FROM 2D EXTRAPOLATION TO 1D INTERPOLATION: CONTENT ADAPTIVE IMAGE BIT-DEPTH EXPANSION

Pengfei Wan, Oscar C. Au, Ketan Tang, Yuanfang Guo, Lu Fang

Department of Electronic and Computer Engineering
Hong Kong University of Science and Technology
Clear Water Bay, Kowloon, Hong Kong
Email: {leoman, eeau, tkt, eeandyguo, fanglu}@ust.hk

ABSTRACT

In this paper, we address the problem of image bit-depth expansion and present a novel method to generate high bit-depth (HBD) images from a single low bit-depth (LBD) image. We expand image bit-depth by reconstructing the least significant bits (LSBs) for the LBD image after it is rescaled to high bit-depth. For image regions whose intensities are neither locally maximum nor minimum, neighborhood flooding is applied to convert 2D interpolation problem into 1D interpolation; for local maxima/minima (LMM) regions where interpolation is not applicable, a virtual skeleton marking algorithm is proposed to convert problematic 2D extrapolation problem into 1D interpolation. At last, a content-adaptive reconstruction model is proposed to obtain the output HBD image. The experimental results show that proposed method significantly outperforms existing methods in PSNR and SSIM without contouring artifacts.

Index Terms— Bit-depth expansion, high dynamic range imaging, virtual skeleton marking

Fig. 1. Bit-depth expansion from 4 bit to 8 bit for pepper image. The top, middle and bottom sub-image is respectively from ground-truth 8-bit image, output of proposed method and output of zero padding. Note false contours as well as chroma distortions are observed in the zero-padded image.

1. INTRODUCTION

Image bit-depth is the number of bits used to represent a quantized pixel value. Nowadays, most images are stored in 8-bit precision, with a dynamic range of order 2 in magnitude. However, luminance levels that human visual system (HVS) can perceive span 12-14 orders in magnitude [1]. As a result high dynamic range imaging (HDRI) draws great attention in recent years [2–4] in multimedia and graphics research.

In this paper we address a HDRI problem, bit-depth expansion from LBD images to HBD images, which is often desirable due to different bit-precisions of photographic devices and image processing systems. For example, incompatibility occurs when 6-bit DVD videos are displayed on 8-bit LCD displays [5], or when 8-bit images are displayed on HDR displays. In the forthcoming video coding standard HEVC, an internal bit-depth increase (IDBI) scheme is also incorporated to improve the precision of data processing [6].

Plenty of methods on bit-depth expansion exist in literature [7–11]. However, contouring artifacts are usually observed in the output HBD images, which greatly degrades the visual quality, see Fig. 1. Though filtering-based methods [8, 10] can reduce contours to some extent, they are not robust to contours with various sizes.

Recently a flooding-based method is proposed by Cheng et al. [9] with encouraging performance. However it fails to reconstruct the LMM regions where interpolation can not be applied. In contrast, our method can effectively and robustly convert problematic 2D extrapolation in LMM regions into 1D interpolation, together with very impressive bit-depth expansion performance, see Fig. 1 and Fig. 2.

This paper is organized as follows. In Section 2 we provide a brief overview of relevant methods. Section 3 elaborates the proposed method and section 4 shows the experi-

This work has been supported in part by the Research Grants Council (RGC) of the Hong Kong Special Administrative Region, China.

978-0-7695-4711-4/12 $26.00 © 2012 IEEE
DOI 10.1109/ICME.2012.118

mental results. Conclusions are given in Section 5.

2. PREVIOUS WORK

Let I_h be the ground-truth HBD image with bit-depth q and I_l be the LBD image with bit depth p ($p < q$). And denote $\hat{I}_h$ as the output HBD image. Note here LBD images are either produced by LSBs truncation of pixel values, or from devices or media with low bit-precision. So conventional inverse tone mapping methods [3, 12], which rely on the knowledge of tone mapping functions, are not applicable for bit-depth expansion problems.

The most straightforward method for bit-depth expansion is zero padding, a.k.a. linear rescaling, where bit '0' is iteratively appended to achieve higher bit-depth:

$$\hat{I}_h^{zp} = I_l \times 2^{q-p} \tag{1}$$

The bit replication method [7] repeats $(b_p, ..., b_1)_2$, the bit representation of I_l, $\lceil \frac{q}{p} \rceil$ times and then take the first q bits as the bit representation of $\hat{I}_h^{br}$.

