SCIS 2016 2016 Symposium on
Cryptography and Information Security
Kumamoto, Japan, Jan. 19 - 22, 2016
The Institute of Electronics,
Information and Communication Engineers

スマートフォンを用いた 透かし入り動画から透かし領域を推定する電子透かし法の性能評価

Performance evaluation of digital watermarking method for estimating the watermark area from watermarked videos using a smart phone

水島 尚良 * 岩田 基 † 黄瀬 浩一 †
Naoyoshi Mizushima Motoi Iwata Koichi Kise

あらまし 近年，スマートフォンやデジタルサイネージ上での広告などが普及している．電子透かしを用いて動画に対し付加情報を埋め込んで透かし入り動画を作成し，スマートフォンで再撮影して情報を取り出せる既存の手法に，岩田らの手法がある．岩田らは透かし入り動画にスマートフォンをかざすことによって関連情報にアクセスできるようにすることを目的としており，iPhone5s で透かし入り動画を撮影して透かしを抽出し，性能を評価している．岩田らの手法では今後の課題の 1 つとして，スマートフォンでの撮影時の手振れの影響の軽減が挙げられていた．これに対し，iPhone6sPlus には光学手振れ補正機能が搭載されており，撮影時の手振れのによる抽出誤りの低減が見込まれる．また，岩田らの性能評価ではノート PC 上での映像を対象としていたのに対し，より実用に近い，70 インチのサイネージを実験に使用する．以上より，本稿では，サイネージ上で再生した透かし入り動画を iPhone6sPlus で撮影して透かしを抽出し，性能を評価する．

キーワード 電子透かし 動画 スマートフォン

1 はじめに

近年，スマートフォンやデジタルサイネージの上での広告などの普及拡大が進んでいる．テレビや街頭のサイネージなどで再生される動画をスマートフォンのカメラで撮影することで，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．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，デジタルコンテンツの中に人が知覚できないように何らかの情報を埋め込む技術のことである [1]．以下，埋め込む情報を透かしと呼び，透かしが埋め込まれた動画を透かし入り動画という．

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，撮影環境，幾何歪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．また，ユーザーが耐えられるレスポンスタイムは 2〜8 秒という報告 [2]，[3] があり，抽出処理は高速に行える必要がある．

このような用途に適した従来法として岩田らが，透かしの痕跡を利用して透かし入り動画から透かし領域を推定する電子透かし法 [4] を提案している．以下，この手法を岩田らの手法と呼ぶ．

岩田らの手法では今後の課題の 1 つとして，スマートフォンでの撮影時の手振れによる影響の軽減が挙げられていた．また，岩田らの性能評価ではノート PC 上での映像を対象としていたため，より実用に近いサイネージを用いた際の性能評価が求められる．その他にも埋め込みビット数を増やすことによる抽出への影響などを調べる必要性がある．

そこで本稿では，手振れ軽減のために，光学手振れ補正機能を搭載した iPhone6s Plus を用いて岩田らの手法の性能を検証する．そのため iPhone5s と iPhone6s Plus を用いて実験を行い，抽出性能の比較を行う．また，埋め込みビット数を増やした透かし入り動画からの抽出性能も調べる．このとき，抽出対象となる透かし入り動画はサイネージ上で再生する．以下，2 節で岩田らの手法

* 大阪府立大学工学域, 〒 599-8531 大阪府堺市中区学園町 1-1. College of Engineering, Osaka Prefecture University, 1-1, Gakuen-cho, Naka-ku, Sakai, Osaka, 599-8531 Japan.
† 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.

について説明し，3 節を実験と考察とし，4 節をまとめとする．

2　岩田らの手法

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 $\boldsymbol{w} = \{w_j | w_j \in \{0,1\}, 1 \leq j \leq N\}$ 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 $\boldsymbol{c} = \{c_i | c_i \in \{-1,1\}, 1 \leq i \leq 2N\}$ を得る．

Step2 図 1 に示すような符号語 $\boldsymbol{c}$ 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 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 1 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．

