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

画像検索を用いて位置検出用マーカのない印刷物から透かしを抽出できる電子透かし法

Digital Watermarking Method Using Image Retrieval to Printed Images without Finder Patterns and Alignment Pattern

小川 広晃 * 岩田 基 * 黄瀬 浩一 *
Hiroaki Ogawa Motoi Iwata Koichi Kise

あらまし 電子透かしを用いて，印刷物に人間には認識できないように情報を埋め込み，それをスマートフォンで撮影して情報を取り出す手法について検討する．SCIS2013 に投稿した手法を基に手法の改良を行った．従来の電子透かし法では位置検出のためにマーカや，枠線をつける必要があった．本来，電子透かしの利点の一つは，見た目を損なわないという点にあるのだが，マーカや，枠線をつけることはその利点と相反している．本稿では，この問題点を解決するために，位置検出の手法を改善し，画像に何も付与せずに位置検出し，透かしから情報を取り出す手法を提案する．画像に何も付与せずに位置検出を行うために，画像検索を用いて撮影画像内にのどの位置に透かし入り画像が存在するのかを検出し，位置検出を行う．画像検索は計算コストが高いため，位置検出はスマートフォン上でなくサーバ上で行っている．位置検出法を変更した結果，従来法と比べ処理時間について従来法が 0.34 秒だったのに対して 0.9 秒を要するようになったものの，同等の抽出率を保持できた．

キーワード 電子透かし　印刷　Android

1 はじめに

電子透かしとはデジタルコンテンツを人間には知覚できないように変更して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画像を透かし入り画像と呼ぶ．

印刷物から携帯端末で情報を読み取るようなシステムとしてよく使われているものに QR コードがある．QR コードは大容量で読み取りも早い一方でコンテンツ以外に QR コードを表示するスペースが必要な点や，白黒パターンのため見た目が悪いという問題点がある．一方，その問題点を解決する技術として，電子透かしがある．電子透かしの場合は，コンテンツと情報を一体化させるため新たなスペースを必要とせず，またコンテンツそのものが情報を表すため見た目を損なわない．過去に，電子透かしを用いて印刷物に情報を埋め込む手法はいくつか提案されている [2][3][4][5]．しかし，一般的な電子透かしは抽出の際に複雑な画像処理を伴うため計算量が多く，読み取り端末に処理能力の低い携帯端末を用いると読取りに時間がかかるため実用的ではない．文献 [6] では抽出処理の計算量が少なく携帯端末でも高速に抽出処理が可能な電子透かし法が提案されている．以下，この手法を従来法と呼ぶ．従来法は携帯端末のみで透かしを取り出すことができ，処理時間が 1 秒以内であるという特徴がある．従来法では，QR コードの四隅に配置されている位置検出用マーカを用いて位置検出を行っている．

本来，電子透かしの利点の一つは，見た目を損なわないことだが，従来法では，画像の四隅に位置検出用マーカを設置する必要がある．そのため，QR コードの問題点である見た目が悪くなるという点を完全に解決することはできていない．

そこで本論文では，この問題点を完全に解決するために，位置検出の手法を改善し，画像に何も付与しないで位置検出することのできる手法を提案する．提案法では，透かしを埋め込む対象となる原画像をあらかじめデータベースに登録し，画像検索によって撮影した透かし入り画像がデータベース上のどの画像かを検索する．ここで，透かし入り画像はデータベースに登録しない．このとき、２つの画像の特徴量を抽出し，データベースの画像の四隅を撮影画像の四隅に変換するような射影変換行列を作成する．この射影変換行列に沿ってデータベース画像を

* 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 Gakuencho, Naka-ku, Sakai, Osaka 599-8531, Japan

図 1: パターン付き透かし入り画像例

変換することによって，撮影画像の透かし入り領域を検出できる．この位置検出手法を用いることによって，位置検出用マーカを透かし入り画像に付与することなく位置検出を行うことができるため，見た目を損なわない．

実験では従来法と提案法の抽出成功率と処理時間を比較することで，提案法の有用性を示す．実験の結果，提案法は従来法のような位置検出用マーカーを必要としないにもかかわらず，従来法と同等の抽出成功率が得られることを示した．反面，提案法は従来法と比べ，高角度で正しく透かしを抽出することができなくなることを確認した．一方処理速度においては，データ通信する時間が追加されたなどの理由により，提案法は従来法と比べて処理時間が増加したが，提案法の処理速度であっても，文献 [7] で示されているユーザーの利便性を損ねる処理時間ではないことを示した．

