Deep Reverse Tone Mapping

YUKI ENDO, University of Tsukuba
YOSHIHIRO KANAMORI, University of Tsukuba
JUN MITANI, University of Tsukuba

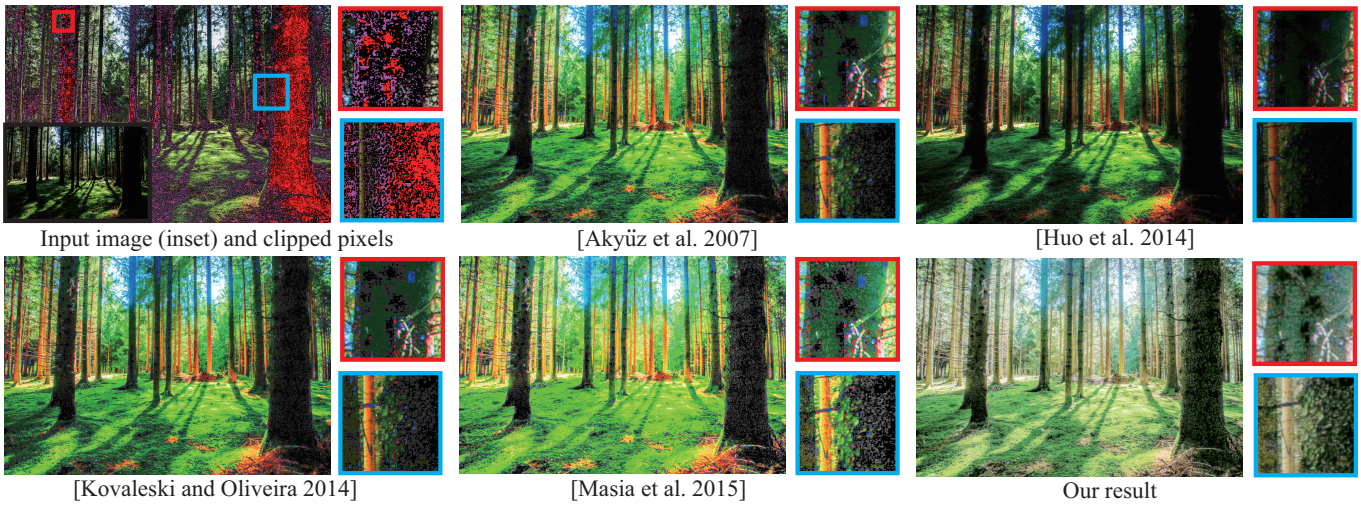

Fig. 1. Comparison of reverse tone mapping operators (rTMOs). A single low dynamic range (LDR) image (top left) is expanded to a high dynamic range (HDR) image using each rTMO and visualized using the consistent tone mapping operator [Kim and Kautz 2008]. While we can observe amplified noise in the dark regions of other results, our result exhibits natural tone reproduction and the least noise. The clipped pixels in the input image are painted in red or magenta if all three channels (or at least one channel) are underexposed, respectively.

Inferring a high dynamic range (HDR) image from a single low dynamic range (LDR) input is an ill-posed problem where we must compensate lost data caused by under-/over-exposure and color quantization. To tackle this, we propose the first deep-learning-based approach for fully automatic inference using convolutional neural networks. Because a naive way of directly inferring a 32-bit HDR image from an 8-bit LDR image is intractable due to the difficulty of training, we take an indirect approach; the key idea of our method is to synthesize LDR images taken with different exposures (i.e., *bracketed images*) based on supervised learning, and then reconstruct an HDR image by merging them. By learning the relative changes of pixel values due to increased/decreased exposures using 3D deconvolutional networks, our method can reproduce not only natural tones without introducing visible noise but also the colors of saturated pixels. We demonstrate the effectiveness of our method by comparing our results not only with those of conventional methods but also with ground-truth HDR images.

CCS Concepts: • **Computing methodologies** → **Image processing**;

Authors' addresses: Yuki Endo, University of Tsukuba, endo@cs.tsukuba.ac.jp; Yoshihiro Kanamori, University of Tsukuba, kanamori@cs.tsukuba.ac.jp; Jun Mitani, University of Tsukuba, mitani@cs.tsukuba.ac.jp.

0730-0301/2017/11-ART177 $15.00
https://doi.org/10.1145/3130800.3130834

Additional Key Words and Phrases: Reverse tone mapping; high dynamic range (HDR) imaging; convolutional neural networks

ACM Reference Format:
Yuki Endo, Yoshihiro Kanamori, and Jun Mitani. 2017. Deep Reverse Tone Mapping. *ACM Trans. Graph.* 36, 6, Article 177 (November 2017), 10 pages. https://doi.org/10.1145/3130800.3130834

1 INTRODUCTION

High dynamic range (HDR) images can convey much richer contrasts, like those found in the real world, than can conventional low dynamic range (LDR) images. HDR techniques have advanced in recent decades, as summarized in a broad review [Mantiuk et al. 2015]. For example, the growing demand for HDR imaging has driven the ongoing development of HDR displays as well as special devices for capturing HDR images or videos. Unfortunately, high-quality HDR cameras are still unaffordable. Meanwhile, photographers have long been accustomed to creating HDR images from multiple LDR images taken with different exposures (i.e., *bracketed images*). A recent, more easy-to-use alternative is to infer an HDR image from a single LDR image.

Single-image HDR inference is referred to as *reverse tone mapping* (or *inverse tone mapping*). Reverse tone mapping has great potential to transform the large amount of legacy LDR content into HDR, ready for enhanced viewing on HDR displays and various applications such as artistic photo editing and image-based lighting with HDR environment maps. Previous methods of reverse tone

mapping employ individual heuristics or optionally use manual intervention to enhance LDR images. Unfortunately, their HDR outputs often significantly deviate from true (i.e., optically-photographed) HDR images. This is because existing methods cannot sufficiently compensate missing information caused by under-/over-exposure and color quantization.

To challenge such an ill-posed problem, we introduce the first deep-learning-based approach to automatically infer a statistically-plausible HDR image from a single LDR input. We employ supervised learning using deep convolutional neural networks (CNNs). A straightforward way is to learn an end-to-end mapping from an 8-bit LDR image directly to a 32-bit HDR image. However, training such neural network models is quite difficult for the following reasons. First, identical HDR images should be output consistently even if input LDR images of the same scene were photographed with different exposures. Second, because HDR pixels have much richer variations than LDR pixels, i.e., $[0, 3.4 \times 10^{38}]$ (`float`) vs [0, 255] (`unsigned char`) per channel, an enormous amount of training data might be required to cover the mapping. However, there are far fewer publicly available HDR image datasets than there are LDR datasets in widespread use for conventional learning tasks. Even worse, subtle changes in loss can affect the output, making training of such models unstable. We discuss the difficulty of the direct approach in Section 6.

Thus, instead of the direct approach, we adopt an indirect one; our key idea is to synthesize bracketed images, that is, to infer a sequence of LDR images with k different exposures, and then reconstruct an HDR image by merging the LDR images [Debevec and Malik 1997]. This approach drastically reduces pixel variations to be inferred from 32-bit floating point values to $k \times 2^8$ (i.e., k intermediate LDR images). Specifically, our model learns the relative changes of each pixel value with increased/decreased exposures by using 3D deconvolutional networks, from a training dataset of synthesized bracketed images. Our method boosts the gradations and thus reproduces more natural tones than previous methods.

Our contributions are as follows:

- The first deep-learning-based framework for automatically inferring an HDR image from a single LDR input, and
- A network model designed to infer bracketed images with increased/decreased exposures, as well as a description of how to generate its training data.

