DEIM Forum 2013 D1-5

ビデオストリームからの対象ビデオの短時間検出

千田　佑真† 　古賀　久志† 　渡辺　俊典†

† 電気通信大学大学院 情報システム学研究科 情報システム基盤学専攻
〒 182-8585 東京都調布市調布ヶ丘 1-5-1
E-mail: †{chida,koga,watanabe}@sd.is.uec.ac.jp

あらまし　近年，ストリーミングビデオからのオンラインビデオ検索に関する研究が盛んに行われている．これらの研究では，著作権違反の発見などを目的として，編集されたビデオを原ビデオの類似物として検出すること，特にどのフレームからどのフレームまでが類似動画であるかの特定精度に力点を置いたものが多い．これに対して本研究では，ストリーミングビデオにおいて，検索対象のビデオ (クエリビデオ) が流れ始めてから検出されるまでの時間の短縮を目標とする．この技術は，ビデオ検出をきっかけとして次のサービスを呼び出すような状況で有用になる．提案手法では，ショットごとにハッシュ値を計算し，連続してハッシュ値が複数回一致することを条件としてクエリビデオを検出する．この時，ハッシュ値が一致するショット数のしきい値はクエリビデオの長さに依存させず，ストリーミングビデオの特性を学習して適応的に変化させる．しきい値をクエリビデオの長さに依存させない理由は，人間が少数のショットから瞬時に認識対象のビデオを認識する能力を模擬することを意図している．

キーワード　ビデオ検出，オンライン処理，ハッシュ，ビデオストリーミング

Fast Detection of Query Videos from Video Streams

Yuma CHIDA†, Hisashi KOGA†, and Toshinori WATANABE†

† Department of Information System Fundamentals, Graduate School of Information Systems,
The University of Electro-Communications
1-5-1 Chofugaoka, Chofu, Tokyo, 182-8585 Japan
E-mail: †{chida,koga,watanabe}@sd.is.uec.ac.jp

1. はじめに

デジタル放送や youtube に代表されるビデオストリーミングサービスの普及により，様々な場面でビデオコンテンツを利用する機会が増えている．そしてビデオコンテンツの量が膨大になるにつれ，動画検索技術が重要になっている．一般に動画検索は人間が付与したタグを用いるタグベースの手法とコンテンツベースの手法に大別されるが，本研究ではコンテンツベースの動画検索を取り扱う．

コンテンツベースの動画検索に関しては，ビデオアーカイブ内でのオフライン検索についてこれまで盛んに研究されてきた．例としては，ユーザーが提示した動画ファイルに対するビデオデータベースからの類似動画検索が挙げられる．一方で，近年ストリーミング中のビデオに対して著作権違反検出やコンテンツ管理を行いたいという要求から，ストリーミングビデオに対するオンライン検索が注目されている．ストリーミングビデオに対するオンライン検索ではオフライン検索と比較して以下の点が異なる．

- ストリーミングビデオからの特徴をリアルタイムで抽出必要がある．
- 検索をリアルタイムで行う必要がある．その際，ストリーミングビデオに含まれる動画の切れ目も検出しなければならない．

ストリーミングビデオに対するオンライン検索に関する従来研究では，

（1） 編集されたビデオを原ビデオの類似物として検出する能力

（2） どのフレームからどのフレームまでが類似動画であるかの特定精度．

に力点を置いた研究が多い．これは著作権違反検出への応用という観点では自然である．

これに対して，本研究ではオンライン検索において，検索対象のビデオ (以下クエリビデオと呼ぶ) が流れ始めてから検出されるまでの認識時間の短縮を研究目標とする．この技術は，

ビデオ検出をきっかけとして次のサービスを呼び出すような状況で有用になる．例えば，監視カメラで不審人物を発見したら警備会社に通報するシステムや，テレビ映像からテレビ CM を検出して，それに関する情報をユーザにその場で提示するようなシステムなどが考えられる．

2. 関 連 研 究

2.1 類似ビデオの検索

Huang[1] らの研究では，オンライン環境においてビデオストリームから，クエリビデオと類似するビデオの検索を行った．クエリとビデオストリームとの重みつき編集類似度 WES(Weighted Edit Similarity) を計算することにより，クエリを多少編集した動画も，検出できるようになっている．この研究では類似動画の検出精度を重視しており，例えば，クエリビデオと ξ%(ξ はしきい値) 以上のショットが類似したビデオセグメントを類似とみなすようにしている．このやり方では，クエリビデオに対する類似ビデオを検索する際には，クエリビデオの ξ%以上が流れ終わらないと類似ビデオが発見されず，検出されるまでの時間がクエリビデオの長さに比例する．本研究ではクエリビデオの長さに無関係に，ビデオが流れ始めてからすぐに同一かどうかを判断し，応答時間の短縮を目標としている。

Dohring[3] らの研究では，オフライン環境において，長時間録画したテレビ映像に対して，類似シークエンスの検索を行った．この研究では，フレームからハッシュ値を計算し，ハッシュ値が一致したそれぞれのフレームから前後のフレームを順次見ていき，ショットの出現パターンが類似していた場合に，類似シークエンスとして検出する．特徴として，長い類似シークエンスでも検出することができるが，シークエンスが長いと F 値が低くなってしまう．

