SCIS 2013 The 30th Symposium on
Cryptography and Information Security
Kyoto, Japan, Jan. 22 - 25, 2013
The Institute of Electronics,
Information and Communication Engineers

印刷物を対象とした電子透かし法の検討
A consideration of Digital Watermarking Method to Printed Images

山中　賢次 *　　岩田　基 *　　黄瀬　浩一 *
Kenji Yamanaka　　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．透かし入りの印刷物にユーザーがスマートフォンをかざすことで情報を得られることを目指している．これを実現するには，スマートフォンをかざしている間，撮影と，情報の抽出処理を並行して行う必要がある．そのため，従来の電子透かしでは重視されていなかった抽出処理の高速性が求められる．ほかの要件としては，撮影映像の解像度が高いと抽出処理に時間がかかるため，解像度が低い撮影映像から情報を抽出できなければならない．また，印刷の際には色合いの変化が生じたり，撮影の際には手ブレや位置ずれなどが生じたりするため，これらの外乱に耐性を持たせなければならない．

キーワード 電子透かし　印刷　Android

1　はじめに

近年，デジタルコンテンツの著作権保護技術として，電子透かしが盛んに研究されている．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よび，透かしが埋め込まれた画像を透かし入り画像とよぶ．電子透かしの特性としては，透かし入り画像の画質劣化を知覚できないこと以外に，透かし入り画像に対して，雑音付加や変形などの様々な変更を施しても，透かしを正しく取り出せることがある．ある変更が施された後でも透かしを正しく取り出せるとき，その変更に対して耐性をもつ，と表現する．

文献 [2] では，電子透かしを応用し，印刷された透かし入り画像から透かしを取り出し，透かしを用いて画像に関する情報へ導くシステムが提案されている．同様なシステムとしては 2 次元コードがあるが，欠点として，コンテンツ以外に 2 次元コード出力領域が必要な点，利用者がコンテンツと 2 次元コードの対応を意識しなければならない点がある．一方で，電子透かし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，またコンテンツを撮影することによって，そのコンテンツに関する情報を得ることができ，ユーザーに直感的な操作感を提供できる．

* 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.

文献 [2] では，撮影にともなう同期はずれや撮影角度による射影ひずみに対処するために，印刷時に透かし入り画像の周りに枠線を付加し，抽出時にその枠線を元に射影変換している．枠線を用いることで性能の低い携帯電話端末上でも高速認識を可能にしている．しかし，この手法の問題点として，枠線が画像全体を囲むためデザイン上の制約が大きいことや回転因子の補正が出来ないことが挙げられる．そこで，これらの問題を解決する方法として QR コードのファインダパターンとアライメントパターンを用いる手法を提案する．そして，これらの手法を Android 端末に実装し性能を評価する．

以下，2 章で文献 [2] の手法について説明し，3 章で透かし入り画像の作成，4 章で Android 端末へのリアルタイム読み取り処理の実装，5 章で実験と考察，6 章でまとめとする．

2　従来方式

本章では文献 [2] で示されている，カメラ付き携帯電話機を用いたアナログ画像からの高速電子透かし検出方式について述べる．この手法の特徴は，印刷された透かし入り画像を携帯電話のカメラで撮影し透かしを正しく取り出せること，処理能力の低い携帯電話機上で透かし抽出処理が可能であり，処理時間が 1 秒以内であることなどである．

印刷された透かし入り画像から透かしを取り出す場合，様々なひずみが影響する．撮影にともなう同期はずれや

撮影角度による射影ひずみには，文献 [3] で示されている，印刷時に透かし入り画像の周りに枠線を付加し，抽出時にその枠線を元に射影変換し対処する．紙面の波打ちやカメラのレンズによるひずみなどの微少なひずみには，安定した特性をもつ 2 次元サインパターン変調を用いて対処する．また，原画像に透かしパターンを加算する方法を埋め込み処理に用いているため，原画像は加法的雑音として影響する，これには，埋め込みの際に，1 次変調としてスペクトル拡散変調を用いることにより対処する．

