

ビデオ検索のための MPEG-7 Video signature の direct-mapped cache への適用法の検討

村 林 昇^{†1} 吉 田 健 一^{†1}

インターネットのブロードバンド化や安価な汎用 PC の高性能化と共に、YouTube などのビデオ投稿サイトの人気が増加している。投稿されるビデオの中には、著作権を侵害し編集処理されたビデオも含まれることから、このような著作権侵害ビデオを検出する技術の重要性が増している。この課題に対応するため、以前の研究で、我々は画像特徴量と direct-mapped cache を組合せた高速ビデオ検出手法である Tiny LSH 法を提案した。本研究では、ビデオ検索のための MPEG-7 video signature を direct-mapped cache への適用法を検討した。提案手法では、MPEG7 video signature の一部を選択し粗量子化するアイデアに基づき Tiny LSH 法を発展させ、更に direct-mapped cache を組合せることで、従来手法と同様な高速検索を可能としながら、検索精度を大幅に改善した。本論文ではビデオ検索の評価実験に基づき改善効果を報告する。

Study of Applying MPEG-7 Video Signature to Direct-mapped Cache for Video Retrieval

NOBORU MURABAYASHI^{†1} and KENICHI YOSHIDA^{†1}

According to the increasing popularity of the video-sharing websites, copy-right infringed videos become the critical issue. These videos are not only the original ones, but also the edited ones. Under these circumstances, the video detection technology increases its importance. In our previous study, we proposed the direct-mapped cache method based on the RGB color space feature to handle the large amount of videos in the internet. Although our method could achieve fast processing speed, the detection accuracy had a room for improvements. In this paper, we proposed new method to improve the accuracy. For this purpose, use of MPEG-7 video signature and a tiny LSH method for MPEG-7 video signature are examined.

1. はじめに

ビデオ投稿サイトの人気が高まるにつれ、著作権を侵害したビデオの存在が問題になっている。そのようなビデオ投稿サイトには、元のビデオを編集した著作権侵害ビデオが多く存在する。編集されたビデオは、元のビデオのフレーム画像の一部をトリミングしたもの、ビデオの記録ビットレートやフレームレートを低下させ画質が劣化したビデオなどである。このような状況から、編集されたビデオの検出技術の重要性が増している。我々は、以前の研究 1) で RGB 色空間における画像特徴量と direct-mapped cache に基づいた高速データ検索手法を用いたビデオ検索手法である Tiny LSH 法を提案した。Tiny LSH 法は、大規模なビデオデータベースを対象にした検索に対応可能であるが、検索対象のビデオの長さが短い場合、検索の正解率は十分ではなく改善の余地があった。

このような課題に対応するため、本論文では MPEG-7 video signatures²⁾ を direct-mapped cache に適用しビデオを高速検索する手法を検討した。また、編集ビデオを模したテストデータを用いた検索性能の評価実験と従来手法との性能比較の結果についても報告する。

以下、2章で関連研究について、3章で提案する適用法に基づくビデオ検索手法について、4章で評価実験について、5章で考察を行い、6章でまとめと今後の課題について述べる。

2. 関連研究

ビデオ検索は、以前から多くの手法が研究されている。編集されたビデオ検索の観点から、検索に用いる特徴量の選択は重要である。画像特徴量は、大局的特徴量と局所的特徴量を考えることができる。

大局的画像特徴量は、例えば、色特徴、テクスチャ特徴など様々な種類がある。我々は先の研究 1) で、RGB 色空間に基づいた画像特徴量を用いた複製ビデオ検索手法である TinyLSH 法を提案した。大局的画像特徴量は、特徴抽出処理が簡単であるが、類似データに対するマッチング性能は、次に述べる局所的特徴量と比較して良好ではない。MPEG-7 video signature²⁾ は、ISO/IEC で制定された新しい国際標準規格で、類似データに対応するために様々な種類の signature データから構成されている。また、MPEG-7 video signature

^{†1} 筑波大学
University of Tsukuba

は大局的特徴量として用いることができ、データ処理が簡単である。

一方、例えば、SIFT³⁾ や SURF⁴⁾ などの局所特徴量は、特徴点が存在する画像に対して、マッチング性能が優れている。このような局所特徴量を用いたビデオ検索については多くの研究が行われている。例えば、5) は SIFT を用いた複製ビデオ検索の研究を行っている。また、SIFT を用いた bag-of-feature 法に基づいた良好な検索性能のビデオ検索⁶⁾ も研究されている。しかし、局所特徴量の抽出や bag-of-feature 法に基づくデータ処理は、大局的特徴の場合に比べて処理が簡単ではなく計算量も多くなる。検索対象のビデオが大規模になると、このようなデータ処理や計算時間は無視できない課題となる。

