

再撮影動画から透かし入り動画の領域を推定する電子透かし法 Digital Watermarking Method Estimating Watermarked Video Region from Recaptured Video

穂西 俊弥 * 岩田 基 * 黄瀬 浩一 *
Toshiya Akinishi Motoi Iwata Koichi Kise

あらまし 動画に対して電子透かしを用い、人に知覚されないように付加情報を埋め込み、それをカメラで再撮影することで情報を取り出す手法について検討する。透かし入り動画にスマートフォンをかざすことで関連情報にアクセスできるようにすることが目的である。透かし入り動画を再撮影した動画には、透かし抽出の対象である透かし入り動画の領域以外に背景領域も含まれる。そこで、電子透かし法を、再撮影動画内の透かし入り動画の領域を推定する処理と透かしの埋め込み・抽出を行う処理に分けて考える。まず、透かしの埋め込み・抽出にはフレーム間の差分を利用する手法を用いる。提案手法は携帯端末上でのリアルタイム処理を目指しているため、小さい計算コストが望まれる。そこで、周波数変換などと比べ計算コストが小さい単純な和算、減算のみを用いて計算できるフレーム間差分を利用する方法を用いた。次に、透かし入り動画の領域推定には、透かしとは別に位置ずれ同期用の情報を埋め込むことも考えられるが、画質劣化の原因になるため好ましくない。そこで本稿では、透かし自体の痕跡を利用して透かしを抽出する方法を新たに提案する。実験では、スマートフォンにより撮影した、手ぶれを含む再撮影動画から透かしを抽出し、提案手法の性能を評価する。

キーワード 電子透かし 動画 再撮影

1 はじめに

近年、スマートフォンやタブレット端末が急速に普及している。それらの多くはカメラを搭載していることに加え、Web ページを見やすい設計になっている。そこで、テレビ番組やCM、街頭の電子看板などの動画をスマートフォンのカメラで再撮影することで、内容に関連のある Web ページへアクセスできると便利である。また、クーポンを取得するという用途も考えられる。

このようなシステムを実現するために、電子透かしを利用できる。電子透かしとは、画像や動画、音楽などのマルチメディアコンテンツに対し、人間が知覚できないほど微小な変更を加えることで付加情報を埋め込む技術であり、この付加情報は透かしと呼ばれる。CM などの原動画に対し、透かしとしてアクセス先の Web ページの情報などを埋め込むことで前述のシステムを実現できる。透かし入り動画から透かしを抽出するときには、再撮影による色の変化や、照明の影響、カメラの傾きによる幾何ひずみなどの影響を受けるため、これらの影響が

あっても抽出可能な手法が求められる。

このような用途に関する研究として、中村らの手法 [1] が挙げられる。中村らの手法は、再撮影フレームから透かし入り動画の領域 (以下、透かし領域) を推定し、その領域から透かしを抽出する。透かし領域の推定には透かし入り動画部分とディスプレイ枠部分の輝度差を利用した Side Trace Algorithm[2](以後 STA とする) を用いており、電子透かし法にはフレームの差分に透かしの埋め込むフレーム間差分法を用いている。このフレーム間差分法は表示、再撮影による色の変化が連続フレームに同様の影響を与えることを利用しているため、再撮影に耐性がある。しかし、STA は透かし入り動画部分とディスプレイ枠部分の輝度差が小さいときには検出に失敗してしまうという問題点がある。これを解消するために、例えば、輝度差が大きくなるような枠線を付加することが考えられるが、動画の見た目を損ねるため望ましくない。

また、山中ら [3] は印刷物をカメラで撮影し、関連情報を得るための手法について提案しているが、位置ずれ同期のために QR コードの位置検出パターンを用いている。これにより精度高く位置ずれ同期を行うことができるが、やはり位置検出パターン自体が目立ってしまい、原コンテンツの価値を損ねる。

* 大阪府立大学大学院工学研究科, 〒 599-8531 大阪府堺市中区学園町 1-1. Graduate School of Engineering, Osaka Prefecture University, 1-1, Gakuen-cho, Naka-ku, Sakai, Osaka, 599-8531 Japan.

