

平成 20 年度卒業論文

論文題目

YouTube への違法な投稿を防止するための
電子透かし

神奈川大学 工学部 電気電子情報工学科
学籍番号 200502718
4 年 A 組 樋口 隆

指導担当者 木下宏揚 教授

目次

1	序論	3
2	基礎知識	7
2.1	YouTube	7
2.1.1	YouTube について	7
2.1.2	投稿推奨形式	8
2.1.3	動画アップロード	9
2.2	YouTube の問題	11
2.3	既存のシステム	12
2.3.1	画像への基本挿入方法	12
2.3.2	画像の周波数特性を利用した透かし	13
2.3.3	各社提案違法動画検出方法	15
2.4	H.264	15
2.4.1	画像の周波数が、高い低いとは	17
2.4.2	GOP	19
2.4.3	SH	19
2.4.4	GOP レイア	19
2.4.5	スライス・レイヤ	19
2.4.6	マクロブロック・レイヤ	19
2.4.7	ブロック・レイヤ	20
2.5	画面内予測符号化	20
3	提案方式	23
3.1	埋め込み方式	23
3.2	取り出し方法	25
4	結論	26
5	謝辞	27
6	参考文献	28
7	質疑応答	30

図目次

1.1	透かしのイメージ	5
2.1	ストリーミング型動画配信	7
2.2	320 × 240 ピクセル	10
2.3	640 × 480 ピクセル	10
2.4	960 × 720 ピクセル	10
2.5	画素と周波数	17
2.6	エンコーダの構成	18
2.7	圧縮データの6階層構成	18
2.8	4 × 4のブロックの予測および符号化順序	21
2.9	H.264の符号化で参照するブロック	21
2.10	4 × 4画素の画面内予測	22
3.1	透かし埋め込みブロック	23
3.2	予測モード選択	24
3.3	提案予測モード選択	24

表目次

2.1	投稿動画形式	9
2.2	アスキーコードを用いた透かし	12
2.3	アスキーコード表	12
2.4	MPEG-4とH.264の画面内予測符号化の違い	20

1 序論

インターネット利用人口の大幅増加、DVDの普及、電子メディアによる情報流通量は著しく増加している。このような環境の中で行き交う情報の多くがデジタル情報である。デジタル情報は、複写、保存、加工が非常に容易に行えるという特徴を持っており、インターネットに代表されるネットワーク技術の発展により、これらのデータを同時に複数のユーザに配布することが容易になった。そのため、デジタル化されたコンテンツが著作権者の思いに反した利用をされる事が起こり、著作権保護という新たな問題が浮上したことにより、透かし技術が提案された。

違法な複製や配布への対策として、コンテンツに著作権侵害に対する警告文や出版元のロゴやお札にある透かしのように可視的なマーキングを施す方法が昔からあるが、この手法ではコンテンツの品質劣化につながる上、マーキングの除去が容易である。そのため、非可視的なマーキングによる著作権保護の手段として、電子透かしが登場したのである [14]。埋め込むコンテンツにはテキスト、音声、画像、動画、プログラムなどがあるが、作者名コピー回数など著作権関連情報を埋め込むことが主となっている。見た目わからない電子透かしは、検出ソフトを用いることで埋め込まれた情報を取り出すことができる。埋め込むコンテンツの種類として

・ 静止画用電子透かし

著作権保護用途 … 静止画デジタルアーカイブシステム

印刷物向け … 社内文書セキュリティ、商用雑誌権利保護

・ 音声用電子透かし

著作権保護用途 … サンプル用音楽 CD 向け流通管理

… 音楽配信用権利保護技術

・ 動画用電子透かし

著作権保護用途 … DVD・VTR 向け流通管理

… 動画配信、アーカイブシステム向け管理システム

また電子透かし手法を用途により以下のように分類できる。

・不可視・高耐性型

埋め込むデータを少ないという欠点があるが、少なくする代わりに、加工、圧縮などの処理に対して、耐性が高く電子透かし情報の除去が困難であるという利点がある。

・不可視・低耐性型

耐性を重視せずに、多くのデータを埋め込みたい場合に用いる。耐性が不必要な分、画質への影響も少なくてすむ。

・可視・可逆型

所有権を明示的にコンテンツ上に表示する電子透かしである。透かしを埋め込んだコンテンツの再利用ができるように、電子透かしを除去することができる。電子透かしを除去するには、特別なプログラムと鍵が必要である。これにより、コンテンツの所有者は、安心してコンテンツをインターネット上で公開したりすることができる。

・可視・非可逆型

所定のコンテンツの所有権をはっきり利用者に示す際に用いる。この型は埋め込んだ透かしデータは取り除くことが出来ない。不正な再利用や販売を防止する目的を持ち、コンテンツに所有者の名前やロゴマークを付加する。 [8, 15, 16]

・電子透かしに必要とされる条件を示す。

- (1) 埋め込みによる品質の低下を抑える
- (2) 付加する情報量が少ない
- (3) 加工によって透かしが消えない
- (4) 非可視的である



図 1.1: 透かしのイメージ

著作権侵害の問題として YouTube が挙げられる。様々な動画が投稿される YouTube だが、TV や映画などの著作物を著作権者の許可を得ずに、不特定多数に対して著作物を公開する事は、著作権の侵害に当たる。「著作物を公開」とは、YouTube に動画を投稿することも該当する。実際、動画ファイルを無制限にしかも無料でアップロード出来る事や、著作権侵害に当たる違法な投稿が大量に見られ、ドラマ、音楽 PV などを簡単に無料で見れることから、YouTube は世に広まったといえる。

また、投稿者によって映像、音楽を断片的につなぎ合わせた、コラージュ作品などは、投稿者の手が加わっているためオリジナル作品と言えるが、元の作品には著作権者がいて無断で活用しているため、これも著作権侵害に当たる。このようにどこまでが OK か NG という線引きが難しく、著作権者側も宣伝してくれていると黙認するパターンが多い。

YouTube に投稿された動画は自動的に H.263 と H.264 というフォーマットにエンコードされる。その内、H.264 は圧縮率の高さが優れており、フォーマットとしては、これから特に活躍していくと思われる。そこで本論文では、違法使用されたくない動画に電子透かしを埋め込むことで、YouTube に投稿される違法性のある動画を検出しやすくするための H.264 に特化した透かし技術を提案する。使うのは、4 × 4 画面内予測符号化の部分を用い、透かしを埋め込む。

