．

本稿では CNN によるリアルタイム物体検出を効率良く実現するための手法として，動画像コーデックの符号化過程で生じた動きベクトルを用いて基準フレームの検出結果を追跡する手法を提案する．提案手法では，動画像コーデックにおける I・P フレーム等のフレーム種別に応じ，各フレームで異なった追跡処理を行う．他フレームを参照せずに独立でフレームを圧縮・復元する I フレームのフレーム画像に対しては CNN 検出器を用いた高精度な物体検出を行い，当該フレームより前のフレームを参照する P フレームでは，前フレームの検出結果に基づいてバウンディングボックスを補間する．I フレーム間隔（GOP サイズ）が N の場合，本手法により CNN による物体検出頻度が毎フレーム検出を行う基準手法の $1/N$ に抑えられ，一定の範囲で検出性能を維持できることを示す．

2. 準備

2.1 Tracking-by-Detection

Tracking-by-Detection [14, 15] は，動画における複数物体追跡で広く用いられるアプローチである．本節では Tracking-by-Detection の典型的な実現例である Geiger らの手法 [14] に基づき, Tracking-by-Detection の概要を示す．

典型的な Tracking-by-Detection は，(1) Detect (2) Predict (3) Associate の三つのステップの処理を行うことで実行される．Tracking-by-Detection の手法全体のフローを要約すると，**図 1** として表される．Detect によって生成された各フレームのバウンディングボックス集合 (BBox Union)（以下, BB 集合）が, Predict 及び Associate によって時系列で伝搬している．

Detect フェーズでは単一フレーム画像を入力とする検出器を用い，物体の候補となる複数のバウンディングボックスを生成する．代表的な検出器としては，DPMv5 [16] などの特徴量記述に基づく従来型の検出器に加え，近年では R-CNN 系 [2, 17, 18]，また Single Shot 系 [3–5] と分類されるような CNN による検出器が知られている．Tracking-by-Detection では検出器の性能は追跡性能全体に大きな影響を及ぼすため，処理速度とのトレードオフを踏まえた上で可能な限り高精度な検出器が利用される．

Predict フェーズでは，Detect フェーズで得られたバウンディングボックスをフィルタリングしてフレーム系列での物体の軌跡における不確実性を取り除く処理が行われる．Predict の典型的な実現として，カルマンフィルタが挙げられる．カルマンフィルタは時系列上での状態遷移が線形変換で表現できるシステムを対象とする時系列データの予測モデルである．カルマンフィルタがターゲットとするシステムの一般形は以下の様に表される [19].

$$x_{k+1} = Ax_k + Bu_k + w_k \tag{1}$$

$$z_k = Hx_k + v_k \tag{2}$$

$$p(w) \approx N(0, Q) \tag{3}$$

$$p(v) \approx N(0, R) \tag{4}$$

それぞれ式 (1) がシステムの状態遷移を表すプロセスモデル，式 (2) が状態を観測した際の変化を表す観測モデル，式 (3), 式 (4) がノイズの確率分布を示す．但し，$N(a, b)$ は a を平均とする分散 b の正規分布を表す．用いられる変数としては x_k が状態，u_k が外乱入力（制御入力），w_k, v_k がプロセスノイズ，観測ノイズを表す．Tracking-by-Detection では，カメラや検出器の不確実性の補正を目的として上式の各モデルにおいて定数行列が用いられる．

Associate フェーズでは，前フレームの Predict を施したバウンディングボックスと次フレームで検出したバウンディングボックスを対応付けて固有の ID を割り当てる処理を行う．Associate の典型的な実現としては, ハンガリアン法 [20] による最適マッチングが挙げられる．ハンガリアン法を適用するにあたって, フレーム間でのバウンディングボックスの遷移しにくさを表すコスト関数 (Affinity) $f(\mathrm{BB_A}, \mathrm{BB_B})$ が予め定められる．二つの連続するフレーム X, Y における BB 集合をそれぞれ $\{\mathrm{BB^X}_i\}_{i=1,\cdots,N}, \{\mathrm{BB^Y}_j\}_{j=1,\cdots,M}$ とすると，両フレームの完全 2 部グラフ対応付けによりフレーム間のバウンディングボックスの遷移に係るコスト行列が算出される．このコスト行列をハンガリアン法の入力として解くとコストの総和が最小となる最適マッチングが得られるため，前フレームのボックス ID をマッチングに則って次フレームに伝搬する．この際，マッチングの対象とならなかったボックスでは ID の生成・破棄が起こる．

