社団法人　情報処理学会　研究報告
IPSJ SIG Technical Reports

2007－ＡＶＭ－59（24）
2007／12／14

複数特徴を用いたディゾルブ，フェードを含むショット境界の高速検出

河合　吉彦†　　住吉　英樹†　　八木　伸行†

† NHK 放送技術研究所　〒157–8510 東京都世田谷区砧 1–10–11
E-mail: †{kawai.y-lk,sumiyoshi.h-di,yagi.n-iy}@nhk.or.jp

あらまし　ショット境界検出は，映像解析における最も基本的な処理のひとつであり，高い検出精度と高速な処理が求められる．本稿では，複数の特徴量に基づいてディゾルブ，フェードを含むショット境界を検出する手法を提案する．提案手法では，明らかにショット境界ではないフレームに対しては処理を省略し，可能性がある部分についてのみ様々な特徴量を算出して解析することにより，精度よく高速にショット境界を検出する．国際的な競争型評価ワークショップである TRECVID 2007 に参加し提案手法を評価したところ，平均再現率が 90.4%，平均適合率が 92.8%という良好な結果が得られた．また，約 425 分のテストデータに対する処理時間は 3 分 28 秒（MPEG1 のデコード時間を除く）となり，実時間の約 1/123 という高速な処理が実現できた．その結果，提案手法は TRECVID 2007 において最も高速な手法と評価された．

キーワード　ショット境界検出，複数特徴，カット，ディゾルブ，フェード，TRECVID

Fast detection method for shot boundary including gradual transition using multiple features

Yoshihiko KAWAI†, Hideki SUMIYOSHI†, and Nobuyuki YAGI†

† Science and Technical Research Laboratories, NHK　1–10–11 Kinuta, Setagaya–ku, Tokyo, 157–8510 Japan
E-mail: †{kawai.y-lk,sumiyoshi.h-di,yagi.n-iy}@nhk.or.jp

Abstract　Shot boundary detection is one of the most fundamental processes in video analysis, and it requires high detection accuracy and high-speed processing. This paper proposes a method for detecting shot boundary including gradual transition based on multiple features. The proposed method enables precise, high-speed detection by omitting the processing of frames that are clearly not shot boundaries, and by analyzing various features only for the parts of the video that are likely to contain shot boundaries. An evaluation on TRECVID 2007, which is an international competition and evaluation project on video retrieval and indexing, resulted in a recall rate of 90.4% and a precision rate of 92.8% on average. About 425 minutes of test data was processed in 3 minutes 28 seconds (excluding the MPEG1 decoding time), 1/123 of the real time. The proposed method was evaluated as the fastest method on TRECVID 2007.

Key words　Shot boundary detection, Multiple features, Cut, Dissolve, Fade, TRECVID

1. まえがき

ショットは映像の基本単位であり，映像をショットに分割する処理は映像解析における最初のステップである．映像をショットに分割するためには，ショットのつなぎ目であるショット境界を検出する必要がある．ショット境界は次の 2 種類に大別できる．ひとつは瞬時ショット切り替えであり，もうひとつは漸次ショット切り替えである．瞬時ショット切り替えは，前のショットから次のショットへ瞬時にショットを切り替えるものであり，カットとも呼ばれる．漸次ショット切り替えは，2 つのフレームの合成の割合を徐々に変化させるディゾルブやフェード，境界を空間的に移動するワイプ，形状や位置を 3 次元的に変化させて切り替える特殊効果（special effect）などに分類できる．

ショット境界は，近接フレーム間の類似度に基づいて検出できる．具体的には，同一のショットに含まれるフレームの類似度は高く，その間にショット切り替えが存在する場合は類似度が低くなるという特徴を利用する．フレーム間の類似度としては，色や輝度ヒストグラムの差分 [1]〜[3]，エッジの変化量 [4]，画素間の相互情報量 [5] などが提案されている．また，漸次ショット切り替えに対しては，画素値の分散 [6] や DCT 係数 [7] に基づく手法が提案されている．近年では，これらの特徴量を組み合わせることにより精度の向上を図った手法も提案されている [8]〜[11]．これらの手法は，各フレームに対して複数の特徴量を算出し，fuzzy c-means や support vector machine などの