以下

2章では YouTube における基本的構造、問題、既存のシステム、H.264 について

3章では H.264 に特化した電子透かしの提案方式について

4章ではまとめ

5章では謝辞

6章では参考文献

を記す。

2 基礎知識

2.1 YouTube

2.1.1 YouTube について

YouTube は、利用者が 1000 万人を超える、大変大きな動画投稿サイトである。[17] アカウントを登録することで、誰でも動画を投稿することができる。YouTube は、これまでのストリーミング配信で主流だった、Windows Media Player や RealPlayer ではなく、Flash 形式で動画を再生している。ストリーミング配信は、ネットワーク帯域に応じたデータを、逐時クライアント端末でバッファリングしながらデータをローカルディスクにファイルとして残さずに配信する仕組みである。そのためデータを受信してからリアルタイムに近い形で動画を再生できる。

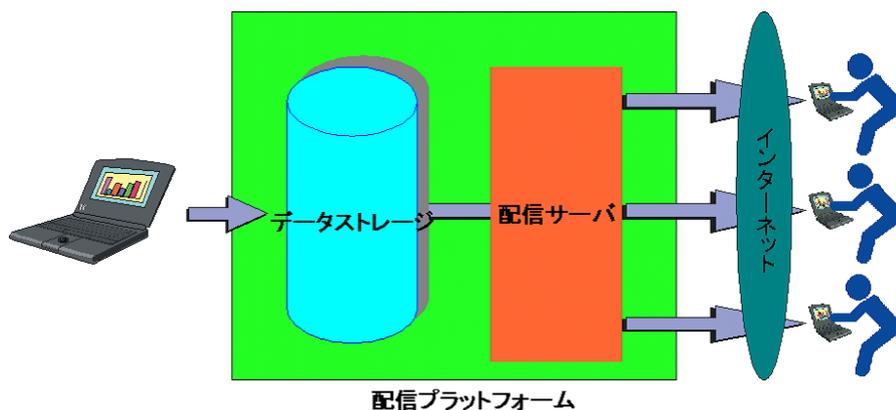


図 2.1: ストリーミング型動画配信

動画ファイルには、AVI や MPEG など多様な形式があるが、YouTube のサイトに動画をアップロードすると、これらのファイルを自動的に Flash 形式に変換するようになっている。現在の一般的なブラウザは Flash に対応しているため、観覧者は手軽に YouTube 動画を観覧できる。さらに、Flash 形式により、通常のストリーミング動画に比べファイルサイズを小さくすることができ、サーバや回線の配信コストも低減することになり、観覧する際も短時間ですべての動画を読み込むことが可能となった。

YouTube に投稿できる動画形式として、映画など著作権のある動画の投稿を防止するために、長さが 10 分以内で容量が 1G 以内の動画しか投稿することができない、

という制限がある。現在、投稿に関して、どのような動画も違法性に関わらずとりあえず全てアップロードできるようになっている。ただし、違法動画が、YouTube上に投稿された場合の違法動画削除方法は、公序良俗に反する動画、暴力的な動画、人を不快にさせる動画などは無条件に削除し、著作権を侵害する動画は、ユーザや著作権者から指摘を受けた場合に違法性を判定し、違法性があると判断した場合、動画を削除をしていくという手段をとっている。また、動画を投稿するユーザは、事前に YouTube アカウントを取得しているため、違法動画を投稿した場合は、削除するだけだが、違法動画を投稿し続けるユーザへの対処として、アカウントを停止、もしくは削除をするという手段もとっている。アカウントを停止、削除されたユーザは新しいアカウントの作成や、YouTube のコミュニティ機能へのアクセスが禁じられる。

2.1.2 投稿推奨形式

YouTube では

- ・動画のフォーマット … MPEG-4
- ・解像度 … 640 × 480 ピクセル
- ・最大ファイルサイズ … 1GB
- ・長さ … 最長 10 分
- ・フレームレート … 30fps

という形式が推奨されている。 [10]

フレームレートは、単位時間当たり何回画面が変わるかを示す指標である。1 秒あたりの画面変化数を数値で表し、fps(Frames Per Second) という単位で表す。この場合、1 秒間に 30 回画面が変わることになる。

2.1.3 動画アップロード

詳細を調べるために推奨形式以外での動画を実際に YouTube にアップロードすることで、視覚的变化を見比べる。今回は解像度と動画のフォーマットに注目し投稿を行った。

アップロードした形式を以下に示す。

投稿動画形式	
解像度	フォーマット
320 × 240 ピクセル	Motion JPEG
640 × 480 ピクセル	Motion JPEG
960 × 720 ピクセル	Motion JPEG
320 × 240 ピクセル	MPEG-2
640 × 480 ピクセル	MPEG-2
960 × 720 ピクセル	MPEG-2
320 × 240 ピクセル	DivX
640 × 480 ピクセル	DivX
960 × 720 ピクセル	DivX

