Reversible Watermarking Algorithm Using Sorting and Prediction

Tables

- **1. Reference**
  - Difference Expansion
- **2. Introduction**
- **3. Proposed algorithm**
  - a. Rationale of Prediction Using a Rhombus Pattern
  - b. Use of Histogram Shift Scheme
  - c. Use of Sorting
  - d. Overflow and Underflow Problem
  - e. Double Embedding Scheme
  - f. Appropriate Threshold Values
- **4. Encoder and Decoder**
- **5. Experiments**

1. Reference

- **Reversible Data Embedding Using a Difference Expansion - Jun Tian**
- which exploits the correlations between adjacent pixels to embed the data bits
- Each pair of pixels is able to embed one data bit, and the maximal embedding capacity can theoretically reach 0.5 bit per pixel (bpp)
- Reversible data hiding ≈ Reversible data embedding ≈ lossless data embedding ≈ Reversible Watermarking ≈ Lossless data hiding
- embeds invisible data (Payload)
- can be applied to digital audio and video as well
  - 1) Payload capacity limit
  - 2) Visual quality
  - 3) Complexity

$\boxed{\text{embed one bit } b = 1}$

$$l = \left\lfloor \frac{206 + 201}{2} \right\rfloor = \left\lfloor \frac{407}{2} \right\rfloor = 203, \; h = 206 - 201 = 5$$

$$h' = \underbrace{2 \times h}_{\text{shifting}} + b = 2 \times 5 \underbrace{+ 1}_{\text{embedding}} = 11.$$

$$x' = 203 + \left\lfloor \frac{11 + 1}{2} \right\rfloor = 209, \; y' = 203 - \left\lfloor \frac{11}{2} \right\rfloor = 198.$$

$\boxed{\text{extract the embedded bit } b}$

$$l' = \left\lfloor \frac{209 + 198}{2} \right\rfloor = 203, \; h' = 209 - 198 = 11.$$

$$b = \mathrm{LSB}(h') = 1, \; h = \left\lfloor \frac{h'}{2} \right\rfloor = 5.$$

$h' = 2 \times h + b$ is called the DE.

2. Introduction

- without using a location map in most cases.
- sorted prediction errors (if needed, though rarely, a reduced size location map allows us)
  - prediction errors to embed data
  - sorting technique is used to record the sorted prediction errors
- two important requirements (the embedding capacity, distortion)

- Tian's difference expansion [15]
  - his embedding capacity is at best 0.5 b/pixel (an uncompressed location map also needs 0.5 b/pixel)
  - The location map consists of flags, which are either 0 or 1
- Alattar [1, 2]
  - expanded one cell from a pair to a triplet [1] or quad [2] to hide two or three bits per cell
  - cell is the unit of pixels in which the data is to be embedded
  - location map covers all triplets or quads instead of pairs, location map size is 1/3, 1/4
- Kamstra and Heijmans [7]
  - reduce location map size by sorting pairs according to correlation measures to facilitate compression
  - its efficient utilization of the correlation between neighboring pixels in the image (highly correlate with average values of neighboring pairs)

- Location maps usually are huge in size and should be compressed

2. Introduction

- without using a location map in most cases.
- sorted prediction errors (if needed, though rarely, a reduced size location map allows us)
  - prediction errors to embed data
  - sorting technique is used to record the sorted prediction errors
- two important requirements (the embedding capacity, distortion)

- Lee at al [9]
  - expandability (which means a possibility of bit shifting operation)
  - changeability (high-frequency wavelet coefficients of each block)
  - least significant bit (LSB) replacement approach for hiding the location map
- Thodi and Rodriguez [13, 14]
  - proposed a histogram shift method for embedding data in prediction errors
  - The location map (or flag bits) used in these schemes covers all ambiguous cells
  - [14] advanced methods based on difference expansion technique
  - [14] is using the JPEG-LS prediction errors as an input signal for the embedding scheme (optimized for lossless compression)
  - The correlation of JPEG-LS prediction errors is higher than that of the difference values between neighbors

- prediction error histogram for correlation

3. Proposed Algorithm

A. Rationale of Prediction Using a Rhombus Pattern

- to exploit the correlation between the pixel and its prediction
- the accurate prediction leads to a better embedding performance
- rhombus predictor
- cross embedding scheme

$$u'_{i,j} = \left\lfloor \frac{v_{i,j-1} + v_{i+1,j} + v_{i,j+1} + v_{i-1,j}}{4} \right\rfloor. \quad (1)$$