第 3 章では，本稿の提案手法を図 1 と対比する形で示す．

2.2 動画像コーデックにおける動きベクトル

提案手法の一部として使用する動画像コーデックの動き補償・予測について，MPEG-2 [21] による符号化処理を例に概要を説明する．

MPEG-2 は動画圧縮コーデックにおける標準規格の 1 つである．動画として連続的に処理されるフレーム画像は，動画コンテナ上で直接保持されるのではなく離散コサイン変換 (DCT), 可変長符号化, DCT 係数の量子化などのデータ圧縮処理を施された上でシーケンスとして保持される．

圧縮処理の中でも特に大きな役割を果たしているのがフレーム間予測による動き補償である．一般に動画圧縮の標

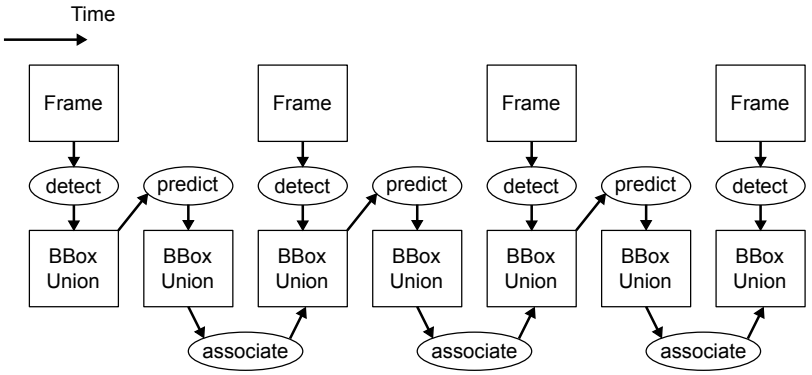

図 1 Tracking-by-Detection

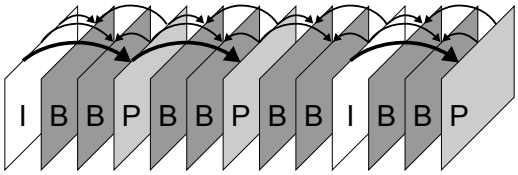

図 2 フレーム間予測

準規格 [21–23] では，動画の各フレームをIフレーム, P フレーム, B フレームと分類しフレーム毎に復号化で行う処理を分ける．Iフレームは他フレームを参照せずに単一のフレームとして復元できる情報を保持しており，動き補償を行う大元の基準となる．P フレームは後方（過去）のフレームを基準にオフセットを表す情報として動きベクトルを保持し，基準フレーム及び動きベクトルによる動き補償処理を施すことで復号化される．B フレームは後方のみならず前方（未来）のフレームも参照し両フレームの補間結果からの差分情報を保持しており，前後のフレームを双方向的に参照し復号化する．**図 2** に，フレーム間の参照例を示す．この際，各フレーム間の矢印は基準フレームと参照しているフレームを表す．一般に P フレームや B フレームの比率が高いほど動画の圧縮率は高くなる一方で，復号化処理は煩雑になり高い処理能力が求められる．

動画中の各フレームは 16×16 ピクセル等のマクロブロックと呼ばれる小領域に分割され，復号化の処理単位となる．動き補償の際には，現フレームから補償先フレームへのマクロブロックの移動オフセットを表す動きベクトルと呼ばれる要素が，マクロブロック毎に参照され処理単位となる．動きベクトルは動画コンテナに符号化された状態で保持され，動き補償過程で復号化される．復号化された動きベクトルは現フレームにおいてマクロブロック内の各ピクセル座標に加算され，補償先フレームを生成する．

規格では，これらデータ圧縮処理について施す処理が階層的に幾つかのサブセットに分けられ，行う処理のサブセットに応じてプロファイル，また求められる圧縮レートに応じてレベルという概念が定義され，実際の符号化の際にはプロファイルとレベルを定めた上で処理を行う．本稿で用いるプロファイルでは，これらの内Iフレームと P フレームのみがコンテナ内に含まれるよう定義され，B フレームにおける双方向的な予測は行わないものとする．

3. 提案手法

3.1 動きベクトルによる検出物体の追跡

第 2.1 章で概観した一般的な Tracking-by-Detection を元に，動きベクトルを用いて動画中の各フレームにおける検出回数を削減する手法を提案する．

