International Journal of Electronics, Communication & Soft Computing Science and Engineering
ISSN: 2277-9477, Volume 2, Issue 4

Image Error-Concealment for JPEG & PICT Format

Abhishek Kumar **Swapnil Isod** **Shailesh N. Sisat** **Vishwajit K. Barbudhe**

Abstract—**Transmission errors in JPEG & PICT can be grouped into three main classes, depending on the affected area: LL, high frequencies at the lower decomposition levels, and high frequencies at the higher decomposition levels. The first type of errors are the most annoying but can be concealed exploiting the signal spatial correlation like in a number of techniques proposed in the past; the second are less annoying but more difficult to address; the latter are often imperceptible. In this paper, we address the problem of concealing the errors when planes are damaged by proposing a new approach This problem has been overcome by applying an adaptive solution, which exploits an edge map to choose the optimal filter mask size. Simulation results demonstrated the efficiency of the proposed approach. Our approach exploits the fact that only a fraction of JPEG code space is actually used by available encoders. Data embedding is performed by mapping a used variable length code (VLC) to an unused VLC. However, standard viewers unaware of the change will not properly display the image. We address this problem by a novel error concealment technique. Concealment works by remapping run/size values of marked VLCs so that standard viewers do not lose synchronization and displays the image with minimum loss of quality. It is possible for the embedded image to be visually identical to the original even though the two files are bitwise different. The algorithm is fast and transparent and embedding is reversible and file-size preserving. Under certain circumstances, file size may actually decrease despite carrying a payload.**

Index Terms—**FEC, data embedding, , JPEG , VLC, PICT.**

I. INTRODUCTION

Data embedding describes a general framework where a payload is embedded within a cover image for authentication, fingerprinting or ownership verification (watermarking), covert communications (steganography) or simply as a vehicle to attach metadata to a cover image (data embedding). All three attempt to exploit an unused communication channel in the cover image but depending on the application there are different approaches and requirements. In digital watermarking, the payload does not have to be large but embedding must be robust, or semi-fragile, and secure. In steganography, the payload can be substantial while the cover image is secondary. Robustness and security are still important but transparency is critical to hide the presence of a covert channel. Digital watermarking of images has traditionally been implemented either in spatial domain [1] or transform domain [2]. However, Multimedia signals are seldom available in uncompressed form because compressed media is often the first generation signal available from digital cameras. It is, therefore, highly desirable to develop watermarking algorithms that work entirely in compressed domain. There has been substantial work in embedding watermarks in JPEG compressed imagery. Examples include classical transform domain watermarking algorithms where the watermark is embedded in appropriately selected transform coefficients. However, we do not consider these algorithms strictly "compressed domain" watermarking because partial decompression is required to gain access to transform coefficients. Examples are JSTEG [3], F5 [4], Outguess [5], and J-Mark [6]. The term JPEG-to-JPEG watermarking has also been used [7], although the proposed algorithm is still not truly compressed domain watermarking. What is desirable is the ability to embed the watermark directly in the bit stream of compressed media with no transcoding or decompression.

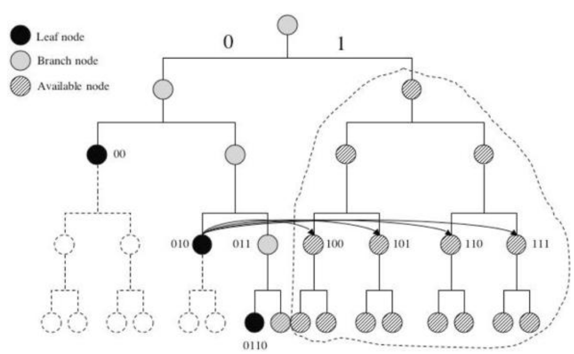


Fig. 1. Code tree for variable length codes.