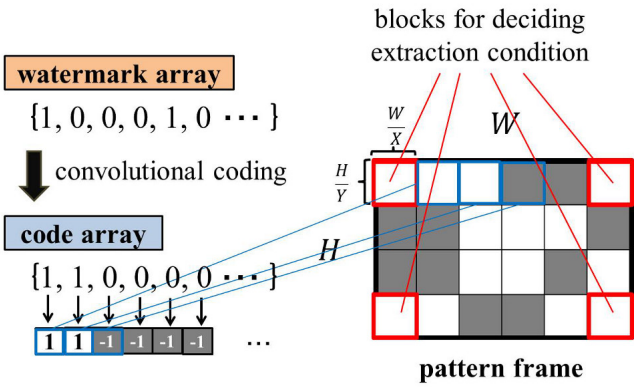

図 1: パターンフレームの作成

- $c_i = 1$ の場合

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

- $c_i = -1$ の場合

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

ここで，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のようになる．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．

2.3 透かし領域推定法

本節では，透かし領域を推定する手法について述べる．岩田らの手法では M 枚の連続した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．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 $\mathrm{B}_x \times \mathrm{B}_y$ 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 $\boldsymbol{a} = \{a_f | 1 \leq f < M-1\}$ 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，上がり下がりを判断する．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．A は上がり幅下がり幅のしきい値である．

$$\begin{aligned} a_f - a_{f+1} &\geq A \\ a_{f+1} - a_{f+2} &\geq A \\ a_{f+2} - a_{f+3} &\leq -A \end{aligned} \tag{3}$$

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．

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．このとき，条件に合う直線がみつからない場合や，交点が画像外の値を取る場合には，Step1 から処理をやり直す．

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．

3　実験と考察

3.1　実験条件

本実験の実験条件として，動画は IHC 委員会が配布している 1920 × 1080 画素で 900 フレーム，30 秒の評価用動画像の Basketball，Lego，Library，Walk1，Walk2 の 5 種類を用いた．これらの動画は IHC 委員会の評価基準 [1] に記載されている FTP サーバーからダウンロード可能である．

透かし入り動画の作成にあたって，透かしの埋め込み対象は Pb 成分で透かし強度は $S = 6$ とし，埋め込む透かしビット列は 0101010100000000 という 16bit と 0101010101010100000000 という 22bit の 2 種類を使用した．そのためパターンフレームの分割数は埋め込みビット数に応じて，16bit の埋め込み時には $X = 6$，$Y = 6$ とし，22bit 埋め込み時には $X = 8$，$Y = 6$ とした．また透かしビット列は動作を安定させるため最後 6 ビットを 0 とした系列を用いた．YPbPr 形式の透かし入り動画像を ffmpeg を用いて 30fps，24Mbps の 30 秒の MP4 形式に変換して再生し実験を行った．

抽出にあたって，推定に用いるフレーム数は $M = 5$ とした．$M = 5$ はおよそ 0.17 秒分のフレーム数にあたる．また推定用ブロックの画素はそれぞれ B_x=5，B_y=5 とし，上がり幅下がり幅のしき値は $A = 1$，パターン合致回数のしきい値は $m = 1$ とした．信頼度判定のしきい値は $R = 3$ とした．

抽出に使用するスマートフォンは iPhone5s と iPhone6s Plus であり，iPhone6s Plus には光学手振れ補正機能が搭載されている．

岩田らの性能評価では透かし入り動画を 2880×1800 画素の解像度を持つ 15 インチの MacbookProRetina ディスプレイモデルに全画面表示し，iPhone5s を使用し手に持った状態で抽出を行っていた．本実験ではより実用に近い条件での抽出を行うため，3840 × 1920 画素の解像度を持つ 70 インチの SHARP 製の PH-H701 上で透かし入り動画を再生し，iPhone5s と iPhone6s Plus を手に持った状態で抽出を行った．撮影フレーム取得時のキャプチャサイズは iPhone5s では 352 × 288 画素，iPhone6s Plus では 640 × 480 画素とした．

また岩田らは透かし入り動画を撮影する際の距離を，液晶テレビの視聴最適距離とされるディスプレイの高さ ×3 として実験を行っていた．MacbookProRetina ディスプレイの高さは約 18cm なのでディスプレイの正面約 54cm の距離から撮影を行っていた．図 2 に岩田らの手法における実験環境を示す．

それに対し，本実験で使用するサイネージの画面の高さは約 90cm なので，サイネージ正面から 270cm 離れた位置で撮影を行った．また最適距離より近い場合と遠い場合についても検証するため 180cm の位置と 360cm の位置からの抽出実験を行った．

