A FPGA Implementation of Dual Images based Reversible Data Hiding Technique using LSB Matching with Pipelining

S. Raveendra Reddy* and S. M. Sakthivel

Department of School of Electronics Engineering, VIT University, Chennai - 600 127, Tamil Nadu, India; Sannapreddy.raveendra2013@vit.ac.in, sakthivel.sm@vit.ac.in

Abstract

Background: In this digital era, the authentication and proof of ownership has become a vital part in all the multimedia data content like audio, image and video. Data Hiding is one of the familiar methodologies used to authenticate and resolve the issues of copyrights of the digital data. Methods: In this paper the FPGA (Field Programmable Gate Array) implementation of Data Hiding using reversible Dual Image concept is carried out on a gray scale image. Here the FPGA implementation is carried out with and without the concept of pipelining. Findings: In the data hiding process the secret key is embedded in the host image content and analyzed with the values of PSNR (Peak Signal to Noise Ratio) and Embedding capacity. In the last, a comparison for the pipelined and non pipelined mode of data hiding process has been done for the values of area, power and timing. From the results it is noted that the pipelined mode of data hiding gives better result in terms of very less embedding time compared to non pipelined mode with a lesser power consumption. As this data hiding methodology involves only simple operation it is easy to implement as FPGA chip using Verilog HDL Modelling language. Here the entire data hiding operation is carried out by the hardware chip, not by software running on hardware, hence the process of data hiding is fast when compared with all other software implementations. As the whole process of embedding is taking place in real time, we can embed this FPGA data hiding chip as a separate co-processor the data hiding operation with any multimedia device.

Keywords: Dual Imaging Concept, FPGA based Data Hiding, LSB Bit Matching, Pipelined Embedding, Reversible Data Hiding

1. Introduction

The originality of the digital data is altered during the process of transmission and sharing over internet\(^1\). Hence there exists a problem in sending the data in an imperceptible manner\(^2\). Data hiding is one of the methods to hide the secret data in the host content and transmit in an effective manner\(^3\). The digital data can be an image, audio or video\(^4,5\). Data hiding is the process of embedding a confidential message. The data hiding process is said to be reversible if the original data can be reconstructed by reversing the algorithm. In this paper the host is considered as a gray scale image and watermark as a sequence of binary data bits.

There are many data hiding techniques available nowadays, dual image hiding method is one of the technique. In dual image hiding method confidential message hidden exclusively in images. The embedding of secret information should be in a manner such that the existence of the information should be invisible. This method helps in avoiding loss of data or copy of data by unauthorized person. After inserting the confidential message it is referred as stego image. The role of stego key is to extract the embedding data in hiding process\(^6\).

The combined LSB matching technique and Dual imaging base data hiding is similar to the methodology of covert communication where people can communicate secretly\(^7\).

*Author for correspondence
All the data hiding algorithms till date are of software implementation, only a few are in hardware based approach. If the data hiding operation is carried out by the Field Programmable Gate Array (FPGA) Chip, the process is quite fast and area, power and timing optimization is possible due to the presence of custom circuitry. The FPGA based data hiding is a real time implementation whereas the software based approaches are doing the data hiding in offline mode of processing.

In this paper reversible data hiding using dual imaging concept with and without has been implemented and its computational efficiency is studied. The relative advantages and disadvantages of the approaches used is presented. The LSB matching method and dual images hiding method algorithms are implemented using MATLAB to illustrate the characteristics of the data hiding techniques such as embedding capacity and peak signal to noise ratio. Finally the entire algorithm is designed using verilog HDL and synthesized using Xilinx ISE.

2. LSB Matching

This Section explains the process of data hiding in a gray scale image using the concept of LSB Matching technique. In the normal LSB embedding method simply the LSB bits of the host content are replaced directly whereas the LSB Matching based data embedding involves the checking of the four conditions and finally the secret data signature is hidden in the host content. These LSB matching method give message in terms of bits and input pixels in terms of decimal values. Instead of inserting the digital signature directly in the binary position as in LSB embedding, here the secret signature insertion is carried out by the comparison with two created of copies of host image.

In this LSB matching based data hiding process the two pixel pairs are selected for the secret data embedding and checked for the four below stated conditions. The same process is repeated for the entire pairs of the pixels in the host image. In case of general LSB embedding technique the embedding is carried out by direct pixel replacement here it is processed by four checking process and then embedded.

Hence this methodology will preserve the host image quality when compared to ordinary LSB embedding technique. The entire LSB matching based data hiding process is represented using a flow chart form in the Figure 1.

