DeepDuoHDR: A Low Complexity Two Exposure Algorithm for HDR Deghosting on Mobile Devices

Kadir Cenk Alpay, Ahmet Oğuz Akyüz, Nicola Brandonisio, Joseph Meehan, and Alan Chalmers

***Abstract*— The increased interest in consumer-grade high dynamic range (HDR) images and videos in recent years has caused a proliferation of HDR deghosting algorithms. Despite numerous proposals, a fast, memory-efficient, and robust algorithm has been difficult to achieve. This paper addresses this problem by leveraging the power of attention and U-Net-based neural representations and using a conservative deghosting strategy. Given two bracketed exposures of a scene, we produce an HDR image that maximally resembles the high exposure where it is well-exposed and fuses aligned information from both exposures otherwise. We evaluate the performance of our algorithm under several different challenging scenarios, using both visual and quantitative results, and show that it matches the state-of-the-art algorithms despite using only two exposures and having significantly lower computational complexity. Furthermore, the parameters of our algorithm greatly simplify deploying its different versions for devices with a variety of computational constraints, including mobile devices.**

***Index Terms*— HDR imaging, deghosting, deep networks, mobile devices.**

I. Introduction

HDR imaging aims to capture a broader dynamic range of the luminance of a scene, enabling more detail to be included in the final image by reducing the number of over/under-saturated pixels. An HDR image/video of a scene is typically produced by merging differently-exposed frames of that scene taken by a low dynamic range (LDR) camera in bracketing mode [1], [2]. However, ghosting artifacts may occur in the merged result if there exists a global camera or local object motion in the scene during the capture process [3].

Addressing the ghosting problem has received much attention in the HDR literature in recent years. On the one hand, traditional algorithms have dealt with this problem by trying to discover pixel correspondences ensued by a warping step [3], [4]. More recent algorithms, on the other hand, heavily relied on deep learning to directly reconstruct an HDR image from the input exposures [5], [6]. The most successful ones amongst these algorithms are generally the most computationally intensive ones, hindering their use for computationally limited applications.

In this study, we approach the deghosting problem with three important design decisions: *(i)* deghosting should be done only when needed, *(ii)* a learning-based solution should be employed only for the most difficult task, leaving room for a more efficient and traditional approach for the simpler steps, and *(iii)* two exposures should be used instead of three or more to further reduce computational complexity while still maintaining HDR quality. As shown in the rest of this paper, these decisions lead to a more lightweight network design and a significantly lower overall processing cost, which we consider as the main novelty of our algorithm. These design choices can become particularly beneficial in hardware-constrained environments such as mobile devices, where power consumption is also critically important for thermal management.

A sample result of our algorithm, together with a description of its overall workflow, is shown in Fig. 1. This video frame taken from the Stuttgart Wide-Gamut HDR Video Dataset [7] represents a very challenging scenario due to the presence of fast-moving and blinking colorful lights and objects at night. The insets show the amount of motion in the input exposures and the reconstruction quality achieved in these regions. We provide a tone-mapped video of this scene in the supplementary materials. To summarize, our main contributions are:

- A hybrid deghosting and HDR image generation algorithm that combines classical and deep learning-based solutions for improved efficiency and modularity.
- A conservative approach which processes only the saturated parts of the reference exposure for providing additional computational benefits.
- A synthetic data method that enables the network to generalize to a new sensor for which no training data is available.

II. Related Work

A. HDR Image Reconstruction

HDR images are typically created by merging multiple exposures. Although there are several variations of this algorithm, its basic principle involves linearizing the input exposures and computing a weighted average of corresponding pixels to obtain an irradiance estimate of the captured

Received 19 September 2023; revised 2 June 2024 and 16 September 2024; accepted 4 November 2024. Date of publication 19 November 2024; date of current version 25 November 2024. The associate editor coordinating the review of this article and approving it for publication was Dr. Keigo Hirakawa. *(Corresponding author: Ahmet Oğuz Akyüz.)*

