Multilevel Structured Approach for Digital Watermarking: Multimedia Data Protection Survey and Perspectives.

Sergey Bezzateev

Saint-Petersburg State University of Aerospace Instrumentation
ITMO University
Saint-Petersburg, Russia
svbezzateev@itmo.ru

This research was funded by the State Assignment grant FSER -2025-003

Agenda

1. Weighted Hamming metric
2. Generalized (L, G) codes in the weighted Hamming metric
3. Applications in information security

Weighted Hamming Metric (WHM)

Definition

The distance between vectors $\boldsymbol{a} = (a_1 a_2 \ldots a_n)$ and $\boldsymbol{b} = (b_1 b_2 \ldots b_n)$ in the WHM is defined by a function $d_{wH}(\boldsymbol{a}, \boldsymbol{b})$ that is given as

$$d_{wH}(\boldsymbol{a}, \boldsymbol{b}) = \sum_{i=1}^{n} w_i \cdot d(a_i, b_i), \quad \text{where} \quad w_i > 0,$$

$d(a_i, b_i) = 1$, if $a_i \neq b_i$ and $d(a_i, b_i) = 0$, if $a_i = b_i$.
The value w_i and vector $\boldsymbol{w} = (w_1, w_2, \ldots, w_n)$ are called as a position i weight and a vector of weights of positions, respectively.

Definition

A set $\boldsymbol{v} = \{v_1, v_2, \ldots, v_l\}$, $l \leq n$ consisting of all the different values of vector $\boldsymbol{w}$ components is called as a set of weights of the weighted Hamming metric.

Generalized binary (L,G) codes agreed with WHM

The set of all binary vectors
$\boldsymbol{a} = (a_1^{(1)}\ a_2^{(1)} \ldots a_{n_1}^{(1)}\ a_1^{(2)} \ldots a_1^{(l)} \ldots a_{n_l}^{(l)})$ of length
$n = n_1 + n_2 + \ldots + n_l$ satisfying relation (1) is called a generalized binary $\Gamma(L, G)$ code agreed with the WHM with set of weights $v = \{r_1, r_2, \ldots, r_l\}$.

$$\sum_{j=1}^{l} \sum_{i=1}^{n_j} a_i^{(j)} \frac{u_i^{(j)'}(x)}{u_i^{(j)}(x)} \equiv 0 \ mod\ G(x)\ , \qquad (1)$$

where $L = \bigcup_{j=1}^{l} L^{(j)},\ L^{(j)} = \left\{ \frac{u_i^{(j)'}(x)}{u_i^{(j)}(x)} \right\}, n_j = \left| L^{(j)} \right|$
$\deg u_i^{(j)}(x) = r_j \leq \deg G(x)\ , (u_i^{(j)}(x), G(x)) = 1$
for all $j = 1, \ldots, l;\ i = 1, \ldots, n_j$,
$u_i^{(j)}(x)$ are irreducible polynomials from $F_{2^m}[x]$ and
$G(x)$ is a separable polynomial from $F_{2^m}[x]$.

Bounds for partial lengths

The maximum power of the subset $L^{(j)}$ for the binary generalized $\Gamma(L, G)$ code is determined by number of irreducible polynomials of degree r_j from $F_{2^m}[x]$.

Theorem

The number of normalized irreducible polynomials from $F_{2^m}[x]$ of the degree l is determined by the value

$$I_{2^m}(l) = \frac{1}{l} \sum_{d|l} \mu(d) 2^{m\frac{l}{d}} \ ,$$

where $\mu(d)$ is the Möbius function:

$$\mu(d) = \begin{cases} 1 \ \ \text{if } d = 1 \ ; \\ (-1)^r \text{if } d \text{ is a product of } r \text{ different prime numbers} \ ; \\ 0 \ \ \text{in all other cases.} \end{cases}$$

Parameters of the generalized binary (L,G) codes agreed with WHM

The length of the generalized binary $\Gamma(L, G)$ code is equal to n :

$$n = n_1 + n_2 + \ldots + n_l \ ,$$

where $n_1 \leq 2^m,\ n_2 \leq l_{2^m}(2)\ , \ldots ,\ n_l \leq l_{2^m}(l)$. Obviously, the weight of the s-th code position w_s in WHM is determined by the degree r_j of the polynomial $u_i^{(j)}(x)$ used for numbering of this position: $w_s = r_j$

Dimension	$k \geq n - m \deg G(x)$,
Minimal Hamming distance	$d \geq \lceil \frac{2 \deg G(x)+1}{r} \rceil$, $r = \max\limits_{j=1,l} r_j$
Minimal weighted Hamming distance	$d_{\omega H} \geq 2 \deg G(x) + 1$

Code example

$G(x) \in F_{2^3}[x]$, $\deg G(x) = 6$, $n = 204$, $k = 186$, $d = 5$, $d_{\omega H} = 13$,

$n_1 = 8$	$n_2 = 28$	$n_3 = 168$	$n = 204$
$r_1 = 1$	$r_2 = 2$	$r_3 = 3$	
t_1	t_2	t_3	$t = t_1 + t_2 + t_3$
0	0	≤ 2	≤ 2
≤ 1	≤ 1	≤ 1	≤ 3
≤ 3	0	≤ 1	≤ 4
0	≤ 3	0	≤ 3
≤ 2	≤ 2	0	≤ 4
≤ 4	≤ 1	0	≤ 5
≤ 6	0	0	≤ 6

Best known codes in ordinary Hamming metric $(204, 188, 5)$, $(204, 186, 6)$ and $(204, 180, 7)$.

