BoostHDR: A novel backward-compatible method for HDR images

Francesco Banterle and Roberto Scopigno[a]

[a] Visual Computing Laboratory, ISTI-CNR, Pisa, Italy

ABSTRACT

In this paper, we present BoostHDR, a novel method for compressing high dynamic range (HDR) images. The algorithm leverages on a novel segmentation-based tone mapping operator (TMO) which relaxes the no seams constraint. Our method can work with both JPEG and JPEG2000 encoders. Moreover, it provides better results compared to the state of the art in HDR images compression algorithms in terms of bit per pixels (bpp), and visual quality using objective metrics.

Keywords: HDR Imaging, HDR Image Compression

1. INTRODUCTION

The HDR content capturing is now becoming very popular, allowing consumers to capture HDR images with compact and DLSR cameras or even mobile phones. Moreover, DSLRs can be used for capturing HDR videos,[12] and HDR video cameras are starting to emerge.[6,19,26] This new era in capturing allows users to represent the full luminance range that the human visual system (HVS) can perceive.

On the other hand, HDR content requires more memory for storing the extra dynamic range information than conventional imaging at 8-bit or low dynamic range (LDR) imaging. For example, an uncompressed HDR pixel, represented using 32-bit floating point, can require four times the amount of memory of an uncompressed LDR pixel at 8-bit. This can negatively affect performances too, because more bandwidth is needed. For instance, it would be prohibitive to manage a photographic gallery of uncompressed HDR images or to play HDR videos.

In recent years, algorithms for HDR content memory compression have been proposed. These methods are typically based on existing compression standards such as JPEG, JPEG2000, and MPEG, which can be modified or extended to handle HDR information. However, the community has not agreed on a common standard encoding yet. This is quite critical, because the HDR imaging has reached the market as extra feature or app for cameras and mobile phones.

In this paper, we present a novel compression algorithm which is based on an segmentation-based TMO, and existing image compression standards such as JPEG and JPEG2000. Our key contributions are:

- *Backward compatibility*: the information in compressed images can be visualized by a JPEG or JPEG2000 standard viewer. This allows users to visualize part of the content when software that can decode our compression algorithm is not present.
- *Improvement over state of the art*: our proposed solution provides better performances in terms of visual quality and bpp than JPEG-HDR compression by Ward and Simmmons,[29,30] JPEG 2000 by Xu and Pattanaik,[32] and Spatial RGBE by Boschetti et al.[5]

The paper is organized as follows. In Section 2, we present an overview on state of the art in HDR images and texture compression. In Section 3, we present our algorithm for HDR images compression. In Section 4, we evaluate our method in terms of visual quality and bpp. Finally, we conclude with limitations and presenting future directions of development in Section 5.

Further author information: (Send correspondence to Francesco Banterle)
Francesco Banterle.: E-mail: francesco.banterle@isti.cnr.it

2. RELATED WORK

In this section, we present an overview on the state of the art in HDR image and texture compression. For a more in depth review on these techniques we refer to Reinhard et al.[21] or Banterle et al.[2]

2.1 Floating Point Formats

An uncompressed HDR image typically requires 96 bpp, which is four times the amount of an uncompressed LDR image. This is because an uncompressed HDR color channel is usually encoded as single precision floating point number. To handle this large amount of data in an efficient way, Ward firstly introduced the RGBE format[31] , a compact and efficient representation of HDR RGB colors. This format stores an 8-bit mantissa for each color channel and a shared 8-bit exponent for all channels, for a total of 32 bpp. Boschetti et al.[5] extended the RGBE encoding scheme for applying it to spatial compression using the JPEG2000 standard.
Another 32 bpp HDR format by Ward, that allows full gamut coverage and fast processing for TMOs, is LogLuv.[10] LogLuv separates luminance and chrominance channels and stores them differently. The former is stored in the logarithm domain using 16 bits, and the latter in the linear domain using the remaining 16 bits. A LogLuv extension by Motra and Thoma[14] exploits image statistics in order to reduce quantization errors and improve the overall quality. This method was also applied to videos with further extensions.[15]
Industrial Light & Magic released their production format, OpenEXR,[8] which has become very popular in the visual effect industry and it is considered the *de facto* standard. This format represents HDR data using 16-bit floating-point per channel for a total of 48 bpp.

