Reversible Image Watermarking Based on Full Context Prediction

Ming Chen, Zhenyong Chen, Xiao Zeng, and Zhang Xiong

School of Computer Science and Engineering, Beihang University

{mingchen@cse.}{chzhyong@}{zengxiao29@cse.}{xiongz@}buaa.edu.cn

Introduction

Reversible Image Watermarking is a kind of watermarking that reversibly embeds covert information into digital images. Its reversibility allows the host images be exactly recovered once embedded data is extracted. It has applications in scenarios where exact host images are crucial, such as medical and military imaging. The work in this paper aims at providing large embedding capacity while high image quality is demanded. The proposed scheme has close relationship to

- Difference Expansion [1]
- Histogram Modification [2]
- Prediction-error Expansion [3]

Methods

Embed data into prediction-errors using additive expansion.

Embedding: $$e' = \begin{cases} e + sign(e) \times b, & |e| = ME \\ e + sign(e) \times 1, & |e| \in [ME+1, LE) \end{cases}$$

Recovering: $$e = \begin{cases} e' - sign(e') \times b, & |e'| \in [ME, ME+1] \\ e' - sign(e') \times 1, & |e'| \in (ME+1, LE] \end{cases}$$

Extracting: $$b = \begin{cases} 0, & |e'| = ME \\ 1, & |e'| = ME+1 \end{cases}$$

Full context prediction

Since image correlation exists in all directions, prediction can be more precise by making full use of correlated pixels in all directions.

Watermarking cannot change pixel value dramatically for concerns of image fidelity, so, watermarked pixel is still highly correlated to its neighboring pixels.

Divide the original image into four interlaced subimages, watermark all subimages separately in sequence. For every single subimage, use pixels in other three subimages for prediction. Process the subimages in the reversed order during extracting and recovering.

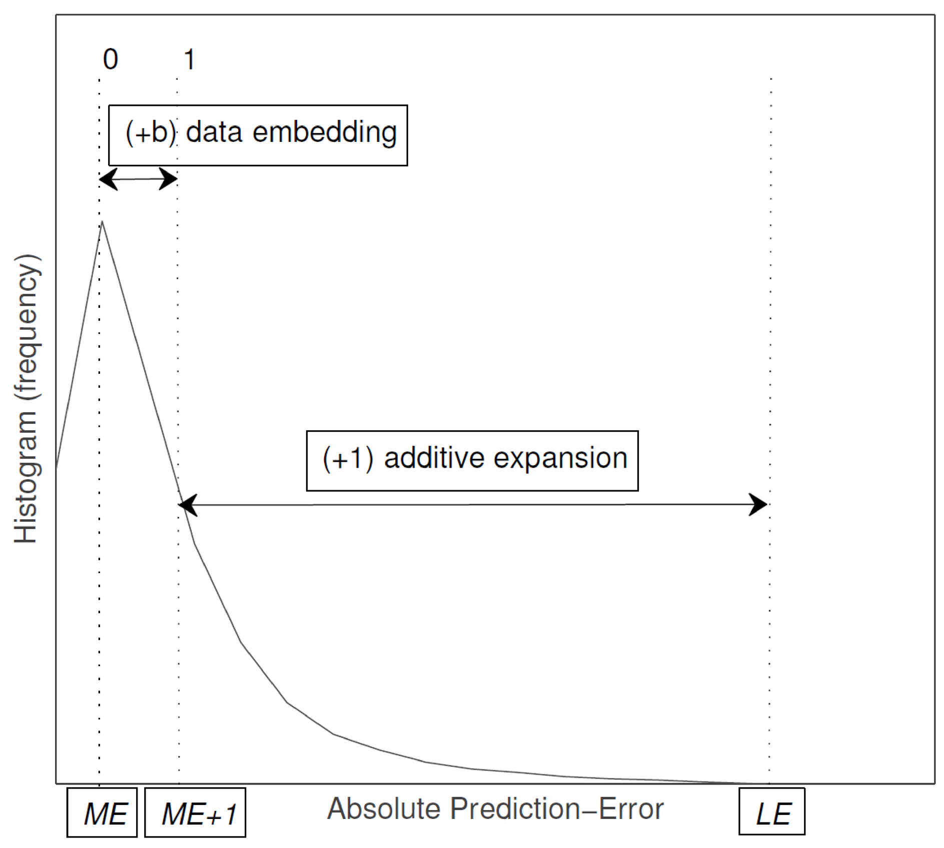

Figure 1: Additive Expansion.

