



High bit depth infrared image compression via low bit depth codecs

Belyaev, Evgeny; Mantel, Claire; Forchhammer, Søren

Published in:

Proc. SPIE 10403, Infrared Remote Sensing and Instrumentation XXV

Link to article, DOI:

[10.1117/12.2275542](https://doi.org/10.1117/12.2275542)

Publication date:

2017

Document Version

Publisher's PDF, also known as Version of record

[Link back to DTU Orbit](#)

Citation (APA):

Belyaev, E., Mantel, C., & Forchhammer, S. (2017). High bit depth infrared image compression via low bit depth codecs. In Proc. SPIE 10403, Infrared Remote Sensing and Instrumentation XXV [104030A] SPIE - International Society for Optical Engineering. DOI: 10.1117/12.2275542

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

PROCEEDINGS OF SPIE

SPIDigitalLibrary.org/conference-proceedings-of-spie

High bit depth infrared image compression via low bit depth codecs

Evgeny Belyaev, Claire Mantel, Søren Forchhammer

Evgeny Belyaev, Claire Mantel, Søren Forchhammer, "High bit depth infrared image compression via low bit depth codecs," Proc. SPIE 10403, Infrared Remote Sensing and Instrumentation XXV, 104030A (30 August 2017); doi: 10.1117/12.2275542

SPIE.

Event: SPIE Optical Engineering + Applications, 2017, San Diego, California, United States

High bit depth infrared image compression via low bit depth codecs

Evgeny Belyaev, Claire Mantel, and Søren Forchhammer

DTU Fotonik, Technical University of Denmark, 2800 Kgs. Lyngby, Denmark

ABSTRACT

Future infrared remote sensing systems, such as monitoring of the Earth's environment by satellites, infrastructure inspection by unmanned airborne vehicles etc., will require 16 bit depth infrared images to be compressed and stored or transmitted for further analysis. Such systems are equipped with low power embedded platforms where image or video data is compressed by a hardware block called the video processing unit (VPU). However, in many cases using two 8-bit VPUs can provide advantages compared with using higher bit depth image compression directly. We propose to compress 16 bit depth images via 8 bit depth codecs in the following way. First, an input 16 bit depth image is mapped into 8 bit depth images, e.g., the first image contains only the most significant bytes (MSB image) and the second one contains only the least significant bytes (LSB image). Then each image is compressed by an image or video codec with 8 bits per pixel input format. We analyze how the compression parameters for both MSB and LSB images should be chosen to provide the maximum objective quality for a given compression ratio. Finally, we apply the proposed infrared image compression method utilizing JPEG and H.264/AVC codecs, which are usually available in efficient implementations, and compare their rate-distortion performance with JPEG2000, JPEG-XT and H.265/HEVC codecs supporting direct compression of infrared images in 16 bit depth format. A preliminary result shows that two 8 bit H.264/AVC codecs can achieve similar result as 16 bit HEVC codec.

Keywords: Infrared image coding, HDR

1. INTRODUCTION

Many infrared sensors nowadays offer a sensitivity of 14 to 16 bits for raw data and reducing the dynamic range of such acquired images means in most cases losing some information. For scientific radiometry applications of infrared remote sensing systems, such as monitoring of the Earth's environment by satellites, infrastructure inspection by unmanned airborne vehicles etc., 16 bit depth infrared images are therefore required and need to be compressed and stored or transmitted for further analysis. There are several image and video coding standards which support 9–16 bit depth format, e.g., H.265/HEVC,¹ JPEG2000² and JPEG-XT.³ However, hardware implementations of the listed codecs are not available in the vast majority of existing low power embedded platforms, while their software implementations are too computationally demanding. At the same time, in most cases such platforms are equipped with a hardware block called video processing unit (VPU) which is able to compress images or videos captured in 8 bit depth format. The main drawback is that such VPU's cannot compress 16 bit depth data directly.

To overcome the limitation mentioned, in this paper we present a simple approach which enables 16 bit depth infrared images compression utilizing codecs designed for 8 bit depth formats. Inspired by JPEG-XT framework,^{4,5} first we apply a separation algorithm which splits an input 16 bit depth image in two 8 bit depth images. The first image contains only the most significant bytes (MSB image) and the second one contains only the least significant bytes (LSB image). Then each image is compressed by an existing image or video codec with 8 bits per pixel input format. Taking into account that the most frequent VPU can compress with JPEG, PNG and H.264/AVC we utilize these codecs in our approach. We show how to select the quantization steps for both images in order to provide the maximum Signal-to-Noise Ratio (SNR) for a given bit rate constraint. Afterwards,