図 3 に提案手法の概要を示す．図中上部のフレーム間の推移は，フレーム間予測による復号化処理を表す．Iフレームからは CNN 等による検出器を用いて物体検出を行い，BB 集合を生成する．P フレームでは，直前の動き補償を行うフレームの BB 集合を参照し，動画像コーデックの動き補償処理より付随情報として抽出した動きベクトルによって各バウンディングボックスを補間する．なお，本手法ではIフレームと P フレームのみをフレーム要素として持つ動画像コーデックで符号化された動画を対象とするため，P フレームが続く間はこの追跡を繰り返す．次のIフレームがきたらそれまで追跡を行い補間的に生成した BB 集合を廃棄し，新規の BB 集合を検出器によって生成する．

今回の提案手法では，Tracking-by-Detection に対応して Detect 処理は上記の様に動きベクトルを用いて簡略化するが，Predict 及び Associate 処理へは特に制限を加えない．即ち既に知られている有効な手法を組み合わせて用いることができるが，今回は知られている手法の内，最も基本的な手法を用いる．

具体的には，Predict は恒等変換としノイズの付加は行わない．また，Associate はハンガリアン法によって行うが，ハンガリアン法のコスト関数としては二つのバウンディングボックスの重なり度合いを表す IoU (Intersection over Union) による下式の関数を用いる．但し，$|BB_x|$ は BB_x の内部領域を表す．

$$f_{\mathrm{IoU}}(\mathrm{BB_A}, \mathrm{BB_B}) = 1 - \frac{|\mathrm{BB_A} \cap \mathrm{BB_B}|}{|\mathrm{BB_A} \cup \mathrm{BB_B}|} \qquad (5)$$

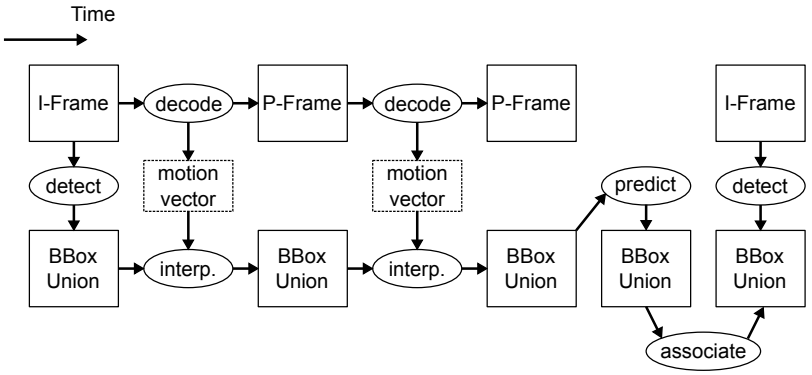

図 3 提案手法における検出フロー

3.2 検出済ボックスの補間演算

先に示した提案手法において，P フレームでのバウンディングボックスの補間演算の実現に際しては，動きベクトルを入力として様々なバリエーションが考えられる．

3.2.1 線形補間

補間演算の素朴な実現として，線形補間を取り上げる．前提として各バウンディングボックス領域内に含まれる動きベクトルの集合が水平・垂直方向に等間隔でそれぞれ N 軸，M 軸配置されるとして，ベクトル集合を $\{\boldsymbol{v}(i,j)\}_{i=1,\cdots N,j=1,\cdots M}$ と表記する，線形補間においては，バウンディングボックスのサイズを固定したままボックスの中心 $\boldsymbol{c}_t$ を新たな中心 $\boldsymbol{c}_{t+1}$ に移す以下の線形式を用いる．

$$\boldsymbol{c}_{t+1} = \boldsymbol{c}_t + \alpha \cdot \frac{\sum_i \sum_j \boldsymbol{v}(i,j)}{N \cdot M} \tag{6}$$

但し α は定数で，ユーザ設定のパラメータとする．

この際，$\alpha = 1/f$ と置くと f はバウンディングボックス内での物体の充填率として解釈することが可能である．単純な場合として，バウンディングボックス内で捕捉される物体領域の動きベクトルが定ベクトル $\boldsymbol{v}$ で，背景領域の動き成分がゼロである状況を想定する．式 (6) の動きベクトルの平均成分は f を用いて

$$\frac{\sum_i \sum_j \boldsymbol{v}(i,j)}{N \cdot M} = f \cdot \boldsymbol{v} \tag{7}$$

と表される．この場合，物体の動きを正しくボックス中心の変位に伝えるには，既に述べたように $\alpha = 1/f$ とすればよい．なお，実際には物体領域と背景領域がそれぞれ異なった動きベクトル成分を持つため，α をこのように設定しても背景領域の成分だけの誤差が含まれる．