predicted value $u'_{i,j}$

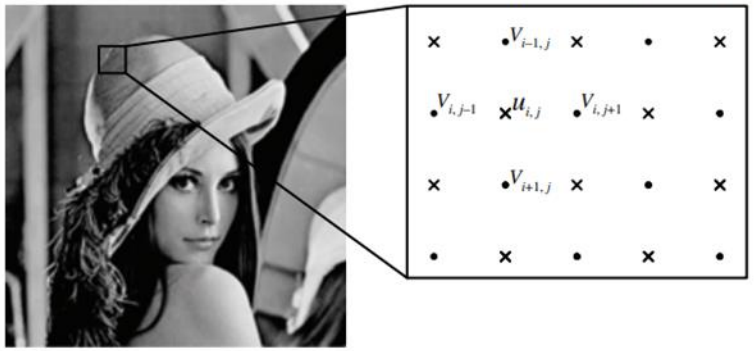


Fig. 1. Prediction pattern. The pixel value u of the Cross set can be predicted by using the four neighboring pixel values of the Dot set and expanded to hide one bit of data.

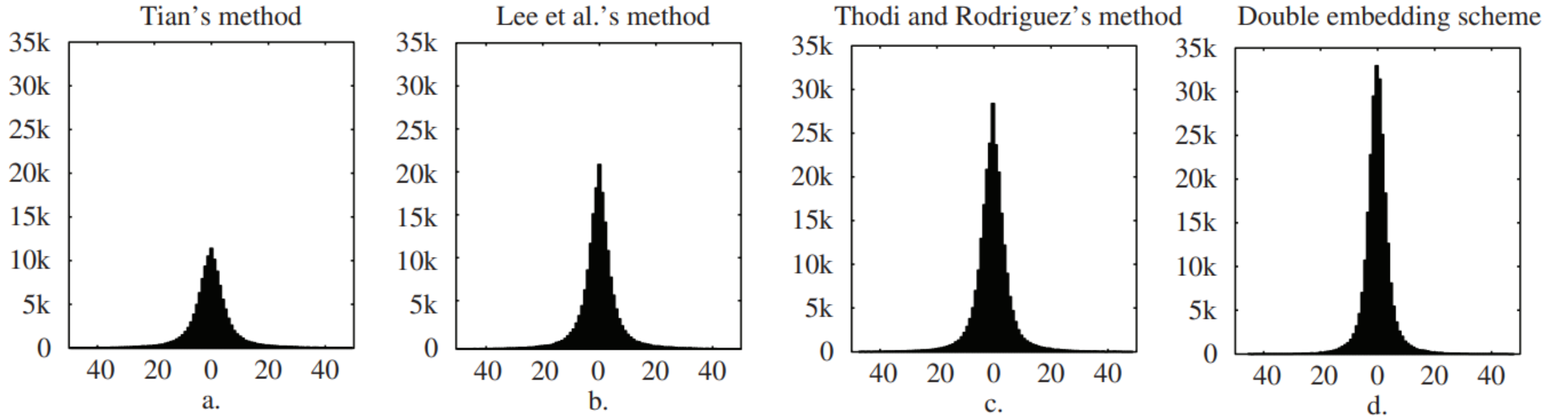


Fig. 2. (a) Histogram of differences between neighboring pixels. (b) Histogram of high-frequency wavelet coefficients. (c) Histogram of JPEG-LS prediction errors. (d) Histogram of prediction errors for Double embedding scheme and (e) for Lena image.

3. Proposed Algorithm

A. Rationale of Prediction Using a Rhombus Pattern

- to exploit the correlation between the pixel and its prediction
- the accurate prediction leads to a better embedding performance
- rhombus predictor
- cross embedding scheme

prediction error $d_{i,j}$

$$d_{i,j} = u_{i,j} - u'_{i,j}. \quad (2)$$

difference expansion

$$D_{i,j} = 2d_{i,j} + b \quad (3)$$

After data hiding

$$U_{i,j} = D_{i,j} + u'_{i,j}. \quad (4)$$

modified prediction error

$$D_{i,j} = U_{i,j} - u'_{i,j}. \quad (5)$$

embedded bit value

$$b = D_{i,j} \bmod 2. \quad (6)$$

original prediction error

$$d_{i,j} = \left\lfloor \frac{D_{i,j}}{2} \right\rfloor. \quad (7)$$

original pixel's value

$$u_{i,j} = u'_{i,j} + d_{i,j}. \quad (8)$$

3. Proposed Algorithm

B. Use of Histogram Shift Scheme

- The histogram shift method is an efficient reversible data hiding technique in terms of low distortion

two threshold values T_n and T_p are used

Expandable set E consists of the predicted errors in $[T_n; T_p]$

The set S is the set of predicted errors that can be shifted

The embedding capacity of the proposed scheme is $|E| - |L|$
where $|L|$ is the size of the location map

histogram shift encoding algorithm

$$D_{i,j} = \begin{cases} 2d_{i,j} + b, & \text{if } d_{i,j} \in [T_n; T_p] \\ d_{i,j} + T_p + 1, & \text{if } d_{i,j} > T_p \text{ and } T_p \geq 0 \\ d_{i,j} + T_n, & \text{if } d_{i,j} < T_n \text{ and } T_n < 0. \end{cases} \quad (9)$$