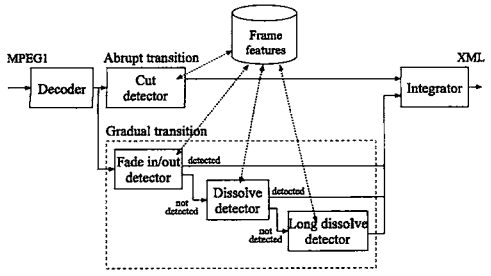


図 1　提案システムの概要

機械学習によってショット境界とそれ以外とを分類する．複数の特徴を用いる従来手法の問題点としては，各フレームに対してすべての特徴量を計算しているため，計算コストが高いという点が挙げられる．特に，複雑な特徴量を利用する場合や，多数の特徴量を利用する場合などには多くの計算時間が必要となる．番組映像に含まれるショット境界は通常，全フレームの1%以下であり，すべてのフレームに対して同一の検出処理を実施することは効率的ではないといえる．

提案手法では，明らかにショット境界でないフレームについては処理を省略し，ショット境界の可能性があるフレームについてのみ様々な特徴量を算出して詳細に解析することにより，精度よく高速にショット境界を検出することを試みる．

2. ショット境界検出

提案システムの概要を図1に示す．まず，デコーダにおいて，入力映像ファイルからフレーム画像を取得する．次に，抽出されたフレーム画像を解析しショット境界を検出する．ショット境界を漏れなく検出できるよう，瞬時切り替えと漸次切り替えの検出処理は並列に実行する．漸次切り替えについては，フェード，ディゾルブ，長いディゾルブの順に検出処理を実施する．このとき，いずれかの漸次ショット切り替えが検出された時点で残りの検出処理を実施しないようにする．また，長いディゾルブについては，検出漏れを避けるため通常のディゾルブと処理を別にしている．さらに，すべての検出処理においてフレーム特徴を共有することにより，同様の特徴量が重複して計算されることがないようにする．最後に統合部では，重なりをもつ漸次切り替えや近接したショット切り替えを統合することによって，最終的な検出結果とする．

以降では，各検出処理について詳細を説明する．

3. カット検出器

カット切り替えは，隣接フレーム間の差分に基づいて検出する．図2にカット検出の手順を示す．まず，R，G，Bの各濃度値の絶対差分和 d_{sad} によって明らかに変化のないフレームを判定する（図2（a））．d_{sad} の算出式を式（1）に示す．

$$d_{sad}(f_{i-1}, f_i) = \frac{1}{|\mathbf{F}|} \sum_{\boldsymbol{r} \in \mathbf{F}} |f_{i-1}(\boldsymbol{r}) - f_i(\boldsymbol{r})| \qquad (1)$$

$f_i(\boldsymbol{r})$ は i 番目のフレームの座標 $\boldsymbol{r}$ における画素の値を表す．また，$\mathbf{F}$ はフレーム全体の画素を表し，$|\mathbf{F}|$ は画素の総数を表す．実際にはR，G，Bの各濃度値について差分和を求めるが，式が煩雑になるため，ここでは簡略化した式を示している．算出された d_{sad} が閾値 T_{sad} より小さい場合はショット境界ではないと判断し処理を終了する．

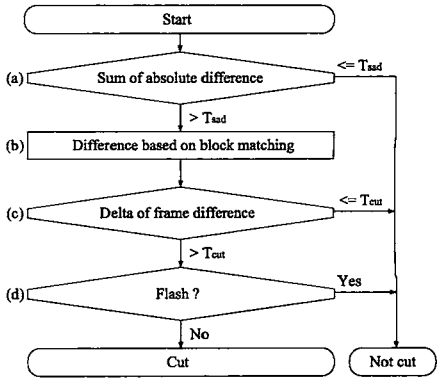


図 2　カット検出の手順

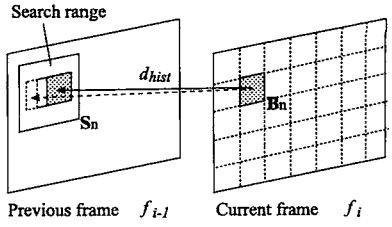


図 3　ブロックマッチング

d_{sad} が閾値以上となった場合は，より精度の高いフレーム間差分を算出しなおす（図2（b））．ここでは，被写体やカメラの動きが考慮できるブロックマッチングに基づくフレーム間差分 d_{bm} を利用する．d_{bm} については後節で詳細を説明する．

次に，算出されたフレーム間差分 d_{bm} に基づいてカット点を判定する（図2（c））．フレーム間差分はカメラや被写体の激しい動きによっても大きく変化し誤検出の要因となるため，提案手法では，フレーム間差分の増加量が閾値以上となった場合にショット境界と判定する．式（2）に判定条件を示す．条件を満たす場合，フレーム $i-1$ と i の間にカットが存在すると判定する．閾値以下の場合はここで処理を終了する．

$$d_{bm}(f_{i-1}, f_i) - d_{bm}(f_{i-2}, f_{i-1}) > T_{cut} \qquad (2)$$

最後に，カットと判定されたフレームに対して，カメラのフラッシュによる誤検出ではないことを確認し，出力結果とする（図2（d））．以下，図2（b），（d）の処理について説明する．