3.2.2 勾配補間

線形補間では補間に要する演算量は非常に軽微であるものの，物体の動き成分と背景の動き成分の分離という観点では非常にナイーブな手法である．一方で，本手法の目的上これらの成分の分離に割くコストを大きくすればするほど，演算量削減の効果は薄れる．以下では比較的軽量な演算で両成分を分離する補間の実現例として，勾配情報を元にした補間（勾配補間）を示す．

勾配補間では，バウンディングボックス内の動きベクトル場の発散（divergence）を取り，その絶対値によって動きベクトルの重み付き平均を取る．発散の計算に際しては三点近似を施した式 (8), (9) の勾配を用いる．

$$\frac{\partial \boldsymbol{v}(i,j)}{\partial x} = \frac{v_x(i+1,j) - v_x(i-1,j)}{2} \tag{8}$$

$$\frac{\partial \boldsymbol{v}(i,j)}{\partial y} = \frac{v_y(i,j+1) - v_y(i,j-1)}{2} \tag{9}$$

式 (8), (9) により計算された発散の絶対値を $\{d(i,j)\}_{i=1,\ \cdots N,j=1,\cdots M}$ と置くと，ボックス中心の遷移式は

$$\boldsymbol{c}_{t+1} = \boldsymbol{c}_t + \frac{\sum_i \sum_j d(i,j) \cdot \boldsymbol{v}(i,j)}{\sum_i \sum_j d(i,j)} \tag{10}$$

と表される．

発散の絶対値は，動きベクトルの変化が大きい所謂エッジ付近の画素にて大きい値をとる．その為，発散の絶対値による重み付き平均は動きベクトルのエッジ付近を強調した平均をとることとなる．提案手法では動きベクトルの平均を取る範囲はバウンディングボックス内に限られており，対象物体を中心として画像の内の狭い範囲のみを対象としているため，ボックス内でエッジを取るような動きはおよそ対象物体の動きであると期待される．従って，勾配補間を行うことで線形補間よりも背景領域の動きを除外した平均によってボックスを遷移させられると期待される．

3.2.3 カルマンフィルタによる平滑化

最後に以上の線形補間及び勾配補間によるボックス中心の移動をカルマンフィルタによって平滑化することを考える．第 2.1 節の式 (1) 〜 (4) において状態 x_k を各バウンディングボックスの中心 c_t，外乱入力 u_k を各ボックスの移動量 $c_{t+1} - c_t$ とおくと，モデルの各行列は単位行列 I により $A = I, B = I, H = I$ と対応付けられ，全体として自明なシステムとなる．この際，式 (1) 及び式 (2) に従って，システムの状態遷移時及び BB 集合の観測時に正規分布ノイズを付加し，c_t を推定する．c_t は I フレーム毎にリセットされる．

第 4 章の評価では，本節で導入した線形補間 (Linear) 及

び勾配補間 (Gradient)，そして両者に対してカルマンフィルタの平滑化を施したバージョン (*Kalman) の計四種類の手法の評価を行う．

4. 評価実験

4.1 データセットの概要

提案手法の効果を，複数物体追跡手法のベンチマークである MOTChallenge の 2016 年版データセットである MOT16 [6] を用いて評価した．MOT16 は，入力として連番画像，FPS，総フレーム数や画面サイズ等のシーケンス全体のメタデータが与えられる他，標準検出器の DPM v5 [16] による各フレームの検出結果が与えられる．出力としては各フレームのバウンディングボックスの識別番号 (ID)，始点座標 $(x_{\mathrm{left}}, y_{\mathrm{top}})$ 及びサイズ (w, h) が求められる．データは test と train の二つのサブセットに分割され，Ground Truth は train にのみ配布されている．

MOT16 では正確な評価のため，複数の指標が評価される．共通して評価される指標は**表 1** に示す通りである．なお，括弧内の矢印はその指標が大きい方が望ましい (↑) のか，小さいほうが望ましい (↓) のかを示す．

共通指標の中で最も良く全体性能を要約する指標は MOTA [24] で，以下のように算出される．

$$\mathrm{MOTA} = 1 - \frac{\sum_t (\mathrm{FN}_t + \mathrm{FP}_t + \mathrm{IDsw}_t)}{\sum_t \mathrm{GT}_t} \qquad (11)$$