高速な類似データ検索については LSH (locality sensitive hashing)⁷⁾ が提案されている。LSH は多次元空間のデータベースから類似データを検索できるが、近傍検索処理を行うために多くのメモリー空間が必要になる。このような課題に対応するため、我々は先の研究 1) で、高速スパムフィルタの研究⁸⁾ で提案された direct-mapped cache をビデオ検索に適用した。

以上のような背景に基づいて、本論文ではビデオ検索のための MPEG-7 Video signature の direct-mapped cache への適用法の検討を行った。

3. MPEG-7 video signature を用いたビデオ検索

ここでは、MPEG-7 video signature²⁾ を direct-mapped cache¹⁾ に適用したビデオ検出手法を提案する。提案手法は、136 年規模のビデオ検索に対応可能で、以下の特徴がある。

- 従来手法と比較して平均正解率は 20%改善。
- 類似ビデオの検出に対応するために MPEG-7 video signature の部分的な選択と元のデータの粗量子化処理。
- MPEG-7 video signature と direct-mapped cache の組み合わせに基づく、ビデオの高速検索。

従来手法では、RGB 特徴量の上位ビットを選択し粗量子化することで、ビデオにおける類似画像フレームの検索に対応していたが、提案手法では、MPEG-7 video signature の部分的な選択と選択したデータの粗量子化によって対応する。

3.1 MPEG-7 video signature

MPEG-7 video signature は、ISO/IEC の国際標準規格として 2010 年に公開された²⁾。MPEG-7 video signature の詳細な内容は規格書²⁾ で説明されているので、ここでは概要を説明する。MPEG-7 video signature は、A type 32 データと D type 348 データの 380 の

データから構成され、各データは 3 値データ (0, 1, 2) である。A type signature は、画像フレーム内のあらかじめ指定された領域における 8 ビットの平均輝度 Y (YCbCr 色空間) とデータレベル $[0, 255]$ 中心値 (128) の差分値としきい値 Tha との比較に基づいて決められる。また、D type signature は画像フレーム内のあらかじめ指定された 2 つの領域の平均輝度の差分値としきい値 Thd との比較に基づいて決められる。例えば、 D_2 は 2 つの領域の平均輝度 Y_1 と Y_3 の差分値としきい値 Thd との比較に基づいて 3 値 (0, 1, 2) のいずれかになる。しきい値は、各 A type signature, 各 D type signature ごとに決められている。

A type と D type の signature データは、それぞれ下式で設定される。

$$A = \begin{cases} 2 & (if \quad Y - 128 > Tha) \\ 1 & (if \quad |Y - 128| \leq Tha) \\ 0 & (if \quad Y - 128 < Tha) \end{cases} \quad (1)$$

$$D = \begin{cases} 2 & (if \quad Y_i - Y_j > Thd) \\ 1 & (if \quad |Y_i - Y_j| \leq Thd) \\ 0 & (if \quad Y_i - Y_j < Thd) \end{cases} \quad (2)$$

MPEG-7 video signature は 380 データで構成されているが、予備実験の結果、検索正解率の改善効果が大きかった 20 の A type signature と、36 の D type signature を選択した。以下の説明では、それら選択した A type signature と D type signature に着目して説明する。

