Data Hiding in Perceptually Masked OpenEXR Image

Kailin Chia, Koksheik Wong, Jean-Luc Dugelay

Data Hiding in Perceptually Masked OpenEXR Image

KaiLin Chia and KokSheik Wong
School of Information Technology
Monash University Malaysia
Bandar Sunway, Malaysia
{kai.chia, wong.koksheik}@monash.edu

Jean-Luc Dugelay
Department of Digital Security
EURECOM
Biot, France
jean-luc.dugelay@eurecom.fr

***Abstract*—High dynamic range (HDR) imaging is able to capture and display significantly more colors when compared to the legacy imaging technology. Recently, HDR environment is gaining popularity and becoming common in our daily life, which resulted in the growing number of HDR images. In this work, a technique is put forward to first perceptually mask a HDR image, and then hide data into the masked HDR image. Specifically, pixels in the OpenEXR file, which are stored in half floating point precision format, are manipulated. First, a predictor is utilized to predict pixel values, where well predicted pixel locations are flagged and utilized as the venues to hide data. On the other hand, the ill predicted pixel locations are flagged as unusable. Next, each pixel is divided into segments of 5 bits, and the segments are XOR-ed and permuted to mask the perceptual semantic of the HDR image. Data hiding then takes place at locations flagged as usable, which further distorts the quality of the image. The proposed method is both reversible and separable. The basic performance of the proposed joint technique is evaluated by using 6 HDR images. In the best case scenario, 91% of pixels in the image can be utilized for data hiding purpose while the perceptual semantic of the original image is completely masked.**

***Index Terms*—HDR, OpenEXR, perceptual masking, data insertion**

I. Introduction

Perceptual masking aims to mask the semantic of a content, and it has been heavily relied on for the purpose of secure communication as well as privacy preservation. Since contents are communicated, shared, and stored online these days, digital content encryption, particularly image, audio and video, have received much attention from the research community [1]. On the other hand, data hiding is the act of inserting data into a digital content to serve specific purpose [2]. Some of the typical applications of data hiding include watermarking, steganography, tamper detection and hyperlinking. Data hiding can be applied to any digital content, including traditional contents such as audio, image, video and text document, or un-traditional contents such as IP packets [3] or laser beam [4].

Among various types of digital content, image remains to be a popular one because it can be easily captured, edited, and shared. With the advancement of smart device, even an entry level model can produce high dynamic range (HDR) image, which is able to present a greater range of luminance levels. A cost effective way to obtain a HDR image is to capture several images, each of different level of exposure (viz., underexposed, normal and overexposed) of the same scene [5]. Details captured in each image are then combined to render the scene more realistically. The higher dynamic range enables digital image to store more information about the scene, which is beneficial for applications such as medical imaging and remote sensing because the increase in stored information can provide more precise analysis. To accommodate the natural of HDR, various file formats are put forward, including JPEG2000, JPEG-XR, JPEG-XT [6], OpenEXR [7], Radiance RGBE [8] and LogLUV TIFF [9].

With the emergence of HDR image, there is an increasing need to have proper techniques for managing HDR image and preserving privacy. In that regard, some perceptual encryption techniques are proposed to mask HDR image. For example, Lin et al. [10] proposed an encryption technique for RGBE image by using elementary cellular automata (ECA) function. Yan et al. [11] proposed a format-compliant technique to encrypte LogLUV TIFF image format. The LogLUV pixel values are extracted from the TIFF file and XOR-ed with a stream cipher prior to recompression into a TIFF file. Ting et al. [12] proposed an encryption technique for JPEG XT compressed image. Specifically, DC and AC coefficients in the base and residual layers are shuffled in multiple rounds to completely mask the perceptual semantic of the image. On the other hand, researcher also invented new techniques to hide data into HDR image. Fujiyoshi et al. [13] put forward a blind and reversible method based on properties of floating point values observed in HDR image. The method is inspired by the histogram shifting method [14] designed for traditional integer-based image. Specifically, histogram of an HDR image is found to be sparse and has many empty bins. This characteristic is exploited as venues for reversible data hiding. Chang et al. [15] designed a method for Radiance RGBE [9] file format. A group of homogeneous representations for each pixel is generated by modifying the quadruplet $\{R, G, B, E\}$, where R, G, B, and E represent the red, green blue and exponent channels, respectively. Basically, a pixel value can be encoded in different representations,