Further author information: (Send correspondence to Evgeny Belyaev)

Evgeny Belyaev, Claire Mantel, Søren Forchhammer: E-mails: {evbel,clma,sofo}@fotonik.dtu.dk

The work is supported by EUDP Project Grant 64015-0072, which is in collaboration with Sky-Watch and Zeuxion.

we present a predefined set of quantization steps which can be applied for real-time scenario. Experimental results show that the proposed encoding scheme provides similar or better rate-distortion performance comparing to HEVC and JPEG-Xt Parts B and C with 'global Reinhard' tone mapping operator.⁹

The rest of the paper is organized as follows. Section 2 introduces the proposed coding algorithm and the encoding parameters selection. Performance evaluation is presented in Sections 3. Conclusions are made in Section 4.

2. PROPOSED CODING SCHEME

2.1 Encoding and decoding

We assume that each pixel x_i of an input image $\mathbf{x} = \{x_1, x_2, \dots, x_N\}$ is a non-negative integer value represented in 16-bit format (two bytes per pixel), i.e., $0 \leq x_i \leq 2^{16} - 1$. First we extract the most significant byte (MSB) of each pixel x_i and generate the MSB image $\mathbf{y} = \{y_1, y_2, \dots, y_N\}$ as

$$y_i = \left\lfloor \frac{x_i}{256} \right\rfloor. \quad (1)$$

Then we extract the least significant byte (LSB) of each pixel x_i and generate LSB image $\mathbf{z} = \{z_1, z_2, \dots, z_N\}$ as

$$z_i = (x_i \bmod 256). \quad (2)$$

One can see that $0 \leq y_i, z_i \leq 2^8 - 1$, i.e., both the MSB and LSB images can be represented by 8 bits per pixel known as YUV 4:0:0 color space format which is supported by the vast majority of existing VPU's.

For \mathbf{y} and \mathbf{z} compression we utilize two different image encoders: the first one is for lossless compression and the second one is for lossy compression. As an example, Figure 1 shows the case when PNG encoder is used for lossless coding and JPEG is used for lossy coding. The MSB image is compressed by both encoders and then the

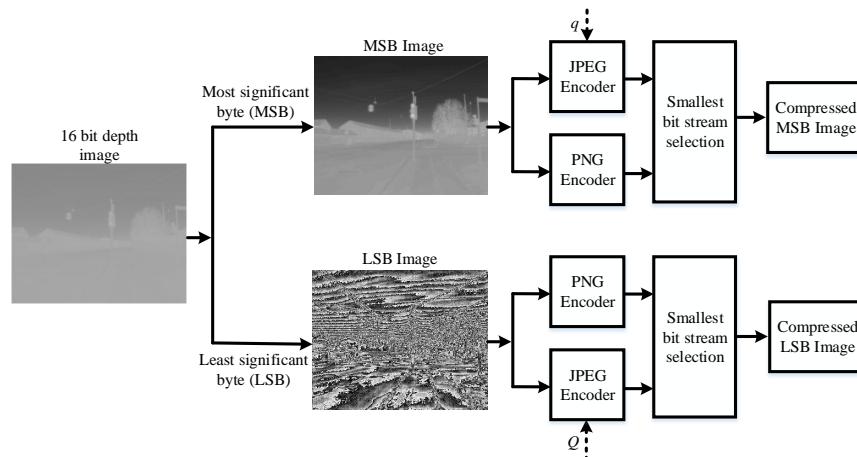
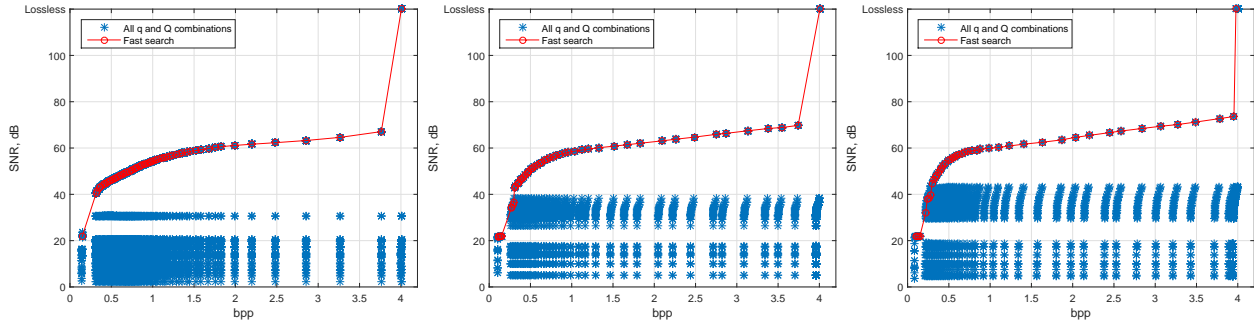


