Self-Supervised High Dynamic Range Imaging: What Can Be Learned from a Single 8-bit Video?

FRANCESCO BANTERLE[*], ISTI-CNR, Italy
DEMETRIS MARNERIDES[†], University of Warwick, United Kingdom
THOMAS BASHFORD-ROGERS[‡], University of Warwick, United Kingdom
KURT DEBATTISTA[‡], University of Warwick, United Kingdom

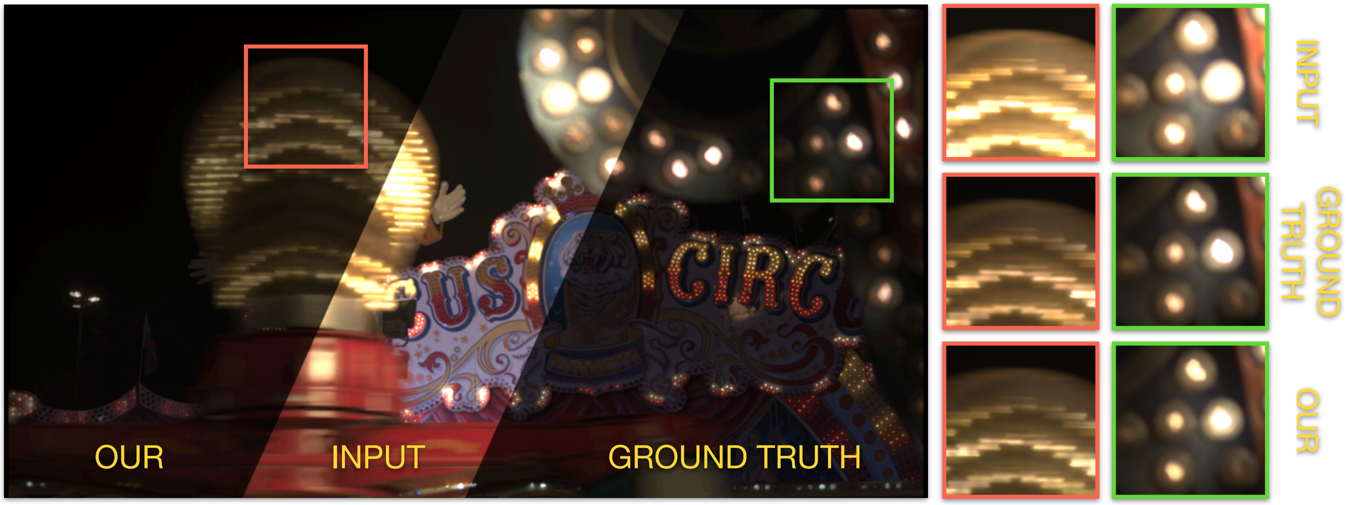


Fig. 1. An example of our inverse tone mapping operator applied to an SDR version frame from the Carousel fireworks 02 sequence [13]. Our method can recover missing texture, colors, and dynamic range details in a convincing way.

Recently, Deep Learning-based methods for inverse tone mapping standard dynamic range (SDR) images to obtain high dynamic range (HDR) images have become very popular. These methods manage to fill over-exposed areas convincingly both in terms of details and dynamic range. To be effective, deep learning-based methods need to learn from large datasets and transfer this knowledge to the network weights. In this work, we tackle this problem from a completely different perspective. What can we learn from a single SDR 8-bit video? With the presented self-supervised approach, we show that, in many cases, a single SDR video is sufficient to generate an HDR video of the same quality or better than other state-of-the-art methods.

CCS Concepts: • **Computing methodologies → Computational photography**; **Image processing**.

Additional Key Words and Phrases: High Dynamic Range Imaging, Inverse Tone Mapping, Deep Learning, Computational Photography

ACM Reference Format:
Francesco Banterle, Demetris Marnerides, Thomas Bashford-Rogers, and Kurt Debattista. 2024. Self-Supervised High Dynamic Range Imaging: What Can Be Learned from a Single 8-bit Video?. *ACM Trans. Graph.* 1, 1 (May 2024), 17 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

To capture the full range of color and shades of brightness in the real world, high dynamic range (HDR) imaging is employed; typically capturing multiple photographs [11, 31] of the same scene at different exposure times. Even though modern sensors setup [8, 24, 25], cameras with smart optics [44, 55], and smartphones [20] can capture HDR imagery, a large amount of content was and still is captured in standard dynamic range (SDR) or is converted to SDR after capture; i.e., smartphones.

When presenting this content on HDR displays [52], or using this imagery for applications where HDR values are required [10], SDR values need to be boosted to HDR; a process known as Inverse Tone Mapping [6].

Researchers have proposed a wide variety of approaches to solving this problem, from straightforward linear functions [1] to, more recently, deep-learning (DL) based solutions [15, 17]. Typically, these DL approaches outperform the existing methods and are mostly

Authors' addresses: Francesco Banterle, francesco.banterle@isti.cnr.it, ISTI-CNR, Pisa, Italy; Demetris Marnerides, dmarnerides@gmail.com, University of Warwick, Coventry, United Kingdom; Thomas Bashford-Rogers, Tom.Bashford-Rogers@warwcik.ac.uk, University of Warwick, Coventry, United Kingdom; Kurt Debattista, k.debattista@warwick.ac.uk, University of Warwick, Coventry, United Kingdom.

0730-0301/2024/5-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

based on training a convolutional neural network (CNN) to encode a mapping from SDR to HDR. To achieve this, a large set of SDR/tone mapped and reference HDR image pairs is required to train a general mapping.

We propose a fundamentally different approach based on the observation that much of the information required for inverse tone mapping may be present in an SDR video sequence. This can be a result of a variety of effects that are present in videos but not in still images. For example, motion in the scene or from the camera can uncover detail that was badly exposed in earlier frames. In addition, changes in the lighting of the scene, or luminance variations due to automatic exposures from the camera can also create a similar effect, where information otherwise lost in some frames exists in some others.

Our approach attempts to gather and distill this information present in a single SDR video to recover information in over-exposed and under-exposed areas of the same video. Figure 1 shows the results of our method. We define a new pipeline for expanding the dynamic range of SDR content using deep-learning approaches. This optimization relies only on the frames of the input SDR video; which are processed in a self-supervised fashion. In the presence of only a single SDR video, there is no ground truth HDR data for training; the method uncovers HDR patterns embedded in the underlying SDR signal using self-supervision. The neural network weights that hold all the knowledge for inverse tone mapping are uniquely learned for each video without relying on external datasets of HDR images or other videos, which are still limited in quantity [51].

In summary, we propose a novel inverse tone mapping operator (ITMO) for expanding SDR videos that self-learns from an input video in a self-supervised fashion. Our approach, even though self-supervised, broadly outperforms state-of-the-art fully supervised ITMO methods both visually and across several metrics. Our main contributions are:

- An self-supervised, straightforward, and effective architecture for expanding SDR videos to obtain HDR videos without the need for a comprehensive dataset.
- A method for generating a tailored dataset starting from a single SDR video;

The source code of this work is available online[1].

2 RELATED WORK

ITMOs generate an HDR image/video from an original SDR version that is quantized at 8-bits [6]. This problem is ill-posed because there is not much information left in under-exposed and over-exposed areas.

2.1 Classic Methods

ITMOs, not employing deep learning, can be classified into three main classes: global, local, and user-based. On one hand, global ITMOs define an expansion function that gathers global statistics from the image and applies it to all pixels. These ITMOs can use linear functions [1], multi-linear functions [45] and gamma functions [7, 30, 42, 43]. On the other hand, local ITMOs define an expansion function that varies per pixel locally exploiting both local and global statistics from the image. Several strategies have been proposed. For example, some operators generate an expand map (a spatially varying function) for guiding the expansion only in certain area of high luminance [6, 21, 29, 49]. In user-based methods, the user drives the expansion and details recovery. For example, Wang et al. [59] proposed a solution in which a user recovers the dynamic range and details of an SDR image using clone-tools and inpainting techniques similar to modern image editors. Another example of such methods is Didyk et al.'s work [13]. In this work, a semi-automatic classification interface allows users to classify pixels into an area consisting of diffuse parts, reflections, and light sources. Then, only reflections and light sources are expanded by applying an adaptive non-linear function.

2.2 Deep Learning-based Methods

Recently, several ITMOs have been proposed using different DL architectures. DL-based methods have largely taken two approaches. The first is to directly reconstruct an HDR image from SDR and the second predicts a set of SDR exposures that are fused to generate an HDR image [12]. Eilertsen et al. [15] masked out well-exposed regions which were reconstructed by a linear operator, and over-exposed regions which were reconstructed by a UNet. Eilerstein et al. [16] extended this work for temporal stability via training regularization. Yu et al. [62] improved Eilertsen et al.'s network [15] by adding an attentive module and using residual blocks in the encoder, they also proposed a novel calibration of HDR images and showed how to extend expansion to environment maps for high-quality image-based lighting. Marnerides et al. [41] used a multi-branch network to directly reconstruct the HDR image where each branch was designed to capture different features for reconstruction. Approaches have also been proposed to reverse the camera pipeline to synthesize HDR images, for example, Yang et al. [61] also used a UNet and Liu et al. [37] reconstruct images using a series of networks. Santos et al. [51] proposed an ITMO based on pretraining a network for inpainting and then specializing this network for ITMO based on masking.

Endo et al. [17] was the first work to predict a set of exposures via an autoencoder that fuses them to generate an HDR image. This approach has been improved using a GAN architecture [33], generalized to create an arbitrary number of different exposure images [34] with cycle consistency, and finally further generalized to arbitrarily change the exposure of the input image by a given target exposure value [23]. The creation of multiple exposures is similar to our work, except their method relied on a large set of training images to learn the mapping from SDR to HDR. Recently, Zhang et al.[63] showed that processing high-frequency and low-frequency parts of an image separately can improve the reconstruction process. The NTIRE 2021 Challenge on High Dynamic Range Imaging [48] presented several supervised HDR reconstruction methods which proposed a range of network architectures and datasets for the evaluation of static images, although these methods are not compared with the state-of-the-art.

In terms of video, Kim et al. [26] proposed a super-resolution and inverse tone mapping approach designed for video applications

[1] https://github.com/banterle/Zeroshot-HDRV

that directly produced HDR frames. They reconstruct low and high-frequency information separately and include upscaling of the high-frequency information, which are then combined into the final frame. They further improved their approach by using a GAN architecture [27]. In both works, dynamic range expansion and super-resolution are computed per frame without an explicit mechanism for enforcing temporal coherence and avoiding temporal artifacts. Recently, Chen et al. [9] introduced three cascade networks for global, local, and highlight enhancement of SDR frames for HDR TV; the method is trained on HDR10 videos. However, the training and the network do not have a mechanism to model temporal data.

Contemporary work [58] focuses on static SDR images. An unsupervised GAN scheme was introduced that used a large dataset of SDR images (approximately 25,000 unique images) during training time to expand the dynamic range of images.

The majority of these approaches are based on the same underlying concept of applying transformations to a ground truth set of HDR images to synthesize an SDR dataset, and then learning the mapping from SDR to HDR or a set of exposures. Unlike other methods, ours uses a zero-shot unsupervised approach that does not require previous data. Furthermore, we focus on video; there are no approaches that harness the information of SDR videos. In this work, we provide a general approach for inverse tone mapping of SDR videos, overcoming the limitations of video methods that cannot be specialized to a particular type of content and require significant dataset sizes and training to learn the mapping.

2.3 Self-supervised Methods for Imaging

Recently, self-supervised methods have become more popular thanks to their performance and the use of limited or no datasets. Shocher et al. [54] introduced zero-shot methods for inverse imaging problems and showed that such strategies can be effective and produce convincing results. The key observation is that the image has repetitions of details at different scales that can be exploited. With a similar aim but a different methodology, Ulyanov et al. [56] proposed the Deep Image Priors framework, where imaging problems such as denoising, inpainting, super-resolution, deblocking, etc. are solved by optimizing the network parameters exploiting a prior degradation function, h, that is known. In this case, no dataset is generated but h (e.g., downsampling operator, blocking method, etc.) has to be defined for each problem.

3 SELF-SUPERVISED EXPANSION

The core concept behind this work is based on the observation that parts of the scene when capturing SDR videos frequently change in their exposedness from frame-to-frame. This may be the consequence of a person/object moving from light to shadow and vice-versa, or the varying exposure time of the shutter speed. This means that information about multiple exposures which can be used for inverse tone mapping is already present in many videos. This indicates that training an ITMO on large datasets, as all deep learning approaches currently do, is not always required.