以下，2 章で従来法について述べる．3 章で提案法について述べ，4 章で実験結果と考察を示し，5 章でまとめとする．

2 従来法

本章では山中らの手法 [6] で示されている，カメラ付き携帯端末を用いた，印刷物を対象とした電子透かし法について述べる．この手法は，携帯端末のみで正しい透かしを取り出せること，処理時間が 1 秒以内であることなどが特徴である．

印刷された透かし入り画像から透かしを取り出す場合，撮影しているどの部分に透かしが埋め込まれているのかを判断する必要がある．従来法では，図 1 のように透かし入り画像の四隅に QR コードの位置検出マーカ配置し，それを用いて位置を検出している．位置検出マーカを用いることで性能の低い携帯端末上でも高速認識を可能にしている．以下，2.1 節で埋め込み法，2.2 節で抽出法について述べる．

2.1 埋め込み法

透かし埋め込み処理について説明する．原画像 I，長さ 4 ビットの透かし情報，埋め込み強度 a の三つを入力とする．ここで，I は $W \times H$ 画素のカラー画像 $I = \{I^R_{x,y}, I^G_{x,y}, I^B_{x,y}\}(0 \leq x < W, 0 \leq y < H)$ で，埋め込み強度とは画質と耐性のバランスを調整するパラメータである．埋め込み強度を大きくすると，透かし入り画像の中の透かし成分が強くなり，画質は劣化する．

step1. 以下では 4 ビットの透かし情報を誤り訂正符号化し 32 ビットの透かしビット列を作成する手順を示す．透かし情報は 4 ビットであるため，16 通りの情報を表すことができる．

step1-1. 系列長が 32 ビットの透かしビット列 $\boldsymbol{\phi^m} = \{\phi^m_j | \phi^m_j \in \{0,1\}, 0 \leq j < 32\}$ を 16 種類用意する $(m = 0, 1, \cdots, 15)$．$\boldsymbol{\phi^m}$ はアダマール行列から生成した系列を用いる．アダマール行列から得られる系列は直交するため，$0 \leq A, B < 16, A \neq B$ を満たす全ての A と B の組み合わせにおいて，ϕ^A と ϕ^B のハミング距離は 16 となる．

step1-2. 透かしビット列 $\boldsymbol{\phi^m}$ と，4 ビットの透かし情報とを 1 対 1 で対応させる．

step2. 以下では透かし系列を周波数成分に変換する手順を示す．

step2-1. あらかじめ定めた整数値 N を用いて，$\boldsymbol{\phi^m_j}$ を $l = N^2/32$ 回繰り返し引き延ばして得られる長さ N^2 の系列を

$$\boldsymbol{b} = \overbrace{\phi^m_0, \ldots, \phi^m_0}^{l}, \phi^m_1, \cdots, \overbrace{\phi^m_{31}, \ldots, \phi^m_{31}}^{l} \quad (1)$$

とする．この際，$\phi^m_j = 0$ の場合は -1，$\phi^m_j = 1$ の場合は $+1$ と読み替えて $\boldsymbol{b} = \{b_k | b_k \in \{0,1\} | 0 \leq k < N^2\}$ を構成する．長さ l の擬似乱数列 $r^{(j)} = \{r^{(j)}_i | r^{(j)}_i \in \{-1, +1\}, 0 \leq i < l, \sum_{i=0}^{l-1} r^{(j)}_i \approx 0\}$ を $j = 0, 1, \ldots, 31$ について用意し，これらを並べた長さ N^2 の系列を，

$$\boldsymbol{r} = r^{(0)}_0, \ldots, r^{(0)}_{l-1}, \cdots, r^{(31)}_0, \ldots, r^{(31)}_{l-1} \quad (2)$$

とする．

step2-2. 式 (3) により直接拡散変調を行い $\boldsymbol{s} = \{s_k | 0 \leq k < N^2\}$ とする．

$$s_k = b_k r_k \quad (3)$$

step2-3. 擬似ランダムに決定される置換

$$\begin{pmatrix} 0 & 1 & 2 & 3 & \cdots & N^2-1 \\ o_0 & o_1 & o_2 & o_3 & \cdots & o_{N^2-1} \end{pmatrix} \quad (4)$$