そこで本稿では、枠線や位置検出パターンなどの位置ずれ同期用の情報を付加することなく透かし領域を推定する方法を提案する。透かしとは別に位置ずれ同期用の情報を付加しなくても、透かし自体が動画になんらかの痕跡を残すと考えられる。そのような痕跡を利用して透かし領域を推定することにより、位置ずれ同期用の情報の付加を必要とせずに再撮影した動画から透かしを抽出できる電子透かし法を実現する。

以下、2章で透かしの埋め込み・抽出法について述べ、3章では透かし領域推定法について述べる。そして、4章で実験結果とその考察を示し、5章でまとめとする。

2 透かし埋め込み・抽出法

本章では、フレーム間の差分を利用した透かしの埋め込み法と抽出法について説明する。再撮影により得られる動画は原動画と比べ、色の情報などが大きく変化する。しかし、再撮影で得られる連続フレームに注目すると、どのフレームにも似た変化が加わる。さらに、動画の連続したフレームの内容の変化は小さいため、フレーム間の差分を取ることで原動画の成分を除去することができる。よって、フレーム間の差分に透かしの埋め込みと透かしのパターンだけが強調されるため、埋め込みによる画素値の変更量を小さくすることができる。

2.1 埋め込み法

Step1 埋め込みたい情報である N ビットの透かしビット列 $w = \{w_j | w_j \in \{0, 1\}, 1 \leq j \leq N\}$ を用意する。これを符号化率 $1/2$ の畳み込み符号により誤り訂正符号化して、 $2N$ ビットの符号語 $c = \{c_i | c_i \in \{-1, 1\}, 1 \leq i \leq 2N\}$ を得る。

Step2 図1に示すような符号語 c を二次元的に配置したパターンフレームを用意する。ここでパターンフレームとは埋め込み対象の動画と同じサイズを持ち、幅方向に X 分割、高さ方向に Y 分割された、合計 XY 個の領域（以下、ブロックと呼ぶ）からなるフレームであり、各ブロック内の全画素値は透かしビットに対応する -1 もしくは 1 の値をとる。このときパターンフレームの四隅のブロックはパターンを正しく取り出すための同期用として用いるため 1 の値を持つブロックにする。従って、埋め込める透かしの長さは $N = (XY - 4)/2$ ビットとなる。

Step3 図1に埋め込みの様子を示す。フレームから差分を取ったときにパターンフレームが得られるように、2枚のフレームを単位として処理を行う。ここで、Step2で作成したパターンを表パターン、表パターンの正負を入れ替えたものを裏パターンと定

義する。一対の原動画フレームに表もしくは裏のパターンを割り当てる。 f 番目のフレームと $f+1$ 番目のフレームの差分からは表のパターンが得られるように、 f 番目のフレームには表パターン、 $f+1$ 番目のフレームには裏パターンを割り当てる。次の $f+2$ 番のフレームと $f+3$ 番目のフレームの差分からは裏パターンが得られるように、 $f+2$ 番目のフレームには裏パターン、 $f+3$ 番目のフレームには表パターンを割り当てる。これを繰り返すことで、全てのフレームに表パターンか裏パターンが割り当てられる。

Step4 割り当てられたパターンフレームにブロックごとに可変の画素値変更量 T を掛けたものを原動画フレームに重畳する。以下では、1つのブロックへの埋め込みを説明する。 i 番目の符号語ビットを埋め込むブロック位置は、四隅を除いたラスタスキャン順で i 番目に当たるブロックである。まず、 f 番目のフレームの i 番目のブロックと、 $f+1$ 番目のフレームの i 番目のブロックのそれぞれのブロック内の画素の平均値の差分をとり、差分平均値 d_i とする。その後、 d_i と透かし強度 S を用いて、式(1)、式(2)により、画素値変更量 T を求める。 $c_i = 1$ かつ $d_i \geq S$ の場合と、 $c_i = -1$ かつ $d_i \leq -S$ の場合は埋め込み条件を満たしているため、 $T = 0$ とする。

