Hindawi Publishing Corporation
Journal of Applied Mathematics
Volume 2013, Article ID 189706, 8 pages
http://dx.doi.org/10.1155/2013/189706

Research Article

A Steganographic Method Based on Pixel-Value Differencing and the Perfect Square Number

Hsien-Wen Tseng[1] and Hui-Shih Leng[1,2]

[1] *Department of Information Management, Chaoyang University of Technology, No.168, Jifeng E. Road, Wufeng District, Taichung 41349, Taiwan*
[2] *Department of Mathematics, Changhua University of Education, No.1, Jin-De Road, Changhua County 50058, Taiwan*

Correspondence should be addressed to Hui-Shih Leng; lenghs@cc.ncue.edu.tw

Received 29 March 2013; Accepted 13 June 2013

Academic Editor: Carlos Conca

The pixel-value differencing (PVD) scheme uses the difference value between two consecutive pixels in a block to determine how many secret bits should be embedded. There are two types of the quantization range table in Wu and Tasi's method. The first was based on selecting the range widths of [8, 8, 16, 32, 64, 128], to provide large capacity. The second was based on selecting the range widths of [2, 2, 4, 4, 4, 8, 8, 16, 16, 32, 32, 64, 64], to provide high imperceptibility. Most of the related studies focus on increasing the capacity using LSB and the readjustment process, so their approach is too conformable to the LSB approach. There are very few studies focusing on the range table design. Besides, it is intuitive to design it by using the width of the power of two. This work designs a new quantization range table based on the perfect square number to decide the payload by the difference value between the consecutive pixels. Our research provides a new viewpoint that if we choose the proper width for each range and use the proposed method, we can obtain better image quantity and higher capacity. In addition, we offer a theoretical analysis to show our method is well defined. The experiment results also show the proposed scheme has better image quantity and higher capacity.

1. Introduction

The pixel-value differencing (PVD) [1] scheme provides high imperceptibility to the stego image by selecting two consecutive pixels and designs a quantization range table to determine the payload by the difference value between the consecutive pixels. Besides, it offers the advantage of conveying a large number of payloads, while still maintaining the consistency of an image characteristic after data embedding.

In recent years, several studies have been proposed to improve the PVD method. Wu et al.'s [2] presented a method combining pixel-value differencing and the LSB replacement method. Yang and Weng [3] proposed a multipixel differencing method that uses three difference values in a four-pixel block to determine how many secret bits should be embedded, and Jung et al.'s [4] proposed an image data hiding method based on multipixel differencing and LSB substitution. Liu and Shih [5] proposed two extensions of the PVD method, the block-based approach and Haar-based approach, and Yang et al. proposed an information hiding technique based on blocked PVD. Liao et al.'s [6] proposed a four-pixel differencing and modified LSB substitution, and Yang et al.'s [7] proposed a data hiding scheme using the pixel-value differencing in multimedia images.

Some studies focused on increasing the capacity [3, 5, 8] using LSB [2, 4] or a readjusted process [6, 7] to improve the embedding capacity or image quantity. Few studies focus on the range table design. Besides, it is intuitive to design it using the width of the power of two.

In this work, we design a new quantization range table based on the perfect square number to decide the payload by the difference value between the consecutive pixels. It differs from the design of Wu and Tsai's scheme, in which the quantization range table is based on the range width of the power of two. The perfect square number provides an elegant mathematical model to develop a new quantization range table, which divides each range into two subranges for embedding different numbers of secret bits.

The remainder of this paper is organized as follows. Section 2 briefly describes Wu and Tsai's PVD approach. Section 3 presents our scheme on how to create a new quantization table based on the perfect square number, how the embedding procedure works, and how to extract the secret data from the stego image. Section 4 offers a theoretical analysis and shows the experiment results. Finally, Section 5 concludes this paper.

