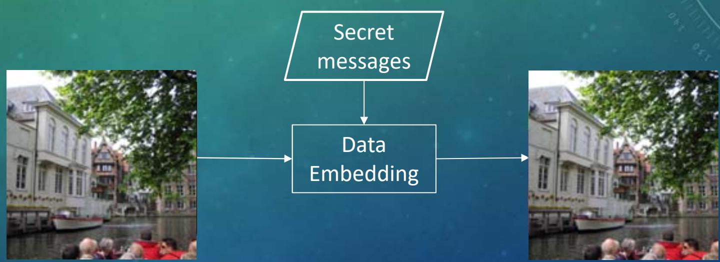
A NOVEL GENERAL MULTIPLE-BASE DATA EMBEDDING ALGORITHM

W. S. CHEN, Y. K. LIAO, Y. T. LIN, AND C. M. WANG, "A NOVEL GENERAL MULTIPLE-BASE DATA EMBEDDING ALGORITHM," INFORMATION SCIENCES, VOL. 358-359, PP. 164-190, SEPTEMBER 2016.

Chung-Ming Wang
Department of Computer Science and Engineering
National Chung Hsing University
Taichung, Taiwan.
cmwang@cs.nchu.edu.tw

INTRODUCTION

- Data Embedding Algorithm: An algorithm that embeds secret messages into a host medium in order to conceal secret information
- Data embedding: (1) Modifying Existing Media or (2) Constructing an existing media
  - Digital image
  - Text
  - Audio
  - Video
  - 3D model

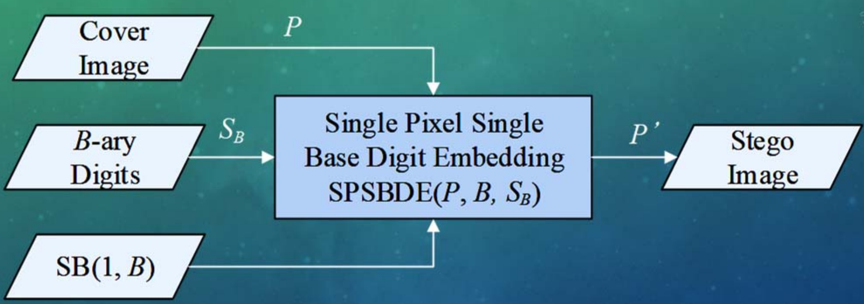


- Embedding Efficiency
  - Ratio = Capacity / Quality
- Image quality assessment, Steganalysis

OUTLINE

- The single-base data embedding approach (SB)
- The multiple-base data embedding scheme (MB)
- The General Multiple-base Data Embedding Algorithm (GMB)

THE **SINGLE-BASE** DATA EMBEDDING APPROACH (SB) 1/6



- Assumption:
- Cover image is given
- B is known (B: base)
- Secret message: B-ary digits, ignoring conversion between a real binary secret message and base B

THE **SINGLE-BASE** DATA EMBEDDING APPROACH (SB) 2/6

- Define a secret ***B*-ary** digit: S_B , cover pixel: P , the embedding function **SPSBDE(P, B, S_B)** .
- First: calculating remainder of the existing pixel value $r = P \bmod B$. Eq. (1)
- Second: calculating the differences value $d = [(S_B - r) + B] \bmod B$. Eq. (2)
- Message Embedding:

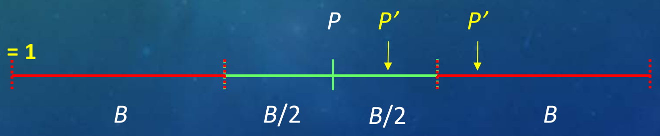
$$P' = \begin{cases} P & \text{if } d = 0 \\ P + d & \text{if } d < B/2 \\ P \pm d & \text{if } d = B/2 \\ P + d - B & \text{if } d > B/2 \end{cases} \qquad \text{Eq. (3)}$$

Saturated pixel:

$$P'' = \begin{cases} P' + B, & \text{if } P' < 0 \\ P' - B, & \text{if } P' > 255 \\ P', & \text{otherwise} \end{cases} \qquad \text{Eq. (4)}$$

P' = 18+3−5 = **16** .