Hamming bound for WHM

The Hamming bound in the WHM can be written in the following way:

$$M \cdot W_n^{\tau} \leq 2^n,$$

where M is a number of codewords, and $\tau = \left\lfloor \frac{d_{WH}-1}{2} \right\rfloor$.

Definition

A binary block code with the length

$$n = n_1 + n_2 + \ldots + n_l$$

and a set of weights $\boldsymbol{v} = (v_1, v_2, \ldots, v_l)$ is called perfect in the WHM if its parameters are on the Hamming bound:

$$M \cdot W_n^{\tau} = 2^n \ .$$

Theorem about perfect code in WHM

Theorem

Binary generalized $\Gamma(L, G)$ *codes with the Goppa polynomial* $G(x) \in F_{2^m}[x],\ \deg G(x) = \tau$ *and the set L:*

$$L = \left\{ \frac{u_i^{(j)\prime}(x)}{u_i^{(j)}(x)} \right\}_{i=1,\ldots,n_j; j=1,\ldots,\tau}$$

are the perfect codes in the WHM :

$$2^k \cdot W_n^\tau = 2^n,$$

where $u_i^{(j)}(x)$ *are irreducible polynomials in* $F_{2^m}[x]$, $\deg u_i^{(j)}(x) = r_j \leq \tau$ *and* $G(x) \neq u_i^{(\tau)}(x)$ *for all i,*

$$n = \sum_{j=1}^{\tau} n_j\,, \quad k = n - m\tau, \quad d_{\omega H} = 2\tau + 1.$$

Number of correctable errors

Corollary

Binary generalized $\Gamma(L, G)$ code perfect in the WHM has the minimum distance $d_{\omega H} = 2\tau + 1$. It can correct any $t \leq \sum_{j=1}^{\tau} t_j$ errors, where $\sum_{j=1}^{\tau} j t_j \leq \tau$ and t_j is the number of errors on the n_j positions of the codeword.

Binary generalized $\Gamma(L, G)$ code with

$$n = 9,\ k = 5,\ d_{wH} = 5$$

$n_1 = 4$	$n_2 = 5$	$n = 9$
$v_1 = 1$	$v_2 = 2$	
t_1	t_2	$t = t_1 + t_2$
0	≤ 1	≤ 1
≤ 2	0	≤ 2

Best known linear code in ordinary Hamming metric is

$$(9, 5, 3)$$

Code example

Example 2

Binary generalized $\Gamma(L, G)$ code with

$n = 135,\ k = 127,\ d_{wH} = 5$

$n_1 = 16$	$n_2 = 119$	$n = 135$
$v_1 = 1$	$v_2 = 2$	
t_1	t_2	$t = t_1 + t_2$
0	≤ 1	≤ 1
≤ 2	0	≤ 2