This work was supported by E-Science grant (01-02-10-SF0327) by MOSTI, and in part by EU Horizon 2020 - Marie Sklodowska-Curie Action through the project entitled Computer Vision Enabled Multimedia Forensics and People Identification (Project No. 690907, Acronym: IDENTITY)

978-1-7281-1817-8/19/$31.00 ©2019 IEEE

where each representation can be decoded to the same value. Each representation is associated with $m = \lfloor \log_2(n) \rfloor$ bits, where n is the number of homogeneous representations for a color. Then, the representation which is associated to the m-bit data to be hidden is selected as the new representation.

Often, both perceptual masking and data hiding are jointly utilized. For example, in the hospital setting, a nurse may need to extract patients' identification information from perceptually masked (or encrypted) CAT scan images. To protect privacy of the patients, medical images are masked, but for management purpose, patients' information is hidden so that it can be extracted without decrypting the images. Similarly, in cloud storage, user's contents are encrypted while information about the contents is embedded for administrative purpose. Therefore, the capability to hide data into perceptually masked HDR image is desired. Some conventional joint techniques include [16]–[18], and to the best of our knowledge, none has been proposed for HDR image.

Nonetheless, in [16], the image pixels are encrypted by using stream cipher, then additional data can be embedded by modifying blocks of the encrypted image. Each block is further divided into two sets $\{S_0, S_1\}$ and the block can be used to hide one bit by flipping the LSB in S_0 or S_1, depending on the bit to be hidden. One disadvantage of this method is that the hidden data cannot be extracted before decryption. Therefore, in [17], Zhang improves the method by making it separable, where data can be extracted without viewing the original image content. In [18], to achieve separability, a specific stream encryption algorithm which can preserve correlation between neighboring pixels is proposed. Specifically, the stream encryption algorithm is followed by a permutation (applied to image blocks) to completely mask the image to ensure higher security while preserving the correlation between pixels in the image blocks. After encryption, any previously proposed DHS (difference histogram shifting) and PEHS (prediction-error histogram shifting) based RDH schemes can be deployed to hide additional information.

In this work, a technique is put forward to first perceptually mask a HDR image, then hide data into the perceptually masked image. Our work is designed for the OpenEXR format, which stores floating point values instead of integer values. The proposed method is *reversible*, where the original image can be perfectly restored, and it is also *separable* where the hidden data can be extracted without restoring (i.e., unmasking) the image. Experiments are conducted to evaluate the basic performance of the proposed joint technique for HDR image.

II. Preliminaries

A. OpenEXR File Format

OpenEXR is an open source HDR image file format developed by Industrial Light & Magic [7]. It supports higher dynamic range and color precision when compared to the traditional 8-bit file format. OpenEXR file is capable of storing pixels in 16- or 32-bit floating-point data type, and the pixels can be compressed in either lossless or lossy mode. OpenEXR image can have a number of user-defined channels, each with a different data type. The most common ones are the R, G and B channels [7].

Figure 1: 16-bit half precision format.

TABLE I: Semantic of 16-bit floating point representation (i.e., *half* format) in OpenEXR.