2.2 CM 検出アルゴリズム

武 [2] らの研究では，オフライン環境において，長時間録画したテレビ映像に対して，CM を自動発見する技術を提案した．この研究では，CM の繰り返し流れる性質を利用し，出現のパターンが似た映像断片の集合を取り出せば，それが CM となっていることを示した．特徴として，CM の長さが 15 秒または 30 秒であることを用いており，発見対象を CM に絞っているため，一般の動画に応用することは難しい．我々の研究では，クエリビデオに対して長さの仮定を置かないことや，オンライン環境のストリーミングビデオからクエリビデオの検出を試みている点が異なっている．

2.3 フレームからの特徴量の計算

類似動画を検索対象とした研究 [1]，[3]，[5]，[6] やその他多くの研究では，フレームからの特徴量の計算に大域特徴量を用いている．大域特徴量とは，フレーム全体から抽出される特徴量であり，フレーム全体の画素値の情報をもとに計算される．また，特徴量がフレーム 1 枚につき 1 つであるため，計算量もデータ量も少ないという利点がある．これに対して，SIFT[4] に代表される局所特徴量というフレームの一部から抽出される特徴量がある．特定の一部分にだけ注目して特徴量を計算するので，注目部分以外が変わった場合でも特徴量が変わらないという利点がある．しかし，1 枚のフレームから抽出される特徴量の数が膨大であることと，その分計算時間がかかるという問題があることから，リアルタイム性を重視している本研究には向いていない．また，類似ではなく同一の動画を検索対象としていることからも，本研究では大域特徴量を用いた．

3. 提 案 手 法

本研究では，複数個の検索対象のビデオ（クエリビデオ）$V_1, V_2, \cdots, V_k$ を用意し, それらを 1 つのストリーミングビデオからオンラインで高速検出することを目的とする．ここで高速とはクエリビデオが流れ始めてから，検出されるまでの時間を短くするという意味である．

基本的に，提案手法ではビデオのショットごとにハッシュ値を計算し，クエリビデオ V_i とストリーミングビデオ間で連続して複数回 (θ_i とする）ハッシュ値が一致したことを以って V_i を検出する．多くの従来研究では，この θ_i は V_i の長さの何割のように定め，θ_i を V_i の長さに比例させる．この方法では，検出されるまでの時間が V_i の長さに依存し，V_i が長いほど検出される速度が遅くなる．しかし，人間がビデオを認識する様態を鑑みると，人間が認識対象のビデオの長さに関係なく，少数のショットから瞬時に対象ビデオを認識できている．例えば，一度見たことのある CM や映画を再び見た時，長さの長い映画の方が思い出すのに時間がかかることはない．そこで本研究では，この人間の能力を模擬して θ_i は V_i の長さには依存させない．高速にクエリビデオを検出するにはハッシュ値が一致するショットのしきい値 θ_i を小さく設定する必要がある．一方で，θ_i が小さすぎると実際には V_i が流れていないにもかかわらず，V_i を誤検出 (つまり，false positive) が発生する可能性がある．そこで θ_i をストリーミングビデオの特性を学習して適応的に変化させる．具体的には，ストリーミングビデオにおいて V_i が流れていないにも関わらず，偶然ハッシュ値が連続して一致する回数を学習する．

提案手法が前提とするストリーミングビデオの条件を以下にまとめる．

- V_i が流れていないにも関わらずストリーミングビデオのショットのハッシュ値が，V_i のショットのハッシュ値とたまたま連続一致する回数は V_i のショット数より小さい．つまり，V_i が流れ終わった後には，V_i が流れたことを正確に判定できる．提案手法は V_i が流れ終わるのを待たないで，短時間で V_i を検出することを目的とする．
- ストリーミングビデオに V_i が出現する場合，全く同一のものが流れるがキャプチャ時エラー等により一部にノイズがのることがある (図 1)．この場合ノイズによりハッシュ値が途中で途切れ，ノイズの前後で V_i が 2 度検出されてしまう可能性がある．そこで提案手法では，これが起こらないように対策をする必要がある．

提案手法の流れは，最初にクエリビデオをハッシュテーブルに登録するオフライン処理と，実際にビデオストリームからクエリビデオを検索するオンライン処理の 2 つに大きく分けられ

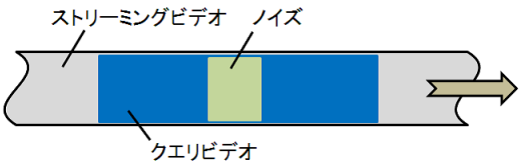

図 1　クエリビデオ V_i の途中にノイズが乗った場合

る (図 2).

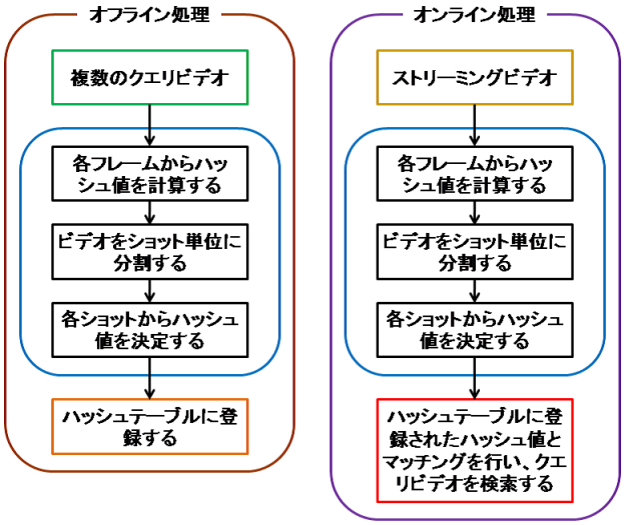

図 2　提案手法の流れ

3.1　クエリビデオのハッシュテーブルへの登録

3.1.1　フレームのハッシュ値の計算

最初にクエリビデオ V_i $(1 \leqq i \leqq k)$ の各フレーム f からハッシュ値を計算する．また、このハッシュ値の計算方法では，武らのアルゴリズム [2] を利用した．f におけるピクセル (x, y) の輝度値を $I_{x,y}$ とし，f のフレーム数を n とする．ハッシュ値の計算手順は以下のようになる．まず，以下の式により，輝度値 $I_{x,y}$ を正規化して，正規化された輝度値 $I^*_{x,y}$ に変換する．