Figure 1: Proposed encoding scheme utilizing JPEG and PNG video processing units (VPU) shortest bit stream is stored or transmitted. The same approach is applied for the LSB image. At the decoder side, the reconstructed pixel is calculated as

$$\hat{x}_i(q, Q) = 2^8 \cdot \hat{y}_i(q) + \hat{z}_i(Q), \quad (3)$$

where \hat{y}_i and \hat{z}_i are reconstructed values of MSB and LSB images, respectively, and q and Q are corresponding JPEG quality factors, $q, Q \in \{1, \dots, 100\}$. If $Q = 1$ then before the encoding we multiply all pixels of LSB image by zero.



(a) JPEG+PNG (b) x264 (ultrafast)+PNG (c) x264 (veryslow)+PNG
 Figure 2: Fast optimization via (6) versus all $\{q, Q\}$ combinations for different VPU's

2.2 Fast selection of coding parameters

The coding performance of the considered scheme depends on quality factors q and Q .⁶ In general case, to obtain operational rate-distortion curve for each image we need to encode it with all possible pairs $\{q, Q\}$, i.e., $|q| \times |Q| = 10000$ times which is too computationally demanding (if H.264/AVC is selected as lossy encoder then $q, Q \in \{0, 1, \dots, 51\}$ and $|q| \times |Q| = 2704$). Instead we propose the following fast optimization. Building on (3) the sum of absolute differences between original and reconstructed images is calculated as:

$$D(q, Q) = \sum_i |x_i - \hat{x}_i(q, Q)| = \sum_i |2^8(y_i - \hat{y}_i(q)) + z_i - \hat{z}_i(Q)|. \quad (4)$$

Equation (4) can be approximated as

$$D(q, Q) \approx \hat{D}(q, Q) = 2^8 \sum_i |y_i - \hat{y}_i(q)| + \sum_i |z_i - \hat{z}_i(Q)| = 2^8 D_y(q) + D_z(Q). \quad (5)$$

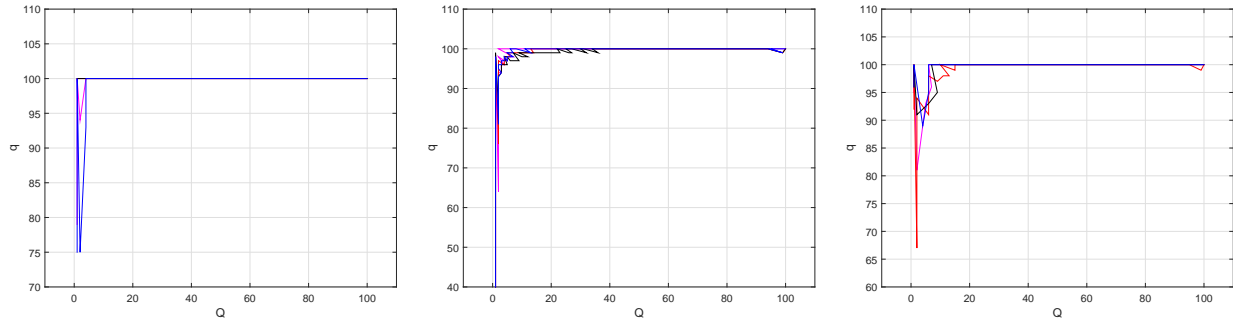
The sums of absolute differences $D_y(q)$ and $D_z(Q)$ as well as corresponding bit stream sizes $R_y(q)$ and $R_z(Q)$ can be determined utilizing $|q| + |Q|$ encodings. Then for each target bit stream size R^* we need to find $\{q^*, Q^*\}$, so that

$$\begin{cases} \min \hat{D}(q, Q) \\ \text{subject to } R_y(q) + R_z(Q) \leq R^*. \end{cases} \quad (6)$$