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は，原画像 I，L ビットの透かし情報 ID，埋め込み強度 a を入力とする．ここで，I は $W \times H$ 画素のカラー画像 $I = \{I^R_{x,y}, I^G_{x,y}, I^B_{x,y}\}(0 \leq x < W, 0 \leq y < H)$，$a$ は画質と耐性のバランスを調整するパラメータとする．

2.1.1 誤り訂正/検出符号化

透かし情報 ID を誤り訂正及び検出符号化し，n ビットの透かしビット列 $\boldsymbol{c} = \{c_j | c_j \in \{0,1\}, 0 \leq j < n\}$ とする．

2.1.2 スペクトル拡散変調

あらかじめ定めた整数値 N を用いて，c_j を $l = N^2/n$ 回繰り返し引き延ばして得られる長さ N^2 の系列を

$$\boldsymbol{b} = \overbrace{c_0, \ldots, c_0}^{l}, c_1, \cdots, \overbrace{c_{n-1}, \ldots, c_{n-1}}^{l} \tag{1}$$

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

$$\boldsymbol{r} = r_0^{(0)}, \ldots, r_{l-1}^{(0)}, \cdots, r_0^{(n-1)}, \ldots, r_{l-1}^{(n-1)} \tag{2}$$

とする．まず，

$$s_k = b_k r_k \tag{3}$$

により直接拡散変調を行う．ここで，$0 \leq k < N^2$ である．次に，擬似ランダムに決定される置換

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

を用意しこの置換を用いて $\boldsymbol{s}$ 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 $\boldsymbol{t}$ を算出する．

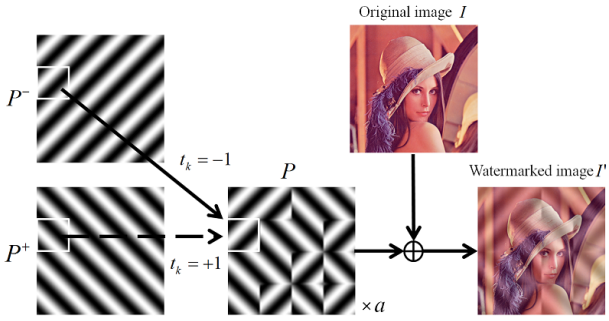

図 1: 2 次元サインパターン変調

ここで，系列 $\boldsymbol{o}$ 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．

$$t_k = s_{o_k} \tag{5}$$

ここで，$0 \leq k < N^2$ である．

2.1.3 2 次元サインパターン変調

図 1 のような，縦横のサイズが原画像と同じ大きさで，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．P^-, P^+ をそれぞれ $N \times N$ 個のブロックに分割して，2 種類のブロックパターン $\{P^{-(h,v)}_{x,y}\}, \{P^{+(h,v)}_{x,y}\}$ を得る．ここで，(h,v) はブロックの位置であり，$0 \leq h < N, 0 \leq v < N$ である．次に，ブロックのラスタスキャン順に沿って，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 $P^{(h,v)}_{x,y}$ を以下のように選択する．

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

これをすべてのブロック位置 (h,v) に対して行うことにより，透かしパターン $P = \{P^{(h,v)}_{x,y}\}$ を得る．

2.1.4 パターン重畳

式 (7) に従い，透かしパターン P，埋め込み強度 a を乗算して I の各色成分に加算し，透かし入り画像 $I' = \{I^{R\prime}_{x,y}, I^{G\prime}_{x,y}, I^{B\prime}_{x,y}\}$ を得る．

$$\begin{aligned} I^{R\prime}_{x,y} &= I^R_{x,y} + aP_{x,y} \\ I^{G\prime}_{x,y} &= I^G_{x,y} + aP_{x,y} \\ I^{B\prime}_{x,y} &= I^B_{x,y} + aP_{x,y} \end{aligned} \tag{7}$$