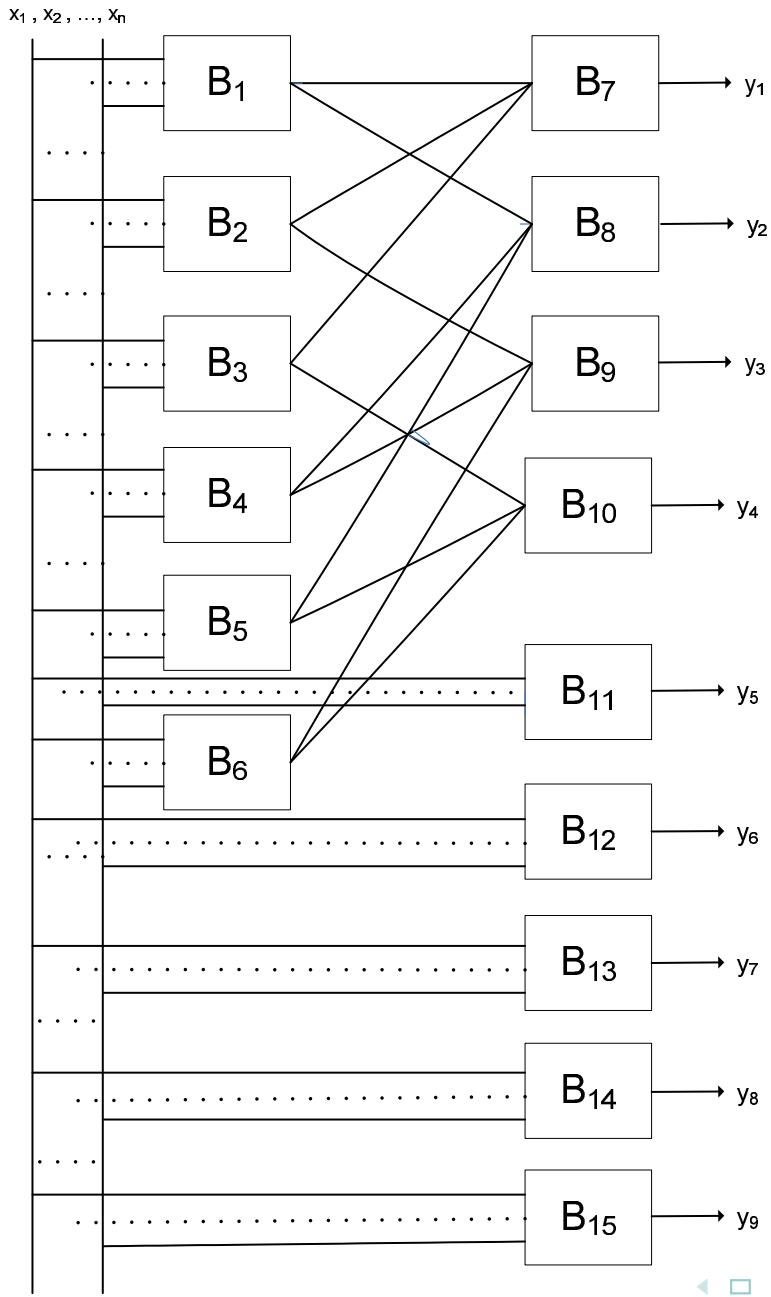
Best known linear code in ordinary Hamming metric is

$(135, 127, 3)$

Logical net application

Let us show how it is possible to use the binary generalized $\Gamma(L, G)$ codes in the WHM to correct the errors from a certain set.
We consider a logical net similar to that given in [Deza, M., Karpovsky, M.G., Milman, V.S.: Codes Correcting Arbitrary Sets of Errors. Revue de Cethedec, 66 (1981)].
There are 15 units $B_1, B_2, \ldots, B_{15}$ and 9 outputs $y_1, y_2, \ldots, y_9$. Let us take that only one unit of the net can function erroneously at every time moment.

Logical net


x1 , x2 , …, xn
B1
B2
B3
B4
B5
B6
B7
B8
B9
B10
B11
B12
B13
B14
B15
y1
y2
y3
y4
y5
y6
y7
y8
y9

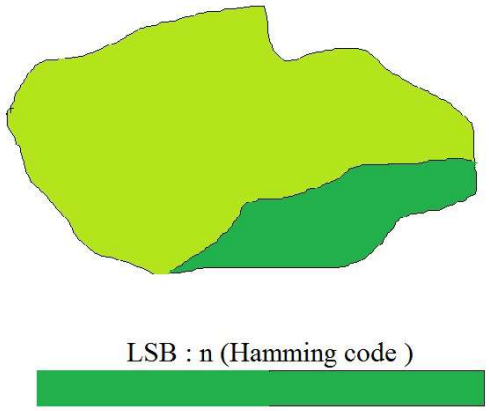
The Set of possible error vectors

$$
\begin{aligned}
F = & \{e_1, e_2, \ldots, e_{15}\} \\
= & \{(110000000), (101000000), (100100000), (011000000), \\
& (010100000), (001100000), (100000000), (010000000), \\
& (001000000), (000100000), (000010000), (000001000), \\
& (000000100), (000000010), (000000001)\} .
\end{aligned}
$$

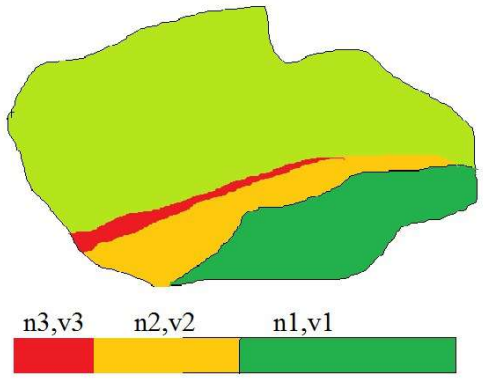
Binary generalized $\Gamma(L, G)$ code with
$n = 9,\ k = 5,\ d_{wH} = 5$