The entire process of data hiding is explained here with an example sample pair of pixels. Consider the pixel pairs

Input \( i,j, k_{01}, p_1, p_2 \)

\[ F(s_{(i,j)}, s_{(i,j+1)}) = \text{LSB} \left( \frac{s_{(i,j)}}{2} + s_{(i,j+1)} \right) \]  

3. Dual Images Hiding Method

In this technique of the LSB matching method dual image hiding method of camouflage pixel modification is used and in order to enhance the image quality the rule table is used

3.1 Embedding Phase

First the original image \( S = \{s_{(0,0)}, s_{(0,1)}, \ldots, s_{(h-1,w-1)}\} \) is duplicated into two identical images \( q_1 = \{q_{1(0,0)}, q_{1(0,1)}, \ldots, q_{1(h-1,w-1)}\}, q_2 = \{q_{2(0,0)}, q_{2(0,1)}, \ldots, q_{2(h-1,w-1)}\} \). Through the LSB matching method, using each two image pixel values \( q_{1(i,j)} \) and \( q_{2(i,j)} \) as a set, we embed all sets to obtain \( S' = \{s'_{(0,0)}, s'_{(0,1)}, \ldots, s'_{(h-1,w-1)}\} \) and \( S'' = \{s''_{(0,0)}, s''_{(0,1)}, \ldots, s''_{(h-1,w-1)}\} \). Subsequently, through the averaging method

\[ \text{Yes} \]

\[ \text{No} \]

\[ s'_{(i,j)} = s_{(i,j)} + s''_{(i,j)} \]

\[ \frac{1}{2} \]

\[ \frac{1}{2} \]

\[ s'_{(i,j+1)} = s''_{(i,j+1)} \]

\[ s''_{(i,j+1)} \]

\[ s''_{(i,j)} \]

Figure 1. Strategy of data hiding.

\( S_{(0,0)} = 49, S_{(0,1)} = 44 \) and confidential information \( P = (10)^2 \) as an example. When \( S_{(0,0)} = 49 \) and \( S_{(0,1)} = 1 \), then \( S_{(0,0)} = 49 \) it is substituted in Equation 1 to give \( F(49,44) = 0 \). The Case-1 condition are met, and this results in modified pixels as \( S'_{(0,0)} = 49 \) and \( S'_{(0,1)} = 44 \). In the same way the four cases are iteratively verified for data hiding row wise and column wise of the image until all the pixels pairs are processed.
Conversely, if the averaged pixel values of the two images are different from the original pixels, we use the rule table to modify the camouflage pixels. The detailed process is described as follows as shown in Figure 2.

- The original image \( S = \{s_{(0,0)}, s_{(0,1)}, \ldots, s_{(h-1,w-1)}\}\) is duplicated into two identical images \( q_1 = \{q_{1(0,0)}, q_{1(0,1)}, \ldots, q_{1(h-1,w-1)}\}\) and \( q_2 = \{q_{2(0,0)}, q_{2(0,1)}, \ldots, q_{2(h-1,w-1)}\}\).

- Through the LSB matching method, we embed \( q_{1(i,j)}^1 q_{1(i,j+1)}^1\) and \( q_{2(i,j)}^1 q_{2(i,j+1)}^1\) get the pixel modification capacity. This is to say \( q_{1(i,j)}^1 q_{1(i,j+1)}^1\) is used to embed confidential message \( p_1\) and \( p_2\), and \( q_{2(i,j)}^1 q_{2(i,j+1)}^1\) is used to embed confidential message \( p_3\) and \( p_4\).

Using the LSB matching method, we first obtain the pixel modified positions and the modified capacity, and we simultaneously hide two pairs. The situation can be summarized as shown in Table 1.

Here ‘0’ means two pairs are hidden, without modifying the pixel values; +1 indicates that the pixel value increases by one. Similarly –1 indicates that the pixel value decreases by 1. Consider case-3 as an example. The first pair \( q_{1(i,j)}^1\) is set 0; and the second pair \( q_{2(i,j)}^1\) is also set to -1. Since the proposed method uses the averaging method to calculate the average of the two camouflage pixels, it takes lower limit to restore the original pixel values. If the two camouflage pixels are simultaneously +2 or +1 or -1, the recovered pixel values obtained through the averaging method will differ from original pixels. Consider case-3 as an example. Take average of \( q_{1(i,j)}^1\) and \( q_{2(i,j)}^1\), the obtained value is higher than original pixel value by 1, which is different from the original pixel value. Furthermore, there is another case in which the pixel values are +1. Using the averaging method and taking the lower limit, the obtained value is less than the original pixel value by 1. As seen in Table 1, we cannot restore the pixels to their original value in seven cases, as case-3, case-6, case-7, case-9, case-10, case-11 and case-16. To deal with these cases, we modify the rules in the table to avoid cases where in the image cannot be restored.