The decoder recovers

$$d_{i,j} = \begin{cases} \lfloor D_{i,j}/2 \rfloor, & \text{if } D_{i,j} \in [2T_n; 2T_p + 1] \\ D_{i,j} - T_p - 1, & \text{if } D_{i,j} > 2T_p + 1 \text{ and } T_p \geq 0 \\ D_{i,j} - T_n, & \text{if } D_{i,j} < 2T_n \text{ and } T_n < 0 \end{cases} \quad (10)$$

$$b = D_{i,j} \bmod 2, \quad D_{i,j} \in [2T_n; 2T_p + 1]. \quad (11)$$

3. Proposed Algorithm

C. Use of Sorting

- sorting is possible only when cells are independent (embedding data into one cell should not affect the other cells)
- Dot and Cross sets of the rhombus scheme are independent each other

$$\mu_{i,j} = \frac{1}{4}\sum_{k=1}^{4}(\Delta v_k - \Delta\bar{v}_k)^2 \qquad (12)$$

Local variance $\mu_{i,j}$

where $\Delta v_1 = |v_{i,j-1} - v_{i-1,j}|$, $\Delta v_2 = |v_{i-1,j} - v_{i,j+1}|$, $\Delta v_3 = |v_{i,j+1} - v_{i+1,j}|$, $\Delta v_4 = |v_{i+1,j} - v_{i,j-1}|$, $\Delta\bar{v}_k = (\Delta v_1 + \Delta v_2 + \Delta v_3 + \Delta v_4)/4$.

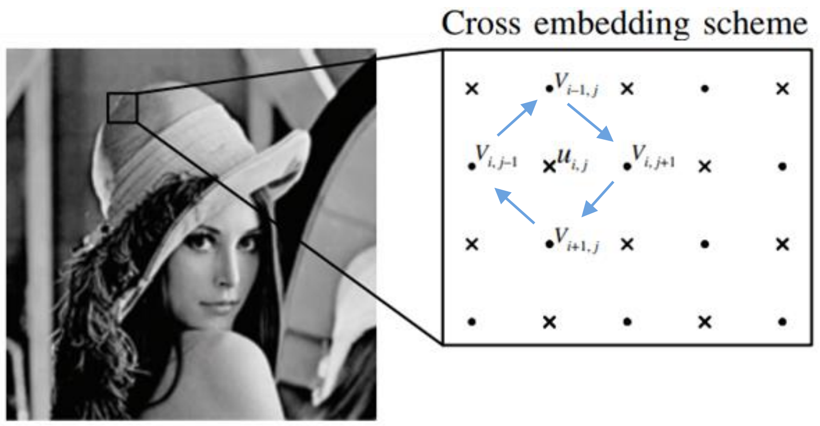


Fig. 1. Prediction pattern. The pixel value u of the Cross set can be predicted by using the four neighboring pixel values of the Dot set and expanded to hide one bit of data.

Assume that d_{sort} is the sorted row of all $d_{i,j}$

- The histogram shift method embeds data with the thresholds
- Cells are sorted in ascending order of the local variance values
- Cells with smaller variance values are better for data hiding

3. Proposed Algorithm

D. Overflow and Underflow Problem

- is used for locating such problematic cells

$$0 \le u'_{i,j} + D_{i,j} \le 255 \qquad (13)$$

S_p is the set of all problematic cells, and S_{op} is the set of overlapping cells with S_p after data hiding.

two-pass testing.

encoder testing (ET)

ET(a) If the current cell is modifiable twice, the corrective location map is not necessary.

ET(b) If the current cell is modifiable once owing to overflow/underflow errors during the second pass, this cell is marked as "0" in the corrective location map. The set of cells that results in this case are denoted as S_{op}. A cells in this category overlaps with the cells in set S_p after modification.

ET(c) If the current cell is not modifiable even once, the cell cannot be used in the embedding phase. This cell is marked as "1" in the location map. The set of such cells is denoted as S_p.

3. Proposed Algorithm

D. Overflow and Underflow Problem

threshold value be 1 (i.e., $T = 1$).