3.1 ブロックマッチングに基づくフレーム間差分

ブロックマッチングの概要を図3に示す．ブロックマッチングでは，現フレームの各ブロック領域に対し，ブロック間のコストが最小となる位置を前フレームにおいて探索する．このコストが閾値以上であったブロックの数を総数で正規化し，フレーム間差分の値とする．式（3）に d_{bm} の算出式を示す．

$$d_{bm}(f_{i-1}, f_i) = \frac{1}{N} \sum_{n=1}^{N} 1 \quad if\ \lambda_n(f_{i-1}, f_i) > T_\lambda \qquad (3)$$

N はブロック領域の総数，λ_n は n 番目のブロックのブロック間コストの最小値を表す．また，T_λ は閾値である．動きベクトルの探索を目的とした場合，ブロック間コストとしては，一般的に輝度の差分二乗和や絶対差分和などが利用されるが，本手

法ではカメラ動きに堅牢な輝度ヒストグラムの絶対差分和を利用する．式（4）に λ_n の算出式を示す．

$$\lambda_n(f_{i-1}, f_i) = \min_{\boldsymbol{v}\in\mathbf{S}_n} \{d_{hist}(f_{i-1}(\boldsymbol{r}+\boldsymbol{v}), f_i(\boldsymbol{r})), \boldsymbol{r} \in \mathbf{B}_n\} \quad (4)$$

ここで，$\mathbf{S}_n$ は探索範囲を表し，$\mathbf{B}_n$ は n 番目のブロック領域に含まれる画素を表す．d_{hist} は輝度ヒストグラムの絶対差分和である．d_{hist} は各ブロック領域について，輝度値の頻度ヒストグラムを作成し，その絶対差分和を計算することで算出する．式（5）にフレーム間のヒストグラム差分和の算出式を示す．

$$d_{hist}(f_{i-1}, f_i) = \frac{1}{N_c}\sum_{c=1}^{N_c} |H_{i-1}(c) - H_i(c)| \quad (5)$$

N_c は輝度ヒストグラムのビンの総数，$H_i(c)$ は c 番目のビンに含まれるフレーム i の画素の総数を表す．

ブロックマッチングにおいてもっとも計算コストが高い処理は，式（4）におけるブロック位置 $\boldsymbol{v}$ の探索処理である．提案手法では，この処理における計算量を削減することによりブロックマッチングの高速化を図る．ブロックの探索アルゴリズムについては，すでに様々な高速化手法が提案されているが，本手法ではその中でも特に高速化が期待できる Dual cross search (DCS) [12] を採用する．さらに，提案手法では通常の DCS に加えて，閾値 T_λ による探索の打ち切りを実施することにより，更なる高速化を図る．予備実験では全ての範囲を探索する全探索に比べマッチング回数を 1/600 程度まで削減することができることが確認できた．

3.2 フラッシュの判定

フラッシュによる急激な輝度の変動は，カットと誤検出される場合があるため，フラッシュの判定を実施する．フラッシュでは，最初の数フレームの間だけ輝度が上昇し，その後，元に戻るといった変化をする．そこでフレーム i が判定対象である場合，まず続く N_f フレームの最小輝度値から構成される画像 f'_i を合成する．式（6）に示す．

$$f'_i(\boldsymbol{r}) = min\left(f_i(\boldsymbol{r}), f_{i+1}(\boldsymbol{r}), .., f_{i+N_f}(\boldsymbol{r})\right) \quad (6)$$

f_i がフラッシュによる輝度上昇であった場合，f_{i-1} と f'_i は類似した画像特徴を持つと考えられる．そこで，フレーム間差分 $d(f_{i-1}, f'_i)$ が閾値 T_{cut} 以下の場合はフラッシュによる誤検出と判定する．フレーム間差分の算出式を式（7）に示す．

$$d(f_{i-1}, f_i) = \begin{cases} 0 & if\ d_{sad}(f_{i-1}, f_i) < T_{sad} \\ d_{bm}(f_{i-1}, f_i) & otherwise \end{cases} \quad (7)$$

カット判定と同様，絶対差分和が閾値以下の場合は，計算コストの高いブロックマッチングを実施しないようにする．

4. フェード検出器

フェードインは，黒フレームから徐々に輝度が上昇していく切り替えであり，フェードアウトは反対に黒フレームに向かって徐々に輝度が減少していく切り替えである．フェードはこの黒フレームを基本にして検出を試みる．検出手順を図 4 に示す．