表 2.1: 投稿動画形式

その結果、フォーマットによる、視覚的变化は見られず、解像度による変化が見られた。

次に投稿した解像度別動画の再生開始直後の画面を記す。



図 2.2: 320 × 240 ピクセル

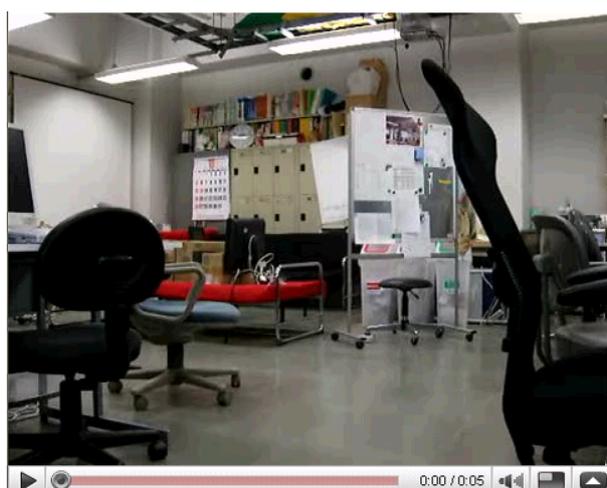


図 2.3: 640 × 480 ピクセル

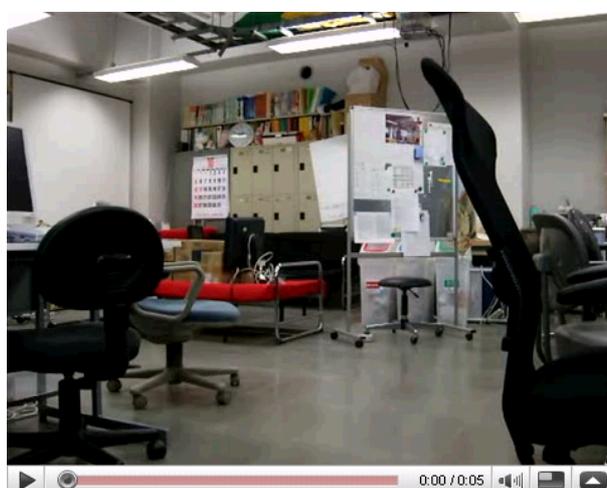


図 2.4: 960 × 720 ピクセル

動画の解像度は、撮影後にサイズ変更したものではなく、動画撮影時の設定でそれぞれの解像度に設定した。

320 × 240 ピクセルでは、椅子の周りの画質が荒くなっているが、640 × 480 ピクセル、960 × 720 ピクセルでは、荒れている部分はなくなり、640 × 480 ピクセル以上の動画であれば、きれいな動画として投稿できることがわかった。

10分以上の動画は、無条件にアップロードできなかった。また、投稿された様々なフォーマットの動画は、アップロード後に

FLV1 H.263

FLV5 H.264

の2種類にエンコードされる。

再生時には前者が表示されるようになっており、特定の操作をすると後者が表示されるようにできる。以下にその方法を示す。

1.FLV1 H.263 を再生している最中に「高画質で表示する」が表示されている場合は、この表示ををクリックする。

2. 見たい動画ページのアドレスの最後に「&fmt=18」を付け加える。 [12, 13]

という2通りの方法があり、どちらかを行うことでフォーマットが H.264 の動画で再生することができる。

2.2 YouTube の問題

YouTube で起こる著作権侵害は、動画を削除していけば解決していくと思うかもしれないが、投稿動画は日に約 35,000 作もアップロードされているという点から、人の力だけで、違法動画を見つけて一つ一つ削除していく手段では、莫大な手間と時間がかかってしまう。また 10 分という投稿制限があるものの、映画などを分割してアップロードをしている現状があり、動画タイトルをそれぞれ類似性のあるものにすることで、まとめて動画検索に引っかかるため、結果映画やテレビ番組が YouTube で見れてしまう現状がある。

2.3 既存のシステム

2.3.1 画像への基本挿入方法

「a」を埋め込む		aのアスキーコードは97(10進数) 2進数に直すと01100001 (アスキーコードは0~127)		
画素番号	画素値	透かし情報	変更後の値	
6行1列	133	0	134	0を埋め込む-画素値を偶数に
6行2列	132	1	133	1を埋め込む-画素値を奇数に
6行3列	132	1	133	1を埋め込む-画素値を奇数に
6行4列	133	0	134	0を埋め込む-画素値を偶数に
6行5列	133	0	134	0を埋め込む-画素値を偶数に
6行6列	133	0	134	0を埋め込む-画素値を偶数に
6行7列	132	0	132	0を埋め込む-そのまま
6行8列	133	1	133	1を埋め込む-そのまま

表 2.2: アスキーコードを用いた透かし

b8	b7	b6	b5	0000	0001	0010	0011	0100	0101	0110	0111
b4	b3	b2	b1								
0	0	0	0	NUL	DLE	SP	0	@	P		p
0	0	0	1	SOH	DC1	!	1	A	Q	a	q
0	0	1	0	STX	DC2	"	2	B	R	b	r
0	0	1	1	ETX	DC3	#	3	C	S	c	s
0	1	0	0	EOT	DC4	\$	4	D	T	d	t
0	1	0	1	ENQ	NAK	%	5	E	U	e	u
0	1	1	0	ACK	SYN	&	6	F	V	f	v
0	1	1	1	BEL	ETB	'	7	G	W	g	w
1	0	0	0	BS	CAN	(8	H	X	h	x
1	0	0	1	HT	EM)	9	I	Y	i	y
1	0	1	0	LF	SUB	*	:	J	Z	j	z
1	0	1	1	VT	ESC	+	;	K	[k	{
1	1	0	0	FF	FS	,	<	L	\	l	
1	1	0	1	CR	GS	-	a	M]	m	}
1	1	1	0	SO	RS	.	>	N	^	n	~
1	1	1	1	SI	US	/	?	O	_	o	DEL

表 2.3: アスキーコード表

基本的な電子透かしとして、画像へ文字を埋め込む方法を記す。

アスキーコードは表3のように「NUL」～「DEL」までの128種類のローマ字、数字、記号、制御コードで構成されている。「NUL」を0番として、順に番号を振っていくと「DEL」は127番となる。

今回埋め込みたい文字として、「a」を用いるとする。アスキーコード表より、「a」は97番目の文字である。10進数97を2進数表記すると、「01100001」となる。この値を透かし情報とし、画像内の画素番号6行1列～6行8列中に埋め込むとする。

2 進数のうち

1. ビット値 0 を埋め込んだ場合は画素値を至近の偶数に変える
2. ビット値 1 を埋め込んだ場合は画素値を至近の奇数に変える
3. 元々条件を満たしている場合は画素値を変更しない

という条件の下透かし情報を埋め込む。

埋め込んだ場所はわかっているので、6 行 1 列 ~ 6 行 8 列から画素値を読み取り、偶数は 0、奇数は 1 として読み取ると、「01100001」という値を得ることができる。それを 10 進数にし、アスキーコード表より「a」が埋め込まれていた、ということがわかる。

2.3.2 画像の周波数特性を利用した透かし

周波数特性を利用し、離散コサイン変換を用いる方法が多く使われている。[20] 例として、画素を離散コサイン変換 (DCT) を用いて周波数領域に変換してから透かし情報を埋め込む方法を用いる。DCT は、個のデータ単位で計算する必要がある。

・以下の式は画素数を $N \times N$ 画素としたときの離散コサイン変換である。

$$F(u, v) = \frac{1}{2}c(u)c(v) \sum_{x=0}^{3} \sum_{y=0}^{3} f(x, y) \cos\left\{\frac{(2x+1)u\pi}{2N}\right\} \cos\left\{\frac{(2y+1)v\pi}{2N}\right\}$$

$$u=0 \text{ のとき } c(u) = \frac{1}{\sqrt{2}} \quad v=0 \text{ のとき } c(v) = \frac{1}{\sqrt{2}}$$

$$u \neq 0 \text{ のとき } c(u)=1 \quad v \neq 0 \text{ のとき } c(v)=1$$

$F(u, v)$: 求める DCT 係数 $f(x, y)$: 画素

・以下に逆離散コサイン変換を示す

$$f(x, y) = \frac{1}{2} \sum_{u=0}^{3} \sum_{v=0}^{3} c(u)c(v)F(u, v) \cos\left\{\frac{(2x+1)u\pi}{2N}\right\} \cos\left\{\frac{(2y+1)v\pi}{2N}\right\}$$

