

Research Article

Hybrid Predictor and Field-Biased Context Pixel Selection Based on PPVO

Hongyin Xiang, Jinsha Yuan, and Sizu Hou

Institute of Electrical and Electronic Engineering, North China Electric Power University, Baoding 071000, China

Correspondence should be addressed to Hongyin Xiang; leo_xianghy@126.com

Received 26 November 2015; Revised 29 January 2016; Accepted 1 February 2016

Academic Editor: Daniel Zaldivar

Copyright © 2016 Hongyin Xiang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Most pixel-value-ordering (PVO) predictors generated prediction-errors including -1 and 1 in a block-by-block manner. Pixel-based PVO (PPVO) method provided a novel pixel scan strategy in a pixel-by-pixel way. Prediction-error bin 0 is expanded for embedding with the help of equalizing context pixels for prediction. In this paper, a PPVO-based hybrid predictor (HPPVO) is proposed as an extension. HPPVO predicts pixel in both positive and negative orientations. Assisted by expansion bins selection technique, this hybrid predictor presents an optimized prediction-error expansion strategy including bin 0 . Furthermore, a novel field-biased context pixel selection is already developed, with which detailed correlations of around pixels are better exploited more than equalizing scheme merely. Experiment results show that the proposed HPPVO improves embedding capacity and enhances marked image fidelity. It also outperforms some other state-of-the-art methods of reversible data hiding, especially for moderate and large payloads.

1. Introduction

Digital watermarking has been used for copyright protection [1, 2], integrity check [3, 4], and self-identity [5]. Important and secret data as watermark was embedded into the host image. However, in some sensitive scenarios, such as medical images, remote sensing images, and military images, even a small amount of deviation is highly detrimental. Reversible data hiding has improved to this demand [6, 7]. Reversibility here means that pixels in marked image can be perfectly recovered to the original ones in the host image.

Three categories of solutions can be summarized to this problem, difference expansion (DE) [8–10], histogram shifting (HS) [11–14], and integer transformation [15–19]. Difference expansion computes and expands the differences between pixels rather than the pixel itself to embed data. This method has considerable capacity and little distortion. However, difference relies on locations on the map occupying spaces large enough for the payload. Histogram shifting uses the peak point bin to carry data and shifts other bins to guarantee reversibility. The peak bin contains the most common pixels. Residual images composed of the difference of adjacent pixels or various neighboring pixels can also be

used to modify the histogram [20]. In this way, gray values are modified at most by 1 . In this way, this method achieves high fidelity. However, it can only be used to hide limited amounts of information. To remedy the above shortages, sorting [21, 22] and prediction-error expansion (PEE) [23] were incorporated into existing methods. Pixel-value-ordering (PVO) in [24] connected these two new stages together. PVO-K in [25] used equivalent multiextreme pixels overlooked by PVO to carry data bits with identical value. Improved PVO (IPVO) in [26] strengthened integrated behavior using a new predictor different from PVO. Notice that all the three aforementioned methods hunt embedding pixels within artificial separated nonoverlapped image blocks. Pixel-based pixel-value-ordering (PPVO) in [27] predictors provided a novel pixel search strategy in a pixel-by-pixel manner. Proximate pixels were selected to form a context pixel vector and predict objective pixel. Experimental results demonstrated markedly higher image quality and larger embedding capacity (EC) than PVO, PVO-K, and IPVO. Nevertheless, only prediction-error 0 was expanded and only some of the context pixel selection strategies have been brought into use. PPVO takes more practical value as an outstanding pixel hunting mode. In this paper, a hybrid predictor is proposed including errors

0 and 1 expanded rather than only error 0 in the previous work of PPVO. Meanwhile, a novel field-biased context pixel selection scheme is here incorporated in for more accurate prediction than equalizing type merely. Furthermore, prediction-error expansion bins selection (EBS) develops an effective mechanism for optimized performance. Experiment results show that the proposed HPPVO may provide higher embedding capacity and better marked image fidelity than conventional PPVO.

The rest of this paper is organized as follows. In Section 2, some related works are briefly described and the primary motivation for improving PPVO is also stated. In Section 3, the concrete scheme of the proposed HPPVO is analyzed in detail. Section 4 lists extensive experiments compared to state-of-the-art studies. Section 5 concludes this paper.

2. Related Work

2.1. Prediction-Error Expansion. Thodi and Rodriguez [23] first proposed PEE method, which predicted pixels and embedded data employing the produced errors instead of pixel differences. This new technique exploited the potential of the correlation inherent in neighborhood pixels. Thus, EC of PEE doubles that of DE-based method. What is more, complex predictors can make much smaller prediction-error for higher fidelity. Afterwards, Li et al. [28] combined adaptive embedding and pixel selection for significant improvement. They tried to adaptively embed data into expanded pixels in the light of different complexity. This technique guaranteed only those small prediction-errors to be applied for expansion and embedding. Experimental results demonstrated that it was an effective way and outperformed some other state-of-the-art methods.

Optimal expansion bins selection was described in some previous works. Wang et al. [29] selected prediction-error expansion in a dynamical way. Embedding distortion was precalculated for all possible embedding pixels and determined by minimal distortion. Hwang et al. [30] upgraded histogram shifting by sorting predicted-errors. This strategy was able to search the optimal threshold values effectively. Wu and Huang [31] modified a pair of histogram bins in the way of selecting histogram bins in sequence. Four prediction modes produced a large amount of prediction-errors from cover image. Ultimately, a number of histogram pairs were enumerated to obtain the best performance after synthesis combination of those four modes. Xuan et al. [32] expanded prediction-errors using histogram pair according to embedding threshold, fluctuation threshold, and left-

and right-histogram shrinking thresholds. The embedding threshold was for selecting prediction-errors not larger than the threshold value. The fluctuation threshold was applied to select prediction-errors, whose associated neighbor fluctuation not exceeding this threshold. The left- and right-histogram shrinking thresholds were used to find the left and right possibly shrink histogram. Just for the condition of satisfied four thresholds, embedding data was carried out.

2.2. PVO-Based Predictors. Recently, based on PEE method, Li et al. [24] presented a high-fidelity data hiding method incorporating a new prediction strategy using pixel-value-ordering. Cover image was partitioned to be equal-sized but nonoverlapped pixel blocks. Pixels in each block were then ordered by gray-scale values. In this pixel sequence, the maximum and minimum pixels were assigned to predict the submaximum and subminimum pixels. Hence, this pixel selection mechanism reduced the number of shifted pixels for deprecating embedding distortion.

Peng et al. [26] pointed that 0 bin was usually the peak in image histogram and took the relative location relationships of pixels in a block. Improved PVO was raised by expanding error 0 for embedding other than positive 1 and negative error -1 in the previous work of PVO. PVO-K came up and Ou et al. [25] took equivalent pixels of all maximum-valued or minimum-valued ones as unit for embedding. Comparing with PVO, in which only two bits can be embedded into each block at most, PVO-K exploited image redundancy better for more allowable bits.

2.3. PPVO Predictor. All aforementioned predictors predicted pixel in a block-by-block manner. To some extent, this mechanism limited the improvement of EC with high PSNR. Qu and Kim [27] raised a novel pixel-based PVO method, with which pixel is predicted in pixel-by-pixel way. This method exploded block constraint and predicted more pixels than the above block-based predictors. Beyond question, PPVO promoted embedding capacity sufficiently. Right here, context pixels, defined to be neighboring pixels inside some scope, assisted to predict the current pixel. Sorting and ordering must be executed first before prediction. Each prediction value was determined by $\max(C)$ or $\min(C)$, the maximum or minimum of context pixel vector C . When $\max(C)$ is different from $\min(C)$, pixel can get prediction if its gray-scale value x is more than $\max(C)$ or smaller than $\min(C)$. When $\max(C)$ is the same as $\min(C)$, equaling a constant number VC , prediction worked only if pixel value is equal to or less than VC . The following formula here is applied to describe this predictor:

$$\hat{x} = \begin{cases} \max(C), & \text{if } \max(C) \neq \min(C), x \geq \max(C), \\ \min(C), & \text{if } \max(C) \neq \min(C), x \leq \min(C), \\ 254, & \text{if } \max(C) = \min(C) = VC, VC = 254, \\ VC, & \text{if } \max(C) = \min(C) = VC, 1 \leq VC \leq 253, x \leq VC. \end{cases} \quad (1)$$

With (1), some conclusions can be drawn as follows:

- (a) Pixels in smooth blocks tend to be predicted. After sorting and ordering, the actual objective becomes to search equivalent or approximate pixels in context pixel vector.
- (b) Different number of context pixels leads to different predicting accuracy, concerned with image distortion. For better embedding performance, as many as possible context pixels should be selected.
- (c) Context pixels locate inside square blocks. Besides the number, context pixel distribution may cause influences on prediction accuracy.

After intensive investigation, we find that when context pixels are the same ones, prediction probability is on the low side for two reasons. Conventional PPVO took 254 as a typical pixel when $VC = 254$, which cut down nearly a half of possible pixel values. Only pixels equal to or smaller than VC can be predicted. The second feature is that only prediction-error 0 can be expanded for carrying data, which wastes some useful errors for expansion and embedding data. Furthermore, there are still some other optimal context pixel selecting strategies which may strengthen data hiding performance in both EC and PSNR. Hence, in next section, a hybrid predictor based on PPVO is proposed.

3. Proposed Method

PPVO generates predictions in a pixel-by-pixel manner. It breaks through block constraints in PVO, PVO-K, and IPVO. Each pixel is predicted by a context vector that contains many sorted neighboring pixels. Huge amounts of pixel redundancy

in smooth image regions that are ignored in other PVO-based methods can be used to embed data. Unfortunately, only prediction-error 0 is involved in prediction. This addresses smooth image region pixels but it does not address common region pixels. Using more prediction values should improve embedding capacity.

3.1. Hybrid Predictor Based on PPVO

3.1.1. Pixel Prediction. For clarity, the same notations used in PPVO were used in this section. In PPVO, pixels other than the current pixel in a dynamic block are considered context pixels for prediction. After sorting in ascending order, a context pixel vector consisting in such neighboring pixels is defined and noted as $C = (c_1, \dots, c_{CN})$. Here, CN is the number of referenced pixels. The current pixel x can be predicted by the maximum $\max(C)$ and minimum $\min(C)$. Pixels are predicted in two different scenarios. One case is that context pixels have distinct numerical values, and if the gray-scale value of the current pixel $x \geq \max(C)$, then the prediction value is $\max(C)$. Similarly, if the current pixel $x \leq \min(C)$, then prediction value is $\min(C)$. If the current pixel value is greater than $\min(C)$ and less than $\max(C)$, no prediction can be made. Another condition is that all the pixels are identical with a constant number VC ; then the number just makes prediction. To prevent overflow and underflow, preprogress must be conducted before prediction. If pixel value is 255 or 0, it must be modified to 254 or 1 accordingly. These pixels should be recorded by a location map preparing for revision at decoder. Hence, this constant number VC is constrained in the range of $[1, 254]$. Obviously, prediction value of the current pixel x is determined by reference pixels in context pixel vector.