2. Review of Wu and Tsai's PVD Approach

The gray-valued cover image is partitioned into nonoverlapping blocks of two consecutive pixels, states p_i and p_{i+1}. From each block, we can obtain a difference value $d_i = |p_i - p_{i+1}|$; then d_i ranges from 0 to 255. If d_i is small, then the block is located within the smooth area and will embed less secret data. Otherwise, it is located on the edge area, and it can embed a greater amount of secret data. The quantization range table is designed with n contiguous ranges, and the range table ranges from 0 to 255. The number of secret bits hidden in two consecutive pixels depends on the quantization range table.

The embedding algorithm is described as follows.

Step 1. Calculate the difference $d_i = |p_i - p_{i+1}|$ for each block of two consecutive pixels p_i and p_{i+1}.

Step 2. Search the quantization range table for d_i to determine how many bits will be embedded. Obtain the range R_i in which $R_i = [l_i, u_i]$, where l_i and u_i are the lower bound and the upper bound of R_i, and $m = \lfloor \log_2(u_i - l_i) \rfloor$ is the number of embedding bits.

Step 3. Read m secret bits from the secret bit stream, and transform it into decimal value b.

Step 4. Calculate the new difference $d_i' = l_i + b$. Ensure both d_i and d_i' are in the same range R_i.

Step 5. Average d_i' to p_i and p_{i+1}. The new pixel values p_i' and p_{i+1}' are obtained by the following formula:

$$(p_i', p_{i+1}') = \begin{cases} \left(p_i + \left\lceil \dfrac{|d_i' - d_i|}{2} \right\rceil, p_{i+1} - \left\lfloor \dfrac{|d_i' - d_i|}{2} \right\rfloor\right), & \text{if } p_i \geq p_{i+1},\ d_i' > d_i, \\ \left(p_i - \left\lceil \dfrac{|d_i' - d_i|}{2} \right\rceil, p_{i+1} + \left\lfloor \dfrac{|d_i' - d_i|}{2} \right\rfloor\right), & \text{if } p_i < p_{i+1},\ d_i' > d_i, \\ \left(p_i - \left\lceil \dfrac{|d_i' - d_i|}{2} \right\rceil, p_{i+1} + \left\lfloor \dfrac{|d_i' - d_i|}{2} \right\rfloor\right), & \text{if } p_i \geq p_{i+1},\ d_i' \leq d_i, \\ \left(p_i + \left\lceil \dfrac{|d_i' - d_i|}{2} \right\rceil, p_{i+1} - \left\lfloor \dfrac{|d_i' - d_i|}{2} \right\rfloor\right), & \text{if } p_i < p_{i+1},\ d_i' \leq d_i. \end{cases} \quad (1)$$

Repeat Steps 1–5 until all secret bits are embedded and the stego image is produced.

In the extracting phase, the same Steps 1 and 2 in the embedding algorithm are used. The difference $d_i' = |p_i' - p_{i+1}'|$ is computed for each two consecutive pixels in the stego image, and then the same quantization range table is searched to find l_i. Compute $b = d_i' - l_i$, and transform b into the binary stream. Repeat until all secret data is completely extracted.

3. Proposed Scheme

In this section, the proposed scheme is described in three parts: the new quantization range table is based on the perfect square number, embedding procedure, and extraction procedure.

3.1. The New Quantization Range Table Based on Perfect Square Number. We design a new quantization range table based on the perfect square number in Table 1. For each pixel value $p \in [0, 255]$, choose the nearest perfect square number n^2 (we will define the nearest perfect square number later), then we have range $n^2 - n \leq n^2 < n^2 + n$ for $n \in [1, 16]$. The width of this range is $(n^2 + n) - (n^2 - n) = 2n$, and the embedding bit length is $m = \lfloor \log_2 2n \rfloor$. For each range $[n^2 - n, n^2 + n)$, if the width of this range is larger than 2^m, then we divide this range into two subranges: $[n^2 - n, n^2 + n - 2^m]$ and $[n^2 + n - 2^m + 1, n^2 + n - 1]$.

For example, if the pixel value is 34, the nearest perfect square number is 36; then we have range: $[30, 41]$. The width of this range is 12, and the embedding bit length is $m = \lfloor \log_2 12 \rfloor = 3$. Since $12 > 2^3$, divide this range into two subranges $[30, 33]$ and $[34, 41]$.

