

フレーム間差分と検出データの遷移に基づく最適動画選択技術

山口俊輔[†] 笠波昌昭[†] 小牧大治郎[†] 篠原昌子[†] 堀尾健一[†] 村上雅彦[†]

概要: 動画や音声などのメディアデータを取り扱うメディア処理（以降フィルタ）を組み合わせることで、アプリケーションを簡単に開発・実行するフレームワークが提案されている。これらのフレームワークでは、フィルタ数の増加に伴って多様なアプリケーションを開発できる反面、多数のフィルタの機能を把握する必要があるため、アプリケーション開発者に大きな負担となる。そこで我々は、動画を解析して何らかの検出データを出力するフィルタを対象に、フィルタの機能把握に寄与可能な動画を選択する技術を提案する。この技術では、予め用意した動画群に対して動画のフレーム間差分と、フィルタを通して動画から出力される検出データの遷移に基づきスコアを算出し、スコアの高い動画を選択する。本稿では、提案技術の設計及び開発と、有効性について評価、考察した結果を報告する。

キーワード: 検出フィルタ, フレーム間差分, 動画, メディア処理基盤

Optimal Video Selection Technology Based on Transition of Frame Difference and Detection Data

Shunsuke Yamaguchi[†] Masaaki Kasanami[†] Daijiro Komaki[†]
Masako Shinohara[†] Kenichi Horio[†] Masahiko Murakami[†]

1. はじめに

ネットワークの普及や通信速度の高速化に伴い、大容量の動画や音声などのメディアデータを取り扱うネットワークアプリケーションの注目が顕著となっている。これまで、ネットワークやセンサデータやメディアを取り扱うアプリケーションを開発、実行するには、それぞれの高い専門知識と多大な工数が必要であったが、近年、アプリケーション開発画面上でメディア処理（以降フィルタ）を組み合わせるだけでアプリケーションを開発、実行可能なフレームワークが提供されており、簡単かつ迅速なメディアアプリケーションの開発、実行が可能となっている。

これらのフレームワークは、様々なメディアの処理に対応する多くのフィルタを備えており、多様なアプリケーション開発に活用できる。しかし活用するにあたっては、以下のいずれかの手段を用いて各フィルタの機能を把握する必要があるため、フィルタ数の増加に伴って、アプリケーション開発者の大きな負担となる。

- フィルタの名称から機能を判断
- 実際にフィルタを組み込んだアプリケーションを試作し、結果を見て機能を判断

特に動画を解析して、動きや色、特定の物体の有無などの特徴を検出したデータを出力するフィルタ（以降検出フィルタ）の場合、動画だけではなく出力される検出データについても理解する必要がある、機能把握による負担が顕著となっている。我々は、検出フィルタの機能を動画で把

握可能とすることで、フレームワークを利用するアプリケーション開発者が検出フィルタ機能を迅速に把握できると考えた。しかし検出フィルタ開発者が、検出フィルタの機能を把握可能な動画を作成する必要が発生し、検出フィルタ開発者に大きな負担となる。

そこで我々は、検出フィルタを対象に、検出フィルタの機能把握に寄与可能な動画を選択する技術を提案する。この技術では、予め用意した動画群に対して、動画のフレーム間差分と、検出フィルタを通して動画から出力される検出データの遷移に基づきスコアを算出し、スコアの高い動画を選択する。提案技術により、アプリケーション開発者は、検出フィルタの機能把握に最適な動画を見て、検出フィルタの機能を直観的に把握できる。また、検出フィルタ開発者が、機能把握に寄与可能な動画を作成する必要がなくなる。

以降、2章では関連研究について紹介する。3章では提案技術とそのシステムアーキテクチャについて説明する。4章ではシステムの実装と、実装から判明した課題、課題の改善案とその実装について説明する。5章では実装したシステムを用いて、動画がフィルタの機能把握に寄与可能かと、提案技術で選択される動画が適切かについて実験を行い、その有効性を評価した。7章で本稿のまとめと今後の課題を述べる。

2. 関連研究

動画検索技術として、キーワード検索をはじめ、類似検索やシーン検索など様々な検索手法について研究、開発が行われてきた。

[†] 株式会社富士通研究所
FUJITSU LABORATORIES LTD.