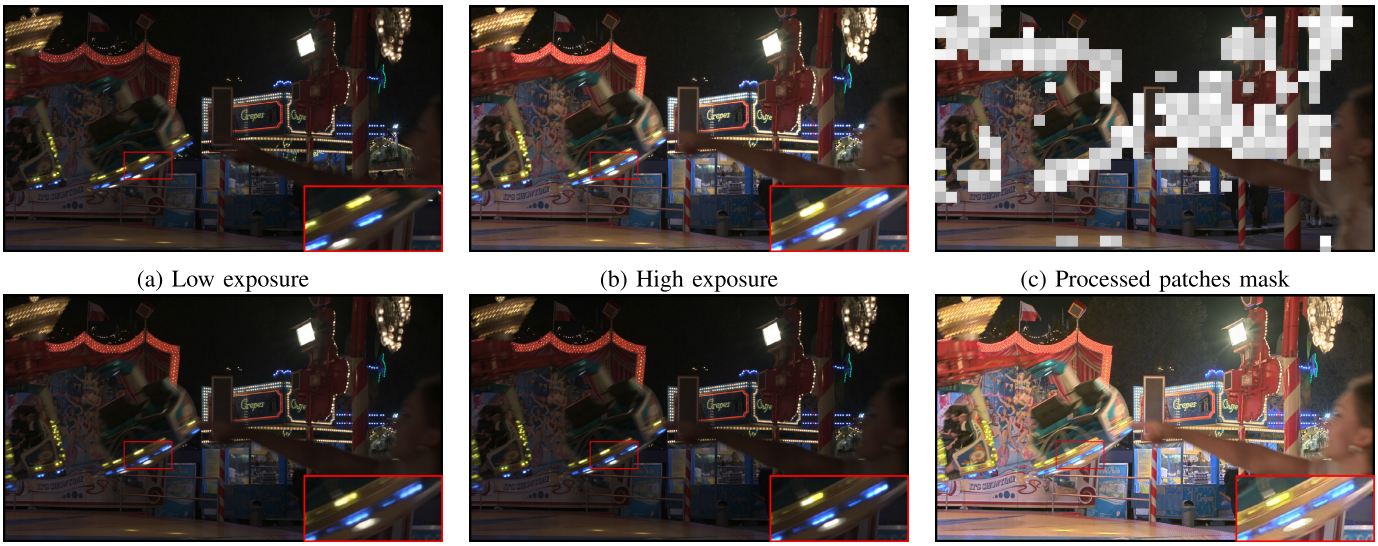
Kadir Cenk Alpay and Ahmet Oğuz Akyüz are with the Department of Computer Engineering, Middle East Technical University, 06800 Ankara, Türkiye (e-mail: akyuz@ceng.metu.edu.tr).

Nicola Brandonisio and Joseph Meehan are with Huawei Technologies, 06250 Mougins, France.

Alan Chalmers is with WMG, University of Warwick, CV4 7AL Coventry, U.K.

This article has supplementary downloadable material available at https://doi.org/10.1109/TIP.2024.3497838, provided by the authors.

Digital Object Identifier 10.1109/TIP.2024.3497838



(a) Low exposure (b) High exposure (c) Processed patches mask

(d) Our reconstructed low exposure (e) Ground-truth low exposure (f) Our tone-mapped HDR

Fig. 1. Our algorithm takes a pair of low (a) and high (b) exposure frames and computes a patch mask from the high exposure (c). The patches corresponding to over-saturated regions are reconstructed by a network, and the well-exposed regions are exposure-normalized (d). This produces a new low exposure that matches a ground-truth low exposure taken at the high's time instance (e). The reconstructed low can be merged with the high to obtain an HDR image (f), which is tone-mapped for display purposes. If desired, the HDR image can be directly reconstructed without creating a new low exposure.

scene [1], [8], [9]. The application of this algorithm requires precise alignment between the input exposures, which otherwise leads to ghosting artifacts. A myriad of HDR deghosting techniques have been proposed in the literature in the last two decades [3], [4], [5]. They all face the same intrinsic physically difficult cases, namely saturation, occlusion, parallax, and large global and local motions that can lead to ghosting artifacts. In some studies, these difficult cases are addressed directly with explicit and specialized methods, including specialized neural network modules or dedicated motion/occlusion/flow detection algorithms [10], [11]. On the other hand, in other research, ghosting artifacts are minimized implicitly within the proposed techniques: for example, during an energy minimization calculation, which may take into account both motion estimation and fusion phase [12].