- $c_i = 1$ の場合

$$T = \text{round} \left(\frac{S - d_i}{2} \right) \quad (1)$$

- $c_i = -1$ の場合

$$T = \text{round} \left(\frac{S + d_i}{2} \right) \quad (2)$$

ここで、 $\text{round}()$ は四捨五入を行う関数である。 f 番目、 $f+1$ 番目のフレームに割り当てられたパターンフレームの i 番目のブロックに T をかけて、フレームの同じブロック位置の値に加算する。このとき、値が0未満になったときは値を0に、画素値が255より大きくなったときは値を255に補正する。これを全ての原動画ブロックに対して行い、四隅の同期用ブロックにも同様の処理を行う。

Step5 Step4を全てのフレームについて行うことで透かし入り動画を得る。

2.2 抽出法

M 枚のフレームを対象として透かしを抽出する方法について述べる。まずは、1枚の差分フレームからの符

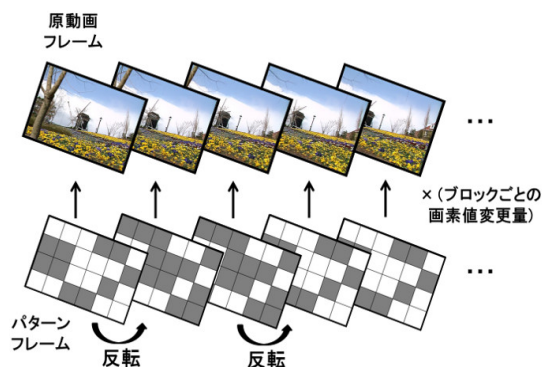


図 1: パターンフレームの重畳

号語抽出処理を示す。抽出処理の対象である差分フレームは、 f 番目のフレームから $f + 1$ 番目のフレームを減算することで得られる。

差分フレームを埋め込み時と同じ XY 個のブロックに分割して各ブロックから符号語ビットを抽出する方法は以下ようになる。符号語ビットの抽出は差分フレームのブロックごとに、平均が 0 以上か 0 未満かによって判定する。このとき、四隅の同期用ブロックの平均値の値によって判定基準を変える。同期用ブロックの平均値が 0 以上なら、ブロック平均値が 0 以上のとき符号語ビットを 1 として抽出し、0 未満のとき -1 として抽出する。逆に、同期用ブロックの平均値が 0 未満なら、ブロックの平均値が 0 未満のとき符号語ビットを 1 として抽出し、0 以上のとき -1 として抽出する。四隅を除く全てのブロックから符号語ビットを抽出し、1 枚の差分フレームからの符号語抽出が完了する。

上記の差分フレームからの符号語抽出を、 M 枚のフレームから得られる $M - 1$ 枚の差分フレームに対して繰り返し、ビットごとの多数決をとり多数派となったビットを並べた符号語ビットを得る。このとき、タイミングによっては、パターンフレームが反転していないところから抽出が行われるが、パターンフレームが反転している差分フレームと比べると値が小さくなる。そこで、差分フレームの絶対平均値の大きさ順で上位半分の信頼性の高い差分フレームからの結果のみを投票に用いる。

そして得られた符号語ビットにビタビ復号で誤り訂正したものを抽出結果の透かしとする。このときビタビ復号時に得られるパスコストを抽出結果の信頼度判定に用いる。パスコストは誤りが多いほど大きい値をとるので、パスコストが R より大きい場合は結果を信頼せず、新たな M 枚のフレームに対して上記の処理を繰り返す。フレーム数 M を増やすほど投票数が増え、多くの情報を抽出結果に反映できるが、抽出にかかる時間も長くなる。

3 透かし領域推定法

本章では背景を含む撮影動画から透かし領域を推定する手法について述べる。まず、従来法である Side Trace Algorithm(STA) について述べ、その後に提案法について述べる。

3.1 従来法

3.1.1 手順

STA を用いて、再撮影により得られた動画フレーム(再撮影フレーム)から透かし領域の四隅の点を検出する従来の方法について説明する。

図 2 は STA での探索手順である。STA は、再撮影フレームの左辺の中心点、右辺の中心点それぞれから再撮影フレームの中央に向かう直線上のエッジを探索し(図 2 の 1, 4)、エッジが見つかったら、そこから上下にエッジを追跡し、エッジが途切れたところ(図 2 の 2, 3, 5, 6)を透かし領域の四隅の点と定める方法である。以下に詳細な手順を述べる。これらの処理はフレームの輝度成分に対して行う。