Tang ら[1]は、動画を分割し、物体の特定の動き、状態の遷移、および状態の期間（その状態がどれだけ続いたか）の遷移を学習させることで、陸上の棒高跳び、バスケットボールのレイアップシュート、テニスのサーブなどスポーツの特定シーンや、誕生日パーティ、クリスマスなどの催し物、車のタイヤ清掃、サンドウィッチの調理などの様々な生活の動作を抽出する技術を提案している。また内海ら[2]は、動画像から物体の特徴量と場所に関する特徴量を抽出し、これらの特徴量からイベントが動画に含まれる確率を計算することで、イベントを検出する技術を提案している。これらの技術では、動画像の特定の動きや動画像から得られる情報を用いてイベントを検出できる。

木村ら[3]は、カメラワーク、映像中の人物数、音の種類、類似尺度を元に入力した動画と類似する動画を検索する技術を提案している。この技術では、入力した動画に対してカメラワークや人物数が似た動画は検索できる。

しかしこれまでの動画検索技術に動画群から人間が見て検出フィルタの機能を把握できる動画を選択、決定するのはない。

3. 提案技術

3.1 最適な動画の定義

アプリケーション開発者が検出フィルタの機能を把握するための最適な動画とは何かを検討し、以下を必要条件として抽出した。

1. 検出フィルタが検出データを出力する動画
 検出データを出力しない動画では、検出フィルタの機能を把握できないため。
2. 出力された検出データに変化がある動画
 検出データに変化が無い動画（例えば、顔検出フィルタに対して常に正面に顔を向けている動画）では、動画の何に反応しているかわからないため。
3. 検出データの出力に同期してある程度の動きが発生する動画
 動画中の動きの前後から、検出データの出力が動画の何に反応しているかが見てわかりやすいため。

1～3 から、人間が見て検出フィルタの機能を把握できる最適な動画を以下の通り定義した。

【定義】

- ① 検出フィルタが検出データを1回以上出力し、検出データの出力タイミングと一定領域以上のフレーム間差分の発生のタイミングが最も同期している（図1）。

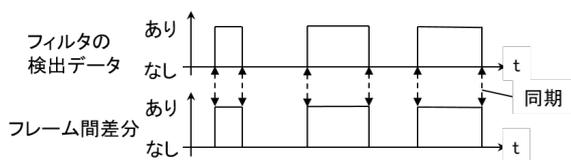


図1 検出データとフレーム間差分の同期

Figure 1 Data synchronization.

3.2 アーキテクチャ

提案技術を実現するためのシステムのアーキテクチャを図2に示す。

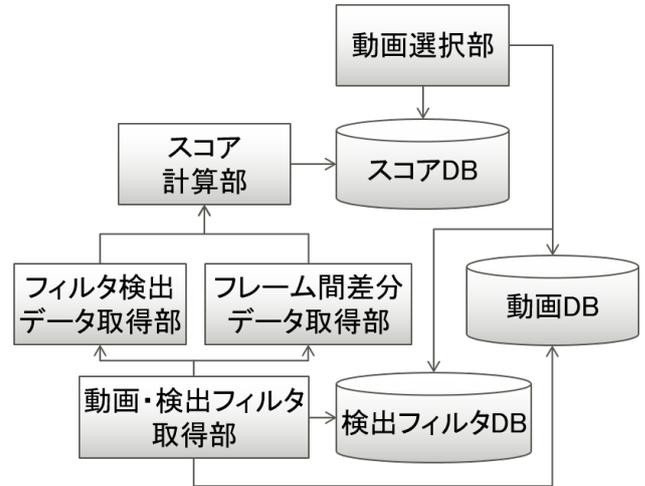


図2 システムのアーキテクチャ

Figure 2 System architecture.