$$\text{輝度値の相加平均 }:\mu = \frac{\sum I_{x,y}}{n} \tag{1}$$

$$\text{輝度値の標準偏差 }:\sigma = \sqrt{\frac{\sum (I_{x,y} - \mu)^2}{n}} \tag{2}$$

$$I^*_{x,y} = \frac{I_{x,y} - \mu}{\sigma} \tag{3}$$

次にフレームを縦方向に 4 分割，横方向に 4 分割して 16 分割し，各ブロックから正規化した輝度の相加平均を計算する．この結果，f に対して，16 次元特徴ベクトルが生成される．輝度値を正規化することにより，画面全体の明るさの変化にロバストな特徴となっている．そして，特徴ベクトルの各要素と，全ブロックの相加平均を比較することにより二値化し，f を 16 ビットの二進値特徴として表現する．これが f のハッシュ値となる．

3.1.2　ショットのハッシュ値の計算

提案手法では，処理の高速化のためにビデオをショット分割し，ショット単位で比較を行う．ショットというのはビデオの 1 シーンのことで，例えばカメラが切り替わったときや，動きの大きい場面でショットが切り替わる．実際にショット分割する際は，連続した 2 枚のフレーム間でハッシュ値を比較し，値が一定値以上変化した場合にショットを分割する．ハッシュ値は 16 ビットの 2 進数なので変化量の値域は 0〜16 までの範囲であるが，様々な動画に対する予備実験の結果により，ショット分割を定めるハッシュ値の変化量の閾値は 3 とした．ショットを決定後，ショットのハッシュ値は，そのショットの中で一番出現回数の多いハッシュ値とした．

3.1.3　ハッシュテーブルへ登録

ショットのハッシュ値をインデックスとしてクエリをショット単位でハッシュテーブルに登録する．つまり，(クエリビデオ ID，ショット番号) のペアをハッシュテーブルへ登録する．

3.2　ストリーミングビデオからのクエリビデオの検出

ここでは，ストリーミングビデオからのクエリビデオの検出方法について説明する．大きな流れは以下のようになる．

（1） ストリーミングビデオからオンラインで最新ショット S を抽出し，そのハッシュ値を計算する．

（2） ハッシュテーブルを検索し，S とハッシュ値が一致する (クエリビデオ ID，ショット番号) のペアを見つける．

（3） ステップ (2) で見つかったクエリビデオに対して，マッチング処理を行い当該クエリビデオが検出されたかどうかを判定する．

（4） S とハッシュ値が異なるが，現在マッチング処理対象になっているクエリビデオに対する処理を行う．

ステップ (1) ではオンラインでショット分割して，抽出された最新のショットに対するハッシュ値を計算する．これは 3.1 節で述べたクエリビデオのショットに対するハッシュ値の計算と同一手順を行えばよい．

ステップ (2) は単純なハッシュテーブルの検索である．S と同じハッシュ値となる (クエリビデオ ID，ショット番号) のペアが複数存在する場合は，ステップ (3) の処理対象となるクエリビデオは複数個になる点に注意されたい．

ステップ (3) では，処理対象のクエリビデオに対して，過去にハッシュ値が一致した回数を 1 増やす．ただし，誤検出を防ぐため，ハッシュ値が一致したショットは時系列順になっていなければならない．そして，過去に θ_i 回ハッシュ値が一致したクエリビデオ V_i が発生すれば，V_i を検出する．

ステップ (4) では，ストリーミングビデオ中で V_i が流れているときに，V_i が流れ終わるか，またはノイズが発生するとハッシュ値が一致しなくなる．このときハッシュ値が連続して一致しない回数を 1 増やし，その回数がしきい値 λ_i を超えたことを以って，現在マッチング途中のクエリビデオに対する処理を行う．また，V_i が流れていないのに過去にたまたまハッシュ値が一致してマッチング処理対象になった V_i に対して、その後ハッシュ値が一致しなかった場合もこの (4) で処理される．

上記の処理を実現するために，各クエリビデオ毎に 3 状態か

らなるステートマシンを設ける．そして，以下の 3 つの状態を設定する．

- マッチング中ではない
- マッチング中
- マッチング確定

例えば，V_i が「マッチング中ではない」という状態は，過去にハッシュ値が一致していないので，マッチング中でない状態である．「マッチング中」という状態は，過去にハッシュ値が一致したが，まだ θ_i 回ハッシュ値が一致していない状態である．「マッチング確定」という状態は，過去にハッシュ値が θ_i 回以上一致し，ストリーミングビデオから V_i が検出された状態である．図 3 にクエリの 3 状態の遷移の様子を示す．状態間の遷移は，図 3 に示すように以下の 4 種類がある．

