社団法人 情報処理学会 研究報告
IPSJ SIG Technical Reports

2009-IOT-4 (43)
2009 / 3 / 6

電子透かしに用いる位置ずれ耐性を有する二次元誤り訂正符号の提案

吉崎 健二† 稲葉 宏幸†

† 京都工芸繊維大学 大学院工芸科学研究科
〒 606-8585, 京都市左京区松ヶ崎御所海道町
E-mail: †{yosizaki,inaba}@ice.is.kit.ac.jp

あらまし 電子透かしはその性質上，様々な改変操作に耐性を有する必要がある．なかでも，抜き取りや切り取りなどの改変操作が行われた場合に，透かしデータの同期が取れなくなることがあり，問題となっている．この問題に対して，同期誤りを訂正可能な誤り訂正符号を用いる方式がいくつか提案されているが，音声信号など一次元のデータを対象とするものであり，画像信号などの二次元のデータを対象とするものは知られていない．そこで本論文では，二つの同期誤り訂正巡回符号を組み合わせることにより，位置ずれ耐性を有する二次元誤り訂正符号を提案する．また，提案方式の電子透かしへの応用についても述べる．

キーワード 電子透かし, 巡回符号, 同期問題

Proposal on 2-Dimensional Error Correcting Code with Self-Synchronization Capability for Digital Watermark

Kenji YOSHIZAKI† and Hiroyuki INABA†

† Kyoto Institute of Technology
Goshokaidoucho, Matsugasaki, Sakyo-ku, Kyoto 606-8585 JAPAN
E-mail: †{yosizaki,inaba}@ice.is.kit.ac.jp

Abstract Generally, a digital watermark should have an ability to resist various attacks such as shifting, clipping, or extraction. These attacks cause a serious synchronization problem in the decoding sequence of the digital watermark. In this paper, we propose a 2-dimensional error correcting code with 2D self-synchronization capability for digital watermark. The proposed code is constructed by combining two cyclic codes that have a self-synchronization capability. This code has robustness against not only unsynchronization attacks such as clipping and picking but also additional noise. We can apply the code for steganography and 2D barcode besides digital watermark.

Key words Digital Watermark, Cyclic Code, Synchronization Problem

1. はじめに

近年，インターネット技術の発展に伴い，ディジタルコンテンツやそれに関連するサービスが普及している．ディジタルコンテンツは複製や加工が容易にできるため便利である一方，不正コピーや不正配布などの著作権侵害が問題となっている．その対策として，不正コピー追跡や改ざん抑制などを目的とする技術として電子透かしが注目されている．

電子透かしとは，ディジタルコンテンツに権利者情報や制御用データなどの副情報を，人間には知覚できないように埋め込む技術である [1] [2]．電子透かしはその性質上，圧縮やノイズ付加，各種フィルタ処理，また，回転・切り取りなどの各種改変操作に対して耐性を有する必要がある．この問題に対して，同期誤りを訂正可能な誤り訂正符号を用いる方式がいくつか提案されているが，音声信号など一次元データを対象とするものであり，画像信号などの二次元データを対象とするものは知られていない．

本論文では，電子透かしに用いる位置ずれ耐性を有する二次元誤り訂正符号について提案する．提案方式では，電子透かしの埋め込み方式そのものについては特定せず，埋め込む情報の符号化について述べる．また，埋め込み情報は 2 値のビット列であり，切り取りや抜き取りが行われた場合には，符号系列に対して切り取りや抜き取りの影響が同じように生じるものとする．そのため，回転や拡大・縮小については想定せずに，切り取りや抜き取りなどによって，透かしデータの復号時に同期が失われてしまう問題に焦点をあてている．

提案する方式はシフト訂正可能な巡回符号を用いて，抜き取りや切り取りなどによって起こる巡回的な位置ずれに対して耐性を有する二次元誤り訂正符号を構成するものである．さらに，その符号を繰り返すことにより，切り取りや抜き取りによる同期ずれに対して耐性を持たせている．

2. シフト訂正可能な巡回符号

巡回符号とはある符号語 $c(x)$ を巡回シフトしたものも符号語となる線形符号であり，効率のよい符号化・復号化回路を実現できるため幅広く用いられている [3] [4]．巡回符号のある符号語を繰り返し配置しておけば，巡回符号の性質より，同期位置がずれてしまっても誤りは検出されないが，一般に復元されるメッセージは異なるものになってしまい，正しい同期位置を知ることもできない．そこで，文献 [5] では巡回符号の生成方法を改良することにより，同期ずれが生じていても正しい同期位置を推定でき，元のメッセージを正しく復元できる符号の構成法が示されている．本論文で提案する手法は，この符号を利用しているので本節ではまずこの符号の符号化法と復号法について簡単に述べる．