提案システムは、動画と検出フィルタを取得する動画・検出フィルタ取得部、動画のフレーム間差分データを取得するフレーム間差分データ取得部と、動画を検出フィルタに適用して出力される検出データを取得するフィルタ検出データ取得部、これら取得したフレーム間差分データと検出データの遷移を比較し、スコアを算出するスコア計算部、検出フィルタに最適な動画を選択する動画選択部をもつ。また、提示候補となる動画を保存するDB、機能を把握する必要がある検出フィルタを登録している検出フィルタDB、動画と検出フィルタと、この組に対して算出されたスコアを保存するスコアDBをもつ。

提案システムでは、予め用意した動画群に対して、動画のフレーム間差分と、検出フィルタを通して動画から出力される検出データの遷移に基づきスコアを算出しておく。また、アプリケーション開発者が検出フィルタを選択した場合、選択した検出フィルタに対して最も高いスコアをもつ動画を、機能把握に寄与可能な動画として提示する。

以下、各機能部の詳細について説明する。

3.2.1 動画・検出フィルタ取得部

動画・検出フィルタ取得部では、動画DBから動画を、検出フィルタDBから検出フィルタをそれぞれ取得する。次に、動画と検出フィルタの全ての組合せに対して、動画をフレーム間差分データ取得部、動画と検出フィルタをフィルタ検出データ取得部へそれぞれ渡す。

3.2.2 フレーム間差分データ取得部

フレーム間差分データ取得部では、動画・検出フィルタ取得部から受け取った動画に対して、フレーム毎に前のフレームとの背景差分を取得し、フレーム間差分データとしてスコア計算部へ渡す。

3.2.3 フィルタ検出データ取得部

フィルタ検出データ取得部では、動画・検出フィルタ取得部から受け取った動画を、同様に受け取った検出フィルタに適用する。検出フィルタからフレーム毎の検出データを取得し、その遷移をスコア計算部へ渡す。

3.2.4 スコア計算部

スコア計算部では、フレーム間差分データ取得部から受け取ったフレーム間差分データと、フィルタ検出データ取得部から受け取った検出データの遷移を比較し、同時間における一致度を計算してスコアの算出する。算出されたスコアは、動画、検出フィルタの情報とともにスコア DB へ保存する。

3.2.5 動画選択部

動画選択部では、検出フィルタに対する動画とスコアの組合せをスコア DB から取得し、最もスコアの高い動画を最適な動画として選択する。

4. 実装と改善

3 章に基づいてシステムのプロトタイプを実装した。なおプロトタイプでは、アプリケーション開発フレームワークのメディア処理基盤[4]を一部で利用した。以下ではまず、実装したプロトタイプで利用しているメディア処理基盤について説明し、次に実装したプロトタイプの詳細について説明する。さらに実装によって判明した課題と、課題に対する改善策、改善策の実装を説明する。

4.1 メディア処理基盤

メディア処理基盤は、メディアの入出力を実現するフィルタやメディア処理を行うフィルタがアイコンで表現され、ブラウザ上の GUI でフィルタを組み合わせることで、メディアデータを取り扱うアプリケーションを簡単に定義、実行可能とするフレームワークである。またメディア処理基盤は、様々な API を提供しており、GUI を利用しなくても、JSON 形式で記述したアプリケーション定義の登録、アプリケーションの実行、アプリケーションで発生する検出データの受信などを API 呼び出しで実現できる。

例えば図 3 のアプリケーション定義では、「再生フィルタ」により指定のファイルパスや URI にあるメディアを取得し、それを「背景差分フィルタ」と「人数カウントフィルタ」に適用するアプリケーションを定義している。

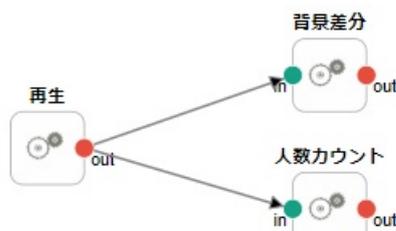


図 3 アプリケーションの定義

Figure 3 Application definition.