“1”

sample 1 with $u_{n+1,m} = 253$, $u'_{n+1,m} = 252$, and $d_{n+1,m} = 1$;

sample 2 with $u_{n-1,m} = 254$, $u'_{n-1,m} = 252$, and $d_{n-1,m} = 2$;

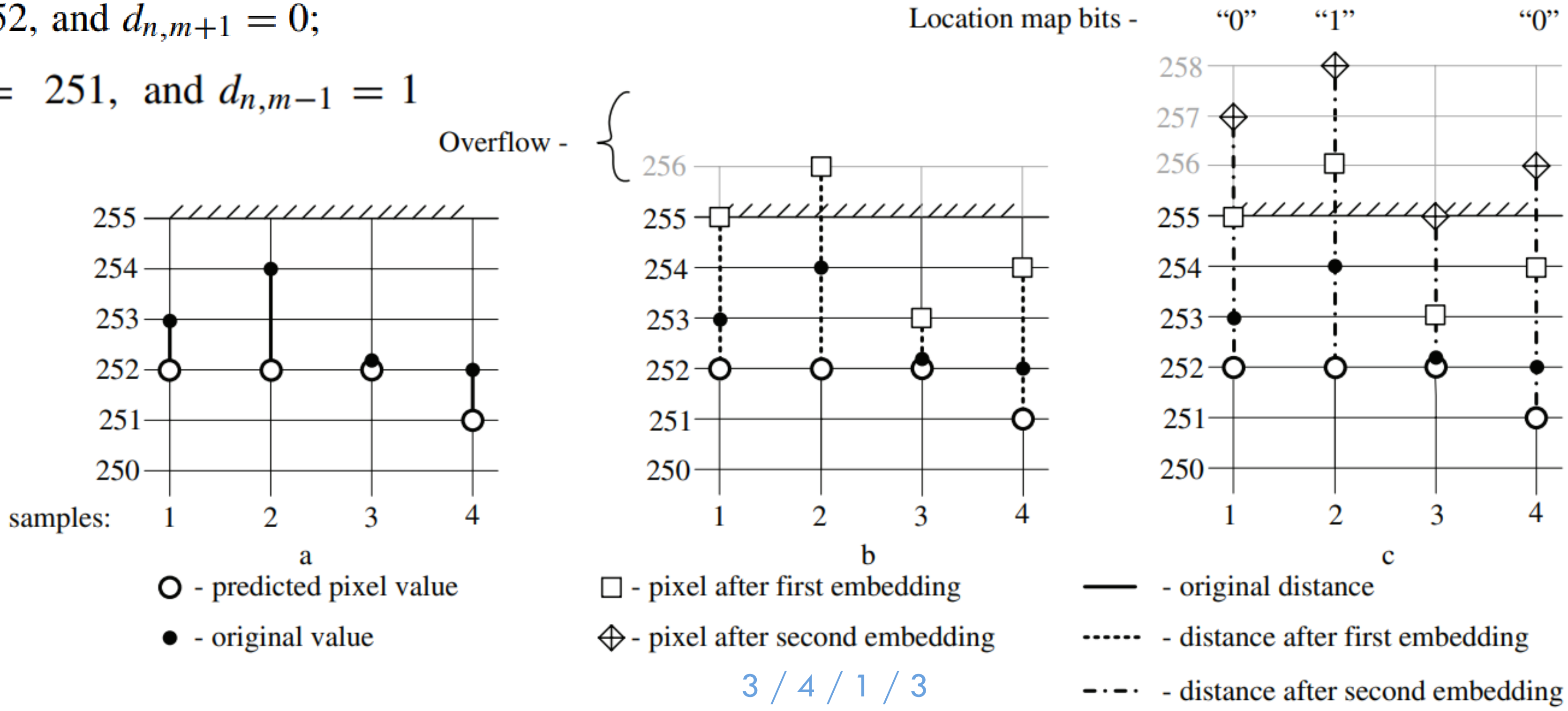
sample 3 with $u_{n,m+1} = 252$, $u'_{n,m+1} = 252$, and $d_{n,m+1} = 0$;

sample 4 with $u_{n,m-1} = 252$, $u'_{n,m-1} = 251$, and $d_{n,m-1} = 1$

After data hiding

$$U_{i,j} = D_{i,j} + u'_{i,j}. \quad (4)$$

$$D_{i,j} = \begin{cases} 2d_{i,j} + b, & \text{if } d_{i,j} \in [T_n; T_p] \\ d_{i,j} + T_p + 1, & \text{if } d_{i,j} > T_p \text{ and } T_p \geq 0 \\ d_{i,j} + T_n, & \text{if } d_{i,j} < T_n \text{ and } T_n < 0. \end{cases} \quad (9)$$



3 / 4 / 1 / 3

Fig. 3. Overflow testing: (a) original data, (b) first time embedding, and (c) second time embedding.

3. Proposed Algorithm

D. Overflow and Underflow Problem

DT(a) If the cell to be decoded is modifiable once, the corrective location map bit is not needed, because the cell was modifiable twice at the encoding phase and thus was not marked. Refer to the Case ET(a).