ME and LE mark the peak and valley of the histogram of absolute prediction-error.

Subimages: $$\begin{aligned} U_1 &= \{x(2i,2j) \mid (i,j) \in [1,H'] \times [1,W']\} \\ U_2 &= \{x(2i,2j+1) \mid (i,j) \in [1,H'] \times [1,W']\} \\ U_3 &= \{x(2i+1,2j) \mid (i,j) \in [1,H'] \times [1,W']\} \\ U_4 &= \{x(2i+1,2j+1) \mid (i,j) \in [1,H'] \times [1,W']\} \end{aligned}$$

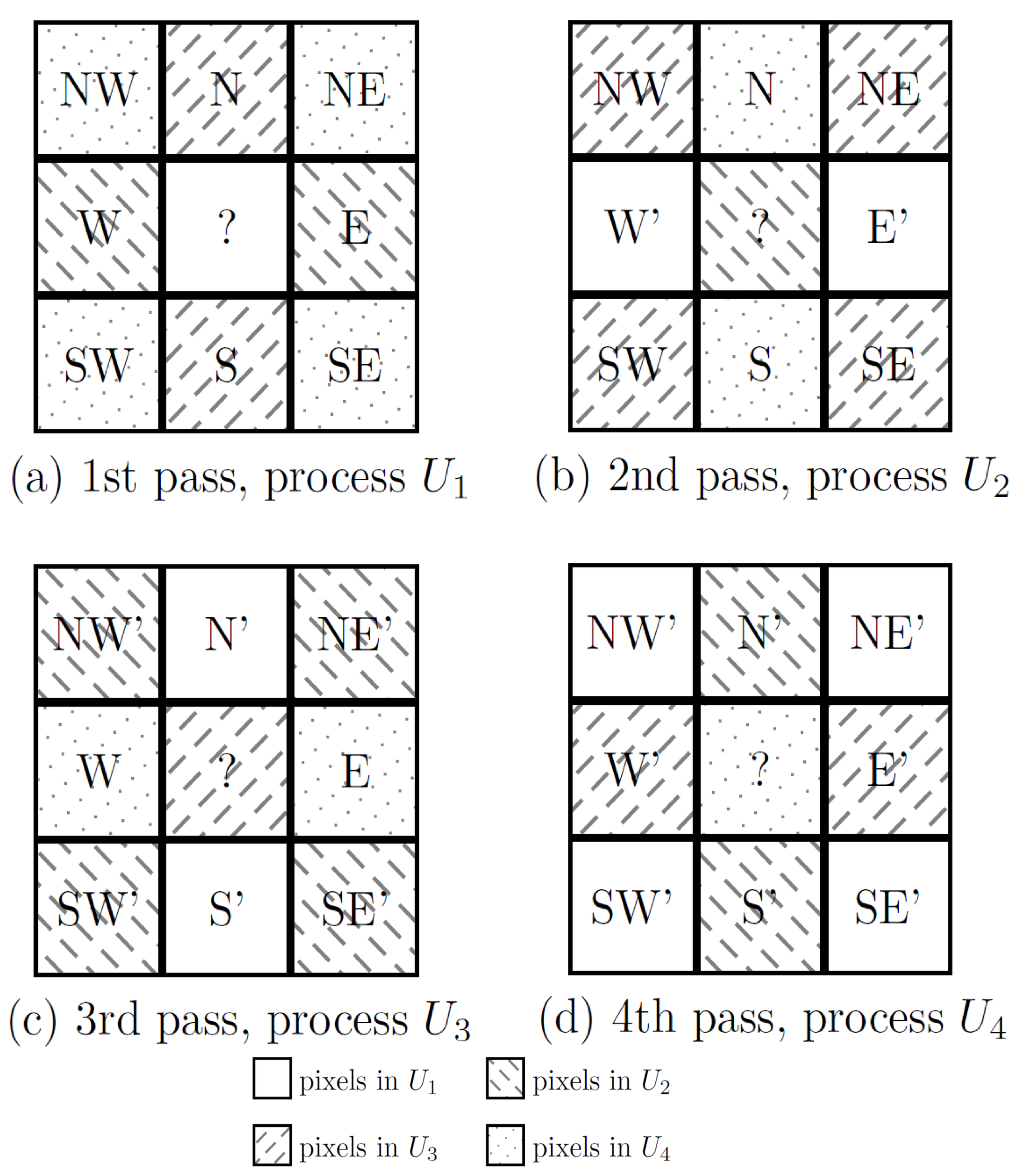

Figure 2: Full Context Prediction.