2.2 抽出法

2.2.1 枠線を利用した同期回復

カメラ撮影時の幾何ひずみによる同期外れに対応するため，枠線を利用した幾何ひずみ補正処理を前処理とし

-1		+1
+1		-1

図 2: 前処理フィルタ用畳込みオペレータ

て行う．まず，透かし入り画像を囲むように枠線を付加し，紙などのアナログ媒体に出力する．透かしの検出を試みる利用者は，枠線付き透かし入り画像を携帯電話のカメラで撮影し再デジタル化する．

再デジタル化された画像を枠線を利用して幾何補正する方法がある [3]．この方法は，枠線を認識して枠のコーナー 4 点の座標を探索し，この 4 点を基準となるテンプレート点と対応付けて射影変換パラメータを推定し，逆射影変換を撮影画像に施すことにより幾何ひずみを補正するものである．補正画像から枠線の内側の透かし入り画像の領域を抽出し，抽出対象画像とする．

2.2.2 グレースケール化と画像サイズの正規化

抽出対象画像の R，G，B の平均値を求めグレースケール画像にする．次にグレースケール画像を $R \times R$ の正方形に正規化する．ここで，R は埋め込み時のパラメーター F を用いて $R = 4F$ とする．$R = 4F$ とするのは，次節で述べる畳込み処理を原画像のサイズに依存せず効率的に行うためである．

2.2.3 前処理フィルタ

前処理フィルタは，畳込み処理とクリップ処理からなる．まず正規化後の画像に対して図 2 の畳込みオペレータによる畳込み処理を施し，透かし信号に対する原画像信号を低減させる．埋込み時に用いた P^-, P^+ の波長は，$R = 4F$ 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 2 の畳込みオペレータを用いると，P^-, P^+ を一度の操作で強調することができる．次に，畳込み処理後の画像の画素値について，値の正負及び 0 によって 3 値化するクリップ処理を行う．図 3 に前処理フィルタ後の画像例を示す.

2.2.4 2 次元サインパターン復調

前処理フィルタ後の画像を $N \times N$ 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埋め込みに用いた P^-, P^+ に対応する周波数エネルギーを求める．ここで，$M = R/N$ である．図 4 に示す畳込み行列 C^- と C^+ による畳込み後の画素値をそれぞれ，$e^{-(h,v)}_{x',y'}, e^{+(h,v)}_{x',y'} (0 \leq x' < M, 0 \leq y' < M)$ とする．次に $e^{-(h,v)}_{x',y'}$ 及び $e^{+(h,v)}_{x',y'}$ についてブロック内のすべての要素の絶対値和をもとめ $E^-_{h,v}$，$E^+_{h,v}$ とする．また $E^-_{h,v}$ と $E^+_{h,v}$ の差を $d_{h,v}$ とする．すべてのブロックに対して $d_{h,v}$ を求めることによって $N \times N$ の正方行列である抽出値行列 $D = \{d_{h,v}\}$ を得る．

図 3: 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 4: 2 次元サインパターン復調のための二つの畳込み行列

$$E^-_{h,v} = \sum_{x'=0}^{M-1} \sum_{y'=0}^{M-1} |e^{-(h,v)}_{x',y'}| \quad (8)$$

$$E^+_{h,v} = \sum_{x'=0}^{M-1} \sum_{y'=0}^{M-1} |e^{+(h,v)}_{x',y'}| \quad (9)$$

$$d_{h,v} = E^+_{h,v} - E^-_{h,v} \quad (10)$$

2.2.5 重み付け

原画像中で 2 次元サインカーブと同じような絵柄をもつ部分では，透かしパターンの振幅よりも原画像の振幅のほうが大きい．そこで 2.2.3 節における畳込み処理後の画像中の $M \times M$ 画素のブロック $B_{h,v}$ を用いてブロック内の画素値 p の絶対値の平均