Step1 図 2 の 1 に示すように再撮影フレームの左辺の中心点から 1 画素ずつ右に進み、図 3 のエッジ検出用フィルタの値がしきい値を超える点を見つける。

Step2 見つかった点から上向きに辺を探索する。Step1 と同様のフィルタを用いて辺を辿り、しきい値を下回る点が見つければ、その点を透かし領域の左上の点とする(図 2 の 2)。この際、辺の傾きを考慮し、真上の点だけでなく、その左右 1 画素も探索範囲とする。

Step3 Step2 と同様に、下向きの探索も行い、透かし領域の左下の点を定める(図 2 の 3)。

Step4 Step1 から Step3 と同様の処理を再撮影フレームの右辺の中心点から行い、透かし領域の右上、右下の点を求める(図 2 の 4~6)。

Step5 検出された 4 点で定義される領域を透かし領域とする。

3.1.2 問題点

黒枠のディスプレイに暗めの動画が映っている場合など、ディスプレイ枠と透かし領域の境界部分の輝度差が小さい場合は STA が失敗しやすい。これは、STA の起点となるエッジを見つけることができないためである。この問題に対処するために輝度差を大きくする枠線を付加することなどが考えられるが、そのような枠線自体が人間にとって目立ってしまう。そこで埋め込みは透かし

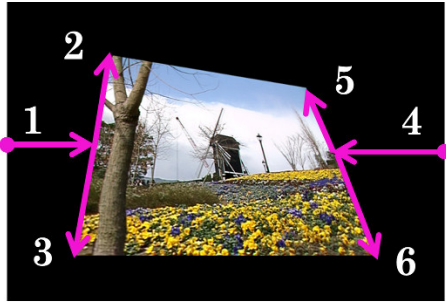


図 2: STA の探索手順

1	0	-1
2	0	-2
1	0	-1

図 3: 辺検出に用いるフィルタ

情報のみとし、再撮影動画からその領域を推定できることが望ましい。

3.2 提案法

本章では、再撮影動画から透かし領域を推定する提案法について述べる。提案法では M 枚の連続した再撮影フレームから透かし領域を推定し、用いた M 枚からの透かしの抽出処理は、その推定された透かし領域に対して行う。再撮影時の手ぶれの影響で、 M 枚のフレームそれぞれにおける透かし領域は異なるが、 M が小さければ手ぶれの影響は小さいと考えられる。以下に詳細な手順を示す。

Step1 M 枚の再撮影フレームから $M - 1$ 枚の差分フレームを得る。

Step2 差分フレーム全体を $B_x \times B_y$ 画素の推定用ブロックに分割する。

Step3 以下に、Step2 で得られた推定用ブロック 1 つに対する処理について述べる。推定用ブロックの平均値 $a = \{a_f | 1 \leq f < M - 1\}$ は時系列方向に $M - 1$ 個の値を持つ。透かしは 2.1 節に示したように、パターンフレームを 2 枚ごとに反転させながら埋め込んでいるので、透かし領域に含まれる推定用ブロックの差分の値は、正の値、0 に近い値、負の値、0 に近い値というパターンを繰り返す。このパターンおよびこれを巡回シフトさせたパターンのうちいずれかに合致するか否かを調べる。これを調べるために、 f 番目のフレームと $f + 1$ 番目のフレームの推定用ブロックの平均値の差分 $a_f - a_{f+1}$ を見て、上がり下がり判断する。式 (3) にパター

ンの判定式を示す。実際にはこれを巡回シフトさせた残り 3 種類の式による判定も行う。 g は上がり幅下がり幅のしきい値である。

$$\begin{aligned} a_f - a_{f+1} &\geq g \\ a_{f+1} - a_{f+2} &\geq g \\ a_{f+2} - a_{f+3} &\leq -g \end{aligned} \quad (3)$$

パターン判定の開始フレーム番号 f を時系列方向に 1 つずつずらしながら探索を行い、 m 回以上合致した場合、その推定用ブロックを透かし領域の候補 (候補ブロック) とする。これを全ての推定用ブロックに対して行う。