- We check if the embedded pixel values can be restored using the averaging method. The formula is as follows:

\[
\left\lfloor \frac{s'_{(i,j)} + s''_{(i,j)}}{2} \right\rfloor,
\]

(2)

\[
\chi_{(i,j)} = \left\lfloor \frac{s'_{(i,j)} + s''_{(i,j)}}{2} \right\rfloor,
\]

a) if \( s_{(i,j)} = s'_{(i,j)}\) and \( s_{(i,j)} = s''_{(i,j)}\), the pixel values after embedding \( s_{(i,j)}, s_{(i,j)}^{(i,j+1)}, s_{(i,j)}^{(i,j+1)}\) and \( s_{(i,j)}^{(i,j+1)}\) are camouflage pixels. We go back to step (2) for the next embedding.

b) If \( s_{(i,j)} \neq s'_{(i,j)}\) or \( s_{(i,j)} \neq s''_{(i,j)}\), the original pixels \( q_{1(i,j)}^1 q_{1(i,j+1)}^1\) and \( q_{2(i,j)}^1 q_{2(i,j+1)}^1\) will be modified to obtain the camouflage pixels using the modification rule table. The modifying rules are shown in Table 2. Let us take rule 2 of Table 2 as an example. The original modification rule of rule 2, which is case-6 of Table 1, is \( q_{1(i,j)}^1\) and the other copies are unchanged. However,
3.2 Information Extraction and Image Restoration Phase

In this phase, the extraction and the image recovery can be carried out separately. The method of information extraction is similar to the LSB matching method. First the camouflage pixel pair values \( s'(i,j) \) and \( s'(i,j+1) \) for the are obtained for the first image. Using the LSB equation we directly obtain the first confidential message \( p_1 \) from \( s'(i,j) \). Using Eq. (1), and substituting \( s'(i,j) \) and \( s'(i,j+1) \) into \( F(.) \) we will get the second confidential message \( p_2 \). Similarly from the second camouflage image pixel pair \( s''(i,j) \) and \( s''(i,j+1) \) we derive information \( p_3 \) and \( p_4 \). For image restoration, we can calculate the identical index location pixel average value of the two camouflage images using Equation 2) to recover the original image. An information extraction and image restoration example is shown in Figures 3 & 4.

3.3 Pipelining

In the pipelined architecture the above stated process are done in a queue based mode. In the first phase the two pixel pairs are fetched and then the two pairs are evaluated for the four cases in the second phase. While during the evaluation of pixel pairs the next two pixels pairs are fetched. In the same manner at the third phase the secret signature is embedded, likewise the fetch, evaluate and embedded phase are pipelined to reduce the time of embedding with a less power consumption, compared with the unpipelined mode. The pipelined architecture is shown in the Figure 5. Here F is Fetching, EV is Evaluation and EM is Embedding.

4. Results and Discussions

The verification of Dual image hiding method effectively hide messages using two camouflaged images. In this the assessment of the pixel gap between camouflage images and original image is carried out using the values of the Peak Signal to Noise Ratio (PSNR). The formula is as follows:

\[
PSNR = 10 \times \log_{10} \left[ \frac{2^{2M} - 1}{\left( \frac{1}{m \times n} \right) \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \left( s(i,j) - s'_{(i,j)} \right)^2} \right] (dB) \tag{3}
\]

| Rule | Case | Pixel value modification statuses | The final modified camouflage pixel values |
|------|------|----------------------------------|-------------------------------------------|
| 1    | 3    | 0 0 -1 0                        | \( q_1(i,j) = 2, q_1(i,j+1) = 1, q_2(i,j) = -1, q_2(i,j+1) = 2 \) |
| 2    | 6    | 0 1 0 1                         | \( q_1(i,j) = 2, q_1(i,j+1) = 1, q_2(i,j) = 2, q_2(i,j+1) = 1 \) |
| 3    | 7    | 0 1 -1 0                        | \( q_1(i,j) = 2, q_1(i,j+1) = 1, q_2(i,j) = -1, q_2(i,j+1) = 2 \) |
| 4    | 9    | -1 0 0 0                        | \( q_1(i,j) = -1, q_1(i,j+1) = 1, q_2(i,j) = 1, q_2(i,j+1) = 1 \) |
| 5    | 10   | -1 0 0 1                        | \( q_1(i,j) = -1, q_1(i,j+1) = 1, q_2(i,j) = 2, q_2(i,j+1) = -1 \) |
| 6    | 11   | -1 0 -1 0                       | \( q_1(i,j) = -1, q_1(i,j+1) = 2, q_2(i,j) = 1, q_2(i,j+1) = -1 \) |
| 7    | 16   | 1 0 1 0                         | \( q_1(i,j) = 1, q_1(i,j+1) = 1, q_2(i,j) = 1, q_2(i,j+1) = 1 \) |