Steganography application

F5 : Two areas with different levels of significance



WF5 : Several areas with different levels of significance



Steganography application

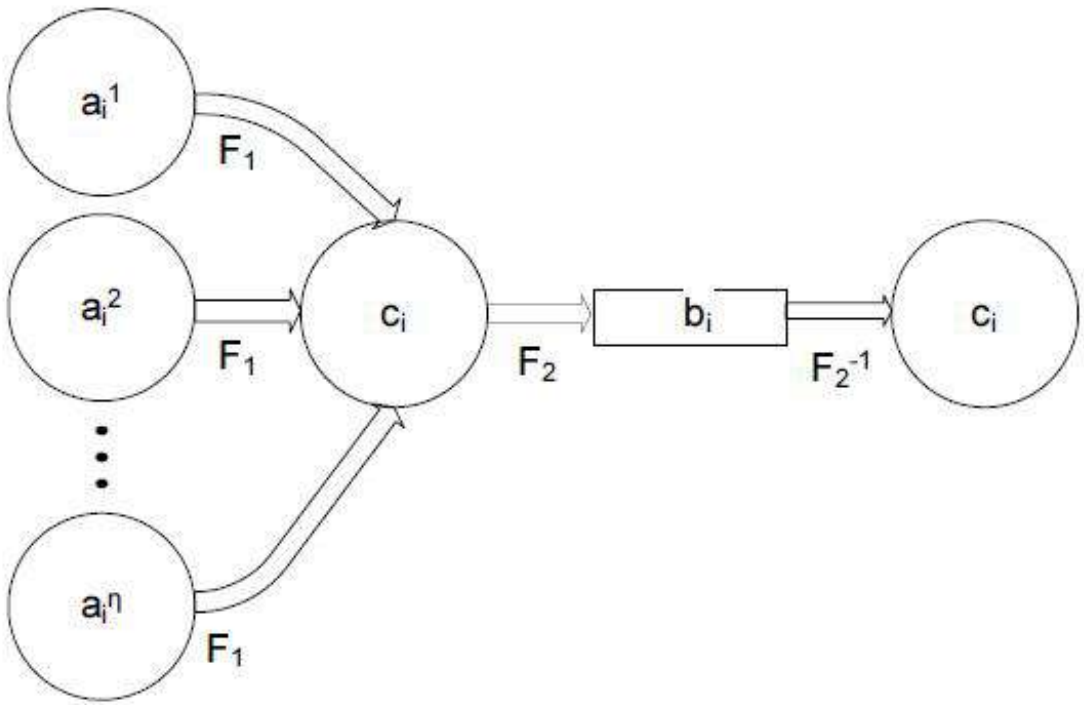
Embedding information into the original container

- determination of the weighted container structure for the original image
- choosing a suitable error correction code **C**
- syndrome1 = embedded information
- source container block —> syndrome2 (by using parity check matrix of the code **C**)
- syndrome1 + syndrome2 = syndrome
- syndrome –> error vector = distortions set (by using decoding algorithm for the code **C**)
- source container block + error vector = output container block

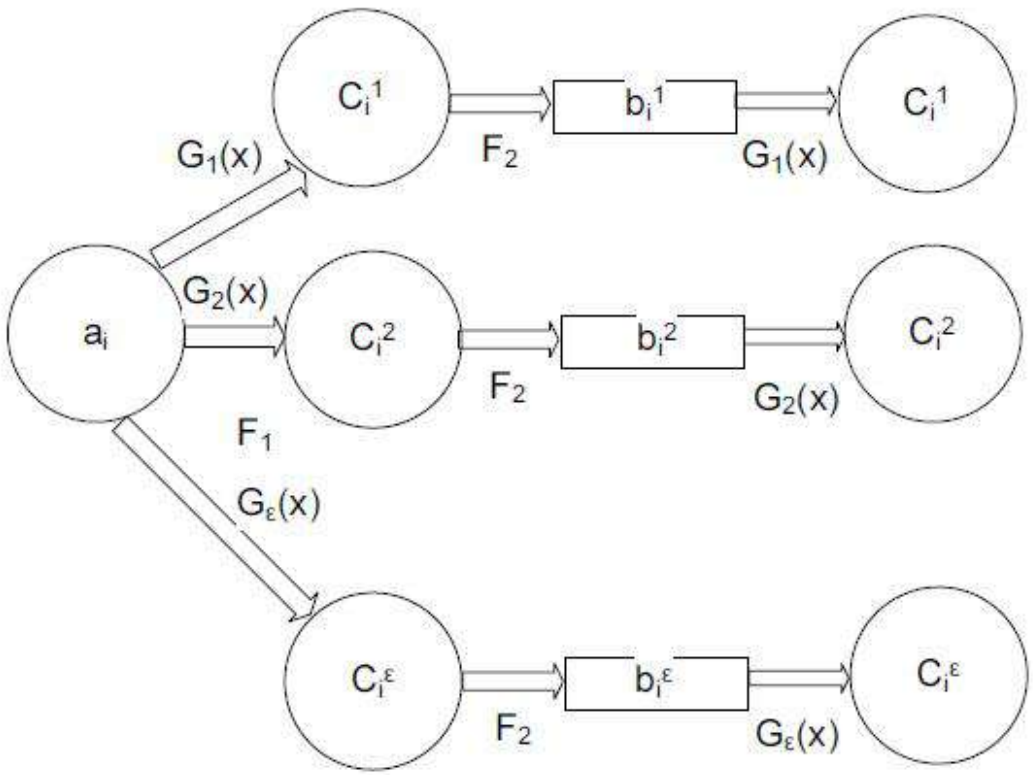
Extracting information from the output container