[1] http://www.ieice.org/iss/emm/ihc/IHC criteriaVer4.pdf

図 2: 岩田らの手法における実験環境

表 1: 各埋め込みビット数における PSNR[dB]

ビット数	Basketball	Lego	Library	Walk1	Walk2
16bit	36.1380	35.3299	35.2398	35.8725	34.7498
22bit	36.1368	35.3294	35.2382	35.8728	34.7494

透かしの抽出回数は 100 回とし，平均抽出時間と平均試行回数，抽出誤り回数を性能評価に用いた．ここで，抽出試行回数とはビタビ復号時のパスコストが信頼度判定をクリアするまでのフレームの取得から誤り訂正を施した系列を得るまでのループ回数であり，抽出誤り回数とは，得られた系列が信頼度判定をクリアしているにもかかわらず埋め込んだ系列と異なっていた回数である．

3.2　埋め込みビット数による画質の比較

まず，埋め込みビット数による画質の比較を行うため，評価指標として PSNR と SSIM を用いる．原動画と 16bit を埋め込んだ動画間，原動画と 22bit を埋め込んだ動画間の各フレームにおける PSNR と SSIM を算出し，各透かし入り動画像内での平均をそれぞれ求めた．表 1 に PSNR，表 2 に SSIM を示す．

表 1 より埋め込みビット数の増加による PSNR の変化は ±0.02 未満であり影響はほぼないことがわかる．また SSIM に関しては，表 2 より埋め込みビット数の増加に伴い，すべての動画で微小ではあるが低下した．これらの指標よりビット数の増加による画質の劣化は小さいと言える．しかし，埋め込みビット数の増加に伴い，パターンフレームの分割数を増やしたため，22bit を埋め込んだ透かし入り動画は，16bit を埋め込んだ動画に比べてブロック状のノイズが目立った．

3.3　iPhone5s と iPhone6s Plus での抽出性能の比較

iPhone 5s と iPhone6s Plus を用いてサイネージ上で再生された 16 ビットの透かしを埋め込んだ透かし入り

表 2: 各埋め込みビット数における SSIM

ビット数	Basketball	Lego	Library	Walk1	Walk2
16bit	0.99155	0.98744	0.98653	0.99199	0.98685
22bit	0.99063	0.98648	0.98577	0.99120	0.98606

表 3: 抽出性能比較 (距離 180cm)

	評価内容	Basketball	Lego	Library	Walk1	Walk2
5s	抽出時間 [msec]	471.76	434.18	423.62	450.00	573.91
	抽出試行回数	1.31	1.20	1.16	1.28	1.73
6s Plus	抽出時間 [msec]	273.81	334.60	285.59	293.83	375.50
	抽出試行回数	1.05	1.31	1.09	1.13	1.51

動画から透かしの抽出を各動画に対して各距離から 100 回行った．各距離から撮影した際の透かし入り動画の大きさは図 3〜図 5 のようになる．また，表 3〜表 5 に各距離から抽出を行った際の平均抽出時間 [msec] と平均試行回数を示す．抽出誤り回数は透かしの抽出ができなかった iPhone5s を用いて Lego を 360cm の距離から撮影し抽出を行った場合を除き，いずれの距離，動画，抽出端末でも 0 回であった．また距離 270cm の距離から撮影し抽出を行った際の透かし領域推定の平均失敗回数と信頼度判定による平均失敗回数を表 6 に示す．ここで，領域推定失敗回数とは 2.3 節の条件を満たさなかった回数であり，信頼度判定による失敗回数とは領域推定は条件を満たすものの，復号時のパスコストが信頼度判定の条件を満たさなかった回数である．

表 3〜表 5 より，ほぼすべての距離，動画で iPhone6s Plus のほうが抽出にかかる時間は短いという結果が得られた．iPhone6s Plus ではキャプチャサイズが大きくなったため，透かし領域推定のための処理量が増加し，抽出にかかる時間が伸びることが考えられたが，iPhone のスペックの向上により，処理時間の増加は見られなかった．実際，領域推定にかかる時間は iphone5s と比較して約 20[msec] ほど短くなっていた．また推定された領域から符号系列を抽出するのにかかる時間も 90[msec] ほど短くなっていた．抽出試行回数については動画や距離により増減が見られるが，いずれも少ない回数で抽出ができた．また表 6 より，Walk2 を除くすべての動画において透かし領域推定失敗回数が iPhone6s Plus では減少している．キャプチャサイズを大きくすることと手振れの軽減により，透かし領域推定の失敗による試行回数の増加を防いでいると考えられる．