2.2 HDR Images Compression

A more compact floating point format can help in reducing memory requirements, but high compression ratios are typically achieved by exploiting spatial coherence and/or removing not perceptible frequencies to the HVS. In the last years, several extensions to standard compression methods such as JPEG and JPEG2000 have been proposed.
Ward and Simmons presented JPEG-HDR[29,30] an extension to JPEG standard to handle HDR images keeping retro-compatibility. This method tone maps an HDR image which is stored using standard JPEG compression. Extra information for recovering HDR data is stored at low resolution in the form of ratio image in the application markers of the standard for a maximum of 64 Kbytes. In order to take into account the lower resolution of the ratio image, the tone mapped image is post-corrected. While JPEG-HDR can compress an HDR image in $0.54 - 3.75$ bpp with JPEG quality settings between 57 and 100, the visual error reaches $0.1 - 4.7\%$ of detectable pixel using VDP[7] with tone mapped images at high JPEG quality settings between 90 and 100. Ward[28] generalized this approach to images and videos for taking into account the reconstruction of high frequencies in the gradient domain.
Li et al.[11] proposed a sub-band architecture for tone mapping and memory compression of HDR images. During the compression stage, an HDR image is firstly tone mapped decomposing the signal in sub-bands, which are separately modified to reduce the dynamic range. Then, the compressed image is stored using a JPEG encoder. There is no need to store extra HDR information, because the process is fully invertible. To improve quality, a feedback loop is introduced to optimize parameters of tone mapping. This method can achieve a bit rate of $1.8 - 4.8$ bpp with a PSNR of $60 - 70$ db using uncompressed HDR images at 12-bit per color channel.
Xu et al.[33] extended JPEG2000 for storing HDR images, exploiting its native support for 16-bit integer values. Their algorithm reduces the range of an input HDR image using a logarithm with base 2. Then, values are remapped in $[0, 2^{16}]$ which are subsequently stored using a JPEG2000 encoder. This compression scheme can achieve a compression rate of $0.48 - 4.8$ bpp, where the visual error reaches $2.3 - 9.6\%$ of detectable pixel using VDP.
Okuda and Adami[17] proposed a compression method based on an analytic inverse tone mapping operator and JPEG. The tone mapping is based on the Hill function and parameters are calculated using a minimization technique. While tone mapped images are encoded using a standard JPEG encoder, residuals, for increasing the quality, are calculated and compressed using wavelets.

Regarding textures, a special image format for computer graphics applications,[1] the main approaches adapt block truncation coding and in particularly S3TC[9] to handle HDR textures. They can leverage on bit operations,[23,24] or color transformation,[?,?] or local block transformations,[25] or histogram LDR parts separations,[27] or analytic inverse tone mapping with minimization processes.[4] All these methods can reach a compression rate of 8 bpp at high visual quality.