Extraction: S_5 = 16 mod 5 = **1**



THE **SINGLE-BASE** DATA EMBEDDING APPROACH (SB) 3/6

EXAMPLE 1: NORMAL PIXEL

- SPSBDE(P, B, S_B) = (18, 5, 1) P: pixel value, B: base, S_B: secret B-ary digit
- Cover pixel P = **18** , a secret **5-ary** digit S_5 = **1** .
- First: calculating remainder of the existing pixel value r = 18 mod 5 = **3** .
- Second: calculating th edifferences value d = [(1 − 3) + 5] mod 5 = **3** .
- **Embedding:** since $3 \geq 5/2$

$$P' = \begin{cases} P & \text{if } d = 0 \\ P + d & \text{if } d < B/2 \\ P \pm d & \text{if } d = B/2 \\ P + d - B & \text{if } d > B/2 \end{cases} \qquad \text{Eq. (3)}$$

P' = 18+3−5 = **16** .

No saturation. P''=P' .

Extraction: S_5 = 16 mod 5 = **1** .

THE **SINGLE-BASE** DATA EMBEDDING APPROACH (SB) 4/6

EXAMPLE 2: SATURATED PIXEL

- SPSBDE(P, B, S_B) = (255, 5, 1)
- Cover pixel ***P* = 255**, a secret ***5*-ary** digit ***S_5* = 1** .
- Remainder of the pixel value ***r*** = 255 mod 5 **= 0** .
- Differences value ***d*** = [(1 − 0) + 5] mod 5 **= 1** .
- **Embedding:** since $1 < \lfloor 5/2 \rfloor$

$$P' = \begin{cases} P & \text{if } d = 0 \\ P + d & \text{if } d < \mathrm{B}/2 \\ P \pm d & \text{if } d = \mathrm{B}/2 \\ P + d - B & \text{if } d > \mathrm{B}/2 \end{cases} \quad \text{Eq. (3)}$$

Saturated pixel:

$$P'' = \begin{cases} P' + B, & \text{if } P' < 0 \\ P' - B, & \text{if } P' > 255 \\ P', & \text{otherwise} \end{cases} \quad \text{Eq. (4)}$$

P' = 255+1 **= 256** .
P'' = 256−5 **= 251** .
Extraction: ***S_5*** = 251 mod 5 **= 1** .

7

EXPECTED MEAN SQUARE ERROR, EMSE(B) 5/6

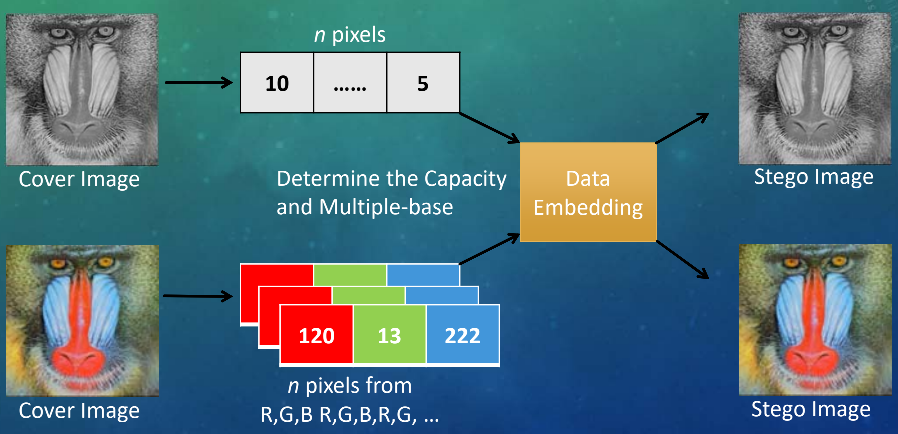
- Given an odd base b, e.g. b=5
- Total Square Error = TSE = $0^2+1^2+2^2+(-2)^2+(-1)^2$
- Expected Mean Square Error for b: EMSE(5)=TSE/5=10/5=2
- A general case for b, TSE = $0^2+1^2+2^2+\ldots+(\frac{(b-1)}{2})^2 + (-\frac{(b-1)}{2})^2 \ldots + (-2)^2+(-1)^2$
- Expected Mean Square Error for a base b : EMSE(b)= TSE/b =$\frac{b^2-1}{12}$
- Verify b=5, EMSE(5)=$\frac{5^2-1}{12}$=2