$$q_{h,v} = \frac{1}{M^2} \sum_{p \in B_{h,v}} |p| \quad (11)$$

を求め，それに基づいて位置 (h, v) のブロックにおける抽出値の重み $w_{h,v}$ を

$$w_{h,v} = f(q_{h,v}) \quad (12)$$

と定める．ここで，重み付け関数である $f(x)$ は単調減少関数とする．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，原画像信号を低減することができる．得られた $w_{h,v}$ を D の各要素 $d_{h,v}$ に乗じて重み付けを行う．

2.2.6　スペクトル拡散復調

2.2.5 節に示す重み付けが施された抽出値行列 D の要素を，ラスタスキャン順に並べた 1 次元行列を $\boldsymbol{g}(0 \leq k < N^2)$ とし，式 (4) のランダム置換を用いてデスクランブルして $\boldsymbol{h}$ を得る．

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

ここで，$0 \leq k < N^2$ である．次に $\boldsymbol{h}$ の長さ $l = N^2/n$ の部分列を

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

として得る．ここで，$0 \leq i < l$ である．更に $\boldsymbol{x^{(j)}}$ の列を平均 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 $\boldsymbol{x^{(j)}}$ と定義する．そして埋め込みに用いた擬似乱数列 $\boldsymbol{r^{(j)}}$ と $\boldsymbol{x^{(j)}}$ 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．

$$\rho_j = \sum_{i=0}^{l-1} x_i^{(j)} r_i^{(j)} \tag{15}$$

そして透かしビット c'_j を

$$c'_j = \begin{cases} 0, & \text{if } \rho_j < 0 \\ 1, & \text{if } \rho_j \geq 0 \end{cases} \tag{16}$$

のようにして抽出する．

2.2.7　誤り訂正/検出復号

$\boldsymbol{c'}$ を誤り訂正及び検出復号し，L ビットの透かし情報 ID' とする．これにより，訂正可能な誤りは訂正される．

3　透かし入り画像の作成

透かし入り画像は PC で作成する．4 ビットの透かし情報 ID を誤り訂正及び検出符号化により冗長化し，32 ビットの透かしビット列とする．また，QR コードのパターンを付加し位置検出に用いる．

3.1　誤り訂正及び検出符号化

4 ビットの透かし情報 ID を誤り訂正及び検出符号化し 32 ビットの透かしビット列を作成する方法を説明する．透かし情報 ID は 4 ビットであるので 16 通りの情報を表すことができる．透かし情報 ID と対応するように系列長が 32 ビットの透かしビット列 $\boldsymbol{\phi^m}$ を 16 種類用意し，透かし情報 ID と透かしビット列を 1 対 1 で対応させる $(0 \leq m < 16)$．$\boldsymbol{\phi^m}$ 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，$0 \leq A, B < 16, A \neq B$ を満たすどのような A, B の組み合わせについても，$\boldsymbol{\phi^A}$ と $\boldsymbol{\phi^B}$ のハミング距離は 16 となる．

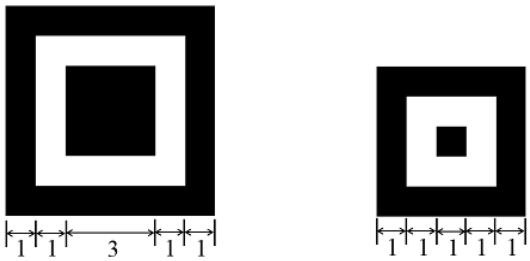

図 5: ファインダパターン (左) とアライメントパターン (右)

抽出の際は，抽出した透かしビット列と $\boldsymbol{\phi^m}$ とのハミング距離を，すべての $\boldsymbol{\phi^m}$ 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 $\boldsymbol{\phi^m}$ が存在する場合，その $\boldsymbol{\phi^m}$ に対応する透かし情報 ID を取り出す．もし，th 以下になる $\boldsymbol{\phi^m}$ が存在しない場合は抽出失敗とする．