The current pixel x acquires prediction value \hat{x} as follows:

$$\hat{x} = \begin{cases} \max(C), & \text{if } \max(C) \neq \min(C), x \geq \max(C), \\ \min(C), & \text{if } \max(C) \neq \min(C), x \leq \min(C), \\ VC + 1, & \text{if } \max(C) = \min(C) = VC, 1 \leq VC \leq 254, x \geq VC + 1, \\ VC, & \text{if } \max(C) = \min(C) = VC, 1 \leq VC \leq 254, x \leq VC. \end{cases} \quad (2)$$

This makes great differences from prediction algorithm in [27] when all the context pixels take the same value. Differences are produced when $\max(C)$ is equal to $\min(C)$, denoted by VC , as illustrated in Figure 1. (a) shows that when $VC = 254$, PPVO predicts the current pixel only if it is 254, and otherwise it is skipped. However, based on that first step of prediction, HPPVO tries further to discover the lower pixel value than 254 and makes prediction. These two steps cover all the possible scenarios. (b) describes that when VC is less than 254 and is more than zero, PPVO employs only one step to estimate the current pixel just using the sole baseline of VC . Meanwhile, HPPVO takes two baselines, $VC + 1$ in the first step and VC in the second step. Obviously, the proposed

hybrid predictor of HPPVO provides more opportunities to get prediction.

Referring to Figure 2, in $(a_1) \sim (a_5)$, PPVO predicts the current pixel only in (a_1) while HPPVO makes predictions in all these five cases. In fact, according to this algorithm, the other 253 cases of the current pixel can all be predicted in HPPVO. In this light, prediction capacity of HPPVO is 254 times that of PPVO. Referring to Figure 2, in $(b_1) \sim (b_5)$, HPPVO still predicts the current pixel in all the cases. Yet, when the current pixel is more than VC , prediction is invalid and has to be skipped in that previous work of PPVO. In this sense, the smaller the VC , the less the prediction cases. So, HPPVO improves obviously prediction

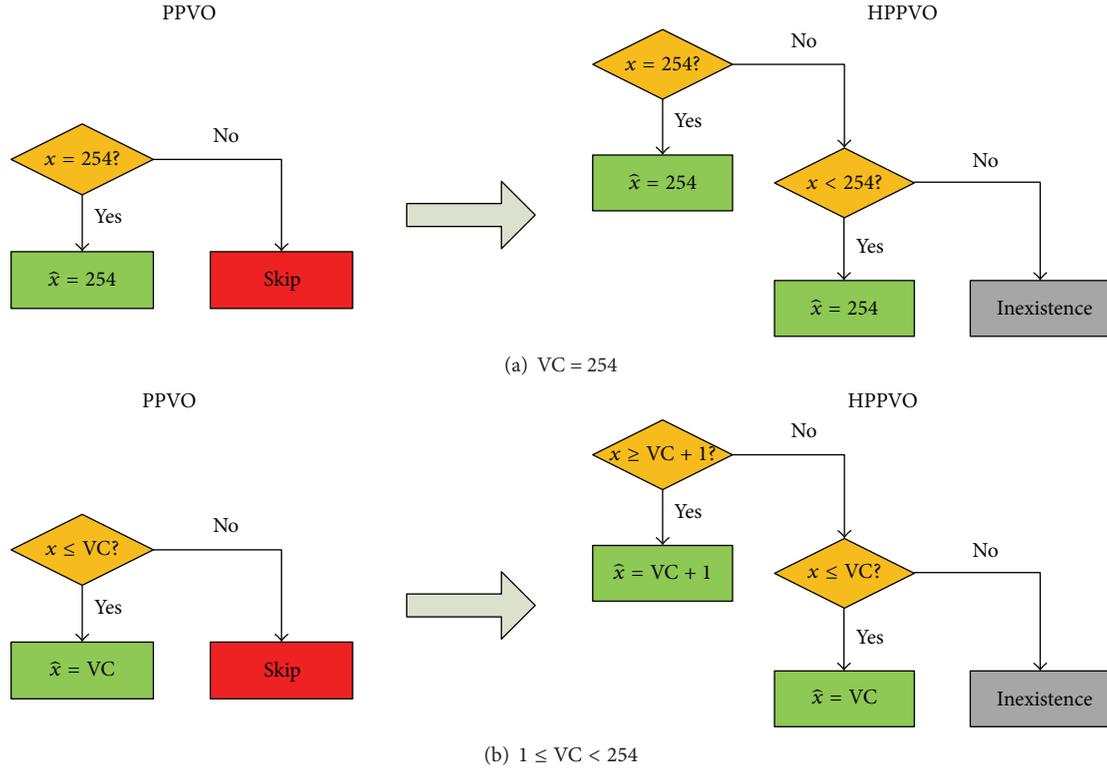


FIGURE 1: The figure puts forward some typical examples and Table 1 illustrates predictions accordingly. Four colors including red, green, yellow, and gray are filled into flowcharts for a straightforward explanation. They, respectively, indicate skip, making prediction, judging conditions, and inexistence.

TABLE 1: Prediction difference of PPVO and the proposed HPPVO.

Types/prediction	a_1	a_2	a_3	a_4	a_5	b_1	b_2	b_3	b_4	b_5
PPVO/ x	254	Skip	Skip	Skip	Skip	Skip	Skip	230	230	230
HPPVO/ x	254	254	254	254	254	231	231	230	230	230

performance compared to that of PPVO. Consistent with (2), VC and VC + 1 are probably taken as the prediction value in all possible cases. Conspicuously, this pivotal prediction modification may explore amazing pixel redundancy.

3.1.2. Data Embedding. Accordingly, the derived prediction-error is noted as follows:

$$e = x - \hat{x}. \quad (3)$$

For predicted pixels, prediction-error can be expanded and validated to embed one bit data b or shift for lossless recovery. For nonpredicted pixel, skip it over right away. When context pixel vector possesses unequal values, only those pixels which is equal to their prediction values can be used for embedding. If a pixel is predicted by the maximum of referenced vector, it remains unchanged when data bit is 0 and it is increased by 1 when data bit is 1. However, if a pixel is predicted by the minimum of referenced vector, it will remain unchanged when data bit is 0 and it will be decreased by 1 when data bit is 1. Otherwise, if prediction-error is unequal to zero, the pixel demands of the unit shift.

Thus, when inequality $\max(C) \neq \min(C)$ is satisfied, the original prediction-error e is expanded to be \tilde{e} as follows:

$$\tilde{e} = \begin{cases} e + b & \text{if } \hat{x} = \max(C), e = 0, \\ e + 1 & \text{if } \hat{x} = \max(C), e > 0, \\ e - b & \text{if } \hat{x} = \min(C), e = 0, \\ e - 1 & \text{if } \hat{x} = \min(C), e < 0. \end{cases} \quad (4)$$

Here, b is the to-be-embedded data bit, $b \in \{0, 1\}$.

Different from the previous work in PPVO, when context pixel vectors contain equivalent values, data may be embedded in two situations. First, if the current pixel is predicted by VC + 1, prediction-error zero is expanded. One-bit data 0 or 1 can be embedded and pixel value may remain the same or be increased by 1 at most. It is worth noting that VC + 1 but not VC is the baseline for prediction. So, actual expanded prediction-error amounts to +1. Second, if context pixels are 254, all the possible current pixel can be predicted by 254 and prediction-error zero will be expanded for data embedding. This is contained by the prediction VC. When bit

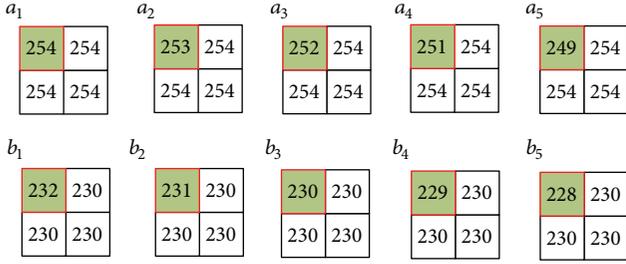


FIGURE 2: The top-left pixel is the current pixel x up for being predicted. Other ones in certain blocks are context pixels. The number of used context pixels is 3. $(a_1) \sim (a_5)$ are mapped from (a) in Figure 1. $(b_1) \sim (b_5)$ are mapped from (b) in Figure 1.

0 is embedded, the pixel remains unchanging and when bit 1 is embedded, the pixel is decreased by 1. Note that though VC is the baseline for judgment, if prediction-error is not equal to 0, the pixel is shifted. When the pixel is greater than predicted value, it is increased by 1 and when pixel is smaller than prediction by 1 or even less, it will be decreased by 1.

When equation $\max(C) = \min(C) = VC$ is established, the initial prediction-error e is expanded to be \tilde{e} as follows:

$$\tilde{e} = \begin{cases} e + b & \text{if } \hat{x} = VC + 1, e = 0, \\ e + 1 & \text{if } \hat{x} = VC + 1, e > 0, \\ e - b & \text{if } \hat{x} = VC, e = 0, \\ e - 1 & \text{if } \hat{x} = VC, e < 0. \end{cases} \quad (5)$$

After embedding, pixel is modified as follows:

$$\tilde{x} = \hat{x} + \tilde{e}. \quad (6)$$

So, in the case of $\max(C) \neq \min(C)$, marked pixels are defined as follows:

$$\tilde{x} = \begin{cases} x + b & \text{if } x = \max(C), \\ x + 1 & \text{if } x > \max(C), \\ x - b & \text{if } x = \min(C), \\ x - 1 & \text{if } x < \min(C). \end{cases} \quad (7)$$

In the case of $\max(C) = \min(C) = VC$, marked pixels are defined as follows:

$$\tilde{x} = \begin{cases} x + b & \text{if } x = VC + 1, \\ x + 1 & \text{if } x > VC + 1, \\ x - b & \text{if } x = VC, \\ x - 1 & \text{if } x < VC. \end{cases} \quad (8)$$

In Figure 3(a), both the current pixel and context pixel are equal to 254 in the first case. Data bit can be embedded on the basis of expanded prediction-error zero. But note that PPVO implements increasing modification $\tilde{x} = x + b$ and HPPVO carries decreasing transformation $\tilde{x} = x - b$. When context

pixels become 253 or other numbers less than 254, the current pixel has to be skipped without prediction in PPVO. However, HPPVO provides a chance for embedding. It takes $VC + 1 = 254$ as the judgment baseline and gets prediction-error zero. After being expanded, this error carries one-bit data. This just makes great difference from PPVO and improves embedding capacity sufficiently. Subsequently, if the current pixel is equal to or less than VC, PPVO and HPPVO execute same arithmetic operations. In Figure 3(b), HPPVO predicts current pixels larger than context pixels. If the current pixel has the value larger by 2 than VC, it will be shifted in HPPVO while being skipped even without prediction in PPVO.