8

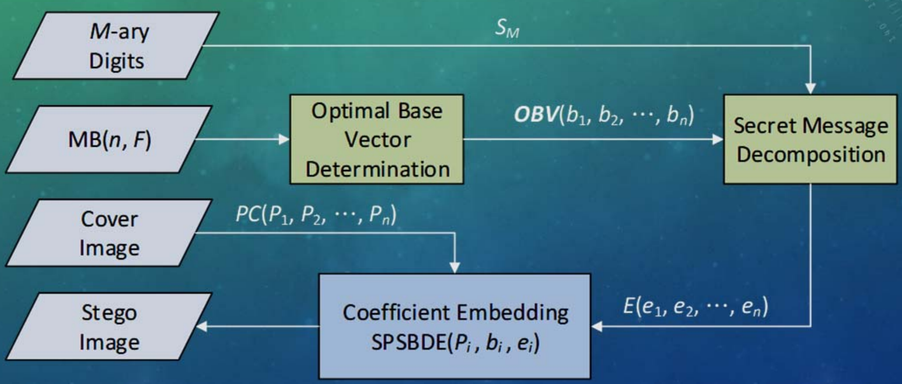
EXPECTED MEAN SQUARE ERROR, EMSE(B) 6/6

- Given an eve base b, e.g. b=6
- Total Square Error = TSE = $0^2+1^2+2^2+3^2+(-2)^2+(-1)^2$
- Expected Mean Square Error EMSE(6)=TSE/6=$\frac{19}{6}$
- A general case for b, TSE = $0^2+1^2+2^2+\ldots+(\frac{b}{2})^2 + [-(\frac{b}{2}-1)]^2\ldots+(-2)^2+(-1)^2$
- Expected Mean Square Error for a base b : EMSE(b)= TSE/b =$\frac{b^2+2}{12}$
- Verify: b=6, EMSE(b)= $\frac{6^2+2}{12}=\frac{19}{6}$
- A general for any base b: MSE(b)=$\frac{b^2-(-2)^{(b+1) \bmod 2}}{12}$

THE **MULTIPLE-BASE** DATA EMBEDDING SCHEME (MB) 1/5



THE **MULTIPLE-BASE** DATA EMBEDDING SCHEME (MB) 1/5



11

THE **MULTIPLE-BASE** DATA EMBEDDING SCHEME (MB) 2/5

SECRET MESSAGE DECOMPOSITION

- Define a desired secret ***F*-ary** digit: S_F , pixel-cluster of *n* pixels: ***PC*(P_1, P_2, …, P_n)**, and assume we have derived an optimal base vector: ***OBV*(b_1, b_2, …, b_n)**.
- Let M= $\prod_{i=1}^{n} b_i$ be the multiplication of every component in OBV
- Since M≥F, we can conceal an M-ary digit rather than a F-nary digit
- First, we use Eq. (14) to calculate message coefficients ***E*(e_1, e_2, …, e_n)** for embedding.
- $$e_i = \begin{cases} S_M \bmod b_i, & \text{if } i = 1 \\ \left\lfloor \dfrac{S_M}{\prod_{j=1}^{i-1} b_j} \right\rfloor \bmod b_i, & \text{if } 2 \le i \le n, \end{cases} \qquad \text{Eq. (14)}$$
- Then embedding each coefficient e_i into pixel P_i by Eqs. (1) to (4).