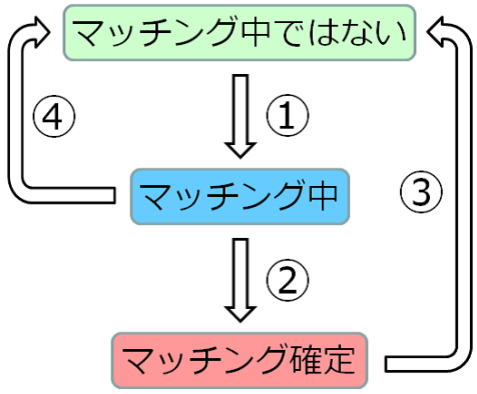

図 3　クエリの状態遷移図

① ハッシュ値が始めて一致する．

② 過去に時系列順にハッシュ値が θ_i 回以上一致する．

③ V_i のショットが最後に検出されてから，λ_i 回ハッシュ値が一致しない．

④ V_i 以外のクエリビデオがマッチング確定する．あるいは V_i のショットが最後に検出されてから，λ_i 回ハッシュ値が一致しない．

①→②→③ が，通常の V_i が検出される流れである．初めて V_i のショットが検出されると，V_i の状態が「マッチング中ではない」から「マッチング中」に変化する．その後，V_i のショットがストリーミングビデオから検出される度に，ハッシュ値が一致した回数が増加し，θ_i を越えると状態が「マッチング中」から「マッチング確定」に変化し，V_i が検出される．

「マッチング確定」状態では，時系列順にその V_i のショットが検出される限りその状態が継続される．一方で，ショットが検出されない回数が λ_i を超えると，状態を「マッチング中ではない」に戻してマッチング終了となる．V_i がストリーミングビデオにおいて流れ終わった時にこの状態遷移が起きる．

V_i が「マッチング中」状態において，V_i 以外のクエリビデオがマッチング確定状態に遷移した時，「マッチング中ではない」に戻る．(遷移 ④) これは 1 個のストリーミングビデオにおいて 2 つのクエリビデオが全く同時に流れるのは不可能であるからである．さらに，ショットが検出されない回数が λ_i を越えた場合も同様に「マッチング中ではない」に戻る．(遷移 ④) これは典型的にはストリーミングビデオ内で，V_i が流れていないにも関わらずたまたまハッシュ値が一致した場合に相当する．偶然，ハッシュ値が一致することで「マッチングでない」から「マッチング中」に状態遷移することはありえるが，偶然ハッシュ値が一致する事象が θ_i 回以上起きる確率は低いので，「マッチング確定」状態に遷移せず誤検出とならない．

3.2.1　しきい値 θ_i の決定アルゴリズム

ストリーミングビデオ内でクエリビデオ内に流れていないのにもかかわらず．たまたまハッシュ値が連続一致する回数は，クエリビデオのハッシュ値とストリーミングビデオのハッシュ値によって決定される．したがって，θ_i はクエリビデオ V_i 毎に設定し，ストリーミングビデオの特性を学習して適応的に変化させる．

θ_i はストリーミング動画中で V_i が流れていないのにショットのハッシュ値が連続して一致する個数（以下，ハッシュ値一致偶然個数と呼ぶ）を学習することで決定され，そのような事象が発生した時に更新される．3 章冒頭で述べた前提条件より，ハッシュ値一致偶然個数は V_i のショット数より小さいので，最長でも V_i の長さだけ時間経過後に，このような事象が発生したことを検知できる．具体的な θ_i の決定アルゴリズムは次のようになる．

ストリーミングビデオの特性が時間の経過に応じて変化することに迅速に対応するため，過去のハッシュ値偶然一致個数の最近の値を重視した加重移動平均 μ_w を求める．ここでは，例えば，TV 映像において，スポーツ番組が流れている時間帯と CM が流れている時間帯でストリーミングビデオの特性が変化するような状況を想定する．そして，ハッシュ値偶然一致個数のばらつきに対応できるよう，標準偏差を μ_w に加算した (式 (4))．

- μ_w：過去に偶然ハッシュ値が連続して一致した個数の加重移動平均
- σ_w：過去に偶然ハッシュ値が連続して一致した個数の標準偏差

$$\theta_i = \mu_w + \sigma_w \tag{4}$$

4 章の実験では，加重移動平均，標準偏差とも過去 3 回のハッシュ値偶然一致個数 (x_1, x_2, x_3 とする) を用いて実装した．特に $\mu_w = \frac{x_1+2x_2+3x_3}{6}$ とした．

しきい値 θ_i の計算タイミング

θ_i はクエリビデオ V_i が流れていないのに，たまたまハッシュ値が一致したときに学習して更新される．具体的には以下の 2 つの事象のいずれかが発生したときに更新する．

（1） V_i が「マッチング中」から「マッチング中ではない」に戻ったとき

（2） V_i が「マッチング確定」から「マッチング中ではない」に戻り，かつマッチしたショット数の合計が V_i の総ショット数の 50%未満の場合

(1) はまさにストリーミングビデオと V_i がたまたまハッシュ値が一致した場合に，「マッチング確定」にならなかったときに相当する．(2) は θ_i が小さいために，クエリビデオが流れていな

いのに誤検出が起きた場合である．マッチしたショット数が少なかったことから，誤検出したことを認識して θ_i を更新している．

3.2.2 しきい値 λ_i の決定アルゴリズム

ストリーミングビデオ中でクエリビデオ V_i が流れているときに，ハッシュ値が連続で一致しなくなるケースは以下の 2 通りである．

（1） クエリビデオ V_i が流れ終わった場合

（2） クエリビデオ V_i の途中にノイズが乗った場合

(1) は通常のクエリビデオ V_i の検出であり，λ_i 回 V_i のショットが検出されないとマッチング終了となる．(2) はビデオのキャプチャ時のエラーにより，クエリビデオ V_i の途中にノイズが乗った場合に発生する．このときのノイズの長さを L とすると，$\lambda_i < L$ の場合，1 つの V_i を 2 回検出してしまう．よって，$\lambda_i > L$ となっている必要がある．しかし λ_i が L に対して必要以上に大きいと，異なる 2 つのクエリビデオ V_i がストリーミングビデオ中で並んだときに，後の V_i の検出が遅くなることや,「マッチング中」状態にあるクエリビデオ V_i の数が増えることから，処理が重くなるというデメリットがある．よって λ_i は L に対して適応的に決定する必要がある．