Totally, HPPVO predicts current pixels by two baselines in the case of equivalent context pixels, which is called double prediction mechanism (DPM). This scheme is able to gain predictions not only for pixels equal to or fewer than VC but also for larger ones. Comparatively speaking, PPVO offers a single prediction mechanism (SPM), which cannot predict pixels larger than VC. Thus, more chances stand for prediction in HPPVO rather than in PPVO. This means embedding capacity will be improved by this hybrid predictor in HPPVO.

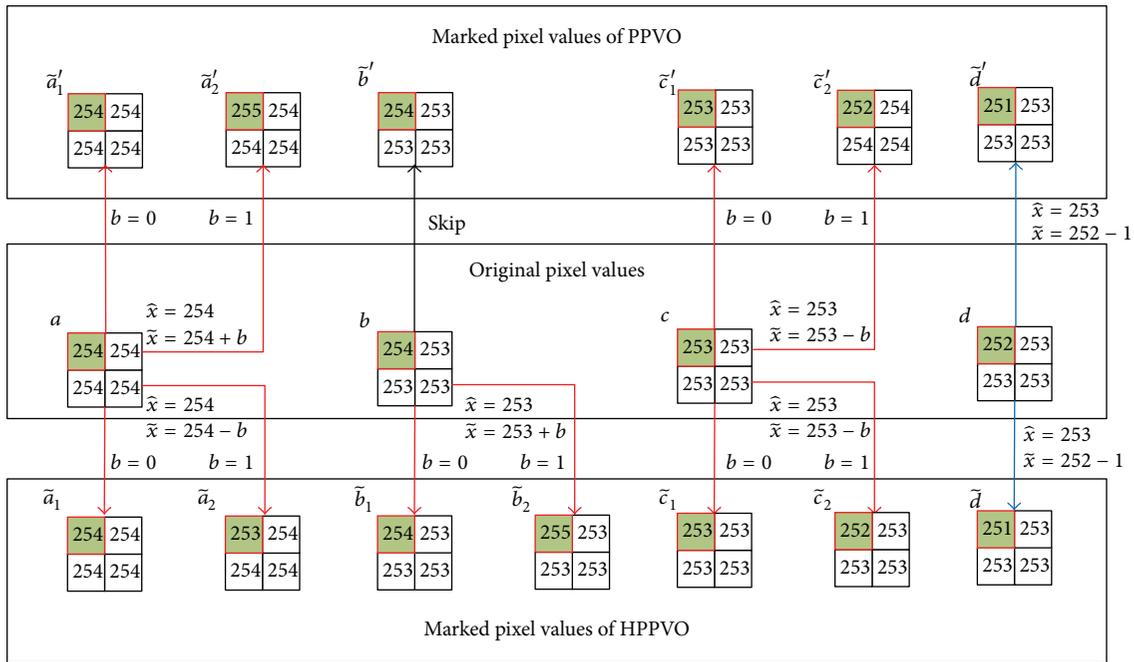
3.1.3. Error Expansion. To testify the advantage of the new proposed hybrid prediction and embedding method, error expansion is offered some contrast in Figure 4. Integer dotted errors are distributed on coordinate axis with finite extension in both the plus and minus directions. Red arrow denotes error expansion and blue arrow indicates error shift. For (a), nonpositive errors are limited in $[-253, 0]$ existing. If VC is 254, only error 0 is expanded. If VC is not equal to 254, error 0 is expanded and errors less than 0 are shifted in negative-going. For (b), integer errors are in the region of $[-253, 253]$. If prediction value is equal to $VC + 1$, error 0 is expanded and other positive errors are shifted in forward direction. If pixel is predicted by VC, error 0 is expanded and other negative errors are shifted in inverse way. Due to the above, zero error here contains abundant significance. In fact, error 0 produced by $VC + 1$ amounts to error 1 judged by VC. 0^+ and 0^- are used to make these two errors 0 with different meaning distinct.

3.1.4. Bit Extraction and Pixel Recovery. In view of the above-mentioned procedure for hiding data, the embedded data and the host image must be perfectly restored from the marked image. Mathematically, inverse calculation of embedding should be implemented. The same prediction logic as embedding is needed first. All the context pixels for prediction must be the same as the embedded pixels, so searching for pixels must involve an anti-raster-scan order.

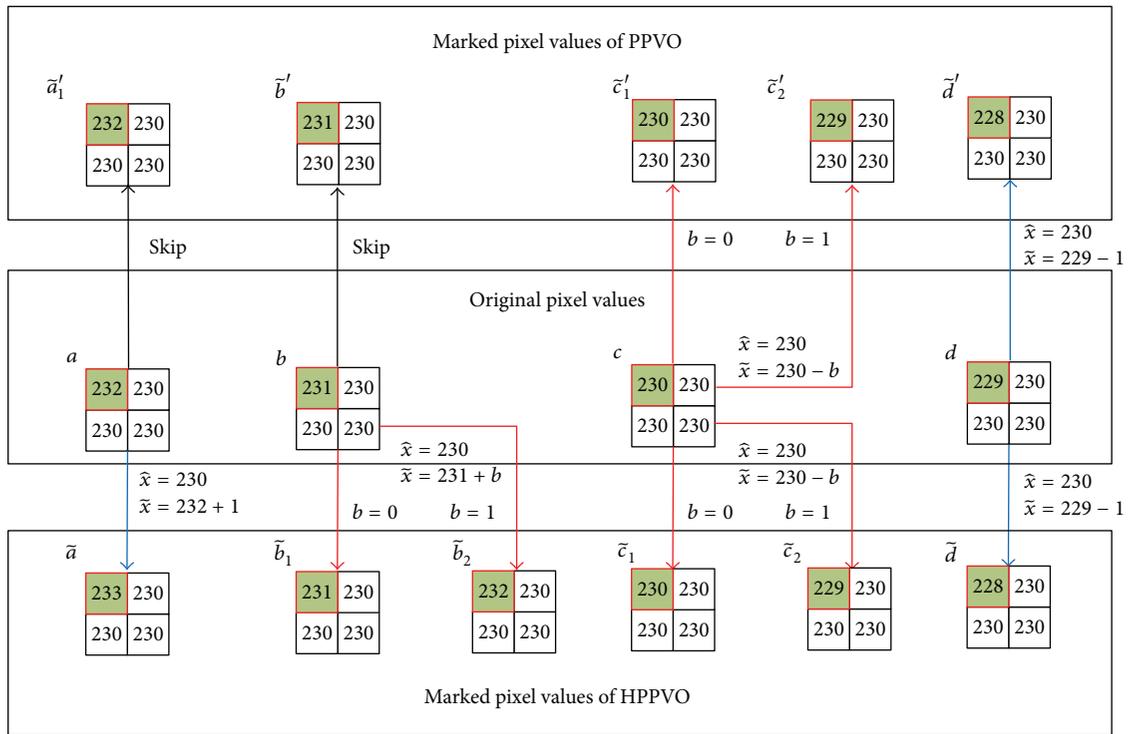
To predict the current marked pixel \tilde{x} , prediction-error is noted as follows:

$$\tilde{e} = \tilde{x} - \hat{x}. \quad (9)$$

Referring to (4), in the case of different values in the context pixel vector, the maximum $\max(C)$ or the minimum $\min(C)$ is taken as prediction. In the first case, when error \tilde{e} is zero, one-bit 0 is extracted and the according original



(a) VC = 254 and VC = 253



(b) VC = 230

FIGURE 3: PPVO and HPPVO embedding schemes in the case of equivalent context pixels. The top-left pixel is the current pixel to be predicted. There are three context pixels and the pixel values are equal to a constant VC. The red solid lines indicate embedded data bits. The blue solid lines indicate shifts. The black solid lines indicate skipping. Rows indicate pixel transformation.

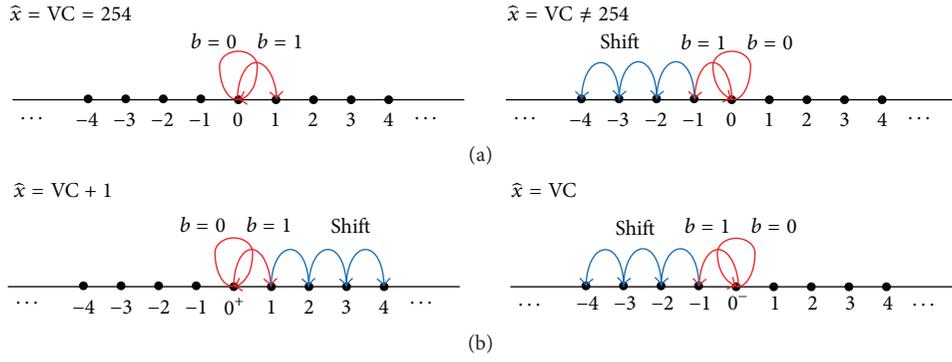


FIGURE 4: Error expansions. Red arrow stands for expanded errors. Blue arrow is shifted. b is the data bit to be embedded. (a) shows two cases in PPVO; (b) stands for the corresponding cases in HPPVO.

pixel is just the same as the current marked pixel. When error \tilde{e} equals one, one-bit 1 is drawn and subtracting 1 from the current pixel makes the original value. When error \tilde{e} is larger than 1, no bit will be extracted, but one unit must be shifted by subtraction. In the second case, if error $\tilde{e} = 0$, the extracted bit $b = 0$ and the pixel holds unchanged. If error $\tilde{e} = -1$, the extracted bit $b = 1$ and the pixel is increased by 1. If error $\tilde{e} < -1$, no bit can be drawn and the pixel is shifted by increasing it by 1. As shown in Section 3.2, only prediction-error 0 is operated. In other cases, no prediction can be obtained and skip this pixel.