3.2 ビデオの類似度

ビデオの類似度を計算するため、提案手法ではビデオ間距離 d を以下のように定義する。A type signature の場合；

$$d = \sum_{f=\text{検出ビデオ開始位置}}^{f+\text{検索ビデオ長}} \{da_f\} \quad (3)$$

ここで、 da_f は A type signature に基づいたフレーム間の Hamming 距離である。

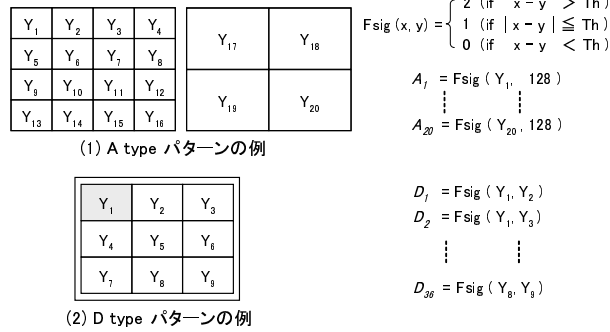


図 1 MPEG-7 video signature の例
Fig.1 Example of MPEG-7 video signature

$$da_f = \sum_{n=1}^{20} \{hma_n\} \quad (4)$$

$$hma_n = \begin{cases} 1 & (\text{if } Ao_n \neq At_n) \\ 0 & (\text{if } Ao_n = At_n) \end{cases} \quad (5)$$

Ao_n : Original video
 At_n : 検索対象ビデオ (Cropped / Rescaled)
(1 ≤ n ≤ 20)

D type signature の場合 ;

$$d = \sum_{f=\text{検出ビデオ開始位置}}^{f+\text{検索ビデオ長}} \{dd_f\} \quad (6)$$

ここで, dd_f は D type signature に基づいたフレーム間の Hamming 距離である .

$$dd_f = \sum_{n=1}^{36} \{hmd_n\} \quad (7)$$

$$hmd_n = \begin{cases} 1 & (\text{if } Do_n \neq Dt_n) \\ 0 & (\text{if } Do_n = Dt_n) \end{cases} \quad (8)$$

Do_n : Original video
 Dt_n : 検索対象ビデオ (Cropped / Rescaled)
(1 ≤ n ≤ 36)

3.3 MPEG-7 video signature に対する Tiny LSH 法の適用

本論文では, 編集されたビデオの検索を対象とする. 編集ビデオは原ビデオから様々な処理が施されている. 具体的には, 画像フレームのトリミング, 記録ビットレートやフレームレートの変更などである. このような編集処理が施されたビデオと原ビデオは対応する画像フレーム同士が同一ではなく類似関係にあり, 画像フレームから生成した特徴量の類似検索が必要になる. LSH は p-stable 分布に基づいた手法でハッシュ関数を複数用いることで, 類似検索に対応し, 我々が先の研究で提案した Tiny LSH 法は, RGB 特徴量の低位ビットを無視し高位ビットを選択する手法で類似検索に対応した.

しかし, MPEG-7 video signature は RGB 特徴量と異なり, 380 次元の特徴ベクトルで各要素は 3 値化されたデータであり, Tiny LSH 法をそのまま適用することはできない. そのため, Tiny LSH 法の情報が顕著な特徴を取り出すという考えに基づき, MPEG-7 video signature の内から検索に大きく寄与する signature データを選択する. 具体的には, 図 1 に示すような 380 データの MPEG-7 video signature の内から type A signature から 20 データと type D signature から 36 データを予備実験の結果に基づいて選択した.

図 2 に MPEG-7 video signature に対する Tiny LSH 法を適用したハッシュ処理を示す. 選択した type A signature 20 データは各データ 2 ビットで, ハッシュ処理を行うために, 全データを 40 ビットのデータ列とする. このデータ列を汎用ハッシュ関数である SDBM ハッシュ関数⁹⁾ に入力してハッシュ値を生成する. このハッシュ値に基づいて, 選択した type A signature の 20 データを video signature database に蓄積する.

type D signature 36 データについては, 次式に基づいて粗量子化を行う.

$$d = \begin{cases} 1 & (\text{if } D = 2) \\ 0 & (\text{if } D = 1 \text{ or } 0) \end{cases} \quad (9)$$

この粗量子化によって, type D signature は 3 値データから 2 値データ d に変換し,

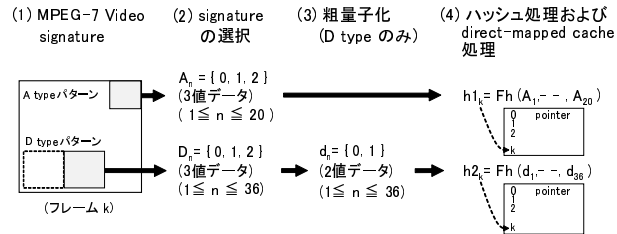


図 2 MPEG-7 video signature への Tiny LSH 処理
Fig. 2 Tiny LSH process (type A and type D signature)