Table 2. Modification rule table.
Where \( m \times u \) is the size of the entire image and \((2^{\text{bit}}-1)^2\) is the image depth. When the pixel difference between camouflage image and original image is too large, it means that the image after embedding confidential messages is seriously distorted. Hence the PSNR value will be lower.

Table 3 shows the image quality measure and total hiding capacity. Here PSNR (1) is the quality of the first camouflage image and PSNR (2) is the quality of the second camouflage image. Consider street.jpg as an example. The total hidden capacity of dual images hiding method was 19456, this is mainly due to the method focused on modifying the second image leading to a higher image distortion.

This section gives the results of the Dual images hiding method using LSB matching process of data hiding using Matlab and the Xilinx. During the Matlab simulation of data embedding the image size of 160 X 160 in Figure 6 is considered for the watermarking process. The entire strategy is modeled in Matlab and the secret signature is embedded. Here the signature is considered as a random pattern. The input image before and after data hiding are shown in Figures 6 & 7 respectively. From the Matlab simulation results it is found that the embedding capacity for this kind of Dual images hiding method using LSB matching based embedding is of 19456bits with a high PSNR(1) ratio of 51.58dB and PSNR(2) ratio of 50.14dB.

After verifying the simulated results the top level entity is modelled and synthesized using the Xilinx ISE 14.3. The device family used is Artix 7(XC7A100t) with a speed grade of -3.

The comparison for the data hiding strategy after the synthesis is shown in Table 4. The Table 3 shows the embedding capacity and PSNR ratio value for the data embedding process. From the synthesis results it is found that there is no critical path in the design.

5. Conclusion

In this paper, the FPGA implementation of watermarking in real time using Dual image hiding method using LSB matching process is proposed. The algorithm does
embedding in the host image in an invisible manner. As it is a FPGA chip, it can be incorporated as a watermarking coprocessor inside a digital camera to carry out the watermarking, in the instant of capturing the picture itself. Dual images hiding method with pipelining technique to improved performance of the design as well as reduced power consumption.

6. References

1. Cox IJ, Miller ML, Bloom JA, Fridrich J, Kalker J. Digital watermarking and steganography. Morgan Kaufmann Publishers; 2007.
2. Li CT. Digital fragile watermarking scheme for authentication of JPEG images. Proceedings – Vision, Image and Signal processing. 2004 Dec; 151(6):460–6.
3. Johnson NF, Jajodia S. Exploring steganography: seeing the unseen. Computer Practices. 1998 Feb; 31(2):26–34.
4. Anderson RJ, Petitcolas FAP. On the limits of steganography. IEEE Journal on Selected Areas in Communications. 2006 Sep; 16(4):474–81.
5. Petitcolas FAP, Anderson RJ, Kuhn MG. Information hiding—a survey. Proceedings of the IEEE special issue on protection of multimedia content. 1999 Jul; 87(7):1062–78.
6. Provos N, Honeyman P. Hide and Seek: An Introduction to Steganography. IEEE computer society. 2003 May-Jun; 1(3):32–44.
7. Johnson NF, Jajodia S. Steganography: Seeing the Unseen. IEEE Computer. 1998 Feb; 31(2):26–34.
8. Mohanty SP, Ranganathan N, Namballa RK. VLSI implementation of visible watermarking for a secure digital still camera design. IEEE Proceedings of the 17th International Conference on VLSI Design (VLSID’04). 2004. p. 1063–8.
9. Karthigaikumar P, Baskaran K. FPGA and ASIC implementation of robust invisible binary image watermarking algorithm using connectivity preserving criteria. Microelectronics Journal. 2011 Jan; 42(1):82–8.
10. Krenn JR. Steganography and Steganalysis. 2004 Jan; p.1–9.
11. Lu TC, Tseng CY, Wu JH. Dual imaging-based reversible hiding technique using LSB matching. Department of information management. chaoyang university of technology. taichung 41349, taiwan, roc. signal processing. 2015 Mar; 108:77–89.