巡回符号の生成多項式を $g(x)$ とし，位数が符号長 n に等しい元 β を根に持つ最小多項式を $q(x)$(ただし，$q(x) \nmid g(x)$) とする．シフト訂正可能な巡回符号の符号化アルゴリズムは次のようになる．

・符号化アルゴリズム

Step 1:
メッセージ多項式 $m(x)$ を用意する．ただし，$\deg(m(x)) < \deg(g(x)) - \deg(g(x)) - \deg(q(x)) - 1$ である．

Step 2:
メッセージ多項式 $m(x)$ から，$q(x)$ を用いて拡大メッセージ多項式 $m'(x)$ を次式のように作成する．
$m'(x) = q(x)m(x) + 1$

Step 3:
符号語 $c(x) = g(x)m'(x)$ を得る．

また，復号アルゴリズムは以下のようになる．

・復号アルゴリズム

Step 1:
抽出した符号語 $r(x)$ に対して，$g(x)$ により生成された巡回符号として (ユークリッド復号法等により) 誤り訂正をしたものを $c'(x)$ とし，次を計算する．
$m'(x) = c'(x)/g(x)$

Step 2:
シフト量 $r = \log_\beta m'(\beta)$ を計算する．

Step 3:
メッセージ多項式 $m(x)$ は次のように得られる．
$m(x) = \left(\left(x^{-r}c'(x)\right) \bmod (x^n + 1)\right)/g(x)$

ここで Step 2 でシフト量が計算できる理由は以下のように説明できる．

ある符号語 $c(x)$ を巡回シフトした符号語は

$$c'(x) = x^r c(x) \bmod (x^n - 1) \tag{1}$$

のようにあらわされる．ここで符号語 $c(x)$ は

$$c(x) = g(x)m'(x) = g(x)(q(x)m(x) + 1) \tag{2}$$

と表されるから，式 (1) を復号して得られる拡大メッセージ多項式 $m'_r(x)$ に β を代入すると，

$$\begin{aligned} m'_r(\beta) &= \beta^r(q(\beta)m(\beta) + 1) \\ &= \beta^r \end{aligned} \tag{3}$$

が得られる．β の位数は n であるため，シフト量 r を推定することができ，最後に式 (1) を次式に従って

$$c(x) = x^{-r}c'(x) \bmod (x^n - 1) \tag{4}$$

r だけ逆シフトすることにより，元の符号語を求めることができる．

3. 提 案 方 式

本章では提案方式の符号化，復号アルゴリズム，符号の性能について述べる．提案する二次元符号の構成の概略は図 1 のようになる．2. 章で述べたシフト訂正可能な BCH 符号を行方向に r 回繰り返し，隣接する BCH 符号のシフト量の差分をシフト訂正可能な RS(Reed-Solomon) 符号の 1 シンボルに割り当て，符号を構成する．そして，RS 符号も列方向に R 回繰り返すことにより，自己同期可能な抜き取り・切り取りに耐性を有する符号を構成する．

提案方式の利点として，次のことがあげられる．

- 巡回符号の符号化復号装置を軽微な変更のみで利用できるため，高速な符号化・復号が可能である．
- 埋め込みデータを正しく復元できるだけでなく，２次元のシフト量を推定できるので画像データなどの同期ずれを補正できる．
- 符号の誤り訂正能力や繰り返し回数等の，パラメータを比較的自由に選ぶことができる．

以下では，BCH 符号の情報多項式を $m(x)$，RS 符号の情報多項式を $M(x)$ とする．ここで，$m(x)$ は $GF(2)$ 上の多項式，$M(x)$ は $GF(2^m)$ 上の多項式である．これらの情報多項式 $m(x)$ と $M(x)$ は，それぞれ独立に選ぶことができる．また，$m(x)$BCH 符号の符号語を $w(x)$，$M(x)$ に対応する RS 符号の符号語を $W(x) = W_0 + W_1x +, \cdots, W_{n-1}x^{n-1}$ とする．ここで，符号長はいずれも $n = 2^m - 1$ である．また，ガロア体の元を整数に一対一に変換する関数 $S(\alpha)(\alpha \in GF(2^m), 0 \leqq s(\alpha) < 2^m)$ と，その逆関数 $S^{-1}(x)(0 \leqq x < 2^m, S^{-1}(x) \in GF(2^m))$ を定義する．そして，上記の BCH 符号と RS 符号を用いて構成する二次元符号ブロック空間を $C(u, v)(0 \leqq u < r, 0 \leqq v < R \cdot n + 1, C(u, v) \in GF(2)$ 上の n 次元ベクトル) とする．ここで，r, R はそれぞれ BCH 符号と，RS 符号の繰り返し回数である．