4.2 実装の詳細

4.2.1 動画・検出フィルタ取得部

プロトタイプでは、予め収集しておいた動画群を、メディア処理基盤が動作するサーバ上のディレクトリに保存しておき、ファイルシステム経由で該当ディレクトリにある動画一覧のファイルパスを取得する。また、メディア処理基盤が提供するフィルター一覧取得 API を用いて、メディア処理基盤で利用可能な検出フィルタの一覧を取得する。

次に動画と検出フィルタの全ての組合せに対して、動画を指定のファイルパスから取得し、4.2.2 節の背景差分検出フィルタと指定の検出フィルタに適用するアプリケーションをメディア処理基盤の JSON 形式で定義する(図 3 の「人数カウント」が指定の検出フィルタになったもの)。さらにメディア処理基盤の API を用いて、アプリケーション定義を登録し、アプリケーションとして実行する。

4.2.2 フレーム間差分データ取得部

予め、メディア処理基盤で扱える背景差分検出フィルタを実装した。背景差分検出フィルタは、入力した動画に対して、フレーム毎に前のフレームとの差分の領域面積を計算する。領域面積が閾値以上か否かを差分の有無とし、時刻、領域面積、およびフレーム間差分の有無を出力する。

フレーム間差分データ取得部では、4.2.1 節で実行したアプリケーションに対して、背景差分検出フィルタから発生した検出データを受信し、時刻とフレーム間差分の有無を抽出して、フレーム間差分データとしてスコア計算部へ渡す。

4.2.3 フィルタ検出データ取得部

フィルタ検出データ取得部では、4.2.1 節で実行したアプリケーションに対して、指定の検出フィルタから発生した検出データを受信し、時刻と検出フィルタ毎に必要な情報を抽出して、検出データとしてスコア計算部へ渡す。なお、検出フィルタ毎に必要な情報は、例えば、顔検出フィルタであれば顔の検出の有無、人数検出フィルタであれば検出人数などである。

4.2.4 スコア計算部

3.1 節の定義①に基づき、動画の全てのフレームに対して、フレーム間差分データと検出データを比較し、互いに 1 で一致する場合スコアを加算する。具体的にはフレーム i の

- ・フレーム間差分データの値を F_i
- ・フィルタの検出データの値を D_i

とすると、スコア V は

$$V = \sum_{i=1}^N v_i$$

$$v_i = \begin{cases} 1 & (D_i = 1 \text{ かつ } F_i = 1) \\ 0 & (D_i \neq 1 \text{ または } F_i \neq 1) \end{cases} \quad (1)$$

となる。

4.3 プロトタイプの改善

4.3.1 実装で判明した課題

サンプルとして収集した 10 本の動画と顔検出フィルタを用いて確認した結果、想定と異なる動画のスコアが高くなる 3 つのパターンが判明した。

1. 動きが激しく検出データが頻繁に変化する動画
 フレーム間差分データと検出データ双方が非常に多く、人間が見てわかりにくい。双方が完全に同期しているよりも、検出データが変化するタイミングでのみ動きがある方が、メリハリがあってわかりやすいこともわかった。
2. 検出フィルタが検出データを出力している時に、検出対象以外が動く動画
 検出フィルタの対象と関係ない動きであってもフレーム間差分データが発生し、検出データが 1 であれば、フレーム間差分データと検出データが一致してしまうため。
3. 時間が長い動画
 フレーム間差分データと検出データが一致する機会が増え、相対的にスコアが高くなりやすいため。

4.3.2 定義の改善

4.3.1 節の結果から、検出フィルタの機能を把握できる最適な動画の条件を見直し、以下②～④へ定義を改善した(図 4)。

【改善定義】

- ② 検出データの出力タイミングに対して、フレーム間差分データが同期して発生した後、一定期間フレーム間差分が発生しない。
- ③ 検出フィルタが出力しない間に、フレーム間差分データが発生しない。
- ④ 動画時間の長さが 30 秒以下。

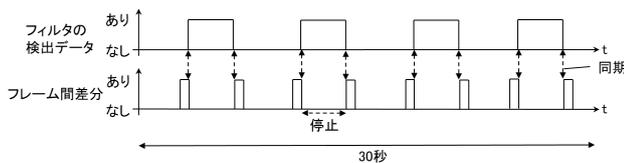


図 4 改善定義を適用した
 検出データのフレーム間差分の同期

Figure 4 Data synchronization with improvement rules.

