Implementation of Algebraic-Geometric Code by a Miura Curve and its Visualization

by Eiko SHIMIZU and Akira KANEKO[1)]
(Department of Information Sciences, Ochanomizu University)

(Received May 16, 2003)

Abstract

In this article, we examine a concrete example of maximal curve represented as a Miura C_3^4 curve, and implement an algebraic-geometric code using rational points of this curve. The implementation is visualized through application to greyscale and color raster images.

§1. Introduction

Reed-Solomon code was introduced in 1960 and has been used practically. Its only deficiency is that the code length is limited to the order of the finite field employed. To obtain longer codes over a fixed field, algebraic-geometric codes of residue type were introduced by Goppa in 1981 and have been studied both theoretically and algorithmically. But its practical implementation is still not popular.

In this report we choose one of the maximal curves found by M. Kawakita[KS] represented as the so called Miura C_3^4 curve, study its properties and construct a Goppa code employing its rational points.

This curve has fairly good properties. For example, the minimum distance of the code subspaces constructed from the natural monomial ordering agrees with the Goppa designed distance for all non-trivial information lengths.

Our implementation is applied to each row of an ASCII pnm format 256 greyscale or RGB color raster image files, prepares code space according to the width of the input image, adds the redundancy bits up to the longest possible in choice, allows to give damage by the mouse operation, and decodes it, thus visualizing how the image is encoded and the damaged image can be, or cannot be recovered, according to the widths of the damaged parts. We have, of course, the Reed-Solomon version. This will serve especially for educational purpose of the theory of error-correcting codes.

The implementation was done with use of `g++` and `Xlib`, which may work on any unix system with `X11`, and also on cygwin which emulates them on Microsoft Wondows.

§2. Algebraic-geometric codes

Let p be a prime, $q = p^m$ and let F_q be a finite field. Before introducing our concrete example, we briefly review the theory of algebraic-geometric codes (AG codes for short) or Goppa codes, specializing to the case of a plane algebraic curve V. Let

$$D = P_1 + \cdots + P_n \tag{1}$$

be the sum of all finite F_q-rational points of V. Let G be another rational divisor of V satisfying

$$\operatorname{supp} D \cap \operatorname{supp} G = \emptyset. \tag{2}$$

Let K be the rational function field of V and let

$$L(G) := \{f \in K \mid G + (f) \geq 0\} \cup \{0\}$$

be the affine version of the linear system defined by the divisor G. Its dimension over F_q is denoted by $l(G)$. Let us define an F_q-linear mapping α by

$$\begin{array}{rccc} \alpha: & L(G) & \longrightarrow & F_q^n \\ & \Psi & & \Psi \\ & f(x) & \mapsto & (f(P_1), \dots, f(P_n)). \end{array}$$

Then the subspace $C_L(D, G) := \operatorname{Image} \alpha \subset F_q^n$ is said to be a function type AG code or L-construction associated to D, G.

[1)]Partially supported by Grant-in-Aid for Scientific Research 12640158.

Next define the space of rational 1-forms on an algebraic curve V by

$$\Omega[V] = \{df \mid f \in F_q[V]\}, \quad \Omega(V) = K\Omega[V].$$

Set

$$\Omega(G) := \{\omega \in \Omega(V) \mid (\omega) \geq G\} \cup \{0\},$$

and $\iota(G) := \dim_{F_q} \Omega(G)$. Set

$$\Omega(G-D) := \{\omega \in \Omega(G-D) \mid (\omega) \geq G-D\}.$$

Then a residue type AG code or Ω-construction is defined as $C_\Omega(D,G) = \mathrm{Image}\,\beta \subset F_q^n$, where β is the following linear mapping:

$$\begin{tikzcd}
\beta: \Omega(G-D) \arrow[r] & F_q^n \\
\rotatebox[origin=c]{90}{$\in$} & \rotatebox[origin=c]{90}{$\in$} \\
\omega \arrow[r, mapsto] & (\mathrm{res}_{P_1}\omega, \ldots, \mathrm{res}_{P_n}\omega).
\end{tikzcd}$$

We have the duality $C_\Omega(D,G) = C_L(D,G)^\perp \subset F_q^n$ induced by the pairing with the residue map

$$\langle f, \omega\rangle := \sum \mathrm{res}_{P_i} f\omega = \sum f(P_i)\mathrm{res}_{P_i}\omega.$$