左上角の成分は DC 成分といって平均的な明るさを示している。この部分を使い透かしの埋め込むと、ブロック全体の明るさが変化してしまう。そのため左上の部分は操作対象から除外する。

また右下の部分は周波数が高い成分を示している。画像の周波数成分の高いところは値が少々変化しても視覚的な違いには見極めがつかなく、違和感を与えない。よって透かしデータを埋め込む場所は 2×2 画素のうち右下角の 2 行 2 列に埋め込むこととなる。

1. 2×2 画素を 1 つのブロックとして考え透かしを埋め込む。

180	185
190	170

2. 現画像データを DCT 変換し、DCT 係数を求める。

362.5	7.5
2.5	-12.5

3. DCT 係数を量子化する。

量子化は、DCT 係数を 4 で割り、四捨五入をすることで求めることができる。

91	2
1	-3

4. 透かし情報として 0 を 2 行 2 列 DCT 係数-3 の位置に埋め込む。

透かしを埋め込む規則より、至近の偶数値に変更する。

91	2
1	-2

5. 逆量子化をする。逆量子化は、DCT 係数に 4 を掛ける。

364	8
4	-8

6. 逆 DCT 変換をする。

184	184
188	172

このような手法で埋め込みを行う。

2.3.3 各社提案違法動画検出方法

角川グループホールディングスでは正規動画と投稿動画を比較する方法を提案している。正規動画を動画投稿サイトに提供しデータベースに蓄積しておく。新規動画がアップロードされると、この新規投稿動画と正規動画を参照することで動画の類似性を判定し、一定以上の類似性がある場合は違法動画と判断して、その投稿動画の公開を止め、投稿者に警告し、コンテンツ事業者に通報することで動画を消去する方法である。この方法は90%の精度で違法性を判別できるとしている。

しかしこの方法では、毎回正規動画を提供していかなければならないため、手間がかかってしまう。また投稿した動画との比較のため、比較の時間がかかってしまう。加工した動画にも強いかどうかという疑問点が浮かぶ。

KDDI 研究所では、動画がプロによって作成されたものか、アマチュアによって作成されたものかを判別する方法を提案している。プロとアマチュアで撮影機器や撮影技術、製作工程などの違いがあることに着目する。これらの違いが表れる映像・音響特徴を解析することにより、その動画コンテンツがプロ制作のものか、アマチュア制作のものかを98%の精度で自動的に判別できるという方法である。

しかしアマチュアの撮影とはいっても、本格的撮影を行うアマチュアも存在するはずなので、違法性のない動画までも検出する可能性がある。

2.4 H.264

H.264の特徴として、圧縮率の高さが優れている点が挙げられる。MPEG-2の2倍の圧縮率を実現してしている。

構成として、MPEG-2、MPEG-4など従来の動画像圧縮符号化方式と同様に、既に符号化された画像フレームからの動きを推定して予測信号を生成し、残差信号を離散コサイン変換/量子化した後、エントロピー符号化を行う、動き補償 + DCT という技術をベースとしている。さらに、以下のような工夫を加えることで、従来を上回る圧縮率を実現した。[11, 21]

1. 動き補償を改善する

図4に示すように、従来のMPEG方式に比べて、画像処理単位を $4 \times 4 = 16$ 画素までとし、細かな画素精度で、きめ細かな動き補償を行う。

2. フレーム内の予測を効率的に行う

画面内符号化モードにおいて、符号化対象ブロックの周囲にある、既に符号化された信号から高度な予測を行う。また、符号化する画像の性質に合わせてさまざまな予測方向の中から最適なものを選択する。

3. 視覚的な画質劣化を抑える

DCTを用いる画像圧縮符号化方式の欠点であったブロック境界の歪を抑えるために、処理単位を 4×4 画素ブロックとし、視覚的な歪を目立ちにくくした。

4. 符号化する情報の性質に適応する技術

DCT/量子化をした後の信号をエントロピー符号化する際に、周囲の情報をもとに可変長符号語表を切り替えるなどし、画像の性質に合った符号化を行う。

2.4.1 画像の周波数が、高い低いとは

DCTは元画像を圧縮しやすくする変換技術であるが、入力される元画像を、数学的な離散コサイン変換を用いて低周波や高周波などの周波数成分に分解する仕組みをもつ。前章2.3.2の、画像の「周波数特性を利用した透かし」のように周波数という言葉が出てくる。画像に表示される画像の周波数とは図2.5(1)に示すように、表示される画面が、絵柄のない白が連続している、真っ白な画面の場合は、画面の中で隣り合っている画素は、まったく同じ白色、となる。このような状態のことを、「相関関係が非常に高い絵柄」という。周波数的にみると、ほとんど画面の変化がない状態ということから、「周波数成分が低い絵柄」という。

また、図2.5(2)に示すような、同一画面内に正反対の色である白と黒が交互に頻繁に登場するように激しい変化が次々に登場する市松模様のような画面は、「相関関係が非常に低い絵柄」という。周波数的に表現すると、激しい変化が次々に登場する画面ということから、「周波数成分が高い絵柄」という。

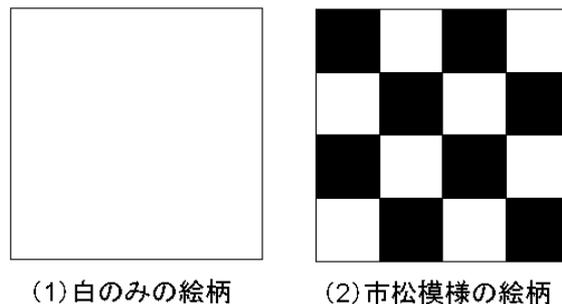


図 2.5: 画素と周波数

(1)

白だけの絵柄の場合、隣同士の画素がその相関関係は最も高い変化がないので周波数は最も低い

(2)