By the definition of subranges, if the to-be-embedded $m+1$ secret bits equal one of the $m + 1$ LSB bits in the first subrange, then we claim it can embed $m + 1$ secret bits. Otherwise, the second subrange's width is always 2^m, and it can embed m secret bits. Therefore, we can guarantee one of the continuous series numbers equals the m bits secret data which we want to embed.

There are two important concepts we want to emphasize here. First, if the difference value is located in the first subrange, there is no modification needed, so this design does not violate the basic concept of PVD and HVS (Human Visual System). Second, we notice almost the difference values belonging to range $[56, 255]$ are used to embed the same size of data, 4 bits of secret data. Our design in Table 1 still coincides with the basic concept of PVD—embedding a lower amount of secret data in the smooth area and a greater amount of secret data in the edge area.

3.2. Embedding Procedure. Before embedding secret data, the function *Nearest_PerfectSquare*(d) is defined to find the nearest perfect square number for difference value d, where d is the difference value of two consecutive pixels.

The function *Nearest_PerfectSquare*(d) returns the nearest perfect square number n, and n is the range number of d. According to range number n, the secret data is embedded into the cover image by the embedding procedure. The

Table 1: The quantization range table based on the perfect square number.

n	Range	Sub-ranges	m
1	$[0,1]$	$[0,1]$	1
2	$[2,5]$	$[2,5]$	2
3	$[6,11]$	$[6,7]$	3
		$[8,11]$	2
4	$[12,19]$	$[12,19]$	3
5	$[20,29]$	$[20,21]$	4
		$[22,29]$	3
6	$[30,41]$	$[30,33]$	4
		$[34,41]$	3
7	$[42,55]$	$[42,47]$	4
		$[48,55]$	3
8	$[56,71]$	$[56,71]$	4
9	$[72,89]$	$[72,73]$	5
		$[74,89]$	4
10	$[90,109]$	$[90,93]$	5
		$[94,109]$	4
11	$[110,131]$	$[110,115]$	5
		$[116,131]$	4
12	$[132,155]$	$[132,139]$	5
		$[140,155]$	4
13	$[156,181]$	$[156,165]$	5
		$[166,181]$	4
14	$[182,209]$	$[182,193]$	5
		$[194,209]$	4
15	$[210,239]$	$[210,223]$	5
		$[224,239]$	4
16	$[240,255]$	$[240,255]$	4

n is the perfect square number; m is the length of embedding bits.

embedding procedure of proposed method is summarized as follows.

The proposed embedding procedure is as follows.

Input. The grayscale cover image pixel value $Gray(i)$, where i is a pixel index. $LSB(p,m)$ is m bits LSB binary stream for pixel value p. $Secret(m)$ represents m bits binary secret data.

Output. The grayscale stego image pixel value $Stego(i)$.

Step 1. For each pair of two consecutive pixels, compute the difference value $d = |(Gray(i+1) - Gray(i)|$.

Step 2. Find the nearest perfect square number n by function $Nearest\_PerfectSquare(d)$, and n is the range number of d in Table 1.

Step 3. If $d \geq 240$, set new pixel-value difference value $d' = 240 + Secret(4)$. According to PVD embedding scheme (Step 5), average d' to $Gray(i)$ and $Gray(i+1)$. The new pixel values $Stego(i)$ and $Stego(i+1)$ are obtained by the following formula:

$$(Stego(i), Stego(i+1)) = \begin{cases} \left(Gray(i) + \left\lceil \dfrac{|d'-d|}{2} \right\rceil, Gray(i+1) - \left\lfloor \dfrac{|d'-d|}{2} \right\rfloor\right), & \text{if } Gray(i) \geq Gray(i+1),\ d' > d, \\ \left(Gray(i) - \left\lceil \dfrac{|d'-d|}{2} \right\rceil, Gray(i+1) + \left\lfloor \dfrac{|d'-d|}{2} \right\rfloor\right), & \text{if } Gray(i) < Gray(i+1),\ d' > d, \\ \left(Gray(i) - \left\lceil \dfrac{|d'-d|}{2} \right\rceil, Gray(i+1) + \left\lfloor \dfrac{|d'-d|}{2} \right\rfloor\right), & \text{if } Gray(i) \geq Gray(i+1),\ d' \leq d, \\ \left(Gray(i) + \left\lceil \dfrac{|d'-d|}{2} \right\rceil, Gray(i+1) - \left\lfloor \dfrac{|d'-d|}{2} \right\rfloor\right), & \text{if } Gray(i) < Gray(i+1),\ d' \leq d. \end{cases} \quad (2)$$