12

THE **MULTIPLE-BASE** DATA EMBEDDING SCHEME (MB) 3/5

EXAMPLE 3:

- MB(n, F)=(3, 35), we assume the optimal base vector: ***OBV*(3, 3, 4)**. M=3*3*4=36 > F=35
- The secret **36-ary** digit: **32_{36}** , pixel-cluster of 3 pixels: ***PC*(90, 100, 110)**, and the optimal base vector: ***OBV*(3, 3, 4)**.
- First, we calculate message coefficients:
- $e_1 = 32 \bmod 3 = 2, \quad e_2 = \left\lfloor \frac{32}{3} \right\rfloor \bmod 3 = 1, \quad e_3 = \left\lfloor \frac{32}{3\times3} \right\rfloor \bmod 4 = 3.$
  - Then embedding the coefficient e_1 = **2** into pixel P_1 = **90** by Eqs. (1) to (4):
  - Remainder of the pixel value r_1 = 90 mod 3 = **0** .
  - Difference value d_1= [(2 – 0) + 3] mod 3 = **2** .
  - Since 2 ≥ ⌊3/2⌋, the stego $P'_1 = P_1 + d_1 - b_1$ = 90 + 2 – 3 = **89**.
  - e_2 , e_3 into P_2 , P_3 , respectively.
- Final stego pixels-cluster = ***PC*(89, 100, 111)**.

THE **MULTIPLE-BASE** DATA EMBEDDING SCHEME (MB) 4/5

FIND THE OPTIMAL BASE VECTOR

- The expected mean squared error (EMSE) can be estimated:
  - $\mathrm{EMSE}(B) = \frac{\left[B^2 - (-2)^{(B+1) \bmod 2}\right]}{12}$ Eq. (5)
  - $\mathrm{Base}(1,5,7) = \cancel{\frac{1^2-(-2)^{(1+1) \bmod 2}}{12}} + \frac{5^2-1}{12} + \frac{7^2-1}{12} = \frac{72}{12} = 6.$ Base(3,3,4) $= \frac{3^2-1}{12} + \frac{3^2-1}{12} + \frac{4^2+2}{12} = \frac{34}{12} = 2.8\overline{3}$
- **Example 4:** Given MB(n, F) = (2, 5)

{SEV (s_1, s_2)} Q_{SEV} = 15	{MBV (v_1, v_2)} Q_{MBV} = 10	EMSE(MBV (v_1, v_2))
(1, 1), (1, 2), (1, 3), (1, 4), (1, 5)	(1, 5)	1.0000
(2, 2), (2, 3), (2, 4), (2, 5)	(2, 3), (2, 4), (2, 5)	0.5833, 1.0000, 1.2500
(3, 3), (3, 4), (3, 5)	(3, 3), (3, 4), (3, 5)	0.6665, 1.0833, 1.3333
(4, 4), (4, 5)	(4, 4), (4, 5)	1.5000, 1.7500
(5, 5)	(5, 5)	2.0000

  - EMSE(MBV(2, 3)) = [EMSE(2) + EMSE(3)] / 2 = [0.5 + 0.66667] / 2 = $0.58\overline{3}$

DETERMINE THE OPTIMAL BASE VECTOR

- Given MB(n, F)
- n: how may pixels are clustered to form a basic embedding unit
- F: the maximal target notational system we intend to convey
- Example: MB(2, 5): 2 pixels are clustered as an embedding unit, the maximal 5-ary notational system
- Brute force approach: four steps
- Step 1: determine the fundamental base component, b, where b=$\lceil \sqrt[n]{F} \rceil$.
- Step 2: construct the fundamental base vector, FBV=(b, b, …b)
- Step 3: evaluate the distortion in terms of the mean square error for every candidate base vector, CBV_i, using n nested "for" loops: the upper bound is +2 and the lower bound is -2
- Step 4: select the vector that has the *minimal* mean square error as the optimal base vector, OBV

DETERMINE THE OPTIMAL BASE VECTOR-EXAMPLE

- Given MB(2, 5)
- Step 1: determine the fundamental base component, b, where b=$\lceil \sqrt[n]{F} \rceil$= $\lceil \sqrt[2]{5} \rceil$= $\lceil 2.236 \rceil$=3
- Step 2: construct the fundamental base vector, FBV=(b, b, …b)=(3, 3)
- Step 3: evaluate the EMSE for every candidate base vector, CBV_i , using n nested "for" loops
- Note: the "for" loops can be more efficient if we setup some conditions