但し，GT は真値として与えられたバウンディングボックスの個数で，変数 t はフレーム番号を表す．

4.2 複数物体追跡性能の比較

4.2.1 実験設定

提案手法の追跡性能を，先に述べた指標に基づいて評価した．入力の連番画像は，FFmpeg 3.4 を用いて符号化し，FFmpeg libavcodec 57.107.100 を用いて動きベクトルを抽出した．各コーデックについて，GOP サイズを明示しない限り I フレームと P フレームの比率は I : P = 1 : 11 に固定してある．

評価対象のバリエーションとして，検出器にデータセット標準の DPMv5 を用いた場合と既存手法 [8] により提供された Faster R-CNN [2]（以下，FRCnn) の検出結果を用いた場合の 2 パターンをとった．

評価に際しての比較手法は以下の三通りである．

Baseline

Detect: 全フレーム検出

Associate: 全フレームについて，ハンガリアン法

Proposed

Detect: I フレームを検出し P フレームを補間

Associate: P→I フレームのとき，ハンガリアン法

Worst

Detect: I フレームを検出し P フレームを固定

Associate: P→I フレームのとき，ハンガリアン法

但し Proposed を参照する際には，第 3.2 節で定義した提案手法のバリエーション名 (Linear, Gradient 等) で参照する．また，線形補間を用いる際の式 (6) のパラメータは $\alpha = 1.0$ とした．

4.2.2 結果

第 3.2 節で示したバリエーション毎の性能差を評価した．この際，動画の符号化は MPEG-2 の Simple プロファイルによって行ってある．今回の評価では提案手法の処理 FPS 計測は動画の各フレーム読み出し，補間演算・ID 割り当て等の追跡にかかる演算を対象とし，検出結果の生成や動画からの動きベクトル抽出においては予め算出した結果を参照している．train split における評価結果を**表 2**，**表 3** に示す．

まずはじめに表 2，表 3 より DPMv5 及び FRCnn の手法それぞれについて，(Hz を除く）全指標で Proposed の方が Worst より改善していることが確認される．特に FP 及び FN の改善に加え IDsw が Baseline よりも大幅に減少している．これは Proposed では定義より P フレームから I フレームに切り替わる際にのみ Associate を行い，P フレームの間には同一物体を自明に追跡していることが要因として挙げられる．この特徴により今回の実験では Associate 由来のエラーが導入される頻度が 1/12 となり，IDsw の減少に繋がっている．

以上の結果では，勾配補間を行った全ての手法で MOTA が約 1.0 程度劣化しているが，定性的に追跡結果を確認すると第 3.2 節で述べた狙いが実現できている場面も確認された．**図 4** に MOT16 (MOT16-09) の追跡の一場面を示す．図中左上の I フレームから始まり，右に向かって P フレームが流れており歩行者の追跡場面を捉えている．歩行者を囲うバウンディングボックスのうち，緑色がカルマンフィルタの平滑化付き線形補間の結果で，ピンク色が同平滑化付き勾配補間の結果である．線形補間の結果ではフレームを追う毎に歩行者の動きからボックスが遅れるのに対し，勾配補間の結果ではより順当に歩行者の動きにボックスが追随しているのが見て取れる．このように勾配補間の狙いは部分的に達成できているが，一方でボックスの遮蔽や動きベクトルのノイズにより鋭敏に反応しており，全体の MOTA が低下していると考えられる．

提案手法では，I フレーム間隔を表す GOP (Group of Picture) サイズを変量として取ることができる．MPEG-2 の Simple プロファイルで GOP サイズを 1〜20 の範囲で変化させた場合の MOTA の変化を**図 5** に示す．なお，これまでの実験で用いてきた GOP サイズが 12 となる点にマーカを付加している．図 5 から見て取れるように，MOTA は GOP サイズにほぼ比例して変化している．

そのため実際のユースケースとしては，求める演算量の

表 1 MOTChallenge における共通評価指標一覧

MOTA (↑)	Multiple Object Tracking Accuracy [24].
MOTP (↑)	Multiple Object Tracking Precision. 真値と True Positive の一致度合い.
FAF (↓)	False Alarm per Frame. 画像当たりの平均誤検出数.
MT (↑)	Mostly Tracked. 軌跡の殆どを追跡できた対象物体の個数.
ML (↓)	Mostly Lost. 軌跡の殆どを追跡できなかった対象物体の個数.
FP (↓)	False Positive. 誤って検出したボックスの個数.
FN (↓)	False Negative. 検出に失敗したボックスの個数.
IDsw (↓)	Identity switch. 割り当てられる ID が切り替わった回数.
FM (↓)	Fragmentations. 軌跡の断片化が起こった回数.
Hz (↑)	1 秒間に追跡処理を行ったフレーム数.

