CH-006

符号化パラメータを用いた動体領域検出手法による圧縮動画検索

Compressed video search: detection of moving object area using coding parameters

祖泉 大河 * 森田 啓義 * 眞田 亜紀子 †
Taiga Soizumi Hiroyoshi Morita Akiko Manada

1 はじめに

デジタルビデオカメラ関連技術の著しい進化によって HD 映像などの高解像度の映像が身近となっている．このような高解像度の映像はフレーム内に小さく写っている人などにおいても動作の解析を可能とする．例えば監視カメラにおいては，カメラが固定されているためフレームいっぱいに人物が映るとは限らず，大抵の場合フレーム内の一部に映り込む形になる．低解像度の映像ではこのような人物の動作解析は困難だが，高解像度の映像はこのような場合においてもその人物についての動作を解析し，監視カメラ映像から特定のシーンを検索することを可能とする．しかし，このように高画質な映像ではその情報量ゆえに，ユーザが見たいシーンを高速に検索する技術が必要である．

2 関連研究

動作の検索では，Alawassel ら [1] が Long Short Term Memory (LSTM) ネットワークを用いて，人間が動画から動作を検索する過程を学習させて動作検索を行う手法を提案している．また，論文 [2] では野球における投球動作との関連性が高い突発音を用いて動作のテンプレート画像を作成し，動画中でマッチングを行うことで，投球動作を検出する研究が行われている．

Derphins ら [3] は歩く，手を振るといった動作を含むビデオクリップ (短い動画) を検索キーとする手法を提案している．キーワードでは動作の表現に限界があるが，ビデオクリップであればどのような動作においても表現をすることが可能であるため，細かい動作の検索を行うことが可能となる．しかし，この研究ではフレームサイズが 50 × 25 でフレームレートが 10fps の検索キー動画を用いて，画面サイズ 144 × 180 でフレームレートが 10fps の検索対象動画からリアルタイムな動作の検索を実現している．しかし，解像度が非常に低く近年普及している HD などの解像度の動画の解析には向かない．

高速な動画検索については広く普及している圧縮規格である MPEG-2 や H.264 で圧縮された動画に対して，圧縮動画から得られる符号化パラメータを直接利用することで復号化をせずに高速化を目指す手法がある [4][5]．論文 [4] では検索キーと検索対象の動画をともに H.264 圧縮動画とし，それぞれから動体領域において発生しやすい符号化パラメータを用いて動体領域の重心曲線を作成し，重心曲線同士のマッチングを行うことで，動作の検索を実現した．しかし，手を振る動作と手を叩く動作において検出の再現率・適合率が低下し区別しづらいという課題がある．

本研究では，符号化パラメータのみを用いた動画を検索キーとする動画検索において，新たな動体領域の検出手法 [5] を導入し従来より高精度で動画検索をできるシステムを提案する．

3 提案手法

昨年 FIT2018（第 17 回情報科学技術フォーラム）にて，MPEG-2 圧縮動画での動体領域には符号化パラメータの 1 つであるマクロブロックタイプ（MBT）の発生パターン（時空間 MBT パターン）があることを発表した [5]．フレームサイズ 720 × 480 の動画では時空間 MBT パターンの発生は動体領域のごく一部であったため，従来動体領域の検出に用いられていたマクロブロックサイズの情報を加えて動体領域の検出を行なっていた．

本研究では動画のフレームサイズを大きくすることで時空間 MBT パターンがより精度良く発生することと H.264 圧縮動画でも発生することを確認し，論文 [4] と同様にその重心情報を用いることでより高精度な動画検索を実現する．