Step 4. If $d < 240$, compute the length of embedding bits $m = \lfloor \log_2(2n) \rfloor$. There are two cases. Search the first subrange $[n^2 - n, n^2 + n - 2^m - 1]$ and find a value p in the subrange such that $LSB(p, m+1) = Secret(m+1)$, and then set $d' = p$. Otherwise, search the second subrange $[n^2 + n - 2^m, n^2 + n - 1]$ and find a value p in the subrange such that $LSB(p,m) = Secret(m)$, and then set $d' = p$. Finally, average d' to $Gray(i+1)$ and $Gray(i)$ as Step 3 does, and then we obtain $Stego(i)$ and $Stego(i+1)$.

We illustrate the embedding procedure in Figure 1.

For example, we choose a pair of two consecutive pixels (47,81) from the cover image; then $d = |81 - 47| = 34$, $n = 6$. The following two conditions are discussed.

Case 1. Secret data is "1110" or "1111" or "0000" or "0001."

In the first subrange $[30,33]$, the 4 LSBs are the same as the secret data ($30 = 0001\mathbf{1110}_2$, $31 = 0001\mathbf{1111}_2$, $32 = 0010\mathbf{0000}_2$, $33 = 0010\mathbf{0001}_2$), so we can embed 4 bits of secret data. Suppose the to-be-embedded bits are "0000"; then the new difference value $d' = 32$ ($32 = 0010\mathbf{0000}_2$). Finally, we modify (47,81) to (48,80).

Case 2. Otherwise, the second subrange $[34,41]$ is used to embed 3 secret bits ranging from "000" to "111". Suppose the to-be-embedded bits are "001"; then the new difference value $d' = 41$ ($41 = 00101\mathbf{001}_2$). Finally, we modify (47,81) to (43,84).

We illustrated the embedding examples in Figure 2.

3.3. Extraction Procedure. The extraction procedure of the proposed method is summarized as follows.

The proposed extraction procedure is as follows.

Input. The grayscale stego image pixel value $Stego(i)$. $LSB(p,m)$ is a decimal number transform from m bits LSB binary stream for pixel value p. $Secret(m)$ represents m bits binary secret data.

Output. Secret data.