3.4 埋め込みビット数増やした際の抽出性能

文献 [4] に記載された岩田らの性能評価では埋め込みビット数は 16bit としていた．今後埋め込める情報量が増えることが望まれるため，埋め込みビット数を増加させた際の抽出性能を調査し評価する．埋め込みビット数を 22bit に増やし，サイネージ上で再生し各距離から iPhone6s Plus を用いて抽出を 100 回行い，平均抽出時間と平均試行回数，抽出誤り回数を調べた．各距離ごとの平均抽出時間と平均試行回数を表 7〜表 9 に示す．またすべての距離，動画で抽出誤り回数は 0 回であった．

16bit を埋め込んだ透かし入り動画から抽出を行った際の結果をまとめた表 3〜表 5 と，22bit を埋め込んだ透かしり動画から抽出を行った結果をまとめた表 7〜表 9 を比較すると抽出試行回数が増加したものが複数見られる．また抽出試行回数の増加に伴い抽出時間も増加している．16bit の透かし抽出に比べると最大で 370[msec] 程度抽出性能が落ちることもあるが，22bit の透かしの抽出にかかる時間はいずれも 1 秒以内であり，リアルタイムで動作できる．また抽出誤りもないため十分実用的である．

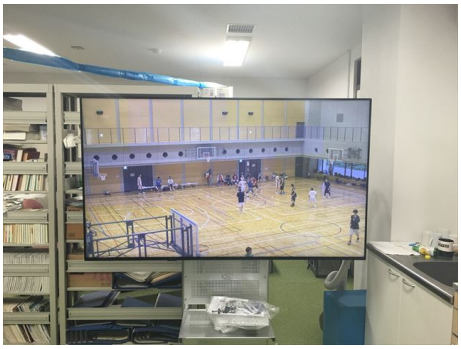

図 3: 距離 180cm

図 4: 距離 270cm

3.5 考察

透かし入り動画の画質に関しては，埋め込みビット数の増加による PSNR や SSIM への影響は小さい．だが PSNR や SSIM ではブロック状のノイズやちらつきなどの画質劣化に対して十分ではないと考えらる．今回の実

図 5: 距離 360cm

表 4: 抽出性能比較 (距離 270cm)

	評価内容	Basketball	Lego	Library	Walk1	Walk2
5s	抽出時間 [msec]	483.06	643.56	486.75	721.44	641.60
	抽出試行回数	1.32	1.81	1.34	2.03	1.84
6s Plus	抽出時間 [msec]	309.54	686.05	319.13	310.90	526.27
	抽出試行回数	1.19	2.63	1.23	1.19	2.09

験では埋め込みビット数を増やすためにパターンフレームの分割数を増やしたが，それにより目視では埋め込みビット数を増やす前よりブロック状のちらつきが目立っていた．

また iPhone5s と iPhone6s Plus での性能比較実験では光学手振れ補正や撮影時のキャプチャサイズを大きくすることで，透かし領域の推定失敗が減少し，それに応じて抽出試行回数が減ったこととスペックの向上による処理速度の向上により iPhone6s Plus では抽出時間が短くなった．しかし，表 4 において Lego や Walk2 では iPhone6s Plus を用いた際，抽出試行回数が増加した．これは，表 6 より信頼度判定の条件を満たさない回数が増加したことが原因だと考えられ，キャプチャサイズの拡大が影響を与えている可能性が考えられる．

ビット数増加による抽出性能評価では，22bit を埋め込んだ動画からの抽出では複数の動画で 16bit を埋め込んだ動画よりも抽出試行回数の増加がみられた．これはパターンフレームの分割数が増え符号語 1 ビットが埋め込まれているブロックサイズが小さくなったことにより，得られた符号語に誤りが生じやすくなり信頼度判定をクリアできないことが増えたことが原因として考えられる．そのためビット数をさらに増やした際には抽出試行回数が増えることが考えられる．