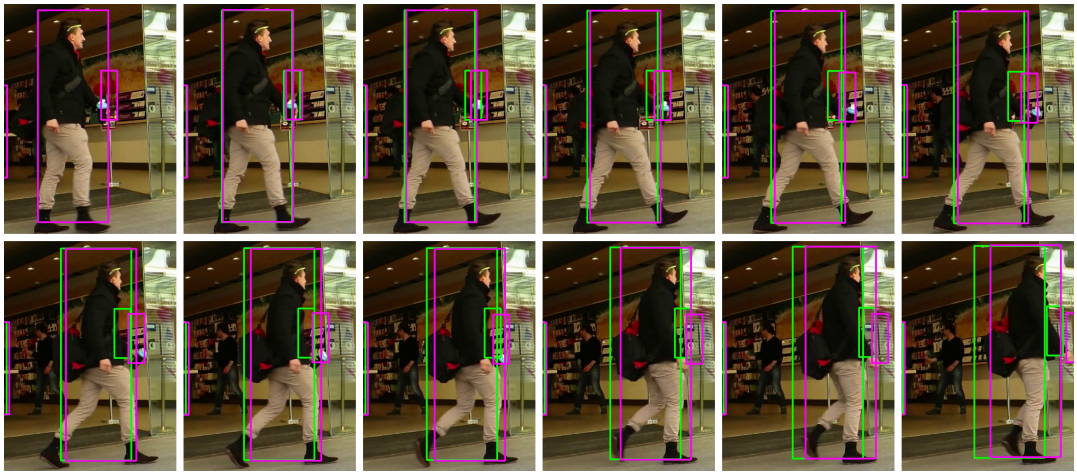

図 4 勾配補間と線形補間の比較：MOT16 における歩行者の例，緑色のボックスがカルマンフィルタの平滑化付き線形補間の結果で，ピンク色のボックスが同平滑化付き勾配補間の結果である．

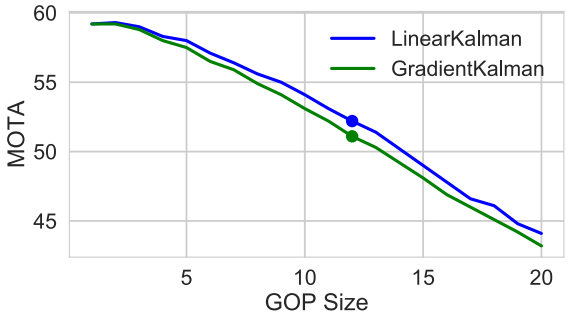

図 5 GOP サイズと MOTA の関係

削減度合いと許容できる追跡性能の低下のトレードオフによって GOP サイズは指定できる．例えば本稿のこれまでの実験の様に GOP サイズとして 12 を用いると，CNN による物体検出頻度を CNN ベースの基準手法に対し 1/12 に抑える事ができる．演算回数で比較すると CNN での演算量に比べ，提案手法で行うボックス単位の補間で行う演算量は大幅に小さいため，演算負荷全体でも物体検出頻度の削減に近い削減効果が期待出来る．

最後に，MOT16 の test split における提案手法 (FRCnn LinearKalman) の評価結果を**表 4** に示す．表 4 には比較としてデータセットで例示されている標準記録及び公式サイト [*1] に公開されている主要な手法の例を載せてある．表中，中線以下の手法がデータセットで例示されている標準記録で，これらの手法は DPMv5 の検出結果を用いている．なお，動画の符号化は MPEG-4 Part2 の Simple プロファイルによって行った．

評価結果としては，まずはじめに提案手法で文献 [8] による Faster R-CNN の検出器を用いて追跡を行うことで，データセットで例示されている DPMv5 による標準記録よりも大幅に MOTA を向上できていることが確認出来る．また，抜粋した上位の手法の内 LMP_p 以外はオンラインの追跡手法であるが，時系列の追跡において CNN を用いていない SORTwHPD16 及び EAMTT と比較すると，提案手法は MOTA で両者の中間にある．SORTwHPD16 は本稿と同じく文献 [8] による Faster R-CNN による検出器を

[*1] https://motchallenge.net/results/MOT16/

表 2 MOT16 提案手法評価結果 (DPMv5) (Evaluated on train split)