まず，現フレームが黒フレームかどうかを判定する（図 4 (a)）．黒フレームの判定は，フレーム平均輝度 $\overline{f_i}$ と，輝度値が閾値 T_{black} 以下である画素の総数 $black(f_i)$ によって判定する．

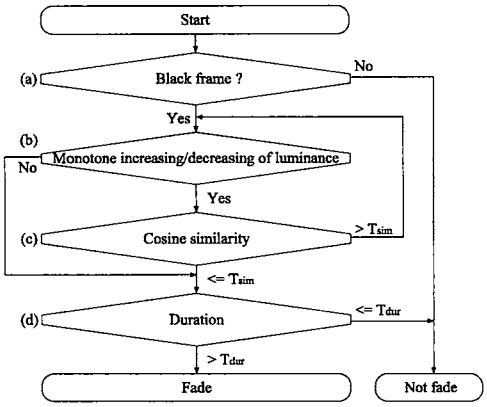


図 4 フェード検出の手順

黒フレームであった場合は，現フレームを終点とするフェードアウトではないかを判定する．あわせて，現フレームを始点とするフェードインではないかを判定する．現フレームが黒フレームでない場合はここで処理を終了する．

フェードイン，フェードアウトの区間は，複数フレームに渡って連続する輝度の単調増加，単調減少に基づいて決定する（図 4 (b)）．単調増加は式（8），単調減少は式（9）による値が閾値 T_{fade} 以上であるかによって判定する．

$$inc(f_{i-1}, f_i) = \frac{1}{|\mathbf{F}|}\sum_{\boldsymbol{r}\in\mathbf{F}} 1 \quad if\ f_i(\boldsymbol{r}) > f_{i-1}(\boldsymbol{r}) \quad (8)$$

$$dec(f_{i-1}, f_i) = \frac{1}{|\mathbf{F}|}\sum_{\boldsymbol{r}\in\mathbf{F}} 1 \quad if\ f_i(\boldsymbol{r}) < f_{i-1}(\boldsymbol{r}) \quad (9)$$

上記の条件は，カメラ動きなどによって輝度の低い被写体がフレーム内に現れる場合などにも成り立ち，誤検出となる場合がある．そこで，フェード区間内における隣接フレームの類似度を算出し，それが閾値 T_{sim} 以下となった場合は誤検出と判定する（図 4 (c)）．式（10）に類似度の算出式を示す．

$$sim(f_{i-1}, f_i) = \frac{\sum_{\boldsymbol{r}\in\mathbf{F}} f_{i-1}(\boldsymbol{r})\cdot f_i(\boldsymbol{r})}{\sqrt{\sum_{\boldsymbol{r}\in\mathbf{F}} f_{i-1}^2(\boldsymbol{r}) \sum_{\boldsymbol{r}\in\mathbf{F}} f_i^2(\boldsymbol{r})}} \quad (10)$$

これは，フレーム画像の余弦による類似度であり，フレーム全体の輝度変動に影響されずに類似性を判定することができる．

最後に，検出されたフェード区間の長さが閾値 T_{dur} 以上であるかを確認して検出結果とする（図 4 (d)）．

5. ディゾルブ検出器

ディゾルブ切り替えは式（11）にようにモデル化できる．

$$f_i(\boldsymbol{r}) = (1-\alpha)\cdot f_b(\boldsymbol{r}) + \alpha\cdot f_e(\boldsymbol{r}),\ \ 0 \leqq \alpha \leqq 1 \quad (11)$$

ここで，f_b はディゾルブの開始フレーム，f_e は終了フレームを表し，α は 0 から 1 まで変化する．提案手法では，このモデルを基本とした 2 種類の特徴量に基づいてディゾルブを検出する．

ひとつは，輝度の単調変化に基づく特徴量である．式（11）にしたがってショットが変化するとき，各画素の値は単調に増加，もしくは単調に減少する．そこで，前後のフレームとの比