Condition	Value
$E = 11111_2 = 31$ and $M > 0$	NaN
$E = 11111_2 = 31$ and $M = 0$	$\pm$ Inf
$E = 0$ and $M > 0$	Subnormal Number
others	Normalized Number

Specifically, the half precision format is a simplified version of the IEEE-754 floating-point specification [19], which is widely known for its single and double precision floating point formats. Half precision format is made up of 1 sign bit (denoted by S), 5 exponent bits (denoted by E), and 10 mantissa bits (denoted by M). Figure 1 further details the arrangement of each group of bits. Similar to single and double precision formats, the half precision format supports denormalized numbers, positive and negative infinities, and NaN (i.e., not a number). The representable magnitude ranges from $2^{-24} \approx 5.96 \times 10^{-8}$ to $(22^{10}) \times 215 = 65504$. Table I summarizes these representations for half precision format based on E and M.

III. Proposed Method

Since the exponent part of a pixel determines its coarse value, the exponents of a natural image are highly correlated, particularly for a group of pixels in the same region. This characteristic can be exploited for reversible data hiding, where the exponent values can be replaced by data bits but at the same time they can be completely recovered later through prediction. In addition, the altered exponent values can result in drastic change to the pixel values, causing severe quality degradation. Figure 2 shows the flow of processes in achieving perceptual masking and data hiding.

A. Preprocessing

The exponent part of all pixels are first predicted by using a pixel predictor. The purpose of this step is to identify well predictable locations, so that they can be replaced by the data to be hidden. In this work, we consider the Median Edge Detection (MED) predictor in JPEG-LS [20]. MED is a simple predictor which relies on the direction of the edges, i.e., horizontal, vertical, or no edges at all. The operation is formally described by Eq. (1), where x is the value to be predicted, a, b and c are the left, top and top-left pixels of x,

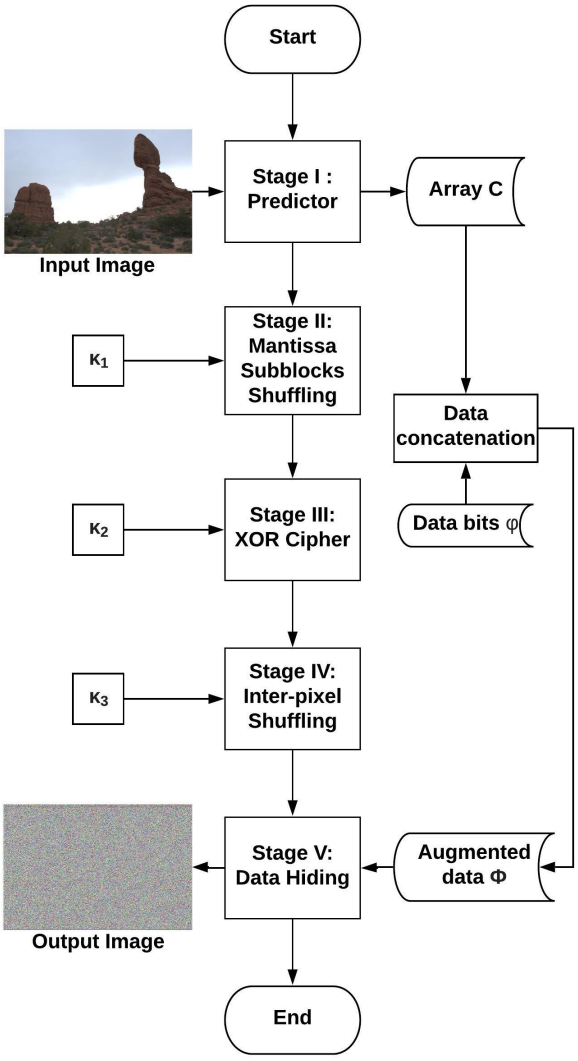


Figure 2: Overall process of proposed method.

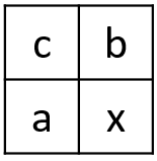


Figure 3: Layout of MED predictor.

respectively, as shown in Figure 3. The predicted value of x is computed as follows:

$$x' = \begin{cases} min(a,b), & \text{if } c \geq \max(a,b); \\ max(a,b), & \text{if } c \leq \min(a,b); \\ a+b-c, & \text{otherwise.} \end{cases} \quad (1)$$

MED predictor predicts all values in one pass following the raster scan order, where pixels at the boarder are utilized as the initial reference pixels.