Step4 Step3 で得られた候補ブロックを 1 の値を持つ画素に、それ以外の推定用ブロックを 0 の値を持つ画素に変換することにより、二値画像 (透かし領域二値画像) を得る。透かし領域二値画像にラベリング処理を施すことによって連結成分を得、最も面積の大きい連結成分のみを残す。ここで、ラベリングは 4 連結に基づいて行う。そして、その領域を囲む最小の矩形を得る。ただし、得られた最大連結成分の画素数が画素全体の 5% 以下の場合には、透かしを取り出せる見込みが少ないため、失敗として処理を終える。

Step5 Step4 で得られた矩形の辺と最大領域との接点を求める。矩形の辺と平行な直線をその接点を中心に回転させる。このとき、接点から遠い位置にある側の矩形の頂点から見て矩形内部の方向に回転させる。こうして、透かし領域二値画像の 1 の値を持つ画素を初めて一定数以上貫通したら、その直線を保存する。これを上下左右全ての矩形の辺で行い、保存した 4 辺の各 2 辺が交わる点を 4 点保存する。

Step6 Step5 で保存した 4 点を元のフレームサイズに合うように座標変換し、これらを四隅の点として透かし領域を定義する。

求めた透かし領域に対して射影変換を施して元のサイズに戻すことにより、透かし入り動画のフレームを得る。

4 実験と考察

再撮影動画から提案法を用いて抽出を行い、透かしの抽出成功回数と所要試行回数、所要時間を調べた。

4.1 実験条件

本実験の実験条件は以下の通りである。原動画には 1920×1035 画素の HD 画質の動画 5 種類を用いた。原

動画は YPbPr 形式で、フレームレートは 30fps である。透かし強度は $S = 6$ とし、埋め込み対象は Pb 成分とした。透かしのパターンフレームの幅方向の分割数は $X = 6$ 、高さ方向の分割数は $Y = 6$ とした。四隅のブロックを同期用に用いるため、埋め込める符号語は $32(2N = XY - 4)$ ビットとなり、透かしは 16 ビットとなる。

提案法の推定用ブロックのサイズは $B_x = 5, B_y = 5$ 、上がり幅下がり幅のしきい値は $g = 1$ 、候補ブロックと判定するためのしきい値は $m = 1$ 、推定に用いるフレーム数は $M = 10$ とした。 $M = 10$ はおよそ 0.33 秒分のフレーム数にあたる。また、ラベリングには大阪大学の井村誠孝氏のプログラム¹ を用いた。

動画の再撮影には iPhone 5 を使用した。iPhone5 のカメラによる再撮影動画の解像度は 1920×1080 画素であり、フレームレートは 30fps である。透かしの抽出は再撮影動画を対象として PC 上で行った。用いた CPU は 3.50GHz の Intel Core i7 2700K であり、メモリは 16GB である。動画を表示するディスプレイには 2880×1800 画素の解像度を持つ Macbook Pro Retina ディスプレイモデルの 15 インチを用い、YPbPr 形式の埋め込み動画を 24Mbps の MPEG 動画として再生した。再撮影は iPhone5 を手に持った状態で、50cm 程度の距離から撮影した。

撮影フレーム 1920×1080 画素のうち、正しい透かし領域は幅、高さ、それぞれ再撮影フレームの幅、高さの 60% 程度の長さであり、透かし領域の面積は再撮影フレーム全体の 35% 程度である。すなわち、透かし領域のサイズはおおよそ 1152×648 画素程度となる。その領域を 6×6 個のブロックに分割して符号語を埋め込んでいため、再撮影された動画における符号語 1 ビットが埋め込まれているブロックのサイズはおおよそ 192×108 画素である。

4.2 透かし入り動画の画質

透かし入り動画を静止した状態で見ると一部平坦な領域でブロックの境界が目立つ部分があった。図 4、図 5 にそれぞれ動画 soccer の原動画のフレーム、透かし入り動画のフレームを示す。図 5 は、前述のようなブロックの境界が目立つ例である。芝生部分のような領域では透かしによるわずかな色の变化が知覚される。しかし、動画再生時にはパターンが高速で反転するため、ブロックの境界はほとんど知覚できない。ただ、パターンが反転することにより、動画によってはちらつきが知覚された。