DT(b) If the cell to be decoded is not modifiable at all, the location map must be consulted, because the cell was either modifiable once or not modifiable at all at the encoding phase (i.e., $S_{op} \bigcup S_p$). If the corrective location map bit is "0," the current pixel belongs to the Case ET(b). The decoder has to extract original value of the cell. However, if the location map bit is "1," the cell is disregarded as it was not modifiable in the encoder. Refer to the Case ET(c).

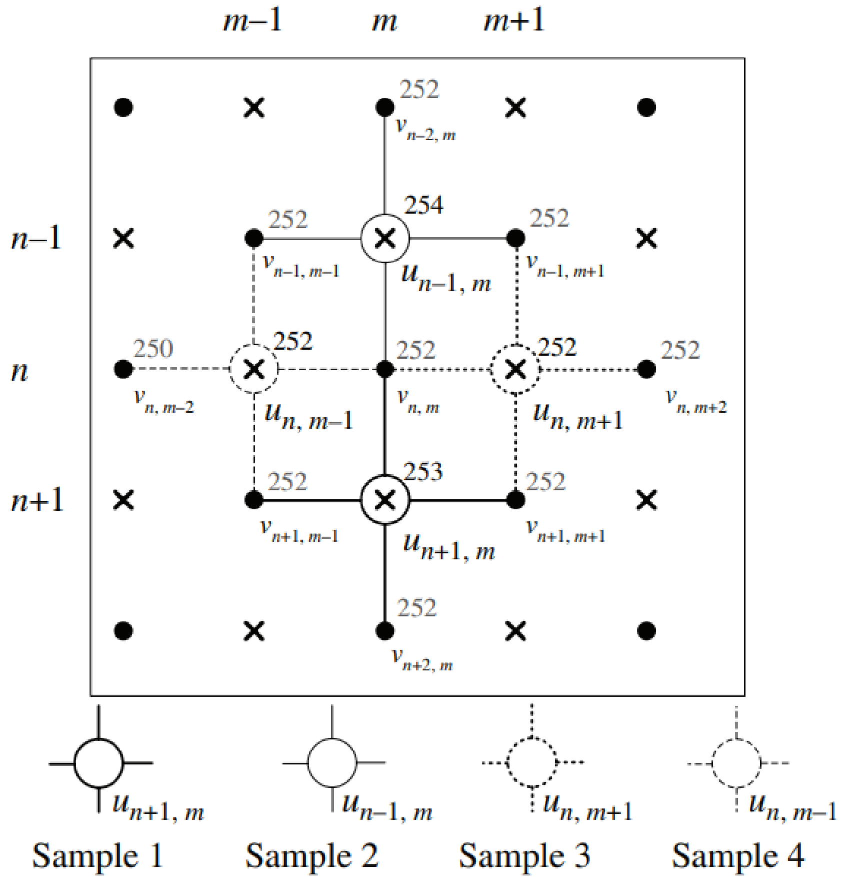


Fig. 4. Example of samples for overflow testing (associated with Fig. 3).

3. Proposed Algorithm

E. Double Embedding Scheme
F. Appropriate Threshold Values

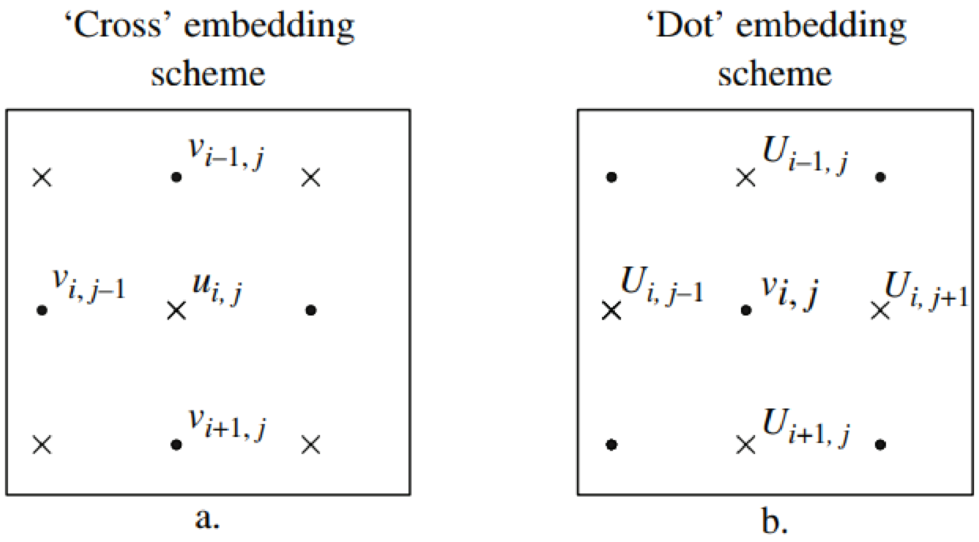


Fig. 5. Relationship among pixels in the Cross set or the Dot set.