The prediction error $e = E_o - E_p$ is computed, where E_p and E_o are the predicted and original exponent values, respectively. To flag the usability of each exponent for data hiding purpose, the value e is analyzed. Specifically, if $e = 0$ (viz., the prediction is perfect), the least significant bit (LSB) of E_o is modified to '0'. Otherwise, the LSB of E_o is set to '1'. The original LSB of an ill predicted E_o is stored in a dynamic array C. As for well predicted E_o, its LSB is not stored because the entire E_o can be predicted perfectly. Array C is important to achieve reversibility. Let E'_o denoted the modified exponent value, where $|E_o - E'_o| \leq 1$. The LSB of E'_o then guides the data hiding process, where '0' denotes that data can be hidden (or from the perspective of the receiver - data is hidden), while '1' means that no data is hidden.

B. Masking The Image

First, the 10-bit mantissa part M is divided and treated as two independent blocks each with 5 bits, namely, M_1 and M_2. The subblocks M_1 and M_2 each assumes a size of 5 bits, which is the same as the length of E_o to ease the masking operations. Therefore, each pixel will assume the form of $[S, E'_o, M_1, M_2]$, but for the purpose for this work, S is ignored. Second, M_1 and M_2 are extracted from all pixels, and shuffled by using a key κ_1. The shuffled M_1 and M_2 form new mantissas (see Fig. 4). The potential outcomes for each pixel include:

$$\begin{aligned} &[E'_o(i), M_1(j), M_2(k)], \\ &[E'_o(i), M_2(j), M_1(k)], \\ &[E'_o(i), M_1(j), M_1(k)], j \neq k, \text{ or,} \\ &[E'_o(i), M_2(j), M_2(k)], j \neq k, \end{aligned} \quad (2)$$

where $M_1(j)$ denotes the M_1 block from the j-th pixel, and $M_2(k)$ as well as $E'_o(i)$ are defined in same manners. The purpose of Stage II is to mask the details or textures (particularly recurring patterns) captured by the mantissa part, although mantissa can only cause small variation of value in comparison to the exponent part.

Next, the 4 most significant bits (4MSB) of the exponents E'_o for each pixel as well as its newly formed mantissa are XOR-ed with a random sequence of 0's and 1's, which is generated with the key κ_2 (Stage III). Figure 5 illustrates an example, where E''_o and M'' refer to the modified exponent and mantissa parts, respectively. Note the LSB of E'_o does not undergo the XOR step to maintain the flag, i.e., $LSB(E'_o) = LSB(E''_o)$. This XOR step is important to break the high correlationship between the pixels, particularly the ill predicted pixels, as they will not be involve the data hiding process in Stage V (see Section III-C for detailed explanation). The newly formed pixels are then shuffled by using the key κ_3 (Stage IV) to further increase the overall entropy of the image pixels.

It is noteworthy that Stages II, III and IV are independent to each other. Therefore, they can be implemented without following any particular order. In addition, a master key κ_m for content masking can be considered to generate the keys κ_1, κ_2 and κ_3.

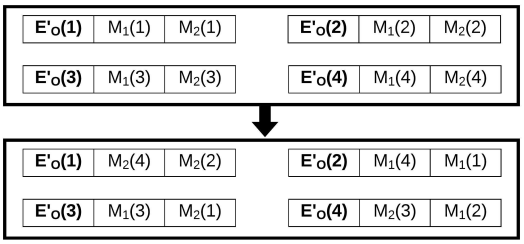


Figure 4: An example output of mixing M_1 and M_2 blocks among pixels (Stage II).

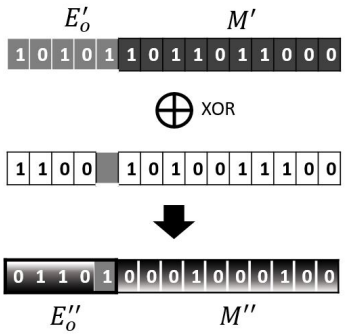


Figure 5: Masking pixel using XOR operation (Stage III).