Method	MOTA	MOTP	FAF	MT (%)	ML (%)	FP	FN	IDsw	FM	Hz
DPMv5 Baseline	27.7	77.3	0.89	7.7	53.6	4707	72606	2487	2586	84.2
DPMv5 LinearKalman	25.2	75.1	1.29	4.4	60.7	6866	75224	512	720	23.4
DPMv5 GradientKalman	24.9	75.2	1.32	3.9	60.9	7029	75371	519	730	22.9
DPMv5 Linear	24.8	74.2	1.33	4.6	60.0	7050	75408	525	787	25.9
DPMv5 Gradient	24.5	74.4	1.36	4.1	60.7	7249	75594	527	783	25.2
DPMv5 Worst	18.2	73.6	1.99	1.2	65.4	10563	78947	811	1350	91.2

表 3 MOT16 提案手法評価結果 (FRCnn) (Evaluated on train split)

Method	MOTA	MOTP	FAF	MT (%)	ML (%)	FP	FN	IDsw	FM	Hz
FRCnn Baseline	59.3	82.0	1.05	36.9	14.9	5597	37296	2097	1890	55.3
FRCnn LinearKalman	52.2	78.6	1.92	22.1	23.0	10207	41884	728	1298	38.4
FRCnn GradientKalman	51.1	78.6	2.02	19.3	22.6	10748	42404	829	1383	37.5
FRCnn Linear	51.0	77.9	2.04	20.9	23.2	10846	42523	729	1419	41.6
FRCnn Gradient	50.2	77.9	2.12	19.1	23.2	11273	42913	795	1482	40.7
FRCnn Worst	34.1	76.7	3.71	7.2	27.1	19746	51109	1908	2860	58.6

用いており，毎フレーム検出を行っているためおよそ以上で Baseline として参照している手法に対応する．EAMTT は検出器に DT-DPM [11] 等を用い粒子フィルタ等で追跡を行っているが，検出器の効果により MOTA で比較した場合に提案手法の方が 2.5 程度高い数値が得られている．

5. まとめ

本稿では CNN によるリアルタイム物体検出を効率良く実現するための手法として，動きベクトルの平均を用いて I フレームの物体検出結果をカルマンフィルタと同等の線形演算によって追跡する手法を提案した．複数物体追跡データセット MOT16 による定量的評価により，I フレーム間隔が 12 の場合，提案手法を用いることで CNN による物体検出頻度を 1/12 に抑え，Faster R-CNN を用いた基準手法比で 84.7%〜88.0%の MOTA を維持可能であることを確認した．

参考文献

[1] Simonyan, K. and Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition, *Computing Research Repository*, Vol. abs/1409.1556 (2014).
[2] Ren, S., He, K., Girshick, R. and Sun, J.: Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, *Proceedings of the Neural Information Processing Systems* (2015).
[3] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y. and Berg, A. C.: SSD: Single Shot MultiBox Detector, *Proceedings of the European Conference on Computer Vision* (2016).
[4] Redmon, J., Divvala, S. K., Girshick, R. B. and Farhadi, A.: You Only Look Once: Unified, Real-Time Object Detection, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2016).
[5] Redmon, J. and Farhadi, A.: YOLO9000: Better, Faster, Stronger, *Computing Research Repository*, Vol. abs/1612.08242 (2016).
[6] Milan, A., Leal-Taixé, L., Reid, I. D., Roth, S. and Schindler, K.: MOT16: A Benchmark for Multi-Object Tracking, *Computing Research Repository*, Vol. abs/1603.00831 (2016).
[7] Tang, S., Andriluka, M., Andres, B. and Schiele, B.: Multiple People Tracking by Lifted Multicut and Person Re-identification, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2017).
[8] Yu, F., Li, W., Li, Q., Liu, Y., Shi, X. and Yan, J.: POI: Multiple Object Tracking with High Performance Detection and Appearance Feature, *Proceedings of the European Conference on Computer Vision Workshop* (2016).
[9] Bewley, A., Ge, Z., Ott, L., Ramos, F. and Upcroft, B.: Simple Online and Realtime Tracking, *Proceedings of the IEEE International Conference on Image Processing* (2016).
[10] Sanchez-Matilla, R., Poiesi, F. and Cavallaro, A.: Online Multi-target Tracking with Strong and Weak Detections, *Proceedings of the European Conference on Computer Vision Workshop* (2016).
[11] Felzenszwalb, P. F., Girshick, R. B., McAllester, D. and Ramanan, D.: Object Detection with Discriminatively Trained Part-Based Models, *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2010).
[12] Kantorov, V. and Laptev, I.: Efficient Feature Extraction, Encoding and Classification for Action Recognition, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2014).
[13] Zhang, B., Wang, L., Wang, Z., Qiao, Y. and Wang, H.: Real-Time Action Recognition With Enhanced Motion Vector CNNs, *Proceedings of the IEEE Computer*