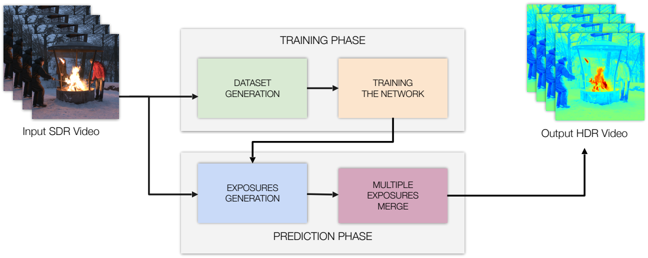


Fig. 2. The full pipeline of our system. To expand the dynamic range of an SDR video, we first generate a tailored dataset, which is employed to train a CNN. Secondly, the trained network predicts higher and lower exposures of the current frame. Finally, these exposures are merged into HDR frames.

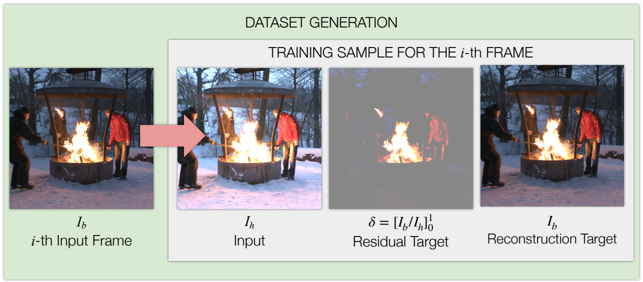


Fig. 3. An example showing how the training data is generated from a single video. Each SDR frame is exposed to a higher exposure. The higher exposures are then used as inputs during training to learn the multiplicative residual mapping $\mathcal{N}$, using the starting SDR frames as targets.

This motivates the design of an approach that can leverage this information for tone mapping. While a patch-based [57] or optical flow [8] method could be used to find the same region of an image in different frames with different exposures, we instead use an approach based on deep learning. This is motivated by the success of deep learning for inverse tone mapping (e.g. [15, 41, 51]) and the use of zero-shot methods with deep learning for single image operations, for example, the super-resolution approach by Shocher et al. [54] based on a similar analysis of similar content in static images [64].

3.1 Overview

Our goal is to predict an HDR video starting from a single input SDR video, V; Figure 2 shows the full pipeline of our work. As a first step, we generate a tailored dataset, $\mathcal{D}_V$, using only V, as shown in Figure 3, and not any other HDR content (Sec.3.2). We then train a network (Sec.3.3), $\mathcal{N}$, that predicts spatially-varying multiplicative residuals, $\hat{\delta}$, such that the lower exposure prediction frame, $\hat{I}_b$, has an e-stop difference from the input, I_h, see Figure 5. The reason why our $\mathcal{N}$ predicts such differences; i.e. $\hat{\delta}$, is in this way the task is more focused only on the missing parts; we show the effectiveness of this approach in Section 4.5. Finally, we employ $\mathcal{N}$ to predict higher and lower exposures; i.e., Figure 7, and we merge them [12] (Sec.3.5).

To summarise, for each original frame at base exposure I_b, we generate a higher exposure frame I_h, that is used as input to the network, $\mathcal{N}$, during training. The network target, δ, is the multiplicative residual of I_h and I_b. During training, the network learns to generate the residual that connects two adjacent exposures via

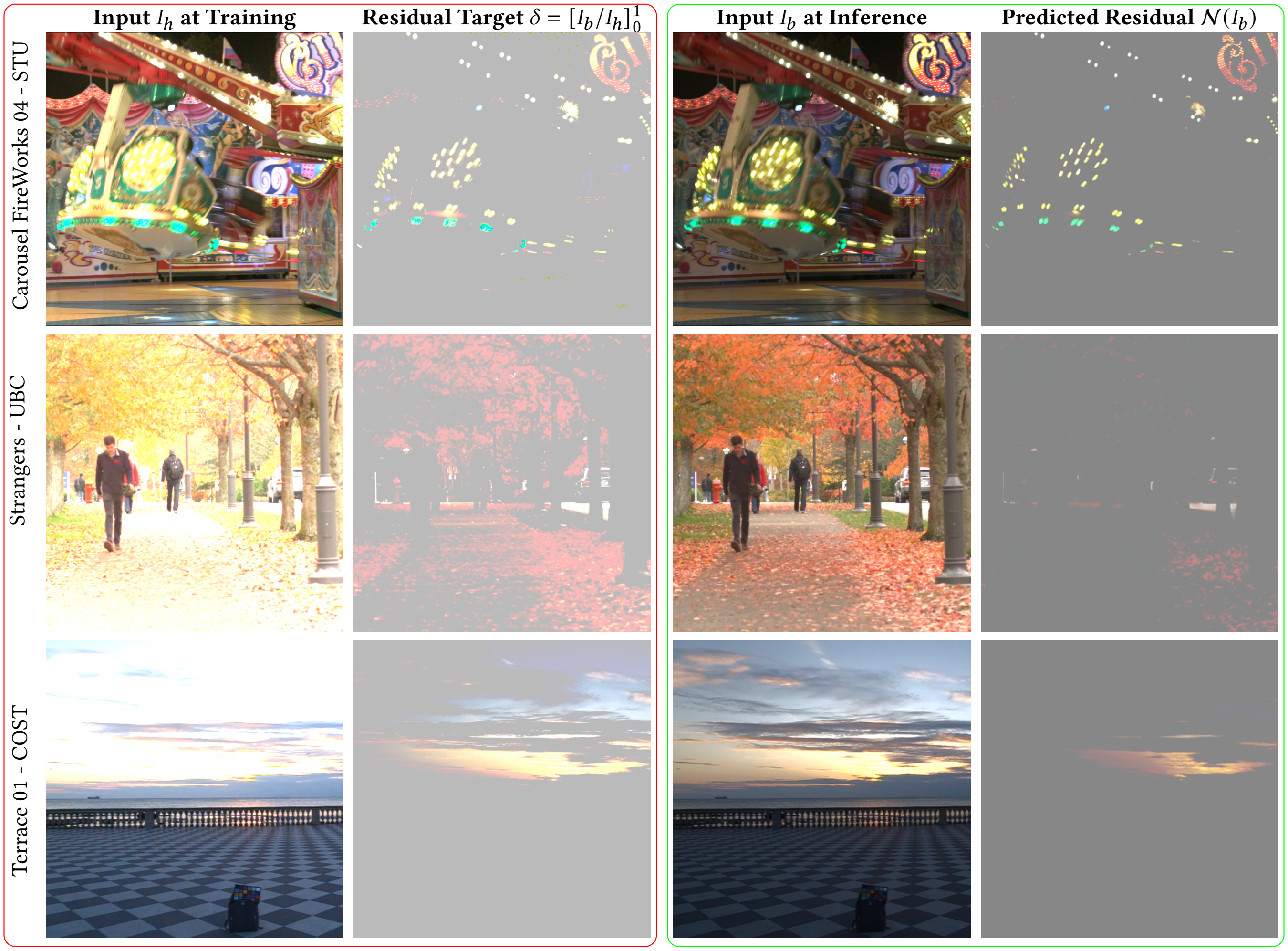

Input I_h at Training
Residual Target $\delta = [I_b/I_h]_0^1$
Input I_b at Inference
Predicted Residual $\mathcal{N}(I_b)$
Carousel FireWorks 04 - STU
Strangers - UBC
Terrace 01 - COST

Fig. 4. Examples of input residuals vs. predicted residuals from our network. At training time (red border), an input frame I_b is re-exposed to create I_h. The network learns the ratio $\delta = [I_b/I_h]_0^1$, where its input is I_h. At inference time (green border), in order to expand the dynamic range of I_b, we apply the network to I_b instead; obtaining a predicted residual, $\mathcal{N}(I_b)$. Note that the predicted residual learns information in the badly exposed regions of the input I_h. For example, it fills in details of the over-exposed sky that are lost in the bottom row.

multiplication or division. At inference, the predicted residual is generated using I_b as input instead, allowing us to generate higher exposures, $\hat{I}_h$, and lower exposures, $\hat{I}_l$, by respectively dividing and multiplying I_b with the predicted residual $\hat{\delta} = \mathcal{N}(I_b)$; see Figure 4. The residual contains information from the whole video as the network is trained on frames across the sequence. This means that temporal information propagation happens via the learned features in the network weights that get information from multiple frames from different times in the same video but also via the temporal consistency loss at each training iteration that takes into account temporally neighboring frames.

The main intuition is we train a network using as input overexposed SDR frames to learn the mapping from overexposed to well-exposed. Depending on the video lighting variations (e.g., motion, exposure time, etc.), this mapping may be fully learned by the network. Therefore, at inference time, we can use the original frames of the video and apply the learned mapping to extend the dynamic range of our video.

Note that apart from the single video that is to be expanded, **no further data needs to be used for training**, making the method self-supervised. The method uses self-supervision from the SDR video as there is no ground truth HDR target. In the absence of a supervisory HDR signal for training, a tailored training dataset is generated from the SDR video that is to be expanded.

3.2 Tailored Dataset Generation

The training dataset is formed by extracting a higher exposure, I_h, from each SDR frame, I_b, of the video at a base exposure value b. To re-expose I_b into I_h, we use the following exposure function:

$$I_{\Delta v} = \left[\left(g^{-1}\left(g(I) \cdot 2^{\Delta v}\right)\right]_0^1 = \left[\left(I \cdot g^{-1}\left(2^{\Delta v}\right)\right)\right]_0^1, \quad (1)$$

where g is the inverse camera response function, $I_{\Delta v}$ is the re-exposed frame I, Δv is the change in exposure value in f-stops, and $[\cdot]_0^1$ is an 8-bit rounding operator with clipping in the range $[0, 1]$. In case the camera response function is unavailable, this can be computed using single image methods [35, 36, 53]. Recently, smartphones and DSLRs encode videos using standard transfer functions such as PQ and HLG [22].

Then, I_h is used to compute the multiplicative residual as:

$$\delta = \left[\frac{I_b}{I_h}\right]_0^1. \tag{2}$$

Note that $I_h > I_b$, therefore $\delta \in [0, 1]$. At evaluation time, $\mathcal{N}$ will instead take the original SDR frame as input, predicting $\hat{\delta}$ to compute higher and lower exposures:

Starting from an SDR input video, V, we assume that it is the result of exposing the ground truth HDR scene, at a base exposure value b. To create a training-time input for $\mathcal{N}$, we re-expose (see Eq.1) the frames I_b to a higher exposure value $h = b + e$ forming a high exposure dataset, $\mathcal{D}_V = \{I_h, I_b\}$, as illustrated in Figure 3. The residual δ (see Eq. 2) will be the target for the residual-predicting network, $\mathcal{N}$, during training. At the inference stage, exposure I_b will be the input of $\mathcal{N}$ to generate higher and lower exposures (see Figure 4):

$$\hat{I}_l = \left[I_b \cdot \mathcal{N}(I_b)\right]_0^1 \qquad \hat{I}_h = \left[\frac{I_b}{\mathcal{N}(I_b)}\right]_0^1. \tag{3}$$

The value of the exposure difference, e, is set to +2 stops; we found this value to be the largest value we could use without leading to too large over-exposed areas in the re-exposed input frame. To ensure model robustness with respect to luminance and exposure variations, we employ a data augmentation technique, where both the starting exposure, I_b, and the higher exposure, I_h, are shifted by a small amount $s \sim \mathcal{U}(0, 0.25)$. This is a standard practice in inverse tone mapping [15, 41, 62] in order to have robust augmentations. In our case, the range of variations is relatively small because the input frames at training time have already been re-exposed by +2-stops. From our experimentation, a larger range than $(0, 0.25)$ may reduce the number of well-exposed pixels such that they would not be sufficient to properly learn the mapping.

3.2.1 Sampling Frames. As the information content between subsequent frames of a video is likely to be very similar, subsampling of frames can be used to speed up training [15]. We propose an approach that views the information content of the frames as a distribution and then samples from this distribution. In this work, we propose two approaches to achieve this.

The first is the use of regular sampling at a rate that is designed to be a common divisor of commonly used frame rates. In this work, we propose a regular subsampling at a rate of 6 (referred to as S6) which is the largest common divisor between the traditional frame rates of 24 and 30 as these are two of the most common frame rates for videos.

The second strategy is to sample a subset of frames proportional to the number of well-exposed pixels in each frame. This is motivated by the observation that the network requires well-exposed pixels to train the network, which can be found via sequential regular sampling, or equivalently the same information can be found by sampling a much smaller subset of frames that broadly contain the same information. We exploit this latter point by creating a discrete distribution H_{we} of frames, where each bin of the distribution is proportional to the percentage of well-exposed pixels in the frame, then sampling a number of frames from this distribution. For each frame, we compute the number of well-exposed pixels as

$$F(t) = \sum_{p \in pixels} \mathbb{1}^t(p), \tag{4}$$

where the indicator function $\mathbb{1}^t(p) = 1$ if the p-th pixel is in the range $[0.05, 0.95]$ and is 0 otherwise; and the sum is over all pixels in the t-th frame in a sequence. The probability of sampling the t-th frame of H_{we} is computed as

$$H_{we}[t] = \frac{F(t)}{\sum_{s \in frames} F(s)}, \tag{5}$$

where $frames$ is the length of the sequence. We use $n_s = 128$ frames sampled from H_{we}, this number being selected in early experiments, which we found balances training time and error, although our method is relatively robust to this value. We refer to this method as SU in this paper. We compare the S6 and SU strategies in an ablation study; see Section 4.5.