λ_i は，ストリーミングビデオ中で V_i が流れている時に、ノイズにより V_i のショットが検出されなくなり，その後再び V_i のショットの続きが検出され始めたときの，ノイズによりハッシュ値が変化したショットの長さを学習することで決定され，そのような事象が発生した時に更新される．

ノイズの長さはキャプチャ時のエラーに依存し，時刻には依存せずハードウェアの特性によって決定される．したがって，ノイズの分布は正規分布に従うと仮定し，正規分布の平均と分散を学習することにした．

- μ_v：過去に V_i を検出中にノイズによりハッシュ値が変化したショットの長さの平均
- σ_v：過去に V_i を検出中にノイズによりハッシュ値が変化したショットの長さの標準偏差

とすると，λ_i は式 (5) により計算される．

$$\lambda_i = \mu_v + 2\sigma_v \tag{5}$$

4 章の実験では，移動平均，標準偏差とも過去 9 回のハッシュ値が変化したショットの長さ ($x_1, \cdots, x_9$ とする) 用いて実装した．特に $\mu_v = \frac{x_1+\cdots+x_9}{9}$ とした．

λ_i の計算タイミング

λ_i は以下の場合に計算される，このときにハッシュ値が変化したショットの長さが学習される．

- V_i が「マッチング中」または「マッチング確定」状態にあるとき，ハッシュ値が一致しなくなり，一定時間が経過しないうちに再びその続きのショットが検出された場合

この一定時間は，ハッシュ値が一致しなくなった時点から V_i の長さ分の時間とする．

4. 実　　験

最初に，現実で起こりうるパターンを想定した人工動画を作り，性能評価を行った。次に、複数のテレビ CM をクエリビデオ，テレビ番組を長時間録画したものをストリーミングビデオとして，現実の環境で実験を行った．

4.1 人工動画での実験

この実験では，ストリーミングビデオ中でクエリビデオ V_i が流れていないにも関わらず，たまたま V_i とハッシュ値が一致することがあった場合を想定した人工動画を作成し，しきい値 θ_i の性能評価を行った．評価項目は以下の 2 つである．

- 誤検出の発生回数
- V_i の認識速度

両者はトレードオフの関係にある．前者を減らすには θ_i を大きく設定すれば良い．しかし，θ_i が大きいと V_i が実際に流れた時に検出するまでの時間が長くなる．したがって，θ_i は誤検出を抑えられる最小の値になっていることが望ましい．

実際のストリーミングビデオでは，たまたまハッシュ値が一致する動画の出現パターンには，規則性があると考えられる．例えば，スポーツ番組やニュース番組など，テレビ番組の種類によってハッシュ値の一致しやすさが異なると考えられる．そこで人工動画では，周期的にハッシュ値の一致が起きやすい期間と起きにくい期間を作った．

人工動画は，ベースとなるストリーミングビデオを用意し，これにクエリビデオ V_i とダミービデオを複数を挿入する．ダミービデオは，V_i とハッシュ値が一致するが，V_i とは異なるものである．したがって，ストリーミングビデオが流れたとき，V_i だけが検出され，ダミービデオが検出されなければ最良の結果となる．ダミービデオは，V_i とハッシュ値が一致するショット数を指定した上で作成する．ハッシュ値が一致するショット数の上限は V_i のショット数の 50%とした．したがって，V_i がダミービデオであるかは，(V_i が流れ終わった後には) 一致するショット数から判定できる (3 章の前提条件参照)．

ダミービデオは V_i をもとにして，ハッシュ値だけが同じになるように白黒のタイル状の模様の動画を作る．図 4 にクエリビデオのフレームに対応するダミービデオのフレームの例を示す．提案手法では，フレームのハッシュ値は 16 個の部分ブロックの相対的な輝度値を 0(暗い) または 1(明るい) で表したものなので，元のフレームの暗い部分が黒になっている．