表 1 に客観的画質評価指標である PSNR を示す。PSNR は原動画を MPEG 圧縮したものと、透かし入り動画を



図 4: 原動画のフレーム



図 5: 透かし入り動画のフレーム

表 1: 客観的画質評価指標 PSNR

flower	34.0dB
square	33.3dB
soccer	31.9dB
leaves	30.6dB
chromak	35.0dB

同様に圧縮したものとの間でフレームごとに算出し、それらの平均をとった。一定の強度で埋めこむために必要な変異量が動画ごとに異なるため、大きな変更が必要となった動画に関しては PSNR が低い。

4.3 抽出成功回数

透かし入り動画を再撮影し、提案法により透かし抽出を行った結果を示す。処理開始の先頭フレームを 1 フレームずつずらしながら、400 回抽出処理を行った。ピタビ復号時のパスコストに基づいた信頼度判定は行わず、得られた透かし全ての抽出成功数を求めた。

表 2 に 400 回の試行中、何回透かしの取り出しに成功したかを示す。ここで成功とは、埋め込んだ透かしビットを全て正しく取り出せることを意味する。動画によるばらつきが大きく、特に leaves の抽出成功回数が少ない。

¹ <http://oshiro.bpe.es.osaka-u.ac.jp/people/staff/imura/products/labeling/source/sample.zip>

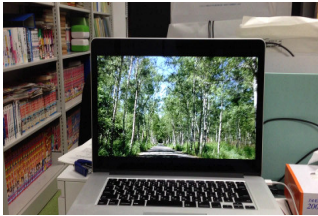


図 6: 撮影フレーム

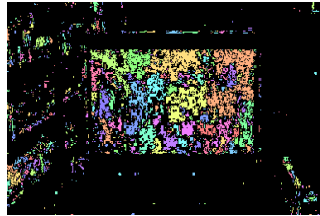


図 7: 領域推定失敗例

表 2: 抽出成功回数

flower	391/400
square	400/400
soccer	355/400
leaves	44/400
chromakey	226/400

図 8~ 図 12 に、差分フレームごとの正しく取り出せたビット数を示す。埋め込んだビット数は 16 ビットである。また、透かし領域推定に失敗した場合の正解ビット数は 0 としている。なお、領域推定に成功した場合、正解ビット数が 0 になることはなかった。この結果から、抽出成功回数が少ない leaves は透かし領域推定に失敗していることが多いことがわかる。また、soccer や chromakey の結果を見ると、透かし領域推定に成功する区間と失敗する区間が分かれていることがわかる。

先に述べたように、leaves では透かし領域推定に失敗することが多かった。図 6 と図 7 に、透かし領域の推定に失敗したときの、leaves の再撮影フレームと、最大連結成分以外を除去する前の候補ブロックのラベリング結果を示す。同じ色の領域が連結成分を表しており、正解の透かし領域から候補ブロックは得られているが、透かし領域内が分断されてしまっているのが失敗の原因だとわかる。このような場合、二値画像処理等で分かれた領域を連結できる可能性がある。

4.4 所要試行回数と所要時間

処理開始の先頭フレームをランダムに決定し、その位置から 10 フレームずつ先頭フレームをずらしながら抽出を繰り返し、抽出が完了するまでの所要試行回数と所要時間を計測した。ピタビ復号時の信頼度判定のしきい値は $R = 4$ とした。この値は予備実験により求めた復号後に誤りが生じない値である。

表 3 に 100 回の試行の平均値結果を示す。flower, square, soccer に関しては、おおよそ 1 回という少ない試行回数で透かしの抽出が可能であった。

leaves に関しては、動画の末尾までに処理が完了しない試行が存在した。leaves 以外の動画は 100 回の試行全

表 3: 所要試行回数と所要時間

	所要試行回数	所要時間
flower	1.03 回	1375msec
square	1.00 回	1322msec
soccer	1.33 回	1815msec
leaves	×	×
chromakey	4.87 回	7257msec

てで透かしの抽出に成功したが、leaves は 44 回のみ成功した。本実験では 10 フレームずつ処理開始のフレームをずらししているため、処理開始のフレームより後ろに 10 フレーム以上連続する抽出に成功する箇所があれば必ず抽出に成功する。図 11 を見ると、後半部分に 10 フレーム以上連続する抽出に成功する箇所がないため、フレーム開始位置によっては、成功する開始フレームから処理を始められなかったと考えられる。成功した試行の平均所要試行回数は 7.48 回、平均所要時間は 12153msec であり、成功する場合でも他と比較して長い時間が必要であった。