Cheng's method [9] can be summarized as:

$$\hat{I}_h^{ch}(k) = \hat{I}_h^{zp}(k) + \lfloor SR(k) \times (2^{q-p} - 1) \rfloor \tag{2}$$

$$SR(k) = \frac{DM(k)}{DM(k) + UM(k)} \in [0, 1] \tag{3}$$

Where k is pixel index, SR means step ratio map. DM/UM is respectively called downward/upward distance map, where each entry $DM(k)$/$UM(k)$ records the distance from pixel k to its nearest contour pixel with smaller/larger intensity.

Although Cheng's method can produce smooth HBD images, serious problems exist for LMM regions. In the local maxima regions, UM values are assumed to be infinity since no upward contour exists, so $SR = 0$ and reconstructed LSBs are simply 0. A similar analysis gives the fact that $SR = 1$ for local minima regions with reconstructed LSBs equals to $2^{q-p} - 1$, see Fig. 2(c).

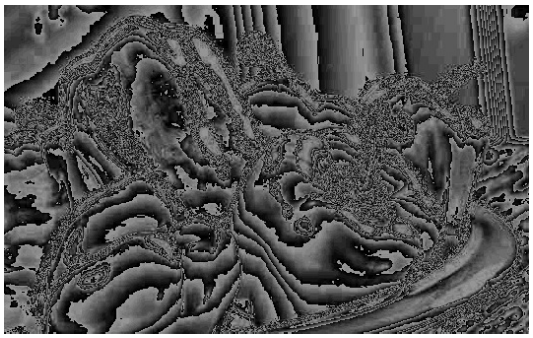
In fact, all above methods can be formulated into one 2D signal reconstruction framework:

$$\hat{I}_h(k) = \hat{I}_h^{zp}(k) + LSB(k) \tag{4}$$

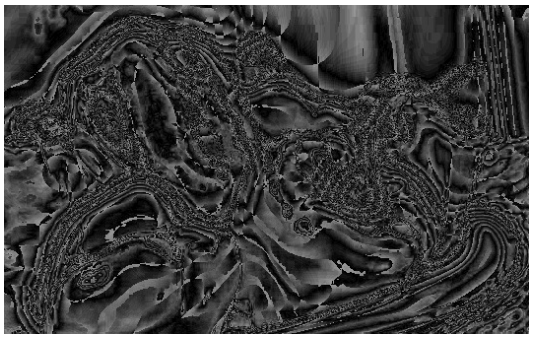
Where LSB is the reconstructed LSBs value for I_l after it is rescaled to $\hat{I}_h^{zp}$. Therefore lost LSBs reconstruction is the key part in bit-depth expansion methods.

3. PROPOSED METHOD

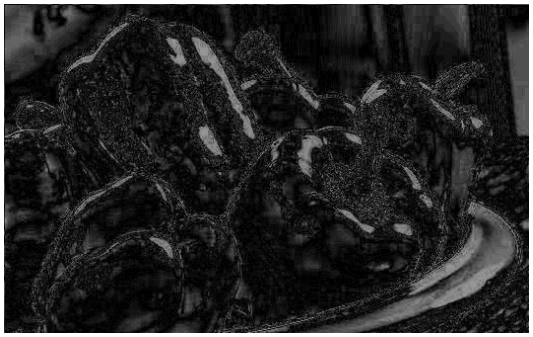
In this section we present our method to reconstruct the lost LSBs by 1D interpolation. Using the same terms as described above, we first obtain SR by neighborhood flooding. LMM regions are then effectively detected by thresholding on SR. For LMM regions, virtual skeletons are marked for converting extrapolation problem into interpolation. At last the LSB is calculated by a content-adaptive reconstruction model. See Fig. 3 for a visual illustration.



(a) Zero padding



(b) Daly's method [8]



(c) Cheng's method [9]

(d) Proposed

Fig. 2. Error maps of selected methods in the case of Fig. 1. Pixel values of error map are proportional to the absolute different between output HBD image $\hat{I}_h$ and ground-truth HBD image I_h.