36 ビットのデータ列とする．その後の処理は，type A signature の場合と同様に，SDBM ハッシュ関数によってハッシュ値を生成し，type D signature 36 データを video signature database に蓄積する．

3.4 Direct-mapped cache に基づいたハッシュテーブル

ハッシュテーブルは，大規模データベースを対象にした高速検索の一般的な手法である．ビデオは類似フレームを多く含むので，video signature データベースに多くの類似データが存在する．もし，従来手法のハッシュテーブルを用いた場合，多くの類似したデータに対して計算し，重複した処理を行うことになる．ハッシュ値の衝突によってハッシュテーブルのリンク長が長くなるため検索時間が長くなる課題がある．

この課題に対応するため，我々は先の研究¹⁾¹⁰⁾で重複した類似データを無視することで高速検索を可能にした Direct-mapped cache のデータ構造を提案した．図 3 に，MPEG-7 video signature に適用した Direct-mapped cache のデータ構造を示す．

次に，ビデオ検索のためのポインタ蓄積処理と検索処理について説明する．これらの処理は，MPEG-7 video signature を用いたことを除き，基本的に文献¹⁾¹⁰⁾で用いたプロセスと同様な処理のため，ここでは，処理のポイントについて説明する．また，type A signature を用いた場合と type D signature を用いた場合の処理は，基本的に同様のため，主に type A signature を用いた場合の処理について説明する．

type A signature を用いた場合のビデオ検索処理を図 5 に示す．ビデオ検索処理は，ポインタに基づいた検出位置を基点に検索目標ビデオの signature データと video signature データベースとのビデオ間距離を計算することで行う．

図 4 に，ポインタ蓄積処理のプロセスを示す．この処理プロセスは，基本的に文献¹⁾¹⁰⁾で用いたプロセスと同じである．

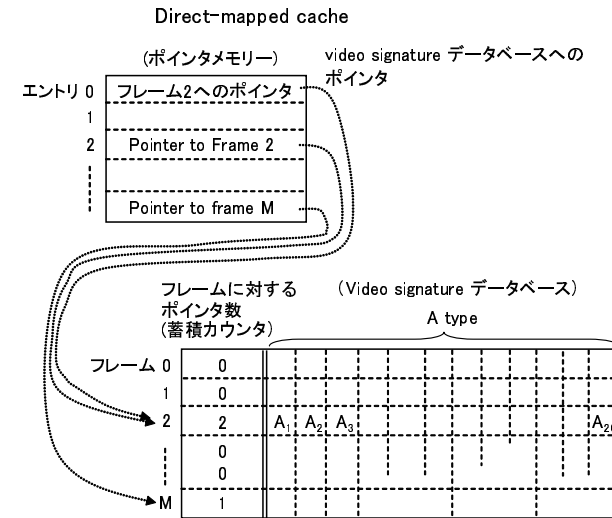


図 3 Direct-mapped cache (A type signature を用いた場合)
Fig. 3 Direct-mapped cache (In case of using type A attribute)

図 4 の 1 行目 ~ 4 行目で，文献¹⁾¹⁰⁾で採用したハッシュ値の生成プロセスと同様の手法に基づいて，k 個のハッシュ値を生成する．具体的には，計算結果 $A_n (1 \leq n \leq 5)$ を長さ 20 の文字列としてまとめた後，そのまとめたデータを汎用ハッシュ関数 (実験では SDBM ハッシュ関数⁹⁾) を使用) の入力データとしてハッシュ処理して，Tiny LSHa.0 の値とした．また Tiny LSHa.(i-1) ($1 \leq i$) の値を汎用ハッシュ関数の入力データとしてハッシュ処理した計算結果を Tiny LSHa.i ($1 \leq i \leq k$) の値とした．