5 おわりに

本稿では、透かし以外の位置ずれ同期用の情報を付加することなく透かし入り動画の領域を推定する電子透かし法を提案した。提案手法は目に見える枠線などを付加しなくても、透かし自体の痕跡を利用して透かし領域を推定できる手法であり、一部動画を除いては少ない試行回数での透かし抽出が可能であった。失敗の多い動画では透かし領域内での分断が起こっており、二値画像処理などを施すことで連結させることができる可能性がある。

今後の課題は、提案法を実際の端末上で動くように処理を軽量化し、実装を行うことである。

参考文献

- [1] 中村高 雄, 山本 奏, 北原 亮, 宮武 隆, 片山 淳, 安野 貴之, 曾根原 登, “リアルタイム検出可能な動画向けモバイル電子透かし,” 画像電子学会, vol.36, no.4, pp.426–434, 2007.
- [2] 片山 淳, 中村高 雄, 山室 雅司, 曾根原 登, “電子透かし読み取りのための i アプリ高速コーナー検出アルゴリズム,” 電子情報通信学会論文誌 D-II, vol.J88-D-II, no.6, pp.1035–1046, 2005.
- [3] 山中 賢次, 岩田 基, 黄瀬 浩一, “印刷物を対象とした電子透かし法の検討,” 2013 年暗号と情報セキュリティシンポジウム予稿集, 3F2-1, 2013.

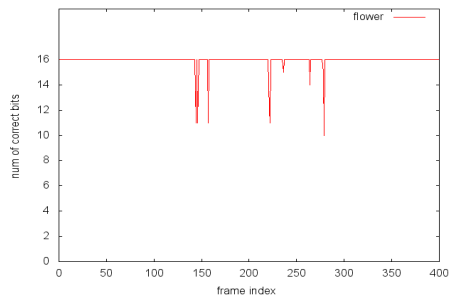


図 8: flower の正解ビット数

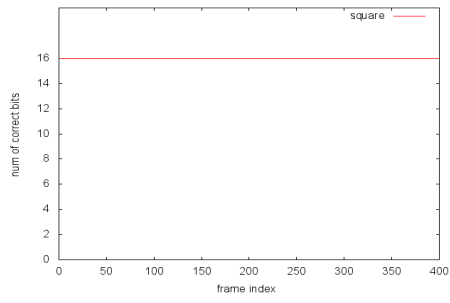


図 9: square の正解ビット数

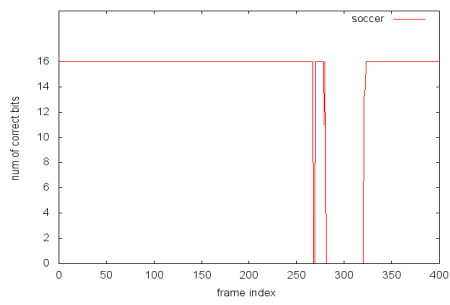


図 10: soccer の正解ビット数

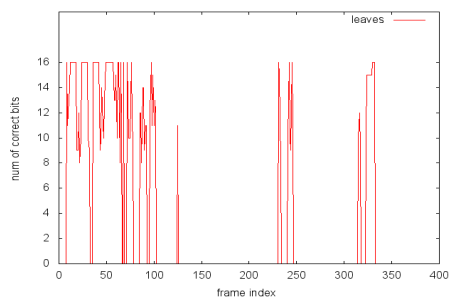


図 11: leaves の正解ビット数

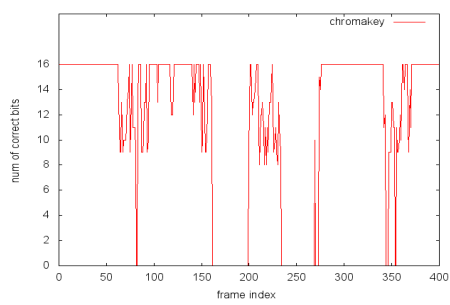


図 12: chromakey の正解ビット数