3.2　QR コードの位置検出パターン付加

従来方式では，撮影にともなう同期はずれや撮影角度による射影ひずみに対処するために，印刷時に透かし入り画像の周りに枠線を付加し，抽出時にその枠線を元に射影変換している．枠線を用いることで性能の低い携帯電話端末上でも高速認識を可能にしている．しかし，この手法の問題点として，枠線が画像全体を囲むためデザイン上の制約が大きいことや回転角度の補正が出来ないことが挙げられる．そこで，これらの問題を解決する方法として QR コードのファインダパターンとアライメントパターンを用いる手法を提案する．

QR コードは 1999 年に規格化され，様々な用途に用いられている．QR コードでは図 5 に示すファインダパターンとアライメントパターンを用いることで，高速に位置検出を行なっている．そこで，この 2 つのパターンを位置検出に利用する．

図 6 に QR コードのパターンを配置した透かし入り画像を示す．透かし入り画像の左上，左下，右上に，ファインダパターンを配置し，右下には，アライメントパターンを配置する．四隅のうち 1 つだけアライメントパターンにすることで回転角度の特定に用いる．

ファインダパターンは中心を横切る直線で走査すると，黒と白が交互に 1:1:3:1:1 の比率で現れる．この比率は

図 6: 透かし入り画像例

図 7: 画像内への配置例

ファインダパターンが回転しても変わらないため，高速に位置を検出できる．

QR コードのパターンを利用することで，図 6 に示すように背景と一体化したデザインや，図 7 に示すように透かし領域を画像内に配置するデザインが可能になる．このように，従来方式の枠線を利用する場合に比べ，デザインの自由度を高くすることができる．

4 Android 端末へのリアルタイム読み取り処理の実装

Android 端末上では透かしの読み取り処理を行う．ユーザーが透かし入り画像に携帯電話端末をかざすだけで自動的に読み取りができるようにする．そのために，ピントがずれたらオートフォーカスをし，常にピントが合うようにする．また，透かしの読み取り処理を常に繰り返す．

読み取り処理は主に位置検出，平面射影変換，透かし抽出，画面表示に分けることができる．以下では各処理について説明する．

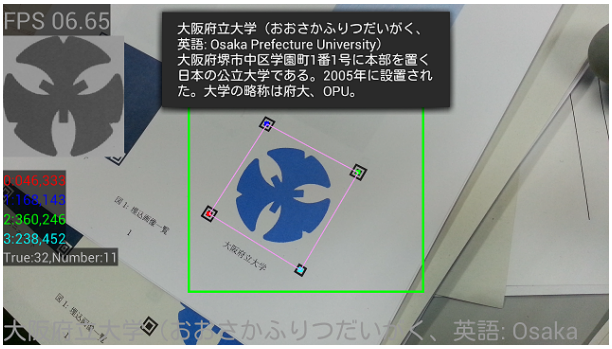

図 8: 画面表示

4.1 位置検出

Zxing という Google が公開しているバーコードリーダーのライブラリを用いる．ライブラリ内の QR コードのファインダパターンとアライメントパターンの検出処理を用い，それぞれのパターンの中心座標を取得する．

4.2 平面射影変換

位置検出で得られたファインダパターン 3 点とアライメントパターン 1 点で囲まれた四角形を，あらかじめ決めておいたサイズの長方形に平面射影変換する．その後，パターンの内側の領域を切り出して抽出対象画像とする．

4.3 透かし抽出

誤り訂正及び検出復号までの処理は従来方式の抽出法と同じである．あらかじめ透かし情報 ID とメッセージを対応させて保存しておき，透かし情報 ID を抽出した際に対応するメッセージに変換する．