3. ALGORITHM

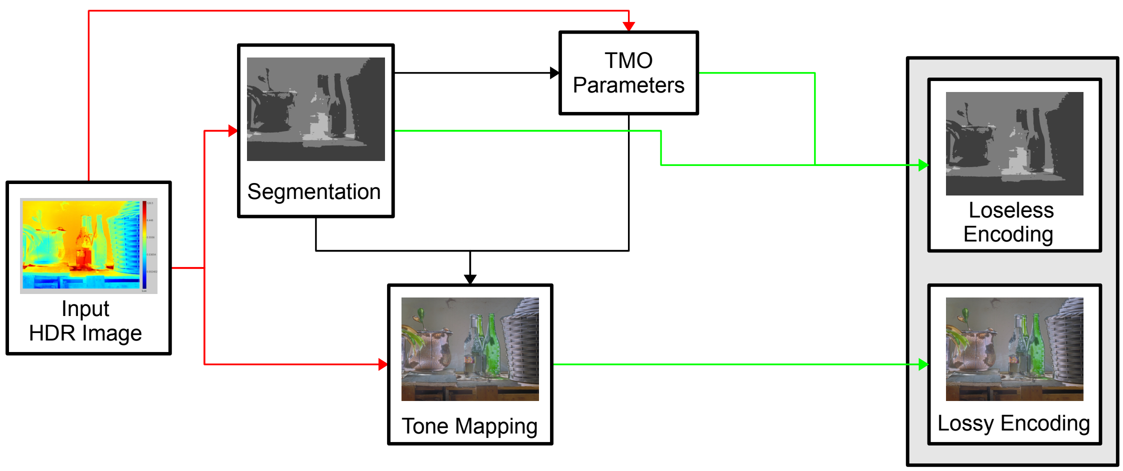


Figure 1. The pipeline for compressing HDR images with our method.

Our approach consists on a novel tone mapping operator which is based on the principle of segmentation operators;[2,21] to segment the image in different areas based on its luminance channel. For each zone of dynamic range, we tone map the image based on its statistics; in our case the average luminance, but other statistics can be employed in order to improve quality. Once the image is tone mapped, we encode it using JPEG or JPEG-2000. Furthermore, we encode the segmentation map, *Compression-Driven Map* (CDM), which segments the image in different areas of dynamic range, and their statistics. The encoding is performed using a loseless compression scheme in order to invert the process during the decoding phase. The full compression scheme of our method is shown in Figure 1.

In our method, we relax a constraint of segmentation TMOs; to avoid seams or halos at the boundary of each segmented zone. While this may produce visual artifacts in the tone mapped image, it increases in the image quality of the HDR images after decompression.

3.1 Compression-Driven Map

The first step of the segmentation algorithm is to extract from the input HDR image the luminance channel, L_{log}, in base 10 logarithm. The luminance is then filtered in order to remove outliers; in order to keep strong edges for segments we used the bilateral filter.[3] Then, each pixel is assigned to an order of magnitude by a floor rounding, this is the first rough segmentation. For each order of magnitude i in the image, the mean luminance $L_{mean,i}$ is calculated. At this point, if two mean luminance values of two segments do not have a distance of 1, they will be merged. This process is iterated until a fixed point is reached. Finally, morphological filtering is applied to remove isolated single pixels and noise.

More sophisticated methods, such as SuperPixels,[22] could be employed for the segmentation. However, the used method was found to produce good results for the compression scheme. The use of SuperPixels could improve the final quality of the tone mapped image with areas following the contour edges in a better way.

The amount of memory required to store the CDM, which is required for decompression, is very small. The encoding of each zone requires a 4-bit unsigned char for a total coverage of 16 levels of magnitude of dynamic range. This is enough for most natural images. In our tests we had on average 3.35 zones per image with a single peak of 8 zones. To further compress the CDM, we apply loseless compression scheme. In our implementation, we used the PNG format in order to have a straightforward access due to the fact it is a popular and well supported format. The CDM can be compressed at around 0.162 bpp, but the use of more efficient compression scheme such as the popular 7z format[18] can compress the CDM down to 0.11 bpp.

(a) (b) (c) (d)

Figure 2. An example of CDM computation: (a) The input HDR image, tone mapped using Reinhard et al.'s operator[20] for visualization purposes. (b) The L_{log} of (a). (c) The rounding on (b). (d) The final CDM after merging segments with a mean distance less than 1. The number of segments in this example is 3.

3.2 A CDM based Tone Mapping Operator

The other important block of our algorithm is the novel TMO. This takes as input the CDM and its associated statistics, which are computed for each different dynamic range zone.

Each dynamic range zone can be encoded differently using different curves such as linear scaling, logarithms, and sigmoids. Nevertheless, we found out in our experiments that a sigmoid for all dynamic range zones was producing satisfying results. For the i-th dynamic range zone, S_i, this is defined as:

$$L_d(\mathbf{x}) = \frac{aL_w(\mathbf{x})}{aL_w(\mathbf{x}) + b} \tag{1}$$

where $L_d(\mathbf{x})$ is the compressed luminance at position $\mathbf{x} \in S_i$, $L_w(\mathbf{x})$ is the HDR luminance value, and a b are the sigmoid parameters. a and b can be automatically computed using optimization techniques[4,13,17] in order to determine the best parameters. On the other hand, this approach can take a lot of time, e.g. up to a minute for a megapixel image. Therefore, we used $a = \frac{L_{mean,i}}{4}$ and $b = 1$ parameters after some experiments. Figure 3 shows a result of the result of tone mapping an HDR image with the described method.