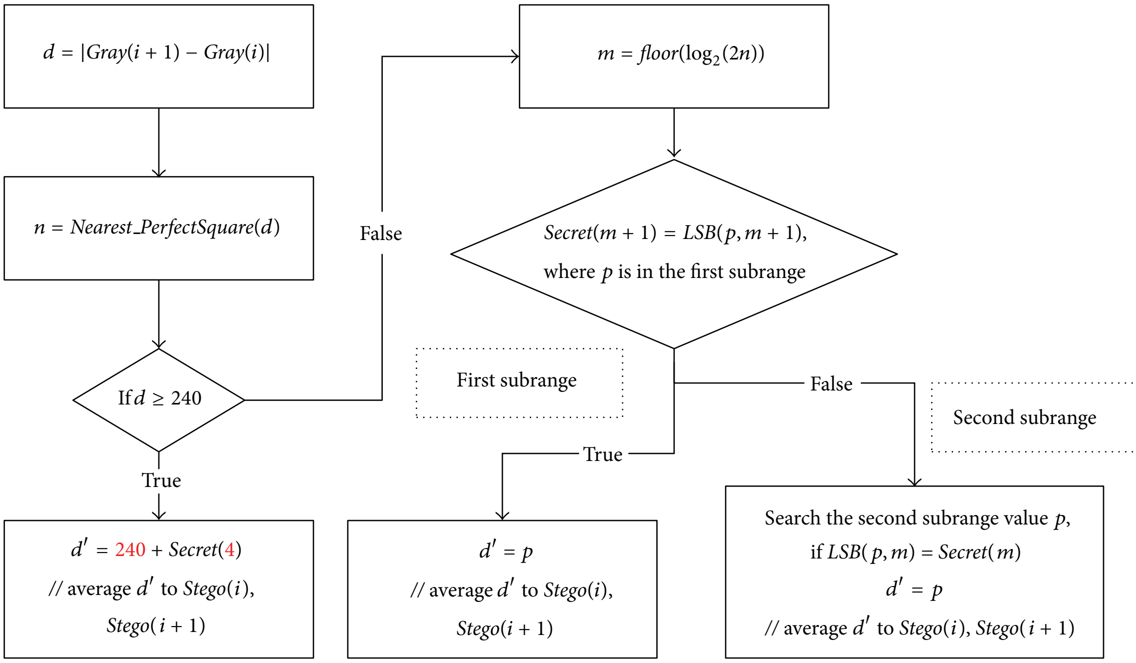

FIGURE 1: The embedding procedure.

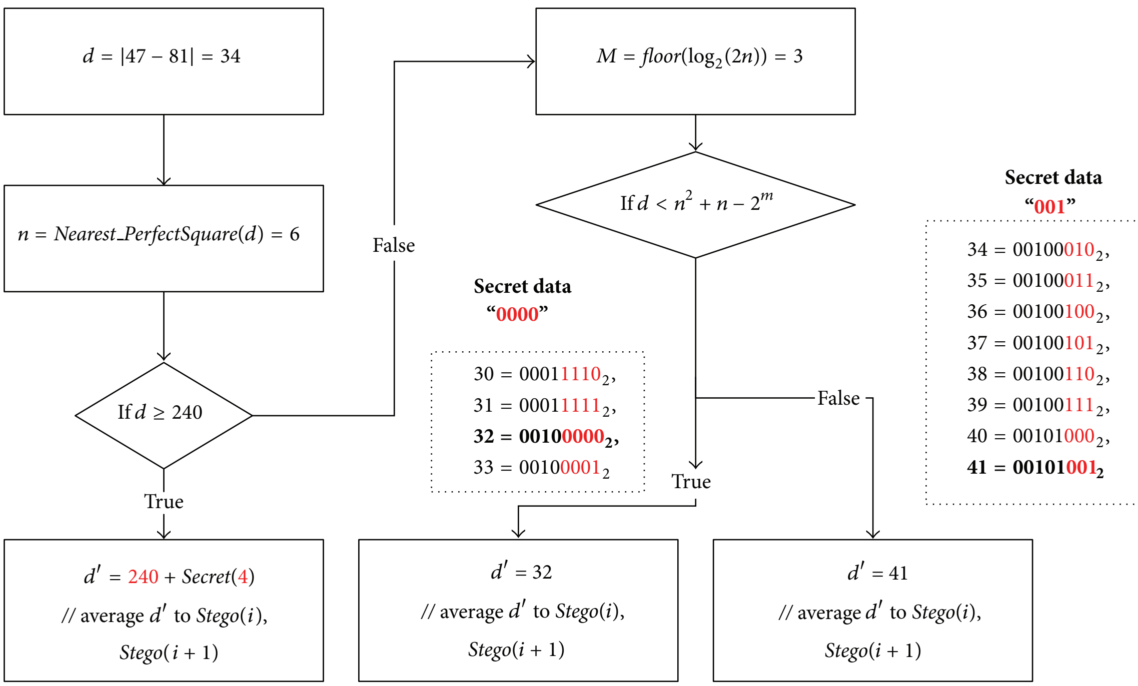

FIGURE 2: The embedding examples of two consecutive pixels (47,81) and two cases of secret data "0000" and "001".

Step 1. For each pair of two consecutive pixels, compute the difference value $d' = |Stego(i+1) - Stego(i)|$.

Step 2. Find the nearest perfect square number n by function $Nearest\_PerfectSquare(d')$, and n is the range number of d' in Table 1.