3.3 Loss Function

The loss function, $\mathcal{L}$, used for optimizing the model (see Figure 5), consists of three terms. The first term, $\mathcal{L}_\delta$, is the loss responsible for directly optimizing the residual mapping and the second, $\mathcal{L}_I$, is responsible for the overall image mapping consistency, and $\mathcal{L}_\tau$ maintains the temporal coherency at current frame t:

$$\begin{aligned}\mathcal{L} =& (1-\alpha)\left(\mathcal{L}_\delta\left(\hat{\delta}^t, \delta^t\right) + \mathcal{L}_I\left(\hat{I}_b^t, I_b^t\right)\right) + \\ & \alpha \mathcal{L}_\tau\left(\hat{I}_b^t, I_b^t, \hat{I}_b^{t+1}, I_b^{t+1}\right),\end{aligned} \tag{6}$$

where $\hat{\delta}^t = \mathcal{N}(I_h^t)$ is the residual prediction, $\hat{I}_b^t = \hat{\delta} I_h^t$ is the resulting base exposure prediction from the higher exposure frames I_h^t in the dataset, and α is a weight between the static and temporal losses (in our pilot experiments $\alpha = 0.95$ gave satisfactory results).

The residual loss, $\mathcal{L}_\delta$, is the L_2 loss because we want to penalize large changes in predicting the multiplicative residuals. The image space loss $\mathcal{L}_I$, consists of an L_1 distance term and a cosine similarity term that helps enforce color consistency:

$$\mathcal{L}_I = \frac{1}{N}\sum_{j=1}^{N} \|I_x^{t,j} - I_y^{t,j}\|_1 + \lambda\left(1 - \frac{1}{N}\sum_{j=1}^{N} \frac{I_x^{t,j} \cdot I_y^{t,j}}{\|I_x^{t,j}\|_2 \|I_y^{t,j}\|_2}\right) \tag{7}$$

where N is the total number of pixels of the image, I^j is the j-th RGB pixel vector of image I, and λ is a constant factor that adjusts the contribution of the cosine similarity term (in our pilot experiments $\lambda = 5$ gave satisfactory results).

The temporal loss $\mathcal{L}_\tau$ ensures temporal coherency by minimizing the temporal differences (i.e., current and next frame) between the target and the estimated frames as:

$$\mathcal{L}_\tau = L_2\left(I_b^{t+1} - I_b^t, \hat{I}_b^{t+1} - \hat{I}_b^t\right), \tag{8}$$

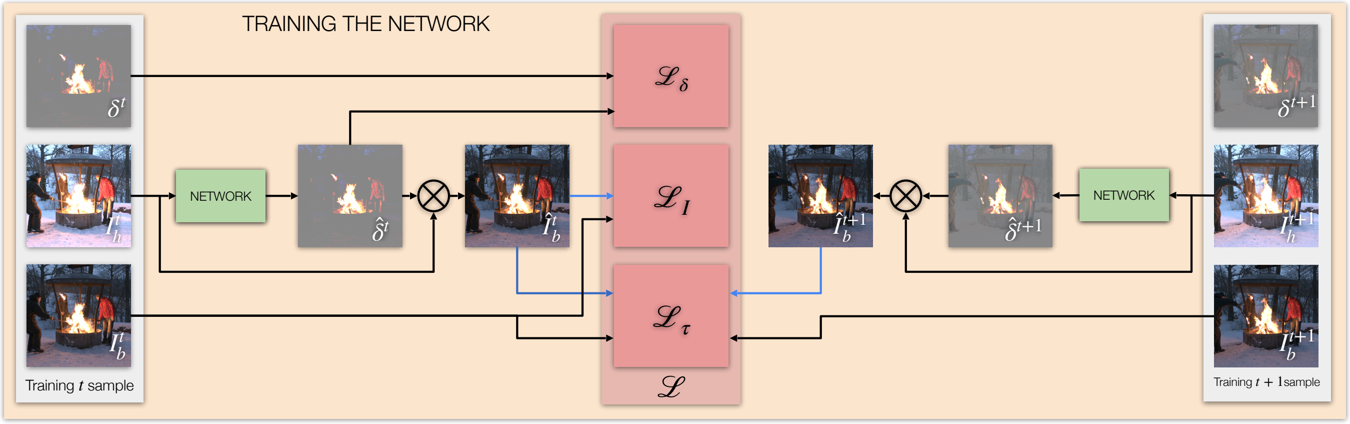


Fig. 5. Tailored training using the generated dataset from the input SDR video. Our goal is to train our network to predict δ values by minimizing a loss, $\mathcal{L}$, composed of three terms. The first term, $\mathcal{L}_\delta$, is the loss responsible for directly optimizing the residual mapping. The second term, $\mathcal{L}_I$, is responsible for the overall image mapping consistency. Finally, $\mathcal{L}_\tau$ maintains the temporal coherency at the current frame t.

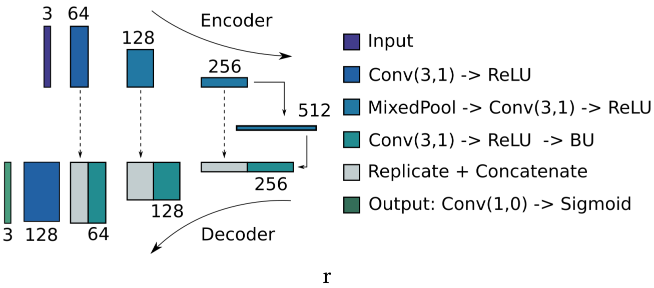


Fig. 6. Diagram of the network architecture used by $\mathcal{N}$. Conv(k,p) is a 2D convolutional layer with kernel size k and padding p. BU denotes bilinear upsampling by a factor of 2.

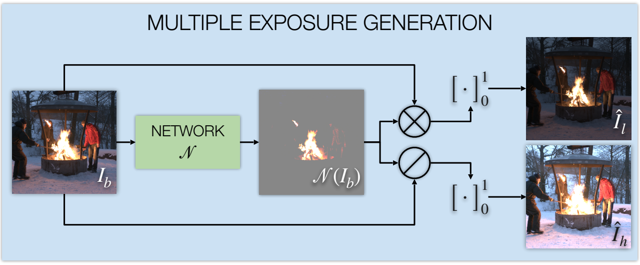


Fig. 7. The generation of exposures step in which the network predicts the residual for an input frame I_b. Then, an underexposed/overexposed frame is created by multiplying/dividing the input frame I_b by the residual, $\mathcal{N}(I_b)$.

we employ L_2 loss because we want to penalize large temporal changes that may introduce temporal artifacts.

3.4 Model

Our model, $\mathcal{N}$, has the goal of predicting δ. $\mathcal{N}$ is based on the UNet architecture [50] and consists of an encoder and a decoder part with skip-connections and 9 convolutional layers in total, see Figure 6. The standard ReLU activation is used but the use of batch normalization (BN) is avoided. This is because the BN layers were found to cause blob-like artifacts in our initial experiments, likely due to the change in input statistics when running at inference mode using a different exposure value as input. Fixed-Pooling [32], which is a learnable combination of max-pooling and average pooling, is used for downsampling in the encoder, while bilinear upsampling is used in the encoder. Figure 6 shows a diagram of the architecture.

To generate a frame at higher exposure, the input frame is divided by the residual (see Figure 7):

$$\hat{I}_h = \hat{I}_{b+e} = \left[\frac{I_b}{\mathcal{N}(I_b)}\right]_0^1. \tag{9}$$

Note that we apply clamping and rounding because we are interested in generating exposures in the SDR domain; such that the method produces only valid values in [0,1]. This process can be repeated on the generated frames. For example, if we want to generate a -4 stops exposure with $e = 2$, we need first to compute a -2 stops exposure:

$$\hat{I}_{b-2} = \left[I_b \cdot \mathcal{N}(I_b)\right]_0^1, \tag{10}$$

and then to compute our goal exposure as :

$$\hat{I}_{b-4} = \left[\hat{I}_{b-2} \cdot \mathcal{N}(I_{b-2})\right]_0^1. \tag{11}$$

3.5 Generating and Merging Exposures

Once the network, $\mathcal{N}$, is trained, we generate new exposure images as:

$$\hat{I}_i = \left[\hat{I}_{i-\mathrm{sgn}(i)}\left(\mathcal{N}(\hat{I}_{i-\mathrm{sgn}(i)})\right)^{-\mathrm{sgn}(i)}\right]_0^1 \qquad \text{where } \hat{I}_0 = I_b. \tag{12}$$

where I_b is the input frame, and sgn is a sign function.

Finally, the radiance map is computed in the logarithm domain using a weighting function as proposed by Debevec and Malik [11]:

$$\log E = \frac{\sum_i w(\hat{I}_i) \cdot \left(\log(g(\hat{I}_i) - \log t_i\right)}{\sum_i w(\hat{I}_i)}, \tag{13}$$

where g is the inverse camera response function, $w(x) = 1 - (2x - 1)^{12}$ is the hat weighting function for reducing reducing noise [2], and $t_i = 2^{ie}$ is the exposure time.

The computation of longer exposure is important in Equation 13 because it helps in improving the overall image quality in certain circumstances. For example, Figure 8 clearly shows our method avoids desaturation. In another example, Figure 9 also demonstrates how longer exposures help with masking noise.

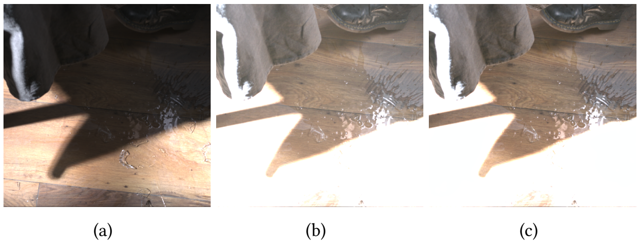


Fig. 8. An example of our method for dark areas for the Bistro_03 sequence: (a) The original SDR frame. (b) The frame in (a) re-exposed using a simple multiplication. (c) +4-stop exposure generated using our method starting from (a). Notice that colors are more desaturated in (b) than (c).

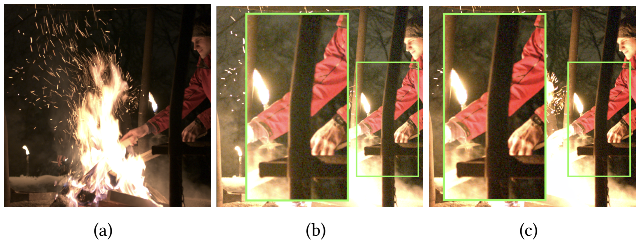


Fig. 9. An example of our method masking noise for the Fireplace_02 sequence: (a) The original SDR frame. (b) The frame in (a) re-exposed using a simple multiplication. (c) +4-stop exposure generated using our method starting from (a). Notice that noise is masked by using our method.

4 RESULTS

In this section, we present quantitative and qualitative results against fully supervised state-of-the-art methods: Santos et al.[51], Eilertsen et al.[15] using retrained parameters for temporal coherency[16], Endo et al. [17], Marnerides et al.[41], Liu et al. [38], Lee et al. [33], and Yu et al. [62], and Chen et al. [9]. The methods will be referred to as SAN (Santos et al.), EIL (Eilerstein et al.), EXP (Marnerides et al.), DRT (Endo et al.), LIU (Liu et al.), DH (Lee et al.), JOU (Chen et al.), LaNet (Yu et al.), and OUR (the presented method). We do not compare with any zero-shot, self-supervised, or semi-supervised methods, as to the best of our knowledge, none exist for inverse tone mapping. Note that we used the original authors' source code and weights for all these methods.

For evaluation, we gathered 43 HDR videos from two popular HDR video datasets: the Stuttgart HDR Video dataset (STU) [18] and the UBC DML-HDR dataset (UBC) [4, 5]. It is important to note that frames from these HDR videos were part of the training set of the state-of-the-art methods we compared against; but due to the scarcity of true HDR videos, not many datasets are available and the community will, commonly, use similar datasets. This is largely unavoidable and may have a detrimental effect on the results of our method in the comparisons. To demonstrate our method in fairer conditions, we employed a set of 4 HDR videos from the IC-1005 project (COST)[2], which are available by request and, to the best of our knowledge, have not been used in training of any of the compared state-of-the-art methods.

4.1 Training: Video Generation

For our method, the training for each video was performed independently, on an NVIDIA DGX Server 5.2.0 machine equipped with four AMD Epyc 7742 (64-core) CPUs i7-7800X (3.50 GHz) with 1 TB of memory and a single NVIDIA DGX A100 GPU with 40 GB of memory (CUDA 11.3). We implemented our model using the PyTorch 1.9.0 deep-learning framework.