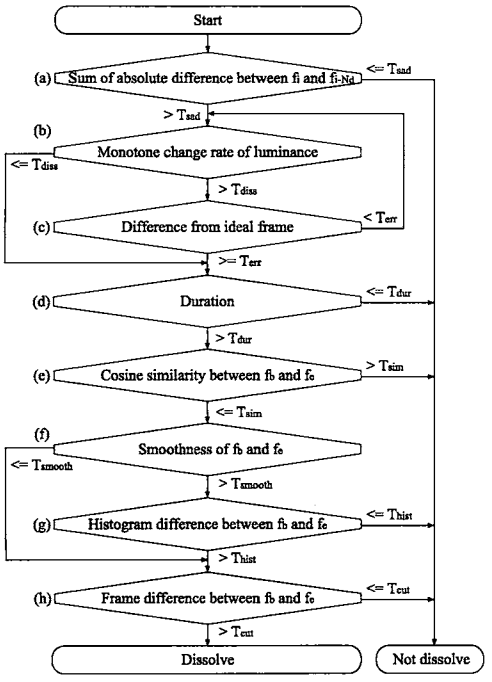


図 5　ディゾルブ検出の手順

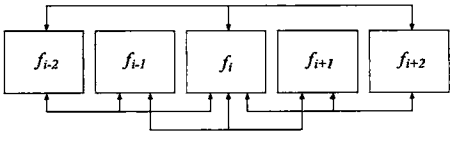


図 6　輝度の単調変化に基づくディゾルブ検出

較から，単調変化している画素の総数を求め，特徴量とする．式 (12) に算出式を示す．

$$diss(f_{i-1}, f_i, f_{i+1}) = \frac{1}{|\mathbf{F}|}\sum_{r\in\mathbf{F}} mono(f_{i-1}(r), f_i(r), f_{i+1}(r)) \tag{12}$$

$$mono(v1, v2, v3) = 1 \quad if \ \ (v1 > v2 > v3) \ \ or \ \ (v1 < v2 < v3) \tag{13}$$

もうひとつの特徴量は，理想的なディゾルブとの差分である．理想的なディゾルブ区間では，式 (14) に示す関係が成り立つ．

$$f_i(r) = \frac{f_{i-1}(r) + f_{i+1}(r)}{2} \tag{14}$$

そこで，式 (15) に示す差分を算出し，これをもうひとつの特徴量として利用する．

$$err(f_{i-1}, f_i, f_{i+1}) = \frac{1}{|\mathbf{F}|}\sum_{r\in\mathbf{F}} \left| \frac{f_{i-1}(r) + f_{i+1}(r)}{2} - f_i(r) \right| \tag{15}$$

図 5 にディゾルブの検出手順を示す．まず，現フレームと N_d フレーム前のフレームとの絶対差分和 $d_{sad}(f_{i-N_d}, f_i)$ を算出し，この間にショット切り替えが含まれているかを判定する（図 5 (a)）．d_{sad} が閾値 T_{sad} 以下の場合はここで処理を終了する．d_{sad} が閾値以上の場合は，前述の 2 種類の特徴量に基づいてディゾルブ区間を検出する．

ディゾルブ区間の検出において，式 (14) (15) のように，連続した 3 フレームのみを考慮して特徴量を算出した場合，カメラや被写体の動きによる誤検出が多く発生してしまう．そこで提案手法では，図 6 に示すように，連続した 5 フレームの 4 箇所から特徴量を算出し利用する．具体的な判定条件を式 (16) (17) に示す．これらの条件が連続的に成り立つ区間をディゾルブ区間とする（図 5 (b), (c)）．

$$diss(f_{i-2}, f_{i-1}, f_i) + diss(f_{i-1}, f_i, f_{i+1}) + diss(f_i, f_{i+1}, f_{i+2}) + diss(f_{i-2}, f_i, f_{i+2}) > T_{diss} \tag{16}$$

$$err(f_{i-2}, f_{i-1}, f_i) + err(f_{i-1}, f_i, f_{i+1}) + err(f_i, f_{i+1}, f_{i+2}) + err(f_{i-2}, f_i, f_{i+2}) < T_{err} \tag{17}$$

続いて，検出されたディゾルブ区間を検証する．まず，検出された区間の長さが十分であるか確認する（図 5 (d)）．区間長が閾値 T_{dur} 以下である場合は検出処理を終了する．

次に，輝度変動による誤検出を判定する．アイリス変動などによって輝度値が時間的に滑らかに変化する場合，ディゾルブと同様に式 (16), (17) の条件が成り立つ場合がある．そこで，検出されたディゾルブ区間の開始フレーム f_b と終了フレーム f_e の類似度を算出し，それが閾値以上の場合には，誤検出と判定する（図 5 (e)）．類似度の算出には，輝度変動に頑健な式 (10) を利用する．