3.1 時空間 MBT パターンによる動体領域の検出

MPEG-2 や H.264 では，動画を圧縮する際にフレーム間予測の技術を用いており，画素の塊であるマクロブロッ

*電気通信大学大学院情報理工学研究科
†湘南工科大学工学部情報工学科

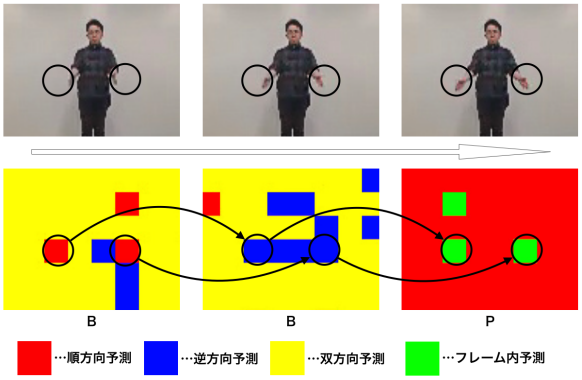

図 1: 時空間 MBT パターン

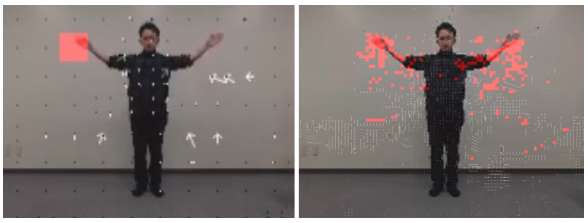

図 2: 時空間 MBT パターン検出例（手を振る動作）

クごとにどのフレームから予測を行うかが決定される．選択された予測方式が圧縮動画データにマクロブロックタイプ（MBT）として記録されている．MPEG-2 ではフレームごとに動体領域で発生しやすい MBT があり，手を叩く様子を撮影した図 1 のようなフレームの並びにおいては 2 枚連続する B フレームの先の B フレームが順方向予測，後の B フレームが逆方向予測となる傾向があり，P フレームではフレーム内予測が発生しやすい [5]．したがってピクチャタイプが BBP という並びの 3 枚のフレームにおいて，「順方向予測→逆方向予測→フレーム内予測」というパターンを示しているマクロブロックを検出することによって，動体の領域の検出が可能となる．このパターンを時空間 MBT パターンと呼ぶ．

図 2 に手を振る動作を撮影した H.264 圧縮動画での時空間 MBT パターンの検出例を示す．赤くオーバーレイされている部分が時空間 MBT パターンが発生したマクロブロックである．図の左は低解像度（160 × 120）の場合，右は高解像度（1440 × 1080）の場合である．H.264 動画においても，手を振る動作の腕の部分等の動体領域に時空間 MBT パターンが発生していることがわかる．低解像度の例においては片腕に時空間 MBT パターンが発生しておらず，低解像度では時空間 MBT パターンが発生しづらく，高解像度での検出が適していると言える．

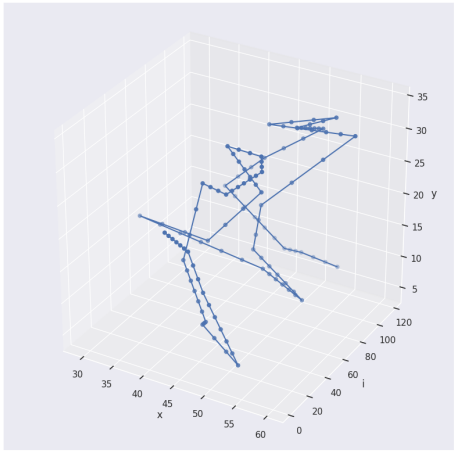

図 3: 重心曲線の例（手を振る動作）

3.2 重心曲線によるマッチング

本研究では論文 [4] と同様に動体領域の重心曲線（図 3）を求め，検索キー動画と検索対象動画でマッチングすることによって動作を検索する．

検索キー動画の時系列重心座標データと検索対象動画の時系列重心座標データ同士で区間ごとに両データ間の距離の和を求める．重心間距離の和が小さいということは，2 つの重心曲線が似ていることを意味する．したがって，重心間距離の和を求め，その値があらかじめ設定した閾値を下回っているフレームについては検索キー動画の動作が行われていると判定する．検索キー動画を f，検索対象動画を g，検索キー動画のフレーム数を n，検索対象動画のフレーム数を N，検索キー動画の時系列重心座標データスライドさせる数を i とすると，検索キー動画と検索対象動画の重心間距離の和の関数は以下のようになる．

$$F(i) = \sum_{x=0}^{n-1} f(x)g(x+i) \quad (i = 0, 1, ..., N-n+1) \quad (1)$$

ただし，$f(x)g(x+i)$ は検索キー動画の x 番目の重心と検索対象動画の $x+i$ 番目の重心間の距離を表す．

検索対象動画とボクシング動作動画の重心間距離の和を図 4 に示す．この例では検索対象動画の $1258 \leq i \leq 1498$ と $2251 \leq i \leq 2512$ の区間でボクシング動作が行われており，その区間で重心間距離の和が低下していることがわかる．