To train our network, we employed mini-batch stochastic gradient descent and the Adam update rule [28] with the learning rate set to 10^{-4}. We left the rest of the parameters set to their default values; i.e., $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$. For each of our trained videos, we set the maximum number of epochs to 128. Typically, we reached a plateau of our loss around epoch 64-100. We trained using a batch size of 4. All the videos are of resolution 512 × 512 both at training and evaluation time. The duration of the training is constant for the SU method independently of the duration of the video because it extracts 128 samples; where an epoch requires on average 7.460 seconds to be completed on our machine. Note that the required time for completing an epoch has a linear relationship with n_s.

In terms of prediction time (the time required for expanding an SDR frame), the model maintains the linear complexity of UNets (i.e., linearly proportional to the number of input pixels). To generate four images at higher and lower f-stop (*i.e.* -4 stops, -2 stops, +2 stops, and +4 stops) from the input frames at HD resolution (i.e., 1920 × 1080) the model requires 6.74 milliseconds of computation.

To highlight that our method expands the dynamic range of input videos, we plot the luminance histogram of our model and the state of the art in Figure 15. Each histogram is computed using all 47 videos from the STU, UBC, and COST datasets. From such plots, we can notice that our method extends the dynamic range of SDR videos both in the bright and dark areas with a behavior very similar to EIL and SAN. This indicates that the methods that perform best according to the objective metrics generate similar histograms; however our method learns from the input video alone rather than being trained on a large corpus of HDR data.

[2] https://www.cost.eu/actions/IC1005/

Table 1. This table reports the mean values over each dataset (i.e., all videos) of PU-PSNR, PU-SSIM, and HDR-VDP2.2, and HDR-VDP3. OUR is the proposed method with SU sampling and the temporal loss. The red font color is for the best method, and the blue font for the second best method. For all metrics, higher values are better.

STU dataset				
Method	**PU-PSNR**	**PU-SSIM**	**HDR-VDP2**	**HDR-VDP3**
OUR	**35.26**	**0.987**	58.31	9.86
SAN	33.51	0.926	**60.63**	**9.90**
EIL	**35.06**	**0.927**	**61.16**	**9.91**
LIU	24.19	0.672	55.59	9.57
EXP	26.64	0.824	54.02	9.53
DRT	17.49	0.418	44.44	7.66
JOU	25.92	0.767	56.00	8.05
DH	22.24	0.620	55.51	9.60
LANet	28.21	0.76-	57.44	9.62
UBC dataset				
OUR	**41.12**	**0.993**	63.028	**9.95**
SAN	33.28	**0.985**	**64.60**	**9.96**
EIL	**33.35**	0.983	**63.77**	**9.95**
LIU	27.14	0.9446	58.82	9.76
EXP	22.77	0.863	56.88	9.66
DRT	17.80	0.736	50.27	7.96
JOU	21.46	0.870	55.99	8.46
DH	22.60	0.899	58.01	9.65
LANet	28.78	0.954	60.51	9.81
COST dataset				
OUR	**44.50**	**0.995**	**71.11**	**9.98**
SAN	**31.53**	**0.97**	68.34	**9.98**
EIL	31.44	0.972	**68.84**	**9.98**
LIU	22.41	0.885	59.96	9.74
EXP	28.92	0.940	62.45	**9.91**
DRT	14.72	0.748	51.12	8.32
JOU	21.12	0.803	58.91	9.85
DH	19.47	0.840	57.70	8.66
LANet	25.21	0.913	61.71	9.11

4.2 Quantitative with Reference

For quantitative results, the generated inverse tone-mapped videos for all the methods (including ours with SU sampling and the temporal loss) were compared with the ground truth using standard metrics for HDR applications and inverse tone mapping: HDR-VDP2.2 [47], HDR-VDP3 [40], PU-PSNR [3], and PU-SSIM [3]; for all these metrics the higher values correspond to better performance. PU-PSNR and PU-SSIM are modified versions of PSNR and SSIM[60] where input images are PU-encoded [39], before being processed by the metric, to handle how the human visual system perceives HDR data. HDR values follow the VESA DisplayHDR1400 standard[3] that has a peak luminance of 1, 400 cd/m^2 and a black level of 0.02 cd/m^2. To generate SDR input frames, we made use of automatic exposure with temporal filtering to reduce flickering for each frame:

$$I_b^t = \left[\left(I_{\mathrm{HDR}}^t \cdot 2^{f^t} \right)^{\frac{1}{2.2}} \right]_0^1 \tag{14}$$

where I_{HDR}^t is the t-th HDR frame, I_b^t is the SDR frame, $b = f^t$ is the exposure value (in f-stop) for the t-th HDR frame, and $[\cdot]_0^1$ is the same rounding and clipping operator as in Eq. 1. Note that the CRF is a simple gamma encoding to reduce the camera response function bias [14] when compared with the state-of-the-art methods.

Table 1 summarizes the comparisons for PU-PSNR, PU-SSIM, HDR-VDP2.2, and HDR-VDP3 for the individual datasets. Means are computed across all videos for each method and metric. Table 2, shows results across the datasets and includes statistical analysis. Using ANOVA there was a main effect for all metrics ($p < 0.05$). Subsequent pairwise comparisons with Bonferroni corrections were conducted. In the table, we group methods that are not significantly different from each other (at $p < 0.05$) together in colored brackets. Methods that are not shown in a group produced results that are significantly different from all others. As can be seen, OUR is in the first group for all metrics. These results show that our method works well in terms of PU-PSNR and PU-SSIM against the state-of-the-art for the STU and UBC datasets. Although it has reasonable results for HDR-VDP2.2 and HDR-VDP3, our method does not outperform SAN and EIL. These results reflect the same ranking as seen in Santos et al.'s work [51].

It is important to note that, apart from our proposed method, *the other methods were trained* using the STU and UBC datasets, which explains their performance with this metric. **Our method does not use any dataset because it self-learns from the input SDR video**.

However, when comparing our method against the state-of-the-art using a dataset that was not used by the other methods during their training, *i.e.* COST in this case, our method performs significantly better than the state-of-the-art across all metrics. This shows the applicability of the proposed method to generalize well as can be seen when comparing results across unseen datasets.

4.3 Quantitative without Reference

We also validated our method using real SDR videos captured using consumer hardware like a smartphone. We captured 21 SDR videos using an iPhone 12 Mini (see Table 3) covering different lighting conditions. Then, we converted these videos into HDR ones using our method and state-of-the-art methods. As in the previous section, HDR values follow the VESA DisplayHDR1400. Since these videos have no HDR reference, we employed PIQUE [46] with PU encoding [39]. PIQUE is a popular SDR no-reference metric that computes an image quality score for an image using a perception-based image quality evaluator. This metric, with HDR values, encoded using PU, is recommended for quality assessment of inverse tone mapping methods without a reference [19]. Table 3 shows the no-reference results using PU-PIQUE for each scene (showing an SDR input frame on top) and the overall average. From this result, our method outperforms the state-of-the-art in terms of overall performance and also achieves first and second place for the majority of the sequences.

4.4 Visual Inspection

We also show qualitative results, comparing our method with the state-of-the-art and the original HDR frames.For all methods, the input is a 0-stop image from the HDR ground truth (GT). All results show frames at different exposure levels in which we applied a simple gamma encoding with $\gamma = 2.2$. There is no tone mapping

[3] https://displayhdr.org/

PU-PSNR	OUR (37.29)	EIL (34.39)	SAN (33.29)	LANet (28.07)	LIU (26.97)	EXP (26.01)	JOU (24.56)	DH (22.07)	DRT (17.32)
PU-SSIM	OUR (0.99)	EIL (0.95)	SAN (0.94)	LIU (0.92)	EXP (0.84)	LANet (0.82)	JOU (0.79)	DH (0.66)	DRT (0.51)
HDR-VDP2	EIL (62.11)	SAN (61.96)	OUR (61.54)	LANet (58.46)	LIU (56.52)	JOU (56.24)	DH (56.23)	EXP (55.33)	DRT (46.27)
HDR-VDP3	EIL (9.92)	SAN (9.92)	OUR (9.89)	LIU (9.63)	LANet (9.62)	EXP (9.59)	DH(9.52)	JOU (8.29)	DRT (7.76)

Table 2. Overall order across all datasets grouped via statistical significance. Colored groupings demonstrate no significant changes using pairwise comparisons with Bonferroni corrections at $p < 0.05$. Individual methods not within a group are significantly different from all others.

Table 3. Results from the no-reference study using real-world data (21 SDR videos) captured with an iPhone 12 Mini smartphone (lower values are better). Results computed with PU-PIQUE[19]; lower values are better. The red font color is for the best method and the blue font for the second best method. See the additional material for more details on the used scene.

SDR Videos dataset									
	OUR	**SAN**	**EIL**	**LIU**	**EXP**	**DRT**	**JOU**	**DH**	**LaNet**
Average	**37.02**	40.34	42.32	55.21	**39.65**	42.70	58.23	42.97	47.84

operator applied to such frames to avoid distortions that tone mapping may introduce in frames both in terms of perceived luminance levels and colors. All frames and videos are corrected using Hanji et al.'s CRF correction[19].

Figure 1 and Figure 10 show an example of our method applied to a challenging scene showing our method reconstructs detail in overexposed areas of the frames including reconstructing texture and colors even in the presence of motion blur.

Figure 17 is a challenging example where there is rapid motion and texture details, colors, and a significant lack of dynamic range in the input. Our method can generate similar details in terms of color, dynamic range, and texture. When compared to EIL, for example, our method manages to recover more texture and details in the flames (avoiding checkerboard artifacts due to transposed convolutions), obtaining similar results to SAN with less unnatural high-frequencies; see Figure 12.

Figure 13 shows an important feature of our method that self-learns from the video. In this figure, we have a sequence, "Windows", from Table 3, the sequence shows a complex window, which is unique. EIL and SAN, the best methods from the state-of-the-art struggle in reconstructing this pattern because it is probably not present in their training dataset. On the other hand, our method self-learns this pattern from the other well-exposed portion of the video during our self-supervised training on the video itself. Therefore, it can manage the reconstruction of that pattern.

Figure 14 has complex light sources that are mostly clipped. Our method can achieve a plausible reconstruction similar to SAN and EIL. Likewise, Figure 11 has clipped light sources and texture details on the dress. These are reconstructed well with our method, and the result is comparable to the other state-of-the-art methods.

4.5 Ablation Studies

We conducted three ablation studies to validate our approach and choices regarding what the network should predict, how to sample a video, the benefits of taking into account temporal coherence.

Residuals: In the first of such studies, we validated the use of predicting residuals for generating higher and lower exposure frames. To achieve this, we compared our method with residuals (Residual),

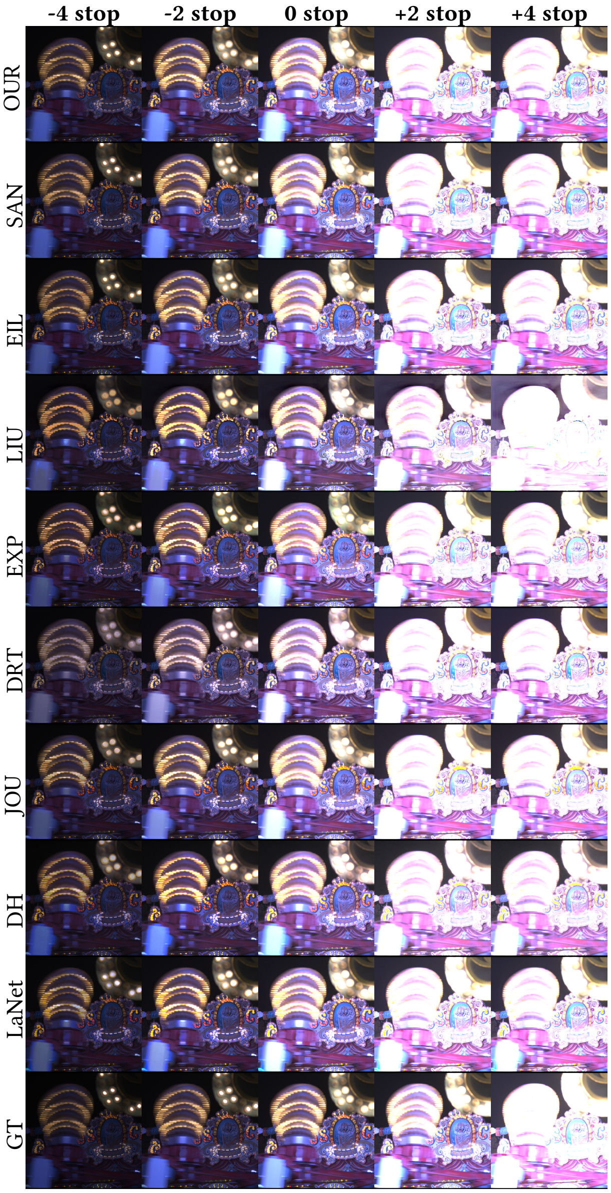


Fig. 10. This figure compares our approach against state-of-the-art methods for the challenging Carousel_Fireworks_02 sequence [18]. This shows that OUR method can reconstruct details in the light sources yet only relies on the original SDR content.