Let g denote the genus of V. Employing the Riemann-Roch theorem and Clifford's theorem, we obtain

Theorem (Goppa bound)

$$\begin{aligned}
d(C_L(D,G)) &\geq d_G(C_L(D,G)) := n - \deg(G), && (3)\\
d(C_\Omega(D,G)) &\geq d_G(C_\Omega(D,G)) := \deg(G) - 2g + 2, && (4)\\
d(C_L(D,G)) &\geq d_G := n - k + 1 - g,\\
d(C_\Omega(D,G)) &\geq d_G := k + 1 - g.
\end{aligned}$$

d_G is called Goppa's designed distance.

The residue type AG codes are practically constructed as the dual of function type AG codes employing the latter as the parity check matrices.

§3. Decoding algorithms

Three algorithms are now available: The first is Pellicaan's theory of t-error correcting pair. The second is Feng-Rao's majority principle. The third is Sudan's list decoding. In this report we present implementation of the first one, Pellicaan's theory.

We sketch the algorithm following [MII]. First we prepare some notation. For two vectors $\boldsymbol{x} = (x_1, \ldots, x_n)$, $\boldsymbol{y} = (y_1, \ldots, y_n)$ we define the product operation $*$ by

$$\boldsymbol{x} * \boldsymbol{y} = (x_1y_1, \ldots, x_ny_n).$$

Clearly from the definition, we have $\boldsymbol{x} * \boldsymbol{y} = \boldsymbol{y} * \boldsymbol{x}$, $\langle \boldsymbol{x}, \boldsymbol{y} * \boldsymbol{z}\rangle = \langle \boldsymbol{x} * \boldsymbol{y}, \boldsymbol{z}\rangle$. Hence for three linear subspaces $A, B, C \subset F_q^n$, we have $A * B \subset C^\perp$ if and only if $A * C \subset B^\perp$. Next for a linear subspace $A \subset F_q^n$ and a subset of indices $I \subset \{1, \ldots, n\}$ we set in general

$$A(I) := \{\boldsymbol{x} \in A \mid x_i = 0 \text{ for } \forall i \in I\}.$$

$\mathrm{supp}\,\boldsymbol{x}$ denotes the positions of indices corresponding to the non-zero components of the vector $\boldsymbol{x}$. $Z(\boldsymbol{x})$ denotes its complement. B' denotes the dual space of B. Given a pair of linear subspaces (A,B), let $E_{\boldsymbol{w}}$ be the linear mapping defined for each $\boldsymbol{w} \in F_q^n$ as follows:

$$\begin{tikzcd}
E_{\boldsymbol{w}}: A \arrow[r] & B' \\
\rotatebox[origin=c]{90}{$\in$} & \rotatebox[origin=c]{90}{$\in$} \\
\boldsymbol{x} \arrow[r, mapsto] & \langle \boldsymbol{w}, \boldsymbol{x} * \cdot\rangle.
\end{tikzcd}$$

We have

$$\mathrm{Ker}\,E_{\boldsymbol{w}} = \{\boldsymbol{x} \in A \mid \langle \boldsymbol{w}, \boldsymbol{x} * \boldsymbol{y}\rangle = 0 \text{ for } \forall \boldsymbol{y} \in B\}, \qquad A(\mathrm{supp}\,(\boldsymbol{w})) \subset \mathrm{Ker}\,E_{\boldsymbol{w}}.$$

Now we have

Definition A pair of linear subspaces (A, B) is called a t-error-correcting-pair if it satisfies the following conditions:
1) $A * B \subset C^{\perp}$.
2) $d(B^{\perp}) > t$.
3) $\dim A > t$.
4) $d(A) + d(C) > n$.

From this definition we immediately obtain

Corollary 1) $\operatorname{Ker} E_{\boldsymbol{y}} = \operatorname{Ker} E_{\boldsymbol{e}}$ if $\boldsymbol{x} = \boldsymbol{y} - \boldsymbol{e} \in C$.
2) $\operatorname{Ker} E_{\boldsymbol{e}} = A(\operatorname{supp}(\boldsymbol{e}))$ if $\operatorname{wt}(\boldsymbol{e}) \leq t$.
3) $A(\operatorname{supp}(\boldsymbol{e})) \neq \{0\}$ if $\operatorname{wt}(\boldsymbol{e}) \leq t$.
4) $\#Z(\boldsymbol{x}) < d(C)$ for $0 \neq \forall \boldsymbol{x} \in A$.