また，空間的に輝度値が滑らかに変化するような画像が，一定の方向に移動した場合にも式 (16), (17) の条件が成り立ち，誤検出となる場合がある．そこで，式 (18) によって f_b および f_e が滑らかな背景を持つ画像でないかを判定する（図 5 (f)）．

$$smooth(f_i) = \frac{1}{4|\mathbf{F}|}\sum_{(x,y)\in\mathbf{F}} \{s_1(f_i) + s_2(f_i) + s_3(f_i) + s_4(f_i)\} \tag{18}$$

$$s_1(f_i) = mono(f_i(x-1,y), f_i(x,y), f_i(x+1,y)) \tag{19}$$

$$s_2(f_i) = mono(f_i(x,y-1), f_i(x,y), f_i(x,y+1)) \tag{20}$$

$$s_3(f_i) = mono(f_i(x-1,y-1), f_i(x,y), f_i(x+1,y+1)) \tag{21}$$

$$s_4(f_i) = mono(f_i(x-1,y+1), f_i(x,y), f_i(x+1,y-1)) \tag{22}$$

$smooth(f_b)$ もしくは $smooth(f_e)$ が閾値 T_{smooth} 以上であった場合は，さらに式 (5) によって f_b と f_e のヒストグラム差分を算出し，これが閾値 T_{hist} 以下となった場合には誤検出と判定する（図 5 (g)）．

最後に，式 (7) によって f_b と f_e のフレーム間差分を計算し，カットと同様に，差分の変動量が閾値 T_{cut} 以上となればディゾルブと判定する（図 5 (h)）．

6. ロングディゾルブ検出器

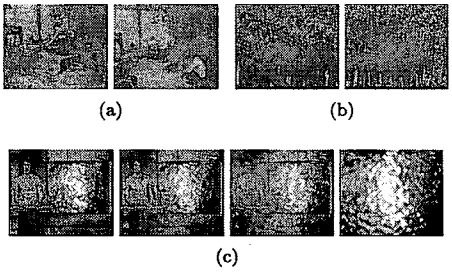
切り替えの区間の長いディゾルブでは，隣接フレームにおいて画素値が変化しない場合があるため，前節による手法では未検出となることがある．そこで，長いディゾルブに対しては，式 (16) (17) におけるフレーム間隔を N_l 倍に拡大することによって検出漏れを防ぐ．他の処理については，通常のディゾルブと同様とする．

表 2　ショット境界検出の評価結果

Run	Overall			Cut			Gradual		
	R	P	F	R	P	F	R	P	F
4	0.937	0.859	0.896	0.961	0.939	0.968	0.680	0.369	0.478
1	0.921	0.913	0.917	0.946	0.951	0.948	0.650	0.554	0.598
2	0.905	0.944	0.924	0.933	0.965	0.949	0.607	0.691	0.646
3	0.888	0.960	0.923	0.916	0.975	0.945	0.578	0.768	0.660
5	0.867	0.966	0.914	0.896	0.980	0.936	0.544	0.767	0.637
Avg.	0.904	0.928	0.915	0.930	0.962	0.946	0.612	0.630	0.604

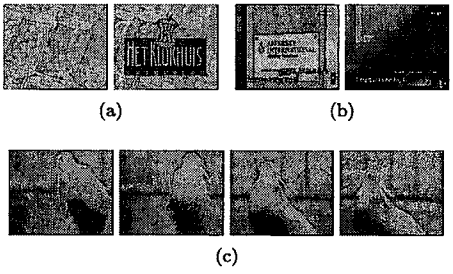
表 1　閾値の設定

Threshold	Run				
	4	1	2	3	5
T_{sad}	15				
T_{cut}	30	35	40	45	50
T_{λ}	30				
N_f	3				
T_{black}	13				
T_{fade}	55				
T_{dur}	5				
T_{sim}	90				
N_d	10				
T_{diss}	30	35	40	45	50
T_{err}	70				
T_{smooth}	40				
T_{hist}	40				
N_l	2				



(a)　(b)　(c)

図 7　未検出の例



(a)　(b)　(c)

図 8　誤検出の例

7. 実　　験

TRECVID 2007 に参加し，提案手法の性能を評価した．TRECVID は米国国立標準技術研究所によって主催されている映像検索や索引付けに関する国際的な競争型評価ワークショップであり [13], [14]，共通のテストデータを用いた実験によって手法を評価する．TRECVID 2007 のテストデータは，オランダで放送された番組映像でドキュメンタリが主となっている．データは合計約 425 分であり，MPEG1 形式で圧縮されている．