B. Optimization-Based Algorithms

HDR deghosting studies that have been proposed up to 2018 are well summarized in several surveys [3], [4], [13]. The most successful algorithms of this era include both multi-scale and iterative energy minimization-based approaches [12], [14]. Optical-flow-based algorithms that can deal with different exposure levels were also proposed [15]. While some of these earlier algorithms expect the input images to be pre-aligned [16] by using multi-exposure image registration algorithms [17], [18], in some research, this limitation does not exist [12], [14]. Several studies were also proposed for video deghosting. They typically combine techniques commonly used in image deghosting such as gradient-based optical flow [19], block-based motion estimation [20], [21], patch-based processing [22] with jitter suppression to avoid flickering artifacts across frames. The primary bottleneck of these optimization-based algorithms is their high computational cost, making them applicable only for offline processing scenarios.

C. Learning-Based Algorithms

These algorithms can be classified into two broad groups, namely the classical deep learning-based approaches and more modern attention and transformer-based models. A comprehensive review of these methods can be found in a recent survey paper by Wang and Yoon [5]. Furthermore, the NTIRE 2022 challenge on HDR imaging reports the performance of these algorithms on benchmark datasets [6].

One of the pioneering works of this group of algorithms is Kalantari et al.'s deep learning-based HDR deghosting algorithm [10]. This study compares several deep learning-based architectures, such as direct estimation, weight estimation, and weight and image estimation approaches. The study also shares a benchmark dataset with training/test separation that is commonly used in follow-up research.

We provide a taxonomy of the most prominent recent publications in Table I. As can be seen in this table, the most recent works are based on deep convolutional neural networks (DCNNs) and employ attention architectures [48]. The use of attention architectures allows these networks to extract the most crucial information from the input exposures [39]. Most methods try to create an HDR image, but a few methods target exposure fusion [44], [46], [49]. Another differentiating factor between the algorithms is the number of input exposures. Most methods work with 3 exposures (low, medium, high), but some algorithms can process only two or a higher number of exposures. Some methods also support HDR video content.

A recent benchmark study [6] compared the results of these recent algorithms according to low-complexity and high-fidelity constraints using Froelich et al.'s dataset [7]. The best-performing method in both tasks used a teacher-student

TABLE I
A CLASSIFICATION OF RECENT HDR DEGHOSTING ALGORITHMS

Study	Year	Video	DCNN	GAN	Attention	Fusion/HDR	Exposures	Bayer	Hardware
Zhang et al. [23]	2024	No	Yes	No	Yes	HDR	3	No	No
Xiao et al. [24]	2024	No	Yes	No	Yes	HDR	3	No	No
Li et al. [25]	2024	No	Yes	No	No	HDR	3	No	No
Yan et al. [26]	2023	No	Yes	No	Yes	HDR	3	No	No
Yan et al. [27]	2023	No	Yes	No	Yes	HDR	3	No	No
Zhou et al. [28]	2023	No	Yes	No	Yes	HDR	3	No	No
Yan et al. [29]	2022	No	Yes	No	Yes	HDR	3	No	No[a]
Catley-Chandar et al. [30]	2022	No	Yes	No	Yes	HDR	N	No	No
Cogalan et al. [31]	2022	Yes	Yes	No	No	HDR	3	Yes	Yes[b]
Jiang et al. [32]	2021	Yes	Yes	No	Yes	HDR	3	Yes	Yes[c]
Liu et al. [33]	2021	No	Yes	No	Yes	HDR	3	No	No
Niu et al. [34]	2021	No	Yes	Yes	Yes	HDR	3	No	No
Anand et al. [35]	2021	Yes	Yes	Yes	No	HDR	2	No	No
Chen et al. [36]	2021	Yes	Yes	No	Yes	HDR	5	No	No
Prabhakar et al. [37]	2021	No	Yes	No	No	HDR	N	No	No
Prabhakar et al. [38]	2020	No	Yes	No	No	HDR	3	No	No
Yan et al. [39]	2020	No	Yes	No	Yes	HDR	3	No	No
Yan et al. [40]	2020	No	Yes	No	No	HDR	3	No	No
Vo et al. [41]	2020	Yes	No	No	No	HDR	2	No	No
Suda et al. [42]	2020	No	Yes	No	No	HDR	3	Yes	Yes[d]
Choi et al. [43]	2020	No	Yes	No	Yes	HDR	3	No	No
Kalantari and Ramamoorthi [11]	2019	Yes	Yes	No	No	HDR	2 or 3	No	No
Prabhakar et al. [44]	2019	No	Yes	No	No	Fusion	N	No	No
Yan et al. [45]	2019	No	Yes	No	Yes	HDR	3	No	No
Martorell et al. [46]	2019	No	No	No	No	Fusion	3-7	No	No
Wu et al. [47]	2018	No	Yes	No	No	HDR	3 or 5	No	No
Kalantari et al. [10]	2017	No	Yes	No	No	HDR	3	No	No