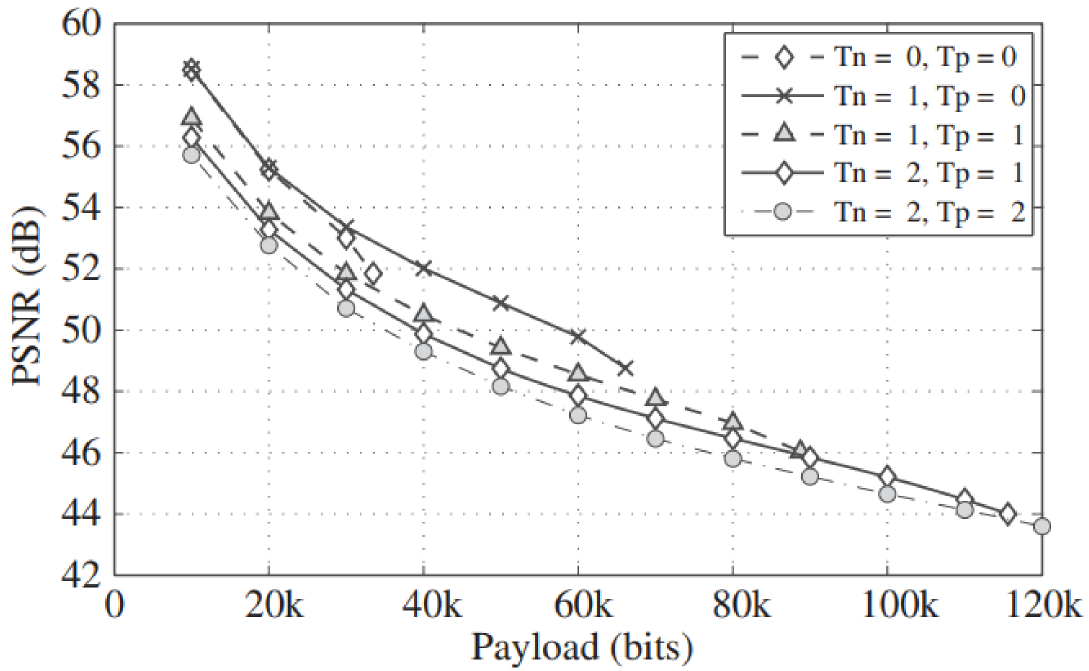


Fig. 6. Effect of threshold values on PSNR given a certain payload for the Lena image.

payload P should be divided into two sets with similar sizes P_{Cross} and P_{Dot} for the Cross and Dot embedding schemes, respectively.

4. Encoder and Decoder

The LSB values of the first 34 prediction errors from d_{sort} are replaced with threshold values $T_{n\text{Cross}}$ (7 bits) and $T_{p\text{Cross}}$ (7 bits), payload size $|P_{\text{Dot}}|$ (20 bits), or $|P_{\text{Cross}}|$ (20 bits). Original 34 LSB values should be collected to a set of collected LSB values S_{LSB}, and included to the payload.