提案手法における閾値は，日本で放送されている様々のジャンル（ドラマ，スポーツ，ドキュメンタリなど）の番組に対する実験から組み合わせを決定し，これを試行 1（Run 1）で使用した．この組み合わせを基本にして閾値を前後に変化させ，合計で 5 回の試行を実施した．実験に使用した閾値を表 1 に示す．また，ブロックマッチングにおけるフレームの分割数は横縦 10 × 10 の合計 100 分割とし，ブロック間コストの算出におけるヒストグラムのビン数は 16 とした．フレームサイズは横縦 176 × 144 ピクセルとした．

7.1 評 価 結 果

評価結果を表 2 に示す．表中の R は再現率，P は適合率，F は F 値を表す．また，Avg. の行は全試行の平均値を表す．全体（overall）の結果としては，平均再現率が 90.4%，平均適合率が 92.8%となり非常に高い精度となった．カット（cut）のみについてはさらに高い検出精度となっている．漸次切り替え（gradual）については再現率，適合率は 60%程度となった．

誤検出，未検出となった箇所について調査した．カットについては，白黒のフィルム映像において，再現率が約 50%と非常に低くなっていた．ショット境界におけるフレーム間の変化が小さいために未検出が多発したものと思われる．図 7 (a) に例を示す．その他には，図 7 (b) のような類似したシーン間での切り替えや，夜などの暗いシーンにおいて未検出となる場合があった．ブロックマッチングにおけるヒストグラムのカテゴリ分けや閾値を動的に変化させるなどの検討が必要である．誤検出については，サイズの大きなスーパーの出現（図 8 (a)）や，カメラの前を物体が横切る場合（図 8 (b)）などがあった．

漸次切り替えについては，切り替え区間内におけるカメラや被写体の動きによる未検出が多くあった．式 (11) のディゾルブのモデルは，フレームが静止していることを前提としている．そのため，カメラ動きなどによってフレーム画像が変動する場合には，ディゾルブ検出のための条件が成り立たず未検出となってしまう．図 7 (c) の例では，水面の変化の影響で未検出となった．フレームに動きがある場合への対応が必要である．また，本手法で対象としていないワイプなどの切り替えは未検出となった．一方，誤検出については，被写体やカメラの動きによる場合があった．図 8 (c) に例を示す．エッジ特徴や周波数特徴などを利用することによってディゾルブとカメラ動きとを区別するなどの処理が必要と思われる．

7.2 処 理 時 間

実験に用いた計算機は，CPU が Intel Core 2 Duo E6600 2.40GHz で，RAM は 2GB である．合計約 425 分のテストデータに対する処理時間を表 3 に示す．表中の Decode の列は MPEG1 映像をデコードしてフレーム画像を取得するのに要し

表 3 処理時間（秒）

Run	Total	Decode	Segmentation
4	1697.4	1482.4	215.0
1	1684.6	1475.5	209.1
2	1683.5	1477.3	206.2
3	1684.0	1478.5	205.5
5	1682.8	1478.6	204.2
Avg.	1686.5 (1/15)	1478.5 (1/17)	208.0 (1/123)

表 4 他の参加チームとの比較結果

#	Team	F-measure (Overall)	Proc. time (Total: sec.)
1	B	0.946	7325.5 [4.3]
2	K	0.941	12974.7 [7.7]
3	M	0.923	5367.0 [3.2]
4	Our team	0.915	1686.5 [1.0]
5	D	0.897	17540.0 [10.4]
6	I	0.885	5517.9 [3.3]
7	C	0.865	4157.9 [2.5]
8	G	0.846	3615.1 [2.1]
9	A	0.834	10586.3 [6.3]
10	E	0.781	611200.0 [362.4]

た処理時間を，Segmentation の列はショット境界を検出するのに要した処理時間を表す．Total はそれらの合計である．また，Avg. の行には 5 回の試行の平均処理時間と，実時間に対する割合を示す．

全体の処理時間は平均 1686.5 秒（約 28 分）であり，実時間の約 1/15 という高速な処理が実現できた．明らかにショット境界ではないフレームをなるべく少ない計算量で判別することによって，計算量を削減することができた．また MPEG1 のデコード時間を除いた処理時間は，わずか 208 秒であり実時間の約 1/123 となった．処理時間のうち約 88%は映像のデコード処理によって占められているため，これらを最適化することにより更なる高速化が期待できる．