を用意しこの置換を用いて s の並びを式 (5) に従いスクランブルすることにより，埋め込み系列 $t = \{t_k | 0 \leq k < N^2\}$ を算出する．ここで，系列 o_k は $0 \sim N^2 - 1$ をランダムに並び替えて得られる系列である．

$$t_k = s_{o_k} \quad (5)$$

step3. 図 2 に示すように，原画像と同じ大きさの，画像サイズに対する相対周波数が F である 90 度回転対称の二つの 2 次元サインカーブ P^-, P^+ を生成する．あらかじめ定めた整数値 N を用いて，P^-, P^+ をそれぞれ $N \times N$ 個のブロックに分割することで，2 種類のブロックパターン $\{P_{x,y}^{-(h,v)}\}, \{P_{x,y}^{+(h,v)}\}$ を得る．(h, v) はブロックの位置であり，$0 \leq h < N, 0 \leq v < N$ ある．また (x, y) は $W \times H$ 画素のカラー画像 I の画素値の座標 $(0 \leq x < W, 0 \leq y < H)$ である．ラスタスキャン順に，埋め込み系列の要素 $t_k = t_{h+vN}$ を選び，t_k の値に応じて透かしパターンのブロックパターン $P_{x,y}^{(h,v)}$ を以下のように選択する．

$$P_{x,y}^{(h,v)} = \begin{cases} P_{x,y}^{-(h,v)} & \text{if } t_k = -1 \\ P_{x,y}^{+(h,v)} & \text{if } t_k = +1 \end{cases} \quad (6)$$

すべてのブロック位置 (h, v) に対してこの作業を行うことにより，透かしパターン $P = \{P_{x,y}^{(h,v)}\}$ を得る．

step4. 図 3 で示すように，式 (7) に従い，埋め込み強度 a を透かしパターン P に乗算して原画像 I の各色成分に加算し，透かし入り画像 $I' = \{I_{x,y}^{R\prime}, I_{x,y}^{G\prime}, I_{x,y}^{B\prime}\}$ を得る．

$$\begin{aligned} I_{x,y}^{R\prime} &= I_{x,y}^{R} + aP_{x,y} \\ I_{x,y}^{G\prime} &= I_{x,y}^{G} + aP_{x,y} \\ I_{x,y}^{B\prime} &= I_{x,y}^{B} + aP_{x,y} \end{aligned} \quad (7)$$

2.2 抽出法

step1. 以下の手順に従い，透かし入り領域を推定する．google が提供している QR コード読み取りライブラリである ZXing[1] を用いて画像の四隅を検出する．この四隅で囲まれた部分を透かし入り領域とする．

[1] https://code.google.com/p/zxing/

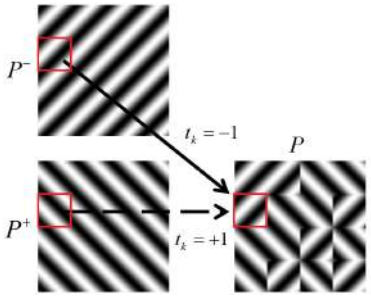

図 2: 2 次元サインパターン変調

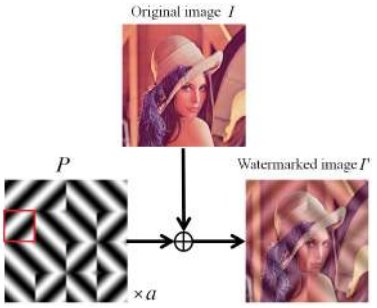

図 3: パターン重畳

step2. 抽出対象画像の R，G，B の平均値を取り，画像をグレースケール化する．グレースケール化した画像を $L \times L$ の正方形に正規化する．ここで，L は埋め込み時に用いた相対周波数のパラメーター F を用いて $L = 4F$ と定める．$L = 4F$ と定めるのは，step3 の透かし強調処理を原画像のサイズに依存することなく行うためである．

step3. 以下の手順に従い，画像に対して透かし強調処理を行う．正規化後の抽出対象画像に対して図 4 の畳込みオペレータによる畳込み処理を施す．これによって原画像信号を低減させ，透かし信号を強調させる．埋込み時に用いた P^-, P^+ の波長は，$L = 4F$ により，正規化後の画像では x, y 方向とも 4 となる．そこで，図 4 の畳込みオペレータを用いると，P^-, P^+ の両方に対して一度の操作で強調することができる．