- output container block —> syndrome1(by using parity check matrix of the code **C**)

Code compression application



Code compression application



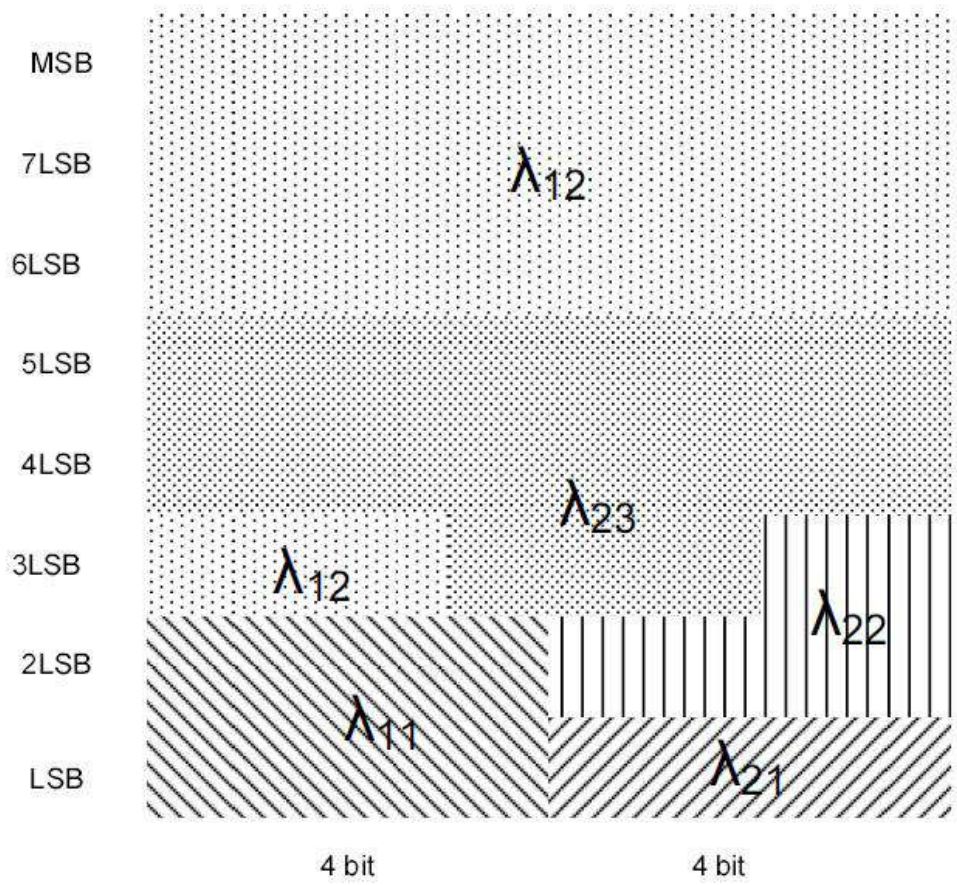
Toy example with two codes

Let us consider as an example the simplest case in which we take two generalized (L,G) codes for a block of the original image , consisting of $m = 8$ pixels, each of which is represented by $\ell = 8$ bits. Thus we have block of $K = 64$ bits. To compress this block we will use two different generalized (L,G) codes C_1 and C_2 with the following parametersdeg $G_1(x) = 2$, $n_1 = 35, k_1 = 29$ and $\lambda_{11} = 8, \lambda_{12} = 27$ (where λ_{11} is the part of the block of the source image having less significance than the part of the block corresponding to λ_{12}) and deg $G_2(x) = 3$ $n_2 = 35, k_2 = 29$ and $\lambda_{21} = 4, \lambda_{22} = 6, \lambda_{23} = 19$ (where λ_{21} is the part of the block of the source image having less significance than the part of the block corresponding to λ_{22} and less than λ_{23}). $G_1(x), G_2(x)$ are irreducible polynomials with coefficients from $GF(2^3)$ and $GF(2^2)$ respectively.

Toy example with two codes

Further, for the selected example, let us divide the block of the original image into three sub-blocks with lengths $N_1 = \lambda_{11} + \lambda_{21} = 12$,$N_2 = \lambda_{12} + \lambda_{22} = 33$, $N_3 = \lambda_{23} = 19$. The weights of the blocks, reflecting the influence of the occurring error on the resulting distortions, are defined as $v_1 = 1, v_2 = 2, v_3 = 3$ respectively. Thus, the greater the weight of a sub-block, the greater the influence of errors occurring in it on the resulting distortions in the image.