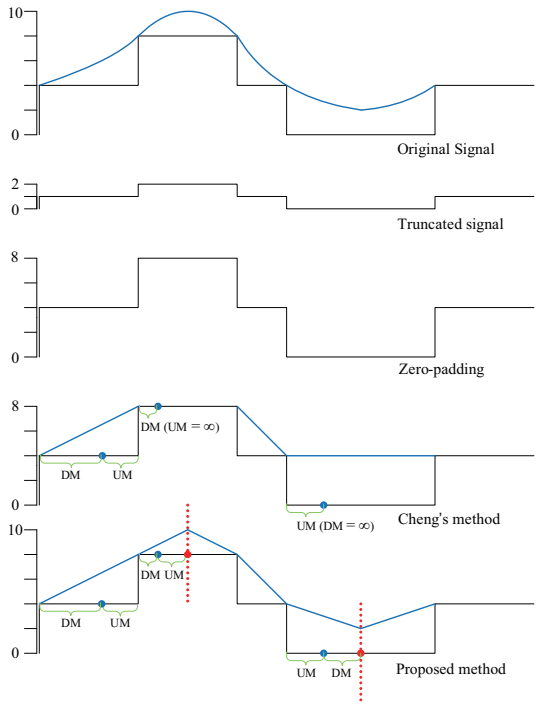


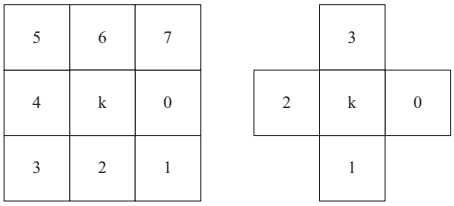
Fig. 3. 1D illustration of selected methods. To reconstruct the original 4-bit signal from its truncated 2-bit version, zero padding introduces severe false contours; Cheng's method can avoid contouring but the information in LMM regions is totally lost. Our method effectively reconstructs the original signal with the aid of LMM virtual skeletons (marked in red).

3.1. Neighborhood Flooding

Within a circle centered at pixel k in I_l, all pixels shall have the same intensity for a sufficient small radius r ($r > 0$). Now expanding r from zero until the circle reaches at least one pixel whose intensity is smaller than $I_l(k)$ but larger than $I_l(k) - \mathcal{T}_e$, then the radius r is the value of downward distance map, i.e., $DM(k)$. Threshold $\mathcal{T}_e$ is to tell false contours from real edges in image. Similarly, the least r to reach at least one pixel whose intensity is in range $I_l(k)$ to $I_l(k) + \mathcal{T}_e$, is the upward distance map value $UM(k)$.

Values of DM and UM can be effectively decided by exchanging information among neighbors, i.e. flooding. In particular, instead of simple 4-neighbor model used in [9], we adopt the 8-neighbor model, see Fig. 4. The combinations of 1 and $\sqrt{2}$ in 8-neighbor model greatly improve the accuracy of DM and UM compared to city block distance used in 4-neighbor model.

Let's denote the absolute pixel index for neigbor j of current pixel k as $\mathcal{N}_k(j)$, and relative distance as $\mathcal{D}(j) \in \{1, \sqrt{2}\}$, where $j = 0, ..., 7$ for 8-neighbor model. Our neighborhood flooding algorithm can then be summarized as follows:



(a) 8-neighbor model (b) 4-neighbor model

Fig. 4. Neighbor models. Numbers in figures are neighbor indices. In (a) the neighbors on diagonal directions have relative distance $\sqrt{2}$ to current pixel k.

1. Initialization step:
   (a) Set $DM = \infty, UM = \infty$ for all pixels.
   (b) For each pixel k, find the following sets for $j = 0, \ldots, 7$:

$$\begin{aligned}\mathcal{S}_k^{dn} &= \{j \mid I_l(k) - I_l(\mathcal{N}_k(j)) \in (0, \mathcal{T}_e)\} \\ \mathcal{S}_k^{up} &= \{j \mid I_l(\mathcal{N}_k(j)) - I_l(k) \in (0, \mathcal{T}_e)\} \\ \mathcal{S}_k^{eq} &= \{j \mid I_l(k) = I_l(\mathcal{N}_k(j))\}\end{aligned}$$

   (c) Set initial values for distance maps:

$$\begin{aligned}DM(k) &= \arg\min_{j \in \mathcal{S}_k^{dn}} \mathcal{D}(j) \\ UM(k) &= \arg\min_{j \in \mathcal{S}_k^{up}} \mathcal{D}(j)\end{aligned}$$

   After initialization, DM and UM have only three possible values: 1, $\sqrt{2}$ and ∞