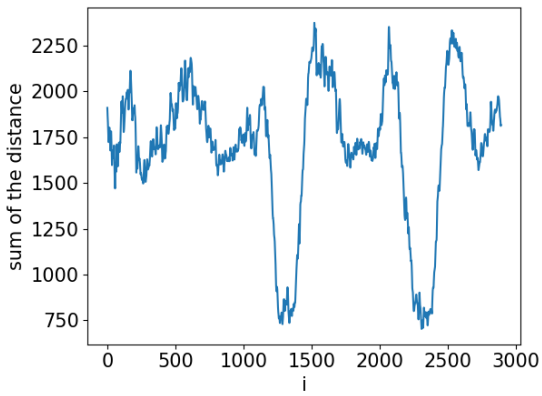

図 4: 重心間距離の和（ボクシング動作）

4　動作の検索実験

4.1　使用動画

論文 [4] で撮影され用いられた動画を使用した．検索キー動画はボクシング，手を叩く，手を振るの 3 種類をそれぞれ約 5 秒間撮影したものである．各動画ともフレームレートは 25fps であるフレームサイズについては 3.1 での考察を踏まえて 160 × 120 であったものを FFmpeg[6] を用いて 1440 × 1080 に変換した．検索対象動画は検索キー動画と同じ 3 種類の動作に加え，それぞれの動作の間に足踏み動作を加えたものである．各動作 10 秒程度で足踏み→手を振る→足踏み→手を叩く→足踏み→ボクシング→足踏み→手を叩く→足踏み→ボクシング→足踏み→手を振るの順番で撮影され，全体としては合計約 120 秒の動画である．

4.2　検索システム

提案手法における，符号化パラメータの用いた時空間 MBT パターンの検出および各フレームにおけるその重心座標の計算を C 言語にて，得られた重心座標の補完および重心曲線同士の距離計算と最終的な動作範囲の判定を Python で実装した．実装においては FFmpeg[6], OpenCV[7] のライブラリを用いた．また，これらのプログラムは次のノート PC 上にて動作させた．

- MacBook Pro(Retina, 15-inch, Mid 2015)
- macOS Mojave バージョン 10.14.5
- プロセッサ　2.5 GHz Intel Core i7
- メモリ　16GB 1600 MHz DDR3

4.3　実験結果

あらかじめ検索対象動画において各検索動作が行われているフレームを目視で判別しておき，閾値を 10 ずつ変化させながら検索システムによって動作発生フレームと判定されたフレームが実際に動作が発生しているフレームであるか照合し，正検出（TP）・誤検出（FP）・未検出（FN）のフレーム数を数え上げた．またこれらをもとにして再現率（Recall）と適合率（Precision）を算出した．これらの結果を閾値をごとにまとめ表 1，2，3 に記した．

$$Recall = \frac{TP}{TP+FN}, Precision = \frac{TP}{TP+FP} \quad (2)$$

表 1: ボクシング動作の検出結果

閾値	正例	TP	FP	FN	Recall	Precision
1410	503	375	4	128	74.6%	98.9%
1420	503	375	4	128	74.6%	98.9%
1430	503	375	4	128	74.6%	98.9%
1440	503	390	4	113	77.5%	99.0%
1450	503	390	4	113	77.5%	99.0%
1460	503	434	4	69	86.3%	99.1%
1470	503	478	1146	25	95.0%	29.4%
1480	503	478	1146	25	95.0%	29.4%
1490	503	478	1146	25	95.0%	29.4%
1500	503	478	1146	25	95.0%	29.4%

表 2: 手を叩く動作の検出結果

閾値	正例	TP	FP	FN	Recall	Precision
910	518	375	40	143	72.4%	90.4%
920	518	389	40	129	75.1%	90.7%
930	518	389	40	129	75.1%	90.7%
940	518	411	40	107	79.3%	91.1%
950	518	411	40	107	79.3%	91.1%
960	518	432	92	86	83.4%	82.4%
970	518	460	540	58	88.8%	46.0%
980	518	460	540	58	88.8%	46.0%
990	518	460	540	58	88.8%	46.0%
1000	518	486	540	32	93.8%	47.4%