表 4 MOT16 提案手法及び主要既存手法評価結果 (Evaluated on test split)

Method	MOTA	MOTP	FAF	MT (%)	ML (%)	FP	FN	IDsw	FM	Hz
LMP_p [7]	71.0	80.2	1.3	46.9	21.9	7880	44564	434	587	0.5
POI [8]	66.1	79.5	0.9	34.0	20.8	5061	55914	805	3093	9.9
SORTwHPD16 [9]	59.8	79.6	1.5	25.4	22.7	8698	63245	1423	1835	59.5
Proposed (FRCnn LinearKalman)	55.0	76.7	2.7	20.4	24.5	15766	65297	1024	1594	16.9
EAMTT [10]	52.5	78.8	0.7	19.0	34.9	4407	81223	910	1321	12.2
TBD [14]	33.7	76.5	1.0	7.2	54.2	5804	112587	2418	2252	1.3
CEM [25]	33.2	75.8	1.2	7.8	54.4	6837	114322	642	731	0.3
DP_NMS [26]	32.2	76.4	0.2	5.4	62.1	1123	121579	972	944	212.6
SMOT [27]	29.7	75.2	2.9	4.3	47.7	17426	107552	3108	4483	0.2
JPDA_M [28]	26.2	76.3	0.6	4.1	67.5	3689	130549	365	638	22.2

Society Conference on Computer Vision and Pattern Recognition (2016).

[14] Geiger, A., Lauer, M., Wojek, C., Stiller, C. and Urtasun, R.: 3D Traffic Scene Understanding from Movable Platforms, *the IEEE Transactions on Pattern Analysis and Machine Intelligence* (2014).

[15] Zhang, H., Geiger, A. and Urtasun, R.: Understanding High-Level Semantics by Modeling Traffic Patterns, *Proceedings of the IEEE International Conference on Computer Vision* (2013).

[16] Sadeghi, M. A. and Forsyth, D.: 30Hz Object Detection with DPM V5, *Proceedings of the European Conference on Computer Vision* (2014).

[17] Girshick, R., Donahue, J., Darrell, T. and Malik, J.: Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2014).

[18] Girshick, R.: Fast R-CNN, *Proceedings of the IEEE International Conference on Computer Vision* (2015).

[19] Welch, G. and Bishop, G.: An Introduction to the Kalman Filter, Technical report, Chapel Hill, NC, USA (1995).

[20] Kuhn, H. W. and Yaw, B.: The Hungarian Method for The Assignment Problem, *Naval Res. Logist. Quart*, pp. 83–97 (1955).

[21] ISO/IEC 14496-2:2004: Information technology – Coding of audio-visual objects – Part 2: Visual, Standard (2004).

[22] ISO/IEC 13818-2:2013: Information technology – Generic coding of moving pictures and associated audio information – Part 2: Video, Standard (2013).

[23] ISO/IEC 14496-10:2014: Information technology – Coding of audio-visual objects – Part 10: Advanced Video Coding, Standard (2014).

[24] Bernardin, K. and Stiefelhagen, R.: Evaluating Multiple Object Tracking Performance: The CLEAR MOT Metrics, *EURASIP Journal on Image and Video Processing*, Vol. 2008, No. 1 (2008).

[25] Milan, A., Roth, S. and Schindler, K.: Continuous Energy Minimization for Multitarget Tracking, *the IEEE Transactions on Pattern Analysis and Machine Intelligence* (2014).

[26] Pirsiavash, H., Ramanan, D. and Fowlkes, C. C.: Globally-Optimal Greedy Algorithms for Tracking a Variable Number of Objects, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2011).

[27] Dicle, C., Camps, O. I. and Sznaier, M.: The Way They Move: Tracking Multiple Targets with Similar Appearance, *Proceedings of the IEEE International Conference on Computer Vision* (2013).

[28] Fortmann, T. E., Bar-Shalom, Y. and Scheffe, M.: Multi-target Tracking Using Joint Probabilistic Data Association, *1980 19th IEEE Conference on Decision and Control including the Symposium on Adaptive Processes* (1980).