Thus, when $\max(C) \neq \min(C)$, error \tilde{e} is decoded as follows:

$$e = \begin{cases} \tilde{e}, b = 0 & \text{if } \hat{x} = \max(C), \tilde{e} = 0, \\ \tilde{e} - 1, b = 1 & \text{if } \hat{x} = \max(C), \tilde{e} = 1, \\ \tilde{e} - 1 & \text{if } \hat{x} = \max(C), \tilde{e} > 1, \\ \tilde{e}, b = 0 & \text{if } \hat{x} = \min(C), \tilde{e} = 0, \\ \tilde{e} + 1, b = 1 & \text{if } \hat{x} = \min(C), \tilde{e} = -1, \\ \tilde{e} + 1 & \text{if } \hat{x} = \min(C), \tilde{e} < -1. \end{cases} \quad (10)$$

Referring to (5), in the condition of all the context pixels equal to a certain number VC , both $VC + 1$ and VC are candidate prediction baselines. Errors 0 and -1 are already decoded for hybrid extraction. When pixel is predicted by $VC + 1$, three cases have to be considered. If error is 0, data bit is 0 and pixel is the same without any modification. If error is 1, $b = 1$ is extracted. Otherwise, error is larger than 1, pixel is shifted, and no bit can be decoded. In the last two cases, pixel will be decreased by 1 for lossless recovery. When pixel is predicted by VC , similar three cases demand for consideration. If error is 0, data bit is 0 and pixel is the same without any modification. If error is -1 , $b = 1$ is extracted. Otherwise, error is less than -1 , pixel is shifted, and no bit can be decoded. For the last two scenarios, pixel will be increased by 1 for image recovery.

When $\max(C) = \min(C) = VC$, error \tilde{e} is restored as follows:

$$e = \begin{cases} \tilde{e}, b = 0 & \text{if } \hat{x} = VC + 1, \tilde{e} = 0, \\ \tilde{e} - 1, b = 1 & \text{if } \hat{x} = VC + 1, \tilde{e} = 1, \\ \tilde{e} - 1 & \text{if } \hat{x} = VC + 1, \tilde{e} > 1, \\ \tilde{e}, b = 0 & \text{if } \hat{x} = VC, \tilde{e} = 0, \\ \tilde{e} + 1, b = 1 & \text{if } \hat{x} = VC, \tilde{e} = -1, \\ \tilde{e} + 1 & \text{if } \hat{x} = VC, \tilde{e} < -1. \end{cases} \quad (11)$$

Recovered pixels are as follows:

$$x = \hat{x} + e. \quad (12)$$

Under the condition of $\max(C) \neq \min(C)$, the current pixel is recovered as follows:

$$x = \begin{cases} \tilde{x}, b = 0 & \text{if } \tilde{x} = \max(C), \\ \tilde{x} - 1, b = 1 & \text{if } \tilde{x} = \max(C) + 1, \\ \tilde{x} - 1, \text{ no bit is extracted} & \text{if } \tilde{x} > \max(C) + 1, \\ \tilde{x}, b = 0 & \text{if } \tilde{x} = \min(C), \\ \tilde{x} + 1, b = 1 & \text{if } \tilde{x} = \min(C) - 1, \\ \tilde{x} + 1, \text{ no bit is extracted} & \text{if } \tilde{x} < \min(C) - 1. \end{cases} \quad (13)$$

Under the condition of $\max(C) = \min(C) = VC$, the current pixel is recovered as follows:

$$x = \begin{cases} \tilde{x}, b = 0 & \text{if } \tilde{x} = VC + 1, \\ \tilde{x} - 1, b = 1 & \text{if } \tilde{x} = VC + 2, \\ \tilde{x} - 1, \text{ no bit is extracted} & \text{if } \tilde{x} > VC + 2, \\ \tilde{x}, b = 0 & \text{if } \tilde{x} = VC, \\ \tilde{x} + 1, b = 1 & \text{if } \tilde{x} = VC - 1, \\ \tilde{x} + 1, \text{ no bit is extracted} & \text{if } \tilde{x} < VC - 1. \end{cases} \quad (14)$$

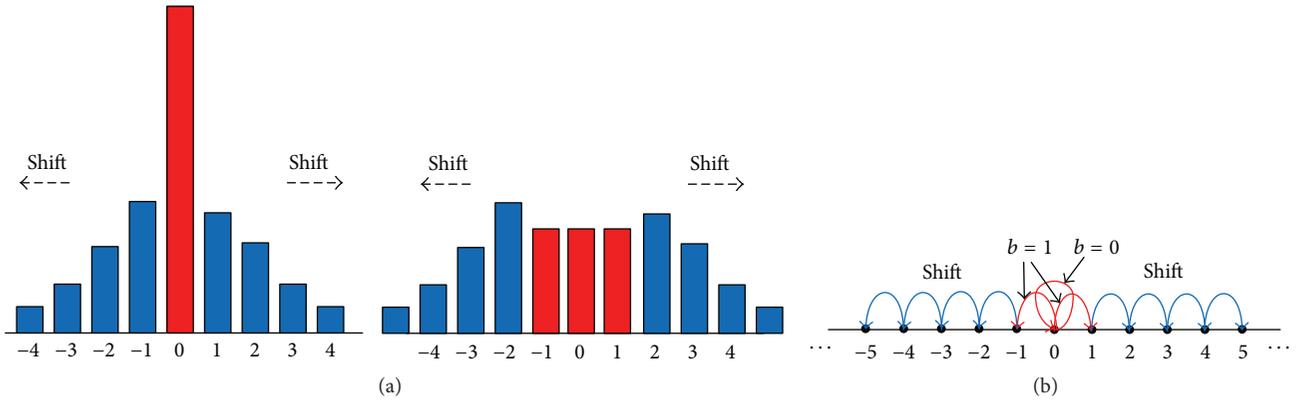


FIGURE 5: Histogram modification mechanism for Lena image without EBS. Red bins stand for expanded errors. Blue bins are shifted. Red arrow stands for expanded errors. Blue arrow is shifted. b is the data bit to be embedded. (a) The left is the PEH before and the right is the according one after embedding. (b) Mapping of bins.

3.2. Expansion Bins Selection. Prediction-error histogram (PEH) is usually to demonstrate the frequencies of prediction-errors for certain image. Suppose that N cover pixels denoted by (x_1, \dots, x_N) are collected for embedding. With (3), prediction-error is calculated by

$$e_i = x_i - \hat{x}_i, \quad (15)$$

where $i \in [1, N]$. PEH can be derived as

$$h(e) = \# \{1 \leq i \leq N : e_i = e\}, \quad \forall e \in \mathbb{Z}, \quad (16)$$

where $\#$ means a cardinal number set. Consider here a PEH for Lena image illustrated by Figure 5, which is generated from the above proposed HPPVO.

Figure 5 describes prediction-error bins in common. With (4) and (5), only error 0 bin is expanded and other bins are shifted to make vacancies for lossless reversibility. For the left figure in (a), 0 bin forms a rather sharp peak before embedding. In line with (16), note histogram bin $h(e_i)$ for error e_i in (e_1, \dots, e_N) . For the right figure in (a), it is partitioned into three parts after embedding. One part will remain in original post for $b = 0$ and the other two parts will shift towards the right and left for $b = 1$, which is caused by (3), (4), and (5). Positive error bins pass on their right locations and negative bins move to adjacent left places. (b) shows error expansion using red arrow and shift by blue arrow. On this occasion, the expected embedding distortion in l^2 -norm can be computed by

$$\begin{aligned} E_{(0,0)}^d &= E \left(\|\tilde{I} - I\|_2^2 \right) = \sum_{i=1}^N E \left((\tilde{x}_i - x_i)^2 \right) \\ &= \frac{2}{3}h(0) + \sum_{\substack{e \notin \{0\}, \\ e \in \{-253, \dots, 253\}}} h(e) = N - \frac{h(0)}{3}, \end{aligned} \quad (17)$$

where I and \tilde{I} denote cover and marked images, respectively. However, one can select the other bin or bins for expansion rather than 0.

From the above, data bit can be embedded for both positive and negative prediction-error either for error 0. Figures 6, 7, and 8 display cases for several distinct coupled expansion bins (e_l, e_r) . e_l and e_r indicate one side of a pair of expansion bins satisfying $e_l \in [-253, 0]$ and $e_r \in [0, 253]$. Therefore, histogram modification mechanism for Lena image without EBS in Figure 5 can be considered to be a typical case $(0, 0)$. When $(e_l, e_r) = (-1, 0)$ illustrated in Figure 6, -1 bin can be employed to carry data bits. Parts of error -1 are unchanged for embedding $b = 0$ and the others are shifted to error -2 for carrying $b = 1$. In this situation, the likely embedding distortion can be formulated as

$$\begin{aligned} E_{(-1,0)}^d &= E \left(\|\tilde{I} - I\|_2^2 \right) = \sum_{i=1}^N E \left((\tilde{x}_i - x_i)^2 \right) \\ &= \frac{2}{3}h(0) + \frac{1}{2}h(-1) + \sum_{\substack{e \notin \{-1,0\}, \\ e \in \{-253, \dots, 253\}}} h(e) \quad (18) \\ &= N - \frac{h(0)}{3} - \frac{1}{2}h(-1). \end{aligned}$$

When $(e_l, e_r) = (0, 1)$ showed in Figure 7, 1 bin can be employed to carry data bits. Parts of error 1 are unmodified for embedding $b = 0$ and the others are moved to the place of error 2 for carrying $b = 1$. This declares more embedding capacity than that in Figure 5. In this case, the conceivable distortion will be counted as

$$\begin{aligned} E_{(0,1)}^d &= E \left(\|\tilde{I} - I\|_2^2 \right) = \sum_{i=1}^N E \left((\tilde{x}_i - x_i)^2 \right) \\ &= \frac{2}{3}h(0) + \frac{1}{2}h(1) + \sum_{\substack{e \notin \{0,1\}, \\ e \in \{-253, \dots, 253\}}} h(e) \quad (19) \\ &= N - \frac{h(0)}{3} - \frac{1}{2}h(1). \end{aligned}$$

When $(e_l, e_r) = (-1, 1)$ showed in Figure 8, both -1 and 1 bins can be used to carry data bits. Parts of them are