(a) (b) (c)

Figure 3. An example of the proposed TMO: (a) The input HDR image, in false color. (b) The CDM of (a). (c) Our TMO.

Finally, the tone mapped image is encoded using a traditional encoder LDR encoder such as JPEG or JPEG-2000.

3.3 Decoding Stage

The decoding stage is very straightforward. Once the tone mapped image and the CDM are decompressed, each pixel is expanded using inverted sigmoid and the parameters of each luminance zone which are encoded in the CDM. The inverted sigmoid in Equation 1 is used to expand the LDR pixels:

$$L_w(\mathbf{x}) = \frac{bL_d(\mathbf{x})}{a\big(1 - L_d(\mathbf{x})\big)} \tag{2}$$

4. EXPERIMENTAL RESULTS

In this section, we compare our algorithm with other state of the art methods for HDR image compression, such as JPEG-HDR,[30] HDR JPEG2000,[33] and Spatial RGBE.[5] In our evaluation, we assessed the numerical accuracy after decompression of the proposed algorithm. In order to achieve this we used two objective metrics such as: the Root Mean Square Error (RMSE) in the log2[RGB] domain,[33] and the Multi-Exposure Peak Signal Noise Ratio (mPSNR).[16] These are popular quality metrics for testing HDR compression methods.[2]

The first metric, RMSE in the base 2 logarithm domain is defined as:

$$RMSE(I, \hat{I}) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\log_2 \frac{R}{\hat{R}}\right)^2 + \left(\log_2 \frac{G}{\hat{G}}\right)^2 + \left(\log_2 \frac{B}{\hat{B}}\right)^2} \tag{3}$$

where I is the original image, $\hat{I}$ the recovered image I after compression,n is the number of pixels in I, and R, G, B are respectively the red, green and blue channels of I.

The second metric, mPSNR, averages PSNR values of an exposure stack extracted from an HDR image. Each exposure is obtained by applying a simple gamma curve and exposure defined as:

$$T(v, c) = \left[255(2^c v)^{\frac{1}{\gamma}}\right]_0^{255} \tag{4}$$

where c is the current f-stop, v is a color value, and $\gamma = 2.2$. Then, the mPSNR is computed as:

$$MSE(I, \hat{I}) = \frac{1}{n \times p}\sum_{c=1}^{p}\sum_{i=1}^{n}\left(\Delta R_{i,c}^2 + \Delta G_{i,c}^2 + \Delta B_{i,c}^2\right) \tag{5}$$

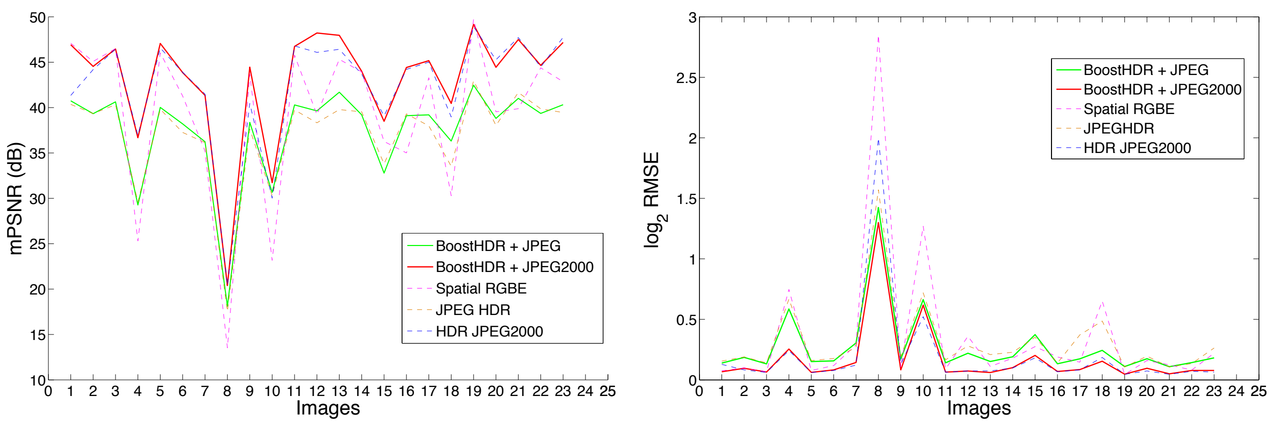
$$mPSNR(I, \hat{I}) = 10\log_{10}\left(\frac{3 \times 255^2}{MSE(I, \hat{I})}\right) \tag{6}$$