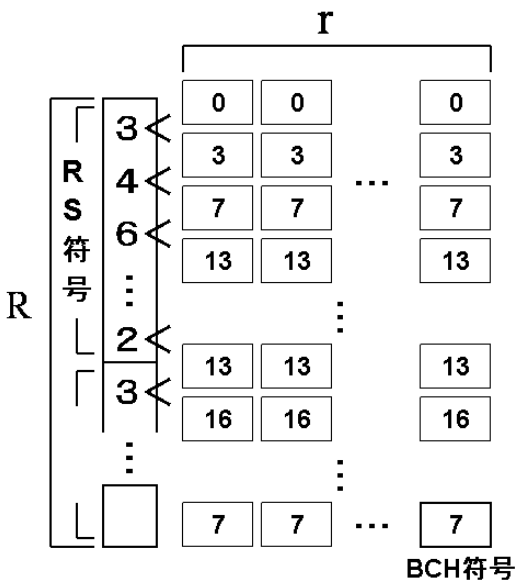

図 1　提案方式図
Fig. 1　Structure of proposed code.

3.1　符号化アルゴリズム

符号化アルゴリズムの概略を図 2 に示す．図中の四角の中の数字は BCH 符号の巡回シフト量である．アルゴリズムは以下のようになる．ここで，符号化アルゴリズム中の丸数字は図 2 中の数字と対応している．

Step 1:
　情報多項式 $m(x), M(x)$ を用意し，2. 章で述べた方法によりそれぞれ BCH 符号，および RS 符号の符号化を行う．
Step 2: ①
　$C(i,0) \leftarrow w(x)\ (i = 0, 1, \cdots, r-1)$
Step 3:
　$j \leftarrow 1,\ w'(x) \leftarrow w(x)$
Step 4: ②
　$s \leftarrow S(W_{j \bmod n})$
　$w'(x) \leftarrow x^s w'(x) \bmod (x^n - 1)$
Step 5:
　$C(i,j) \leftarrow w'(x)\ (i = 0, 1, \cdots, r-1)$
Step 6:
　$j \leftarrow j + 1$
　if　$j < n \cdot r + 1$
　　goto Step 4
　else
　　end

3.2　復号アルゴリズム

3.1 で述べた符号の復号アルゴリズムを以下に述べる．復号アルゴリズムの概略を図 3 に示す．図中の斜線の四角は誤って復号した BCH 符号をあらわしている．透かしから抽出した二次元符号ブロック空間を $C'(u,v)(0 \leqq u < U', 0 \leqq v < V', C'(u,v) \in GF(2)$ 上の n 次元ベクトル) とする．ここで，

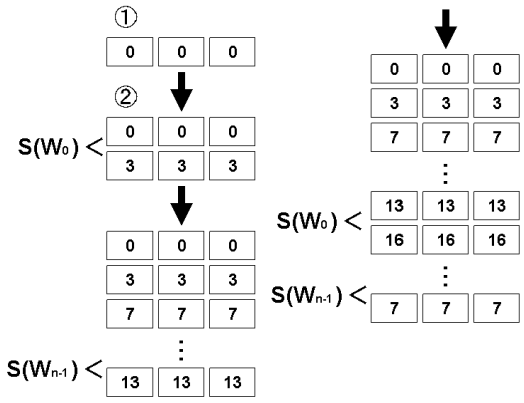

図 2　符号化アルゴリズム
Fig. 2　Encoding algorithm.

U', V' は切り取りなどにより，埋め込んだ符号のサイズと異なる場合もある．また，符号ブロック空間を抽出する際に，その抽出元である符号空間の幅 $\hat{X}$ が BCH 符号長の正整数倍，または高さ $\hat{Y}$ が RS 符号の符号長の正整数倍 +1 ではない場合には，符号語のシンボルとして抽出可能な部分だけを抽出し，抽出不可能な符号語のシンボルは消失誤りとする．以下でも同様に，図 3 中の丸数字と復号アルゴリズム中の丸数字は対応している．