Given a t-error-correcting pair (A, B), the t-designed distance decoding, that is, the algorithm finding the original vector $\boldsymbol{x} = \boldsymbol{y} - \boldsymbol{e}$ under the assumption that the error vector $\boldsymbol{e}$ contained in the received word $\boldsymbol{y}$ has $\operatorname{wt}(\boldsymbol{e}) \leq t$, is as follows: By the above corollary $A(\operatorname{supp} \boldsymbol{e}) = \operatorname{Ker} E_{\boldsymbol{e}} = \operatorname{Ker} E_{\boldsymbol{y}} \neq \{0\}$ is first determined from $\boldsymbol{y}$. We can find a concrete element $\boldsymbol{a}$ therein by solving a system of linear equations. Then $\operatorname{supp}(\boldsymbol{e}) \subset Z(\boldsymbol{a})$, and we can estimate the location of erroneous coordinates. Further, by 4) $\#Z(\boldsymbol{a}) \leq d(C) - 1$, hence the column vectors of the parity-check matrix H of C corresponding to the components of $Z(\boldsymbol{a})$ are linearly independent, and the system of linear equations

$$H\boldsymbol{e} = H\boldsymbol{y}, \quad \operatorname{supp} \boldsymbol{e} \subset Z(\boldsymbol{a})$$

is uniquely solvable. Thus we can specify $\boldsymbol{e}$. The complexity of this algoritm is dominated by the solution of linear equations and of $O(n^3)$.

When we apply this algorithm to function type AG codes, we choose

$$t = \left\lfloor \frac{n - \deg(G) - 1 - g + 2l(G - D)}{2} \right\rfloor,$$

hence $n \geq 2t + 1 + g > t + g$. Now fix an F_q-rational positive divisor H such that

$$\operatorname{supp}(H) \cap \operatorname{supp}(D) = \emptyset, \quad \deg(H) = t + g. \tag{5}$$

For one-point AG codes, $H = (t + g)P_{\infty}$ will do. Then we have $\deg(H - D) < 0$, hence $l(H - D) = 0$. Thus the following is a t-error-correcting pair for the function type AG code $C = C_L(D, G)$:

$$A := C_L(D, H), \quad B := C_{\Omega}(D, G + H).$$

Similarly, if we put

$$t = \left\lfloor \frac{\deg(G) + 1 - 3g + 2\iota(G)}{2} \right\rfloor,$$

then for H satisfying (5), the following is a t-error-correcting pair for the residue type AG code $C = C_{\Omega}(D, G)$:

$$A := C_L(D, H), \quad B := C_L(D, G - H).$$

§4. Theory of Miura curve.

Let a, b be a pair of coprime natural numbers satisfying $2 \leq a < b$. Miura[M] gave a way to implement AG codes employing algebraic curves of the following concrete form with coefficients in F_q:

$$f(X, Y) := \sum_{0 \leq ai + bj \leq ab} c_{ij} X^i Y^j = 0, \quad \text{where } c_{b0} \neq 0,\ c_{0a} \neq 0. \tag{6}$$

We call a curve of this form a C_a^b curve or a Miura curve. (6) can be rewritten as follows:

$$Y^a + \sum_{0 \leq ai + bj \leq ab, i \neq 0} a_{ij} X^i Y^j = X^b + \sum_{0 \leq i < b} a_i X^i. \tag{7}$$

$a = 2$, $b = 3$ in (7) just corresponds to the Weierstrass normal form of the elliptic curve. $a = 2$, $b = 2g + 1$ corresponds to a hyperelliptic curve of genus g. We have in general

Proposition 1) Miura curve is absolutely irreducible over F_q, that is, any extension of coefficients does not allow factorization.

2) C_a^b has only one point $P_\infty = (0 : 1 : 0)$ at infinity. This is singular in general, but of degree 1, and resolution of singularity does not cause new points.

3) Let $F_q[x, y] := F_q[X, Y]/(f(X, Y))$ be the coordinate ring of C_a^b, K be its quotient field. Then we have

$$(x)_\infty = aP_\infty, \quad (y)_\infty = bP_\infty, \quad F_q[x, y] \subset L(\infty P_\infty) \subset K = F_q(x, y),$$

hence, $N_{P_\infty} := -\nu_{P_\infty}(L(\infty P_\infty)) \supset -\nu_{P_\infty}(F_q[x, y]) = aN + bN$. Here $N = \{0, 1, 2, \ldots\}$.