市松模様の場合、隣同士の画素は正反対の白と黒なので相関関係は最も低い急激な変化があるので周波数は最も高い

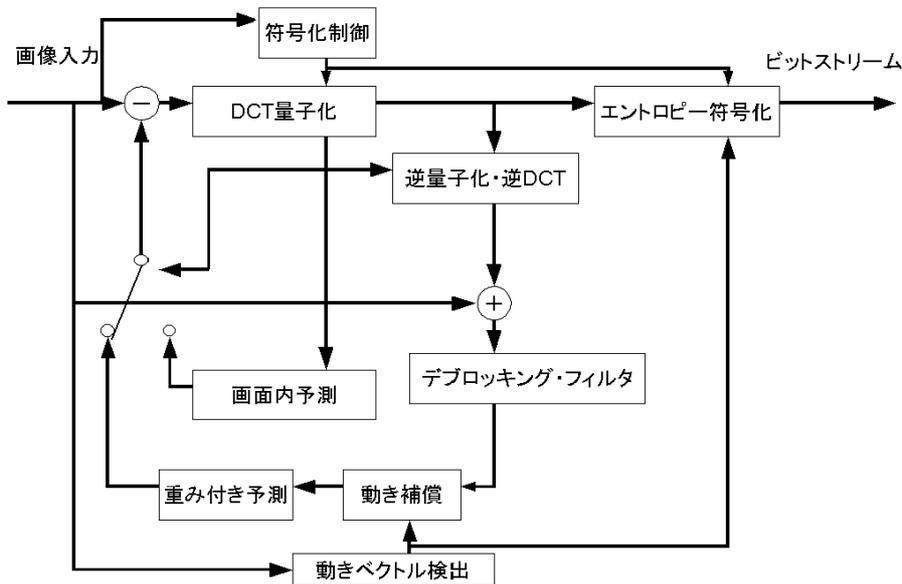


図 2.6: エンコーダの構成

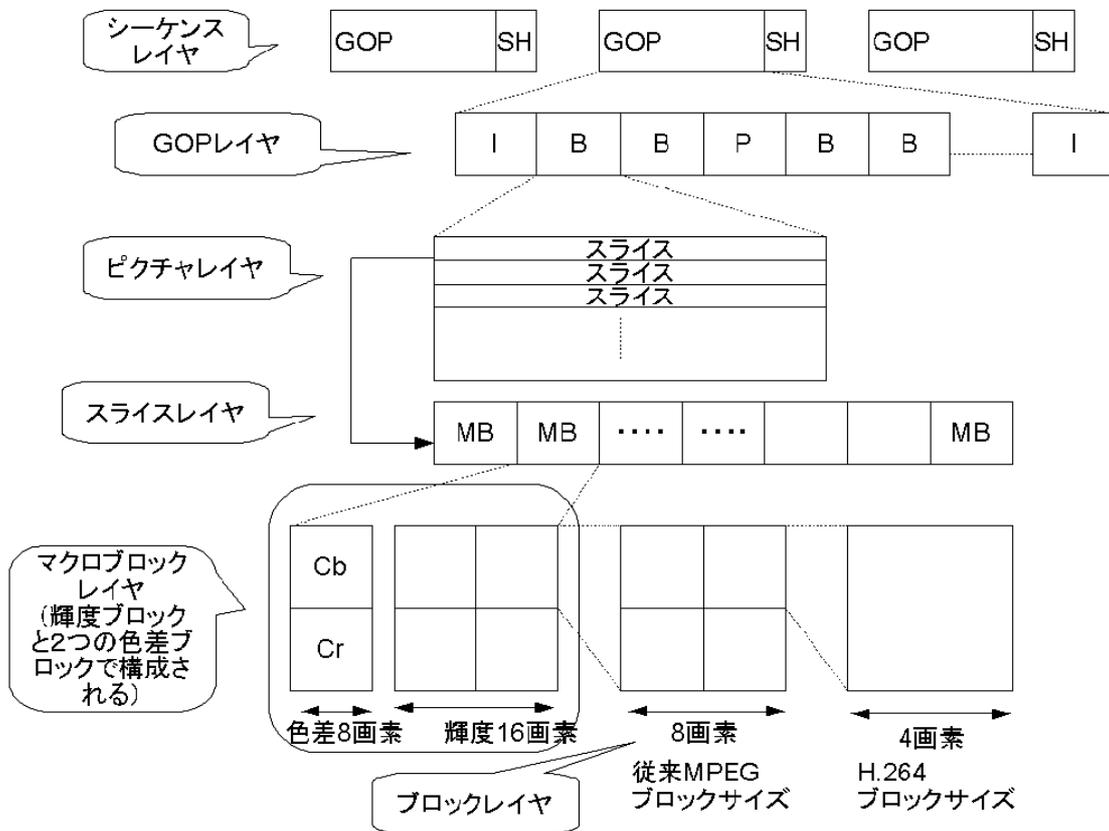


図 2.7: 圧縮データの6階層構成

2.4.2 GOP

GOPとは、動画を構成している最小の単位構造のことである。タイプの異なる複数のピクチャがある規則に基づいて並んでいるもの。

I・B・Pという3つのピクチャ構造を採用しており、ひとつのIピクチャに対してBピクチャとPピクチャを用いることで、編集やランダムアクセスが可能となっている。

2.4.3 SH

シーケンス・スヘッダの略で、GOP単位での独立再生ができるようにするため、GOP毎にシーケンス・ヘッダを設ける。シーケンス・ヘッダには画像の大きさ、画素の縦横比などデコーダが再生のために必要な初期データが入っている。

2.4.4 GOP レイア

1つのGOPの構成であり、I、B、Pピクチャでできている。

Iピクチャは他の画面を参照することなく、単独で符号化することによって画像を生成することができる。また、PピクチャとBピクチャの参照フレームとして用いられる。

Bピクチャは、両方向予測符号化画面であり、IピクチャまたはPピクチャを用いて、過去と未来の双方向からの予測符号化によって得られる画面である。

Pピクチャは、前方向予測符号化画面であり、時間的に過去に位置する、IピクチャもしくはPピクチャを予測画面として、画面間予測を用いて符号化処理が行われる。

2.4.5 スライス・レイヤ

ピクチャ内の 16×16 画素からなるマクロブロックを水平方向に接続した細長い帯状の領域。スライスは、量子化器に使用されるステップ・サイズを変更する単位となっている。

2.4.6 マクロブロック・レイヤ

マクロブロックは、輝度は4ブロック(16×16)と、2つの色差ブロック(各 8×8)で構成される。色差はブロックは、 $1/2$ にサブサンプルされているので、輝度4ブロックと色差1ブロックが、画面上で同じ大きさになる。符号化の仕組み、予測や符号化のモードは、 16×16 のマクロブロックを単位として示され、マクロブロックタイプで表される。それに対して、離散コサイン変換は、 8×8 のブロックを単位としている。そのため、マクロブロックは6個のブロックをまとめて扱うことになる。