Step 1:
　透かしから符号ブロック空間 $C'(u,v)$ を抽出する．
Step 2:
　$C'(u,v)$ 中の全ての BCH 符号のブロックを復号し，復号できたものの中から多数決を取り，それを $m(x)$ とする．ここで，正しく復号できた符号語の割合がある一定数 γ より小さい場合は，信頼性が低いと考えられるので，エラー (復号失敗) を出力する．
Step 3: ③
　全ての BCH 符号のブロックのシフト量 $s_{i,j}\ (0 \leqq i < U', 0 \leqq j < V')$ を計算する．
Step 4: ④
　$W_j \leftarrow S^{-1}(s_{i,j+1} - s_{i,j})\ (j = 0, 1, \cdots, Y)$
Step 5:
　列ブロック $W_0, W_1, ..., W_Y - 1$ に含まれる全ての RS 符号の復号を行う．
Step 6: ⑤
　Step 4, 5 をすべての列ブロックについて行う．
Step 7:
　Step 2 と同様に，復号できた RS 符号ブロックの中から多数決を取り，それを $M(x)$ とする．ここでも，正しく復号できた符号語の割合がある一定数 γ より小さい場合は，信頼性が低いと考えられるので，エラー (復号失敗) を出力する．

3.3　符 号 性 能

提案した符号の性能について述べる．ここでは想定される改変操作として誤り付加，上書き，抜き取り，切り取りを考え

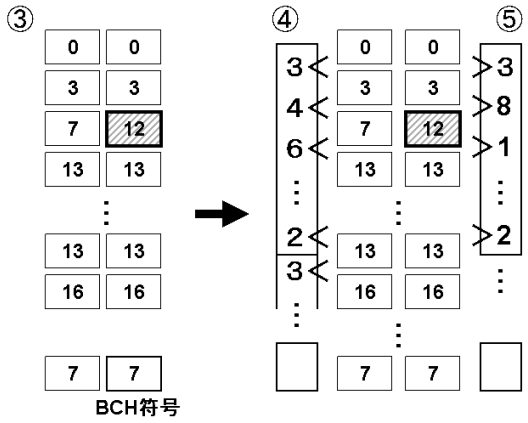

図 3　復号アルゴリズム
Fig. 3　Decoding algorithm.

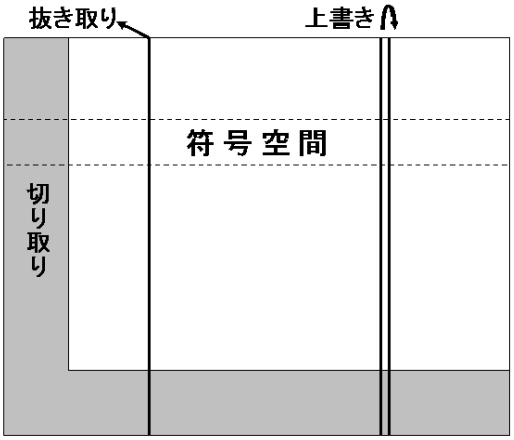

図 4　各種操作
Fig. 4　Various alterations.

る．誤り付加については構成に使用した BCH 符号の誤り訂正能力以下の誤りは明らかに訂正可能であり，さらに BCH 符号が復号に失敗した場合には RS 符号によって誤り訂正が試みられることになる．抜き取りにおいては，1 行抜き取る場合，抜き取った部分以降は順次シフトされることになるので，抜き取られた行が含まれている RS 符号は正しく復号できる保証はないが，そのほかの RS 符号のブロックにおいてはシフト訂正可能な巡回符号の性質により，正しく復号することができる．また，1 列抜き取る場合はその列が含まれていた BCH 符号が正しく復号されるかどうかは保障されないが，その他のブロックにおいてはシフト訂正可能な巡回符号の性質により，正しく復号することができるため，多数決復号により結果として正しいメッセージ $m(x)$ と $M(x)$ を復号できることになる．

この方式における符号化率は情報記号数をそれぞれ k_{bch}, k_{rs} とすると，

$$R = \frac{k_{bch} + k_{rs} \cdot m}{n \cdot r(n \cdot R + 1)} \tag{5}$$

となる．

4. 実　　験

前章で提案した二次元符号について具体的なパラメータを設定し，各種改変操作を加えた上で復号を試みる計算機シミュレーションを行い，想定される性能が達成されているかどうかを調べる．

4.1 実験条件

実験では上書き・抜き取り・切り取りの各操作を行い，その後さらに誤りを付加した (図 4)．ここで，上書きとは，ある行または列を隣接する行または列へ上書きする操作を示している．実験を行った際のパラメータは表 1，2 とした．提案する符号の性質上，BCH 符号の復号精度がその後の RS 符号の復号精度へと大きく影響するため，BCH 符号の最小距離を大きく取っている．このパラメータにおいては，BCH 符号は 6 個の誤りを訂正できる．また，RS 符号は 4 個の誤りを訂正できるが，シフト量の差分を符号語の各シンボルに割り当てているため，BCH 符号の 1 ブロックが誤ると，RS 符号は 2 シンボル誤ることになる．そのため，BCH 符号の 2 ブロックまでの誤りを訂正できることになる．このパラメータにおける符号化率は，$R = (24 + 54 \cdot 6)/(189 \cdot 190) = 0.01$ である．

表 1　巡回符号パラメータ ($GF(2^6)$)
Table 1　Parameter of cyclic code over $GF(2^6)$.