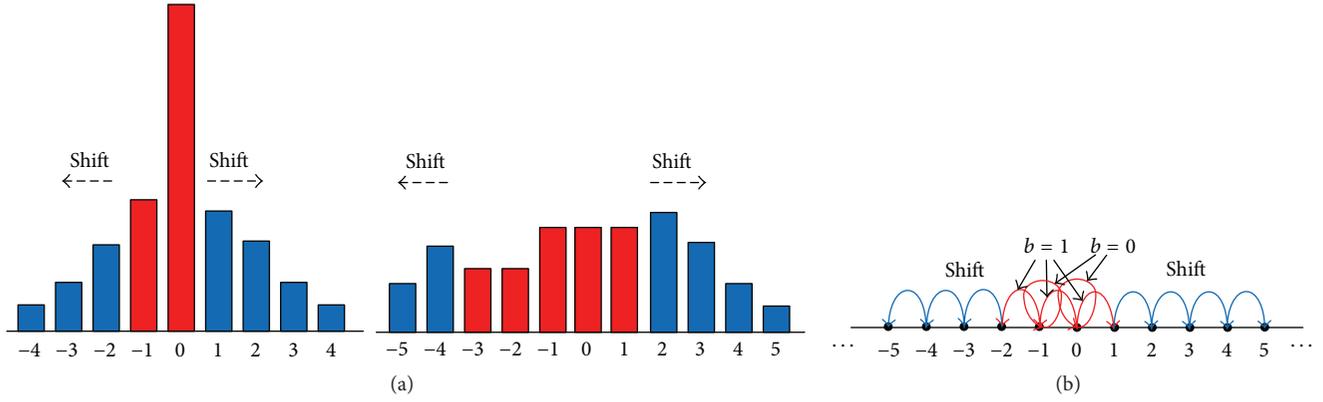


FIGURE 6: Histogram modification mechanism for Lena image with coupled expansion bins $(-1, 0)$. (a) The left is the PEH before and the right is the according one after embedding. (b) Mapping of bins.

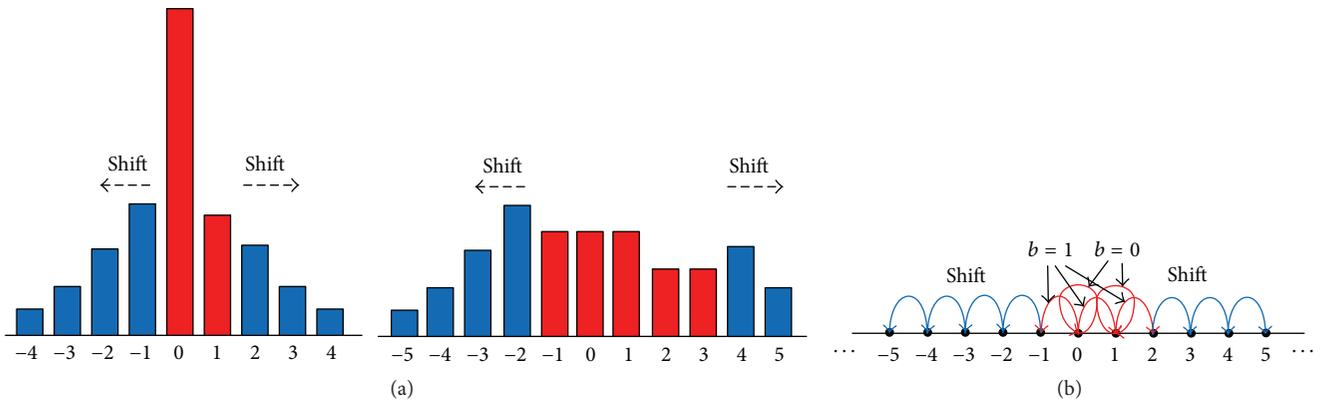


FIGURE 7: Histogram modification mechanism for Lena image with coupled expansion bins $(0, 1)$. (a) The left is the PEH before and the right is the according one after embedding. (b) Mapping of bins.

unmodified for embedding $b = 0$. But -1 has to move to left and 1 must shift towards right for adjacent post to embed a bit $b = 1$. It is a remarkable fact that 0 bin occupying overwhelming pixels is skipped. This will improve embedding performance sufficiently for the following expression:

$$\begin{aligned}
 E_{(-1,1)}^d &= E \left(\|\tilde{I} - I\|_2^2 \right) = \sum_{i=1}^N E \left((\tilde{x}_i - x_i)^2 \right) \\
 &= \frac{1}{2} (h(-1) + h(1)) + \sum_{\substack{e \notin \{-1,0,1\}, \\ e \in \{-253, \dots, 253\}}} h(e) \quad (20) \\
 &= N - h(0) - \frac{1}{2}h(-1) - \frac{1}{2}h(1).
 \end{aligned}$$

On the whole, EBS will improve embedding capacity shown in Figures 5–8. In addition, taking a comparison with (17) by (18), (19), and (20), the inequation denoted by (21) comes into being. So, it can be concluded that expansion bins selecting strategy contributes to bringing

down embedding distortion. Thus, it is probable for better embedding performance using EBS that

$$\begin{aligned}
 N - \frac{h(0)}{3} &\leq \min \left(\left(N - \frac{h(0)}{3} - \frac{1}{2}h(-1) \right), \right. \\
 &\quad \left. \left(N - \frac{h(0)}{3} - \frac{1}{2}h(1) \right) \right) \\
 &\leq \max \left(\left(N - \frac{h(0)}{3} - \frac{1}{2}h(-1) \right), \right. \\
 &\quad \left. \left(N - \frac{h(0)}{3} - \frac{1}{2}h(1) \right) \right) \leq N - h(0) - \frac{1}{2}h(-1) \\
 &\quad - \frac{1}{2}h(1).
 \end{aligned} \quad (21)$$

3.3. Field-Biased Context Pixel Selection (FCPS). According to different functions, pixels can be categorized into two types. One is to-be-predicted type and the other one is auxiliary prediction type. The latter pixels are also known as context pixels. For simplicity, we call the to-be-predicted pixel the current pixel. Weighted example introduced in [27] will be analyzed further. To be more intuitive, the different

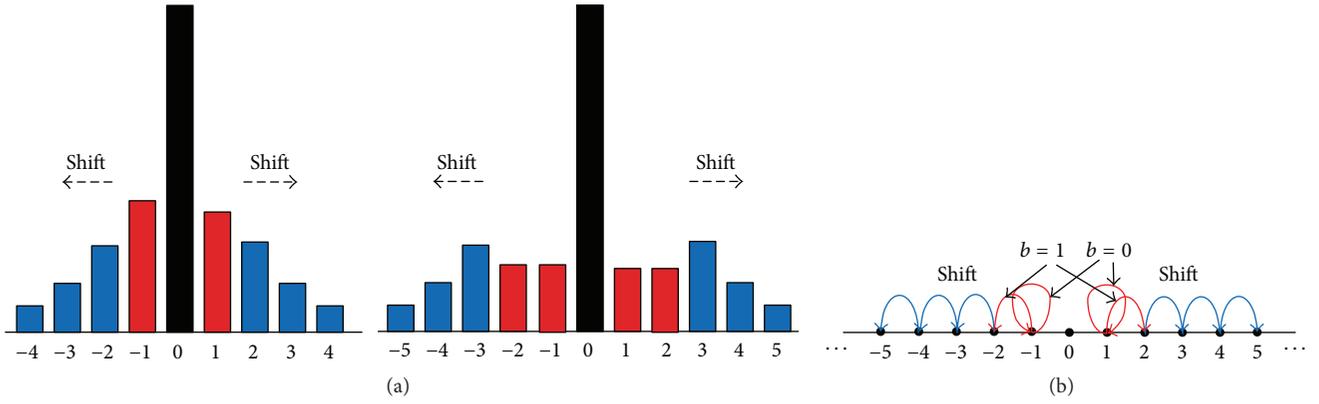


FIGURE 8: Histogram modification mechanism for Lena image with coupled expansion bins $(-1, 1)$. Red bins stand for expanded errors. Blue bins are shifted. Black bins are skipped. (a) The left is the PEH before and the right is the according one after embedding. (b) Mapping of bins.

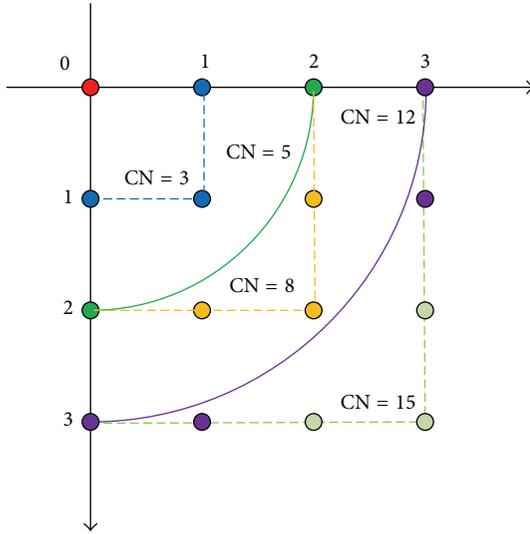


FIGURE 9: Equalizing distribution of context pixels in PPVO (CN = 3, 5, 8, 12, and 15).

distribution of context pixels is given which is illustrated in Figure 9 for different pixels number denoted by CN. When CN = 3, 8, and 15, context pixels locate in the inner and marginal regions of a square, laid at the top-left vertex by the current pixel and lengthened by pixel intervals of 1, 2, and 3 times unit pixel intervals, respectively. While CN = 5 or 12, context pixels are distributed in the inner and marginal regions of a right angle sector, whose radii are 2 or 3 times unit pixel intervals. In the sight of pixel locations, context pixels dot certain organized areas, such as squares and right-angle sectors. So, this is called well-balanced distribution or equalizing distribution.

Actually, the number and distribution of context pixels represent some correlation information about the current pixel. It can be interpreted as gray value variation of image pixels. Difference $e_i = x_i - x_o$ may be employed to represents this changing process, where x_i is the current pixel and x_o is a selected context pixel. This difference value stands for

the similarity degree of two pixels. If e_i tends to zero still more, the according two pixels are more similar. Otherwise, similar degree declines. The larger CN is, the more the specifically rounded pixels evolve.

In terms of pixel similarity, distance potential can be defined as follows.

Definition. Based on a certain baseline pixel, distance potential of another pixel is measured by

$$E = \bar{\varepsilon} \sqrt{k_h^2 + k_v^2} \cdot d_u, \quad (22)$$

where ε is pixel value gradient, d_u indicates the unit horizontal or vertical pixel distance, and k_h, k_v are unit distance times. Assuming that image resolution is $W \times H$, a set is so denoted as $K = \{1, 2, \dots, \max(W, H)\}$. Obviously, $k_h \in K_h, k_v \in K_v$, and $K_h = K_v = K$.