2.4.7 ブロック・レイヤ

8 × 8 の DCT(離散コサイン変換) を行うための空間領域である。
H.264 は、さらに 4 × 4 に細分化して処理を行う。 [9, 18, 19]

2.5 画面内予測符号化

H.264 では、圧縮率を向上させるため、フレーム間予測を用いないマクロブロックに対して、上や左などに隣接するマクロブロックの隣接画素から補間によって予測画像を生成し、その予測画像との差分を符号化して画面内予測符号化を行います。予測画像の生成単位となるブロックサイズは、輝度成分については 4 × 4 および 16 × 16 画素の 2 種類であり、色差 (Cb,Cr) 成分の 8 × 8 画素については 8 × 8 画素単位の 1 種類である。また、複雑な画像は、4 × 4 画素の小さなブロック単位で予測し、平坦な画像は 16 × 16 画素の大きなマクロブロック単位で予測することで、8 × 8 画素単位のみで予測する MPEG-4 と比べて複雑な画像も平坦な画像も効率よく予測できる。MPEG-4 で導入されている AC/DC 予測では、予測する係数が DCT 係数の行列のうちの最上列ないし最左行の係数に限られているため、縦方向もしくは横方向の画素変化に対してしか予測効率を高めることができない。これに対して、H.264 の画面内予測では DCT 係数ではなく画素値での予測を行う。4 × 4 画素単位の画面内予測符号化では、図 2.8 に示す 16 × 16 画素を構成する 0 ~ 15 のブロックの順序に符号化を行う。さらに、12 番目のブロックを符号化する場合は、図 2.9 のように左ブロック、上ブロック、右上ブロックの 4 画素と左上ブロックの 1 画素の値からブロック内の 4 × 4 画素の値を予測し符号化するため、予測効率が大幅に向上している。予測方向は図 2.10 のように 9 通りの中から適切に予測できる予測方向を 4 × 4 画素のブロックごとに 1 つ選択し、選択した予測方向を 4 × 4 画素のブロック単位で符号化する。平均値については、左ブロックの 4 画素と、上ブロックの 4 画素の合計 8 画素の平均値より 4 × 4 画素全てを予測する予測方法である。

	MPEG-4	H.264
予測を行うデータ	直交変換された係数の直流成分と 低周波数の交流成分のみ	全ての画素値
参照するブロック	左、上、左上、の 3 箇所	左、上、右上、左上、の 4 箇所
予測方向の符号化	なし、方向は参照ブロックの係数の値で決定	あり
予測するブロックの単位	8 × 8 画素	4 × 4 画素または 16 × 16 画素