where p is the number of exposures (sampling uniformly the dynamic range of the image), and $\Delta R_{i,c} = T(R(\mathbf{x}_i), c) - T(\hat{R}^*(\mathbf{x}_i), c)$ for the red color channel, and so on for the green and blue channels.

In our tests, we had a database of 22 HDR images, these can be seen in Figure 4. We implemented our compression framework with JPEG and JPEG2000 encodings in Matlab R2010b using the HDR Toolbox.[2] Our testing machine was an Intel dual core with 3Gb of RAM on a 32-bit Windows 7.

The results of our comparisons test are showed in graphs in Figure 5. In these tests, we used similar quality setting, 95, for both JPEG-HDR and BoostHDR-JPEG, and the same compression rate, 15, for HDR JPEG2000, BoostHDR-JPEG2000, and Spatial RGBE and JPEG-2000 using 16-bit encoding. The results are summarized in Table 1. As it can be seen from Table 1, our method with JPEG encoding provides better results than JPEG-HDR, with around 0.4 dB more for mPSNR, and 0.035 less in the RMSE error. This is achieved by saving around 0.5 bpp. Regarding the JPEG2000 encoding, our method provided better results of Spatial RGBE and HDR JPEG200. Compared to HDR JPEG2000, the second best method, we achieved nearly 0.7 dB of mPSNR more, and 0.2 less RMSE error. However, our method consumed more memory; around 0.37 bpp. This is due to the fact that using a sigmoid as TMO function can enhance very dark areas, which may increase the file size because seen as important by the encoder. In fact, HDR JPEG2000 uses a linear quantization which does not enhance dark areas.

Figure 4. The dataset of 22 HDR images used in our tests, these images are tone mapped using Reinhard et al.'s operator.[20]



(a) (b)

Figure 5. The compression results for each image in the dataset in Figure 4: (a) the mPSNR results; i.e. higher is better. (b) the RMSE results expressed in residual error; i.e. lower is better.

Method	**mPSNR**	**RMSE**	**bpp**
BoostHDR + JPEG	37.468	0.272	3.149
BoostHDR + JPEG2000	43.122	0.171	3.352
Spatial RGBE	39.230	0.3621	4.246
JPEG-HDR	37.055	0.317	3.687
HDR JPEG2000	42.438	0.198	3.192

Table 1. The overall compression results based on the dataset in Figure 4. mPSNR is expressed in decibel (dB); i.e. higher is better. The RMSE in base 2 logarithm is expressed in residual error; i.e. lower is better.

5. CONCLUSIONS AND FUTURE WORK

We presented BoostHDR, a novel compression scheme for HDR images. This method is based on a segmentation tone mapping operator which can show seams. Our method shows on average higher quality at similar bpp, around 3 bpp, when compared to JPEG-HDR when using JPEG encoder, and Spatial RGBE and HDR JPEG2000 when using a JPEG2000 encoder. The main limitation of the current method is compatibility, which can fail in some cases. This is due to the fact we relaxed the condition of no seams which can create unpleasant images, but they can still be inspected by final users. Another limitation is the use of a sigmoid for all dynamic range zones; this creates an enhancement in dark areas increasing the bpp count without need.