ALGORITHM TO DETERMINE OBV: MB(2, 5)

```
for i=-2 to +2
  for j=-2 to +2
    CBV_i=(b, b)+(i, j)
    if legal (multiply >=F) and not repeat (b1<=b2) and legal component (>=2)
      Dist_i=EMSE(CBV_i)
    else
     next i, j
OBV= Dist_i that has the min{Dist_i}
```

- Setup three conditions to speedup the computing
- Condition 1: multiply all components $\geq$ F
- Condition 2: components are in ascending order, b1$\leq$b2$\leq$ …,
- Condition 3: components $\geq$ 2

17

THE OPTIMAL BASE VECTOR-MB(2, 5): OBV(2, 3)

n	F	i	j	b1'	b2'	Distortion	OBV
2	5	-1	0	2	3	0.583333	Yes
2	5	0	0	3	3	0.666667	
2	5	-1	1	2	4	1.0	
2	5	0	1	3	4	1.083333	
2	5	-1	2	2	5	1.25	
2	5	0	2	3	5	1.333333	
2	5	1	1	4	4	1.5	
2	5	1	2	4	5	1.75	
2	5	2	2	5	5	2.0	

18

See the values shown in the excel file

THE EMBEDDING CAPACITY MAY BE CHANGED

- Given MB(n, F), we determine an optimal base OBV=$(b_1, b_2, \ldots, b_n)$
- Our initial intention is to embed an F-ary digit in an n-pixel cluster
- The embedding capacity (EC) per n-pixel cluster in bits is: EC=$log_2(F)$
- The embedding rate in bpp is ER=$\frac{EC}{n} = \frac{log_2(F)}{n}$
- Let M= $\prod_{i=1}^{n} b_i$ be the multiplication of every component in OBV
- Note that $M \geq F$ but not necessary M=F.
- Mmaking use of M rather than F provides any benefits?

19

MAKING USE OF M-ARY BUT NOT F-ARY

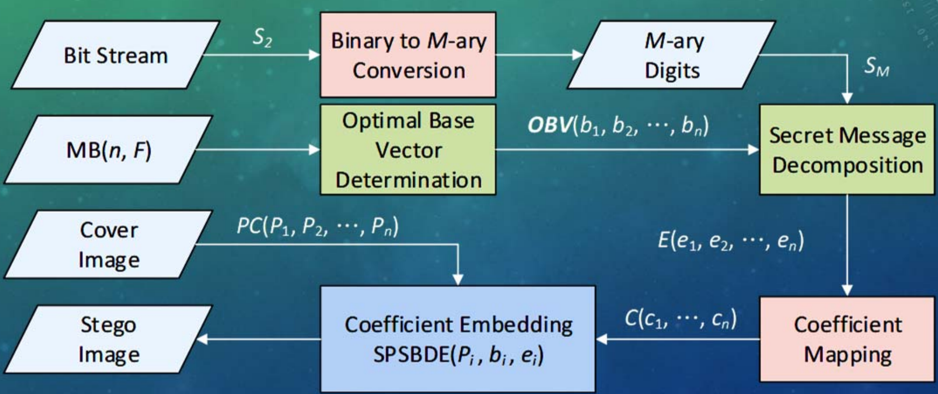
- We can make use of M rather than F to embed more secret bits without increasing the distortion
- The updated embedding capacity per n-pixel cluster in bits is: EC=$log_2(M)$
- The updated embedding rate is ER=$\frac{EC}{n} = \frac{log_2(M)}{n}$
- In our example
- Original EC=$log_2(F)$=$log_2(5)$=2.3219 bits
- Updated EC=EC=$log_2(M)$=$log_2(6)$=2.5850 bits