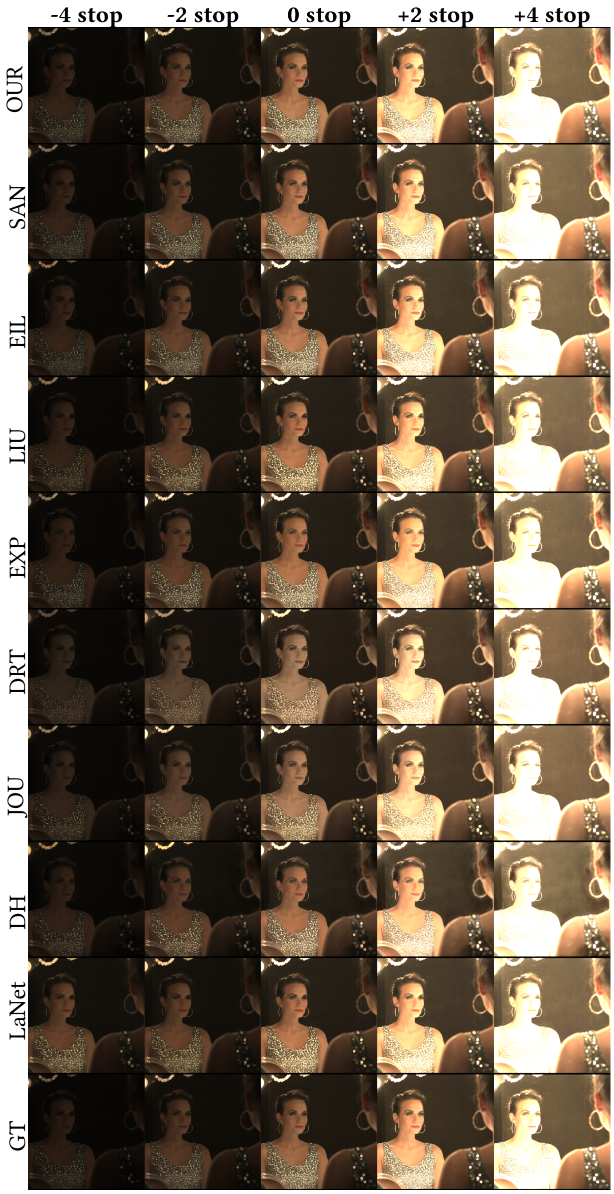


Fig. 11. A visual comparison of all tested methods. The frame is part of the sequence Showgirl_01 [18]. This shows that the OUR method can reconstruct details in the lights and dress and performs similarly to other methods yet only relies on the original SDR content.

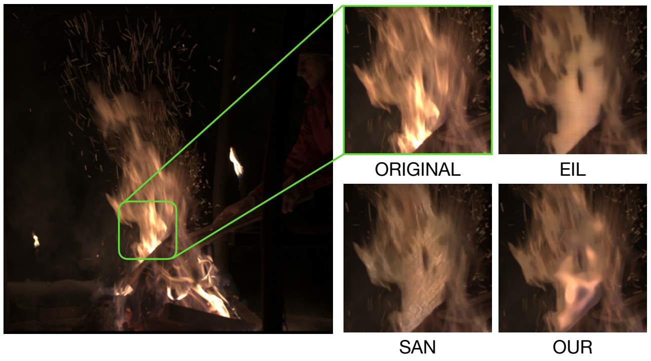


Fig. 12. An example at -4-stop from Figure 17 comparing EIL, SAN, and OUR method in the detailed region. EIL shows fewer details than the original HDR image and checkerboard artifacts due to the use of transposed convolutions. SAN creates more details, and it adds a lot of high-frequency details that may look unnatural. OUR approach sits in the middle providing a trade-off between details and smoothness.

Table 4. This table reports the results of the ablation study when predicting residuals (Residual) or an exposure (End2End); the mean values over each dataset (i.e., all videos) of PU-PSNR, PU-SSIM, and HDR-VDP2.2, and HDR-VDP3. This study was computed with S6 sampling and without the temporal loss. The red font color highlights the best method. For all metrics, higher values are better.

STU dataset				
Method	**PU-PSNR**	**PU-SSIM**	**HDR-VDP2**	**HDR-VDP3**
Residual	**35.04**	**0.985**	**58.28**	**9.83**
End2End	34.08	0.982	57.72	9.79
UBC dataset				
Residual	**39.19**	**0.992**	**63.56**	**9.94**
End2End	35.81	0.986	61.12	9.83
COST dataset				
Residual	**44.15**	**0.993**	**59.20**	**9.91**
End2End	34.20	0.957	57.77	9.78

see Figure 7, against the same network trained on predicting higher and lower exposures (End2End). In this study, we employed S6 sampling and we disabled the temporal loss for the sake of simplicity. Table 4 reports the results of this study and it clearly confirms that predicting residuals instead of an entire image produces higher-quality results.

Sampling Frames: In the second study, we determined the effectiveness of the SU sampling strategy versus S6; the regular sampling every six frames. Therefore, we trained our network with SU and S6 for all datasets (we disabled the temporal loss for the sake of simplicity); Table 5 shows the results of this study. From these results, the SU sampling method performs overall better than S6 for the majority of metrics. This shows that the method provides high-quality results and saves computational time.

Temporal Loss: In the third ablation study, we further tested our method to understand if the temporal loss is beneficial. For this test, we added the Smoothness metric (S) [14] to determine if the video is less smooth ($\mathbf{S} < 1.0$) or smoother ($\mathbf{S} \geq 1$) than the original HDR video. Table 6 reports the results of this study. This elicits that the temporal loss always generates videos that are temporally smooth $\mathbf{S} \geq 1$ as much as the original HDR video in the reconstructed areas. Furthermore, the temporal loss provides, in the overall, better values of PU-PSNR, PU-SSIM, HDR-VDP2.2, and HDR-VDP3 than without it; in the majority of cases. An important thing to highlight is that the method without the temporal loss can achieve, in the majority

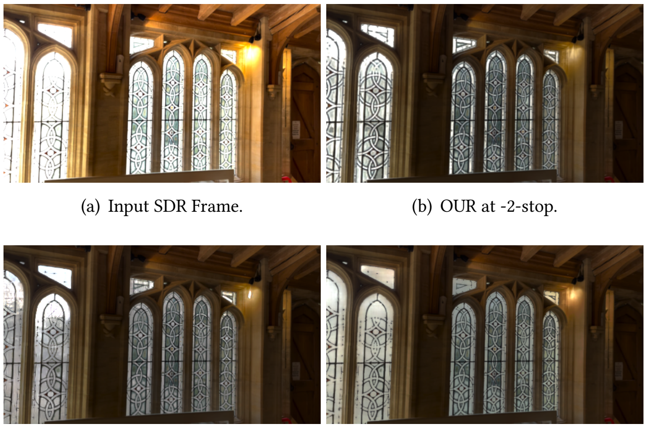

(a) Input SDR Frame.
(b) OUR at -2-stop.
(c) SAN at -2-stop.
(d) EIL at -2-stop.

Fig. 13. An example of the video Windows used in Section 4.3 and Table 3. In this example, the input SDR frame (a) exhibits a window with a complex pattern. Since this pattern is unique it may be difficult to reconstruct properly if it is not present in the learning dataset for dataset-based techniques. OUR technique can reconstruct this pattern better than SAN and EIL (the best methods from the state-of-the-art) because it self-learns this pattern from its input video.

Table 5. Results from the ablation study predicting residuals using the two sampling strategies S6 and SU; the mean values over each dataset (i.e., all videos) of PU-PSNR, PU-SSIM, HDR-VDP2.2, and HDR-VDP3. The temporal loss is disabled for the sake of clarity. As before, the red font color is for the best method. For all metrics, higher values are better.

Method	PU-PSNR	PU-SSIM	HDR-VDP2	HDR-VDP3
		STU dataset		
Residual+SU	**35.07**	**0.987**	**58.36**	**9.83**
Residual+S6	35.04	0.985	58.28	**9.83**
		UBC dataset		
Residual+SU	**41.24**	**0.993**	62.12	**9.95**
Residual+S6	39.19	0.992	**63.56**	9.94
		COST dataset		
Residual+SU	43.72	**0.996**	**69.86**	**9.91**
Residual	**44.15**	0.993	59.20	**9.91**

of cases reasonable performance. However, the use of temporal loss improves the quality of the outcome.

4.6 Improving Quality and Computational Efficiency

In this section, we show how to further improve quality and timings for our method using straightforward strategies. With the method described in Section 3 the training time for a single epoch is on average 7.460 seconds on the hardware used in this study. This means that reaching the plateau (64-128 epochs) requires between 8-15 minutes, and this is independent of the length of the video.

Table 6. This table shows the results of the temporal loss ablation study (with/without); the mean values over each dataset (i.e., all videos) of PU-PSNR, PU-SSIM, HDR-VDP2.2, HDR-VDP3, and Smoothness S. The red font color is for the best method. For all metrics, higher values are better.

Method	PU-PSNR	PU-SSIM	HDR-VDP2	HDR-VDP3	S
			STU dataset		
Residual+SU+Temporal	**35.26**	**0.987**	**58.31**	**9.86**	**1.29**
Residual+SU	35.07	**0.987**	58.36	9.83	1.28
			UBC dataset		
Residual+SU+Temporal	41.12	**0.993**	**63.03**	**9.95**	**1.00**
Residual+SU	**41.24**	0.992	62.12	**9.95**	0.98
			COST dataset		
Residual+SU+Temporal	**44.50**	**0.996**	**71.11**	**9.98**	1.02
Residual+SU	43.72	**0.996**	69.86	9.91	**1.03**

Therefore, we firstly propose a method that uses batches of videos to further improve quality, and then focus on computational efficiency by proposing two strategies to significantly speed up the self-training phase: 1) pre-training on a dataset of SDR videos followed by fine-tuning when expanding a video; 2) reducing the samples needed to train the network.

Batch Training: In many real-world scenarios, users capture similar videos (for example at the same location), and they may need to expand all of them together. Therefore, we studied what happens when all our videos are trained together (STU, UBC, and COST datasets); i.e., picking 128 samples from each video. Table 7 reports the results of this experiment titled “128-sample Batch”. From these results, the quality has improved when compared to OUR method which trains each video separately. However, the training time for epoch increases greatly to 164 seconds. The following section helps mitigate this increase.

Table 7. Results from the optimization strategy reducing samples; the mean values over each dataset (i.e., all videos) of PU-PSNR, PU-SSIM, HDR-VDP2.2, and HDR-VDP3. The temporal loss is disabled for the sake of clarity. As before, the red font color is the best method, and the blue font color is the second best method. For all metrics, higher values are better.