(a) クエリビデオ　　(b) ダミービデオ

図 4　ハッシュ値が同じフレーム

このようにして作成した長さの異なるダミー動画をベースとなるストリーミングビデオに挿入していく．

以下に作成した人工ストリーミングビデオの構成を示す．

- ベースビデオ：1 時間の動画 ×1 個

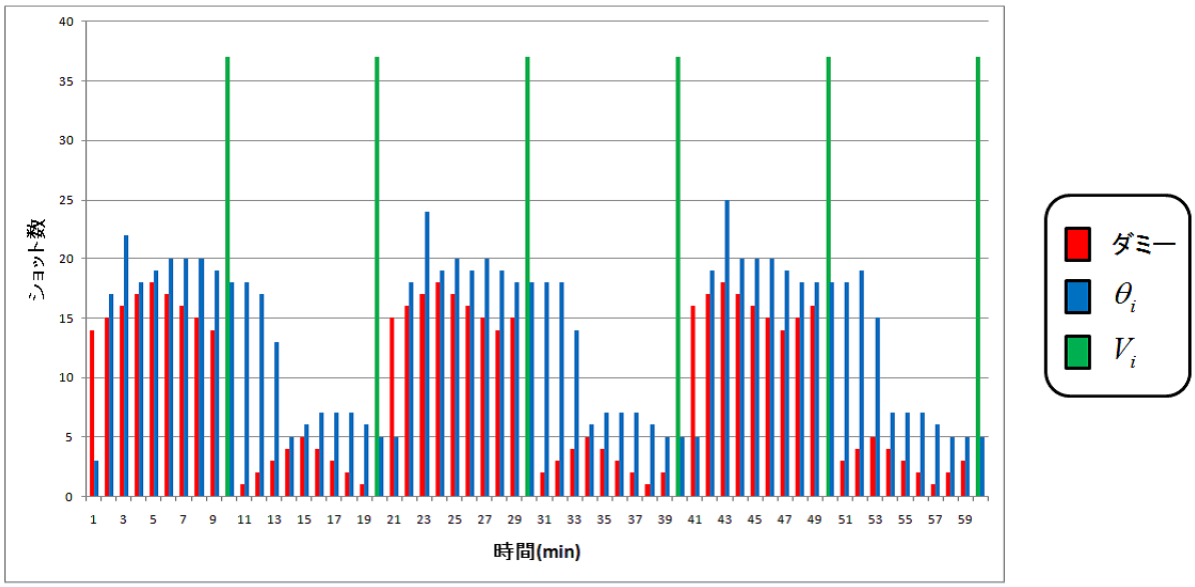

図 5　ダミーの長さに揺らぎのある場合

- クエリビデオ V_i：テレビ CM(12 秒，37 ショット)×6 個
- ダミービデオ：V_i から作成したショット数 (1,2,3,4,5,14,15,16,17,18) が異なるビデオ ×54 個

1 時間のベースビデオに対して 1 分周期でダミービデオを挿入した．ダミービデオにおいて V_i とハッシュ値が一致するショット数は，10 分周期に大きくなったり，小さくなったりする．これが上述したハッシュ値の一致が起きやすい期間と起きにくい期間に対応する．ただし，それぞれの期間内でもダミーのハッシュ値が一致する長さは一定ではなく多少の揺らぎがある．また，10 分おきに V_i を挿入した．

実験結果

この動画に提案手法を適用した結果を図 5 に示す．このグラフでは横軸が時刻，縦軸がショット数である．ダミー動画でハッシュが一致するショット数，θ_i をそれぞれ赤，青の棒で表す．また，クエリは緑色の棒で示されている．

赤，青の棒の長さを比較すると，まず全体的には，ダミー動画の長さに応じて θ_i が適応的に変化しており，ストリーミングビデオの性質に応じて誤検出が起きない範囲で短い時間で V_i を検出することができていることがわかる．赤の棒の長さが青の棒の長さ (θ_i) より長い時，ダミー動画で θ_i 個以上のショットのハッシュ値が一致したことになり，誤検出となる．図 5 より，赤の棒の長さが青の棒の長さより長い誤検出は，3 回起きている．ダミー動画は 54 個含まれているので，誤検出の割合は $\frac{3}{54} \times 100$ =5.6%であった．誤検出はダミー動画のハッシュ値が一致しにくい区間から一致しやすい区間に切り替わった時に 1 回のみ発生している．ハッシュ値が一致しにくい区間では適応的に学習した結果，θ_i が小さくなっているので，ハッシュ値が一致しやすくなった直後に誤検出が起きるのは仕方がない．しかし，1 度誤検出と判定すると，次からは θ_i が急激に大きくなった結果，誤検出は発生していない．

しきい値 θ_i は過去 3 回分の加重移動平均をとっているのにも関わらず，誤検出が 1 回しか発生しない理由は，式 (4) で分散の項を加えたためである．分散の項によりハッシュ値の一致しやすさが増大すると θ_i が速く上昇する．しかし，逆に，分散の項の影響で，ハッシュ値が一致しにくくなる場合は θ_i の下がり方が遅く，V_i が検出されるまでの時刻が，実際のダミー動画の長さと較べて相対的に遅くなってしまっている．ハッシュが一致しにくく変化する場合の検出時間の短縮が今後の課題である．

4.2　現実動画での実験

提案手法のアルゴリズムが現実動画でも正しく動くかどうかを検証するために，テレビ放送を録画したものをストリーミングビデオとして実験を行った．以下に使用したビデオの構成とファイル情報 (表 1，表 2) を示す．

- ストリーミングビデオ：テレビ放送 (アナログ) を 24 時間録画した動画 ×1 個
- クエリビデオ $V_i(1 \leqq i \leqq 5)$：ストリーミングビデオから抽出したテレビ CM×5 個

表 1　使用したビデオのファイル情報

	長さ (sec)	サイズ (pixel)	フレームレート (fps)	ショット数
ストリーミングビデオ	24hour	720*480	29.97	43758
クエリビデオ V_1	12	720*480	29.97	37
クエリビデオ V_2	14	720*480	29.97	30
クエリビデオ V_3	15	720*480	29.97	23
クエリビデオ V_4	30	720*480	29.97	33
クエリビデオ V_5	15	720*480	29.97	25

表 2　ストリーミングビデオ中に含まれるクエリビデオ V_i の数

V_1	V_2	V_3	V_4	V_5
10	20	2	2	6

ただし，アナログ放送の動画ではフレームの上下に黒帯が入っているので，これを除外するため上下高さ 60 ピクセルずつカットして使用した．ファイルとして録画したビデオを再生