2. Flooding step:
   (a) For each pixel k, update distance maps according to the following rules:

$$\begin{aligned}DM(k) &= \min(DM(k), \overline{DM}(k)) \\ UM(k) &= \min(UM(k), \overline{UM}(k))\end{aligned}$$

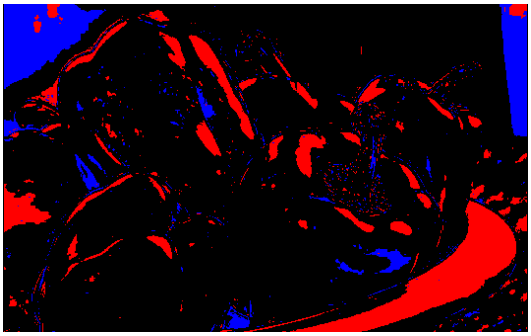
   Where

$$\begin{aligned}\overline{DM}(k) &= \arg\min_{j \in \mathcal{S}_k^{eq}} DM(\mathcal{N}_k(j)) + \mathcal{D}(j) \\ \overline{UM}(k) &= \arg\min_{j \in \mathcal{S}_k^{eq}} UM(\mathcal{N}_k(j)) + \mathcal{D}(j)\end{aligned}$$

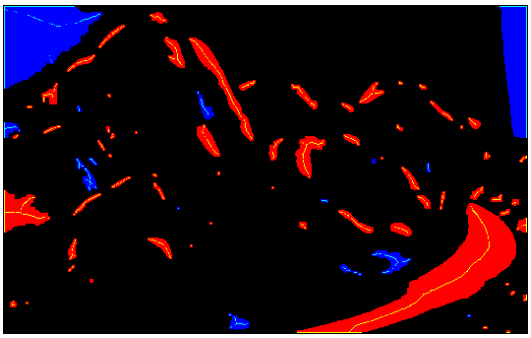
   (b) If update occurs in DM or UM, repeat the flooding step until convergence.

A stable state is reached at the end of neighborhood flooding, together with completed DM and UM. Then $SR(k)$ is calculated by (3), representing the relative position of pixel k on the tightest 1D curve which passes through k and connects k's upward and downward contour. So 2D signal reconstruction becomes 1D interpolation problem in non-LMM regions. Later we will see SR is the only variable in our LSBs reconstruction model.

3.2. LMM Regions Detection and LMM Map Refinement



(a) Original LMM map



(b) Denoised LMM map with virtual skeletons

Fig. 5. LMM maps. Local maxima and minima regions are respectively marked in red and blue, with virtual skeletons in yellow and cyan. Compared with (a), noisy LMM regions are effectively removed in (b), resulting in topologically meaningful LMM regions. Thin but complete virtual skeletons are marked on the refined LMM regions ($\lambda = 2$).

Thanks to the fact that $UM = \infty$ (i.e. $SR = 0$) for local maxima regions and $DM = \infty$ (i.e. $SR = 1$) for local minima regions, a simple thresholding on SR offers accurate detection of LMM regions.

However, original LMM map can be very noisy, as is shown in Fig. 5(a). In order to get a clean and meaningful LMM map for virtual skeleton marking, we apply a multi-pass method as follows:

1. Morphological opening operation to remove tiny LMM regions caused by image noise or random textures.
2. Removal of irregular LMM regions caused by unexpected shading. In Fig. 5(a), they are red/blue spots in blue/red regions. Which are harmful for extracting meaningful structures for LMM regions. Here each LMM pixel is assigned with locally dominant color (red or blue) until convergence.
3. Morphological closing operation to fill small holes in LMM regions caused by the first step.

3.3. Marking Virtual Skeletons for LMM Regions

In this section we present our virtual skeleton marking algorithm for LMM regions. First, we make the following assumptions on virtual skeletons:

- Pixels on virtual skeletons have the highest/lowest intensities respectively in local maxima/minima regions.
- For a LMM region containing no image boundary pixel, pixel intensities are symmetrically distributed along its topological skeleton. So its virtual skeleton is simply the topological skeleton.
- For a LMM region containing image boundary pixels, a mirror object should exists in the scene with the boundary being the symmetry axis. So its virtual skeleton contains both the image boundary pixels and topological skeleton.