PICT & JPEG Code Space: The entropy-coded portion of *PICT & JPEG* can either be Huffman or arithmetic coded. Because most images use Huffman encoding, that is the focus of our discussion. The *PICT & JPEG* standard does not require that a specific Huffman table be used for every image. Instead, to maximize compression, the standard encourages customizing the Huffman table based on the specific run/size occurrences for each individual image. However, most commercial JPEG encoders bypass this customization step and instead use example tables provided in the standard itself. These standard tables include VLC assignments for all 162 possible run/size combinations. The VLC assignment to each of the 162 possible run/sizes was determined based on a large library of images, approximately 100 000 according to the standard.

Embedding:

1. Parse the bitstream, extract VLCs.
2. Build the code tree, identify used and unused code space.
3. Identify qualified VLC pairs for mapping.
4. To embed a 1, map one used VLC to its unused counterpart.
5. Remap run/size of the unused VLC to minimize or eliminate
visual impact.
6. To embed a 0, do not map a qualified VLC.
7. Embed a hash of the message in the header.

Code mapping:

The concept of error concealment is generally associated with MPEG video. The idea is to replace corrupted macro blocks with some form of replenishment, through a variety of ways including inter or intra prediction or simple repetition of then corresponding data from previous frames. In this section we point out that error concealment is also relevant in the context of PICT & JPEG when embedding is treated as forced bit errors. One of the key objectives in this work is that embedded PICT & JPEG stream must be viewable by any PICT & JPEG viewer.

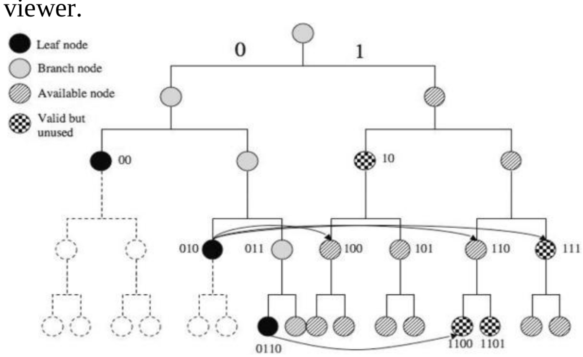


Fig. 2. In many PICT & JPEG images, a large number of VLCs are defined but never used in the image.

Therefore, we define two classes of viewers, 1) *embedding-aware* viewer, meaning that the viewer is also an authorized decoder and 2) *standard* viewer, meaning all other publicly available PICT & JPEG viewers.

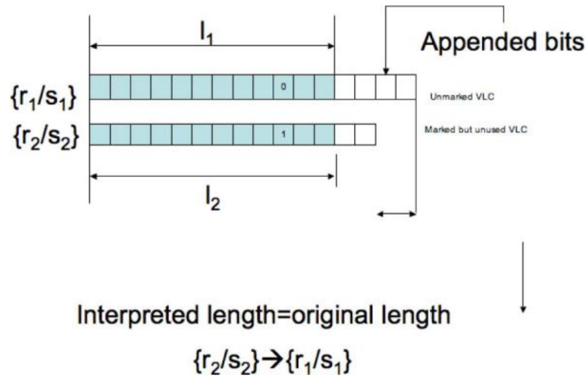


Fig. 3. Mapped VLC appears as another VLC of the same length. The new size S_2 must be remapped to S_1 to maintain synchronization. Also r2 = r1.

Run Size Remapping:

Since mapped VLCs are only mapped to unused VLCs, modifying their runs/sizes will only affect how the image is displayed. No other part of the image is affected by this modification. Ideally, run/size should be changed to match the original run/size. However, this cannot be done in all cases. The interpreted length of the mapped VLC may be equal to, shorter, or longer than the length of the original VLC.