We demonstrate that our method can reproduce HDR images more faithfully than conventional methods as shown in Figure 1.

2 RELATED WORK

The most well-known approach to HDR photography is to merge multiple LDR photographs taken with different bracketing exposures to synthesize a single HDR image [Debevec and Malik 1997; Mann et al. 1995]. While this approach was designed to target static scenes, it has been extended to handle dynamic scenes while reducing ghosting and tearing artifacts (see, e.g., [Kalantari and Ramamoorthi 2017]). To enable single-shot HDR photography, several devices have been proposed such as cameras with multiple sensors [Kronander et al. 2013; Tocci et al. 2011] and off-the-shelf cameras with coded exposures [Serrano et al. 2016; Zhao et al. 2015]. Whereas these approaches require multiple input images of the same scene or special devices, our input is only a single LDR image, which can be obtained much more easily from, for instance, cameras on mobile phones or the Internet.

Reverse tone mapping. Transforming a single LDR image to an HDR image is a relatively new topic in the field of computer graphics. This transformation is achieved by expanding the contrast range of the LDR image. This operation has often been referred to as a *reverse tone mapping operator* (rTMO) (or *inverse tone mapping operator*). Constructing an rTMO is an ill-posed problem because information is missing due to under-/over-exposure and color quantization for limited contrast ranges of LDR images.

In the early stage of the development of this technique, Banterle et al. [2006; 2008; 2007] proposed rTMOs as approximate inversion of the tone mapping operator by Reinhard et al. [2002]. They also create an expand map using density estimation of the light source to enhance the brightness of an LDR image. Rempel et al. [2007] focused on online rTMOs for videos that can be integrated directly into display hardware. After the contrast stretching of the input image, their method enhances the brightness in the saturated regions by not only blurring images but also preserving strong edges using Gaussian and edge stopping functions. Kovaleski and Oliveira accelerated their method using the faster edge-preserving filtering [2009] and further improved the technique to support a wide range of exposures robustly [2014]. Akyüz et al. [2007] presented a simple linear expansion method. They conducted psychophysical experiments and found that such simple operations often outperform a true HDR image for HDR display. Masia et al. [2009; 2015] also proposed simple expansion operators based on a gamma curve, the gamma value of which is determined automatically via regression. Huo et al. presented two algorithms based on the dodging and burning approach [2013] and the physiological approach [2014] for the luminance channel. Wang et al. [2015] segmented an input LDR image into four regions based on the accumulated histogram, and enhanced each region separately. These methods cannot reproduce missing details in clipped regions.

Whereas all of the above solutions are automatic, with the exception of parameter tuning, there are also interactive techniques. The method by Didyk et al. [2008] classifies saturated regions as light, reflections, or diffuse surfaces using a classifier based on user markups. After that, the different expansion functions are applied to the classified regions. Masia et al. [2010; 2016] also presented an interactive approach wherein the user adjusts regional tonal balance of the final HDR image by using a piecewise linear function.

To enrich badly exposed regions in expanded LDR images, texture details are transferred from appropriately exposed patches in the input image [Wang et al. 2007] or images obtained from the Internet [Jain et al. 2014; Savoy et al. 2014]. Although these methods can generate appealing results, they require the user to select reference images of very similar scenes and to annotate target regions. Our method does not require user markups.

Concurrently, Zhang and Lalonde [2017] proposed a deep-learning approach to directly infer HDR from a single LDR image, designed specifically for daytime outdoor panoramas in sunlight. They explicitly incorporate sun elevation in the loss function, assuming

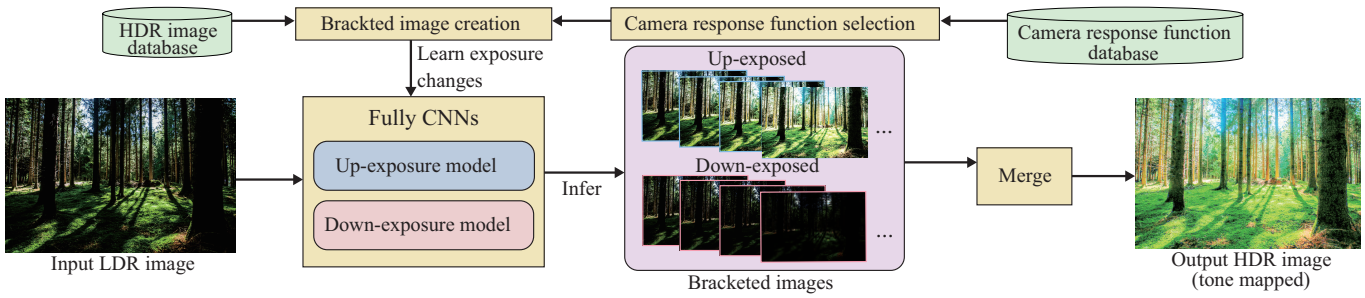

Fig. 2. Overview of the proposed method. The flow is decomposed into the learning and inference phases. In the learning phase, the bracketed LDR images are first created from HDR databases by simulating cameras. Next, we let our fully CNNs learn the changes in the exposures of the bracketed images. In the inference phase, the learned CNNs compute LDR images with different exposures from a single input LDR image. The final HDR image is then generated from these bracketed LDR images.

that sun azimuth is fixed. Their method plausibly reproduces the extremely high dynamic range caused by the sun. However, reconstructing fine details of scenes is difficult from their small resolution inputs (e.g., 64×128). In contrast, our method is more versatile; it handles not only daytime outdoor images but also night and indoor images, among others, of much larger resolution (e.g., 512×512). Eilertsen et al. [2017] also adopted a deep-learning approach. Their proposed CNN, which consists of an encoder and decoder that operate in the logarithm domain, can directly create a high-quality HDR image from an LDR image containing saturated areas. While one of their limitations is the difficulty in recovering dark regions, our method handles both dark and bright regions using the indirect approach.

3 ALGORITHMIC PIPELINE

Figure 2 illustrates the overall flow of our pipeline. As introduced in Section 1, we reconstruct an HDR image from a single LDR input indirectly by inferring bracketed LDR images and merging them. The flow is decomposed into the learning and inference phases. In the learning phase, the bracketed LDR images are first created from HDR databases by simulating cameras (Section 3.1). Next, we let our neural network models (i.e., up-/down-exposure networks) learn the changes in the exposures of the bracketed images (Section 4). In the inference phase, the learned models compute LDR images with different exposures from a single input LDR image. Brighter/dimmer bracketed images are inferred in our up-/down-exposure networks, respectively. The final HDR image is then generated from these bracketed LDR images (Section 3.2).

3.1 Creating Bracketed Images for Training

Our training dataset consists of ground-truth HDR images and corresponding sets of bracketed LDR images. To account for color variations in LDR images caused by different non-linear camera response functions (CRFs), we synthesize a set of LDR images with different CRFs and exposures from each HDR image. Toward this, we simulate cameras using the following equation [Debevec and Malik 1997]:

$$Z_{i,j} = f(E_i \Delta t_j), \tag{1}$$

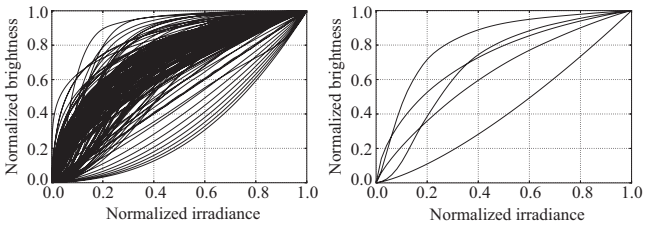

Fig. 3. Camera response curves for creating training data. Using all response curves in the database [Grossberg and Nayar 2003] is redundant (left), so we choose five representative curves using k-means clustering (right).

where $Z_{i,j}$ denotes pixel values for each pixel i and exposure duration index j; f, E_i, and Δt_j denote CRFs, film irradiance values, and exposure durations, respectively. In this paper, $Z_{i,j}$ and E_i represent our LDR and HDR images.