step4. 原画像中で 2 次元サインパターンと同じようなテクスチャをもつ部分では，原画像の振幅が透かしパターンの振幅よりも大きくなってしまう．そこで以下の手順に従い，重み付け処理を行う．透かし強調処理後の画像中のすべての画素のブロック $B_{h,v}$ に対してブロック内の画素値 p の絶対値の平均 $q_{h,v}$ を次式に従って求める．

$$q_{h,v} = \frac{1}{M^2} \sum_{p \in B_{h,v}} |p| \quad (8)$$

そして $q_{h,v}$ に基づいて位置 (h,v) のブロック $B_{h,v}$ における抽出値の重み $w_{h,v}$ を

$$w_{h,v} = f(q_{h,v}) \tag{9}$$

と定める．重み付け関数である $f(x)$ は単調減少関数である．これにより絶対値の大きなブロックに対する抽出値の重みを小さくすることができ，原画像中で 2 次元サインパターンと同じようなテクスチャをもつ部分の信号を低減することができる．

step5. step3 で処理を施された画像に対して画素値が正であれば +1，負であれば −1，0 であれば 0 にするクリップ処理を行う．図 1 の画像に対して透かし強調処理，重み付け処理，クリップ処理を行ったときの画像例を図 5 に示す．この図では変化がわかりやすいように画素値が −1 であれば 0，0 であれは 128，+1 であれば 256 に置換している．以下透かし強調処理，重み付け処理，クリップ処理をまとめて前処理と呼ぶ．

step6. 以下の手順に従い，二次元サインパターンの復調を行う．$N \times N$ 個のブロックに分割し，$M \times M$ 画素の各ブロックについて，図 6 に示す畳込み行列 C^- と C^+ で前処理後の画像に畳込み処理を行った後の画素値をそれぞれ，$e^{-(h,v)}_{x',y'}, e^{+(h,v)}_{x',y'}$ とする．(h,v) はブロックの位置であり，$0 \le h < N, 0 \le v < N$ である．また (x',y') は $M \times M$ 画素のブロック内での座標 $(0 \le x' < M, 0 \le y' < M)$ である．埋め込みに用いた P^-, P^+ に対応する周波数エネルギー $E^-_{h,v}, E^+_{h,v}$ を式 (2.9)，式 (2.10) によって求める．

$$E^-_{h,v} = \sum_{x'=0}^{M-1}\sum_{y'=0}^{M-1} |e^{-(h,v)}_{x',y'}| \tag{10}$$

$$E^+_{h,v} = \sum_{x'=0}^{M-1}\sum_{y'=0}^{M-1} |e^{+(h,v)}_{x',y'}| \tag{11}$$

$E^-_{h,v}$ と $E^+_{h,v}$ の差を $d_{h,v}$ とする．

$$d_{h,v} = E^+_{h,v} - E^-_{h,v} \tag{12}$$

$d_{h,v}$ をすべてのブロックに対して求めることによって，抽出値行列 $D = \{d_{h,v}\}$ を得る．

step7. 以下では周波数成分を透かし系列に変換する手順を示す．

step7-1. 抽出値行列 D の要素をラスタスキャン順に並べた 1 次元行列を $g = \{g_k | 0 \le k < N^2\}$ とし，式 (4) のランダム置換を用いて g の並びを式 (2.13) に従いデスクランブルして $h = \{h_k | 0 \le k < N^2\}$ を得る．

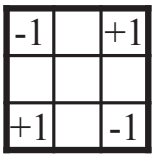

図 4: 前処理フィルタ用畳込みオペレータ

$$h_{o_k} = g_k \tag{13}$$

step7-2. h の長さ $l = N^2/n$ の部分列 $x^{(j)} = \{x^{(j)}_i | 0 \le i < l\}$ を

$$x^{(j)}_i = h_{i+jl} \tag{14}$$

として得る．

step7-3. $x^{(j)}$ の各要素を平均 0 分散 1 になるように正規化したものを，改めて抽出対象系列 $x^{(j)}$ と定義する．そして埋め込みに用いた擬似乱数列 $r^{(j)}$ と $x^{(j)}$ の相関値 ρ_j を以下の式 (15) ですべての j について求める．

$$\rho_j = \sum_{i=0}^{l-1} x^{(j)}_i r^{(j)}_i \tag{15}$$

step7-4. 透かし系列 $\boldsymbol{\varphi} = \{\varphi_j | \varphi_j \in \{0,1\} | 0 \le j < 32\}$ を

$$\varphi_j = \begin{cases} 0, & \text{if } \rho_j < 0 \\ 1, & \text{if } \rho_j \ge 0 \end{cases} \tag{16}$$