4.4 画面表示

図 8 に画面表示の例を示す．QR コードのパターンが検出可能なときに，検出したパターンの座標とそれらを結ぶ四角形を表示し，画面の左側に射影変換後の抽出対象画像，得られた 4 点の座標値を表示する．また，抽出に成功した際に，透かし情報 ID に対応するメッセージを表示する．

5 実験と考察

実験では Android 端末に Galaxy S III GT-I9300 を用いた．原画像は 512 × 512 画素の 2 種類の画像を用い，$F = 48$, $N = 32$ で透かしを埋め込む．誤り訂正で使用するパラメータは th = 5 とする．印刷で使用するプリンターは EPSON M5000 で用紙は上質紙を用いる．図 9 に原画像と透かし入り画像を示す．透かし読み取りアプリを Android 端末で実際に動作させて透かし入り画像を読み取り，抽出処理にかかる時間を測定する．表 1

表 1: 処理時間 (100 回試行)

処理	時間 [ms]
位置検出	97.99
平面射影変換	232.77
透かし抽出	215.70
合計	546.46

表 2: 抽出成功率 (100 回試行)

	a	5cm	15cm
lenna	4	2%	86%
	8	98%	100%
	12	100%	100%
OPU	4	0%	23%
	8	19%	99%
	12	95%	100%

に処理時間を示す．この処理時間は 100 回の試行の平均値である．この表より，一回の処理時間の合計は 546ms であることがわかる．ここから，この手法はユーザーに遅延を感じさせない手法であるといえる．

次に，ロバスト性を評価するために，原画像 2 種類（lenna, OPU），埋め込み強度 3 種類（$a = 4, 8, 12$），印刷サイズ 2 種類（5cm 四方, 15cm 四方）のについて，透かしを正しく取り出せるか検証した．撮影はフリーハンドで行い，カメラをかざす際に透かし入り画像が出来るだけ大きく写るようにした．そして，位置検出パターンを検出できた 100 回の試行において，透かしの抽出を成功した割合を求めた．

表 2 に抽出成功率を示す．印刷サイズが 5cm 四方と 15cm 四方では，15cm 四方のほうが抽出成功率が高いことがわかる．これは，印刷サイズが小さいと，印刷の際に生じるのインクのにじみや紙面のシワの影響を受けやすいためだと考えられる．また，撮影時に近接撮影となるため手ぶれの影響も大きくなるためだと考えられる．OPU と lenna を比べると OPU の方が抽出成功率が低い．これは，OPU のマークは青と白で構成されており，白の部分に加算されたサインパターンの模様が，印刷の際に消えてしまうためためだと考えられる．

6 まとめ

本稿では，文献 [2] で従来手法で用いられている，枠線を用いた空間同期補正の問題点を解決するために，QR コードのファインダパターンとアライメントパターンを用いる手法を提案した．また，Android 端末上で動作する読み取りアプリを実装し，性能を評価した．実験により，一回の読み取りの処理時間が 546ms であり，十分高速なことを確認した．また，抽出成功率は埋め込み強度が $a = 12$ の場合，OPU の印刷サイズ 5cm 四方で 95%となる以外は 100% となることを確認した．今後の課題としては，位置検出パターンの形状の改良や配置の検討などがあげられる．

参考文献

[1] 松井甲子雄, “電子透かしの基礎, ” 森北出版, 1998.

[2] 中村高雄，片山淳，山室雅司，曽根原登, “カメラ付き携帯電話機を用いたアナログ画像からの高速電子透かし検出方式, ” 電子情報通信学会論文誌 D-II, vol.J87-D-II, no.12, pp.2145–2155, 2004.

[3] 片山淳, 中村高雄, 山室雅司, 曽根原登, “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.

lenna(原画像)

OPU(原画像)

lenna(透かし入り画像)

OPU(透かし入り画像)

図 9: 原画像と透かし入り画像 ($a = 12$)