Pixels marked by '?' are current pixels to be predicted. Watermarked pixels are apostrophized.

Without loss of generality, we process the subimages in the sequence of U_1, U_2, U_3, and U_4, namely we watermark U_2 after U_1 is watermarked and so forth. For a pixel in every single subimage, its full context consists of the eight nearest pixels, which are within the other three subimages. Let U_k be the k-th subimage to be watermarked and U'_k be the watermarked result of U_k, then the full context of U_k contains pixels in $\{U'_1 \ldots U'_{k-1} U_{k+1} \ldots U_4\}$. In the above case, the full contexts of the four subimages U_1, U_2, U_3, and U_4 are formed by pixels in $\{U_2 U_3 U_4\}$, $\{U'_1 U_3 U_4\}$, $\{U'_1 U'_2 U_4\}$ and $\{U'_1 U'_2 U'_3\}$.

The simple full context predictor used in this paper is

$$\hat{x} = \frac{x_w + x_s + x_e + x_n}{4}$$

Procedures

Embedding procedures:

1. Predict the value of every pixel using a full context. Calculate prediction-errors and find out ME and LE.
2. Apply additive expansion to all pixels equal to 1 or 254, embed part of the watermark message $\mathcal{W}$, and record the indices of pixels whose values become 0 or 255 to form overhead bitstream $\mathcal{O}$.
3. Concatenate $\mathcal{O}$ and the residual watermark message $\mathcal{W}'$ to form a new bitstream $\mathcal{N}$. Apply additive expansion to all pixels lie between 2 and 253 and embed $\mathcal{N}$.

Extracting and recovering procedures:

1. Apply the inverse additive expansion to all non-boundary pixels. Put extracted bits into bitstream $\mathcal{L}_1$ if the restored pixels lie between 2 and 253, or else put the bits into bitstream $\mathcal{L}_2$. Once overhead information is available in $\mathcal{L}_1$, decode it and remove it from $\mathcal{L}_1$.
2. Identify those pseudo-boundary pixels using decoded overhead information. Apply the inverse additive expansion to them and put extracted bits into bitstream $\mathcal{L}_3$.
3. Reassemble $\mathcal{L}_2$ and $\mathcal{L}_3$ to form bitstream $\mathcal{L}_4$ by ordering the indices of their corresponding pixels. Concatenate $\mathcal{L}_4$ and $\mathcal{L}_1$ to obtain the watermark data.

Results

Table 1: Performance of The Proposed Scheme

Image (8-bit grayscale)	Capacity (bits)					PSNR (db)
	U_1	U_2	U_3	U_4	Pure	
Lena	18,341	17,333	17,904	17,097	70,627	49.6
Baboon	6,342	6,213	6,102	6,025	24,634	48.6
Plane	19,658	19,658	19,577	19,453	78,154	50.1
Boat	9,740	9,474	9,822	9,440	38,316	48.9
Tiffany	12,202	11,609	11,734	11,519	46,056	49.1
Barbara	12,765	11,877	12,622	11,797	48,997	49.1
Peppers	10,588	9,980	10,040	9,269	39,733	48.9
Goldhill	11,482	11,478	11,470	11,248	45,630	49.0
Average	**12,640**	**12,203**	**12,409**	**11,981**	**49,018**	**49.2**