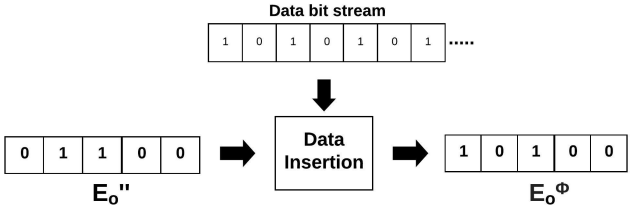
C. Data Hiding

Let ϕ be the external data to be hidden, which is usually encrypted with a key κ_4. For reversibility purpose, the original LSB of the exponent, denoted by C needs to be stored. Therefore, the augmented data $\Phi = [C|\phi]$ is constructed, where $|$ refers to the concatenation operation. Φ is then processed in segments of 4-bit. Data hiding is carried out by considering pixel locations where $LSB(E''_o) = 0$ for E''_o being the exponent part of the pixel at Stage IV. Specifically, for each usable pixel, the most significant 4 bits of its exponent are replaced by a 4-bit segment from Φ. Data hiding will not take place if $LSB(E''_o) = 1$. The process is repeated until all segments of Φ are inserted or when one runs out of usable pixels. Figure 6 illustrates the data insertion process, where the new exponent value is denoted as E^{Φ}_o.

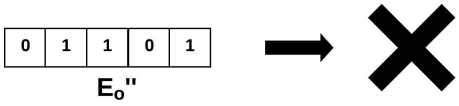
D. Data Extraction and Image Recovery

The data extraction and image recovery steps are exactly the reverse of the encryption and data insertion processes. Specifically, the inserted data, i.e., Φ is first extracted, and it is processed to obtain ϕ and C. The intended receiver will be able to decrypt ϕ using κ_4. Note that the data can be extracted without the need to decrypt the image.

To recover the original image, the encryption processes need to be reversed, i.e., inverse shuffling in Stage IV using κ_3, XOR operation in Stage III using the same binary sequence generated by κ_2, and inverse shuffling of the M_1's and M_2' blocks in Stage II using κ_1. Next, the LSB of the exponent part for each pixel is examined to reconstruct the image. Specifically, when LSB(E) = '1', that means 4MSB planes



(a) Case 1: E''_o with LSB of '0'.



(b) Case 2: E''_o with LSB of '1'

Figure 6: Hiding data into preprocessed exponent part.

were not replaced, i.e., the original sequence is intact, and the first bit of C is removed and used as the LSB for E. On the other hand, if LSB(E) = '0', then 4MSB planes are recovered through prediction. Note that the predict-to-recover process must be carried out in raster scan order again to ensure the bits from C is reinserted at the correct position, as only pixels flagged as unpredictable have their original LSB stored in C (length of array $C \neq$ total pixels in image).

IV. Experiment

The proposed algorithm is implemented in Matlab R2018a. 6 images from the dataset [21] are selected randomly for experiment purpose. Since the original image in [21] is large in dimension (i.e., larger than 4K), the selected images are down-scaled to 6.25% of the original size. These images are shown in Fig. 7. It is confirmed that the inserted data can be extracted, and the original image can be completely reconstructed from its processed counterpart.