Variant of splitting the source image block



Possible distortions

$\ell_{678} = \{MSB, 7LSB, 6LSB\}, \ell_{45} = \{5LSB, 4LSB\}, l_3 = \{3LSB\},$
$\ell_2 = \{2LSB\}, \ell_1 = \{LSB\}$

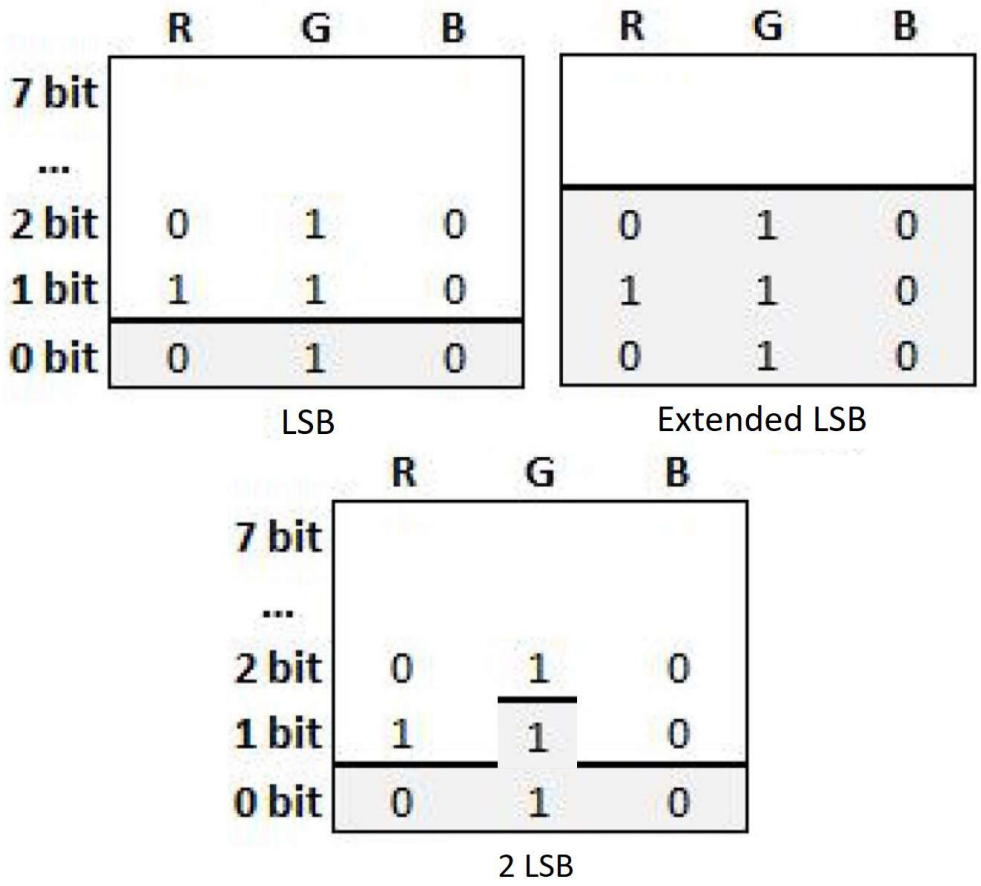
l_{678}	0	1	1	0	0	0	0	0	0	0	0	0	0	0
l_{45}	0	0	1	1	0	0	0	0	0	0	0	0	0	0
l_3	0	0	0	0	1	1	1	1	1	1	2	1	0	0
l_2	0	0	0	0	0	1	1	0	0	0	0	2	1	1
l_1	0	0	0	0	0	0	1	1	2	3	0	0	0	1

l_{678}	0	0	0	0	0	0	0	0	0	0	0	0	0
l_{45}	0	0	0	0	0	0	0	0	0	0	0	0	0
l_3	0	0	0	0	0	0	0	0	0	0	0	0	0
l_2	1	1	1	2	2	2	3	3	0	0	0	0	0
l_1	2	3	4	1	2	3	0	1	1	2	3	4	5

Agenda: Multimedia data unequal zones structures for DWM embedding

1. LSB approach
2. MLSB approach: structure based and semantic based
3. ECC code based MLSB weighted DWM embedding

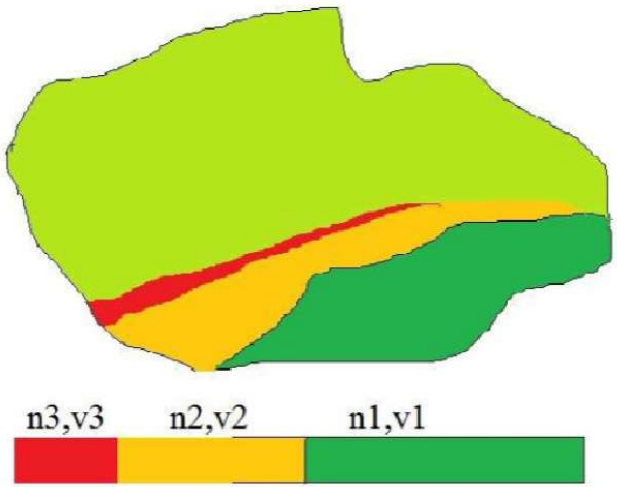
Least significant bit (LSB) embedding container model