図 4 の 5 行目 ~ 17 行目は signature データに基づくポインタの蓄積処理である．図 4 の 6 行目 if 文 ~ 9 行目はハッシュ値の衝突がない場合の処理で，単にポインタを蓄積し，蓄積したフレーム (すなわちポインタが参照しているフレーム) に対する蓄積カウンタを増やす．各フレームが何個のポインタで参照されているかは，この蓄積カウンタによって判定することができる．図 4 の 10 行目 if 文 ~ 16 行目はハッシュ値が衝突する場合の処理である．さらに，11 行目の if 文では，文献¹⁾¹⁰⁾による重要性の判定基準に基づいて，重要性が低いポインタがある場合にはそのポインタを上書きして，上書きしたポインタが参照する蓄積カウンタを減らし，書き込んだポインタが指すフレームの蓄積カウンタを増やしている．

```
蓄積処理 store(f) f:フレーム
1: for(i=0 ; i<k ; i++){
2:   signature のハッシュ値 pa[i]=TinyLSHa_i(f);
3: }
4: for(i=0 ; i<k ; i++){
5:   // signature データの蓄積処理
6:   if(pa[i] ハッシュ値の衝突がない){
7:     signature によるポインタを蓄積;
8:     (蓄積した f に対する蓄積カウンタ)++;
9:   }
10:  else{
11:    if(重要性が低い signature によるポインタがある){
12:      重要性が低い signature のポインタを上書きして蓄積;
13:      (上書きした f に対する蓄積カウンタ)--;
14:      (蓄積した f に対する蓄積カウンタ)++;
15:    }
16:  }
17: }
```

図 4 データ蓄積処理 (type A signature の場合)
Fig.4 Data storing process (In case of type A signature)

検索処理は Direct-mapped cache に記憶したポインタを用いて, signature データベースに蓄積されている signature データを用いてビデオフレームシーケンス間の距離を計算する. 検索処理では処理時間の短縮化のため, 文献¹⁾¹⁰⁾と同様に, 処理の間引きを行っている. すなわち, 図 5 の 1 行目の foreach 処理により, 検索目標ビデオに含まれるフレームの第 1 から最終フレームまで均等間隔で L 箇所の各フレーム位置 q ($=q_1, q_2, \dots, q_L$) ごとに以下の処理を行う.

以下で説明する処理は, 文献¹⁾¹⁰⁾と同様なので, ここでは処理のポイントについて説明する. 4 行目の for 処理は, q を中心に前後 m の範囲 (すなわち $(q-m) \sim (q+m)$ の間) で位置を変えて最小距離を検索することで, 検索目標ビデオの signature データと signature データベースの Original ビデオにおける signature データとがもっとも一致する位置を検索する.

図 5 の 6~7 行目で, 検索位置にポインタが記憶されているかチェックし, 8 行目の if 文でポインタが記憶されている場合に, 9~12 行目でビデオ間距離の計算に基づいて, 画像フレームのシーケンスが一致するか各 signature に基づいて距離計算を行い, 最小距離 d_1 とその位置 t_1 を逐次更新して検索したデータをメモリに記憶して行く. ここで, 類似フレームの検索結果は q, m に対して複数あるので, 最小距離を与えるビデオの開始位置 t (q, m によるずれを補正した値) を検索結果とする.

```
検索処理 retrieve(v) (v: 検索目標ビデオ)
ds:signature によるビデオ間距離
1: foreach (q:フレーム位置 in {q1,q2, ... qL}){
2:   // 第 1 から最終フレームまで均等間隔で L 箇所;
3:   // signature に基づく検索
4:   for ( t'=(q-m) ; t' <= (q+m) ; t'++) {
5:     t1=t' に対する signature データベース中のフレーム位置;
6:     位置 t1 に対応したポインタが Direct-mapped
7:       cache に記憶されているかチェック;
8:     if(位置 t1 に対応するポインタが記憶されている){
9:       t1 を基準に v と Original ビデオ signature データベース中の
10:      データとのビデオ間距離を計算;
11:      最小距離 d1, 開始位置 t1 "を更新しメモリに記憶;
12:      (ここで t1 "=t1-q)
13:    }
14:  }
15: }
16: return(メモリに記憶された最小距離 ds, 開始位置 t1 );
```

図 5 ビデオ検索処理
Fig.5 Data retrieval process

4. 評価実験

4.1 テストデータ

提案手法の性能評価を行うため, 表 1 に示す 3 種類のテストデータを用いた実験を行った. これらテストデータは, "the Open Video Project" の Web サイト¹¹⁾ からダウンロードしたビデオ ("ダウンロードビデオ"とする) を用いて準備した. ダウンロードの条件は, {All field, Any Genre, More than 10 minutes, sound, color, MPEG2} であり, 全部で 109.5 時間分のビデオを収集した. ダウンロードビデオは, 様々な記録条件のビデオが含まれているため, 記録条件を整えた原ビデオ (Original ビデオ) を初めに生成した. Original ビデオを加工処理し, 検索対象の編集ビデオとして Cropped ビデオと Rescaled ビデオの 2 種類を生成した (図 6).