Step 3. If $d' \geq 240$, $Secret(4) = LSB(d', 4)$. Otherwise, compute the length of embedding bits $m = \lfloor \log_2(2n) \rfloor$, search subrange n from Table 1 to determine which subrange it belongs to, and extract the secret data ($Secret(m+1) = LSB(d', m+1)$ for the first subrange and $Secret(m) = LSB(d', m)$ for the second subrange). Finally, we extract all secret data.

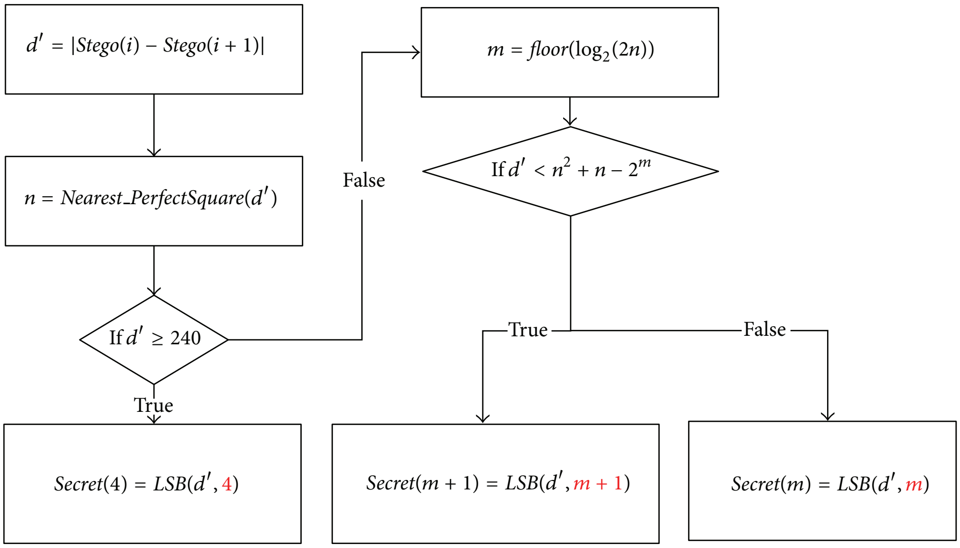

Figure 3: The extraction procedure.

We illustrate the extraction procedure in Figure 3.

For example, we choose a pair of two consecutive pixels (48,80) from the stego image; then $d' = |80 - 48| = 32$, $n = 6$. Since d' is located in the first subrange, we can extract the secret data "**0000**" from 4 LSBs of 32 = $0010\mathbf{0000}_2$.

For another example, we choose a pair of two consecutive pixels (43,84) from the stego image; then $d' = |84 - 43| = 41$, $n = 6$. Since d' is located in the second subrange, we can extract the secret data "**001**" from 3 LSBs of 41 = $00101\mathbf{001}_2$.

4. Theoretical Analysis and Experiment Results

Lena, Baboon, Peppers, Jet, SailBoat, and Tiffany from the SIPI Image Database are chosen as the cover images. First, we give a theoretical analysis to show our method is well defined, and then the experiment results show the proposed scheme has higher imperceptibility.

4.1. Theoretical Analysis. The stego image quality is measured by the peak signal-to-noise ratio (PSNR). The PSNR formula is defined as

$$\text{PSNR} = 10 \times \log_{10} \frac{255^2}{\text{MSE}} \text{ (dB)}, \quad (3)$$

where MSE is the mean square error between the cover and stego images. For a cover image, whose width and height are w and h, MSE is defined as

$$\text{MSE} = \frac{1}{w \times h} \sum_{i=1}^{w} \sum_{j}^{h} \left(Stego(i,j) - Cover(i,j) \right)^2, \quad (4)$$

where $Stego(i, j)$ and $Cover(i, j)$ are the pixel values of the stego and cover images, respectively.