Method	PU-PSNR	PU-SSIM	HDR-VDP2	HDR-VDP3
		STU dataset		
Residual+SU+Temporal	35.26	0.987	58.31	**9.83**
128-Sample Batch	**36.54**	**0.989**	**58.99**	**9.84**
4-Sample	34.89	0.987	58.28	9.82
Small-Sample Batch	**35.80**	**0.988**	**58.71**	**9.83**
		UBC dataset		
Residual+SU+Temporal	41.12	0.993	63.03	9.95
128-Sample Batch	**43.28**	**0.996**	**66.45**	**9.97**
4-Sample	41.02	**0.994**	**65.86**	**9.96**
Small-Sample Batch	**41.86**	**0.994**	64.16	**9.96**
		COST dataset		
Residual+SU+Temporal	**44.50**	**0.996**	**71.11**	**9.98**
128-Sample Batch	**45.82**	**0.997**	65.23	**9.99**
4-Sample	39.62	0.993	60.95	9.937
Small-Sample Batch	43.59	**0.996**	**71.16**	**9.98**

Reducing Samples: To speed up the computation time, we exploit the trade-off between training time and the number of sample frames, n_s. Therefore, we ran our method using $n_s = 4$ instead of

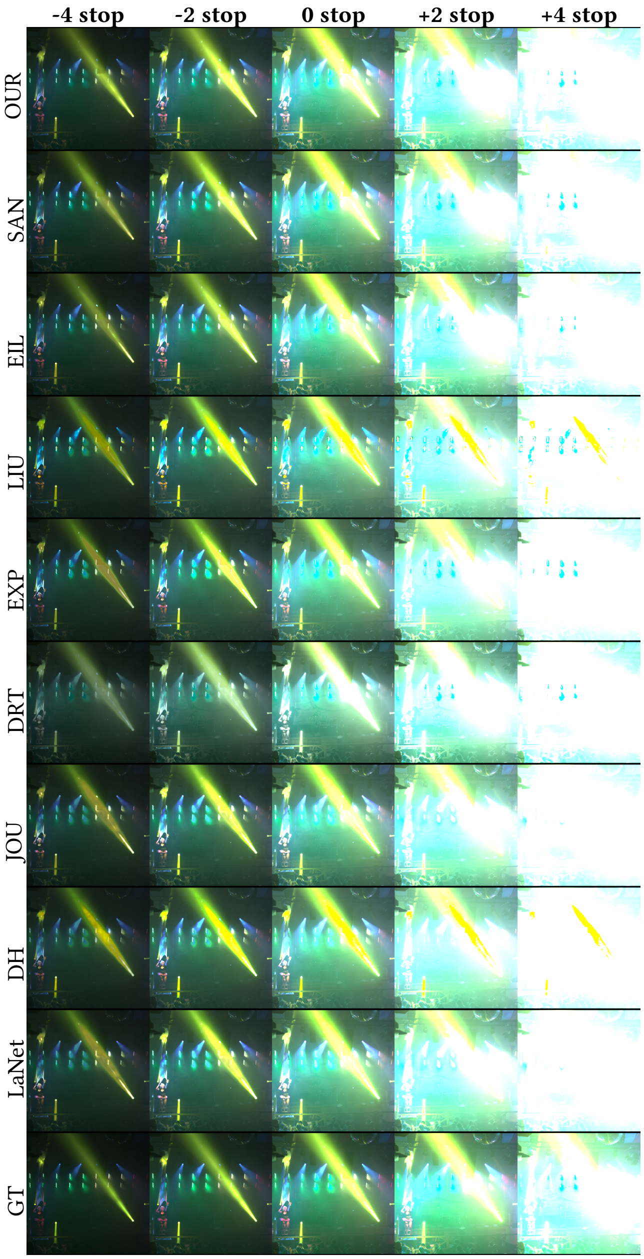


Fig. 14. This image shows a frame from the sequence Beerfest_04 [18]. Our method has a reconstruction performance similar to SAN and EIL and is able to plausibly reconstruct the lighting in this scene.

$n_s = 128$ as proposed in Section 3.2.1 using our sampling strategy SU. Table 7 shows the results of this study titled "4-Sample". From this, it is clear that $n_s = 4$ reduces the overall quality, although the results are still better than some the state of the art methods; see Table 1. With regards to timings, the method takes less than 0.659 seconds to complete a single epoch; completing the entire training in less than two minutes.

In light of this encouraging result we extended our batching strategy using a total of $n_s = 128$, by taking 2-3 frames for each video using our sampling strategy SU. The idea is to train together all videos from our datasets (STU, UBC, and COST) at the same cost as a single video. Taking only 7.46 seconds per epoch, this speeds the process up by a factor of 47 times compared to train each single video with our method. Table 7 shows the results of this study titled "Small-Sample Batch". From this test, it is clear that quality is higher than training each single video separately, and it is a viable optimization when a batch of videos needs to be expanded together.

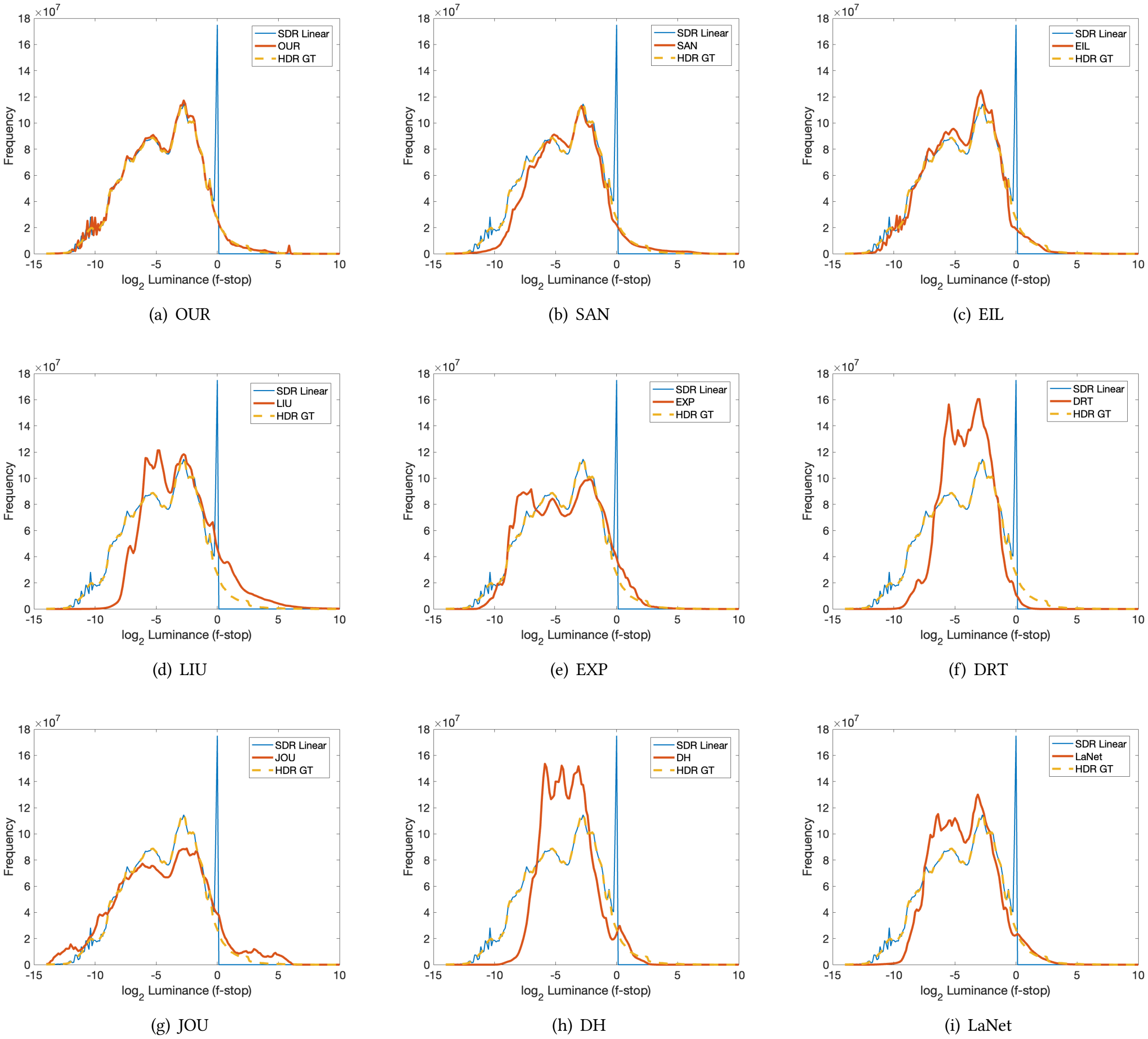
Pre-training with Fine-Tuning: Another viable approach for improving computational efficiency is pre-training followed by fine-tuning. The first step of this strategy is to pre-train our network using a dataset of SDR videos. Then, when a novel SDR video (not present in the pre-training dataset) needs to be expanded, we load the pre-trained weights, and fine-tune the weights using our self-supervised scheme for a few epochs. For this study, we used 23 SDR videos as a pre-training dataset. The pre-training required less than 7 hours to reach a plateau. We show results for 1, 4, and 8 epochs of fine-tuning and these take, respectively 7.46 seconds, 30.01 seconds, and 58.83 seconds to train after the pre-training stage. Table 8 shows the results of this experiment. From these results, it is noticeable that pre-training achieves further improved results and the 8-epoch fine-tuning requires less than a minute. These results are to be expected as this solution combines the strengths of dataset-based methods with the proposed approach to tailor the weights to the specific video.

4.7 Hanji Correction

For further comparison, we analyzed our data applying Hanji et al.'s CRF correction [19]. This correction improves the CRF inversion by exploiting the ground truth or reference image, which is strictly required for the correction. While the reference image is not typically available for the larger characterization of the inverse tone mapping problem, we include these results for completeness. Table 9 reports the results of such comparisons. This shows that even in the presence of the additional non-linearity; i.e., the correction, our method is still competitive with the state-of-the-art.

4.8 Limitations

To learn texture and dynamic range details from a single SDR video in an effective way, our network needs to view moving people/objects and/or to view the scene from different points of view through camera motion. This is because over-exposed or under-exposed parts of the video may become well-exposed when these parts are not static. When the motion in an input SDR video is limited, our network may not be able to discover how to recover texture and dynamic range in under-exposed and over-exposed parts of the video. Figure 16 shows a frame from the sequence Bistro_01 [18] and the expanded frames at exposures -4-stop and -2-stop. This scene has very limited motion, only the candles in the background (green square), which limits the recovery capabilities of our method to only the flames of the candles.



(a) OUR

(b) SAN

(c) EIL

(d) LIU

(e) EXP

(f) DRT

(g) JOU

(h) DH

(i) LaNet

Fig. 15. Pixel values from our model and the start of the art methods for COST, STU, and UBC datasets. For each method, we plot the original SDR data histogram (“SDR Linear”) in blue, the HDR ground-truth data (“HDR GT”) in yellow, and the expanded HDR data of that method in red. (a) shows that our method extends the dynamic range for both underexposed and underexposed pixel values.

5 CONCLUSIONS AND FUTURE WORK

In this work, we have shown that a self-supervised approach can expand the dynamic range of SDR videos and it is possible to recover both missing details in terms of texture and dynamic range. To achieve this, we have employed self-supervised strategies. The proposed method can achieve high-quality results that improve on fully-supervised state-of-the-art techniques both visually and in terms of several metrics. This is particularly useful as it does not require reliance upon an external HDR dataset. One important advantage of our method is the ability to learn patterns that are unique in the video and are not present in a dataset; see Figure 13.

The method performs best when there is motion in the video; ideally both in terms of people/objects and camera motion that exhibit different exposures across the frames such that the training process can form a fuller understanding of the scene’s dynamic range. To improve quality and computational efficiency, we have proposed different strategies to improve computational efficiency including pre-training followed by fine-tuning, and training batches of videos together while reducing the number of samples n_s. Both approaches achieve a significant speedup whilst also maintaining high quality.

This work confirms our hypothesis that SDR videos can be expanded without an external dataset and produce reasonably high-quality results that are competitive with fully-supervised methods. In future

Table 8. Results from the optimization strategy using Pre-training and fine tuning on the video itself at different epochs; the mean values over each dataset (i.e., all videos) of PU-PSNR, PU-SSIM, HDR-VDP2.2, and HDR-VDP3. The temporal loss is disabled for the sake of clarity. As before, the red font color is the best method, and the blue font color is the second best method. For all metrics, higher values are better.

STU dataset				
Method	**PU-PSNR**	**PU-SSIM**	**HDR-VDP2**	**HDR-VDP3**
Residual+SU+Temporal	35.26	**0.987**	58.31	9.87
Pre-train + 1-epoch Fine Tuning	37.08	**0.991**	59.81	**9.88**
Pre-train + 4-epoch Fine Tuning	**37.41**	**0.991**	**59.90**	**9.88**
Pre-train + 8-epoch Fine Tuning	**37.55**	**0.991**	**60.13**	**9.89**
UBC dataset				
Residual+SU+Temporal	41.12	**0.993**	63.03	**9.96**
Pre-train + 1-epoch Fine-Tuning	**41.90**	**0.995**	**66.34**	**9.96**
Pre-train + 4-epoch Fine-Tuning	41.86	**0.995**	66.16	**9.96**
Pre-train + 8-epoch Fine-Tuning	**42.36**	**0.995**	**66.33**	**9.97**
COST dataset				
Residual+SU+Temporal	44.50	**0.995**	**71.11**	**9.98**
Pre-train + 1-epoch Fine-Tuning	**44.92**	**0.997**	64.64	**9.98**
Pre-train + 4-epoch Fine-Tuning	**45.21**	**0.997**	**64.89**	**9.98**
Pre-train + 8-epoch Fine-Tuning	43.83	**0.997**	64.67	**9.97**

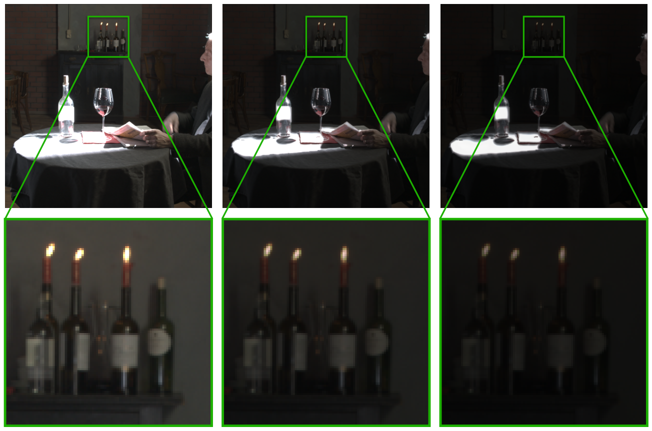


Fig. 16. A limitation example (Bistro_01) [18]). Here our method can only reconstruct the candles in the green square because they move while the rest of the overexposed scene (the table and the bottles) does not have motion. Top: On the left, the input SDR frame; In the middle, the recovered frame at -2 stops using our method; On the right, the recovered frame at -4 stops using our method. Bottom: the zooms of the respective green square areas.

work, we would like to generalize our method and apply it to existing ITMOs for fine-tuning to provide temporal coherency and optimize training weights to the content of the input video.