A. Effective Payload

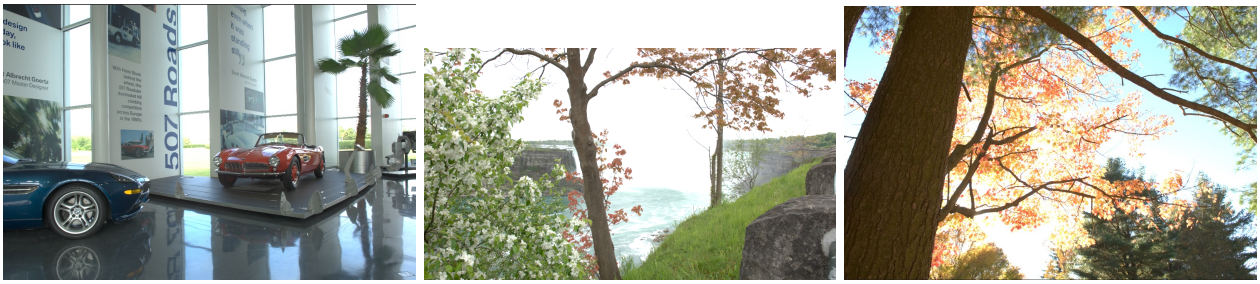
First, the raw capacity achieved for each image using the proposed algorithm is computed as:

$$\frac{\textit{number of usable pixels}}{\textit{total pixels}} \times 100\%, \tag{3}$$

while the percentage of side information is computed as:

$$\frac{\textit{number of unusable pixels}}{\textit{number of usable pixels}} \times 100\%. \tag{4}$$

Here, raw capacity refers to the actual number of pixels that are flagged as suitable for hiding data, while side information percentage is the fraction of raw capacity spent on coding the array C. Table II records the capacity for each test image. Results suggest that the percentage of usable pixels is high for all HDR images considered, with an average of more than



(a) 507 (b) CanadianFalls (c) CemeteryTree(2)

(d) Frontier (e) BigFootPass (f) BalancedRock

Figure 7: Six HDR test images from [21].

TABLE II: Hiding capacity for different test images.

Image	raw capacity (%)	side info capacity (%)
507	89.2549	2.9442
CanadianFalls	74.0750	8.6622
CemeteryTree(2)	63.5323	14.2582
Frontier	86.3199	3.8944
BigfootPass	90.5954	2.5238
BalancedRock	88.5727	3.1595

70%. However, some images (e.g., CemeteryTree(2)) yield lower raw capacity because there are less well predictable pixels. In other words, higher number of ill predictable pixels result in relatively larger side information C, and hence less available space to insert data.

B. Image Quality

This section presents the quality of the processed image from Stage V in two scenarios: the first one hides data after the prediction stage by skipping Stage II, III and IV, and; the second one hides data after the encryption stage, viz., stepping through Stage II, III and IV. By visual inspection, output image in the first scenario is already severely distorted. However, some details of the image are still visible (e.g., see outline of wheel at the lower left region in Fig. 8(c)). Nonetheless, the image produced by in the second scenario is completely distorted, i.e., there is no trace of the original perceptual semantic. This observation verifies that although replacing the exponent part with payload bits already result in a highly distorted image, Stage II, III and IV are crucial to further intensify distortion to completely mask the perceptual semantic of an image. Figure 8 shows two representative output images for both scenarios.

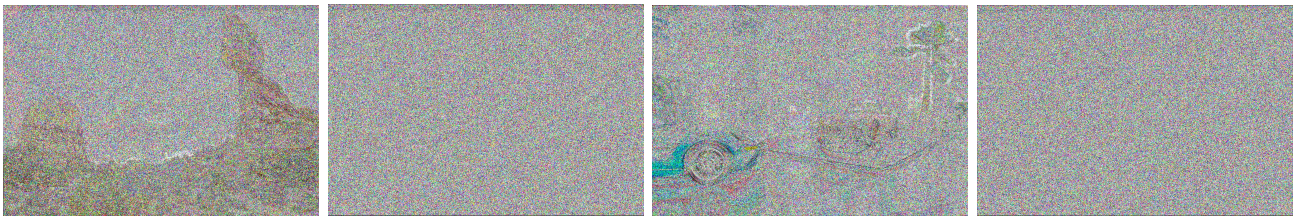
To quantify distortion, log-based PSNR [22] is computed for the processed images for both scenarios, and the results are recorded in Table III. The logarithmic function is an approximation on how Human Visual System (HVS) responds to the visible luminance range. The capability to mimic how human eye response to light makes this quality metric more suitable in capturing the quality of HDR image. As expected, the PSNR values are low (i.e., close to 0) and this confirms that the processed images are highly distorted after applying the proposed image masking and data hiding technique.