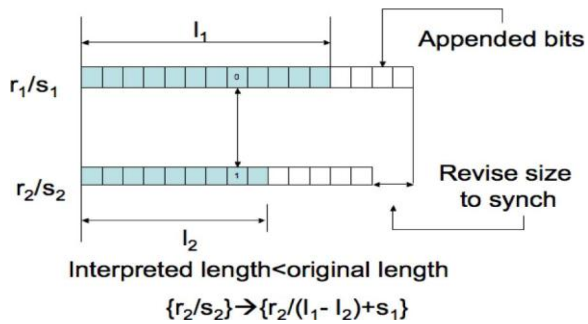


Fig. 4. Valid, but unused, VLC appears as a prefix in a mapped VLC.

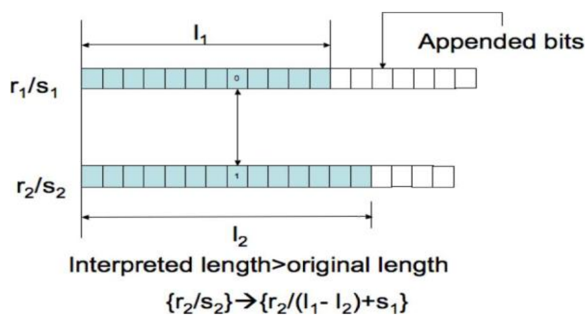


Fig. 5. Interpreted length of the mapped VLC is now a longer VLC.

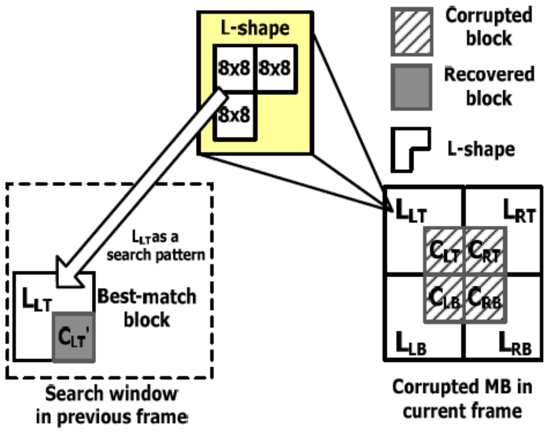
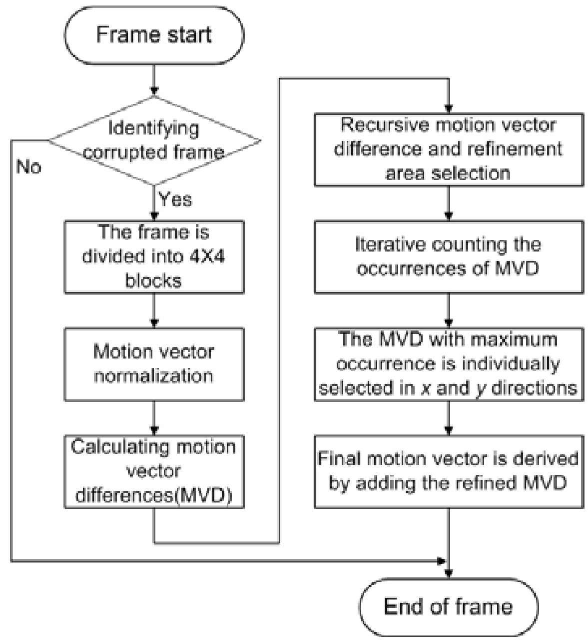


Fig 6. The illustration of corrupted MB in the block matching principle with the L-shape as a search pattern.