Inspecting the right previous example, the current pixel is the baseline pixel. For a specified CN, here is $\forall k_h \in K_h, \exists k_v \in K_v$ satisfying $k_h = k_v$. Of course, $\max(k_h) = \max(k_v)$ and $\min(k_h) = \min(k_v)$. Considering different CN, when CN = 3, $\vec{k}_h = \vec{k}_v = [1]^T$. When CN = 5 and 8, $\vec{k}_h = \vec{k}_v = [1 \ 2]^T$. When CN = 12 and 15, $\vec{k}_h = \vec{k}_v = [1 \ 2 \ 3]^T$. Thus, context pixels not only depict the similarity between surrounding pixels versus the current pixel, but also demonstrate according distance potential for a certain similar degree.

Comparing with equalizing schemes, field-biased context pixels, shown in Figure 10, have different distance potential. In other words, $\forall k_h \in K_h, \exists k_v \in K_v$ is not always tenable. As far as different CN, k_h and k_v take distinct rules. When CN = 5, $\vec{k}_h = [1 \ 2]^T$ and $\vec{k}_v = [1]^T$. When CN = 7, $\vec{k}_h = [1 \ 2 \ 3]^T$ and $\vec{k}_v = [1]^T$. When CN = 11, $\vec{k}_h = [1 \ 2 \ 3]^T$ and $\vec{k}_v = [1 \ 2]^T$. These six aforementioned schemes discriminatively compress and amplify the variation region of similarity between the current pixel and context pixels in the two dimensions of horizontal and vertical orientation to some extent. These pixels associate some zonal features and therefore we called them field-biased context pixels. From distribution, field-biased context pixels collected difference of horizontal or vertical pixel transformation, which is conducive for describing

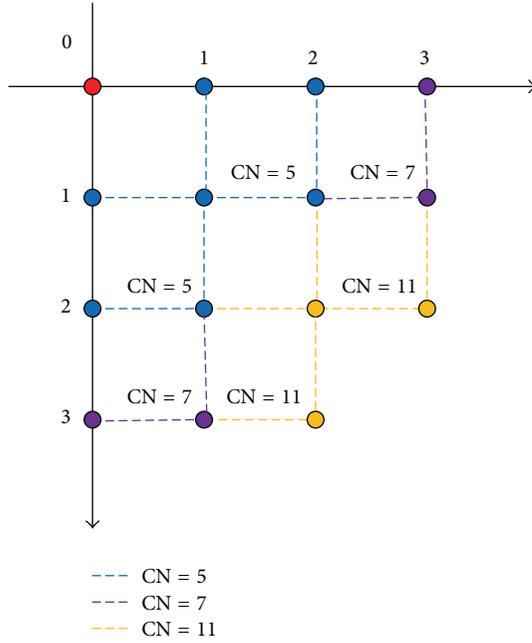


FIGURE 10: Optional field-biased context pixels (CN = 5, 7, and 11).

the image gray value changes in different directions. After introducing offset type context pixels, the current pixel tends to be able to predict more accurately than only equalizing schemes.

4. Experimental Results

Considering the embedding capacity and the peak signal-to-noise ratio (PSNR) value, the performance of the proposed HPPVO is tested with four previous works in [10, 24, 27, 33]. Standard formatted images of eight-bit gray-scale pictures in the size of 512×512 are selected from the SIPI image database. In Figure 11, Lena, Airplane (F16), Baboon, Barbara, Boat, and Peppers meet this demand. Certainly, random bit streams make the payloads for objective test.

4.1. PSNR Performance. Regarding PSNR value, HPPVO outperforms other four involved schemes in most cases, which is shown in Figure 12. Two methods in two previous works in [10, 33] were both found to be capable of embedding large amounts of data. However, PVO and its evolution methods provided much higher PSNR values. PPVO had considerable superiority in both embedding capacity and fidelity relative to three algorithms given above. In all the tested images, HPPVO increases the embedding capacity without decreasing the PSNR value regardless of small or large payload. Especially in smooth and very smooth images, Lena, Barbara, and Airplane (F16) show relatively obvious improvements in moderate and large payload sizes.

4.2. Embedding Capacity Performance. In terms of embedding capacity, Table 2 shows the results in different context pixels. Generally speaking, HPPVO obtains higher EC than

PPVO in most cases. For a certain image, context pixel number has effect on capacity improvement. As shown in Figure 13, Lena is improved in embedding capacity by 446 bits, 15 bits, 1 bit, 4017 bits, and 0 bits and F16 receives capacity gains by 2032 bits, 291 bits, 26 bits, 3325 bits, and 0 bits when the number of context pixels was 3, 5, 8, 12, and 15. This indicates that more data can be embedded into images with HPPVO than that with conventional PPVO. Yet, different CN causes different EC improvement. For more detailed comparison, the same and the adjacent numbers of context pixels are taken together. Here, 3, 5, 8, 12, and 15 are defined as central points. Accordingly, comparisons are drawn by maximum capacity of C_{5h} and C_{5v} with that of C_5 , maximum capacity of C_{7h} and C_{7v} with that of C_8 , and maximum capacity of C_{11h} and C_{11v} with that of C_{12} . Blue marked lines and circles in Figure 13 show that Lena increases by 446 bits, 1991 bits, 1974 bit, 6004 bits, and 0 bits, and F16 image gains improvements of 2032 bits, 3332 bits, 3320 bits, 11948 bits, and 3 bits' capacity. This demonstrates that the proposed hybrid works strikingly with similar context numbers. It is a remarkable fact that when CN is near to 12, prominent improvement can be achieved. And so, in Figure 14, this item will help us to analyze how notable results will be obtained from images with different complexity.

We define the improvement ratio by

$$r_{IEC} = \frac{EC_h - EC_c}{EC_c}, \quad (23)$$

where the subscript h indicates the new proposed hybrid predictor and c denotes the conventional predictor. Based on Table 2, we drafted Figure 14. Clearly, the improved EC ratio to CN = 12 is outstanding among other cases. On this occasion, all images appear well. Lena, Barbara, Boat, and Peppers obtain 0.33, 0.25, 0.26, and 0.26 improvements in common complexity. F16, the smoothest one, gets the maximum ratio value of 0.44. Even the most rough image, Baboon, takes the minimum 0.18. These splendid performances make good annotation for EC improvement of the proposed HPPVO.

From above, some conclusions can be drawn as follows:

- The number of context pixels has an effect on capacity increases. Moderate and large number, here CN = 12, obtain the most prominent performance. This is because nearly half of predictable opportunities from the two-way hybrid predictor will be appended for pixel vacancies. Simultaneously, capacity can be improved for neighboring pixels in field-biased schemes to a greater extent than in equalizing schemes.
- Image complexity takes influence on embedding capacity. Smooth images receive higher capacity more than that of rough ones. This mainly relies on the above simulated examples, F16 and Lena, the most and the second-rate smooth images, acquire the best and the suboptimal capacity improvement, respectively.
- Texture feature produces distinct capacity gains. The texture is more visible, the better enhancement in

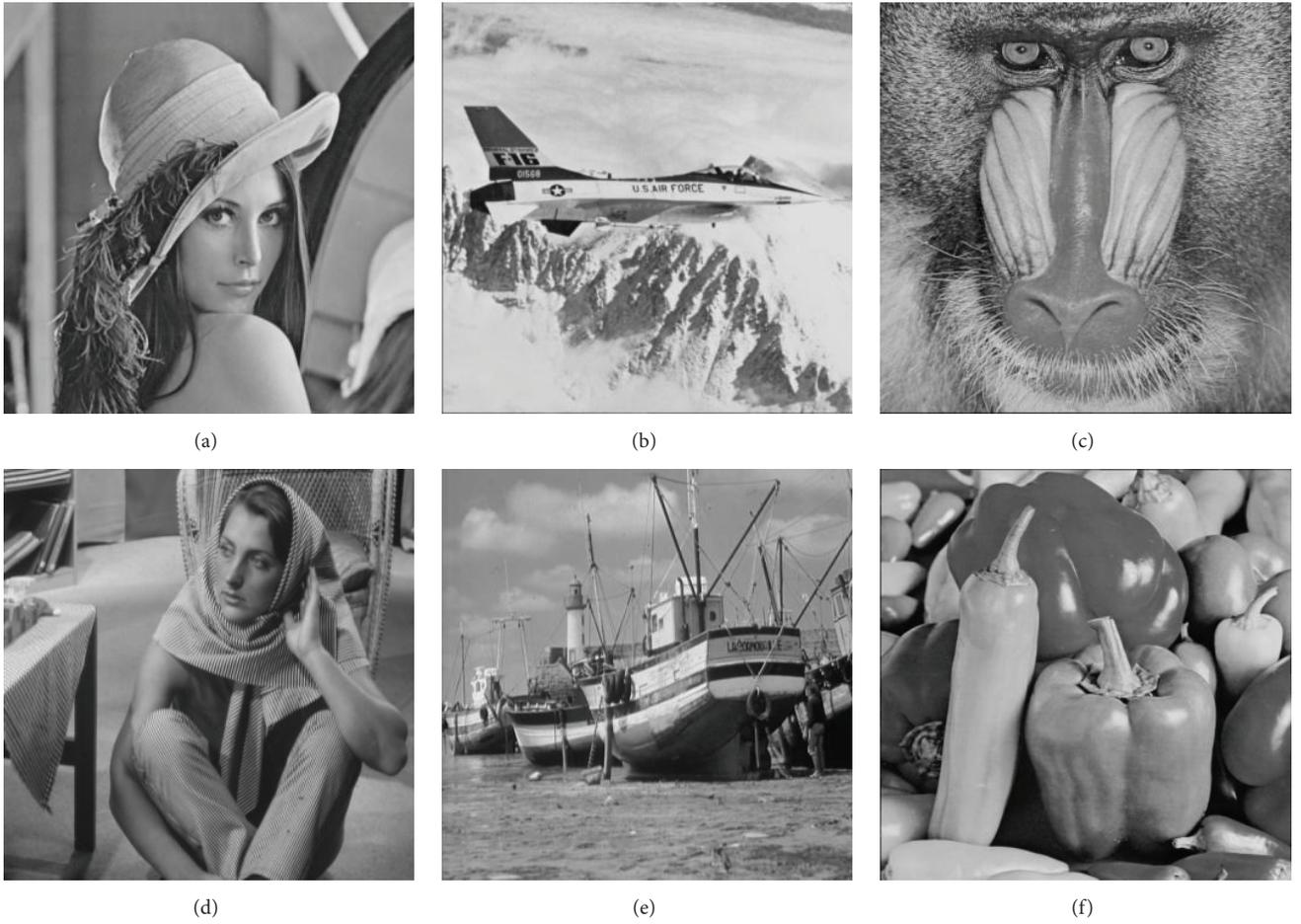


FIGURE 11: Test images of SIPI image data set. (a) Lena. (b) Airplane (F16). (c) Baboon. (d) Barbara. (e) Boat. (f) Peppers.