TABLE III: Quality assessment after processing (dB).

Image	Log_PSNR	
	Scenario 1	Scenario 2
507	3.3252	2.7679
CanadianFalls	4.4658	3.0105
CemeteryTree(2)	4.8320	2.7357
Frontier	3.3314	2.5717
BigfootPass	3.9878	3.4865
BalancedRock	3.4036	2.7252

C. Discussion

Last but not least, the proposed method is suitable to be applied for floating point numbers, specifically half precision format. The conventional methods [16]–[18] are not suitable for floating point numbers because they are designed for integer pixels, which have different intrinsic properties. For example, 3LSB are manipulated to hide data in [16], but the technique is not readily applicable to pixels stored in floating point format. Specifically, the mantissa part of the pixel in half-precision format is rather random and it is challenging to predict them precisely.



(a) BalancedRock: Data hidden after Stage I (Scenario 1)

(b) BalancedRock: Data hidden after Stage IV (Scenario 2)

(c) 507: Data hidden after Stage I (Scenario 1)

(d) 507: Data hidden after Stage IV (Scenario 2)

Figure 8: Output images for both scenarios.

V. Conclusions

In this paper, image masking and data insertion techniques are put forward for HDR image stored in the OpenEXR format. Specifically, the exponent part of each pixel is classified as predictable or not predictable. The predictable pixels are modified to hide data, while side information is stored as part of the payload for perfect image recovery purpose. To further intensify the distortion, mantissa subblocks shuffling, XOR cipher and inter-pixel shuffling are applied to the pixels. Experiments suggest that, on average, the number of usable pixels is greater than 70% of the total number of pixels. Results also suggest that the quality of an image can be completed distorted and perfectly recovered later when required.

As future work, more detailed analysis will be performed to investigate into the security aspect of the proposed method. In addition, the file structure of OpenEXR will be further explored to improve capacity and secrecy.

References

[1] N. S. Kulkarni, B. Raman, and I. Gupta, *Multimedia Encryption: A Brief Overview*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 417–449. [Online]. Available: https://doi.org/10.1007/978-3-642-02900-4_16

[2] I. Cox, M. Miller, J. Bloom, J. Fridrich, and Kalker Ton, *Digital watermarking and steganography*. Morgan Kaufmann Publishers, 2008.

[3] O. I. Abdullaziz, V. T. Goh, H.-C. Ling, and K. Wong, "AIPISteg: An active IP identification based steganographic method," *Journal of Network and Computer Applications*, vol. 63, pp. 150 – 158, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S108480451600059X

[4] F. Yue, C. Zhang, X.-F. Zang, D. Wen, B. D. Gerardot, S. Zhang, and X. Chen, "High-resolution grayscale image hidden in a laser beam," *Light: Science &Amp; Applications*, vol. 7, pp. 17 129 EP –, 01 2018. [Online]. Available: https://doi.org/10.1038/lsa.2017.129

[5] P. E. Debevec and J. Malik, "Recovering high dynamic range radiance maps from photographs," in *Proceedings of the 24th annual conference on Computer graphics and interactive techniques - SIGGRAPH '97*. ACM Press, 1997, pp. 369–378. [Online]. Available: http://portal.acm.org/citation.cfm?doid=258734.258884

[6] T. Richter, A. Artusi, and T. Ebrahimi, "JPEG XT: A New Family of JPEG Backward-Compatible Standards," *IEEE MultiMedia*, vol. 23, no. 3, pp. 80–88, jul 2016. [Online]. Available: http://ieeexplore.ieee.org/document/7535096/

[7] F. Kainz, R. Bogart, P. Stanczyk, and P. Hillman, "Technical Introduction to OpenEXR," 2013. [Online]. Available: http://www.openexr.com/TechnicalIntroduction.pdf