Therefore in our method, virtual skeletons contain any LMM pixel k which is on image boundary, or satisfies:

$$\sum_{j=0}^{3} f(M(k)-M(\mathcal{N}_k(j)), M(k)-M(\mathcal{N}_k(j+4))) \geqslant \lambda \quad (5)$$

Where $f(x_1, x_2) = 1$ if and only if $x_1 > 0$ and $x_2 > 0$. Distance map M is DM for local maxima regions and UM for local minima regions. Note neighbor j and neighbor $j + 4$ ($j = 0, \ldots, 3$) are on opposite directions in Fig. 4(a), so λ thresholds the number of directions along which $M(k)$ is locally maximal. In our implementation, $\lambda = 2$ or 3 offers pleasing virtual skeletons, see Fig. 5.

Advantages of our virtual skeleton marking algorithm are two-fold: first, calculations in (5) are independent of geometric properties of LMM regions; second, besides boundary pixels, only LMM pixels exactly on the topological skeletons are marked. Therefore it can guarantee thin but complete virtual skeletons for LMM regions with arbitrary shapes and sizes.

Then DM and UM for LMM regions are updated based on the first assumption. As is shown in Fig. 3, all pixels except those on virtual skeletons now have well-defined SR. As a result, in our method problematic 2D extrapolation in LMM regions becomes 1D interpolation.

3.4. Content-adaptive LSBs Reconstruction

After obtaining the updated step ratio map SR, we are ready for reconstructing the LSBs for $\hat{I}_h^{zp}$. For non-LMM regions where no prior knowledge on image contents is assumed, we reconstruct the LSBs using (2). However, the intensity pattern can be very different in LMM regions. For example, usually the lost LSBs for local minima regions can not reach 0 because of existence of ambient scene light, and lost LSBs for local maxima regions usually do not reach $2^{q-p} - 1$ if they are not super-saturated.

Those facts require us to reconstruct the LSBs according to image contents. In this paper a unified adaptive model is proposed for reconstructing LSBs for $\hat{I}_h^{zp}$. According to (4), the output HBD image is given by:

$$\hat{I}_h(k) = \hat{I}_h^{zp}(k) + \lfloor g(k) \times (2^{q-p} - 1) \rfloor \tag{6}$$

$$g(k) = \begin{cases} SR(k) & \text{if } k \in \mathcal{R}_{nl} \\ \cos(1 - SR(k))^{\alpha} & \text{if } k \in \mathcal{R}_{ss} \\ 0.5 & \text{if } k \in \mathcal{R}_{ad} \\ 0.5 \times SR(k) & \text{if } k \in \mathcal{R}_{max} \\ 0.5 + 0.5 \times SR(k) & \text{if } k \in \mathcal{R}_{min} \end{cases} \tag{7}$$

Where

- $\mathcal{R}_{nl}$ is the set of pixels in non-LMM regions.
- $\mathcal{R}_{ss}$ is the set of pixels in super-saturated regions. A pixel k satisfying $I_l(k) = 2^p - 1$ is recognized as super-saturated. Based on Phong's specular reflection model [13], we use a cosine function with tunable power α to reconstruct the LSBs.
- $\mathcal{R}_{ad}$ is the set of pixels in absolute-dark regions. A pixel k satisfying $I_l(k) = 0$ is recognized as absolute-dark, which usually occurs in RGB images that lack certain color. Expected LSBs value is reconstructed.
- $\mathcal{R}_{max}$/$\mathcal{R}_{min}$ is respectively the set of pixels in the remaining local maxima/minima regions.

Note SR is the only variable in our LSBs reconstruction model which is based on 1D interpolation. After getting a HBD image $\hat{I}_h$, a simple bilateral filtering is applied on virtual skeleton pixels whose SR values are ill-posed (0 or 1). Since the amount of virtual skeleton pixels are very limited, this post-processing is efficient in computation.

In Fig. 2(d), very clean reconstruction error map is produced by our content-adaptive model, in which the reconstruction error is negligible except for very noisy regions where pixel intensity patterns can hardly be modelled.

4. EXPERIMENTS

We verify the effectiveness of proposed method by comparing with a variety of bit-depth expansion methods [7–9], using peak signal-to-noise ratio (PSNR) and the structural similarity index (SSIM [14]) as metrics.

Three sets of experiments (6-bit to 8-bit, 4-bit to 8-bit and 2-bit to 8-bit) are conducted on a set of 8-bit test images. Test images include four gray-scale images and three RGB images. The input LBD images are produced by LSBs truncation of test images. PSNR and SSIM are then calculated between output HBD (8-bit) images and the original test images. See Table. 1 for detailed comparisons.