7.3 他の TRECVID 参加チームとの性能比較

他の TRECVID 参加チームとの比較結果を表 4 に示す．表 4 では，検出精度が上位の 10 チームについて、精度が高いものから順に並べている．検出精度の比較には，全体結果（overall）における F 値の平均値を用いた．比較の結果，提案手法の精度は全参加チームの中で 4 位となった。上位のチームとの差もわずかであり非常に高い検出精度となった．また，表 4 の "proc. time"の列に各手法の平均処理時間（秒）と，提案手法を 1 とした場合の比率を角括弧内に示す．処理時間の比較では，提案手法は全参加チームの中で最も高速な手法となった．提案手法の処理時間は，他手法の 1/2〜1/360 程度となっている．また，検出精度が上位の 3 つの手法については，提案手法に比べ検出精度が 1〜3%高いものの，処理時間は 3〜8 倍程度となっている．処理時間と検出精度を統合的に判断した場合，提案手法は非常に有効な手法といえる．

8. あとがき

本稿では，複数の特徴に基づいてショット境界を検出する手法を提案した．提案手法では，明らかにショット境界ではないフレームについては処理を省略し，可能性がある部分についてのみ詳細な特徴量を算出することにより，正確かつ高速にショット境界を検出した．国際的な競争型ワークショップである TRECVID 2007 に参加し提案手法を評価したところ，平均再現率が 90.4%，平均適合率が 92.8%という良好な結果が得られた．また，約 425 分のテストデータに対する処理時間は 3 分 28 秒（MPEG1 のデコード時間を除く）となり，実時間の約 1/123 という高速な処理が実現できた．その結果，提案手法は TRECVID 2007 において最も高速な手法と評価された．

今後の検討課題としては，各処理における閾値の決定方法が挙げられる．学習データから最適な組み合わせを決定するような手法について検討したい．また，白黒映像などの変動の小さい映像に対する対応策や，切り替え区間においてカメラなどの動きがあるディゾルブの検出手法などについても検討が必要と考える．さらに，今回，本手法では対象としなかったワイプや CG を用いた特殊な切り替えなど，その他の切り替えの検出手法についても検討を進めたい．

文 献

[1] H.Zhang and S.S.A.Kankanhalli, "Automatic partitioning of full-motion video," ACM Multimedia Systems, vol.1, no.1, pp.10–28 (1993).

[2] J.S.Boreczky and L.A.Rowe, "Comparison of video shot boundary detection techniques," in Proc. SPIE, vol.2664, pp.170–179 (1996).

[3] B.T.Truong, C.Dorai and S.Venkatesh, "New enhancements to cut, fade, and dissolve detection processes in video segmentation," in Proc. ACM Multimedia, pp.219–227 (2000).

[4] R.Zabih, J.Miller and K.Mai, "A feature-based algorithm for detecting and classifying production effects," Multimedia Systems, vol.7, no.2, pp.119–128 (1999).

[5] Z.Cernekova, I.Pitas and C.Nikou, "Information Theory-Based Shot Cut/Fade Detection and Video Summarization," IEEE Trans. Circuits and Systems for Video Tech., vol.16, no.1, pp.82–91 (2006).

[6] J.Nam and A.H.Tewfik, "Detection of Gradual Transitions in Video Sequences Using B-Spline Interpolation," IEEE Trans. Multimedia, vol.7, no.4, pp.667–679 (2005).

[7] R.A.Joyce and B.Liu, "Temporal Segmentation of Video Using Frame and Histogram Space," IEEE Trans. Multimedia, vol.8, no.1, pp.130–140 (2006).

[8] X.Gao and X.Tang, "Unsupervised Video-Shot Segmentation and Model-Free Anchorperson Detection for News Video Story Parsing," IEEE Trans. Circuits and Systems for Video Technology, vol.12, no.9, pp.765-776 (2002).

[9] C-W.Ngo, "A robust dissolve detector by support vector machine," in Proc. ACM Int. Conf. Multimedia, pp.283–286 (2003).

[10] H.Feng, W.Fang, S.Liu and Y.Fang, "A New General Framework for Shot Boundary Detection Based on SVM," in Proc. IEEE ICNN&B, vol.2, pp.1112–1117 (2005).

[11] K.Matsumoto, M.Naito, K.Hoashi and F.Sugaya, "SVM-Based Shot Boundary Detection with a Novel Feature," in Proc. IEEE Int. Conf. Multimedia and Expo, pp.1837–1840 (2006).

[12] X.-Q.Banh and Y.-P.Tan, "Adaptive Dual-Cross Search Algorithm for Block-Matching Motion Estimation," IEEE Trans. Consumer Electronics, vol.50, no.2, pp.766–775 (2004).

[13] A.F.Smeaton, P.Over and W.Kraaij, "Evaluation campaigns and TRECVid," in Proc. ACM Int. Workshop on Multimedia Information Retrieval, pp.321–330 (2006).

[14] http://www-nlpir.nist.gov/projects/trecvid/.