4) The genus g of C_a^b satisfies

$$g \leq \frac{(a-1)(b-1)}{2} = \#\{N \setminus N_{P_\infty}\}.$$

Here the equality holds if and only if C_a^b is non-singular as an affine curve, which is equivalent to $F_q[x, y] = L(\infty P_\infty)$.

§5. Realization of algebraic-geometric codes by Miura curves.

Now we construct concrete AG codes employing Miura curves. The first task is to choose a plane curve V over F_q which has as many F_q-rational points $V(F_q)$ as possible. The known theoretical bound due to Hasse-Weil-Serre is

$$\#V(F_q) \leq q + 1 + g\lfloor 2\sqrt{q} \rfloor.$$

Actually this final form is due to Serre, and Hasse-Weil gave the estimate without $\lfloor \cdot \rfloor$. Traditionally, we call a curve maximal if the equality holds in the latter weaker estimate. (Hence necessarily q should be an even power of p.) We call a curve optimal if it attains the maximum number of rational points for a given fixed genus g. Maximal curves are necessarily optimal, but optimal curves need not be maximal. See van der Geer's website ([vG]) for actual record.

The curve we are going to consider is one found by M. Kawakita [KS], which is a C_3^4 curve over F_{2^8} with the defining equation

$$Y^3 + \alpha^{101}Y = X^4 + \alpha^{29}X^2 + \alpha^{18}X + 1. \tag{8}$$

Here $\alpha \in F_{2^8}$ is a primitive element satisfying $\alpha^8 + \alpha^4 + \alpha^3 + \alpha^2 + 1$. This is non-singular except for the point at infinity, with the genus $g = 3$. The number of rational points including one at infinity is equal to 353. It was the first known concrete example of maximal curve for this genus. A part of the list of finite rational points is given below. Here, the hexadecimal numbers correspond to the coefficients of elements of F_{2^8} represented as polynomials of order 7 in α over F_2. For example, the first ce represents $\alpha^7 + \alpha^6 + \alpha^3 + \alpha^2 + \alpha$.

Table Partial list of rational points of C_3^4 curve (8).

(00,ce)	(00,37)	(00,f9)	(01,f0)	(01,fd)	(01,0d)	(02,e5)	(08,fc)	(10,4e)	(40,95)
(80,fe)	(80,85)	(80,7b)	(1d,be)	(1d,cc)	(1d,72)	(3a,87)	(3a,d3)	(3a,54)	(74,05)
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
(e9,84)	(cf,07)	(1b,3b)	(6c,08)	(d8,2f)	(ad,42)	(47,98)	(47,50)	(47,c8)	(8e,75)
(8e,d7)	(8e,a2)								

To construct a concrete function type AG code, we choose as the divisor D the sum of all the above rational points. Hence, $n = 352$. We choose $G = 300P_\infty$ among the region where $\iota(G) = 0$, $l(G - D) = 0$ are satisfied. Then by the Riemann-Roch theorem $k = \deg(G) + \iota(G) - g + 1 - l(G - D) = 298$. (In our implementation the latter value is automatically determined from the input image width and the code is designed after that. The value 298 corresponds to the concrete example considered below.) Then the Goppa bound gives $d(C_L(D, G)) \geq n - \deg(G) + 2l(G - D) = 52$. Together with the Singleton bound we have $52 \leq d \leq 55$. To determine codewords, we fix the basis of $L(\infty P_\infty)$ to

$$\{X^iY^j \mid 0 \leq j \leq 2, i = 0, 1, 2, \ldots\}.$$

Now $k = l(G) - l(G - D) = l(G)$, hence those belonging to $L(300P_\infty)$ therein has degree $3i + 4j \leq 300$, and they are the following 298 monomials:

$$\{1, X, Y, X^2, XY, Y^2, X^3, X^2Y, XY^2, X^4, X^3Y, X^2Y^2, \ldots, X^{99}, X^{98}Y, X^{97}Y^2, X^{100}\}. \tag{9}$$