4.4　先行研究との比較

提案手法による実験結果から，再現率と適合率の調和平均である F 値（F-measure）を基準にして動作ごとのベ

表 3: 手を振る動作の検出結果

閾値	正例	TP	FP	FN	Recall	Precision
1240	497	345	0	152	69.4%	100.0%
1250	497	377	16	120	75.9%	95.9%
1260	497	377	16	120	75.9%	95.9%
1270	497	377	16	120	75.9%	95.9%
1280	497	377	16	120	75.9%	95.9%
1290	497	377	16	120	75.9%	95.9%
1300	497	377	16	120	75.9%	95.9%
1310	497	377	16	120	75.9%	95.9%
1320	497	377	16	120	75.9%	95.9%
1330	497	394	445	103	79.3%	47.0%

ストスコアを選び表 4 にした．また，論文 [4] での実験結果においても同様にしてベストスコアを選び表 5 にした．

$$F - measure = \frac{2 * Recall * Precision}{Recall + Precision} \tag{3}$$

表 4: 提案手法ベストスコア

動作	Precision	Recall	F-measure
ボクシング	89.3%	99.1%	92.3%
手を叩く	79.3%	91.1%	84.8%
手を振る	75.9%	95.9%	84.7%

表 5: 論文 [4] ベストスコア

動作	Precision	Recall	F-measure
ボクシング	91.1%	86.7%	88.8%
手を叩く	66.2%	63.5%	64.8%
手を振る	33.7%	73.7%	46.3%

論文 [4] と比較して 3 動作すべてについて F 値が上回りボクシング動作においては F 値が 3.5%，手を叩く動作では 20.0%，手を振る動作では 38.4%上回る結果となった．

4.5 処理時間

今回の実験での要した処理時間について記す．圧縮動画から符号化パラメータを抽出して時空間 MBT パターンを検出し，各フレームにおける重心を計算するのに要した時間はボクシング動作動画が 0.04 秒，手を叩く動作動画が 0.04 秒，手を振る動作動画が 0.05 秒，検索対象動画では 1.10 秒であった．重心曲線同士のマッチングによって動作を検索する過程において，閾値ごとの平均処理時間はボクシング動作が 1.1 秒，手を叩く動作が 1.6 秒，手を振る動作が 1.3 秒であった．

5 まとめ

本研究では動画のフレームサイズを大きくすることで時空間 MBT パターンがより精度良く発生することと H.264 圧縮動画でも発生することを確認し，その重心曲線のマッチングを行うことで従来より高精度な動画検索を実現した．提案手法では手を叩く動作と手を振る動作という区別の難しい動作においても，F 値において 80%台での検索を実現することができた．使用動画やマッチングの手法は論文 [4] と同じものであるから，今回の結果は H.264 圧縮動画での符号化パラメータを用いた動体領域検出において，時空間 MBT パターンが有用であることを示している．

本研究では動作の時間的な検出について行なったが，空間上での検出については至っておらず，人物は 1 人のみでの検出である．今後についてはこれらの課題の克服について検討していきたい．

参考文献

[1] H. Alwassel *et al.*,“Action Search: Spotting Actions in Videos and Its Application to Temporal Action Localization ”, In: ECCV, 2018.

[2] 三上弾, 紺谷精一, 森本正志,“突発音検出と教師なし動きクラスタリングを用いた野球映像からの投球イベント検出 ”, 電子情報通信学会論文誌 D Vol. J90–D No. 2 pp. 526–534, pp.21-26, 2007.

[3] K. G. Derpanis *et al.*,“Action spotting and recognition based on a spatiotemporal orientation analysis”, Trans. on PAMI, vol. 35, pp. 527-540, 2013.

[4] 岡村和磨, 森田啓義, 眞田亜紀子,“人の動作ビデオクリップを検索キーとする圧縮動画検索 ”, 2018 年 電子情報通信学会総合大会情報・システム講演論文集 2, D-12-59, pp. 98, 2018.

[5] 祖泉大河, 森田啓義, 眞田亜紀子,“時空間マクロブロックタイプパターンを用いた圧縮動画検索”, FIT2018(第 17 回情報科学技術フォーラム) 第 3 分冊, CH-002, pp. 5-8, 2018.

[6] FFmpeg, htts://www.ffmpeg.org

[7] OpenCV, htt://opencv.jp/