4.2 検索性能

Rescaled ビデオと Cropped ビデオを長さ 30 秒 ~ 600 秒のビデオ断片をクエリとして, Original ビデオにおけるビデオ開始位置を検出する性能評価実験を行った. 例えば, 30 秒のビデオ断片をクエリとする検索では, 13,140 ($= 109.5 \times 3600 / 30$) のビデオ断片が含まれる. 性能評価は, 平均正解率と平均検出時間を計測して行った. 平均正解率は下式で求める.

表 1 テストデータ
Table 1 Test data set

Original ビデオ	720×480 画素, 6Mbps, 30fps, MPEG2 ビデオ (ダウンロードビデオの画像サイズと記録ビット レートを正規化)
Rescaled ビデオ	360×240 画素, 1Mbps, 30fps, MPEG2 ビデオ (画像サイズを縮小し記録ビットレートを下げ再 エンコード)
Cropped ビデオ	360×240 画素, 320kbps, 24fps, MPEG2 ビデオ (Original ビデオ画像フレーム中心 576×384 画素 を切出し, さらに 縮小画像にして記録ビットレ ートとフレームレートを下げ再エンコード)

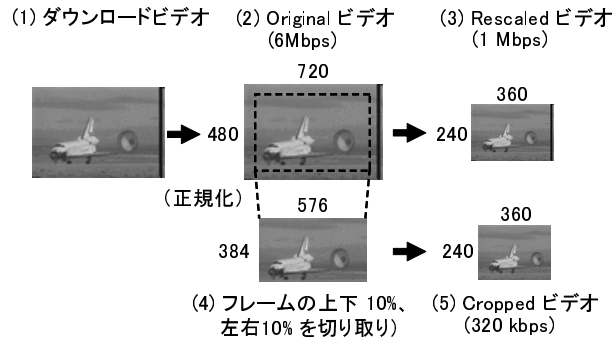


図 6 テストデータセットの生成
Fig.6 Data set preparation

$$P_{det} = 100 \times \frac{N_{ok}}{N_{all}} \quad (10)$$

ここで;

N_{ok} : 検索位置が正解した検索対象ビデオの数

N_{all} : 検索対象ビデオの総数.

図 7, 図 8 に, 平均正解率の結果を示す. "RGB" のデータは, 従来手法による実験結果
で, "A", "D" が提案する MPEG-7 video signature による実験結果である. 実験結果か
ら, Cropped ビデオ, Rescaled ビデオ両方の場合で平均正解率は改善されていることが分

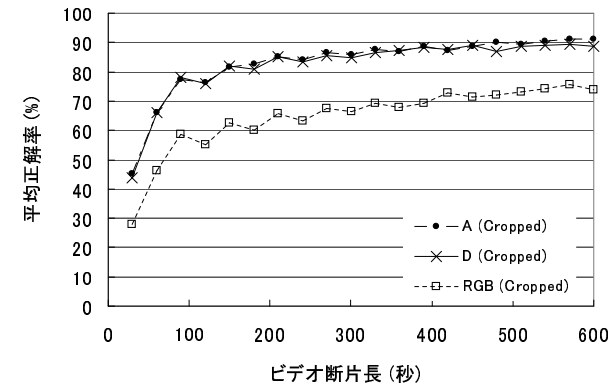


図 7 検出正解率 (Cropped ビデオ)
Fig.7 Detection accuracy (Cropped video)

かる.

図 9, 図 10 は, 平均検出時間の結果を示す. 用いた PC の仕様は, CPU は Pentium4
3GHz, メモリーは 2GB である.

表 2, 表 3 に提案手法と従来手法の検索性能を詳しく整理した. クエリとするビデオ断片
長が短いほど平均正解率の改善効果は大きいことが分かる. 検出時間は従来手法に比べて約
1/2 となっている. 実験の検出処理では, ビデオ間距離の計算ごとに最小データを記憶して
行き, 記憶されているデータより大きな値となる場合は, それ以降の計算を行わない処理を
実装している. そのため, 検索正解率が良いほど目標位置を早く検索できて検索時間が短く
なっている.