表 2.4: MPEG-4 と H.264 の画面内予測符号化の違い

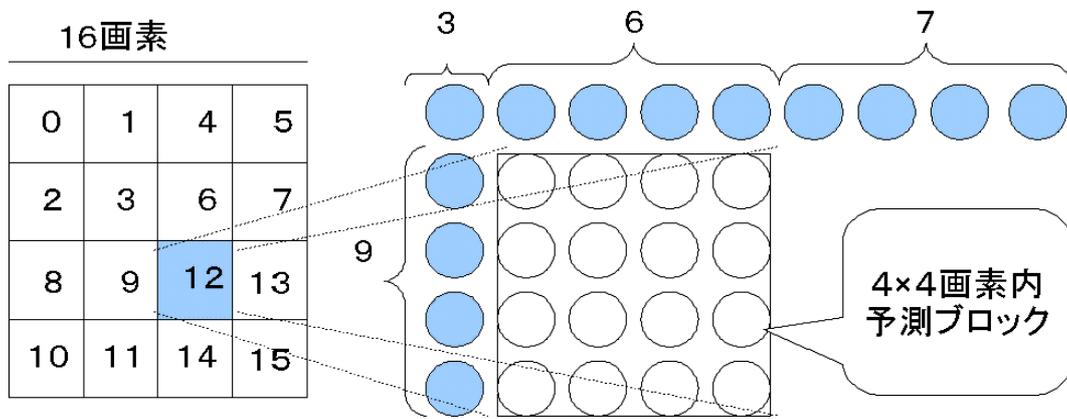


図 2.8: 4 × 4 のブロックの予測および符号化順序

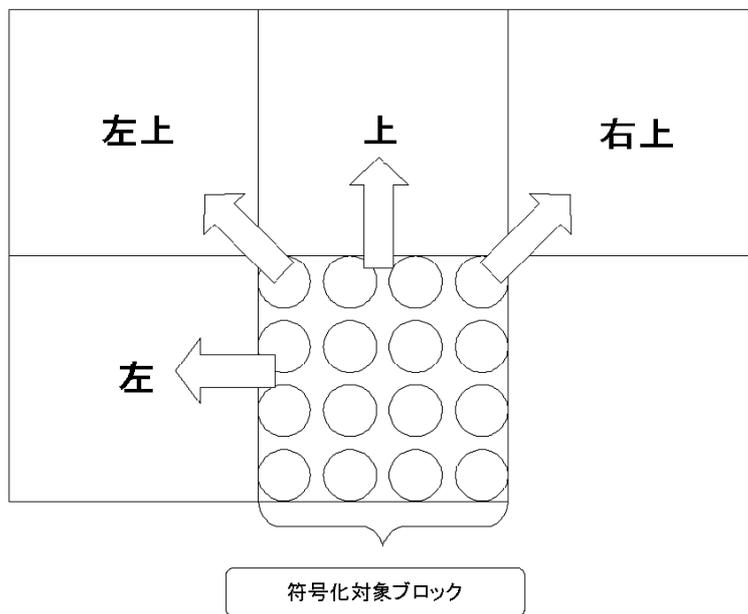


図 2.9: H.264 の符号化で参照するブロック

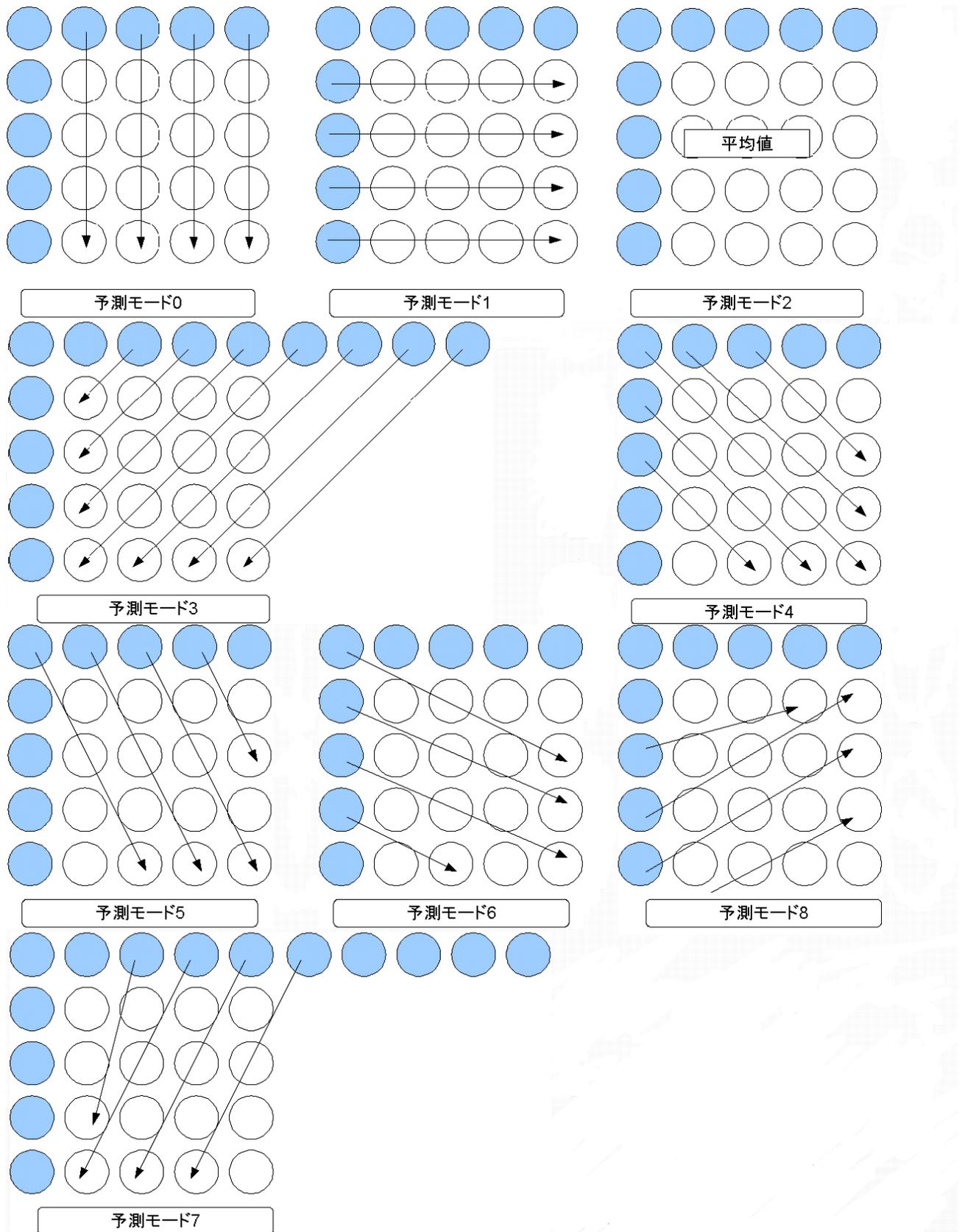


図 2.10: 4 × 4 画素の画面内予測

3 提案方式

3.1 埋め込み方式

今回提案する方式は、H.264 の特徴である 4×4 画素の画面内予測符号化の処理部分を用いる。そのため埋め込みはエンコーディング中に行う必要がある。

各ブロックごとにモード 0~8 の、9 通りの予測モード (図 2.10 参照) の中から適切と判断した予測モードを選択し予測符号化している点に着目し、電子透かし埋め込み方法を提案する。

電子透かしデータとして、1 ビット (0 と 1) の 2 通りの値を用い、ある条件下において、埋め込むこととする。

まず、透かしデータを埋め込む場所を決める。YouTube に動画を投稿するということは、2 章で示したように、「長さ 10 分までの動画をアップロードできる」ということが前提となっているため、投稿される動画は、切り取りなどの加工が行われているはずである。そのため、透かしの消失しないためにフレーム全てに透かしの埋め込むこととする。

ただし埋め込む場所を、毎フレーム同じ場所にしてしまうと、どれか一つのフレームで、透かしブロックの位置を特定された際に、全てのフレームの透かしがわかってしまうこととなる。よって、各フレームごとにフレーム違うブロックを選択することとする。

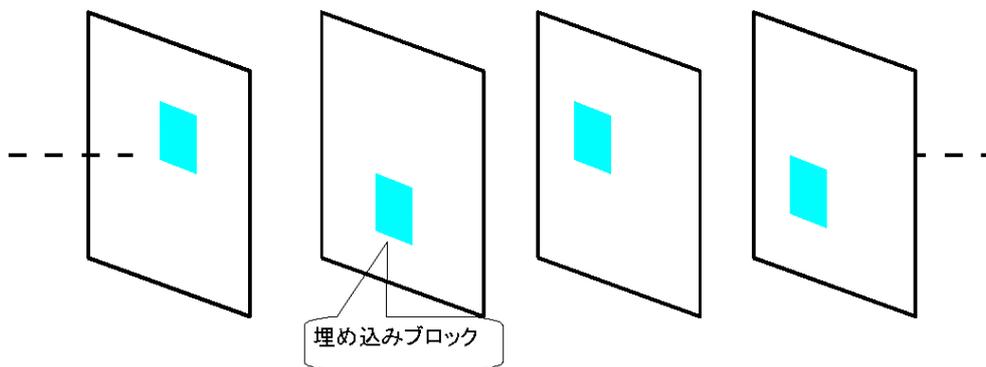


図 3.1: 透かし埋め込みブロック

透かし情報として、データ1、0のビット値を以下の条件により埋め込む。どのブロックに埋め込むかは埋め込み位置参照テーブルにより決定する。

全てのフレームに透かしを埋め込む際に、予測方法は、普段は図3.2に示すようにモード0~8の中から選択するのだが、今回は、図3.3に示すように埋め込むデータビット値が0のときは、偶数のモードから選択埋め込むデータビット値が1のときは、奇数のモードから選択することとする。