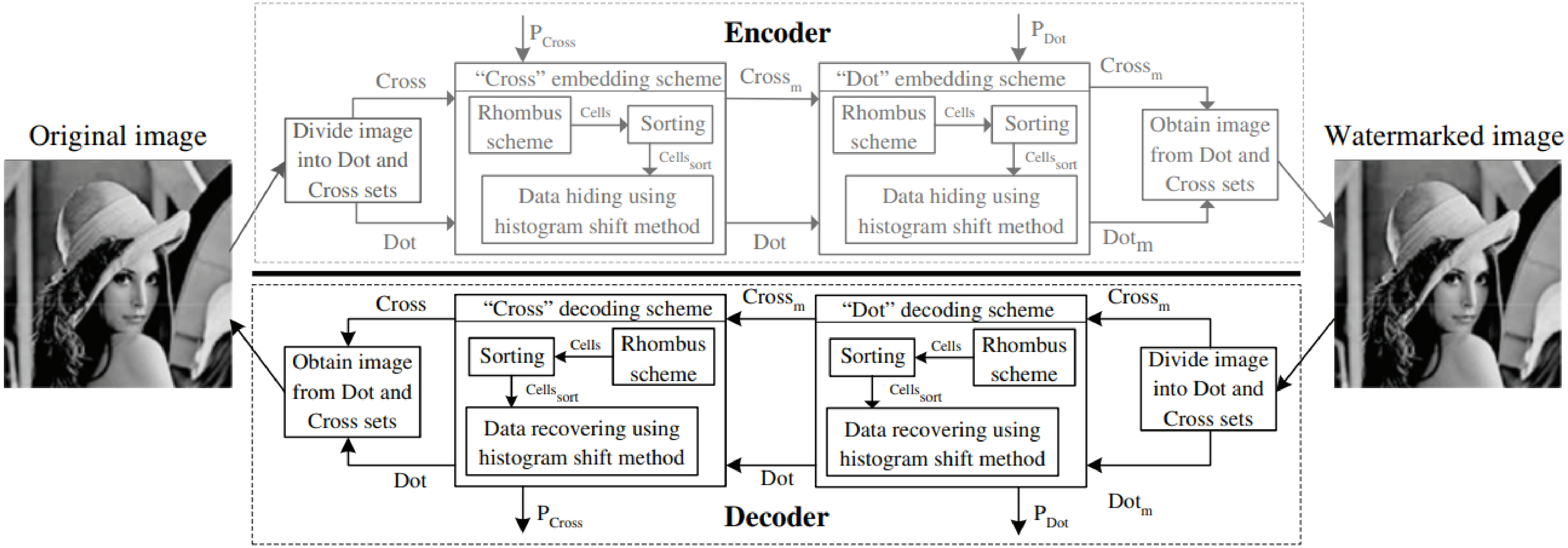
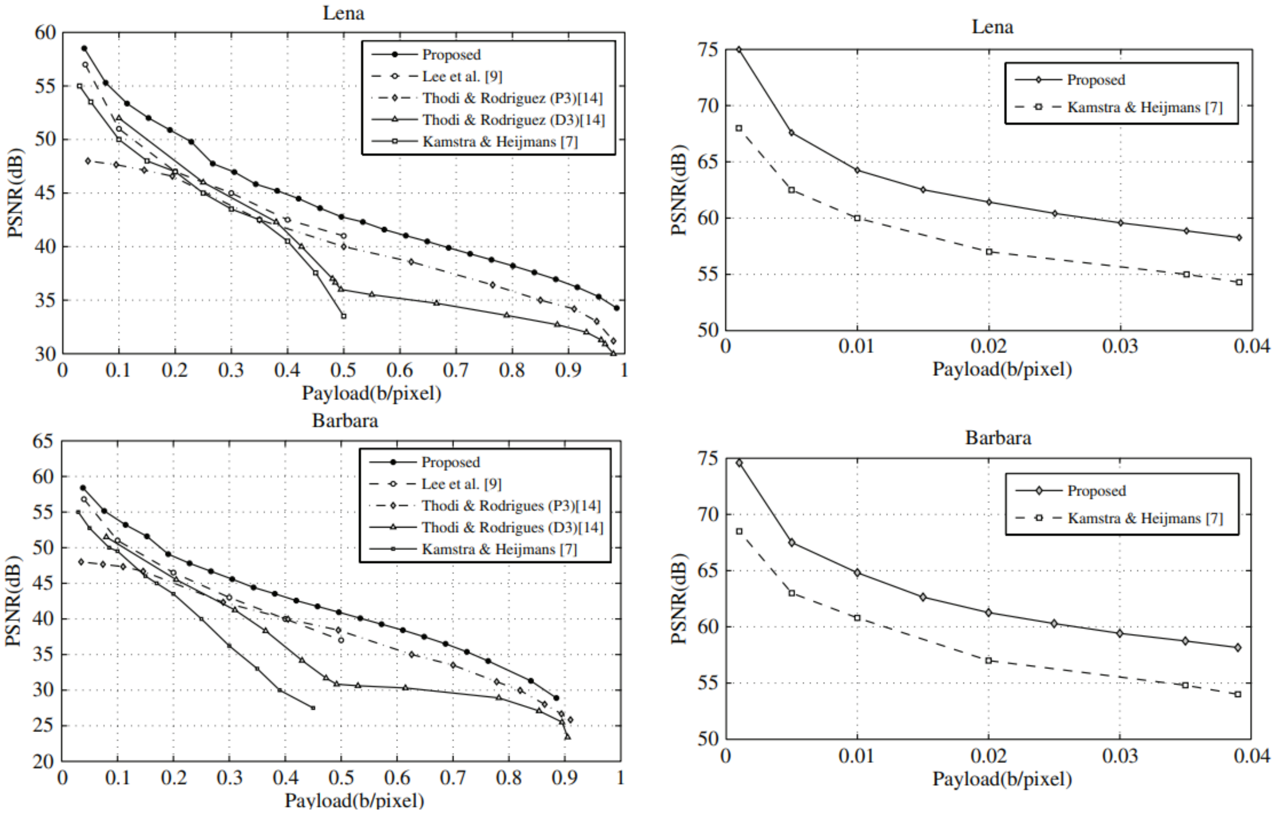


Fig. 7. Framework of the double encoding and decoding scheme.