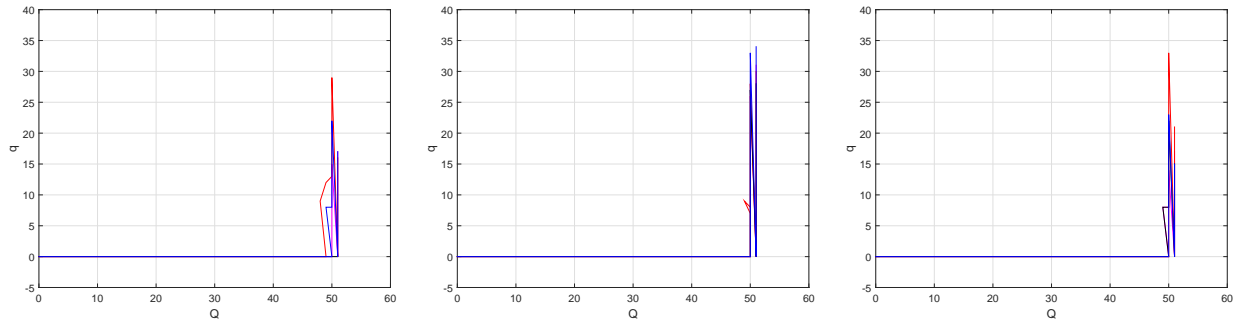
The proposed optimization is $\frac{|q| \times |Q|}{|q| + |Q|}$ times faster than the full search. Figure 2 shows rate-distortion points for all possible combinations of $\{q, Q\}$ and optimization via (6) for test image Fluke 1 (see Figure 5, a)). Here we considered three combinations of lossless and lossy encoders. In all of the cases the PNG encoder has been used for lossless encoding, while as lossy encoder we used JPEG (Figure 2, a)), x264 in *ultrafast* profile (Figure 2, b)) and x264 in *veryslow* profile (Figure 2, c)). One can see that in all cases the proposed optimization via (6) is close to the operational rate-distortion curves.

2.3 Predefined set of coding parameters

The proposed optimization via (6) is much faster than the full search. However, it is still computationally demanding for real-time applications, since it requires $|q| + |Q|$ encodings. To overcome it we propose to use predefined $\{q_i, Q_i\}$ pairs so that an increase of index i corresponds to reduction of bit rate and increase of distortion close to the operational rate-distortion curve. Figure 3 shows pairs $\{q^*, Q^*\}$ obtained via (6) for different test image sets in a case when PNG and JPEG are selected as lossless and lossy codecs. Each curve starts from the pair $\{100, 100\}$ which corresponds to the maximum quality and then connects all the remaining pairs in the bit rate reduction order. We can observe that in most of the cases the following set of pairs $\{100, 100\}, \{100, 99\}, \dots, \{100, 20\}, \{95, 1\}, \{94, 1\}, \dots, \{1, 1\}$ correspond to the rate-distortion curves obtained via (6) for all test images. Building on this observation we propose to select the encoding parameters $\{q, Q\}$ only from the listed set of pairs. Similarly, in a case when PNG and x264 are utilized as lossless and lossy codecs, each curve starts from the pair $\{0, 0\}$ and the following set $\{0, 0\}, \{0, 1\}, \dots, \{0, 45\}, \{0, 51\}, \{1, 51\}, \dots, \{51, 51\}$ provides the best performance.



(a) Fluke 1, 2, 3 and 4 (b) Boson 1, 2, 3 and 4 (c) Drone 1, 2, 3 and 4
 Figure 3: $\{q, Q\}$ pairs selected via (6) for JPEG+PNG scheme



(a) Fluke 1, 2, 3 and 4 (b) Boson 1, 2, 3 and 4 (c) Drone 1, 2, 3 and 4
 Figure 4: $\{q, Q\}$ pairs selected via (6) for x264 (ultrafast)+PNG scheme

3. PERFORMANCE EVALUATION

Experimental results were obtained for infrared image is captured by three different infrared cameras: Fluke Ti32, Flir Boson and Flir Tau2 with resolutions 320×240 , 320×256 and 640×512 , respectively. For each camera we selected four images called Fluke 1–4, Boson 1–4 and Drone 1–4* (see Figure 5). The proposed infrared image coding approach is compared with H.265/HEVC,¹ JPEG2000² and JPEG-Xt³ video and image coding standards supporting 16 bit depth images. For H.265/HEVC results we used Main profile of HM 16.14 reference software⁷ compiled with high bit depth support (RExt_HIGH_BIT_DEPTH_SUPPORT=1). For JPEG20000 results we used OpenJpeg software version 2.1.2.⁸ The JPEG-Xt compression was performed using the reference software (part of ISO/IEC 18477-5) release 1.53 (Feb. 2017). Profiles B and C were used with the embedded 'global Reinhard' tone mapping operator to obtain the Low Dynamic Range image. The JPEG-Xt compression is defined by two quantization parameters, q for the JPEG encoding of the residual layer and Q for the quantization of the extension layer. For each profile, all possible $\{q, Q\}$ pairs were applied and the best results in terms of rate-distortion were selected. The proposed approach was executed in two versions. In the first version denoted as 'JPEG+PNG' we used PNG encoder as lossless encoder and JPEG as lossy encoder. In the second version denoted as 'x264+PNG' we used PNG encoder as lossless encoder and x264 `veryslow` as lossy encoder. In both cases we utilized the predefined sets of encoding parameters introduced in Section 2.3.