ACKNOWLEDGEMENTS

We thank Jan Fröhlich and his team for the Stuttgart HDR Video dataset, and Panos Nasiopoulos and his team for the UBC HDR Video dataset. We also thank Gabriel Eilertsen, Marcel Santos, and Param Hanji for help regarding comparisons, and Rafał Mantiuk, Massimiliano Corsini, and Fabio Pellacini for insightful comments on the manuscript.

Table 9. This table reports the mean values over each dataset (i.e., all videos) of PU-PSNR, PU-SSIM, HDR-VDP2.2, and HDR-VDP3. OUR is the proposed method with SU sampling and temporal loss. The red font color is the best method, and the blue font color is for the second best method. For all metrics, higher values are better. This table reports results with application of the Hanji et al.'s CRF correction[19].

STU dataset				
Method	**PU-PSNR**	**PU-SSIM**	**HDR-VDP2**	**HDR-VDP3**
OUR	**34.08**	**0.971**	59.57	9.87
SAN	33.60	0.968	**61.01**	**9.90**
EIL	**35.805**	**0.978**	**61.17**	**9.91**
LIU	26.41	0.915	55.45	9.71
EXP	28.77	0.906	55.04	9.64
DRT	27.03	0.912	53.40	9.50
JOU	31.32	0.958	56.95	9.77
DH	29.42	0.945	55.78	9.71
LANet	29.24	0.921	56.51	9.73
UBC dataset				
OUR	**42.25**	**0.995**	**66.90**	**9.96**
SAN	**44.39**	**0.997**	**68.67**	**9.96**
EIL	41.82	**0.985**	66.62	**9.95**
LIU	32.00	0.969	61.40	9.90
EXP	34.85	0.985	61.47	9.91
DRT	36.31	0.986	60.76	9.92
JOU	37.53	0.988	61.51	9.93
DH	36.04	0.988	63.41	9.92
LANet	34.15	0.980	63.29	9.88
COST dataset				
OUR	**42.05**	**0.996**	**64.79**	**9.98**
SAN	39.59	**0.995**	65.44	**9.99**
EIL	**41.26**	**0.995**	**66.25**	**9.99**
LIU	32.00	0.969	61.40	9.90
EXP	34.85	0.985	61.47	9.91
DRT	36.31	0.986	60.76	9.92
JOU	37.53	0.988	61.51	9.93
DH	34.47	0.983	60.07	9.90
LANet	35.45	0.982	62.65	9.93

REFERENCES

[1] Ahmet Oguz Akyüz and Erik Reinhard. 2007. Noise reduction in high dynamic range imaging. *J. Vis. Commun. Image Represent.* 18, 5 (2007), 366–376. https://doi.org/10.1016/j.jvcir.2007.04.001

[2] Ahmet Oğuz Akyüz and Erik Reinhard. 2007. Noise Reduction in High Dynamic Range Imaging. *Journal of Visual Communication and Image Representation* 18, 5 (2007), 366–376.

[3] Tunç Ozan Aydın, RafałMantiuk, and Hans-Peter Seidel. 2008. Extending quality metrics to full luminance range images. In *Human Vision and Electronic Imaging XIII, San Jose, CA, USA, January 27, 2008 (SPIE Proceedings, Vol. 6806)*, Bernice E. Rogowitz and Thrasyvoulos N. Pappas (Eds.). SPIE, 68060B.

[4] Maryam Azimi, Amin Banitalebi-Dehkordi, Yuanyuan Dong, Mahsa T. Pourazad, and Panos Nasiopoulos. 2014. Evaluating the Performance of Existing Full-Reference Quality Metrics on High Dynamic Range (HDR) Video Content. In *ICMSP 2014: XII International Conference on Multimedia Signal Processing, Venice, Italy*. –.

[5] A. Banitalebi-Dehkordi, M. Azimi Hashemi, M.T. Pourazad, and P. Nasiopoulos. 2014. Compression of high dynamic range video using the HEVC and H.264/AVC standards. In *10th International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness, QShine 2014, Rhodes, Greece, August 18-20, 2014*. IEEE, 8–12. https://doi.org/10.1109/QSHINE.2014.6928652

[6] Francesco Banterle, Patrick Ledda, Kurt Debattista, and Alan Chalmers. 2006. Inverse Tone Mapping. In *GRAPHITE '06* (Kuala Lumpur, Malaysia). ACM, New York, NY, USA, 349–356. https://doi.org/10.1145/1174429.1174489

[7] Cambodge Bist, Rémi Cozot, Gérard Madec, and Xavier Ducloux. 2017. Tone expansion using lighting style aesthetics. *Comput. Graph.* 62 (2017), 77–86. https://doi.org/10.1016/j.cag.2016.12.006

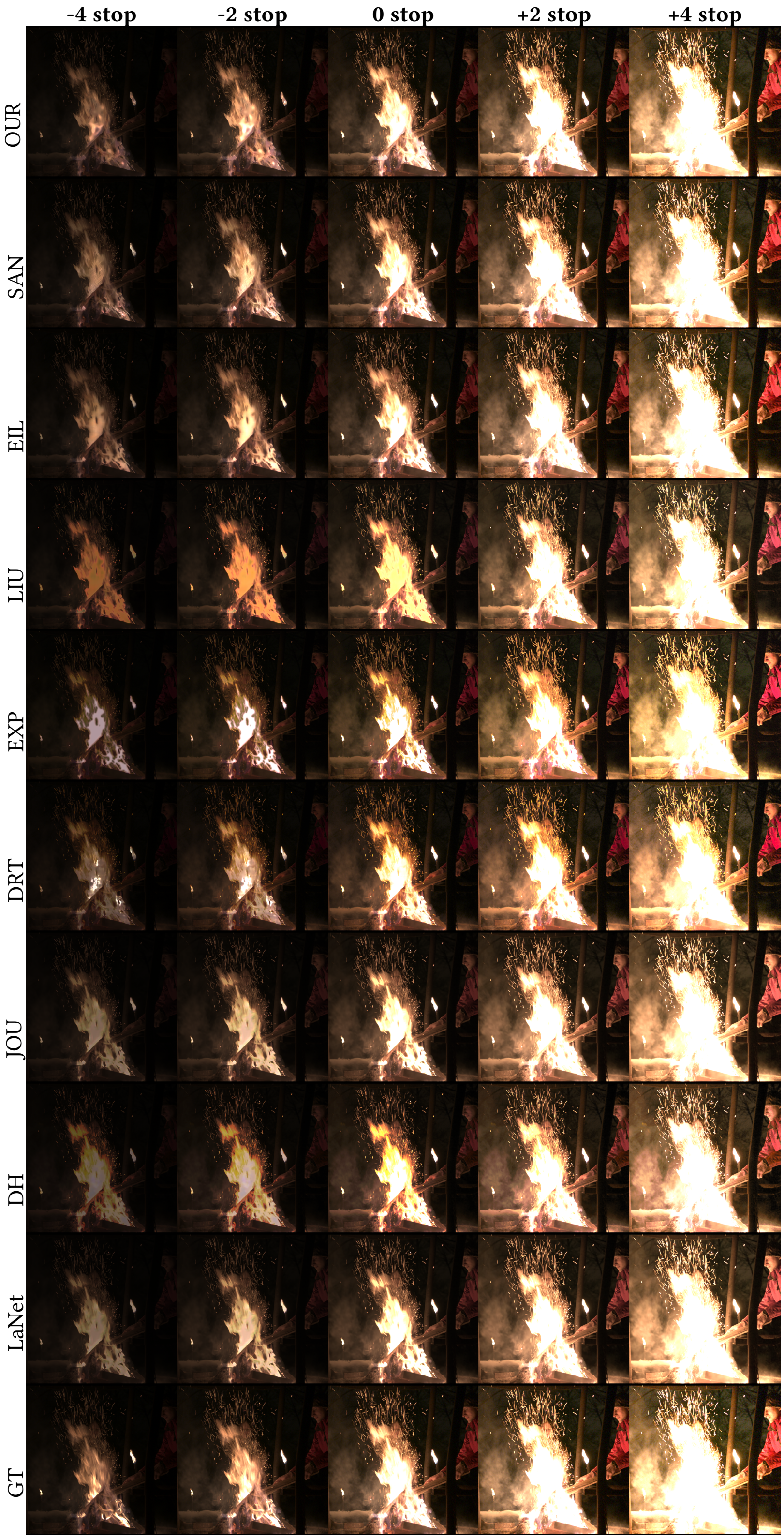


Fig. 17. This shows a frame from the sequence Fireplace_02 [18]. Different exposures of the reconstructed HDR image are shown from left to right, and different ITMOs are shown vertically. This illustrates that our method can generate similar details to the Ground Truth in terms of color, dynamic range, and texture.

[8] Guanying Chen, Chaofeng Chen, Shi Guo, Zhetong Liang, Kwan-Yee K. Wong, and Lei Zhang. 2021. HDR Video Reconstruction: A Coarse-to-fine Network and A Real-world Benchmark Dataset. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*. IEEE, 2482–2491. https://doi.org/10.1109/ICCV48922.2021.00250

[9] Xiangyu Chen, Zhengwen Zhang, Jimmy S Ren, Lynhoo Tian, Yu Qiao, and Chao Dong. 2021. A New Journey From SDRTV to HDRTV. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. IEEE, 4500–4509.

[10] Paul Debevec. 2002. Image-Based Lighting. *IEEE Comput. Graph. Appl.* 22, 2 (2002), 26–34. https://doi.org/10.1109/38.988744

[11] Paul E. Debevec and Jitendra Malik. 1997. Recovering high dynamic range radiance maps from photographs. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1997, Los Angeles, CA, USA, August 3-8, 1997*, G. Scott Owen, Turner Whitted, and Barbara Mones-Hattal (Eds.). ACM, 369–378. https://doi.org/10.1145/258734.258884

[12] Paul E. Debevec and Jitendra Malik. 1997. Recovering High Dynamic Range Radiance Maps from Photographs. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '97)*. ACM Press/Addison-Wesley Publishing Co., USA, 369–378. https://doi.org/10.1145/258734.258884

[13] Piotr Didyk, Rafał Mantiuk, Matthias Hein, and Hans-Peter Seidel. 2008. Enhancement of Bright Video Features for HDR Displays. *Computer Graphics Forum* 27, 4 (2008), 1265–1274.

[14] Gabriel Eilertsen, Saghi Hajisharif, Param Hanji, Apostolia Tsirikoglou, Rafal K. Mantiuk, and Jonas Unger. 2021. How to cheat with metrics in single-image HDR reconstruction. In *IEEE/CVF International Conference on Computer Vision Workshops, ICCVW 2021, Montreal, BC, Canada, October 11-17, 2021*. IEEE, 3981–3990. https://doi.org/10.1109/ICCVW54120.2021.00445

[15] Gabriel Eilertsen, Joel Kronander, Gyorgy Denes, Rafał K. Mantiuk, and Jonas Unger. 2017. HDR image reconstruction from a single exposure using deep CNNs. *ACM Trans. Graph.* 36, 6 (2017), 178:1–178:15. https://doi.org/10.1145/3130800.3130816

[16] Gabriel Eilertsen, Rafał K. Mantiuk, and Jonas Unger. 2019. Single-Frame Regularization for Temporally Stable CNNs. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*. Computer Vision Foundation / IEEE, 11176–11185. https://doi.org/10.1109/CVPR.2019.01143

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

[18] Jan Froehlich, Stefan Grandinetti, Bernd Eberhardt, Simon Walter, Andreas Schilling, and Harald Brendel. 2014. Creating Cinematic Wide Gamut HDR-Video for the Evaluation of Tone Mapping Operators and HDR-Displays. In *Proceedings IS&T/SPIE Electronic Imaging, 2014*, Vol. 9023. SPIE, 9023 – 9023 – 10. https://doi.org/10.1117/12.2040003

[19] Param Hanji, Rafal Mantiuk, Gabriel Eilertsen, Saghi Hajisharif, and Jonas Unger. 2022. Comparison of Single Image HDR Reconstruction Methods — the Caveats of Quality Assessment. In *ACM SIGGRAPH 2022 Conference Proceedings* (Vancouver, BC, Canada) *(SIGGRAPH '22)*. Association for Computing Machinery, New York, NY, USA, Article 1, 8 pages. https://doi.org/10.1145/3528233.3530729

[20] Samuel W. Hasinoff, Dillon Sharlet, Ryan Geiss, Andrew Adams, Jonathan T. Barron, Florian Kainz, Jiawen Chen, and Marc Levoy. 2016. Burst Photography for High Dynamic Range and Low-Light Imaging on Mobile Cameras. *ACM Trans. Graph.* 35, 6, Article 192 (nov 2016), 12 pages. https://doi.org/10.1145/2980179.2980254

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

[22] ITU-R. 2018. Recommendation ITU-R BT.2100-2: Image parameter values for high dynamic range television for use in production and international programme exchange.