[8] G. Ward, "Real Pixels," in *Graphics Gems II*. Elsevier, 1991, pp. 80–83. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/B9780080507545500256

[9] G. W. Larson and G. Ward, "LogLuv Encoding for Full-Gamut, High-Dynamic Range Images," *Journal of Graphics Tools*, vol. 3, no. 1, pp. 15–31, jan 1998. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1080/10867651.1998.10487485

[10] K.-S. Lin, T.-H. Chen, C.-H. Lin, and S.-S. Chang, "A tailor-made encryption scheme for high-dynamic range images," in *Genetic and Evolutionary Computing*, J.-S. Pan, P. Krömer, and V. Snášel, Eds. Cham: Springer International Publishing, 2014, pp. 183–192.

[11] J. Yan, T. Chen, and C. Lin, "Encryption in high dynamic range images for RGBE format," in *2013 Ninth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, Oct 2013, pp. 493–496.

[12] J. Ting, S. Y. Ong, and K. S. Wong, "Format-compliant perceptual encryption method for JPEG XT," in *2019 IEEE International Conference on Image Processing*, to appear.

[13] M. Fujiyoshi and H. Kiya, "A Blind Reversible Data Hiding Method for High Dynamic Range Images Taking Advantage of Sparse Histogram," in *Digital Forensics and Watermarking: 16th International Workshop , IWDW 2017, Magdeburg, Germany, August 23-25, 2017, Proceedings*, C. Kraetzer, Y.-Q. Shi, J. Dittmann, and H. J. Kim, Eds. Cham: Springer International Publishing, 2017, pp. 347–361. [Online]. Available: https://doi.org/10.1007/978-3-319-64185-0{_}26

[14] J. Hwang, J. Kim, and J. Choi, "A Reversible Watermarking Based on Histogram Shifting," in *Digital Watermarking: 5th International Workshop, IWDW 2006, Jeju Island, Korea, November 8-10, 2006. Proceedings*, Y. Q. Shi and B. Jeon, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 348–361. [Online]. Available: https://doi.org/10.1007/11922841{_}28

[15] C.-C. Chang, T.-S. Nguyen, and C.-C. Lin, "A new distortion-free data embedding scheme for high-dynamic range images," *Multimedia Tools and Applications*, vol. 75, no. 1, pp. 145–163, jan 2016. [Online]. Available: http://link.springer.com/10.1007/s11042-014-2279-5

[16] X. Zhang, "Reversible data hiding in encrypted image," *IEEE Signal Processing Letters*, vol. 18, no. 4, pp. 255–258, April 2011.

[17] ——, "Separable reversible data hiding in encrypted image," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 2, pp. 826–832, April 2012.

[18] F. Huang, J. Huang, and Y. Shi, "New framework for reversible data hiding in encrypted domain," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 12, pp. 2777–2789, Dec 2016.

[19] W. Stallings, *Computer Organization and Architecture - Designing for Performance (7. ed.)*. Pearson / Prentice Hall, 2006.

[20] M. J. Weinberger, G. Seroussi, and G. Sapiro, "The LOCO-I lossless image compression algorithm: principles and standardization into jpeg-ls," *IEEE Transactions on Image Processing*, vol. 9, no. 8, pp. 1309–1324, Aug 2000.

[21] M. D. Fairchild, "The HDR Photographic Survey." [Online]. Available: http://rit-mcsl.org/fairchild/HDRPS/CIC15HDRSurvey.pdf

[22] T. O. Aydn, R. Mantiuk, and H.-P. Seidel, "Extending quality metrics to full luminance range images," B. E. Rogowitz and T. N. Pappas, Eds., vol. 6806. International Society for Optics and Photonics, feb 2008, p. 68060B. [Online]. Available: http://proceedings.spiedigitallibrary.org/proceeding.aspx?doi=10.1117/12.765095