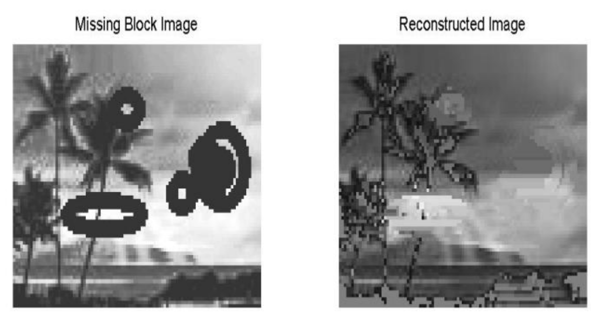
II. Error Concealment Algorithm

In this section, our proposed whole frame loss error concealment algorithm is described in detail. As mentioned before, the motion copy method is a simple and effective whole frame loss error concealment method. However, the motion copy method is based on the assumption that the motion vector difference between consecutive frames is slight. Based on this assumption, the motion copy method directly uses the motion vectors from the previous frame to conceal the lost frame. Nevertheless, there is no refinement operation performed for the motion vectors obtained by the motion copy algorithm. Based on this observation, the main idea of our proposed whole frame loss error concealment algorithm is to further refine the motion vectors derived by the motion copy method and consequently improve the quality of concealed frames.



Flowchart of the proposed algorithm

III. Results



Nature.JPEG

Fig 1 PSNR: 11.61 dB, Total Time Required: 2.79 s

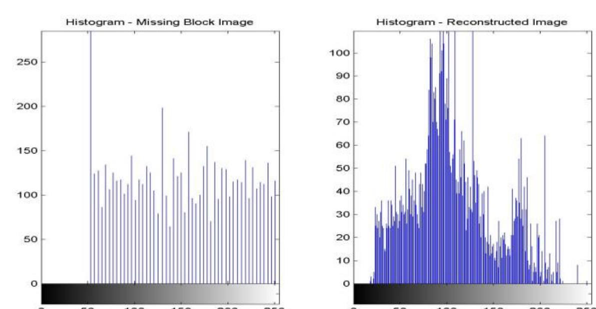
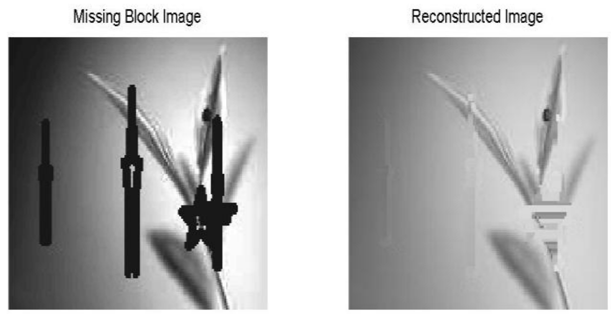