In future work, we would like to apply different compression curves depending on the dynamic range zone. Furthermore, we would like to investigate compression for HDR videos performing an extensive study on temporal coherent TMO curves; i.e. filtering TMO's statistics. In the case of HDR videos, the CDM can be further compressed taking into account temporal coherence leading to a small overhead.

Acknowledgements

We thank Massimiliano Corsini for early development of the system and fruitful discussions. We thank Alessandro Artusi for his precious comments on the draft of the paper. We thank Greg Ward, Kimmo Roimela, Frédéric Drago, Max Lyons, Raanan Fattal, Fredo Durand, Paul Debevec, and ILM Ltd for HDR images used in our paper. Authors are funded by the V-MusT from the European Community's Seventh Framework Programme (FP7 2007/2013) under the Grant Agreement 270404. This work was partially supported by the European COST Action IC-1005.

REFERENCES

[1] Tomas Akenine-Möller, Eric Haines, and Natty Hoffman. *Real-Time Rendering 3rd Edition*. A. K. Peters, Ltd., Natick, MA, USA, 2008.

[2] Francesco Banterle, Alessandro Artusi, Kurt Debattista, and Alan Chalmers. *Advanced High Dynamic Range Imaging: Theory and Practice*. AK Peters, Ltd, first edition edition, 2011.

[3] Francesco Banterle, Massimiliano Corsini, Paolo Cignoni, and Roberto Scopigno. A low-memory, straightforward and fast bilateral filter through subsampling in spatial domain, February 2012.

[4] Francesco Banterle, Kurt Debattista, Patrick Ledda, and Alan Chalmers. A gpu-friendly method for high dynamic range texture compression using inverse tone mapping. In *Proceedings of graphics interface 2008*, GI '08, pages 41–48, Toronto, Ont., Canada, Canada, 2008. Canadian Information Processing Society.

[5] Alberto Boschetti, Nicola Adami, Riccardo Leonardi, and Masahiro Okuda. Flexible and effective high dynamic range image coding. In *ICIP*, pages 3145–3148, 2010.

[6] Alan Chalmers, Gerhard Bonnet, Francesco Banterle, Piotr Dubla, Kurt Debattista, Alessandro Artusi, and Christopher Moir. High-dynamic-range video solution. In *ACM SIGGRAPH ASIA 2009 Art Gallery & Emerging Technologies: Adaptation*, SIGGRAPH ASIA '09, pages 71–71, New York, NY, USA, 2009. ACM.

[7] Scott Daly. The visible differences predictor: an algorithm for the assessment of image fidelity. *Digital images and human vision*, pages 179–206, 1993.

[8] Industrial Light & Magic. OpenEXR. `http://www.openexr.org/`.

[9] Günter Knittel, Andreas G. Schilling, Anders Kugler, and Wolfgang Straßer. Hardware for superior texture performance. *Computers & Graphics*, 20(4):475–481, July 1996.

[10] Gregory Ward Larson. Logluv encoding for full-gamut, high-dynamic range images. *J. Graph. Tools*, 3(1):15–31, 1998.

[11] Yuanzhen Li, Lavanya Sharan, and Edward H. Adelson. Compressing and companding high dynamic range images with subband architectures. *ACM Transaction on Graphics*, 24(3):836–844, 2005.

[12] Magic Lantern. Magic lantern firmware. `http://magiclantern.wikia.com/wiki/Magic_Lantern_Firmware_Wiki`, 2011.

[13] Zicong Mai, Hassan Mansour, Rafał Mantiuk, Panos Nasiopoulos, Rabab Kreidieh Ward, and Wolfgang Heidrich. On-the-fly tone mapping for backward-compatible high dynamic range image/video compression. In *ISCAS*, pages 1831–1834, 2010.

[14] A. Motra and H. Thoma. An adaptive logluv transform for high dynamic range video compression. In *ICIP*, pages 2061–2064, 2010.

[15] Ajit Motra and Herbert Thoma. An adaptive logluv transform for high dynamic range video compression. In *ICIP*, pages 2061–2064, 2010.