As an objective quality metric we used Signal-to-Noise Ratio (SNR) computed as

$$SNR = 10 \log_{10} \frac{x_{max}^2}{d}, \quad (7)$$

where x_{max} is maximum value in input image \mathbf{x} , and d is mean square error computed as $d = \frac{1}{N} \sum_i (x_i - \hat{x}_i)^2$.

*We would like to thank DroneSystems, Denmark for providing the IR images, Drone1–4.

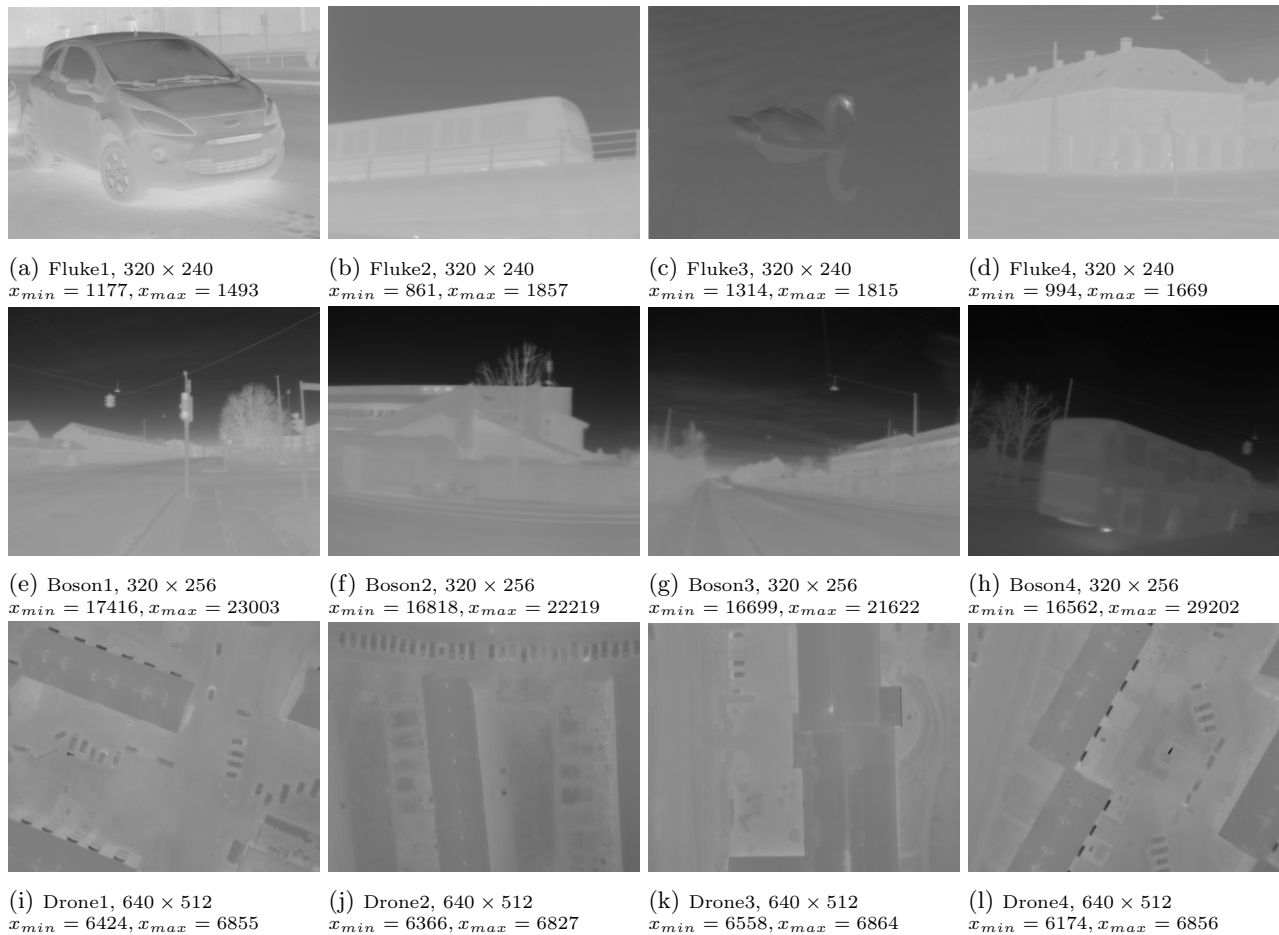


Figure 5: Test infrared images taken by a Fluke Ti32 camera, Bson xxx camera, Drone xxx camera