TABLE 2: Comparisons of embedding capacity (/bit) in terms of different context pixels between PPVO and HPPVO.

Image	C_3	C_5 C_{5h}/C_{5v}	C_8 C_{7h}/C_{7v}	C_{12} C_{11h}/C_{11v}	C_{15}
Lena	44,118/44,564	34,210/34,226 35,825/36,201	28,561/28,562 30,535/30,452	18,195/22,212 24,199/23,806	20,175/20,175
Baboon	13,548/13,576	10,361/10,361 10,830/10,630	8,367/8,367 9,172/8,876	5,995/6,532 7,047/6,958	5,779/5,779
Barbara	34,930/35,266	27,013/27,024 28,416/29,056	23,127/23,127 24,480/25,174	16,036/18,429 19,873/19,982	17,147/17,147
Boat	29,194/29,369	22,930/22,937 24,005/23,741	18,946/18,946 20,151/20,096	12,622/14,575 15,827/15,891	13,126/13,126
Peppers	33,439/33,628	26,341/26,344 27,311/27,368	21,524/21,524 23,267/23,265	14,469/16,865 18,226/18,249	15,435/15,435
F16	66,923/68,955	53,372/53,663 56,643/56,704	45,455/45,518 48,319/48,775	26,888/35,540 38,170/38,836	32,411/32,414
Cluster Average	37,025/37,560	29,021/29,093 30,505/30,617	24,330/24,356 25,987/26,106	15,701/19,026 20,557/20,620	17,346/17,346

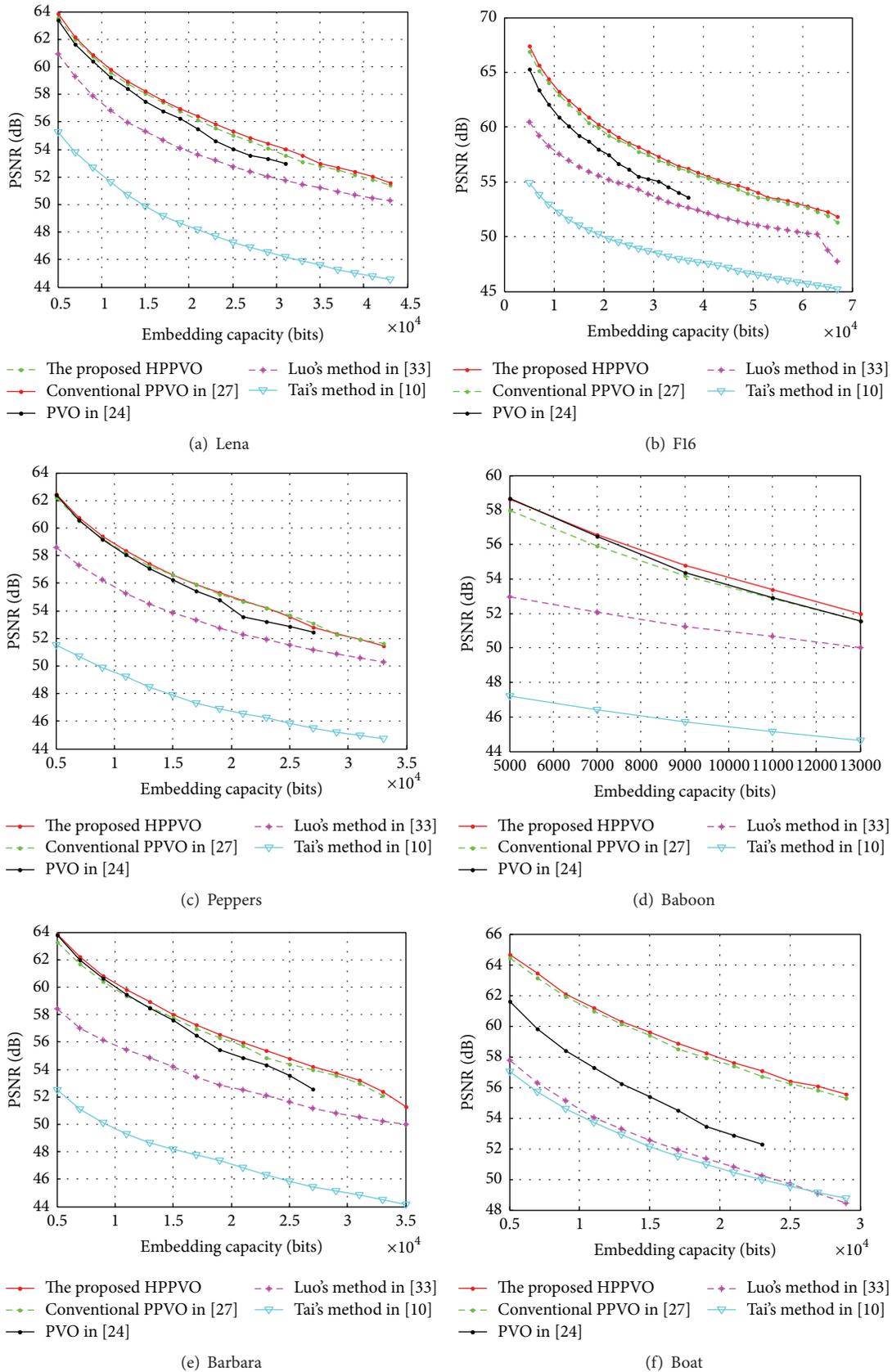


FIGURE 12: The performance comparison of proposed scheme and four state-of-the-art methods in [10, 24, 27, 33]. All the test images are from the SIPI image data set.

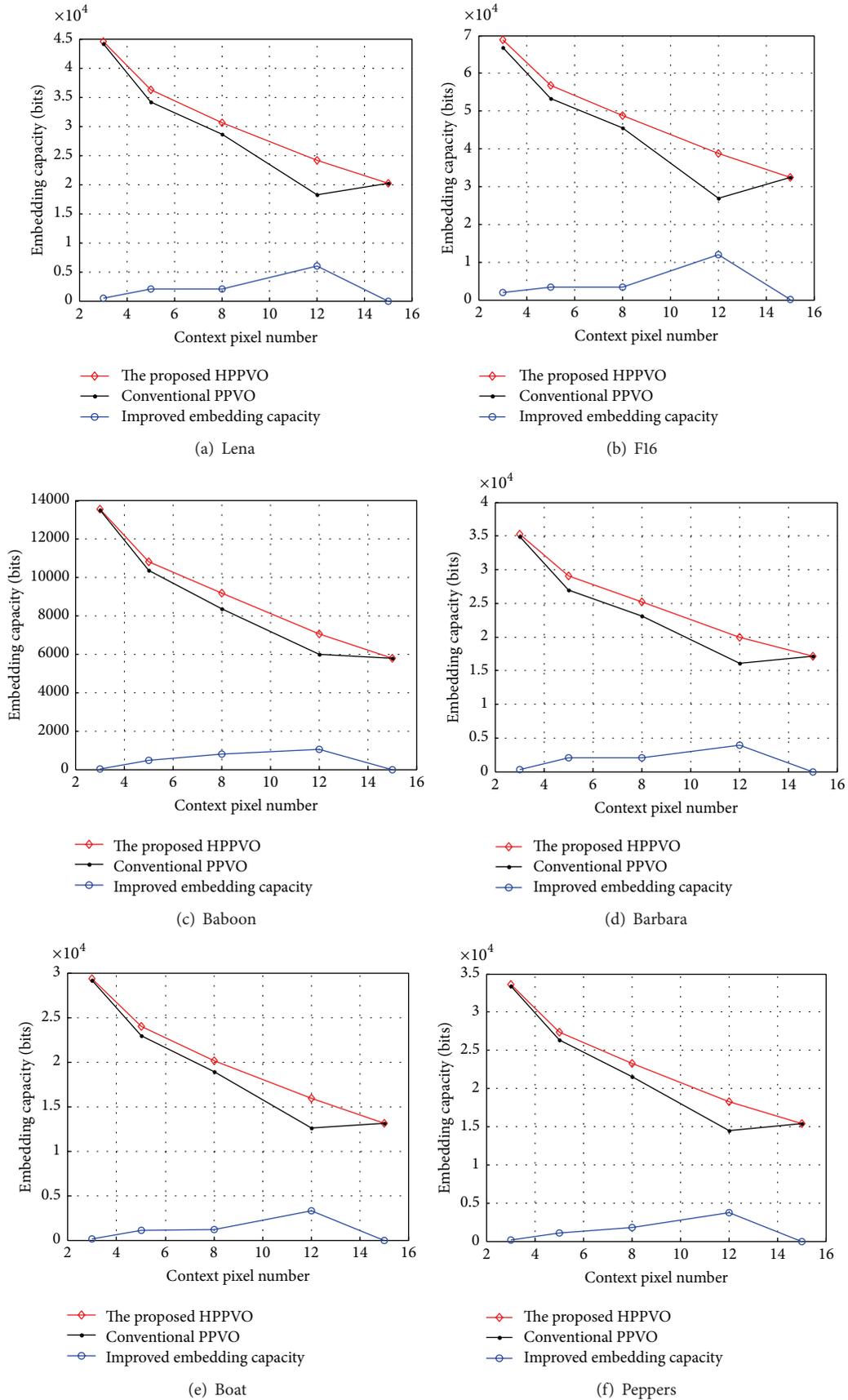


FIGURE 13: Improved capacity comparison for different numbers of context pixels. All the test images are from SIPI image data set.