5. Experiments



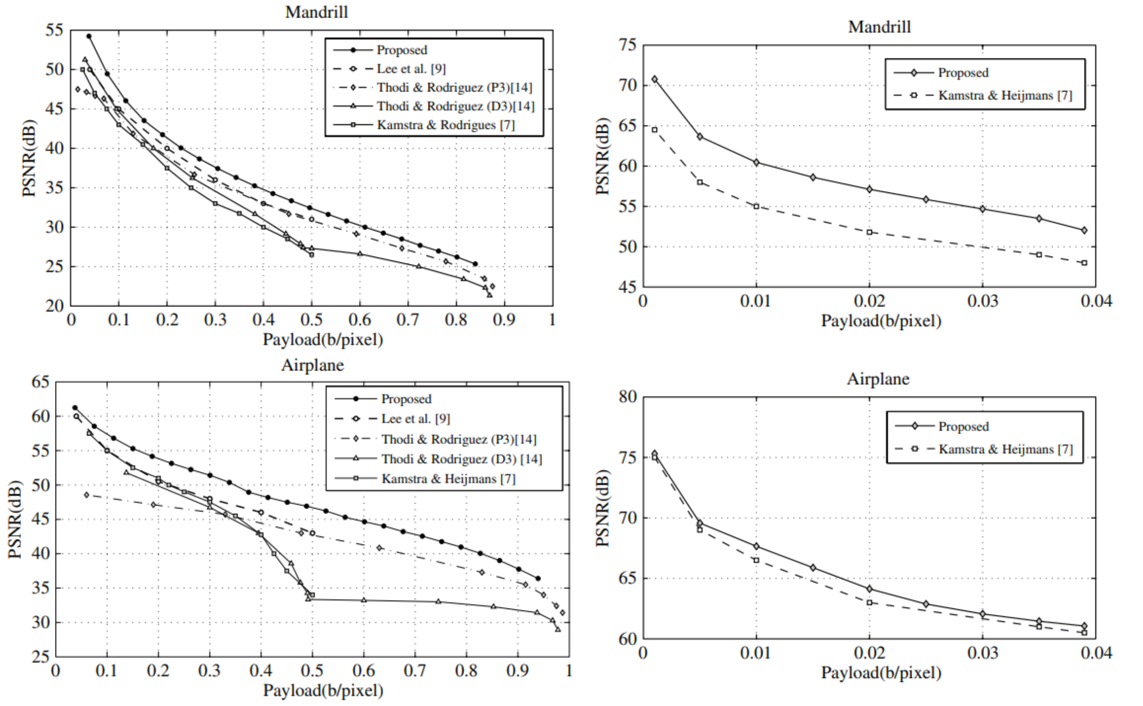


Fig. 8. Distortion vs. capacity graphs of the tested methods for each of the four images. Right column of graphs shows results for the small capacities (0–0.04 b/pixel).

5. Experiments

TABLE I

IMPROVEMENTS OVER EXISTING METHODS FOR LENA IMAGE IN DB

Payload (b/pixel)	Improvements over Kamstra and Heijmans [7]	Improvements over Thodi and Rodriguez [14] based on		Improvements over Lee *et al.* [9]
		Difference expansion	Prediction errors	
0.040	3.97 dB	—	10.05 dB	1.51 dB
0.100	3.36 dB	2.05 dB	6.25 dB	3.02 dB
0.300	3.46 dB	2.21 dB	3.22 dB	1.96 dB
0.450	6.04 dB	3.98 dB	2.06 dB	1.85 dB
0.650	—	5.67 dB	2.47 dB	—
0.800	—	2.63 dB	2.19 dB	—

TABLE II

LOCATION MAP SIZE VS. PAYLOAD FOR TESTED IMAGES

Payload (bits/b/pixel)	Lena			Barbara			Mandrill			Airplane		
	Thresholds		Location map (bits)	Thresholds		Location map (bits)	Thresholds		Location map (bits)	Thresholds		Location map (bits)
	T_n	T_p		T_n	T_p		T_n	T_p		T_n	T_p	
100 k/0.38	−2	1	0	−2	2	0	−6	5	31	−1	1	0
150 k/0.57	−3	3	0	−6	5	2	−11	10	278	−2	2	0
200 k/0.76	−5	5	0	−9	8	115	−20	19	1758	−4	3	0
220 k/0.83	−6	6	2	−15	15	1253	−29	29	6782	−5	5	0
230 k/0.87	−8	7	6	−25	25	5602	—	—	—	−7	6	0
250 k/0.95	−12	12	9	—	—	—	—	—	—	−13	13	13