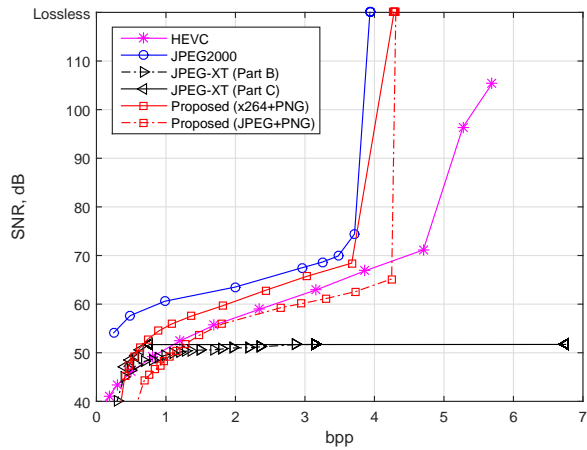
Figures 6,7 show rate-distortion performance of different infrared image coding approaches. One can note that in most cases the proposed 'x264+PNG' version provides similar or better rate-distortion performance compared to HEVC and JPEG-XT Parts B and C, while the proposed 'JPEG+PNG' version shows worse performance compared to both HEVC and JPEG-XT. Finally, JPEG2000 achieves the best performance for all test images.

4. CONCLUSION

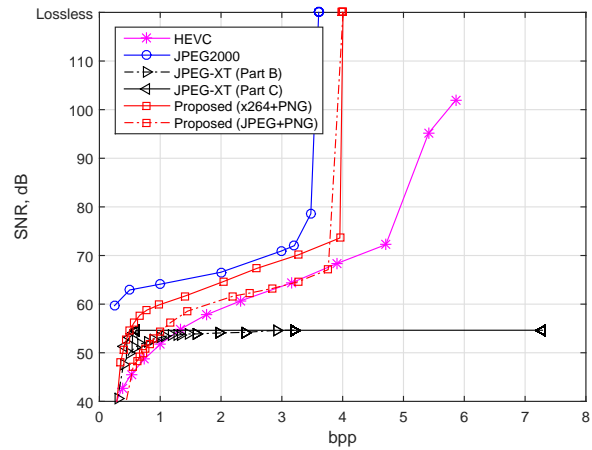
In this paper we presented 16 bit depth infrared image coding based in 8 bit depth image or video codecs. The main advantage of the proposed approach is it provides relatively good rate-distortion performance and enables a high bit depth infrared image coding at the vast majority of existing embedded platforms equipped by 8 bit depth codecs only. This makes it a good candidate for real-time high bit depth infrared image coding systems.

REFERENCES

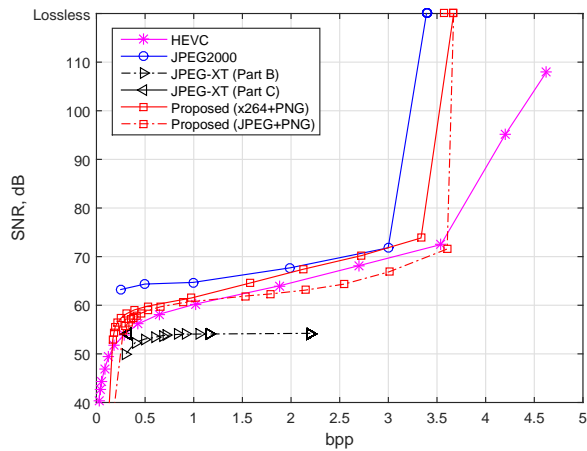
- [1] High Efficiency Video Coding, *document ITU-T Rec. H.265 and ISO/IEC 23008-2*, Oct. 2014 .
- [2] ITU-T and ISO/IEC JTC 1, "JPEG 2000 image coding system: Core coding system, ITU-T Recommendation T. 800 and ISO/IEC 15444-1," JPEG 2000 Part 1, 2000.
- [3] ISO/IEC 18477, "Scalable compression and coding of continuous-tone still images (JPEG-XT)", JTC 1 / SC 29 / WG 1, 2016.
- [4] T.Richter, "Backwards Compatible Coding of High Dynamic Range Images with JPEG", *Data Compression Conference (DCC)*, pp. 153–160, 2013.