By the way, the value of $-\nu_{P_\infty}$ for these is $\{0, 3, 4, 6, 7, 8, \ldots, 300\}$. Hence the gap values are $\{1, 2, 5\}$, in concordance with the general formula $5 = 2g - 1$. The basis of the codeword subspace is given as the numerical vectors obtained from the polynomials in (8) by substituting the coordinate values of 352 points in Table. The general codewords are obtained as linear combinations of these, hence there are $q^k = 2^{8\times 298}$ words. Therefore to determine the minimum distance directly from the weights of these is impracticable. Executing the fundamental deformation to the basis, however, we obtain an echelon form matrix

$$\begin{array}{cccccccccc}
 & \multicolumn{6}{c}{\longleftarrow\ 298\ \longrightarrow} & \multicolumn{3}{c}{\longleftarrow 54 \longrightarrow}\\
\uparrow & 1 & 0 & 0 & 0 & \cdots & 0 & * & \cdots & *\\
 & 0 & 1 & 0 & 0 & \cdots & 0 & * & \text{general} & *\\
298 & 0 & 0 & 1 & 0 & \cdots & 0 & * & \text{comp.} & *\\
\downarrow & & & & & \ddots & & & \vdots & \\
 & 0 & 0 & 0 & 0 & \cdots & 1 & * & \cdots & *
\end{array}$$

which contains a row with two zeros in the uncleaned "general" part to the right. Thus we obtain a better estimate $52 \leq d \leq 53$. Finally, a probe by computer readily gives a new codeword of weight 52, the first row + the 234-th row× f4. Thus we conclude that $d = 52$.

Note that for this curve a general algorithm finding the true minimum distance is not known[2]. Computer experiments show that for each of the residue type code spaces in the range $4 \leq k \leq 347$ ($k \neq 345$), we can find a vector with the Goppa designed distance just in the same way. As a matter of fact, in most cases the basis already contains a vector of minimum weight. in a few exceptional cases the linear combination of two basis vectors provides one as in the example above, and in a very rare exception of the case $k = 342$ the linear combination of three does. For $k = 345$ and the remaining marginal values of k the true minimum distances are really bigger than the Goppa designed distances. But they are of no importance for practical use. (In the case $k = 345$ the true minimum distance is $6 =$ the Goppa bound $+1$, which was determined by verifying that any 5 columns of the parity check matrix are linearly independent.

Employing the deformed basis we can realize systematic encoding, and embed an image as it is. As an example of application to images, we show below one with 256 grey scale, with 298 pixels in each row. (The image is taken from the standard file "lena". We contracted and cut it to 298× 307 pixels for our experiment. Of course the standardness of the employed image has no significance for the purpose of this article.) Decoding algorithm is Pellikaan's one explained in §3. We have

$$t = \left\lfloor \frac{n - \deg G - 1 - g}{2} \right\rfloor = \frac{352 - 300 - 1 - 3}{2} = 24.$$

According to §3, we have, for $H = (t + g)P_\infty = 27P_\infty$, $A = C_L(D, H) = C_L(D, 27P_\infty)$. We can assign the first 25 basis element of $C_L(D, G)$ to A. Hence $\dim A = 25$. Further, $B = C_\Omega(D, G + H) = C_L(D, G + H)^\perp$. We apply the construction method for $C_L(D, G)$ to a larger one $C_L(D, 327P_\infty)$, and from the generating matrix with 325 rows we calculate its parity check matrix to obtain the basis of B. This gives $\dim B = 352 - 325 = 27$. The remaining parts are realized by simple matrix calculations. For this example, the theoretical decoding bound $\lfloor (d_G - 1)/2 \rfloor = 25$, which can be realized by a more sophisticated algorithm of Feng-Rao, is not so big than our t. The damages are given by mouse operation to replace the indicated rectangles by white. Those places where decoding failed has size of damage exceeding the decoding bound 24 pixels. Notice that if we employ Reed-Solomon code on $\boldsymbol{F}_{2^8}$ allowing the same correctability, the maniable image width will be at most 256-49+1=208.