20

THE MULTIPLE-BASE DATA EMBEDDING SCHEME (MB) 5/5

	Elaine	Boat	Baboon	Lena
	MB(5, 109)	MB(4, 81)	MB(3, 1008)	MB(5, 499409)
OBV	(2,3,3,3,3)	(3,3,3,3)	(9,11,11)	(13,13,14,15,15)
M or F	M=162 > F=109	M=81=F=81	M=1089>F=1008	M=532350>F=499409
MSE_P	0.633	0.667	8.889	16.367
MSE_R	0.633	0.668	8.908	16.355
$\|Diff\|$	**0.0**	**0.001**	**0.019 (0.02%)**	**0.012 (0.06%)**

GENERAL MULTIPLE-BASE DATA EMBEDDING ALGORITHM (GMB) 1/14



GENERAL MULTIPLE-BASE DATA EMBEDDING ALGORITHM (GMB) 2/14

BINARY TO *M*-ARY CONVERSION

$\longleftarrow$ *M* $\longrightarrow$

MC	$0 \sim (2^k - 1)$		$2^k \sim (M - 1)$
	k		0
TC	$0 \sim (M - 2^k - 1)$	$(M - 2^k) \sim (2^k - 1)$	$2^k \sim (M - 1)$
	$k + 1$	k	$k + 1$

- An Example of the **Minimal Concealment (MC)**:
- Given GMB(2, 7), so n = 2, M = 7-ary, then MC: $\boldsymbol{k} = \lfloor \log_2 7 \rfloor = \mathbf{2}$
- But theoretical: $\log_2 7 = 2.807$

23

GENERAL MULTIPLE-BASE DATA EMBEDDING ALGORITHM (GMB) 3/14

BINARY TO *M*-ARY CONVERSION

$\longleftarrow$ *M* $\longrightarrow$

TC	$0 \sim (M - 2^k - 1)$	$(M - 2^k) \sim (2^k - 1)$	$2^k \sim (M - 1)$
	$k + 1$	k	$k + 1$

- An Example of the **Triplet Concealment (TC)**:
- Given GMB(n, M)=(2, 7), so n = 2, M = 7-ary, then $\boldsymbol{k} = \lfloor \log_2 7 \rfloor = \mathbf{2}$
- The range can embed ***k*+1** bits = 0~2 (Segment I) or 4~6 (Segment III).
- Assumed message bit string = "101001111…".
- First pick "10" and check range.
- Package the third bit "1" into the MSB = "110".
- The embedding message = **6**.

m	*k*+1	*k*	*bits*
0	0	**00**	3
1	0	**01**	3
2	0	**10**	3
3		**11**	2
4	1	00	3
5	1	01	3
6	1	10	3

Exp. Avg. bits
(3+3+3+2)/4
= 2.75

Theo. **2.807**

24

GENERAL MULTIPLE-BASE DATA EMBEDDING ALGORITHM (GMB) 4/14

BINARY TO *M*-ARY CONVERSION

- An Example of the **Notational Concealment (NC)**
- Given GMB(2, 7), so n = 2, M = 7-ary, the $\boldsymbol{k} = \lfloor \log_2 7 \rfloor = \mathbf{2}$, and in theory $\log_2 7 = 2.807$
- Image resolution $H \times V = 16$, number of group $U = \lfloor (H \times V)/n \rfloor = 8$.

P_1	P_2	P_3	P_4
P_5	P_6	P_7	P_8
P_9	P_{10}	P_{11}	P_{12}
P_{13}	P_{14}	P_{15}	P_{16}

D = 2
MC: $\lfloor \log_2 7 \rfloor + \lfloor \log_2 7 \rfloor = 4$ bits.
NC: $\lfloor \log_2 7 \times 7 \rfloor = 5$ bits. (Avg. Cap. = 2.5)

D = 3
MC: $\lfloor \log_2 7 \rfloor + \lfloor \log_2 7 \rfloor + \lfloor \log_2 7 \rfloor = 6$ bits.
NC: $\lfloor \log_2 7 \times 7 \times 7 \rfloor = \lfloor \log_2 343 \rfloor = 8$ bits. (Avg. Cap. = $2.\overline{6}$)

- $D_{opt} = \arg\max_D \left\{ \left\lfloor \frac{U}{D} \right\rfloor \times L \right\}$,