[a] Simple alternating exposures. [b] Camera that can vary the exposure time in every column.
[c] Tri-exposure quad Bayer sensor. [d] Different neutral density filters in front of different pixel blocks.

knowledge distillation strategy [50]. A common trait of well-performing methods was to use residual feature distillation blocks (RFDB) [51] and dilated residual dense blocks (DRDB) [45]. It was also found that a large number of the algorithms used a U-Net-style network architecture due to its efficiency and performance benefits. It should be noted that most of the benchmarked studies were proprietary where only high-level descriptions of the algorithms were available.

D. Single Image HDR

As an alternative to HDR image deghosting, single image HDR reconstruction methods are also proposed. These methods aim to capture the luminance distribution of the scene in a single time instance, thereby circumventing the ghosting problem. Earlier designs involve hardware modifications such as split aperture imaging where the input light beam is split and redirected into different sensors [52], [53] as well as per-pixel exposure variation using optical filters with varying transmittance [54], [55], [56]. Dual-ISO [57], [58], [59], [60] and dual-exposure [61], [62], [63], [64], [65], [66] methods that use learning-based deinterlacing algorithms to construct full resolution HDR images are also proposed.

Relatedly, inverse tone-mapping techniques take a single LDR image and produce an HDR image where missing details are hallucinated [67]. While earlier methods relied on inverting tone mapping operators [68] and non-linear brightness adjustments [69], recent studies focused on using neural networks to predict the details in saturated regions [70], [71], [72], [73].

Finally, burst photography techniques generate HDR images from a large number of LDR images taken in rapid succession. These techniques rely on reducing the noise in dark regions instead of recovering details in over-saturated ones [74], [75].

III. APPROACH

A. Rethinking Deghosting

In our analysis of existing algorithms, we found that *(i)* most of the recent methods aim to perform deghosting on the entire image, *(ii)* they aim to reconstruct the final HDR image in an end-to-end manner, and *(iii)* that most algorithms require at least three input exposures, and having a well-exposed reference image (the medium exposure) is important for them. While these features can be desirable for some applications, they also bring about extra challenges. As for *(i)*, processing the entire image incurs a significant performance cost while at the same time risking the appearance of artifacts in regions that do not need deghosting. For *(ii)*, learning to produce an HDR image poses a more difficult learning task and necessitates a more complex network compared to learning to produce an LDR image. The latter also allows using any desired HDR reconstruction algorithm instead of the one embedded in the network. Finally, for *(iii)*, using more than two images may be impractical for performance-critical applications such as real-time HDR image/video reconstruction in mobile devices. As a result, our algorithm is designed around the following design decisions to circumvent these problems:

- Only use two exposures, namely the low and the high exposure. Any of these can be set as the reference, although we use the high exposure in this paper.