To define CRFs, we use Grossberg and Nayar's Database of Response Functions (DoRF) [Grossberg and Nayar 2003]. This database consists of 201 response curves for common brands of films, charge-coupled devices (CCDs), and digital cameras collected by the authors. All of the CRFs in the database are monotonic, normalized in the range of $[0, 1]$, and sampled with 1,000 points. All of the CRFs are shown in the left panel of Figure 3. However, using all of the CRFs is redundant and unnecessarily increases the training time. We thus use only representative CRFs selected using k-means clustering (Figure 3, right). In our experiments, we use five CRFs interpolated using spline interpolation.

Because the CRFs in DoRF are normalized, we must determine the absolute standard for E_i and Δt_j. For example, if the range of E_i is from 0 to 1,000 and Δt_j is 1, most pixels of $Z_{i,j}$ might become clipped whites. Clipped regions are almost useless for learning exposure-based changes, and should thus be avoided. We adjust the range of the observed signal $E_i \Delta t_j$ appropriately. Specifically, we set Δt_j as a power of τ between stops

$$\Delta t_j = \frac{1}{\tau^{T/2}}, ..., \frac{1}{\tau^2}, \frac{1}{\tau}, 1, \tau, \tau^2, ..., \tau^{T/2}, \tag{2}$$

Fig. 4. Examples of bracketed images created from the HDR datasets. Exposures are selected to avoid completely white or black regions.

where T is an even integer and $j = 1, 2, \ldots, T+1$. We then normalize $E_i \Delta t_j$ so that the average pixel value of $E_i \Delta t_{T/2+1}$ ($= E_i$ because $\Delta t_{T/2+1} = 1$) equals 0.5. In our experiment, we used $T = 8$ and $\tau = \sqrt{2}$. Figure 4 shows examples of the synthesized bracketed images.

Although the normalization of E_i and the choice of Δt_j determine the dynamic range of the inferred HDR image, linear scaling of E_i or Δt_j can be compensated by linear scaling of the inferred HDR values (see Appendix A for details). Therefore, if the inferred HDR image is a little too dim or bright, the user can adjust it by linearly scaling pixel values.

3.2 Merging Inferred Bracketed Images

Feeding an input LDR image to our model, our up-exposure network outputs N up-exposed images while our down-exposure network outputs N down-exposed images (Section 4). We thus have $2N + 1$ LDR images including the input image, and we can choose k (up to $2N + 1$) images to construct an HDR image.

We must choose the inferred LDR images carefully. If the input LDR image is already bright/dim, inferred LDR images with too high/low exposures, respectively, should be avoided because they tend to contain artifacts, mainly due to the lack of training data. For example, inferred LDR images with exposures that are too high or low contain erroneously bright/dim pixels, respectively. To avoid such erroneous LDR images, we adopt the following heuristic. Starting from the input LDR image, we accept the $(j + 1)$-th brighter/dimmer LDR image until each pixel value v_{j+1} in each color channel is larger/smaller than v_j or the absolute difference $|v_{j+1} - v_j|$ is smaller than a certain tolerance η. We set $\eta = 64$ for 256 levels.

To combine the inferred bracketed LDR images, we can use several merging methods [Debevec and Malik 1997; Mertens et al. 2007]. The existing method [Debevec and Malik 1997] requires an exposure duration for each bracketed image. The exposure duration of the input image can be obtained from its Exif data, and that of up-/down-exposed images can be computed by multiplying the input exposure duration by a factor of τ (or $\frac{1}{\tau}$). If Exif data is not available, the user can set a fixed value for τ, and linearly scale the HDR values later, as mentioned in the previous section and Appendix A. We can also use the method by Mertens et al. [2007], which does not require exposure durations as input, to generate a tone-mapped LDR image without generating an HDR image.

4 UP-/DOWN-EXPOSURE MODELS

In this section we describe our neural network architecture, which increases/decreases the exposure of an input LDR image. We build two models for up- and down-exposure using the same architecture. Assuming that our up-/down-exposure networks learn different features, we train them separately.

4.1 Architecture

Figure 5 shows the overall architecture of our network. The input to the network is a $W \times H \times c$ LDR image where W, H, and c are width, height, and the number of color channels. We basically used 512×512×3 RGB images as input for training our network. In the inference phase, larger images can also be given to the network as input. The pixel values of the input image are normalized in the range of $[0, 1]$. The proposed network starts by encoding an input LDR image into latent semantic features using 2D convolutional neural networks; the results are $\frac{W}{2^l} \times \frac{H}{2^l} \times c_l$ three-dimensional tensors on the l-th convolutional layer. In the last layer of the encoder, the input image is reduced to one pixel; a flattened 512-dimensional latent vector is obtained. Next, the network decodes the semantic features into LDR images taken with different exposure durations using 3D deconvolutional neural networks; the output is a $N \times W \times H \times c$ four-dimensional tensor, where N denotes the number of exposure durations. As described in Section 3.1, the up-/down-exposure models sequentially increase/decrease the camera exposure of the input photograph by τ (or $\frac{1}{\tau}$) times, that is, τ, τ^2, ... (or $\frac{1}{\tau}$, $\frac{1}{\tau^2}$, ...), by learning the changes of the exposures of the bracketed images in the database. In the following, the architectures of the CNNs are described in more detail.

Encoder. For the encoder, we adopt an architecture similar to the pix2pix model by Isola et al. [2016], with the exception of the size of the input image and the number of layers. Specifically, the encoder consists of nine layers of 4×4 convolution with a stride of (2, 2) and spatial padding of (1, 1). From the first layer to the last layer, the numbers of filter kernels (i.e., the numbers of output channels) is 64, 128, 256, 512, 512, 512, 512, 512, and 512, respectively. In the second and subsequent layers, batch normalization [Ioffe and Szegedy 2015] is applied to the convolved output to improve the learning in the networks by normalizing the distribution of each input feature. In our network, the batch normalization normalizes an input batch by using only its statistics both in the learning and inference phases. The activation function in each layer is the leaky ReLU function [Maas et al. 2013].

Decoder. After the 2D CNN, the decoder uses convolved features as input to generate an $N \times W \times H \times c$ four-dimensional tensor that consists of N images with different exposure. To generate consistent images with different exposure, we employ 3D deconvolution neural networks. The 3D CNNs are extensions of 2D CNNs and were introduced to obtain temporal-coherent videos by performing convolutions in both time and space. In our case, we employ convolutions in exposure and space. Specifically, our decoder consists of nine layers. The first three layers are $4 \times 4 \times 4$ deconvolution (in the order of exposure, width, and height) with a stride of (2, 2, 2) and