4.3.3 改善定義の実装

改善定義②～④に基づき、プロトタイプを修正した。スコア計算部では、まず定義②を満たすため、動画の全てのフレームに対して、フレーム間差分データと検出データを比較した際に、検出データが 0 から 1 に変化した時(検出開始時)と、1 から 0 に変化したとき(検出終了時)に、フレーム間差分データが 0 から 1 に変化する場合スコアを加算する。具体的にスコア V は

$$V = \sum_{i=1}^N v_i$$

$$v_i = 1 \quad (D_i \neq D_{i+1} \text{ かつ } F_i = 0 \text{ かつ } F_{i+1} = 1) \quad (2)$$

となる。

なお人数検出フィルタなど、検出データが 0, 1 以外を出力する検出フィルタの場合、様々な数値が出力されている動画の方が最適と考えられる。そこでこれまでに出力されていない値の検出データが出力される度に、スコアをさらに 1 加算し、合計 2 加算する。具体的には、既に出力された検出データの集合を A とすると、スコア V は

$$V = \sum_{i=1}^N v_i$$

$$v_i = 2 \quad (D_i \notin A \text{ かつ } D_i \neq D_{i+1} \text{ かつ } F_i = 0 \text{ かつ } F_{i+1} = 0) \quad (3)$$

となる。

また検出データが 0 以上の検出状態で継続している期間においては、フレーム間差分データも 0 で継続している場合にスコアを加算する。具体的には検出後に継続して検出中か否かを判定する状態 S を

$$S_{i+1} = \begin{cases} 0 & (D_i > 0 \text{ かつ } D_{i+1} = 0 \\ & \text{かつ } F_i = 0 \text{ かつ } F_{i+1} = 1) \\ 1 & (D_i \geq 0 \text{ かつ } D_{i+1} > 0 \text{ かつ } D_i \neq D_{i+1} \\ & \text{かつ } F_i = 0 \text{ かつ } F_{i+1} = 1) \\ S_i & (\text{上記以外の時全て}) \end{cases}$$

と定義することで、スコア V は

$$V = \sum_{i=1}^N v_i$$

$$v_i = 1 \quad (D_i > 0 \text{ かつ } D_i = D_{i+1} \text{ かつ } S_i = 1 \text{ かつ } F_i = 0 \text{ かつ } F_{i+1} = 0) \quad (4)$$

となる。ただし長時間継続するとわかりにくいと考え、加算期間には一定の制限を設ける。

次に、定義③を満たすため、検出データが 0 の期間において、フレーム間データ差分が 0 から 1 に変化した場合にスコアを減算する。具体的にスコア V は

$$V = \sum_{i=1}^N v_i$$

$$v_i = -1 \quad (D_i = 0 \text{ かつ } D_{i+1} = 0 \text{ かつ } F_i = 1 \text{ かつ } F_{i+1} = 1) \quad (5)$$

となる。ただし短時間の変化はノイズと考え、一定期間継続した場合にのみ減算する。

最後に定義④を満たすため、動画・検出フィルタ取得部

において、30秒よりも長い動画を扱わないようにした。これは、アプリケーション開発者が検出フィルタの機能を把握するために確認する動画が短くなるメリットもある。

最終的にスコア V は

$$V = \sum_{i=1}^N v_i$$

$$v_i = \begin{cases} 1 & ((2)\text{または}(4)\text{の時}) \\ 2 & ((3)\text{の時}) \\ -1 & ((5)\text{の時}) \\ 0 & (\text{上記以外の時全て}) \end{cases}$$

となる。

5. 実験

5.1 実験環境

提案技術のアプローチである動画がフィルタの機能把握に寄与可能か否かと、提案技術で選択される動画が適切か否かについて有効性を評価するため、17人の被験者に対して1人ずつ2つの実験を実施した。

実験では、オープンソースのマルチメディアアプリケーション開発用のフレームワーク GStreamer[5]のプラグインとして提供されている顔検出フィルタと人数検出フィルタを利用した。

また、DeepMindが公開している youtube の動画のデータセット Kinetics[6]から、10秒程度の動画を検出フィルタ毎に4本ずつ選んだ。各動画の概要を以下に示す。