今回の実験結果から光学手振れ補正機能を持つ iPhone6s Plus を用いることで動画や撮影距離，埋め込みビット数によらず，より実用的な条件下で誤りなく 1 秒に抽出以内の可能であることが示された．

表 5: 抽出性能比較 (距離 360cm)

	評価内容	Basketball	Lego	Library	Walk1	Walk2
5s	抽出時間 [msec]	536.72	-	596.28	1005.00	595.39
	抽出試行回数	1.49	-	1.66	2.76	1.63
6s Plus	抽出時間 [msec]	469.37	515.66	362.66	371.77	529.37
	抽出試行回数	1.81	1.97	1.40	1.42	2.07

表 6: 抽出試行時の平均失敗回数の内訳 (距離 270cm)

	内訳	Basketball	Lego	Library	Walk1	Walk2
5s	領域推定失敗	0.07	0.17	0.08	0.26	0.25
	信頼度判定による失敗	0.31	0.76	0.31	0.88	0.74
6s Plus	領域推定失敗	0.03	0.10	0.04	0.01	0.26
	信頼度判定による失敗	0.18	1.57	0.20	0.18	0.94

4 終わりに

本稿では，岩田らの手法を取り上げ，課題として挙げられていた手振れ補正による抽出性能の評価と埋め込みビット数を増やした際の抽出性能を評価した．16bit と 22bit の埋め込みでは PSNR と SSIM の値に大きな変化は見られなかったが，パターンフレームの分割数の変更によるブロック状のノイズが目立った．抽出性能に関しては，手振れ補正機能をもち，スペックの向上した iPhone6s Plus を用いて iPhone5s と比較することで評価を行った．領域推定で 20[msec]，推定された領域からの符号系列抽出で 90[msec] 程度の処理時間の短縮と，手振れ補正により領域推定失敗による抽出試行回数の増加を抑られるという結果が得られた．埋め込みビット数の増加による抽出性能の評価では，多少抽出時間が長くなることがあるが，いずれも 1 秒以内に抽出が可能で実用性があると言える結果となった．

今後の課題として，透かし入り動画の画質向上やさらなる埋め込みビット数の増加に対する埋め込みや抽出への影響を調べることが挙げられる．動画全体にパターンフレームを埋め込む岩田らの手法ではブロック状のノイズが目立つため，埋め込みの際に動きベクトルを利用することで埋め込み領域を狭めるなど知覚されにくいように改善していく必要がある．

謝辞

本研究は，JST 研究成果最適展開支援プログラム（A-STEP）探索タイプ（課題番号：AS262Z02045H）の助成を受けたものです．

参考文献

[1] 松井甲子雄, “電子透かしの基礎,” 森北出版, 1998.

表 7: 22bit 埋め込みに対する抽出性能 (距離 180cm)

評価内容	Basketball	Lego	Library	Walk1	Walk2
抽出時間 [msec]	340.52	701.44	307.482	314.48	354.44
抽出試行回数	1.30	2.74	1.18	1.22	1.35

表 8: 22bit 埋め込みに対する抽出性能 (距離 270cm)

評価内容	Basketball	Lego	Library	Walk1	Walk2
抽出時間 [msec]	292.14	534.67	302.60	417.13	311.50
抽出試行回数	1.12	2.03	1.16	1.59	1.19

[2] Zona Research, “The need for speed II,” Zona Market Bulletin, Issue 05, http://www.keynote.com/downloads/Zona_Need_For_Speed.pdf, 2001.

[3] S. Shunmuga Krishnan and Ramesh K. Sitaraman, “Video stream quality impacts viewer behavior: inferring causality using quasiexperimental designs,” IMC ’12 Proceedings of the 2012 ACM conference on Internet measurement conference, pp.211–224, 2012.

[4] 岩田 基，黄瀬 浩一, “透かしの痕跡を利用して再撮影動画像から透かし入り動画像の領域を推定する電子透かし法の実装実験,” 日本画像学会誌, vol.53, no.5, pp.436-447, 2014.

表 9: 22bit 埋め込みに対する抽出性能 (距離 360cm)

評価内容	Basketball	Lego	Library	Walk1	Walk2
抽出時間 [msec]	308.39	550.37	309.53	386.60	365.42
抽出試行回数	1.18	2.15	1.19	1.48	1.40