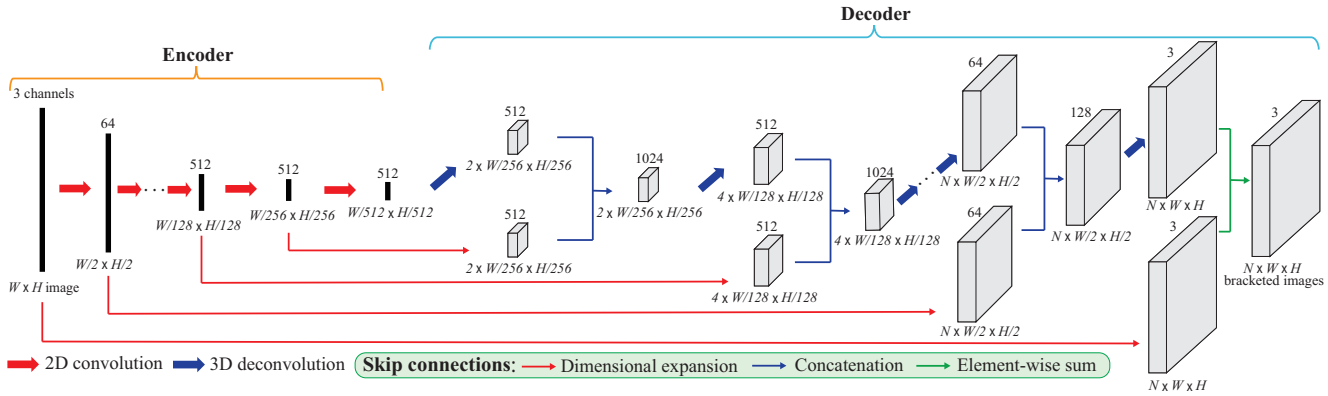

Fig. 5. Our network architecture. The same architecture is used both for up- and down-exposure models.

padding of (1, 1, 1). The remaining layers are $3 \times 4 \times 4$ deconvolution with a stride of (1, 2, 2) and padding of (1, 1, 1). That is, the first three layers double the layer inputs across the exposure and spatial axes, and the remaining layers do the same thing across the spatial axis only. From the first layer to the last layer, the number of filter kernels are 512, 512, 512, 512, 512, 256, 128, 64, and 3, respectively. In the layers other than the last one, batch normalization is applied, and the activation function is the ReLU function [Nair and Hinton 2010]. The last layer outputs a four-dimensional tensor with pixel values in [0, 1] through the sigmoid function.

The 3D CNNs are often used for video analysis and classification [Ji et al. 2013; Karpathy et al. 2014; Maturana and Scherer 2015; Tran et al. 2015; Wu et al. 2015]. For video prediction, recent work [Brock et al. 2016; Vondrick et al. 2016] has generated more successful results compared to previous work. However, the output video is small (64×64) and still contains inconsistent noise across the temporal axis. This is because the network is more complex than 2D CNN and learning is more difficult than for a single image. In our case, because we use a larger image and deeper deconvolution network, this problem becomes more severe. We thus extend the network with *skip connections*, as described below.

Skip connections. In the above encoder-decoder network, the decoder uses a fully encoded vector, which means a latent feature of a whole image. To incorporate local and low-level information from an input image into the decoder step-by-step, we add skip connections following the extension of the U-Net [Ronneberger et al. 2015] and residual units [He et al. 2016]. The U-Net has skip connections between each layer i and layer $n - i$, where n is the total number of layers, which concatenates all channels at the two connected layers. This architecture enables the decoder to utilize local information and accelerates learning. The residual units can be expressed in a general form as $x_{l+1} = f(h(x_l) + F(x_l))$, where x_l and x_{l+1} denote the input and output of the l-th unit, f denotes an activation function, and F is a residual function. Intuitively, the residual units can learn variations from the input. We assume that learning exposure variations from the input is easier than learning how to generate new images from scratch.

The U-Net was recently used for image generation [Isola et al. 2016], and the residual units also have compelling accuracy for image classification [He et al. 2016]. However, they are designed for 2D convolution and deconvolution and cannot be applied to 2D convolution and 3D deconvolution directly due to the different dimensions of the outputs on the layers with skip connections. We thus expand the dimensions of the input image and intermediate features in the encoder. Specifically, we duplicate and concatenate the input image and encoded features so that the dimensions of each tensor match those of the corresponding layer of the decoder. That is, we convert the $W \times H \times c$ input image tensor and $\frac{W}{2^l} \times \frac{H}{2^l} \times c_l$ encoded features into $N \times W \times H \times c$ and $N \times \frac{W}{2^l} \times \frac{H}{2^l} \times c_l$. For the U-Net, the $N \times \frac{W}{2^l} \times \frac{H}{2^l} \times c_l$ encoded features and output of the connected deconvolution layers are concatenated. For the residual units, the $N \times W \times H \times c$ input image tensor is added to the last layer before the activation function. In the residual unit $x_{l+1} = f(h(x_l) + F(x_l))$ of our network, x_l is an input image, x_{l+1} is an output, the function f is the sigmoid function, h is an identity mapping, and F is the function of the whole network without the last sigmoid function.

4.2 Learning and Implementation

As described in Section 3.1, we synthesize a set of $T + 1$ bracketed images for each scene and each CRF. Let D be such a set of $T + 1$ bracketed images and $I_j \in D$ be an LDR image with exposure duration index j (as used in Equation (2)). Given I_j, the up-exposure model learns the relative changes due to increased exposure by referencing the rest of the images in D with higher exposures, that is, $I_{j+1}, I_{j+2}, \ldots, I_{j+1+N}$, as ground-truth data. The loss function for the up-exposure model is then defined as:

$$\sum_{D} \sum_{j=1}^{T} \| I^{up}_{j+1 \to j+1+N} \oplus O_j - M_j \circ G(I_j, \theta) \|_1, \tag{3}$$

where $I^{up}_{j+1 \to j+1+N}$ denotes a $\min\{N, T+1-j\} \times W \times H \times c$ tensor obtained by concatenating images from I_{j+1} to $I_{\min\{j+1+N, T+1\}}$. O_j and $\oplus$ are a $\min\{j+N-T-1, 0\} \times W \times H \times c$ zero tensor and a concatenation operator. M_j and $\circ$ denote a $N \times W \times H \times c$ tensor and

element-wise product. To mask the regions where the data do not exist, each element of M_j is one if the index of the first dimension is less than $T+2-j$ and zero otherwise. $G(I_j, \theta)$ is an output $N \times W \times H \times c$ tensor of our network and θ denotes network weights. We use L1 distance instead of L2 distance because it can reduce blurring in the results [Zhao et al. 2017]. For the down-exposure model, the network learns the images in reverse from the same training data, and its loss function is defined as:

$$\sum_{D} \sum_{j=1}^{T} \| I^{down}_{T+1-j \to T+1-j-N} \oplus O_j - M_j \circ G(I_{T+2-j}, \theta) \|_1, \tag{4}$$

where $I^{down}_{T+1-j \to T+1-j-N}$ denotes a $\min\{N, T+1-j\} \times W \times H \times c$ tensor obtained by concatenating the images in reverse from I_{T+1-j} to $I_{\max\{1, T+1-j-N\}}$.

We train our network with a stochastic gradient descent of batch size of one using the Adam optimizer [Kingma and Ba 2014] with a fixed learning rate of 0.0002 and momentum term of 0.5. We initialize all weights of the 2D convolution and 3D deconvolution with zero mean Gaussian noise with standard deviation 0.02. A 50% dropout rate [Srivastava et al. 2014] is applied to the first three deconvolution layers in the decoder after the batch normalization in order to make the decoder robust against noise in the encoded features. The optimization procedure can be easily implemented using common libraries without modifying the solvers. We implemented our algorithm and neural network models using Python language and Chainer library. In our experiments, the L1 loss-based optimization took about 60 epochs for a certain level of convergence (about a month on a GeForce GTX 1080).

5 EXPERIMENTS

We compared our method with several existing reverse tone mapping methods [Akyüz et al. 2007; Huo et al. 2014; Kovaleski and Oliveira 2014; Masia et al. 2015] that do not require manual annotations on images. To perform experiments using these existing methods, we used the HDR Toolbox [Banterle et al. 2011], which is a MATLAB-based library for processing HDR content. For the parameter settings, we enabled multilinear regression to determine the γ value for the method of Masia et al. and used $\gamma = 2.2$ for inverse gamma correction in all existing methods. The other parameters are default unless otherwise noted. The programs of ours and the existing methods were run on a PC with Core i7-5960X CPU, 128GB RAM, and GeForce GTX 1080. For inference, each of our up-/down-exposure models took about two seconds for the $1,536 \times 1,024$ (Figure 1) as well as $1,536 \times 1,536$ images (Figure 10), and about 0.3 seconds for 512×512 images. The whole process took about five seconds and one second, respectively.