パラメータ	BCH	RS
符号長	$n = 63$	$n = 63$
メッセージ長	$k_{bch} = 24$	$k_{rs} = 54$
最小距離	$d_{bch} = 13$	$d_{rs} = 9$

表 2　シミュレーション条件
Table 2　Parameter of simulation.

パラメータ	値
繰り返し回数	$r = R = 3$
符号サイズ	189×190 ($W \times H$)
多数決閾値	$\gamma = 2/3 = 0.67$

4.2 実験結果

上書き・抜き取り・切り取りを行った実験結果を表 3〜6 に示す．ここで，表 6 における切り取り領域とは，符号空間の面積に対する切り取りの割合であり，切り取った部分について復号を行っている．また，表中の値は正しく復号できた割合 (%) であり，BCH/RS の順である．両方とも 100%のときは省略して 100 としている．

今回用いたパラメータでは，繰り返し回数をどちらも 3 回としているため，RS 符号は 9 ブロック含まれていることになる．1 列または 1 行抜き取った場合には，RS 符号 3 ブロックに影響が生じることになる．抜き取られたブロック以外はシフト訂正可能な巡回符号の性質により，正しく復号できることになる．そのため，2 つの抜き取りまでは正しく復号できることになるが，順次シフトされるため，BER が高い場合には正しく復号できない場合もある．また，切り取りに対しては切り取り面積を 70%までとしているため，RS 符号は少なくとも 4 ブロック程度は残っていることになるため，正しく復号できる．以上より，実験結果と想定した符号の性能が近いことが確認できた．

表 3 上 書 き
Table 3 Results of decoding after overwrite.

上書き本数		1	2	3
B	0.01	100	100	100
E	0.02	100	100	100
R	0.03	100	100	100

表 4 抜き取り (行)
Table 4 Results of decoding after picking (Line).

抜き取り本数		1	2	3
B	0.01	100	100	98/92
E	0.02	100	100	100/90
R	0.03	100	100/96	98/82

表 5 抜き取り操作に対する復号率 (列)
Table 5 Results of decoding after picking (Column).

抜き取り本数		1	2	3
B	0.01	100	100/98	100/84
E	0.02	100	100	98/80
R	0.03	100	100/98	100/74

表 6 切り取り操作に対する復号率
Table 6 Results of decoding after clipping.

切り取り領域 (%)		70	75	80	85	90
B	0.01	100	98/98	100	100	100
E	0.02	100	98/98	100	100	100
R	0.03	100	98/98	100	100	100

5. む す び

本論文では位置ずれ耐性を有する二次元誤り訂正符号の提案を行った．そして，実際にパラメータを設定し，各種改変操作を加えてその復号率を求め，想定した符号の性能に近いかどうかを調べた．

本方式の応用例の一つとして，電子透かしの他にステガノグラフィも考えられる．ステガノグラフィとは通信している事実そのものを隠す技術である [2] [6]．ステガノグラフィはその性質上，故意に操作が加えられることはないので，最も簡単な埋め込み法として，空間領域に透かしを埋め込む方法が挙げられる．ただし，例えば最下位ビットに埋め込んだ場合には，繰り返し符号のため同じパターンが複数回現れることになり，注意が必要である．

本論文では電子透かしの具体的な埋め込み方法については触れなかったが，今後，この符号の特徴を生かした電子透かしについて検討していく必要がある．

文 献

[1] 松井甲子雄, ”電子透かしの基礎-マルチメディアのニュープロテクト技術-”, 森北出版株式会社, 1998
[2] 画像電子学会編, ”電子透かし技術 ディジタルコンテンツのセキュリティ”, 東京電機大学, 2004
[3] F.J. Macwilliams, N.J.A. Sloane, ”The Theory of Error-Correcting Codes”, North-Holland, 1977
[4] W. Wesley Peterson, E.J. Weldon, Jr., ”Error-Correcting Codes” Mit Pr, 1972
[5] Hiroyuki Inaba, ”Notes on Rotation-Resistent Digital Watermark using Radon Transform”, ISITA 2004 pp.310-315, 2004
[6] 宮地充子, 菊池浩明, ”情報セキュリティ”, オーム社, 2003