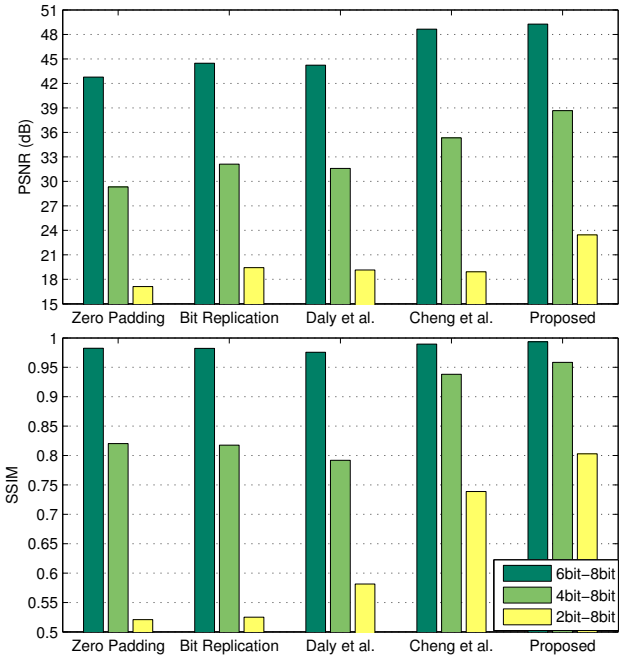


Fig. 6. Mean values of PSNR and SSIM for different methods in different experiments. Note even for a 2-bit image, our method can produce a full 8-bit image with acceptable quality.

As is shown in Fig. 6, remarkably superior performance is achieved by proposed method in our experiments. On average, proposed method gains 0.63, 3.33, 4.01 dB in PSNR and 0.004, 0.020, 0.064 in SSIM over the best of remaining methods for 6-bit, 4-bit, 2-bit to 8-bit expansion respectively. It's also noteworthy that compared with zero padding, our gain is respectively 6.49, 9.35, 6.32 dB in PSNR and 0.011, 0.138, 0.282 in SSIM.

In addition, proposed method has no constrains on the bit-depths of input LBD images and output HBD images. So it can be easily applied on common 8-bit images to generate HDR images with much larger bit-depth (e.g. 12-bit).

5. CONCLUSION

In this paper we propose a novel method to expand the image bit-depth with high accuracy without contouring artifacts. Our main contribution is the virtual skeleton marking algorithm, which robustly converts the problematic 2D extrapolation in LMM regions into simple 1D interpolation.

6. REFERENCES

[1] R. Mantiuk, G. Krawczyk, K. Myszkowski, and H. Seidel, "Perception-motivated high dynamic range video encoding," in *ACM SIGGRAPH*, 2004, pp. 733–741.

Table 1. Performance Comparisons in PSNR (dB) and SSIM [14]