  $1 \le D \le U$ and $L = \lfloor D \times \log_2 M \rfloor \le 64$ Eq. (18)
- $L_{opt} = \lfloor D_{opt} \times \log_2 M \rfloor$ Eq. (19)

GENERAL MULTIPLE-BASE DATA EMBEDDING ALGORITHM (GMB) 5/14

BINARY TO *M*-ARY CONVERSION

P_1	P_2	P_3	P_4
P_5	P_6	P_7	P_8
P_9	P_{10}	P_{11}	P_{12}
P_{13}	P_{14}	P_{15}	P_{16}

Case of the GMB(2, 7) total Cap.
EC_{MC} = 16 bits.
EC_{NC} = 22 bits. ($D_{opt} = 4, L_{opt} = 11$)
EC_{THE} = 22.458 bits.

- Expected capacity, *EC*:
- $EC_{MC} = k \times U$ Eq. (30)
- $EC_{TC} = \left(k + \frac{M - 2^k}{2^k} \right) \times U$ Eq. (32)
- $EC_{NC} = \lfloor D_{opt} \times \log_2 M \rfloor \times \left\lfloor \frac{U}{D_{opt}} \right\rfloor = L_{opt} \times \left\lfloor \frac{U}{D_{opt}} \right\rfloor$ Eq. (33)
- $EC_{THE} = (\log_2 M) \times U$ Eq. (34)

GENERAL MULTIPLE-BASE DATA EMBEDDING ALGORITHM (GMB) 6/14

BINARY TO M-ARY CONVERSION

n	M	D_{opt}	L_{opt}	C_{NC}	C_{THE}	Difference	Loss (%)
4	13	10	37	242461	242512.0	51.02	0.021
5	93	2	13	340782	342835.0	2053.02	0.599
6	250	8	63	344043	348025.1	3982.12	1.144
7	789	5	48	359472	360404.7	932.74	0.259
8	5329	3	37	404114	405656.3	1542.34	0.380
9	7125	4	51	371331	372787.0	1455.99	0.391
10	25489	2	29	380103	383709.7	3606.71	0.940
11	59871	4	63	375291	378187.7	2896.72	0.766
12	934587	3	59	429579	433273.1	3694.06	0.853
13	3025894	2	43	433526	434109.3	583.34	0.134

GENERAL MULTIPLE-BASE DATA EMBEDDING ALGORITHM (GMB) 7/14

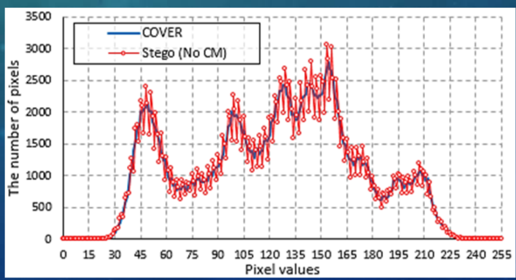
Images	Type	n	M	Cap.	MSE_P	MSE_R	PSNR	Efficiency
Elaine	NC	5	162	384472	0.633	0.633	50.12	2.31697
Boat	NC	4	81	415055	0.667	0.668	49.89	2.37022
Pepper	NC	6	1215	445638	0.889	0.887	48.65	1.91654
Couple	DC	6	9375	571146	1.778	1.81	45.55	1.20373
Tiffany	MC	9	2343750	611667	2.13	2.161	44.78	1.07974
Gold hill	MC	8	4235364	720896	3.792	3.777	42.36	0.72809
Barbara	TC	9	67765824	757576	4.63	4.634	41.47	0.62363
Baboon	TC	3	1089	879339	8.889	8.908	38.63	0.37656
Jet	DC	5	131769	891578	9.35	9.351	38.42	0.36372
Lena	TC	5	532350	996942	16.367	16.355	35.99	0.23253