In the following, we first describe the training dataset we used and then show our experimental results as well as comparisons with the results of the existing methods.

5.1 Dataset

To optimize the weights of our network, we constructed a training dataset by collecting HDR images from [Funt and Shi 2010a,b; Nemoto et al. 2015; Xiao et al. 2002] and other online databases[1]. The datasets contain indoor scenes as well as outdoor scenes. In some datasets, HDR images are environment maps in the longitude/latitude format. From these environment maps, we augmented HDR images by trimming several regions and rendering them with perspective projection. Specifically, we fixed the elevation angle and view angle to 10 and 80 degrees while using six azimuths (0, 60, 120, 180, 240, and 300 degrees). We used 1,043 HDR images for training. These HDR images are used to generate bracketed LDR images with five types of tone curves and nine exposure durations. The total number of LDR images for training is 46,935. All of the training images are resized to 512×512. Note that the inputs of the resulting images are not included in the training data.

5.2 Results

We first visualized the obtained HDR images by using an existing tone mapping operator (TMO). Among many TMOs, we chose Kim et al.'s TMO [Kim and Kautz 2008] for all results because it can generate consistent tone-mapped results without adjusting parameters for each new image to achieve high-quality results.

Figures 1 and 6 show the results of applying the proposed method and the existing methods to various scenes. The input test images are obtained from Places205 dataset [Zhou et al. 2014] and SUN360 dataset [Xiao et al. 2012]. Because the images in the SUN360 dataset are panoramas and too large (9104×4552), we trimmed them as described in Section 5.1 and generated 512×512 images for Figure 6 as well as $1,536 \times 1,536$ images for Figure 10. To visualize the clipped pixels in the input LDR images, pixels are painted in red/green if they are completely under-/overexposed (i.e., black/white) or in magenta/cyan if one or two channels are under-/overexposed, respectively. In the results from the existing methods, some noises and discontinuity appear in the under/overexposed regions, whereas our results look natural and have enriched gradations. For example, in the top row in Figure 6, smooth gradation can be observed around the entrance of the castle in our result, while this is not the case for the other results. In the top three rows, red and blue noise appears in the underexposed regions, but such artifacts are not observed in our results. In the third row from the top, the results of the existing methods appear too reddish, while our result retains rich color variations. Additionally, in the three bottom rows, we used significantly overexposed LDR images as inputs. These are challenging examples because this task is essentially inpainting of missing regions (painted in green). Although not perfect, our method can restore more plausible color gradations than the existing methods.

We also conducted a quantitative evaluation by comparing inferred and ground-truth HDR images. As an evaluation metric, we used HDR-VDP-2 [Mantiuk et al. 2011], which can compute visual difference based on human perception rather than a mathematical differences such as root-mean-square-error between two HDR images.[2] The input LDR images for each method were created from ground-truth HDR images (not included in the training data) using

[1] The online HDR datasets that we used are: EMPA HDR database (http://empamedia.ethz.ch/), pfstools HDR image gallery (http://pfstools.sourceforge.net/) and sIBL Archive (http://www.hdrlabs.com/sibl/).
[2] We used the latest version 2.2.1 downloaded from http://hdrvdp.sourceforge.net/wiki/.

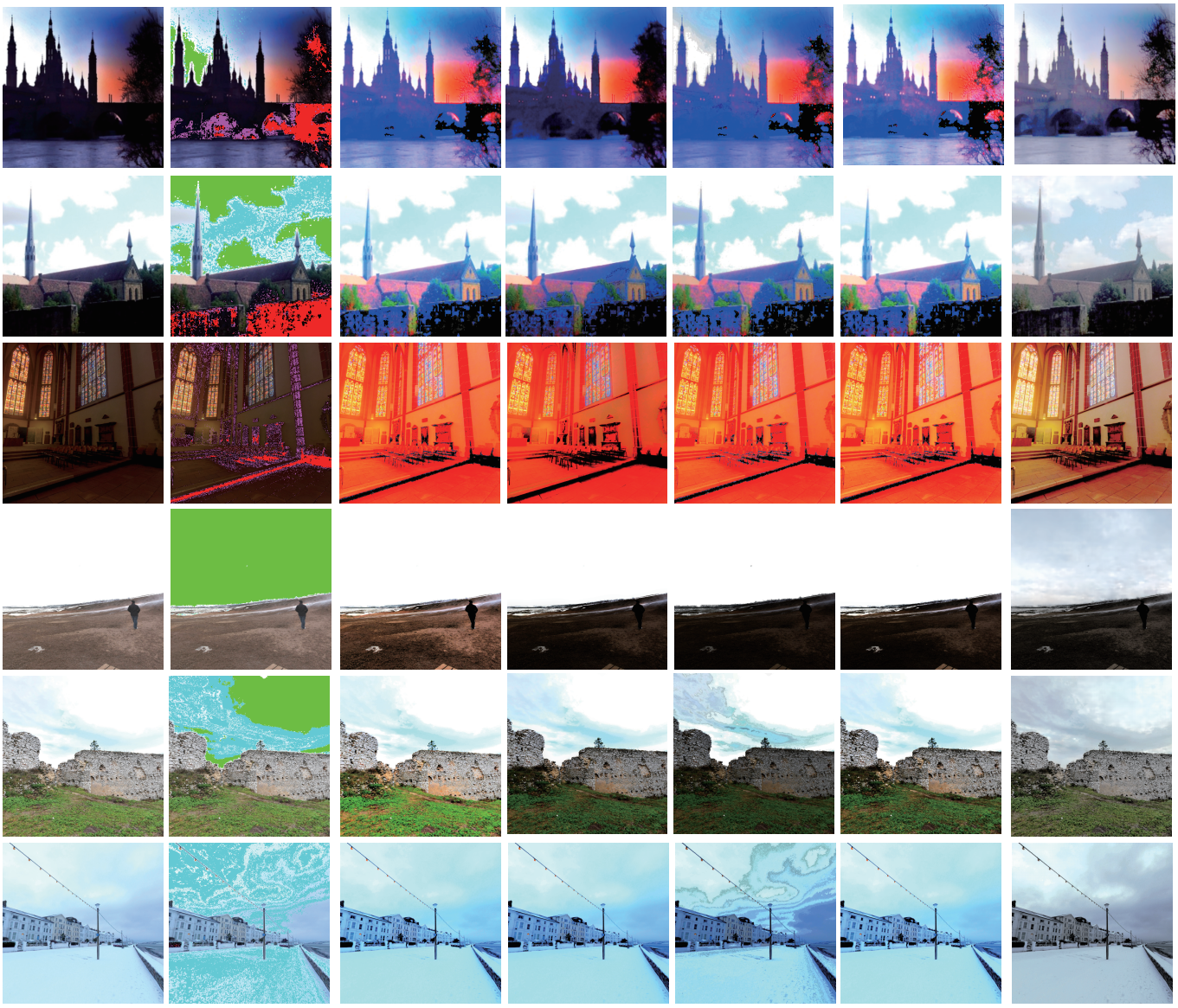

Fig. 6. Tone-mapped results of HDR images generated by each rTMO. To visualize the clipped pixels in the input LDR images, pixels of the images in the second column are painted in red/green if they are completely under/overexposed (i.e., black/white) or in magenta/cyan if one or two channels are under/overexposed, respectively. While there is amplified noise in the dark regions of other results, our results exhibit natural tone reproduction with the least noise. Please enlarge images in the electronic version.

the CRFs in the database. For evaluation using HDR-VDP-2, the maximum luminance value of each output HDR image is adjusted to that of the ground truth image by specifying the corresponding parameter (e.g., `maxOutput` in the HDR toolbox) for the compared methods. Our results are also normalized using ground-truth maximum luminance values.