For the practical implementation, $B^\perp = C_L(D, G{+}H)$ is bigger than the requred code space $C_L(D, G)$, and calculating its dual B is heavy though B itself is small. For this reason, it is more advantageous to employ the residue type codes: We first construct $C^\perp = C_L(D, G)$ with small size by a procedure as explained above, and then employing it as a parity check matrix, calculate the dual code C of the required

[2] For maximal curves of Hermite type this fact is well known ([KY]). But we expect that our example does not reduce to that class, though it is not yet checked.

size. In the above example, $G = 56P_\infty$, $C^\perp = C_L(D,G)$ has dimension $\deg(G) + \iota(G) - g + 1 - l(G-D) = 54$, hence C has dimension $k = n - \dim C_K(D,G) = 352 - 54 = 298$ as required. Given that the parity check matrix is reduced to the echelon form, its dual C is also obtained as such and systematic encoding is instantaneous.

Our implementation equally works with color images, by means of direct sum codes of three copies of the above, each applied to RGB component. Actually our program distinguishes whether the input image is monochrome or color, and chooses the suitable code.

Acknowledgements

The authors expess their sincere gratitude to Dr. Shinji Miura for guiding us to the theory of AG codes through many personal lectures. They also thank Dr. Motoko Kawakita who kindly informed us her unpublished new results and stimulated us.

References

[E] E. Shimizu : Implementation of Algebraic-Geometric Codes and its Applications, *Master's thesis*, Ochanomizu University, 2002.

[G] V. G. Goppa: Codes on algebraic curves, *Soviet Math. Doklady*, **24** (1981), 170–172.

[KS] M. Kawakita (Suzuki) : Algebraic Curves of Genus Three over Finite Fields of Characteristic Two with Many Rational Points, *Master's thesis*, Ochanomizu University, 1999.

[KY] P. V. Kumar and K. Yang : On the true minimum distance of Hermitian codes, in *AGCT-3, Springer Lecture Notes in Math.* **No. 1518**, pp.99–107, 1992.

[M] S. Miura : Study of Error Correcting Codes based on Algebraic Geometry (in Japanese), *Thesis*, University of Tokyo, 1997.

[MII] S. Miura, Y. Iwadare, H. Imai : Mathematical theory of algebraic geometric codes (in Japanese), *IEICE Trans.* Vol. **J82-A, No.8** (1999), 1223–1238.

[P] R. Pellikaan : On decoding linear codes by error correcting pairs, *preprint*, Eindhoven University of Technology, 1988.

[SV] A. N. Skorobogatov, S. G. Vladut : On the decoding of algebraic-geometric codes, *IEEE Trans. Information Theory*, **36** (1990), 1451–1463.

[vG] G. van der Geer, M. van der Vlugt: Table for the function $N_q(g)$, `http://www.wins.uva.nl/~geer`.

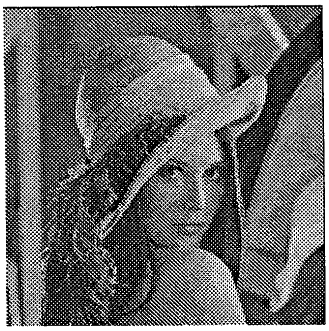

Fig. 1 : Original Image

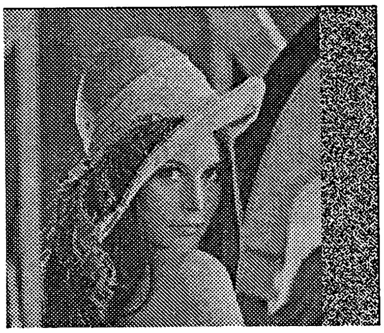

Fig. 2 : Redundancy added

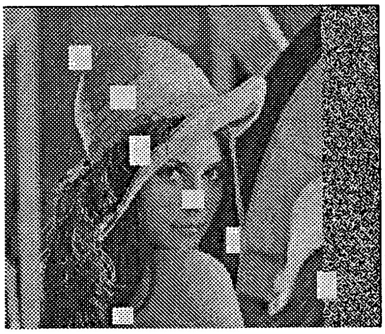

Fig. 3 : Damaged (at 4 places)

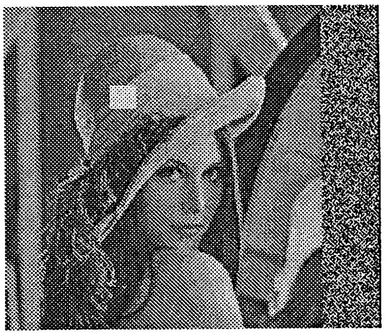

Fig. 4 : Decoded (recovering 3 places)