GENERAL MULTIPLE-BASE DATA EMBEDDING ALGORITHM (GMB) 8/14

COEFFICIENT MAPPING

- Solve MC, DC, and TC, message coefficients problem and histogram fluctuation
- $$e_i = \begin{cases} S_M \bmod b_i, & \text{if } i = 1 \\ \left\lfloor \dfrac{S_M}{\prod_{j=1}^{i-1} b_j} \right\rfloor \bmod b_i, & \text{if } 2 \le i \le n, \end{cases}$$ Eq. (14)

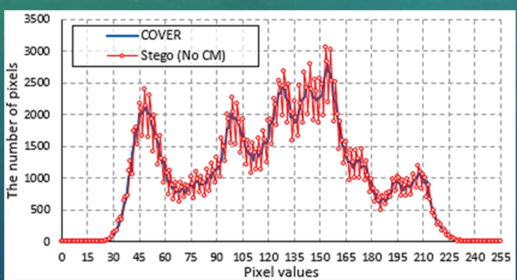
P_1	P_2	P_3	P_4
P_5	P_6	P_7	P_8
P_9	P_{10}	P_{11}	P_{12}
P_{13}	P_{14}	P_{15}	P_{16}



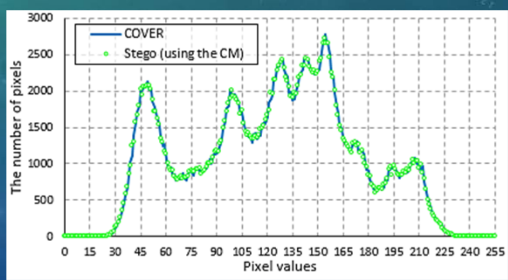
GENERAL MULTIPLE-BASE DATA EMBEDDING ALGORITHM (GMB) 9/14

COEFFICIENT MAPPING

- The concept of coefficient mapping (CM): shift message by a key.
- Lena, no CM :



Using CM:



GENERAL MULTIPLE-BASE DATA EMBEDDING ALGORITHM (GMB) 10/14

COEFFICIENT MAPPING

- Cyclic substitution function (CSF). Denote cover accumulate number is EN_j, stego is XN_j, where $j = 1, 2, ..., U$, $U = \lfloor (H \times V) / n \rfloor$.

P_1	P_2	P_3	P_4
P_5	P_6	P_7	P_8
P_9	P_{10}	P_{11}	P_{12}
P_{13}	P_{14}	P_{15}	P_{16}

$EN_1 = 1$, $EN_2 = 2$, $EN_3 = 3$, ..., $EN_8 = 8$

- **Embedding:**
  - $r_i = EN_j \bmod b_i$, $i = 1, 2, ..., n$ and $j = 1, 2, ..., U$ Eq. (25)
  - $c_i = \mathrm{CSF}(e_i, r_i, b_i) = (e_i + r_i) \bmod b_i$, $i = 1, 2, ..., n$ Eq. (26)
- **Extraction:**
  - $r_i' = XN_j \bmod b_i$, $i = 1, 2, ..., n$ and $j = 1, 2, ..., U$ Eq. (27)
  - $e_i' = \mathrm{RCSF}(c_i', r_i', b_i) = (c_i' - r_i') \bmod b_i$, $i = 1, 2, ..., n$ Eq. (28)

Embedding: $(e_i = 0, EN_{12})$

$r_i = EN_{12} \bmod 5 = 2$

$c_i = (0 + 2) \bmod 5 = 2$

Extraction: $(XN_{12}, c_i = 2)$

$r_i' = XN_{12} \bmod 5 = 2$

$e_i' = (2 - 2) \bmod 5 = 0$

CONCLUSIONS

- ***(1)*** we provide a multiple-purpose style producing a **high quality** of images or providing a **large payload** for feasible applications, depending on the selection of the parameters n and M.
- ***(2)*** the algorithm **conveys a number of secret bits closer to the theoretical maximum** contributed from the triplet and notational concealment schemes we introduce.
- ***(3)*** it is possible to **accurately predict** the overall **capacity** and the stego **image quality** using mathematical expressions without conducting a real message embedding.
- ***(4)*** a coefficient mapping technique we recommend **increases the security and resists steganalytic attacks** from visualizing the histogram of stego images.