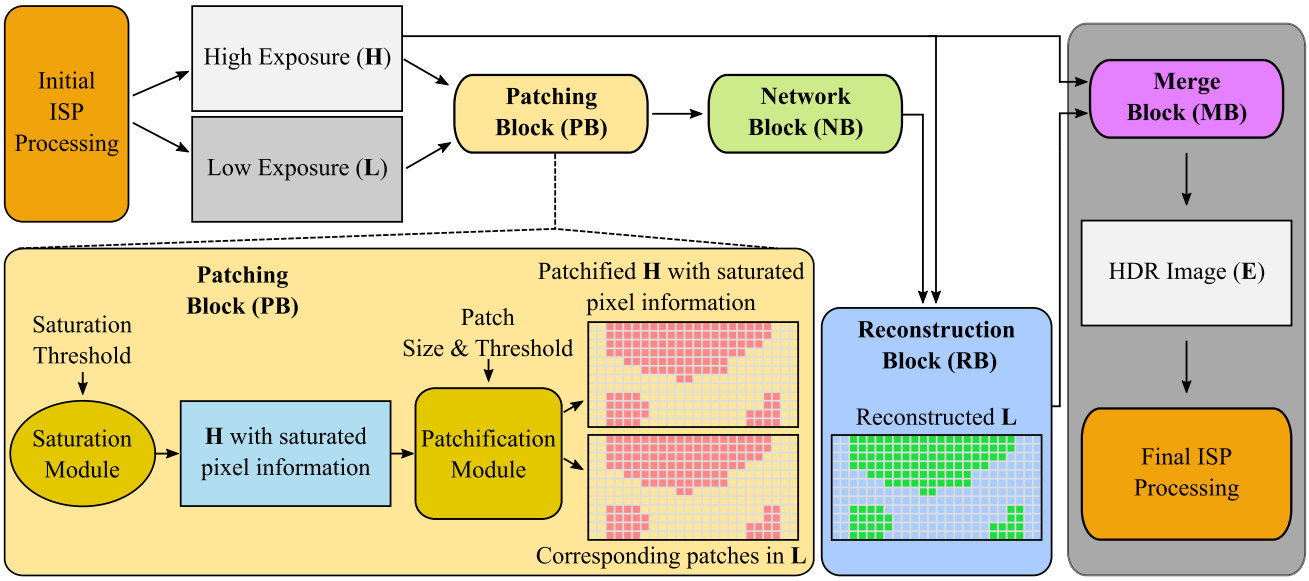


Fig. 2. Overview of the proposed deghosting framework. The inputs to our pipeline are two exposures produced by an initial ISP processing stage. The high exposure is then analyzed to find its saturated patches, which are paired with the corresponding low exposure patches (patching block). These paired high and low patches are then processed by our previously trained network to produce new low patches that are aligned with the content in high patches (network block). Then a new low exposure is reconstructed keeping the high's non-saturated patches and using the newly produced patches from the network when saturated (reconstruction block). The original high exposure and the reconstructed low exposure are then merged to create the final HDR image (merge block). The final HDR image can undergo subsequent ISP processing such as tone-mapping.

- Only process parts of the reference exposure that lack details. For a high exposure reference, this corresponds to over-saturated regions.
- As output, produce a new deghosted exposure instead of the final HDR image. The HDR image can be reconstructed using any off-the-shelf HDR merge [1] or exposure fusion [49] algorithm, not only simplifying the task of the learning algorithm but also making the results suitable for further ISP processing. In other words, our algorithm can run as an early step in the ISP pipeline, and the rest of the pipeline, including HDR image generation, can proceed as if the exposures are captured without any ghosting problems.

All of these design decisions allow for a more efficient deghosting algorithm compared to state-of-the-art deep learning-based ones and, as shown in the results section, without compromising image quality. The overall pipeline of our framework that implements these design decisions is depicted in Fig. 2. The following sections explain the details of each stage.

B. Initial ISP Processing

We assume that the processing begins with the camera's internal image signal processor (ISP). This stage is responsible for reading digital sensor values in Bayer domain and converting them to RGB low and high exposures. The ISP processing includes demosaicking, white-balancing, color correction, and in some cases denoising and sharpening of the sensor values. Each ISP may perform different operations, and the specific details are not critical to our algorithm as long as they output two bracketed RGB images with known exposure values.

C. Patching Block

Our algorithm begins with the patching block in which we identify the patches whose details need to be recovered. Assuming that the high exposure is the reference, we first identify its over-saturated regions that need to be reconstructed from the low exposure. To this end, we apply a saturation test to the high exposure to detect its over-exposed pixels. This test compares each pixel's RGB values with a *saturation threshold* S_t such that the pixels with any color component greater than this threshold are marked as saturated.

In the next step, both high and low exposures are patchified using the value of the *patch size* parameter, P_s, in both horizontal and vertical axes. As both exposures are of the same resolution, each patch of each exposure corresponds to the identically located patch in the other exposure. Then, the patch selection step is conducted on the patches of the high exposure. The patches that include more than *saturated count threshold*, C_s, number of pixels are selected together with their corresponding patches from the low exposure. A pair of high and low patches selected in this manner constitute the inputs to the following learning stage. In this paper, we use the following values for the parameters defined above: $S_t = 0.97$ for pixel values in [0, 1] range, $P_s = 64 \times 64$, and $C_s = 1$.

D. Network Block

The purpose of the network block (Fig. 3) is to reconstruct details missing in a high exposure patch using information from both high and low exposure patches. Specifically, given two input patches as h and l, the network block predicts $\hat{l}$ which resembles a low exposure captured at high's time instance.