5. 考 察

5.1 提案手法に必要なメモリー量

必要なメモリー量は重要なポイントである. ここでは, 提案手法の必要なメモリー量
の見積もりを行う. PC は 64 bit システム (32 ビットポインター) を用いると仮定すると,
direct-mapped cache は 4G フレーム (すなわち 136 年分) を蓄積できる. type A signature
の場合, 各フレームのデータ量は 11 バイト (すなわち 5+4+2 バイト) なので, 必要なメ
モリー量は 44G バイトになる. type D signature の場合, 各フレームのデータ量は 15 バイ

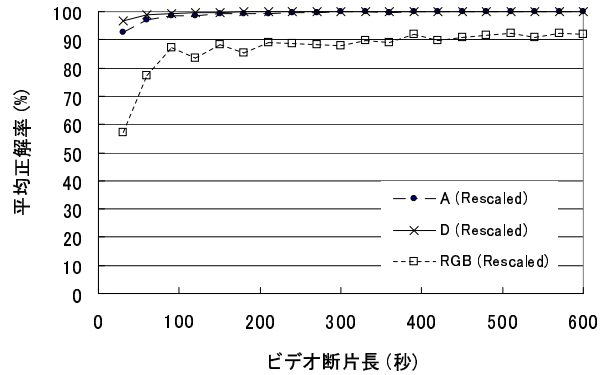


図 8 検出正解率 (Rescaled ビデオ)
Fig. 8 Detection accuracy (Rescaled video)

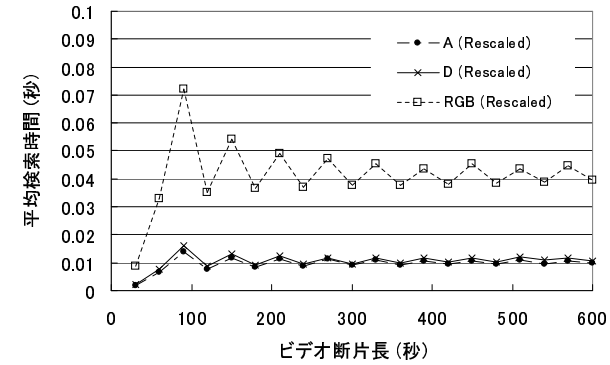


図 10 検出時間 (Rescaled ビデオ)
Fig. 10 Detection time (Rescaled video)

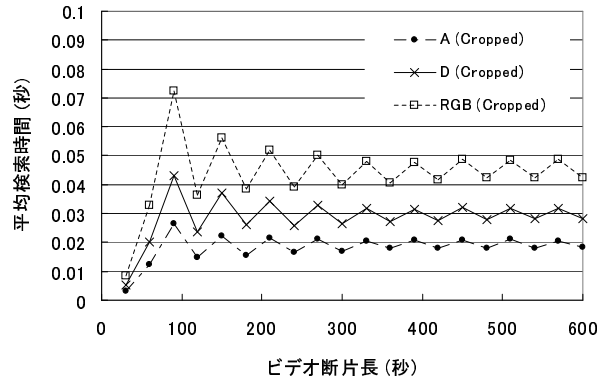


図 9 検出時間 (Cropped ビデオ)
Fig. 9 Detection time (Cropped video)

ト (すなわち 9+4+2 バイト) なので、必要なメモリ量は 60G バイトになる。

提案手法に必要なメモリー量は、現状の汎用 PC でも実用的である。このメモリー量の見積もりから、提案手法は 136 年規模の大規模データの検索にも対応できることが分かる。

表 2 検索正解率の性能比較

Table 2 Performance comparison (detection accuracy)

	30(秒)	60(秒)	300(秒)
(Cropped ビデオ)			
A type (%)	45	66	86
D type (%)	44	66	85
従来手法 (%)	28	46	66
(Rescaled ビデオ)			
A type (%)	93	97	99
D type (%)	97	99	99
従来手法 (%)	57	77	88

6. まとめと今後の課題

本論文ではビデオ検索のための MPEG-7 Video signature の direct-mapped cache への適用法の検討を行った。適用した提案手法の特徴は以下の通りである。

- (1) 画像特徴量として、MPEG-7 Video signature の一部を選択して用いる。
- (2) 編集されたビデオ検索に対応するために MPEG-7 Video signature に Tiny LSH 処理法を適用する。
- (3) MPEG-7 Video signature と direct-mapped cache を組み合わせて高速検索を行う。