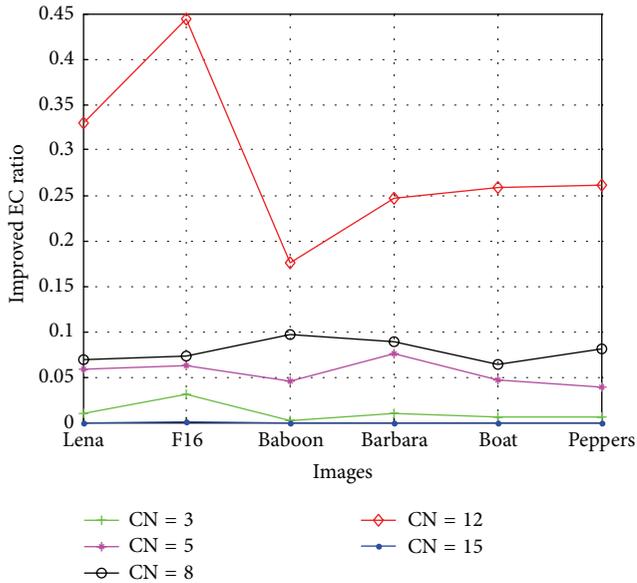


FIGURE 14: Comparison of improved EC ratios to images for different CN.

field-biased context pixel selections. This is because of the variation describing for texture pixels. Pixels in the zonal massif region in F16 image and the diagonal headwear area in Lena image contribute vast expandable prediction-errors.

Totally, HPPVO is more suitable for marked images claiming for high fidelity and large embedding capacity.

5. Conclusion

HPPVO, which is designed to create large embedding capacity without reducing the PSNR value, combines hybrid prediction and field-biased context pixel selections. New embedding logic enhances embedding data, and new context pixel scheme is found to improve PSNR values. Experimental results show that HPPVO takes outstanding advantages in smooth and very smooth images. This suggests that field-biased context pixel schemes merit further research.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

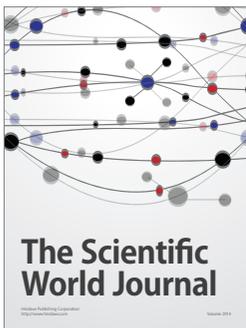
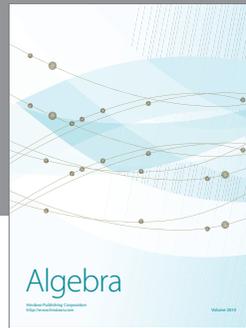
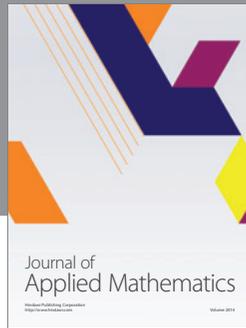
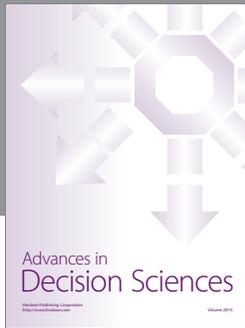
Acknowledgments

This work was supported by National Natural Science Foundation of China (no. 61302106 and no. 61201393), the Fundamental Research Funds for the Central Universities (no. 2014MS105 and no. 13MS66), and Hebei Province Natural Science Foundation of China, Youth Science Fund (no. E2013502267). Here, the authors are obliged to the above foundations, their responsible persons, and all their lab associates.

References

- [1] I. J. Cox, J. Kilian, F. T. Leighton, and T. Shamon, "Secure spread spectrum watermarking for multimedia," *IEEE Transactions on Image Processing*, vol. 6, no. 12, pp. 1673–1687, 1997.
- [2] C. I. Podilchuk and E. J. Delp, "Digital watermarking: algorithm and application," *IEEE Signal Processing Magazine*, vol. 18, no. 4, pp. 33–46, 2001.
- [3] G. C. Langelaar, I. Setyawan, and R. L. Lagendijk, "Watermarking digital image and video data," *IEEE Signal Processing Magazine*, vol. 17, no. 5, pp. 20–46, 2000.
- [4] S. Y. Chen, H. Tong, Z. Wang, S. Liu, M. Li, and B. Zhang, "Improved generalized belief propagation for vision processing," *Mathematical Problems in Engineering*, vol. 2011, Article ID 416963, 12 pages, 2011.
- [5] M. Barni, F. Bartolini, and A. Piva, "Improved wavelet-based watermarking through pixel-wise masking," *IEEE Transactions on Image Processing*, vol. 10, no. 5, pp. 783–791, 2001.
- [6] S. Y. Chen, J. Zhang, Q. Guan, and S. Liu, "Detection and amendment of shape distortions based on moment invariants for active shape models," *IET Image Processing*, vol. 5, no. 3, pp. 273–285, 2011.
- [7] H. Liu, S. Y. Chen, and N. Kubota, "Guest editorial special section on intelligent video systems and analytics," *IEEE Transactions on Industrial Informatics*, vol. 8, no. 1, p. 90, 2012.
- [8] J. Tian, "Reversible data embedding using a difference expansion," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 8, pp. 890–896, 2003.
- [9] A. M. Alattar, "Reversible watermark using the difference expansion of a generalized integer transform," *IEEE Transactions on Image Processing*, vol. 13, no. 8, pp. 1147–1156, 2004.
- [10] W.-L. Tai, C.-M. Yeh, and C.-C. Chang, "Reversible data hiding based on histogram modification of pixel differences," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 6, pp. 906–910, 2009.
- [11] Z. C. Ni, Y.-Q. Shi, N. Ansari, and W. Su, "Reversible data hiding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 3, pp. 354–362, 2006.
- [12] Y.-S. Juang, L.-T. Ko, J.-E. Chen, Y.-S. Shieh, T.-Y. Sung, and H. C. Hsin, "Histogram modification and wavelet transform for high performance watermarking," *Mathematical Problems in Engineering*, vol. 2012, Article ID 164869, 14 pages, 2012.
- [13] Y.-Y. Tsai, D.-S. Tsai, and C.-L. Liu, "Reversible data hiding scheme based on neighboring pixel differences," *Digital Signal Processing*, vol. 23, no. 3, pp. 919–927, 2013.
- [14] X. L. Li, B. Li, B. Yang, and T. Y. Zeng, "General framework to histogram-shifting-based reversible data hiding," *IEEE Transactions on Image Processing*, vol. 22, no. 6, pp. 2181–2191, 2013.
- [15] L.-T. Ko, J.-E. Chen, H.-C. Hsin, Y.-S. Shieh, and T.-Y. Sung, "Haar-wavelet-based just noticeable distortion model for transparent watermark," *Mathematical Problems in Engineering*, vol. 2012, Article ID 635738, 14 pages, 2012.
- [16] S. Lee, C. D. Yoo, and T. Kalker, "Reversible image watermarking based on integer-to-integer wavelet transform," *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 3, pp. 321–330, 2007.
- [17] X. Wang, X.-L. Li, B. Yang, and Z.-M. Guo, "Efficient generalized integer transform for reversible watermarking," *IEEE Signal Processing Letters*, vol. 17, no. 6, pp. 567–570, 2010.
- [18] L.-T. Ko, J.-E. Chen, Y.-S. Shieh, M. Scialia, and T.-Y. Sung, "A novel fractional-discrete-cosine-transform-based reversible

- watermarking for healthcare information management systems,” *Mathematical Problems in Engineering*, vol. 2012, Article ID 757018, 17 pages, 2012.
- [19] D. Coltuc, “Low distortion transform for reversible watermarking,” *IEEE Transactions on Image Processing*, vol. 21, no. 1, pp. 412–417, 2012.
- [20] S.-K. Lee, Y.-H. Suh, and Y.-S. Ho, “Reversible image authentication based on watermarking,” in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME '06)*, vol. 1–5, pp. 1321–1324, IEEE, Toronto, Canada, July 2006.
- [21] L. Kamstra and H. J. Heijmans, “Reversible data embedding into images using wavelet techniques and sorting,” *IEEE Transactions on Image Processing*, vol. 14, no. 12, pp. 2082–2090, 2005.
- [22] V. Sachnev, H. J. Kim, J. Nam, S. Suresh, and Y. Q. Shi, “Reversible watermarking algorithm using sorting and prediction,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 7, pp. 989–999, 2009.
- [23] D. M. Thodi and J. J. Rodriguez, “Expansion embedding techniques for reversible watermarking,” *IEEE Transactions on Image Processing*, vol. 16, no. 3, pp. 721–730, 2007.
- [24] X. L. Li, J. Li, B. Li, and B. Yang, “High-fidelity reversible data hiding scheme based on pixel-value-ordering and prediction-error expansion,” *Signal Processing*, vol. 93, no. 1, pp. 198–205, 2013.
- [25] B. Ou, X. L. Li, Y. Zhao, and R. R. Ni, “Reversible data hiding using invariant pixel-value-ordering and prediction-error expansion,” *Signal Processing: Image Communication*, vol. 29, no. 7, pp. 760–772, 2014.
- [26] F. Peng, X. L. Li, and B. Yang, “Improved PVO-based reversible data hiding,” *Digital Signal Processing*, vol. 25, no. 1, pp. 255–265, 2014.
- [27] X. Qu and H. J. Kim, “Pixel-based pixel value ordering predictor for high-fidelity reversible data hiding,” *Signal Processing*, vol. 111, pp. 249–260, 2015.
- [28] X. Li, B. Yang, and T. Y. Zeng, “Efficient reversible watermarking based on adaptive prediction-error expansion and pixel selection,” *IEEE Transactions on Image Processing*, vol. 20, no. 12, pp. 3524–3533, 2011.
- [29] C. Wang, X. Li, and B. Yang, “Efficient reversible image watermarking by using dynamical prediction-error expansion,” in *Proceedings of the 17th IEEE International Conference on Image Processing (ICIP '10)*, pp. 3673–3676, September 2010.
- [30] H. J. Hwang, H. J. Kim, V. Sachnev, and S. H. Joo, “Reversible watermarking method using optimal histogram pair shifting based on prediction and sorting,” *KSII Transactions on Internet and Information Systems*, vol. 4, no. 4, pp. 655–670, 2010.
- [31] H.-T. Wu and J. Huang, “Reversible image watermarking on prediction errors by efficient histogram modification,” *Signal Processing*, vol. 92, no. 12, pp. 3000–3009, 2012.
- [32] G. Xuan, X. Tong, J. Teng, X. Zhang, and Y. Q. Shi, “Optimal histogram-pair and prediction-error based image reversible data hiding,” in *Digital Forensics and Watermarking: 11th International Workshop, IWDW 2012, Shanghai, China, October 31–November 3, 2012, Revised Selected Papers*, vol. 7809 of *Lecture Notes in Computer Science*, pp. 368–383, Springer, Berlin, Germany, 2012.
- [33] L. X. Luo, Z. Y. Chen, M. Chen, X. Zeng, and Z. Xiong, “Reversible image watermarking using interpolation technique,” *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 1, pp. 187–193, 2010.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