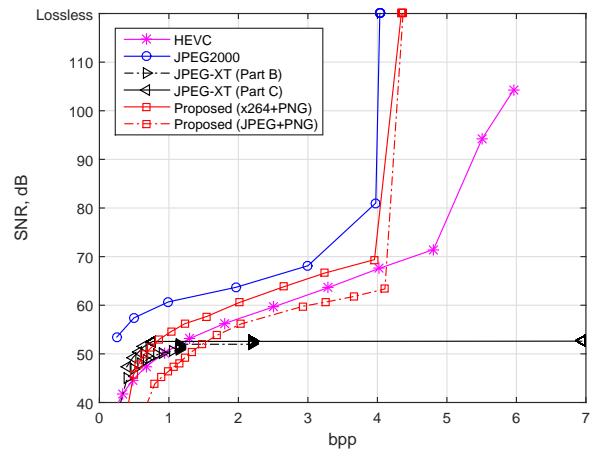
(a) Fluke 1



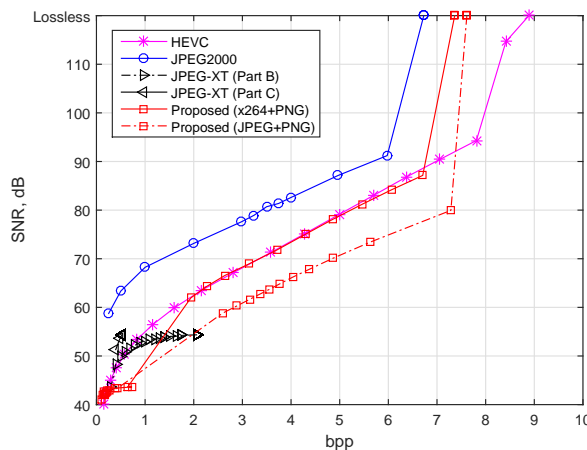
(b) Fluke 2



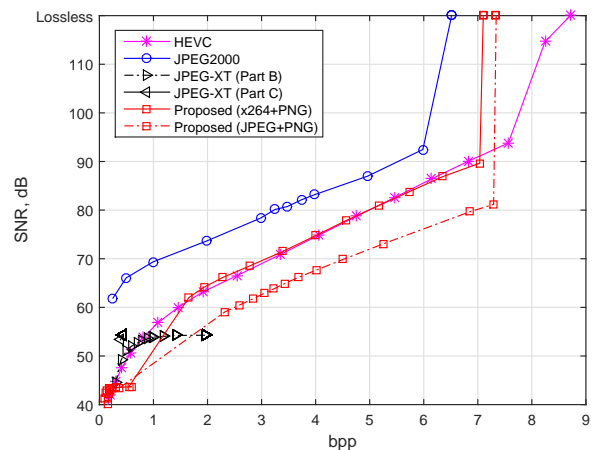
(c) Fluke 3



(d) Fluke 4

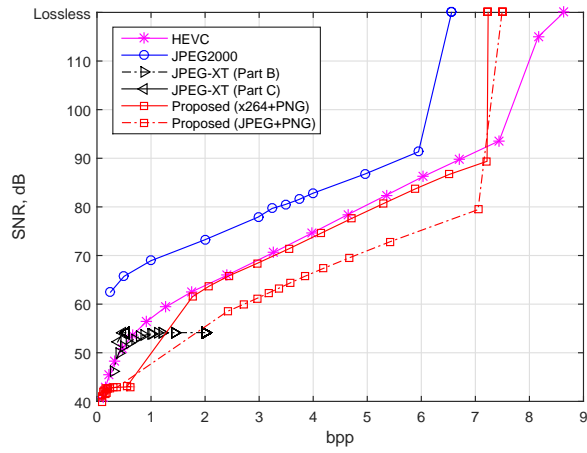


(e) Boson 1

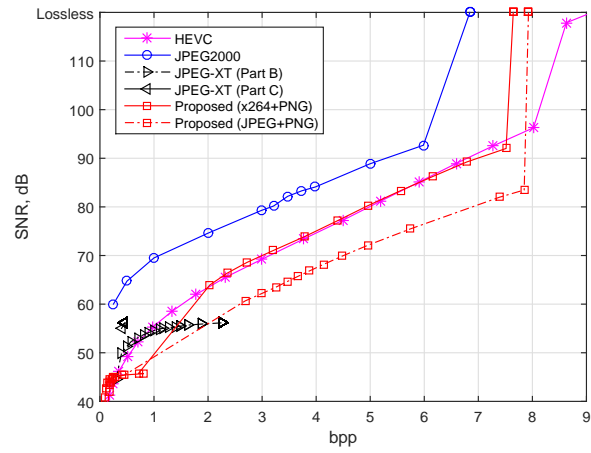


(f) Boson 2

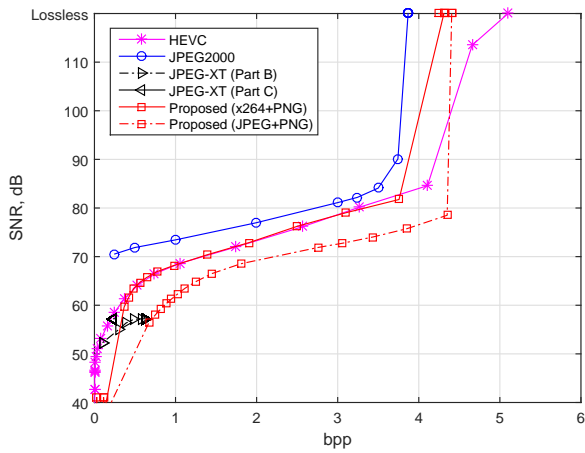
Figure 6: Rate-distortion comparison of different encoding algorithms



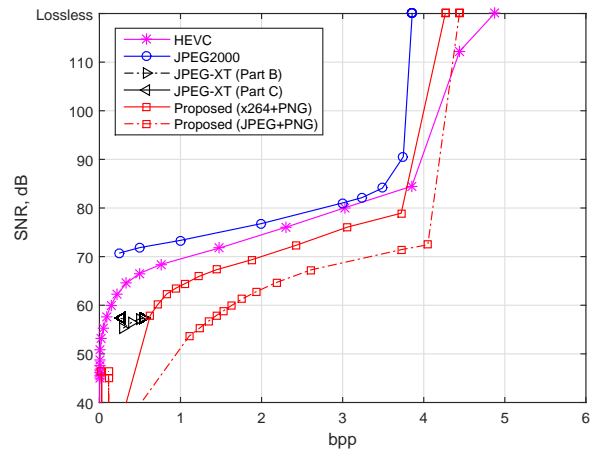
(a) Boson 3



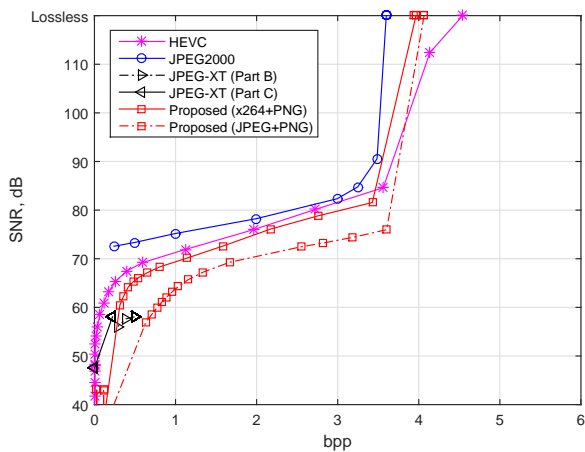
(b) Boson 4



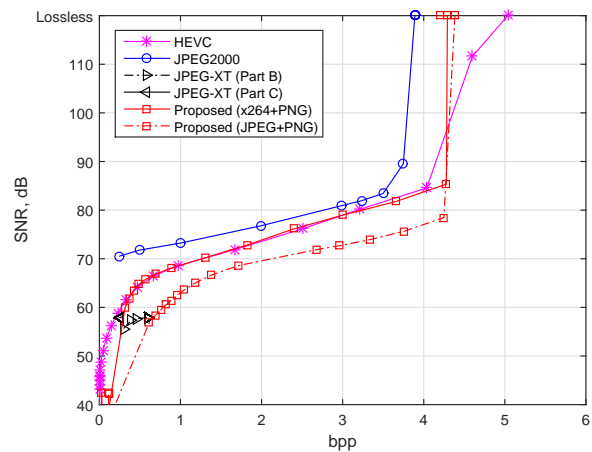
(c) Drone 1



(d) Drone 2



(e) Drone 3



(f) Drone 4

Figure 7: Rate-distortion comparison of different encoding algorithms

- [5] M. Narwaria, M. Silva, P. Callet and R. Pepion, "Tone mapping-based high-dynamic-range image compression: study of optimization criterion and perceptual quality", *Optical Engineering*, vol. 52, no. 10, pp. 102008-1–102008-15, 2013.
- [6] T. Richter, "Rate Allocation in a Two Quantizer Coding System", *Data Compression Conference (DCC)*, pp. 83 – 92, 2014.
- [7] HM Software, https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/
- [8] An open-source JPEG 2000 codec written in C, www.openjpeg.org/
- [9] E. Reinhard, M. Stark, P. Shirley, and J. Ferwerda, "Photographic tone reproduction for digital images," *ACM Trans. Graph.*, vol. 21, no. 3, pp. 267-276, Jul. 2002.