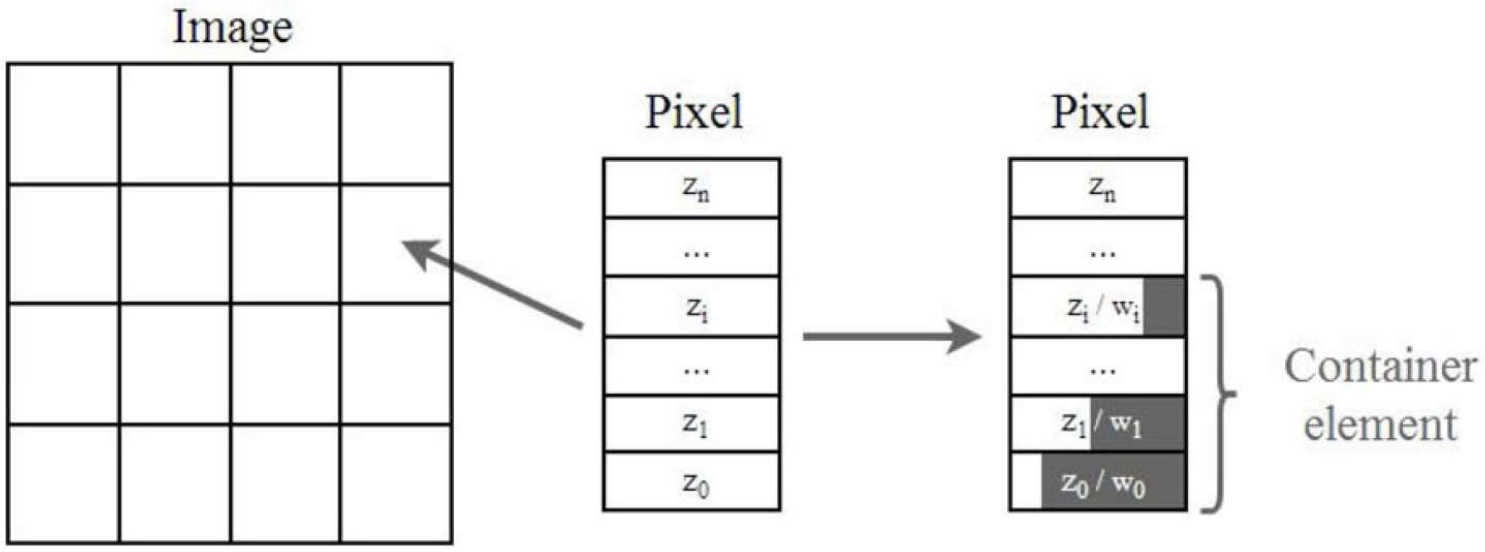
LSB approach

Multilevel significant bit (MLSB) embedding

1. Structure based MLSB image model
2. Semantic based MLSB image model
3. MLSB image models schema



Structure based MLSB image weighted model



Pixel zones example

Semantic based MLSB image weighted model



Most significant zone - w3

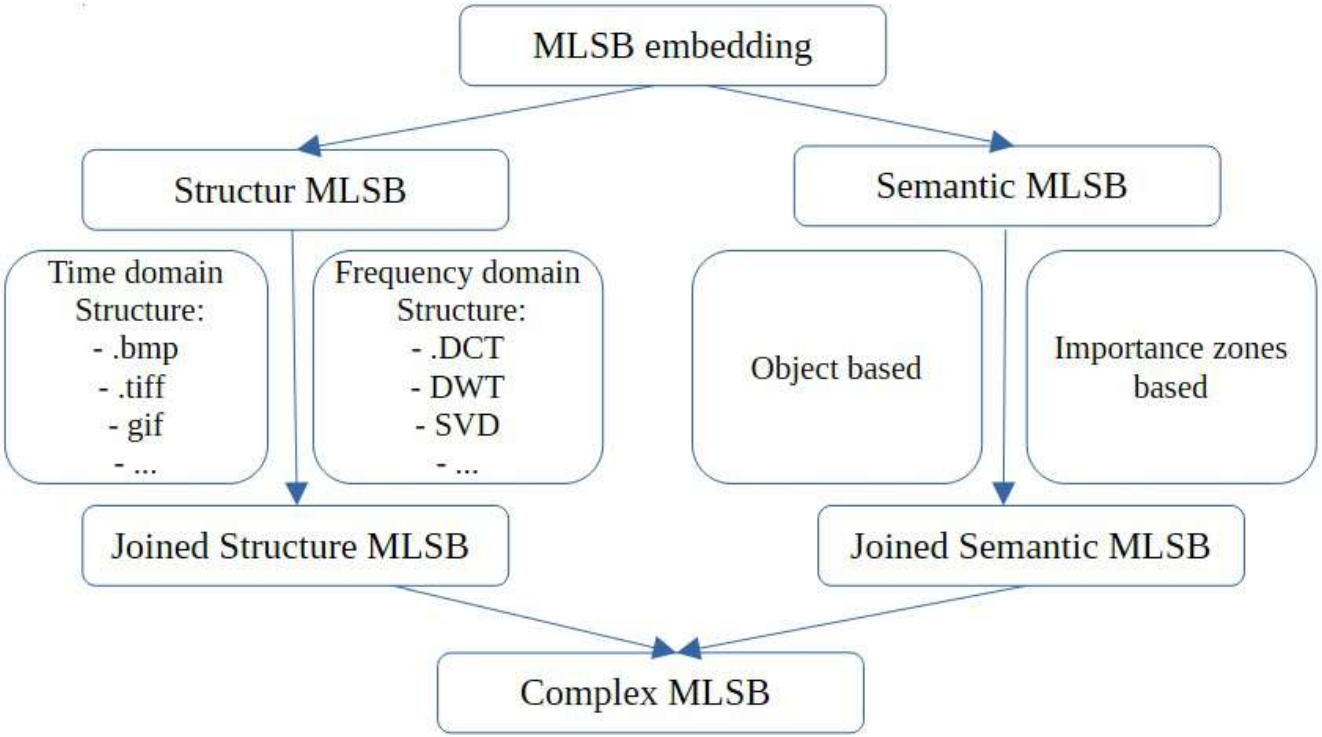
Middle significant zone - w2

Least significant zone - w1

$w1 < w2 < w3$

Object semantic zones example

MLSB image weighted models schema



MLSB image models schema.jpg

ECC code based MLSB weighted DWM embedding

$$P = \sum_{i=1}^{\ell} t_i p_i,$$

t_i- average number of distortions in zone I_i