<顔検出フィルタ用>

動画①：赤ん坊が四つん這いで下を向いている状態から顔を上げる

動画②：顔を左右に早く振っている画面から、高速でアングルが回転し顔以外が映る

動画③：常に正面を向いたまま体を揺らす

動画④：2人が並んでいる状態から、上下左右に高速に揺れる

<人数検出フィルタ>

動画⑤：1人が正面を向いて体を前後左右に揺らしてあり、その前後をもう1人が通る

動画⑥：2人が顔を左右に高速に振る

動画⑦：4人が並んで演奏しており、右へティルトする

動画⑧：前後に5人ずつ並び、左右に揺れながら演奏している（後列の人は見え隠れする）

なおプロトタイプを用いて確認したところ、顔検出フィルタには動画①が、人数検出フィルタには動画⑧が、それぞれ機能把握に最適と選択された。

5.2 実験内容

【実験1】

- 適用している検出フィルタを伏せた状態で、顔検出フィルタを適用したときの検出データを重畳した動

画①を視聴させる。

Q1: 何の検出フィルタに対して最適と選択された動画か（選択肢は、笑顔検出フィルタ、人数検出フィルタ、顔検出フィルタ）

- 顔検出フィルタに対して選択された動画であったことを教える。

Q2: 動画①は顔検出フィルタに対する最適な動画として適切か。

- 顔検出フィルタを適用したときの検出データを重畳した動画②～④を視聴させる。

Q3: 顔検出フィルタに対して最適と感じる動画はどれか。

【実験2】

- 適用している検出フィルタを伏せずに、人数検出フィルタを適用したときの検出データを重畳した動画⑤～⑧を視聴させる。

Q4: 人数検出フィルタに対して最適な動画はどれか。

- 人数検出フィルタに対して選択された動画が動画⑧であったことを教える。

Q5: 動画⑧は人数検出フィルタに対する最適な動画として適切か。

5.3 実験結果

実験1の結果を表1に示す。※印は想定する回答である。

表1 実験1の結果

Table 1 Results of Experiment 1.

Q1. 何の検出フィルタに対して最適と選択された動画か	笑顔検出フィルタ	3人
	人数検出フィルタ	1人
	顔検出フィルタ[※]	13人
Q2. 動画①は最適な動画として適切か	適切である[※]	14人
	適切ではない	3人
Q3. 顔検出フィルタに最適と感じる動画はどれか	動画①[※]	11人
	動画②	4人
	動画③	2人
	動画④	1人

Q1の結果より、17人中13人が顔検出フィルタに対して最適と選択された動画だと判断できており、動画が検出フィルタの機能把握に有効であると言える。

Q2の結果より、17人中14人が動画①が最適な動画として適切としており、提案技術による適切な動画選択の有効性が伺える。適切ではない理由としては、「動画で人物が笑っていたため、笑顔検出フィルタだと思った（笑顔検出）」という意見があった。検出対象に包含関係（今回であれば顔⇔笑顔）がある検出フィルタ間においては、動画によっては検出フィルタの機能を間違えて把握する危険性があることがわかる。包含関係がある他の検出フィルタの検出データとの重複を利用し、紛らわしい動画を除外するような改善が必要である。また「顔検出にも使えると思うが全身

が映っていたので(人数検出)」という意見もあった。動画における検出対象の占有領域の割合が機能把握に寄与している可能性があり、今後より詳細に調査する。

Q3の結果より、17人中11人が動画①が最適としており、提案技術による適切な動画選択の有効性が伺える。なお動画③と④を複数選択した被験者が1名いたため、総数が18になっている。動画②を最適とした理由としては、「動画に顔以外の要素が少なく、顔の有無がはっきりパートで分かっていた」という意見だった。Q2でも述べた通り、検出対象の占有領域の割合の他、検出対象の有無が時間的に分離していることが機能把握に寄与する可能性も見受けられる。

次に実験2の結果を表2に示す。※印は想定する回答である。

表2 実験2の結果
 Table 2 Results of Experiment 2.