Figure 7 shows the visual differences computed by HDR-VDP-2 between the HDR images generated by each method and the ground-truth HDR images. Obviously, our results are better than those of the existing methods. Table 1 shows the average Q (quality correlate) scores of the HDR-VDP-2 results for 690 scenes[3] generated with five CRFs. The model of "Ours (srgb)" was trained using a dataset generated with only a single sRGB curve as a CRF (Section 3.1). Our model trained with multiple CRFs has the best score compared to the other existing methods and to "Ours (srgb)". We

[3]The original dataset is publicly-available via http://www.cr-market.com/?p=842.

Table 1. Average Q (quality correlate) scores of HDR-VDP-2 for 690 scenes. Larger values are better (up to 100). Q scores can be negative. The values after ± are standard deviations.

Methods	Q scores
[Akyüz et al. 2007]	54.30 ± 6.70
[Huo et al. 2014]	53.44 ± 6.01
[Kovaleski and Oliveira 2014]	54.36 ± 6.73
[Masia et al. 2015]	54.05 ± 7.23
Ours (srgb)	56.83 ± 7.19
Ours	**58.52 ± 7.15**

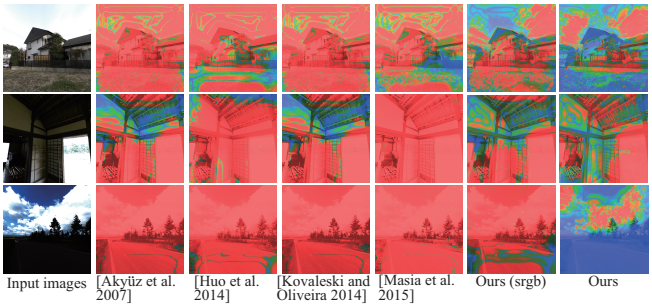

Fig. 7. Comparisons with the HDR images generated by each method and the ground-truth HDR images. We used the HDR-VDP-2 metric, which calculates visibility differences based on human perception. The visualized differences increase from blue to red.

also performed a z-test, and the scores of our method and the other methods were significantly different at $p < 0.001$.

6 DISCUSSION

Here we discuss alternatives to our network architecture.

Direct inference approach. As we mentioned in Section 1, an alternative approach to reverse tone mapping with deep learning is to directly learn an end-to-end mapping function from LDR to HDR. The difficulty of this approach was described in Section 1. Figure 8 shows an example of a failed result from such direct inference, where we used a conditional adversarial network, similar to [Isola et al. 2016]. The HDR outputs are inconsistent and quite noisy. Inferring an identical image from different inputs is challenging, a difficulty which is also recognized in, for instance, the task of frontalizing a human face from different views [Huang et al. 2017]. Although high-quality results are reported with recent face frontalization techniques, our task seems much more difficult than face frontalization because the inputs of face frontalization are limited to LDR face images, that is, variations are smaller than ours.

Decoder design. We considered several approaches to generating multiple images from a single image, and we decided to use 3D CNNs for the following reasons. One possible approach is to repeatedly apply 2D deconvolution to the encoded features of a previous output. Every time the deconvolution is applied, an image with increased (or decreased) exposure is obtained. This approach is simple and can be easily realized by using current generative

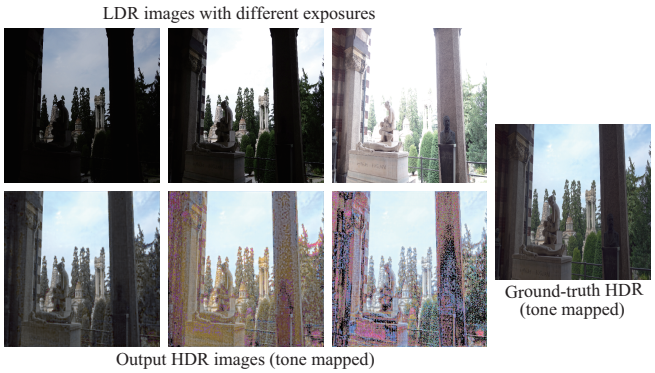

Fig. 8. Failed results of directly inferring an HDR image from a single LDR image. We used the three LDR images (upper left) with different exposures and the HDR image (right) as training data for the generative neural network model. We then fed in the same LDR images as input to the model but consistent HDR images were not generated (lower left). This exemplifies the great difficulties of learning a mapping function from different LDR inputs for the same HDR output as well as directly recovering missing information from much higher dynamic ranges.

Fig. 9. Artifacts of repeated patterns caused by using adversarial loss. From left to right, the input LDR image; the downexposed LDR image, and the up-exposed LDR image.

models for images [Isola et al. 2016; Pathak et al. 2016]. However, in our preliminary experiment this approach often failed to generate high-quality images because noise and artifacts in the output accumulate with repeated deconvolution. Another approach is to incorporate recurrent structures such as long short-term memories (LSTMs) [Gers et al. 2000] into the 2D convolution and deconvolution network. This approach was recently adopted for video prediction [Lotter et al. 2016; Srivastava et al. 2015]. Although recent generative models for 2D images can generate a realistic image of relatively large size, it is still difficult for this approach to generate multiple images that are consistent with each other from input natural images of tens of thousands of pixels or more.

Adversarial loss. While our method uses the L1 loss function for training the network, adversarial loss [Denton et al. 2015; Goodfellow et al. 2014; Isola et al. 2016; Radford et al. 2015] has recently shown significant improvement in generating sharper images. We also tried the adversarial loss function for various networks. However, we were not able to reconstruct plausible images in under-/over-exposed regions, and some repeated artifacts were observed

Fig. 10. Our results with larger input images selected from Figure 6. Tiling artifacts appear if the completely-saturated regions are large (left and middle) while natural results can be obtained if at least one channel is not saturated (right).

as shown in Figure 9. Presumably this is because our dataset is crucially small compared to the potential variations of scenes and the training of the adversarial loss function is unstable in this task.

Limitations and future work. Currently our method is not sufficiently capable of handling scenes with extremely high dynamic range because our models are trained with a fixed range of exposures. We can confirm that, for example, the brightness of the sky or the daylight outside the door in Figure 7 is not reproduced plausibly. This issue can be alleviated by increasing the number of inferred bracketed images, but at the cost of a larger memory footprint. Considering the logarithm of pixel values might be helpful [Eilertsen et al. 2017] .

If the given LDR image has wide clipped regions (especially in the case of high-resolution images), the newly synthesized content (e.g., clouds) exhibits tiling artifacts, as shown in Figure 10. This is probably because the spatial sizes of convolution kernels become relatively too small for test images that are larger than those used for training. One workaround is to use larger kernels and larger training images, resulting again in a larger memory footprint and longer training times. A better approach would be to synthesize new contents on a local patch basis while accounting for the global context of the scene.

Although we have demonstrated the effectiveness of our approach compared to existing rTMOs, its expressive power is still limited due to the rather small dataset, containing small variations, compared to the variations in conventional tasks, for example, image classification. To enrich our training dataset, synthesizing HDR images using 3D computer graphics represents a promising avenue. We would also like to incorporate stochastic factors, such as generative adversarial nets (GAN), to synthesize plausible images even with large under/overexposed regions.