		Zero padding		Bit Replication [7]		Daly et al. [8]		Cheng et al. [9]		Proposed	
Setup	Image	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
6-bit to 8-bit	pepper_g	42.68	0.9876	44.70	0.9872	44.17	0.9773	48.18	0.9935	**48.35**	**0.9938**
	pepper_c	42.82	0.9784	44.01	0.9782	44.13	0.9714	45.94	0.9682	**47.72**	**0.9901**
	teapot_g	42.66	0.9875	44.53	0.9871	44.07	0.9748	49.68	0.9948	**49.80**	**0.9951**
	teapot_c	42.68	0.9876	44.23	0.9873	44.08	0.9759	49.49	0.9947	**49.61**	**0.9950**
	kid_g	42.99	0.9828	45.32	0.9824	44.18	0.9750	48.02	0.9919	**48.22**	**0.9923**
	kid_c	42.89	0.9811	44.91	0.9807	44.22	0.9744	47.99	**0.9914**	**48.12**	0.9907
	synth_g	42.73	0.9724	43.55	0.9723	44.82	0.9807	51.16	0.9916	**53.06**	**0.9970**
4-bit to 8-bit	pepper_g	29.04	0.8927	32.40	0.8887	31.68	0.8383	35.42	0.9647	**38.13**	**0.9702**
	pepper_c	29.45	0.8551	31.98	0.8532	31.63	0.8069	31.68	0.8701	**36.27**	**0.9009**
	teapot_g	29.12	0.9060	32.42	0.9051	31.53	0.8442	37.42	0.9777	**38.14**	**0.9797**
	teapot_c	29.29	0.9017	32.09	0.9013	31.49	0.8497	37.57	**0.9766**	**38.03**	0.9762
	kid_g	29.40	0.7322	32.32	0.7272	32.02	0.7788	35.22	0.9294	**38.34**	**0.9579**
	kid_c	29.42	0.7359	32.35	0.7314	32.08	0.7815	34.47	0.9196	**38.32**	**0.9586**
	synth_g	29.50	0.7190	31.21	0.7161	30.69	0.6440	35.57	0.9298	**43.43**	**0.9650**
2-bit to 8-bit	pepper_g	16.52	0.6611	19.46	0.6626	19.38	0.6966	20.42	0.8043	**24.18**	**0.8463**
	pepper_c	17.02	0.6041	19.27	0.6084	19.19	0.6523	16.83	0.6711	**22.08**	**0.7281**
	teapot_g	16.18	0.5194	19.08	0.5321	19.04	0.6047	20.54	0.8119	**23.09**	**0.8684**
	teapot_c	16.31	0.4674	18.85	0.4757	18.78	0.5683	20.21	0.7943	**22.96**	**0.8496**
	kid_g	17.85	0.5530	20.64	0.5561	19.42	0.6315	18.61	0.6950	**22.46**	**0.7438**
	kid_c	17.79	0.5270	19.58	0.5285	19.14	0.5996	18.48	0.6911	**23.09**	**0.7394**
	synth_g	18.17	0.3140	19.16	0.3117	19.09	0.3169	17.39	0.7039	**26.23**	**0.8450**

$*$ *image names ending with _g and _c stand for gray-scale images and RGB color images respectively*

[2] P. E. Debevec and J. Malik, "Recovering high dynamic range radiance maps from photographs," in *ACM SIGGRAPH 2008 classes*, 2008, pp. 31:1–31:10.

[3] F. Banterle, P. Ledda, K. Debattista, and A. Chalmers, "Inverse tone mapping," in *GRAPHITE '06*, 2006, pp. 349–356.

[4] G. W. Larson, H. Rushmeier, and C. Piatko, "A visibility matching tone reproduction operator for high dynamic range scenes," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 3, no. 4, pp. 291–306, 1997.

[5] R. Mantiuk, A. Efremov, K. Myszkowski, and H. Seidel, "Backward compatible high dynamic range mpeg video compression," in *ACM SIGGRAPH*, 2006, pp. 713–723.

[6] "Meeting report of the fourth meeting of the joint collaborative team on video coding, daegu, kr," *Joint Collaborative Team on Video Coding*, 20-28 January, 2011.

[7] R. Ulichney and S. Cheung, "Pixel bit-depth increase by bit replication," *Color Imaging: Device Independent Color. Proc. of SPIE*, vol. 3300, pp. 232–241, 1998.

[8] S. Daly and X. Feng, "Decontouring: Prevention and removal of false contour artifacts," in *Proc. SPIE Human Vision and Electronic Imaging IX*, 2004, vol. 5292, pp. 130–149.

[9] C. H. Cheng, O. C. Au, C. H. Liu, and K. Y. Yip, "Bit-depth expansion by contour region reconstruction," in *Proc. of IEEE Int. Sym. on Circuits and Systems*, 2009, pp. 944–947.

[10] C. H. Liu, O. C. Au, H. W. Wong, M. C. Kung, and S. C. Chao, "Bit-depth expansion by adaptive filter," in *Proc. of IEEE Int. Sym. on Circuits and Systems*, 2008.

[11] M. Winken, D. Marpe, H. Schwarz, and T. Wiegand, "Bit-depth scalable video coding," in *Proc. IEEE Int. Conf. Image Processing*, 2007, vol. 1, pp. I–5–I–8.

[12] F. Banterle, P. Ledda, K. Debattista, and A. Chalmers, "Expanding low dynamic range videos for high dynamic range applications," in *the 24th Spring Conference on Computer Graphics*, 2010, pp. 33–41.

[13] B. T. Phong, "Illumination for computer generated pictures," *Commun.ACM*, vol. 18, no. 6, pp. 311–317, June 1975.

[14] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *Image Processing, IEEE Transactions on*, vol. 13, no. 4, pp. 600–612, 2004.