Q4. 人数検出フィルタに最適な動画はどれか	動画⑤	11人
	動画⑥	1人
	動画⑦	0人
	動画⑧[※]	5人
Q5. 動画⑧は最適な動画として適切か	適切である[※]	7人
	適切ではない	10人

Q4の結果より、17人中5人しか動画⑧を最適としておらず、11人は動画⑤を最適としていた。またQ5の結果より、動画⑧が最適な動画として適切としたのは17人中7人で、残り10人は適切ではなかったとしていた。

適切でない理由として、「動画⑧は多すぎて人数を把握できない」、「動画⑤の方が少人数で人数が増えたタイミングが見やすくわかりやすい」、「実際の人数と検出人数が合っていないように見える」といった意見があった。以下に本技術が⑤より⑧を最適な動画と選択した理由を考察する。

動画⑧のスコアが高くなった理由を調査したところ、以下の2つの過剰な加算が見られた。

- 認識しきれない人数を検出した場合の加算
 0, 1だけでなく複数の数値の出力がある検出フィルタの場合、様々な数値が出る方が適していると考え、4.3.3節の式(3)でスコアを加算していたことが逆効果となっていた。今後は人間の視覚で瞬時に判断可能な数を考慮する必要があると考えられる。
- 認識しきれない人数変化を検出した場合の加算
 動画⑧は、前列5人の動きが後列5人に被ることで、人として認識されなくなり、検出データが頻繁に変化する。その結果、4.3.3節の式(2)でスコアが多く加算されていた。人間の視覚は一時的に視認できなくても人と把握しているため、検出領域の遷移から一時的に検出できないデータを補完して検出データの変化を抑制する必要も考えられる。

動画⑤のスコアが低くなった原因を詳細に調査したところ、人が常に動いてフレーム間差分データが常に発生していることから、人数検出フィルタの検出データに変化が発生しても、4.3.3節の式(2)の条件を満たさず、スコアに加算が一切無かったことが分かった。一定時間内に指定の面積以下でフレーム間差分が発生し続ける領域は停止している状態と同じと判断することで改善できると考えられる。

6. まとめ

本稿では、検出フィルタを対象に、検出フィルタの機能把握に寄与可能な動画を選択する技術を提案する。この技術では、予め用意した動画群に対して、動画のフレーム間差分と、検出フィルタを通して動画から出力される検出データの遷移に基づきスコアを算出し、スコアの高い動画を選択する。提案技術により、アプリケーション開発者は、検出フィルタの機能把握に最適な動画を見て、検出フィルタの機能を直観的に把握できる。また、検出フィルタ開発者が、機能把握に寄与可能な動画を作成する必要がなくなる。実装したシステムを用いて、動画がフィルタの機能把握に寄与可能かと、提案技術で選択される動画が適切かについて実験を行い、それぞれ一定の有効性を確認できた。

しかし、検出対象に包含関係がある検出フィルタ間においては、どちらの検出フィルタに対しても有効そうな紛らわしい動画を選択してしまうことや、人間の視覚では一時的に視認できなくても認識しているものが、検出フィルタでは認識できずに検出データと一致しない、などの課題があった。

今後は、他の検出フィルタの検出データとの重複を利用し、包含関係がある紛らわしい動画の除外や、一時的検出データの不足をデータの補完によって人間の認識と検出データを一致させることで課題に対応していく。また、メディア処理基盤への組み込み、社内試行での活用を目指し、さらなる改善と技術開発を進める。

参考文献

- [1] K. Tang, L. Fei-Fei, D. Koller: Learning latent temporal structure for complex event detection. 2012 IEEE Conference on Computer Vision and Pattern Recognition, pp.1250-1257, (2012).
- [2] 内海ゆづ子, 勝手美紗, 岩村雅一, 黄瀬浩一: 物体検出器の検出傾向に基づく動画像のイベント検出. 電子情報学会論文誌 D, J99-D, No.5, pp.583-587, (2016)
- [3] 木村彰吾, 林貴宏, 尾内理紀夫. 類似理由の提示機能を具備した類似動画検索システムの構築: 第49回プログラミングシンポジウム予稿集 2008, pp.97-106 (2008)
- [4] 小牧大治郎, 山口俊輔, 篠原昌子, 堀尾健一, 村上雅彦, 松井一樹: IoTアプリケーション開発のためのメディア処理・制御フレームワーク, マルチメディア, 分散協調とモバイルシンポジウム 2016 論文集, 2016, pp. 79-88 (2016).
- [5] GStreamer: open source multimedia framework, <https://gstreamer.freedesktop.org/>
- [6] Kinectics, <https://deepmind.com/research/open-source/open-source-datasets/kinetics/>