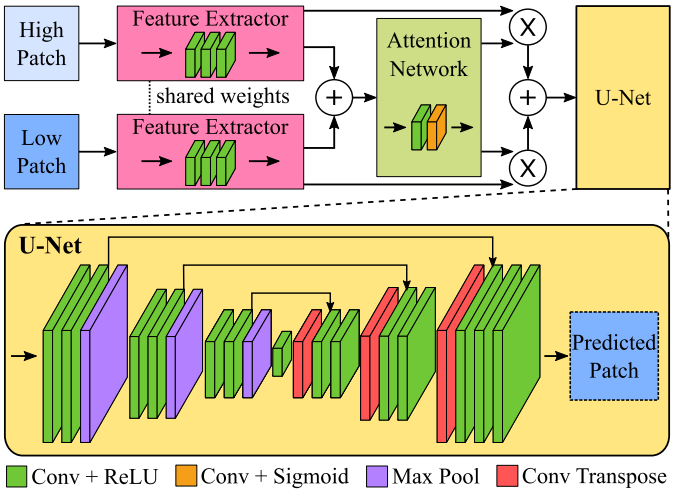
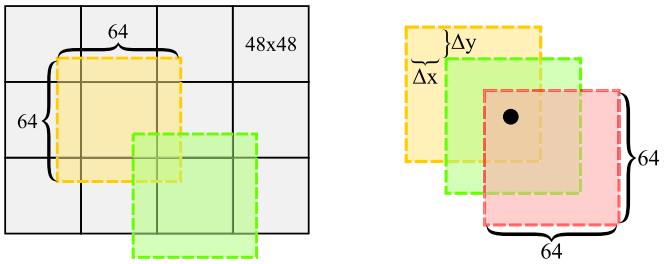


Fig. 3. Our network architecture is a combination of feature extraction and attention blocks followed by a U-Net model with skip connections. Plus and cross symbols represent channel-wise concatenation and Hadamard product, respectively.



(a) Overlapped patches (b) Shifted patches

Fig. 4. Two methods to eliminate patch artifacts: (a) the image is divided into 48×48 blocks, and processed patches are centered around these blocks. (b) processed patches are shifted by Δx and Δy units yielding multiple estimates for a pixel. We found the strategy in (a) to work better.

1) Exposure Normalization: After experimenting with several alternatives, we found that the network learns better if it is fed with a pair of exposures that are normalized to the same exposure value (EV) range. Assuming that initially both h and l are in linear $[0, 1]$ domain, we compute an exposure normalized low patch as follows:

$$l_{l \to h} = l 2^{EV_h - EV_l}. \tag{1}$$

The network then receives h and $l_{l \to h}$ as input and produces an estimate of the low exposure normalized to the high's exposure level:

$$\hat{l} = g(h, l_{l \to h}), \tag{2}$$

If desired, the final low patch may be obtained by bringing the network output back to the original EV range of the low exposure:

$$\hat{l}_{\text{low}} = \hat{l} 2^{EV_l - EV_h}. \tag{3}$$

Alternatively, the HDR result can be obtained by directly merging h and $\hat{l}$.

2) Patch Processing: Patching artifacts may occur at the border between two predicted patches, or between a predicted and a simply high-to-low normalized patch. To eliminate this problem, we experimented with two approaches shown in Fig. 4. In the first approach, shown in Fig. 4a, the underlying image is logically divided into blocks of 48×48 pixels. The 64×64 patches processed by our algorithm are centered around these blocks. As a result, each processed patch gets reconstructed by being aware of the content of neighboring patches. After processing, only the center 48×48 region is cropped and placed into its corresponding location in the final image. In our implementation, this process is executed in parallel for multiple patches.

An alternative solution to eliminate tiling artifacts would be to shift the patches by Δx and Δy units, creating multiple estimates for a single pixel – one for each patch. For example, the pixel indicated by the black dot in Fig. 4b, would receive 3 estimates. These are then averaged to find the final estimate. Amongst these two alternatives, we found the first solution to be more efficient and produced results without noticeable patch artifacts. The second solution also eliminates patch artifacts, but may introduce a slight blur due to the averaging process and is computationally more complex.

3) Network Details: As shown in Fig. 3, our network accepts a pair of low and high 3-channel (RGB) exposure patches of size 64×64 as input. A feature extraction block with shared weights is employed as the first step to produce features for each patch. This block is composed of three convolutional layers with a ReLU activation function at the end of each layer. The aim of this block is to extract expressive feature representations from the pixels of the input patch pair.