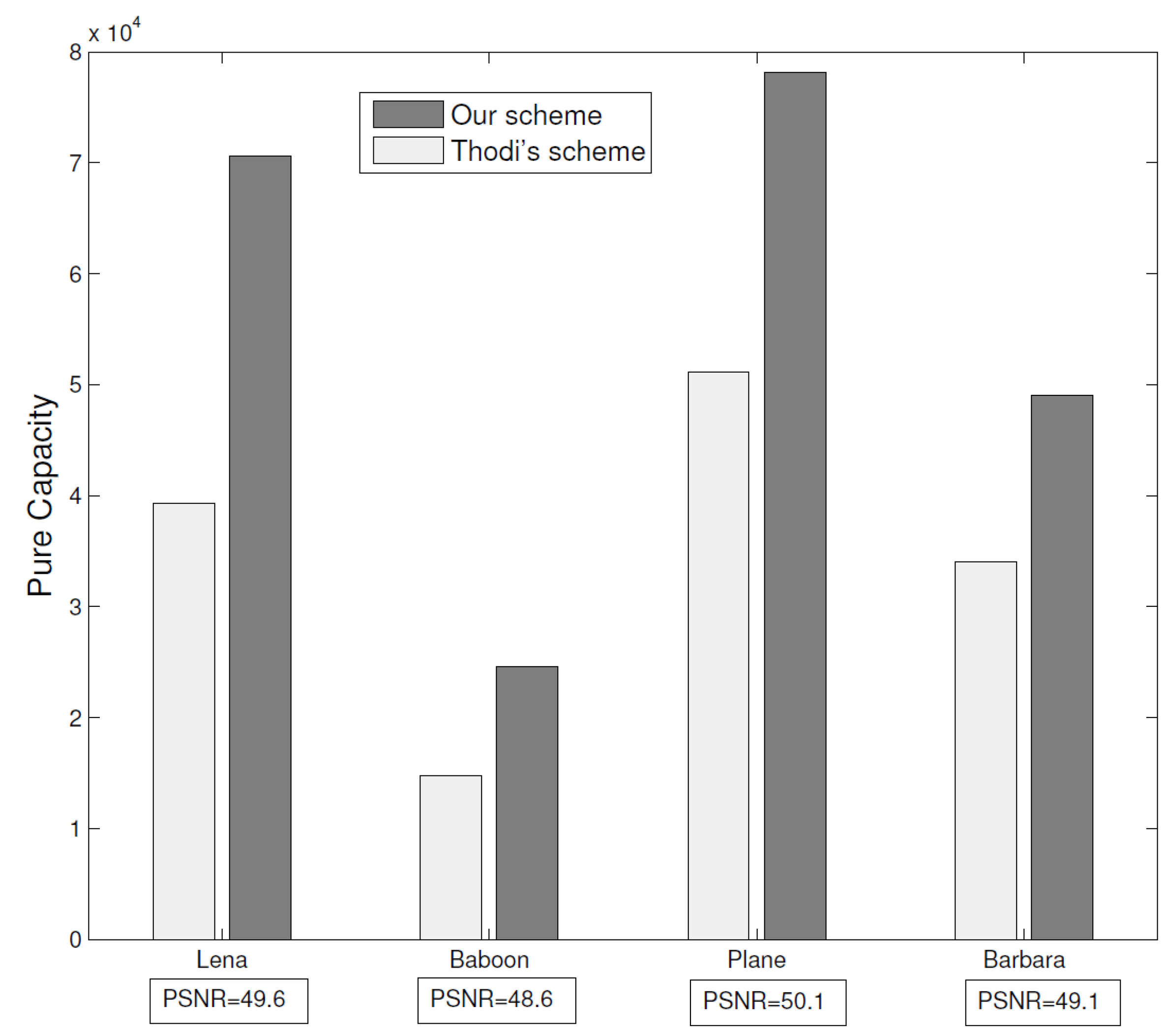

Figure 3: Performance Comparison.

(a) Original Lena

(b) Watermarked Lena (49.6db)

Figure 4: Image Quality.

References

[1] J. Tian, "Reversible data embedding using a difference expansion," *IEEE Trans. Circuits and Systems for Video Technology, vol. 13, no. 8, pp. 890–896, 2003.*

[2] Z. Ni, Y.-Q. Shi, N. Ansari, and W. Su, "Reversible data hiding," *IEEE Trans. Circuits and Systems for Video Technology, vol. 16, no. 3, pp.* 354–362, 2006.

[3] D. M. Thodi and J. J. Rodriguez, "Expansion embedding techniques for reversible watermarking," *IEEE Trans. Image Processing, vol. 16, no. 3, pp. 721–730, 2007.*

[4] D. Coltuc, "Improved capacity reversible watermarking," in *IEEE International Conference on Image Processing ICIP 2007, pp. 249–252.*

[5] M. Chen, Z. Chen, X. Zeng, and Z. Xiong, "Reversible data hiding using additive prediction-error expansion," in *ACM Workshop on Multimedia and Security 09, Princeton, New Jersey, 2009, pp. 19–24.*