表 3 検索時間の性能比較

Table 3 Performance comparison (detection time)

	30(秒)	60(秒)	300(秒)
(Cropped ビデオ)			
A type (秒)	0.003	0.012	0.017
D type (秒)	0.005	0.020	0.026
従来手法 (秒)	0.009	0.033	0.040
(Rescaled ビデオ)			
A type (秒)	0.002	0.007	0.009
D type (秒)	0.002	0.008	0.010
従来手法 (秒)	0.009	0.033	0.040

表 4 必要なメモリー量の見積もり

Table 4 Memory requirement

64 bit システム (32 ビットポイント)
4G フレーム = 60(秒)×60(分)×24(時間)×365(日)×136(年)
・ type A の場合 ;
44 GByte = (5 Byte(=20) signature × 2bit / 8bit)
+ 4 Byte フレーム ID
+ 2 Byte 蓄積カウンタ)
× 4G フレーム
・ type D の場合 ;
60 GByte = (9 Byte(=36) signature × 2bit / 8bit)
+ 4 Byte フレーム ID (Mem)
+ 2 Byte 蓄積カウンタ (Cnt))
× 4G フレーム

109.5 時間のテストデータを用いた検索の性能評価実験では、以下の結果を得た。

- (1) ビデオ長 60 秒の Cropped ビデオ と Rescaled ビデオを用いた実験では、A type, D type signature とともに、平均正解率は 66% であり、従来手法に比べて 20%性能改善できた。
- (2) ビデオ長 30 秒の Rescaled ビデオを用いた実験において、D type signature では平均正解率は 97% であり、従来手法に比べて 40%性能改善できた。Cropped ビデオよりも性能改善が大きい。
- (3) A type, D type signature とともに、検索対象ビデオの長さがどの場合でも 0.1 秒以内で検索が可能

(4) 検索速度を維持して 136 年分のビデオ検索に対応できる。

本論文で適用法の性能評価のために用いたテストデータは、記録ビットレート、フレームレート、画像サイズの変更や画像の切り取りなどの編集ビデオであった。その他の様々な変更ビデオやさらに大規模なビデオデータに対する性能評価、性能改善は今後の課題である。

参 考 文 献

- 1) Yoshida, K. and Murabayashi, N.: Tiny LSH for Content-Based Copied Video Detection, *Proceedings of the International Symposium on Applications and the Internet 2008 (SAINT 2008)*, pp.89-95 (2008).
- 2) ISO/IEC 15938-3:2002/Amd.4:2010 Multimedia content description interface – Part 3: Visual – AMENDMENT 4: Video signature tools, (Oct. 2010).
- 3) Lowe, D.G.: Distinctive Image Features from Scale-Invariant Keypoints, *International Journal of Computer Vision*, Vol.60, No.2, pp.91-110 (2004).
- 4) Bay, H., Tuytelaars, T. and Van, L.G.: SURF: Speeded Up Robust Features, *In: Computer Vision-ECCV 2006*, pp.404-417 (2006).
- 5) Liu, Z., Liu, T., Gibbon, D. and Shahraray, B.: Effective and Scalable Video Copy Detection, *Proceedings of the international conference on Multimedia information retrieval (MIR 2010)*, pp.119-128 (2010).
- 6) Douze, M., Gaidon, A., Jegou, H., Marszalek, M. and Schmid, C.: INRIA-LEAR's Video Copy Detection System, *TRECVID Workshop*, pp.119-128 (2008).
- 7) Datar, M., Indyk, P., Immorlica, N. and Mirrokni, V.: *Locality-sensitive hashing using stable distributions*, MIT Press (2006).
- 8) Yoshida, K., Adachi, F., Washio, T., Motoda, H., Homma, T., Nakashima, A., Fujikawa, H. and Yamazaki, K.: Density-based spam detector, *KDD2004*, pp.486-493 (2004).
- 9) "SDBM." <http://search.cpan.org/src/NWCLARK/perl-5.8.8/ext/SDBM-File/sdbm/README> (Reference:2010-01-25).
- 10) 村林昇, 吉田健一: Tiny LSH に基づいた大規模ビデオ検索, *信学論 (D)*, Vol.J94-D, No.8, pp.1461-1472.
- 11) "The Open Video Project." <http://www.open-video.org> (Reference:2009-10-01).