のようにして抽出する．

step8. 以下の手順に従い，誤り訂正を行う．$\boldsymbol{\varphi}$ と ϕ^m とのハミング距離を，全ての ϕ について求める．そして，ハミング距離が最小でそのときのハミング距離が th 以下になる ϕ^m が存在する場合，その ϕ^m に対応する透かし情報を取り出す．もし，th 以下になる ϕ^m が存在しない場合は抽出失敗とする．

3 提案法

従来法には，携帯端末のみで透かしを抽出可能であったり，処理時間が 1 秒以内であるという特徴があったが，一方で画像の四隅に位置検出用マーカを付与しなければならないという問題点があった．そこで，提案法では，画像検索を用いた位置検出を行うことで，画像の四隅に位置検出マーカを付与せずに透かしを取り出す方法を提案する．ただし提案法では携帯端末のみでは透かしを抽出できず，サーバとの通信を必要とする．

図 5: 前処理フィルタ後の画像例

		-1		
	-1		+2	
-1		+2		-1
	+2		-1	
		-1		

C^-

		-1		
	+2		-1	
-1		+2		-1
	-1		+2	
		-1		

C^+

図 6: 2 次元サインパターン復調のための二つの畳込み行列

3.1 埋め込み法

2.1 節に従い，画像に透かしを埋め込む．その後，透かしを埋め込んだ原画像の SIFT 特徴量 [7] をデータベースに登録する．SIFT 特徴量とは回転や照明変化に頑健な特徴量である．

3.2 抽出法

step1. 携帯端末で透かし入り画像を撮影し，撮影した画像を携帯端末からサーバに送る．

step2. 以下の手順に従い，撮影した画像内にどのデータベース画像が含まれているかを判断する．

step2-1. 撮影した画像から SIFT 特徴量を抽出する．

step2-2. 各特徴量について近似最近傍探索を用いてデータベースから最近傍を求める．ここで，近似最近傍探索には BDH[8] を用いる．

step2-3. 最近傍となった特徴量を含むデータベース画像に投票する．このときに求められた特徴量間の距離を，投票されたデータベース画像ごとに，特徴量間距離のリストとして保存しておく．

step2-4. 最も多く投票された画像を検索結果画像とする．

step2-5. 得られた検索結果画像について，step2-3 で保存しておいた特徴量間距離のリストを調べる．最も短い距離より S 以上長い距離となった特徴量は誤検出として無視する．その結果，特徴量が T 以上のこらなかった場合，位置検出失敗と判断し，step1 に戻る．S, T はあらかじめ定めた整数値である．

step3. 以下の手順に従い，位置検出に必要な射影変換行列を得る．

step3-1. 検索結果画像の全特徴量から特徴量を 4 つ取り出す．検索結果画像の特徴点の座標を (x_1, y_1)〜(x_4, y_4)，撮影した画像の対応する特徴点の座標を (u_1, v_1)〜(u_4, v_4) とする．また射影変換行列 H のパラメータを h_1〜h_8 とする．

$$H = \begin{pmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & 1 \end{pmatrix} \tag{17}$$

step3-2. 以下の 8 次元連立一次方程式を解き，パラメータ h_1〜h_8 を求める．

$$\begin{aligned}
u_1 &= x_1h_1 + y_1h_2 + h_3 - x_1h_7u_1 - y_1h_8u_1 \\
v_1 &= x_1h_4 + y_1h_5 + h_6 - x_1h_7u_1 - y_1h_8u_1 \\
u_2 &= x_2h_1 + y_2h_2 + h_3 - x_2h_7u_2 - y_2h_8u_2 \\
v_2 &= x_2h_4 + y_2h_5 + h_6 - x_2h_7u_2 - y_2h_8u_2 \\
u_3 &= x_3h_1 + y_3h_2 + h_3 - x_3h_7u_3 - y_3h_8u_3 \\
v_3 &= x_3h_4 + y_3h_5 + h_6 - x_3h_7u_3 - y_3h_8u_3 \\
u_4 &= x_4h_1 + y_4h_2 + h_3 - x_4h_7u_4 - y_4h_8u_4 \\
v_4 &= x_4h_4 + y_4h_5 + h_6 - x_4h_7u_4 - y_4h_8u_4
\end{aligned}$$