[23] So Yeon Jo, Siyeong Lee, Namhyun Ahn, and Suk-Ju Kang. 2021. Deep Arbitrary HDRI: Inverse Tone Mapping with Controllable Exposure Changes. *IEEE Transactions on Multimedia* 24 (2021), 2713–2726.

[24] Nima Khademi Kalantari and Ravi Ramamoorthi. 2019. Deep HDR Video from Sequences with Alternating Exposures. *Comput. Graph. Forum* 38, 2 (2019), 193–205. https://doi.org/10.1111/cgf.13630

[25] Nima Khademi Kalantari, Eli Shechtman, Connelly Barnes, Soheil Darabi, Dan B. Goldman, and Pradeep Sen. 2013. Patch-Based High Dynamic Range Video. *ACM Trans. Graph.* 32, 6, Article 202 (nov 2013), 8 pages. https://doi.org/10.1145/2508363.2508402

[26] Soo Ye Kim, Jihyong Oh, and Munchurl Kim. 2019. Deep sr-itm: Joint learning of super-resolution and inverse tone-mapping for 4k uhd hdr applications. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. IEEE, 3116–3125.

[27] Soo Ye Kim, Jihyong Oh, and Munchurl Kim. 2020. Jsi-gan: Gan-based joint super-resolution and inverse tone-mapping with pixel-wise task-specific filters for uhd hdr video. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. AAAI Press, 11287–11295. Issue 7.

[28] Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*.

[29] Rafael Pacheco Kovaleski and Manuel M. Oliveira. 2014. High-Quality Reverse Tone Mapping for a Wide Range of Exposures. In *27th SIBGRAPI Conference on Graphics, Patterns and Images*. IEEE Computer Society, New York, 49–56.

[30] Hayden Landis. 2002. Production-Ready Global Illumination. In *SIGGRAPH Course Notes* 16. ACM, New York, 87–101.

[31] Bruno Lecouat, Thomas Eboli, Jean Ponce, and Julien Mairal. 2022. High Dynamic Range and Super-Resolution from Raw Image Bursts. *ACM Trans. Graph.* 41, 4, Article 38 (jul 2022), 21 pages. https://doi.org/10.1145/3528223.3530180

[32] Chen-Yu Lee, Patrick W. Gallagher, and Zhuowen Tu. 2018. Generalizing Pooling Functions in CNNs: Mixed, Gated, and Tree. *IEEE Trans. Pattern Anal. Mach. Intell.* 40, 4 (2018), 863–875. https://doi.org/10.1109/TPAMI.2017.2703082

[33] Siyeong Lee, Gwon Hwan An, and Suk-Ju Kang. 2018. Deep recursive hdri: Inverse tone mapping using generative adversarial networks. In *proceedings of the European Conference on Computer Vision (ECCV)*. 596–611.

[34] Siyeong Lee, So Yeon Jo, Gwon Hwan An, and Suk-Ju Kang. 2020. Learning to generate multi-exposure stacks with cycle consistency for high dynamic range imaging. *IEEE Transactions on Multimedia* 23 (2020), 2561–2574.

[35] Han Li and Pieter Peers. 2017. CRF-Net: Single Image Radiometric Calibration Using CNNs. In *Proceedings of the 14th European Conference on Visual Media Production (CVMP 2017)* (London, United Kingdom) *(CVMP 2017)*. Association for Computing Machinery, New York, NY, USA, Article 5, 9 pages. https://doi.org/10.1145/3150165.3150170

[36] Stephen Lin, Jinwei Gu, Shuntaro Yamazaki, and Heung-Yeung Shum. 2004. Radiometric Calibration from a Single Image. In *CVPR 2004: Proceedings of the 2004 IEEE Conference on Computer Vision and Pattern Recognition (CVPR2004)*. IEEE Computer Society, Washington, DC, USA, 938–945.

[37] Yu-Lun Liu, Wei-Sheng Lai, Yu-Sheng Chen, Yi-Lung Kao, Ming-Hsuan Yang, Yung-Yu Chuang, and Jia-Bin Huang. 2020. Single-image hdr reconstruction by learning to reverse the camera pipeline. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, 1651–1660.

[38] Yu-Lun Liu, Wei-Sheng Lai, Yu-Sheng Chen, Yi-Lung Kao, Ming-Hsuan Yang, Yung-Yu Chuang, and Jia-Bin Huang. 2020. Single-Image HDR Reconstruction by Learning to Reverse the Camera Pipeline. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 1648–1657.

[39] Rafał K. Mantiuk and Maryam Azimi. 2021. PU21: A novel perceptually uniform encoding for adapting existing quality metrics for HDR. In *Picture Coding Symposium, PCS 2021, Bristol, United Kingdom, June 29 - July 2, 2021*. IEEE, 1–5. https://doi.org/10.1109/PCS50896.2021.9477471

[40] Rafal K. Mantiuk, Dounia Hammou, and Param Hanji. 2023. HDR-VDP-3: A multi-metric for predicting image differences, quality and contrast distortions in high dynamic range and regular content. arXiv:2304.13625 [eess.IV]

[41] Demetris Marnerides, Thomas Bashford-Rogers, Jonathan Hatchett, and Kurt Debattista. 2018. ExpandNet: A Deep Convolutional Neural Network for High Dynamic Range Expansion from Low Dynamic Range Content. *Comput. Graph. Forum* 37, 2 (2018), 37–49. https://doi.org/10.1111/cgf.13340

[42] Belen Masia, Sandra Agustin, Roland W. Fleming, Olga Sorkine, and Diego Gutierrez. 2009. Evaluation of Reverse Tone Mapping Through Varying Exposure Conditions. *ACM Trans. Graph.* 28, 5 (2009), 1–8. https://doi.org/10.1145/1618452.1618506

[43] Belen Masia, Ana Serrano, and Diego Gutierrez. 2017. Dynamic range expansion based on image statistics. *Multimedia Tools and Applications* 76, 1 (2017), 631–648. https://doi.org/10.1007/s11042-015-3036-0

[44] Christopher A. Metzler, Hayato Ikoma, Yifan Peng, and Gordon Wetzstein. 2020. Deep Optics for Single-Shot High-Dynamic-Range Imaging. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*. Computer Vision Foundation / IEEE, 1372–1382. https://doi.org/10.1109/CVPR42600.2020.00145

[45] Laurence Meylan and Sabine Süsstrunk. 2006. High Dynamic Range Image Rendering with a Retinex-Based Adaptive Filter. *IEEE Transactions on Image Processing* 15, 9 (2006), 2820–2830.

[46] Venkatanath N., Praneeth D., Maruthi Chandrasekhar Bh., Sumohana S. Channappayya, and Swarup S. Medasani. 2015. Blind image quality evaluation using perception based features. In *Twenty First National Conference on Communications, NCC 2015, Mumbai, India, February 27 - March 1, 2015*. IEEE, 1–6. https://doi.org/10.1109/NCC.2015.7084843

[47] Manish Narwaria, Rafał K. Mantiuk, Mattheiu Perreira Da Silva, and Patrick Le Callet. 2015. HDR-VDP-2.2: A calibrated method for objective quality prediction of high dynamic range and standard images. *Journal of Electronic Imaging* 24, 1 (2015), 1050.1–1050.3.

[48] Eduardo Pérez-Pellitero, Sibi Catley-Chandar, Ales Leonardis, and Radu Timofte. 2021. NTIRE 2021 Challenge on High Dynamic Range Imaging: Dataset, Methods and Results. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2021, virtual, June 19-25, 2021*. Computer Vision

Foundation / IEEE, 691–700.

[49] 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 (2007), 39. https://doi.org/10.1145/1276377.1276426

[50] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015 - 18th International Conference Munich, Germany, October 5 - 9, 2015, Proceedings, Part III (Lecture Notes in Computer Science, Vol. 9351)*, Nassir Navab, Joachim Hornegger, William M. Wells III, and Alejandro F. Frangi (Eds.). Springer, 234–241. https://doi.org/10.1007/978-3-319-24574-4_28

[51] Marcel Santana Santos, Tsang Ing Ren, and Nima Khademi Kalantari. 2020. Single Image HDR Reconstruction Using a CNN with Masked Features and Perceptual Loss. *ACM Trans. Graph.* 39, 4, Article 80 (2020), 10 pages. https://doi.org/10.1145/3386569.3392403

[52] Helge Seetzen, Greg Ward, Lorne Whitehead, and Wolfgang Heidrich. 2004. High Dynamic Range Display System. In *ACM SIGGRAPH 2004 Emerging Technologies* (Los Angeles, California) *(SIGGRAPH '04)*. Association for Computing Machinery, New York, NY, USA, 8. https://doi.org/10.1145/1186155.1186164

[53] Aashish Sharma, Robby T. Tan, and Loong-Fah Cheong. 2020. Single-Image Camera Response Function Using Prediction Consistency and Gradual Refinement. In *Computer Vision - ACCV 2020 - 15th Asian Conference on Computer Vision, Kyoto, Japan, November 30 - December 4, 2020, Revised Selected Papers, Part VI (Lecture Notes in Computer Science, Vol. 12627)*, Hiroshi Ishikawa, Cheng-Lin Liu, Tomás Pajdla, and Jianbo Shi (Eds.). Springer, 19–35. https://doi.org/10.1007/978-3-030-69544-6_2

[54] Assaf Shocher, Nadav Cohen, and Michal Irani. 2018. "Zero-Shot" Super-Resolution Using Deep Internal Learning. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*. IEEE Computer Society, 3118–3126. https://doi.org/10.1109/CVPR.2018.00329

[55] Qilin Sun, Ethan Tseng, Qiang Fu, Wolfgang Heidrich, and Felix Heide. 2020. Learning Rank-1 Diffractive Optics for Single-Shot High Dynamic Range Imaging. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*. Computer Vision Foundation / IEEE, 1383–1393. https://doi.org/10.1109/CVPR42600.2020.00146

[56] Dmitry Ulyanov, Andrea Vedaldi, and Victor S. Lempitsky. 2020. Deep Image Prior. *Int. J. Comput. Vis.* 128, 7 (2020), 1867–1888. https://doi.org/10.1007/s11263-020-01303-4

[57] Tu Van Vo and Chul Lee. 2020. High Dynamic Range Video Synthesis Using Superpixel-Based Illuminance-Invariant Motion Estimation. *IEEE Access* 8 (2020), 24576–24587.

[58] Chao Wang, Ana Serrano, Xingang Pan, Bin Chen, Hans-Peter Seidel, Christian Theobalt, Karol Myszkowski, and Thomas Leimkuehler. 2022. Glow-GAN: Unsupervised Learning of HDR Images from LDR Images in the Wild. arXiv:2211.12352 [cs.CV]

[59] Lvdi Wang, Li-Yi Wei, Kun Zhou, Baining Guo, and Heung-Yeung Shum. 2007. High Dynamic Range Image Hallucination. In *SIGGRAPH '07: ACM SIGGRAPH 2007 Sketches* (San Diego, California). ACM, New York, NY, USA, 72. https://doi.org/10.1145/1278780.1278867

[60] Zhou Wang, Eero P. Simoncelli, and Alan C. Bovik. 2003. Multi-scale Structural Similarity for Image Quality Assessment. In *37th IEEE Asilomar Conference on Signals, Systems and Computers* (San Diego, California). IEEE, New York, NY, USA, 1398–1402. https://doi.org/10.1109/ACSSC.2003.1292216

[61] Xin Yang, Ke Xu, Yibing Song, Qiang Zhang, Xiaopeng Wei, and Rynson WH Lau. 2018. Image correction via deep reciprocating HDR transformation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 1798–1807.

[62] Hanning Yu, Wentao Liu, Chengjiang Long, Bo Dong, Qin Zou, and Chunxia Xiao. 2021. Luminance Attentive Networks for HDR Image and Panorama Reconstruction. *Computer Graphics Forum* 40, 7 (2021), 181–192.

[63] Yang Zhang and Tunç Ozan Aydin. 2021. Deep HDR estimation with generative detail reconstruction. *Comput. Graph. Forum* 40, 2 (2021), 179–190. https://doi.org/10.1111/cgf.142624

[64] Maria Zontak and Michal Irani. 2011. Internal statistics of a single natural image. In *CVPR 2011*. IEEE, 977–984.