Suppose, the probability of distribution is uniform. Then we calculate the average payload and average MSE for each range $R_i (i = 1 \sim 16)$ (or the perfect square number n) according to Table 1. The average payload is computed by the following formula:

$$\text{average payload} = \frac{C(n_{i,1})}{2n} \times n_{i,1} + \frac{C(n_{i,2})}{2n} \times n_{i,2}, \quad (5)$$

where $C(n_{i,1})$ is the total number in the first subrange, $n_{i,1}$ is the embedding bits in the first subrange, $C(n_{i,2})$ is the total number in the second subrange, and $n_{i,2}$ is the embedding bits in the second subrange. The average error for each range is calculated by the following formula:

$$\text{average error}_i = \frac{C(n_{i,1})}{2n} \times 0 + \frac{C(n_{i,2})}{2n} \times \frac{\sum_{i=0}^{C(n_{i,2})-1} i}{C(n_{i,2})}. \quad (6)$$

For example, $n = 6$, average payload is $(4/12) \times 4 + (8/12) \times 3 = 3.33$, and the average error is $(4/12) \times 0 + (8/12) \times (\sum_{i=0}^{7} i)/8 = 2.33$.

The total MSE is estimated by

$$\text{MSE} = \sum_{i=1}^{16} |n_i| \times (\text{average error}_i)^2, \quad (7)$$

where $|n_i|$ and average error_i are the width and average error for each range.

Therefore, we obtain the average payload and average MSE using the perfect square number, as illustrated in Table 2.

The same theoretical analysis using Wu and Tsai's method is shown in Table 3.

From Tables 2 and 3, we calculate the payload and PNSR for Lena, Baboon, Peppers, Jet, SailBoat, and Tiffany and compare Wu and Tsai's method and the proposed method by theoretical analysis in Table 4.

Our method clearly has greater capacity and higher PSNR than Wu and Tsai's method, which proves the proposed method is well defined.

Table 2: Distributions of pixel-value difference, average payload, and average MSE for images using the proposed method.

R_i	Lena	Baboon	Peppers	Jet	SailBoat	Tiffany	Average payload	Average MSE
1	79276	72714	77375	90611	77179	83887	1	0.5
2	28604	14659	27386	17330	21513	28642	2	1.5
3	13861	15316	17509	11449	16168	10717	2.33	1
4	5013	11053	5467	3894	8215	3953	3	3.5
5	2275	7539	1593	2026	3787	1768	3.2	2.8
6	1128	4754	726	2049	2104	897	3.33	2.33
7	566	2841	353	1138	1127	400	3.43	2
8	224	1466	195	1259	577	535	4	7.5
9	100	561	150	586	270	170	4.11	6.67
10	21	149	109	281	58	87	4.2	6
11	4	18	57	408	19	8	4.27	5.45
12	0	2	51	41	22	7	4.33	5
13	0	0	92	0	33	0	4.38	4.62
14	0	0	9	0	0	1	4.43	4.29
15	0	0	0	0	0	0	4.47	4
16	0	0	0	0	0	0	4	7.5

Table 3: The distributions of pixel-value difference, payload, and MSE for images using Wu and Tsai's method.

Range width	Lena	Baboon	Peppers	Jet	SailBoat	Tiffany	Payload	MSE
2	79276	72714	77375	90611	77179	83887	1	0.5
2	17302	7786	15376	17330	12301	18372	1	0.5
4	18177	12904	20501	11449	16226	15669	2	1.5
4	6986	9285	9018	3894	9154	5318	2	1.5
4	3266	6437	3786	2026	5174	2556	2	1.5
8	2889	8147	2574	2049	4944	2354	3	3.5
8	1392	5010	869	1138	2378	1040	3	3.5
16	1174	5201	733	1259	2200	885	4	7.5
16	385	2288	292	586	907	386	4	7.5
32	214	1207	273	281	504	551	5	16
32	11	91	120	408	46	45	5	16
64	0	2	155	41	59	8	6	32
64	0	0	0	0	0	1	6	32

Table 4: Comparison between Wu and Tsai's method and the proposed method by theoretical analysis.