7 CONCLUSION

This paper has presented a data-driven method for reverse tone mapping based on CNNs. Most of the existing methods are limited because they rely on specific assumptions or heuristics; the alternative approach utilizes reference images to cope with loss of data, but it requires user markups and reference images of very similar scenes. Unlike these existing approaches, our approach is a first attempt at supervised learning, which automatically infers an HDR image from a single LDR image by learning exposure changes. The training dataset, consisting of various bracketed images, is created using HDR images and CRF databases. In addition we presented a 2D convolution- and 3D deconvolution-based neural network architecture with skip connections for generating over- and underexposed images.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for giving insightful comments.

REFERENCES

Ahmet Oğuz Akyüz, Roland Fleming, Bernhard E. Riecke, Erik Reinhard, and Heinrich H. Bülthoff. 2007. Do HDR Displays Support LDR Content?: A Psychophysical Evaluation. *ACM Trans. Graph.* 26, 3, Article 38 (July 2007).

Francesco Banterle, Alessandro Artusi, Kurt Debattista, and Alan Chalmers. 2011. *Advanced High Dynamic Range Imaging: Theory and Practice.* AK Peters (CRC Press), Natick, MA, USA.

Francesco Banterle, Patrick Ledda, Kurt Debattista, and Alan Chalmers. 2006. Inverse tone mapping. In *Proc. of GRAPHITE'06.* 349–356.

Francesco Banterle, Patrick Ledda, Kurt Debattista, and Alan Chalmers. 2008. Expanding low dynamic range videos for high dynamic range applications. In *Proc. of SCCG'08.* 33–41.

Francesco Banterle, Patrick Ledda, Kurt Debattista, Alan Chalmers, and Marina Bloj. 2007. A framework for inverse tone mapping. *The Visual Computer* 23, 7 (2007), 467–478.

André Brock, Theodore Lim, James M. Ritchie, and Nick Weston. 2016. Generative and Discriminative Voxel Modeling with Convolutional Neural Networks. *CoRR* abs/1608.04236 (2016). http://arxiv.org/abs/1608.04236

Paul E. Debevec and Jitendra Malik. 1997. Recovering High Dynamic Range Radiance Maps from Photographs. In *Proc. of SIGGRAPH'97.* 369–378.

Emily L. Denton, Soumith Chintala, Arthur Szlam, and Rob Fergus. 2015. Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks. In *Proc. of NIPS'15.* 1486–1494.

Piotr Didyk, Rafal Mantiuk, Matthias Hein, and Hans-Peter Seidel. 2008. Enhancement of Bright Video Features for HDR Displays. *Comput. Graph. Forum* 27, 4 (2008), 1265–1274.

Gabriel Eilertsen, Joel Kronander, Gyorgy Denes, Rafal Mantiuk, and Jonas Unger. 2017. HDR image reconstruction from a single exposure using deep CNNs. *ACM Trans. Graph. (Proc. of SIGGRAPH ASIA 2017)* 36, 6 (Nov. 2017).

Brian V. Funt and Lilong Shi. 2010a. The effect of exposure on MaxRGB color constancy. In *Human Vision and Electronic Imaging XV, part of the IS&T-SPIE Electronic Imaging Symposium.* 75270.

Brian V. Funt and Lilong Shi. 2010b. The Rehabilitation of MaxRGB. In *Proc. of Color and Imaging Conference 2010.* 256–259.

Felix A. Gers, Jürgen Schmidhuber, and Fred A. Cummins. 2000. Learning to Forget: Continual Prediction with LSTM. *Neural Computation* 12, 10 (2000), 2451–2471.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. 2014. Generative Adversarial Nets. In *Proc. of NIPS'14.* 2672–2680.

Michael D. Grossberg and Shree K. Nayar. 2003. What is the Space of Camera Response Functions?. In *Proc. of CVPR'03.* 602–612.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *Proc. of CVPR'16.* 770–778.

Rui Huang, Shu Zhang, Tianyu Li, and Ran He. 2017. Beyond Face Rotation: Global and Local Perception GAN for Photorealistic and Identity Preserving Frontal View Synthesis. *CoRR* abs/1704.04086 (2017). http://arxiv.org/abs/1704.04086

Yongqing Huo, Fan Yang, and Vincent Brost. 2013. Dodging and Burning Inspired Inverse Tone Mapping Algorithm. *Computational Information Systems* 9, 9 (2013), 3461–3468.

Yongqing Huo, Fan Yang, Le Dong, and Vincent Brost. 2014. Physiological inverse tone mapping based on retina response. *The Visual Computer* 30, 5 (2014), 507–517.

Sergey Ioffe and Christian Szegedy. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *Proc. of ICML'15.* 448–456.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. 2016. Image-to-Image Translation with Conditional Adversarial Networks. *arxiv* (2016).

G. Jain, A. Plappally, and S. Raman. 2014. InternetHDR: Enhancing an LDR image using visually similar Internet images. In *Proc. of Twentieth National Conference on Communications.* 1–6.

Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 2013. 3D Convolutional Neural Networks for Human Action Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* 35, 1 (2013), 221–231.

Nima Khademi Kalantari and Ravi Ramamoorthi. 2017. Deep High Dynamic Range Imaging of Dynamic Scenes. *ACM Transactions on Graphics (Proc. of SIGGRAPH*