しながら処理して，オンライン環境を検証した．提案手法の可変しきい値 θ_i と λ_i の効果を検証するため，θ_i と λ_i を固定値にした場合と，可変にした場合とでそれぞれ V_i の検出精度と検出速度の比較を行った．

検出精度の評価は再現率と適合率，および F 値によって行った．適合率 P，再現率 R，F 値はそれぞれ以下の式で表わされる．

$$P = \frac{\text{検索結果のうち正しく検出された } V_i\text{の数}}{\text{検出された } V_i\text{の数}} \tag{6}$$

$$R = \frac{\text{検索結果のうち正しく検出された } V_i\text{の数}}{\text{ストリーミングビデオ中に含まれる } V_i\text{の数}} \tag{7}$$

$$F = \frac{2PR}{P+R} \tag{8}$$

V_i の検出速度の評価は，ストリーミングビデオ中で V_i が流れ始めてから V_i が検出されるまでにかかったショット数で評価する．ただし，誤検出だったときはカウントせず，正しく検出された場合のみで評価する．

4.2.1 固定しきい値での実験

最初に，V_i 状態に影響するしきい値 θ_i と λ_i をそれぞれ固定値 3 として実験を行った．検出速度に関しては，θ_i が固定値となっているため評価は行っていない．V_i の検出結果は以下の表のようになった．

表 3　V_i の検出結果 (固定しきい値)

	V_1	V_2	V_3	V_4	V_5	V_i 全体
検出数	12	51	4	3	9	79
正解	10	20	2	2	6	40
誤検出	2	31	2	1	3	39
適合率 P	0.83	0.39	0.50	0.67	0.67	0.51
再現率 R	1	1	1	1	1	1
F 値	0.90	0.56	0.67	0.80	0.80	0.68

結果から，再現率は 1 となり，ストリーミングビデオに含まれている V_i はすべて正しく検出することができたが，誤検出が多く，適合率は 0.51 となった．特に V_2 については適合率 0.39 とが低くなっていた．この原因は，クエリ検出後にノイズが乗ってマッチングが終了し，ノイズ後のクエリビデオの続きに対してもクエリを検出してしまったためであった．

その他の誤検出の例として，V_i とまったく無関係のショットがたまたま連続で一致して V_i が検出されてしまうことがあった．例えば，白い背景に企業のロゴが入った物や，全体が黒で塗りつぶされたフレームにおいてハッシュ値が一致することが多かった．

4.2.2 可変しきい値での実験

次に，提案手法をそのまま用いて，ストリーミングビデオの特性を学習して適応的に変化するしきい値 θ_i と λ_i が，クエリビデオ V_i の検出に実際どれほど効果があるのかを検証した．V_i の検出結果は以下の表のようになった．

結果から，再現率は 1 となり，ストリーミングビデオに含まれている V_i はすべて検出することができ，適合率は 0.82 と固定しきい値を利用した場合よりも改善された．特に V_2 に関しては，可変しきい値 λ_i を適用したことにより，F 値が 0.56 から 0.93 へ大きく改善された．V_4 以外は多少精度が改善されたがほとんど変わらない結果となった．誤検出の原因は大きく分けて以下の 3 つに分類された．

表 4　V_i の検出結果 (可変しきい値)

	V_1	V_2	V_3	V_4	V_5	V_i 全体
検出数	11	23	3	4	8	49
正解	10	20	2	2	6	40
誤検出	1	3	1	2	2	9
適合率 P	0.90	0.87	0.67	0.50	0.75	0.82
再現率 R	1	1	1	1	1	1
F 値	0.95	0.93	0.80	0.67	0.86	0.90

（1） クエリビデオ V_i の途中にノイズが乗って，1 つの V_i に対して 2 回 V_i が検出された場合

（2） クエリビデオ V_i のバージョン違いが流れた場合

（3） しきい値 λ_i が大きい時に，断続的にハッシュ値が偶然一致することが繰り返され，V_i が流れていないのに V_i を検出した場合

(1) はストリーミングビデオの初期の学習段階の誤検出であり，ストリーミングビデオの始めの方で発生していた．(2) は CM のバージョン違いによるもので，V_i の最後に流れる企業のロゴなどでショットが一致してしまいそれが誤検出につながった．(3) は V_i とまったく無関係のショットがたまたま複数一致した場合に発生していた．この対策として，λ_i を学習する条件として，V_i の長さより短い範囲ではなく，現在のマッチング中の V_i のショット番号から最終ショットまでより短い範囲にすることなどが考えられる．

全体的には固定しきい値の場合と比べて，F 値が 0.68 から 0.90 に増加したので，可変しきい値を導入したことによる効果が確認できた．

検出速度

正解の V_i が検出されたとき，V_i が流れ始めてから何ショットで検出したかで，検出速度を評価した．図 6 に V_i ごとに検出されるまでにかかったショット数を比較した．この結果ほとんどの V_i は θ_i の最小値 3 で高速に検出されているが，V_i ごとにみるとそれぞれで検出されるまでのショット数に偏りがあることがわかった．特に V_1 がマッチ確定までの時間が遅く，逆に V_2 はマッチ確定までの時間が速い傾向が見られた．したがって V_i によってたまたまハッシュ値が一致する確率や，ノイズの大きさが異なるということなので，このことから，提案手法で V_i ごとにしきい値を設定したことは，ストリーミングビデオ中での V_i の特性に合わせて効率よく V_i を検索できていると考えられる．