p_i - significance (penalty) of the zone I_i

ℓ- overall zones number

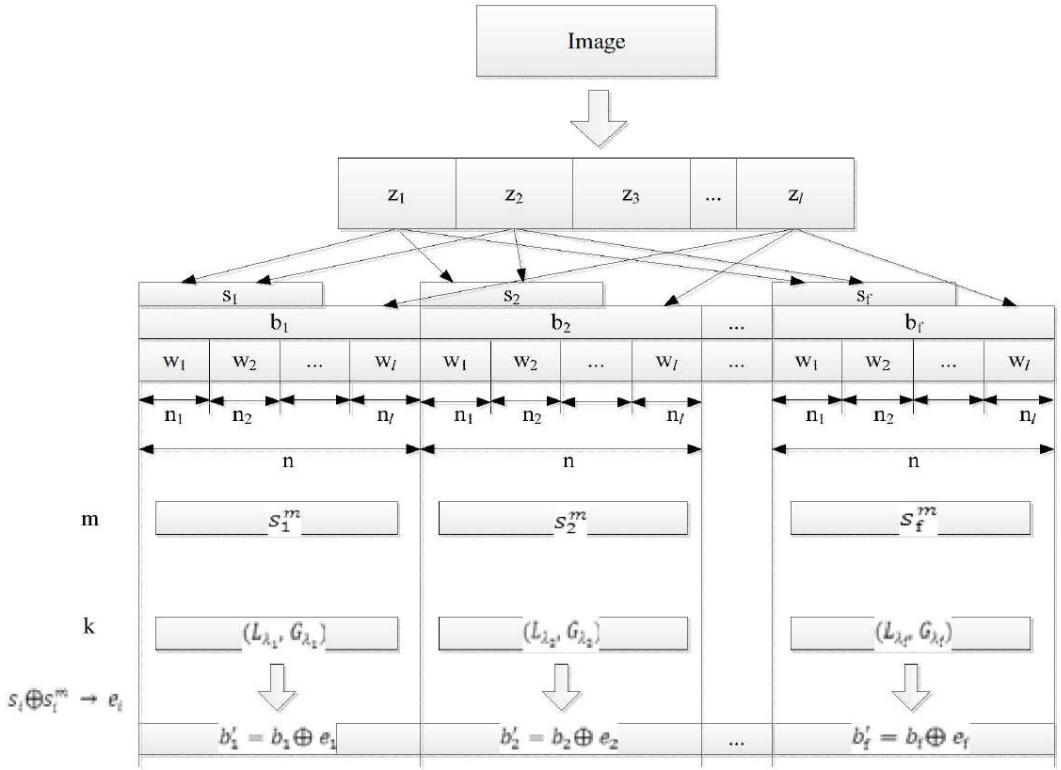
$$t = wt_H(\boldsymbol{\tau})$$

t - Hamming weight of the vector $\boldsymbol{\tau}$

$$\boldsymbol{\tau}^{(i)} = \boldsymbol{e}^{(i)} - \boldsymbol{c}^{(i)}$$

MLSB weighted DWM Penalty function

ECC code based MLSB weighted DWM embedding



MLSB weighted DWM embedding examples

MLSB weighted DWM embedding perspectives for AI technologies

MLSB weighted DWM embedding implementation for:

1. Multimedia data sources identification
2. Multimedia AI training process data sets/data bases protection
3. AI generated multimedia data watermarking and detection

Discussion

Thank You!

Questions ...