step3-3. 検索結果画像の全ての特徴点から取り出した 4 つの特徴点を除いた特徴点の座標に求めた射影変換行列を当てはめる．そうして得られた座標と，撮影した画像の対応する特徴点の座標の距離を求める．

step3-4. step3-1〜3-3 を全ての特徴量の取り方について繰り返し，step3-3 で得られる距離が最小となったときの射影変換行列を位置検出に用いる．これによって誤検出された特徴量を使わないような射影変換行列を作成することができる．

step4. 検索結果画像の四隅の座標を step3 で求めた射影変換行列で変換することで撮影した画像内の透

図 7: 原画像

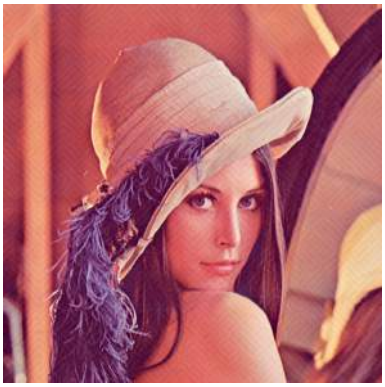

図 8: 透かし入り画像

かしが埋め込まれている領域の四隅の座標が得られる．この四隅で囲まれた領域を透かし入り領域と呼ぶ．

step5. 透かし入り領域をあらかじめ決めておいたサイズの正方形になるように平面射影変換する．その後，正方形の領域を切りだして抽出対象画像とする．

step6. 抽出対象画像をサーバから携帯端末に送る．

step7. 2.2 節に従って抽出対象画像から透かしを抽出する．

4 実験と考察

従来法と提案法について，抽出成功率と処理時間の性能の比較実験を行った．

4.1 抽出成功率の比較実験

本節では，提案法で推定された四隅の精度と従来法で推定された四隅の精度，さらに両手法の抽出成功率を比較することにより提案法の有効性を示す．抽出成功率とは透かしを抽出した際に正しい透かしが抽出できた比率のことであり，次の式で示される．

$$\text{抽出成功率} = \frac{\text{正しい透かしが抽出できた回数}}{\text{透かしを抽出した総回数}}$$

4.1.1 実験条件

原画像は 512 × 512 [pixel] の 1 種類 (lenna)，印刷サイズ 15cm 四方で印刷した．図 7，図 8 に原画像と透かし入り画像を 15cm 四方で印刷したものをそれぞれ示す．印刷した紙を壁に貼り付け，正面左 60 度から正面右 60 度まで 15 度刻みで角度をつけて撮影を行った．図 9 に示すように，撮影はフリーハンドで行った．50 回の試行において，抽出成功率を求めた．

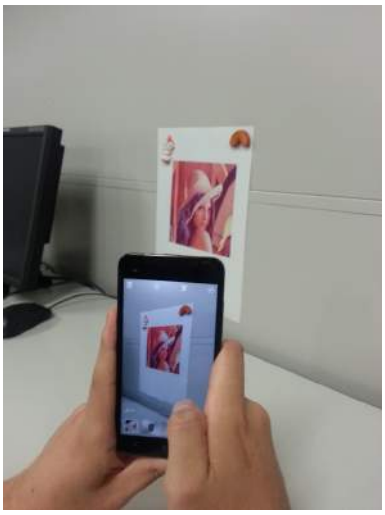

図 9: 撮影風景

4.1.2 実験結果

表 1 に抽出成功率を示す．×は何も抽出されなかったことを表している．表 1 が示すように，従来法では左右 30 度まで，提案法では正面で正しい透かしを 100%抽出することができた．ただし，提案法では 45 度まで抽出成功率 92%以上で抽出することができている．今後，我々は従来法のように連続的に画像を取得して透かしを抽出する手法にしようと考えている．その場合，画像を複数回送信し，同じ透かしを抽出できたときのみ抽出成功とするなどすれば，30 度や 45 度の抽出成功率は 100%に限りなく近づく．例えば抽出成功率が 92%のとき，2 回の試行で同一の結果が得られれば抽出成功としたとき，抽出成功率は 99.99%となる．