The attention block takes the channel-wise concatenated features produced at the end of the feature extractor as input and produces an attention map. The attention block is composed of two convolutional layers. The first one is followed by a ReLU activation function, and the second one is followed by a Sigmoid activation function to bound the output values of the attention map to $[0, 1]$ range. The features of both high and low exposures are then multiplied by the attention map to suppress the unimportant details and keep the important ones unchanged. The scaled features are then concatenated channel-wise and fed to our U-Net block.

All convolutional layers of the feature extractor and the attention block use 64 filters with kernel size 3×3. We employ zero-padding to keep the spatial resolution of the features unchanged so that the spatial resolution requirement of 64×64 of the first layer of the U-Net block is ensured.

The U-Net block's details are visualized in the bottom half of Fig. 3. Each encoding level of the encoder part employs a stack comprised of two convolutional layers with ReLU activations and a max pooling layer that halves each spatial dimension. The decoder part leverages transposed convolutions to double the spatial dimensions. Each decoding level employs two convolutional layers with ReLU activations right after the transposed convolution layer except the top level where three are employed. We employ skip connections between the corresponding levels of the encoder and decoder parts by channel-wise concatenating the encoded features with the outputs of the transposed convolutions. This enables us to

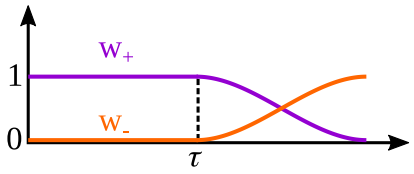


Fig. 5. Weighting functions used for blending the high and reconstructed low patches. τ parameter defines the threshold above which low exposure contributes. Note that there are two weighting functions, but the input to both functions is the high exposure (see (4)).

prevent important details from being lost during the encoding process.

The convolutional layers of the first encoding level have 16 filters. The number of filters is doubled as the spatial dimensions are reduced and halved at each decoding level. The last convolutional layer of the decoder part contains 3 filters to produce the predicted 3-channel (RGB) patch at the spatial resolution of 64×64.

E. Reconstruction and Merge Blocks

Our algorithm can produce a new low exposure that is content-wise aligned to the high exposure and exposure-wise aligned to the original low exposure. For this purpose, unsaturated patches of the high exposure can simply be scaled down to the low exposure's EV and the saturated patches can be replaced by the network output. This way, any desired HDR reconstruction algorithm can be used providing modularity. We considered the decoupling of the deghosting process from HDR assembly as an important feature of our algorithm.

Alternatively, we can directly reconstruct an HDR image from the high exposure and the network output in the following manner. We use a reconstruction function that computes a weighted blending of the high patch and the reconstructed patch using a custom weighting function for patches that are processed by our network. Otherwise, we directly use the high exposure's pixel value as it was found to be not saturated:

$$E(x,y) = \begin{cases} \dfrac{w_+(h(x,y))h(x,y) + w_-(h(x,y))\hat{l}(x,y)}{w_+(h(x,y)) + w_-(h(x,y))} & \text{if } (x,y) \in \Omega, \\ h(x,y) & \text{otherwise,} \end{cases} \tag{4}$$

where Ω represents the set of pixels that are processed by our network. The w_+ and w_- weighting functions control the degree of blending between pixels. They are defined as Hermite-like blending functions (Fig. 5):

$$w_+(x) = \begin{cases} 1 & \text{if } x < \tau, \\ 1 + 2\left[\dfrac{x-\tau}{1-\tau}\right]^3 - 3\left[\dfrac{x-\tau}{1-\tau}\right]^2 & \text{if } x \geq \tau, \end{cases} \tag{5}$$

$$w_-(x) = \begin{cases} 0 & \text{if } x < \tau, \\ -2\left[\dfrac{x-\tau}{1-\tau}\right]^3 + 3\left[\dfrac{x-\tau}{1-\tau}\right]^2 & \text{if } x \geq \tau. \end{cases} \tag{6}$$

τ represents the threshold value beyond which we allow the contribution of pixels from the reconstructed low exposure patch, and x is the luminance of the pixel instead of its individual color channels to prevent color cast [76]. In our implementation, we set $\tau = 0.5$ for pixel values in [0, 1] range.