Fig 2. Output Histogram



Tree.JPEG
Fig 3 PSNR: 12.80 dB, Total Time Required: 1.65 s

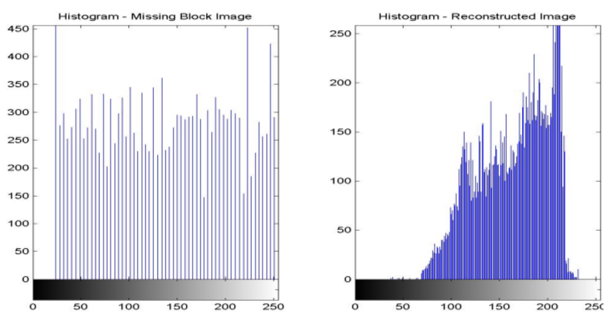


Fig 4 Output Histogram

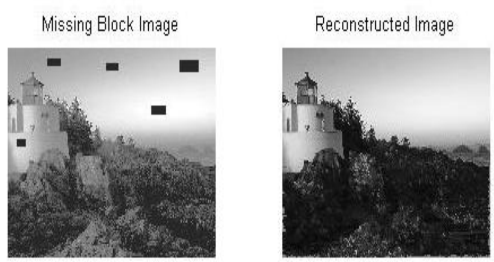


Fig 5. PSNR: 17.95 dB, Total Time Required: 314.79 s: Lighthouse.JPEG

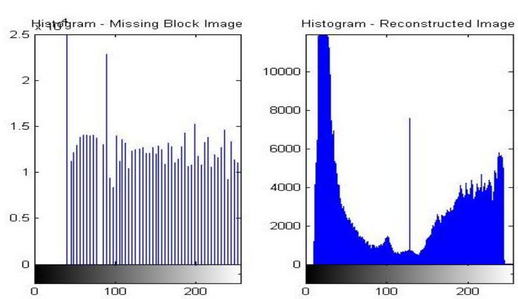


Fig 6. Output Histogram

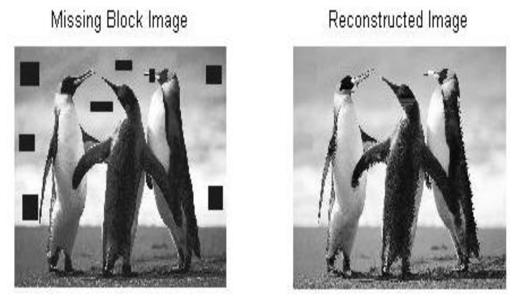


Fig 7. PSNR: 7.25 dB, Total Time Required: 130.39 s: Penguins .PICT

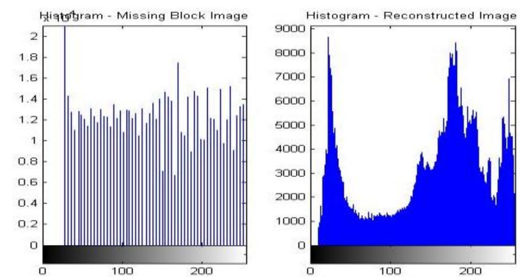


Fig 8. Output of dib format histogram

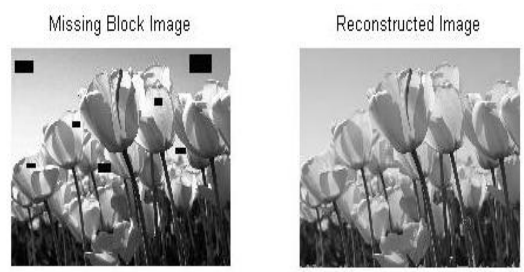


Fig 9. Flowers.PICT

PSNR:19.27 dB, Total Time Required: 20.86 s

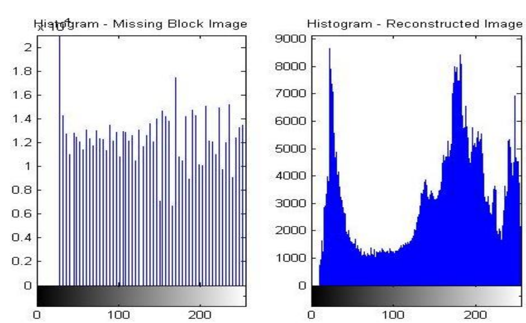


Fig 10. Output Histogram

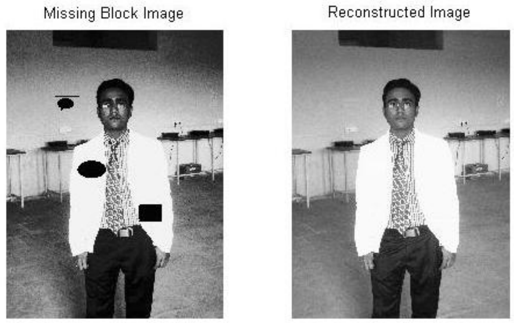


Fig 11. V2.PICT

PSNR: 20.65 dB, Total Time Required: 451.84 s: v2.eps

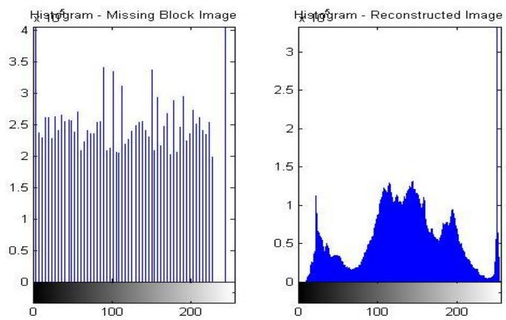
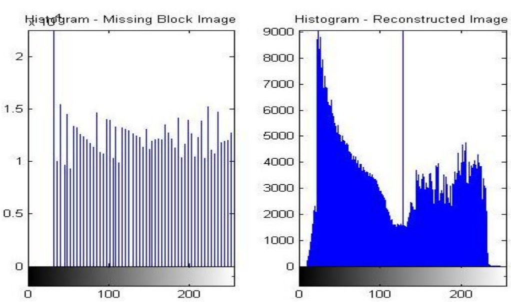


Fig 12. Output Histogram



PSNR	DCT	Harrison	Inpainting	FEC
	48.46	9.39	34.29	8.12
	12.78	14.74	10.70	3.56
	11.37	11.26	11.20	12.1
	7.71	8.56	8.85	11.7
	12.13	12.77	12.27	15.9
	36.01	11.19	12.93	5.81

CONCLUSION

In this paper, we have advanced the state of the art in JPEG data embedding on several fronts. The data is embedded directly in the bit stream and executes considerably faster than existing techniques, which require full or partial decompression. Embedding is lossless in the sense that once the data is removed, the image can be restored to its original state with no changes. The stream, despite carrying a payload, remains syntax-compliant and, hence, viewable by standard viewers. Notably, marked images can be made mathematically and, thus, visually identical to the original image.

REFERENCES

[1] F. Hartung and B. Girod, "Digital watermarking of uncompressed and compressed video," *Signal Process.*, vol. 66, pp. 283–301, May 1998.