4.2 処理時間の比較実験

本節では，従来法と提案法で撮影から透かし抽出までの処理時間の比較を行い，提案法の有用性を示す．

表 1: 抽出成功率

角度	左			正面	右		
	60 度	45 度	30 度		30 度	45 度	60 度
従来法	×	×	100%	100%	100%	×	×
提案法	×	98%	100%	100%	94%	92%	×

表 2: 処理時間 [ms]

	データ送信	四隅検出	抽出処理	合計
従来法	無し	56	284	340
提案法	224	392	294	900

4.2.1 実験条件

提案法の画像検索に用いるデータベースには 1000 枚の画像を用意した．山中らの手法の実験には Galaxy S3 GT-I9300 を用い，提案法の実験には同じ端末と，CPU は Opteron6238 2.80GHz，メモリは 512GB のサーバを用いた．従来法では端末上で全ての処理を行い，提案法では端末上で抽出処理，サーバ上で四隅検出を行う．図 9 に示すように透かし入り画像を印刷したものを壁に貼り付け通常の室内照明下で撮影し，透かしを抽出するまでの処理時間を測定した．50 回試行し，その平均を取った．また，提案法の実験結果はそれぞれ四隅検出，抽出処理が端末，サーバで独立に処理されたときの時間である．

4.2.2 実験結果

表 3 に処理時間の実験結果を示す．提案法は従来法に比べ，データ送信にかかる時間が追加されている．さらに画像検索を用いた位置検出はデータベース内の画像検索に時間がかかるため，四隅のマーカーを用いた位置検出より時間がかかる．以上より合計の処理時間は増加した．文献 [7] ではユーザーの利便性を考えるとプログラムの応答時間は 2 秒以内でなければならないと示されている．提案法の処理時間は十分使用に耐えうるといえる．また 4.1.2 節で述べたように，将来 2 回の試行で同一の結果が得られれば抽出成功としたとき，処理時間は 2 倍の 1.596 秒となる．2 回の試行で同一の結果が得られれば抽出成功としたときでも，2 秒以内に透かしを抽出できているので，これも十分使用に耐えうる処理時間だといえる．

5 おわりに

本論文では文献 [6] の従来法で用いられている，位置検出マーカを用いた透かし抽出法の問題点を解決するために，位置検出に画像検索を用いる手法を提案した．そして，提案法と従来法の性能を実験に基づき比較した．実験により，提案法は従来法のような位置検出用マーカーを必要としない代わりに，従来法と比べ抽出可能な撮影角度が狭いことを確認した．また処理速度は従来法よりも遅いものの，実用可能な程度に高速であることを確認した．今後の課題として 4.1.2 節で挙げた連続的に透かし抽出を行うことによる抽出成功率の向上などが挙げられる．

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.

[2] 堀内陽一郎, 棟安実治, “拡散方式と DCT を用いた印刷画像へのデータ埋め込みの一手法, ” 電子情報通信学会技術研究報告, vol.105, no.295, pp.19–24, 2005.

[3] 水本匡, 松井甲子雄, “DCT を用いた電子透かしの印刷取込み耐性の検討, ” 電子情報通信学会論文誌, vol.85, no.4, pp.451–459, 2002.

[4] 中西康二, 棟安実治, “携帯電話を用いたデータ埋込印刷画像からのデータ検出, ” 電子情報通信学会技術研究報告, vol.107, no.547, pp.51–56, 2008.

[5] 正野隆文, 棟安実治, 花田良子, “携帯電話による情報検出を目的とした印刷画像へのデータ埋め込み, ” 電子情報通信学会技術研究報告,vol.109,no.78,pp.13–18,2009.

[6] 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, ” SCIS2013 論文集, 3F2-1, 2013.

[7] R.B.Miler, “Response time in man-computer conventional transactions,” Proc. AFIPS Fall Joint Computer Conference, vol.33, pp,267, 1968.

[8] N.C.Pauline, and H.Steven. “SIFT: Predicting amino acid changes that affect protein function,” Nucleic acids research 31.13, 3812–3814, 2003.

[9] M. Iwamura, T. Sato, and K. Kise, “What is the most efficient way to select nearest neighbor candidates for fastapproximate nearest neighbor search?,” Proc. 14th International Conference on Computer Vision (ICCV 2013),pp.535–542, 2013.