5. まとめと今後の課題

本研究ではオンラインのビデオストリームに対して，クエリとして用意した複数のビデオをリアルタイムで高速に検索する

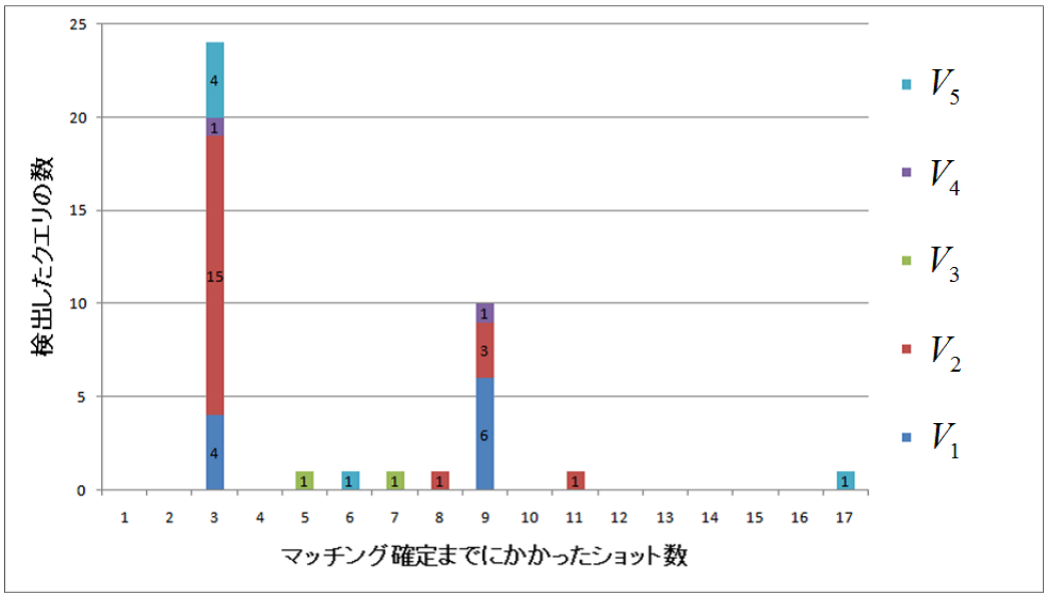

図 6　V_i ごとの検出速度の比較

手法を提案した．人間が過去に一度見たことのあるビデオをビデオの長さに依存せず瞬時に認識検出できる能力があることから，しきい値 θ_i を V_i の長さに比例させないことで、より人間が実際目で見たときの判断に近い検索が可能となった．また、ストリーミングビデオの特性を学習することで、その場面に合わせてしきい値を変化させ，誤検出の割合を小さくすることができた．特にストリーミングビデオ中でたまたまクエリビデオが流れていないにもかかわらずハッシュ値が一致した場合や，逆にクエリビデオの中でノイズが入って部分的にハッシュ値が変化した場合でも検出できるように 2 種類の可変しきい値を設定した．

人工動画の実験では，可変しきい値の変化の様子を人工のストリーミングビデオを作成して評価した．この実験から，検出速度に関しては，ストリーミングビデオの特性に応じて誤検出が起きない範囲で短い時間で V_i を検出できることが確認できた．また，検出精度に関しては，最初の学習期間では誤検出が発生してしまうものの，それ以降は誤検出が起きずしきい値が適応的に変化することが確認できた．また，しきい値に分散の項を足しているので，しきい値の上がるスピードは速いが下がるのが遅くなっていた．そこでストリーミングビデオの長期的な特性を学習したり，しきい値の上げる方と下げる方で計算方法を変えるなどの対策が考えられる．

現実動画での実験では再現率は 1 となり，ストリーミングビデオ中に含まれる V_i はすべて検出することができた．適合率も可変しきい値を適用することで，0.51 から 0.82 に改善され，F 値は 0.9 となった．適合率が 1 にならない理由は，初期の学習期間や CM のバージョン違いによるものと，クエリの長さよりも短い範囲でハッシュ値がたまたま一致することが原因であった．よって，後者の場合は λ_i を学習する条件として，V_i の長さより短い範囲ではなく，現在のマッチング中の V_i のショット番号から最終ショットまでより短い範囲にするなどによって対策をしていきたい．検出速度は V_i によって異なるが，一部のクエリビデオ内にノイズが乗っているケースを除いて，大部分のケースでは 3〜数ショットで高速に検出することができた．

謝辞　本研究は科学研究費基盤研究（C）課題番号 24500111 の支援を受けて行った．

文　献

[1] Zi Huang, Heng Tao Shen, Jie Shaoy, Bin Cuiz, Xiaofang Zhou, “Practical Online Near-Duplicate Subsequence Detection”, IEEE Transactions on Multimedia, Vol.12, No.5, pp.386-398, 2010

[2] 武小萌, 佐藤真一, “時間的共起ハッシュアルゴリズムに基づいた CM 検出・同定”, 画像の認識・理解シンポジウム (MIRU2010)

[3] Ina Dohring, Rainer Lienhart, “Mining TV Broadcasts for Recurring Video Sequences”, Conference on Image and Video Retrieval, pp.327-356, 2009

[4] D. G. Lowe, “Distinctive Image Features from Scale-Invariant Keypoints”, Journal of Computer Vision, pp91-110, 2004.

[5] Klaus Schoeffmann, Laszlo Boeszoermenyi, “Video Sequence Identification in TV Broadcasts”, International Conference on Multimedia Modeling, pp129-139, 2011

[6] Xiangmin Zhou, Lei Chen, “Monitoring Near Duplicates Over Video Streams”, ACM Multimedia, pp521-530, 2010