[16] Jacob Munkberg, Petrik Clarberg, Jon Hasselgren, and Tomas Akenine-Möller. High dynamic range texture compression for graphics hardware. *ACM Transaction on Computer Graphics*, 25(3):698–706, 2006.

[17] Masahiro Okuda and Nicola Adami. Two-layer coding algorithm for high dynamic range images based on luminance compensation. *J. Vis. Comun. Image Represent.*, 18(5):377–386, 2007.

[18] Igor Pavlov. 7z format. `http://www.7-zip.org/7z.html`, 2012.

[19] RED. Hdrx. In *http://www.red.com/*, 2010.

[20] Erik Reinhard, Michael Stark, Peter Shirley, and James Ferwerda. Photographic tone reproduction for digital images. *ACM Trans. Graph.*, 21(3):267–276, 2002.

[21] Erik Reinhard, Greg Ward, Sumanta Pattanaik, Paul Debevec, Wolfgang Heidrich, and Karol Myszkowski. *High Dynamic Range Imaging, Second Edition: Acquisition, Display, and Image-Based Lighting (The Morgan Kaufmann Series in Computer Graphics)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2010.

[22] Xiaofeng Ren and Jitendra Malik. Learning a classification model for segmentation. In *Proc. 9th Int'l. Conf. Computer Vision*, volume 1, pages 10–17, 2003.

[23] Kimmo Roimela, Tomi Aarnio, and Joonas Itäranta. High dynamic range texture compression. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers*, pages 707–712, New York, NY, USA, 2006. ACM.

[24] Kimmo Roimela, Tomi Aarnio, and Joonas Itäranta. Efficient high dynamic range texture compression. In *SI3D '08: Proceedings of the 2008 symposium on Interactive 3D graphics and games*, pages 207–214, New York, NY, USA, 2008. ACM.

[25] Wen Sun, Yan Lu, Feng Wu, and Shipeng Li. Dhtc: an effective dxtc-based hdr texture compression scheme. In *GH '08: Proceedings of the 23rd ACM SIGGRAPH/EUROGRAPHICS symposium on Graphics hardware*, pages 85–94, Aire-la-Ville, Switzerland, Switzerland, 2008. Eurographics Association.

[26] Jonas Unger and Stefan Gustavson. High dynamic range video for photometric measurement of illumination. In *In Proceedings of Sensors, Cameras and Systems for Scientific/Industrial Applications X, IS&T/SPIE 19th Inernational Symposium on Electronic Imaging*, volume 6501, 2007.

[27] Lvdi Wang, Xi Wang, Peter-Pike Sloan, Li-Yi Wei, Xin Tong, and Baining Guo. Rendering from compressed high dynamic range textures on programmable graphics hardware. In *I3D '07: Proceedings of the 2007 symposium on Interactive 3D graphics and games*, pages 17–24, New York, NY, USA, 2007. ACM.

[28] Greg Ward. A general approach to backwards-compatible delivery of high dynamic range images and video. In *CIC 14th: Proceedings of the Fourteenth Color Imaging Conference*. The Society for Imaging Science and Technology, 2006.

[29] Greg Ward and Maryann Simmons. Subband encoding of high dynamic range imagery. In *APGV '04: Proceedings of the 1st Symposium on Applied perception in graphics and visualization*, pages 83–90, New York, NY, USA, 2004. ACM.

[30] Greg Ward and Maryann Simmons. JPEG-HDR: a backwards-compatible, high dynamic range extension to jpeg. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Courses*, page 2, New York, NY, USA, 2005. ACM.

[31] Gregory Ward. Real pixels. In *Graphics Gems*, volume 2, pages 80–83. Morgan Kaufmann Publishers Inc., 1991.

[32] Ruifeng Xu, Sumanta N. Pattanaik, and Charles E. Hughes. High-dynamic-range still-image encoding in jpeg 2000. *IEEE Computer Graphics and Applications*, 25(6):57–64, 2005.

[33] Ruifeng Xu, Sumanta N. Pattanaik, and Charles E. Hughes. High-dynamic-range still-image encoding in JPEG 2000. *IEEE Comput. Graph. Appl.*, 25(6):57–64, 2005.