Cover images (512 × 512)	Wu and Tsai's method with the range widths of 2, 2, 4, 4, 4, 8, 8, 16, 16, 32, 32, 64, and 64			Proposed method		
	Capacity	Payload	PSNR	Capacity	Payload	PSNR
Lana	173640	0.66	47.75	198209	0.76	49.23
Baboon	213681	0.82	42.31	229459	0.88	46.17
Peppers	176685	0.67	45.71	200831	0.77	49.10
Jet	163312	0.62	46.01	191383	0.73	47.96
SailBoat	188086	0.72	44.86	209494	0.80	47.87
Tiffany	167645	0.64	46.83	191290	0.73	49.18

Figure 4: Six test images: (a) Lena, (b) Baboon, (c) Peppers, (d) Jet, (e) SailBoat, and (f) Tiffany.

Table 5: The experiment results use Figure 4 as the cover image.

Cover images (512 × 512)	Capacity (bits)	Payload (bpp)	PSNR (dB)
Lena	215740	0.82	50.70
Baboon	241719	0.92	48.57
Peppers	217290	0.83	50.57
Jet	204682	0.78	50.89
SailBoat	224915	0.86	49.86
Tiffany	210935	0.80	50.86

4.2. Experiment Results. We also use the same test images as the real test shown in Figure 4, and the experiment results are shown in Table 5.

From Table 5, we found the experiment results have larger capacity and better PSNR than those of the theoretical analysis. The capacity and PSNR seem to be affected by the secret data, with more pixel-value difference falling in the first subranges and matching the secret data; we can obtain more capacities and less distortion.

5. Conclusions

This work designs a new quantization range table based on the perfect square number. In particular, we propose a new technology to design the range table. The width of the range is no longer a power of two, and if the difference value is located in the first subrange, there is no modification needed. Therefore, this design has not violated the basic concept of PVD and HVS (Human Visual System). If we choose a proper width for each range and use the proposed method as mentioned above, we can obtain better image quantity and higher capacity. The theoretical analysis shows the proposed scheme is well defined and has larger capacity and higher PSNR than those of Wu and Tsai's second type range table design. The experiment results also show the proposed scheme provides large capacity and high imperceptibility. In addition, our study ingeniously uses the perfect square number to achieve the goal.

References

[1] D.-C. Wu and W.-H. Tsai, "A steganographic method for images by pixel-value differencing," *Pattern Recognition Letters*, vol. 24, no. 9-10, pp. 1613–1626, 2003.

[2] H. C. Wu, N. I. Wu, C. S. Tsai, and M. S. Huang, "Image steganographic scheme based on pixel-value differencing and LSB replacement methods," *IEE Proceedings Vision Image and Signal Processing*, vol. 152, no. 5, pp. 611–615, 2005.

[3] C. H. Yang and C. Y. Weng, "A steganographic method for digital images by multi-pixel differencing," in *Proceedings of International Computer Symposium*, pp. 831–836, Taipei, Taiwan, 2006.

[4] K.-H. Jung, K.-J. Ha, and K.-Y. Yoo, "Image data hiding method based on multi-pixel differencing and LSB substitution methods," in *Proceedings of International Conference on Convergence and Hybrid Information Technology (ICHIT '08)*, pp. 355–358, kor, August 2008.

[5] J.-C. Liu and M.-H. Shih, "Generalizations of pixel-value differencing steganography for data hiding in images," *Fundamenta Informaticae*, vol. 83, no. 3, pp. 319–335, 2008.

[6] X. Liao, Q.-Y. Wen, and J. Zhang, "A steganographic method for digital images with four-pixel differencing and modified LSB substitution," *Journal of Visual Communication and Image Representation*, vol. 22, no. 1, pp. 1–8, 2011.

[7] C.-H. Yang, C.-Y. Weng, H.-K. Tso, and S.-J. Wang, "A data hiding scheme using the varieties of pixel-value differencing in multimedia images," *Journal of Systems and Software*, vol. 84, no. 4, pp. 669–678, 2011.

[8] C. H. Yang, S. J. Wang, C. Y. Weng, and H. M. Sun, "Information hiding technique based on blocked PVD," *Journal of Information Management*, vol. 15, no. 3, pp. 29–48, 2008.