[2] I. J. Cox, J. Kilian, F. T. Leighton, and T. Shamoon, "Secure spread spectrum watermaking for multimedia," *IEEE Trans. Image Process.*,
vol. 6, no. 12, pp. 1673–1687, Dec. 1997.

[3] Steganography Software for Windows [Online]. Available: http://www.stegoarchive.com

[4] A. Westfeld and A. Pfitzmann, I. S. Moskowitz, Ed., "High capacity despite better steganalysis (F5-a steganographic algorithm)," in *Proc. Information Hiding 4th Int. Workshop*, New York, 2001, vol. 2137, pp. 289–302, Springer-Verlag.

[5] N. Provos, "Defending against statistical steganalysis," presented at the 10th USENIX Security Symp., Washington, DC, 2001.

[6] P. H. W. Wong, O. C. Au, and J. W. C. Wong, "A data hiding technique in JPEG compressed domain," in *Proc. SPIE Security and Watermarking of Multimedia Contents III*, 2001, vol. 4314, pp. 309–320.

[7] P. H. W. Wong and O. C. Au, "A capacity estimation technique for JPEG-to-JPEG image watermarking," *IEEE Trans. Image Process.*, vol. 13, no. 8, pp. 746–752, Aug. 2003.

[8] *"ATSC Digital **Television** Standard,"* ATSC Standard *Al53*, Apr. 1995.

[9] S. Aign and K. Fazel, "Temporal and Spatial Error Concealment Techniques for Hierarchical MPEG Video Codec," *IEEE International Conference on Communication*, Vol. *3*, pp. 1778-1783, 1995.

[10] J. W. Suh and Y. **S.** Ho, "Error Concealment based on directional interpolation," *IEEE Trans. on Consumer Electronics*, Vol. 43, No. 3, pp.295-302, Aug. 1997.

[11] S. Aign and K. Fazel, "Error Detection and Concealment Measures in MPEG-2 Video Decoder," in *Proc. Int. **Workshop** on HDTV '94*, Turino, Italy, pp.169-l8l, Oct. 1994.

[12] H. Sun, K. Challapali and J. Zdepski, "Error Concealment in Digital Simulcast AD-HDTV Decoder," *IEEE Trans. on Consumer Electronics*, Vol.