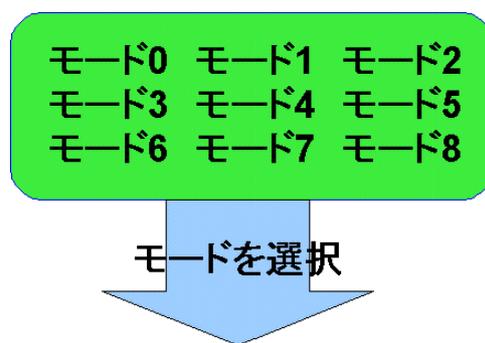


図 3.2: 予測モード選択

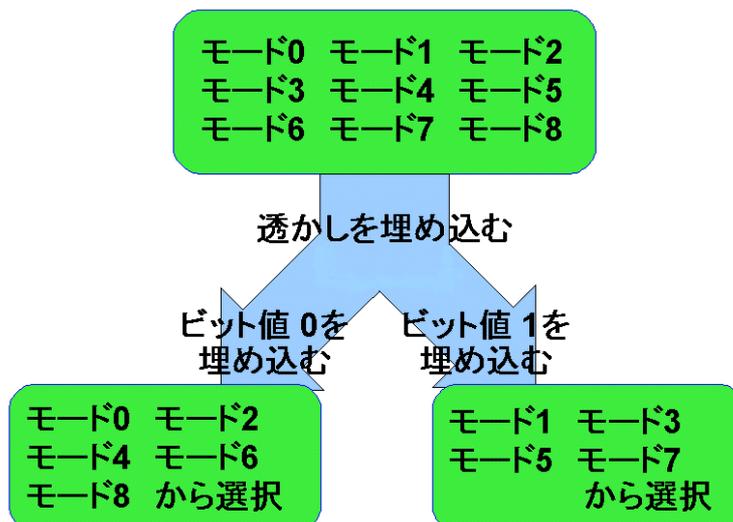


図 3.3: 提案予測モード選択

また、H.264 は適したモード番号を選択してくれるため、

- ・電子透かしデータのビット値が1 のとき、偶数のモード番号が選択されたら、モード番号を奇数に変更
- ・電子透かしデータのビット値が0 のとき、奇数のモード番号が選択されたら、モード番号を偶数に変更
- ・最初からそれぞれの条件を満たす場合にはそれぞれのモード番号を変更しないとする。

3.2 取り出し方法

透かしを埋め込む際には、エンコーディング中に行ったが、透かしを取り出すときは、デコーダを用いず、ストリームを解析することで透かしを取り出す。

透かしを埋め込んだブロックは埋め込み位置参照テーブルによりわかっているので、埋め込んだ動画のフレームから、そのブロックのビットストリームを抽出する。そこで使われた予測モードを知ることで、偶数の予測モードを使って符号化をしていけば、0 を、また奇数の予測モードを使って符号化をしていけば、1 をそれぞれ透かし情報として埋め込んだことがわかる。

4 結論

本論文では、動画投稿サイトの一つである YouTube へ実際に動画を投稿することで、アップロードの仕組みや、エンコード後の形式について述べ、非可視型電子透かし方法を提示した。

また YouTube に動画を投稿することで、高い圧縮率を持つという利点のある H.264 にエンコードされることから、H.264 に特化した電子透かしを提案した。その際に MPEG シリーズとは違う、H.264 ならではの符号化方式である 4×4 画面内予測部分を用いた。

9通りの予測モードを条件によって偶数モード、奇数モードに分け、1ビットの透かしの埋め込む。埋め込み位置参照テーブルを用いる事により透かしの取り出しを高速に行えるようにした。

今後更に普及していくと考えられる H.264 への埋め込む透かしであるので、不正投稿の防止が期待できる。

今後の課題として、提案方法を用い実際に動画を投稿することにより透かしへの影響を調べる事があげられる。

5 謝辞

本研究を行うにあたり、終止熱心にご指導して頂いた木下宏揚教授に心から感謝致します。また、様々な面で数多くの有益な御助言をして頂いた、鈴木一弘助手に深く感謝致します。さらに、公私にわたり、良き研究生生活を送らせて頂いた木下研究室の方々に感謝致します。

6 参考文献

参考文献

- [1] 動画像の高能率符号化
著者：小野定康・村上篤道・浅井幸太郎 出版社：オーム社
- [2] H.264/AVC 教科書
著者：大久保榮 出版社：インプレス
- [3] 電子透かしとコンテンツ保護
著者：小野束 出版社：オーム社
- [4] 動画職人になる本
著者：勝田有一朗・前田尊 出版社：工学社
- [5] WebSite expert
出版社：技術評論社
- [6] 電子透かし
著者：スコット・モスコウィッツ 出版社：小学館
- [7] 電子透かしの基礎
著者：松井甲子雄 出版社：森北出版株式会社
- [8] <http://dev.sbins.co.jp/watermark/usage.html>
電子透かし技術の種類と使用法
- [9] http://www.axiscom.co.jp/files/whitepaper/wp_h264_32205_jp_0805_lo.pdf#search=H.264 画像圧縮標準
- [10] http://code.google.com/intl/ja/apis/youtube/developers_guide_protocol.html#User_Uploaded_
デベロッパーガイド
- [11] <http://ja.wikipedia.org/wiki/H.264>
Wikipedia
- [12] <http://learning-life.net/blog/?cat=14>
iPhone, Flash, H.264 and YouTube

-
- [13] http://matinoakari.net/news/item_68468.html
YouTube **さんが高画質化へ**
- [14] <http://it.jeita.or.jp/eltech/report/2001/01-jou-04.pdf>
電子透かし報告書
- [15] <http://www.datahide.com/BPCSj/>
KIT ステガノグラフィ研究グループ
- [16] <http://hp.vector.co.jp/authors/VA017815/insideof4.htm>
ステガノグラフィー
- [17] <http://www.atmarkit.co.jp/news/200703/22/youtube.html>
CGMの強い誘導力
- [18] <http://www.avcc.or.jp/e-kyouiku/e-10/e-10-06.html#10>
AVCC
- [19] <http://www.acr.atr.co.jp/~hirata/Doc/MPEG.htm>
MPEG
- [20] <http://www.jpo.go.jp/shiryousonota/map/denki14/frame.htm>
デジタル動画像圧縮技術
- [21] <http://mikilab.doshisha.ac.jp/dia/research/report/2005/0918/003/report20050918003.html>
動画圧縮技術

7 質疑応答

Q:他のフォーマットでは予測モードを用いた技術は使えないのか。

A:4 × 4 画素内符号化は H.264 のみの技術なので、この方式は適用できない。

ただし、例として、MPEG-4 は 8 × 8 予測なので、予測モードを得ることができれば、同じ方式でできる。