2017) 36, 4 (2017).
Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. 2014. Large-Scale Video Classification with Convolutional Neural Networks. In *Proc. of CVPR'14.* 1725–1732.
Min H. Kim and Jan Kautz. 2008. Consistent Tone Reproduction. In *Proc. of CGIM'08.* 152–159.
Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *CoRR* abs/1412.6980 (2014). http://arxiv.org/abs/1412.6980
Rafael Kovaleski and Manuel M. Oliveira. 2014. High-Quality Reverse Tone Mapping for a Wide Range of Exposures. In *Proc. of SIBGRAPI'14.* 49–56.
Rafael Pacheco Kovaleski and Manuel M. Oliveira. 2009. High-quality brightness enhancement functions for real-time reverse tone mapping. *The Visual Computer* 25, 5-7 (2009), 539–547.
Joel Kronander, Stefan Gustavson, Gerhard Bonnet, and Jonas Unger. 2013. Unified HDR reconstruction from raw CFA data. In *Proc. of ICCP'13.* 1–9.
William Lotter, Gabriel Kreiman, and David Cox. 2016. Unsupervised Learning of Visual Structure using Predictive Generative Networks. In *ICLR'16 workshop.*
Andrew L. Maas, Awni Y. Hannun, and Andrew Y. Ng. 2013. Rectifier nonlinearities improve neural network acoustic models. In *ICML Workshop on Deep Learning for Audio, Speech, and Language Processing.*
Mann, Picard, S. Mann, and R. W. Picard. 1995. On Being 'undigital' With Digital Cameras: Extending Dynamic Range By Combining Differently Exposed Pictures. In *Proc. of IS&T.* 442–448.
Rafal Mantiuk, Kil Joong Kim, Allan G. Rempel, and Wolfgang Heidrich. 2011. HDR-VDP-2: a calibrated visual metric for visibility and quality predictions in all luminance conditions. *ACM Trans. Graph.* 30, 4 (2011), 40:1–40:14.
R. K. Mantiuk, K. Myszkowski, and H.-P. Seidel. 2015. High Dynamic Range Imaging. In *Wiley Encyclopedia of Electrical and Electronics Engineering.* John Wiley & Sons Inc., 1–42.
Belen Masia, Sandra Agustin, Roland W. Fleming, Olga Sorkine, and Diego Gutierrez. 2009. Evaluation of Reverse Tone Mapping Through Varying Exposure Conditions. *ACM Transactions on Graphics (Proc. of SIGGRAPH Asia)* 28, 5 (2009), 160:1–160:8.
Belen Masia, Roland W. Fleming, Olga Sorkine, and Diego Gutierrez. 2010. Selective Reverse Tone Mapping. In *Congreso Español de Informatica Grafica.* Eurographics.
Belen Masia and Diego Gutierrez. 2016. Content-Aware Reverse Tone Mapping. In *Proc. of ICAITA 2016.*
Belen Masia, Ana Serrano, and Diego Gutierrez. 2015. Dynamic range expansion based on image statistics. *Multimedia Tools and Applications* (2015), 1–18.
Daniel Maturana and Sebastian Scherer. 2015. VoxNet: A 3D Convolutional Neural Network for Real-Time Object Recognition. In *Proc. of IEEE/IROS'15.* 922–928.
Tom Mertens, Jan Kautz, and Frank Van Reeth. 2007. Exposure Fusion. In *Proc .of Pacific Graphics 2007.* 382–390.
Vinod Nair and Geoffrey E. Hinton. 2010. Rectified Linear Units Improve Restricted Boltzmann Machines. In *Proc. of ICML2010.* 807–814.
Hiromi Nemoto, Pavel Korshunov, Philippe Hanhart, and Touradj Ebrahimi. 2015. Visual attention in LDR and HDR images. In *International Workshop on Video Processing and Quality Metrics for Consumer Electronics.* http://mmspg.epfl.ch/hdr-eye
Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros. 2016. Context Encoders: Feature Learning by Inpainting. In *Proc. of CVPR'16.* 2536–2544.
Alec Radford, Luke Metz, and Soumith Chintala. 2015. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *CoRR* abs/1511.06434 (2015). http://arxiv.org/abs/1511.06434
Erik Reinhard, Michael M. Stark, Peter Shirley, and James A. Ferwerda. 2002. Photographic tone reproduction for digital images. *ACM Trans. Graph.* 21, 3 (2002), 267–276.
Allan G. Rempel, Matthew Trentacoste, Helge Seetzen, H. David Young, Wolfgang Heidrich, Lorne Whitehead, and Greg Ward. 2007. Ldr2Hdr: On-the-fly Reverse Tone Mapping of Legacy Video and Photographs. *ACM Trans. Graph.* 26, 3, Article 39 (July 2007).
O. Ronneberger, P.Fischer, and T. Brox. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI) (LNCS)*, Vol. 9351. 234–241.
Florian M. Savoy, Vassilios Vonikakis, Stefan Winkler, and Sabine Süsstrunk. 2014. Recovering badly exposed objects from digital photos using internet images. In *Proc. of Digital Photography X, part of the IS&T-SPIE Electronic Imaging Symposium.* 90230W.
Ana Serrano, Felix Heide, Diego Gutierrez, Gordon Wetzstein, and Belen Masia. 2016. Convolutional Sparse Coding for High Dynamic Range Imaging. *Comput. Graph. Forum* 35, 2 (2016), 153–163.
Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15, 1 (2014), 1929–1958.
Nitish Srivastava, Elman Mansimov, and Ruslan Salakhutdinov. 2015. Unsupervised Learning of Video Representations using LSTMs. In *Proc. of ICML'15.* 843–852.
Michael D. Tocci, Chris Kiser, Nora Tocci, and Pradeep Sen. 2011. A versatile HDR video production system. *ACM Trans. Graph.* 30, 4 (2011), 41:1–41:10.
Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. 2015. Learning Spatiotemporal Features with 3D Convolutional Networks. In *Proc. of ICCV'15.* 4489–4497.
Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. 2016. Generating Videos with Scene Dynamics. In *Proc. of NIPS'16.* 613–621.
Lvdi Wang, Li-Yi Wei, Kun Zhou, Baining Guo, and Heung-Yeung Shum. 2007. High Dynamic Range Image Hallucination. In *Proc. of EGSR'07.* 321–326.
T. H. Wang, C. W. Chiu, W. C. Wu, J. W. Wang, C. Y. Lin, C. T. Chiu, and J. J. Liou. 2015. Pseudo-Multiple-Exposure-Based Tone Fusion With Local Region Adjustment. *IEEE Transactions on Multimedia* 17, 4 (2015), 470–484.
Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 2015. 3D ShapeNets: A deep representation for volumetric shapes. In *Proc. of CVPR'15.* 1912–1920.
Feng Xiao, Jeffrey M. DiCarlo, Peter B. Catrysse, and Brian A. Wandell. 2002. High Dynamic Range Imaging of Natural Scenes. In *Proc. of Color and Imaging Conference 2002.* 337–342.
J. Xiao, K. A. Ehinger, A. Oliva, and A. Torralba. 2012. Recognizing scene viewpoint using panoramic place representation. In *Proc. of CVPR'12.* 2695–2702.
Jinsong Zhang and Jean-François Lalonde. 2017. Learning High Dynamic Range from Outdoor Panoramas. (2017). arXiv:arXiv:1703.10200
H. Zhao, O. Gallo, I. Frosio, and J. Kautz. 2017. Loss Functions for Image Restoration With Neural Networks. *IEEE Transactions on Computational Imaging* 3, 1 (2017), 47–57.
Hang Zhao, Boxin Shi, Christy Fernandez-Cull, Sai-Kit Yeung, and Ramesh Raskar. 2015. Unbounded High Dynamic Range Photography Using a Modulo Camera. In *Proc. of ICCP'15.* 1–10.
Bolei Zhou, Àgata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. 2014. Learning Deep Features for Scene Recognition using Places Database. In *Proc. of NIPS'14.* 487–495.

A COMPENSATING INFERRED HDR IMAGES BY LINEAR SCALING

Here, we explain why the scaling of E_i or Δt_j in Section 3.1 can be compensated by linear scaling of the inferred HDR values.

In the method of Debevec and Malik [1997], an estimated HDR value $\hat{E}_i$ is calculated as

$$\ln \hat{E}_i = \frac{\sum_j w(Z_{ij})(g(Z_{ij}) - \ln \Delta t_j)}{\sum_j w(Z_{ij})}, \tag{5}$$

where $w(\cdot)$ is a hat function for weighting and $g = \ln f^{-1}$. Without loss of generality, suppose that Δt_j is linearly scaled to $\alpha \Delta t_j$ with a coefficient α. The corresponding estimated HDR value $\hat{E}'_i$ is then

$$\ln \hat{E}'_i = \frac{\sum_j w(Z_{ij})(g(Z_{ij}) - \ln \alpha \Delta t_j)}{\sum_j w(Z_{ij})} \tag{6}$$

$$= \frac{\sum_j w(Z_{ij})(g(Z_{ij}) - \ln \Delta t_j)}{\sum_j w(Z_{ij})} - \ln \alpha \tag{7}$$

$$= \ln \hat{E}_i - \ln \alpha, \tag{8}$$

which means that it is a linear relationship, that is, $\hat{E}'_i = \hat{E}_i \exp(-\alpha)$. Suppose that $\hat{E}$ is the ground-truth HDR image and Δt_j is the ground-truth exposure duration. When we train our model with the ground truth HDR image $\hat{E}$ and the corresponding set of bracketed LDR images, for example, the inferred HDR image $\hat{E}'$ becomes dimmer with a longer exposure duration (i.e., $\alpha > 1$) than the ground-truth image. This is because, in this setting, it is assumed that the longer duration was required to record the ground-truth LDR value in a dimmer scene. Conversely, $\hat{E}'$ becomes brighter with $\alpha < 1$. Although α is unknown in general, the user can adjust the inferred HDR image $\hat{E}'$ by multiplying a value close to $\exp(\alpha)$ to obtain a plausible HDR image.