IV. Dataset Generation

A. Stuttgart Dataset

Stuttgart HDR video dataset contains ghost-free HDR frames captured for 33 different scene video cuts using a beam-splitter-based two-camera capture rig [7]. Due to the size, diversity, and photographic quality of this dataset, it serves as a convenient starting point to train and test our network model. However, because each frame is a ghost-free HDR image, the ghosting phenomenon needs to be simulated. For this purpose, we select pairs of contiguous HDR frames from each scene (e.g., frame E_i and E_{i+1}). The first image in each pair is considered as the low exposure and the second one the high exposure. The high exposure's value, EV_h, is selected as both 0 and 2 EV to simulate a well and a highly exposed high exposure. The corresponding high exposure is found by:

$$H_{i+1} = 255\left[2^{EV_h}\frac{0.217E_{i+1}}{\overline{E}_{i+1}}\right]^{1/\gamma}_{\text{clip}(0,1)}, \tag{7}$$

where $\overline{E}$ indicates the mean value and the factor 0.217 ensures that when EV_h is 0 the mean value of the gamma corrected high exposure, H, will be 127.5. We assume $\gamma = 2.2$ and clip the result to [0, 1] range prior to scaling it with 255.

The corresponding low exposures, L_i, are created in the same manner by taking EV_l from $\{EV_h - 1, EV_h - 2, EV_h - 3, EV_h - 4\}$ and using the previous HDR frame, E_i. Finally, for each low exposure, we create a *ground-truth* low exposure, L_{i+1}, by using frame E_{i+1} and EV_l. The ground-truth lows are perfectly aligned to the high exposure (due to being created from the same HDR image), but their exposure values match the low exposures generated from the previous frame. The triplet (H_{i+1}, L_i, L_{i+1}) thus constitutes a training sample for our network. In practice, we do not use all of the pixels from these images but only take the patches that meet the saturation criteria in the high exposure as explained in Section III-C.

B. HDR Photographic Survey Dataset

HDR photographic survey contains a rich collection of 106 high quality HDR images created from multi-bracketed input exposures captured by a Nikon D2X camera [77]. This dataset well complements the Stuttgart dataset as it contains many outdoor landscape photographs, a characteristic missing in the Stuttgart dataset. Thus, we use the survey dataset for improving the generalization performance of our network to a diverse set of scene conditions. However, because the survey dataset is comprised of static images, we employ a warping strategy to simulate motion.

For this purpose, we create single-octave Perlin noise fields using the improved Perlin noise algorithm [78], because it can lead to controllable and continuous random warped images. Two noise fields are created, one for each of x- and y-axis.

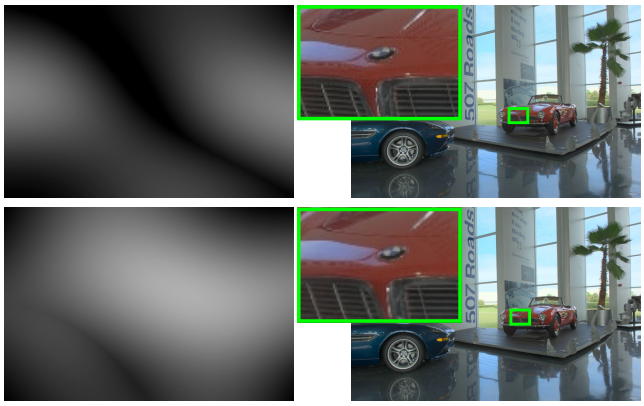


Fig. 6. Left: x and y warp fields. Right: Original and warped images.

Assuming that n represents the noise value in $[-1, 1]$ range, we use the following mapping to compute the final noise value:

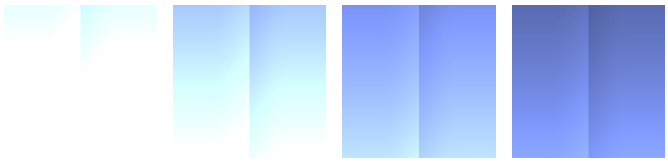
$$n' = s((2|n| - 1) + n). \tag{8}$$

The parameter s controls the desired amount of warping. We set $s = 8$, to achieve a total of ± 16 pixels shifts in either direction. Perlin noise maps are created in 256×256 resolution and scaled to the resolution of the input image, which is approximately full-HD in our case (we resize all survey images to 1920 pixels wide by preserving the aspect ratio). Different pairs of noise maps are created for each input image. A sample result is shown in Fig. 6, where the warp fields for the x- and y-axis are shown on the left. In this illustration, mid-gray tones denote little to no warping, whereas black and white tones denote negative and positive shifts, respectively. Using these warp fields, the original image shown in the top-right is remapped to the final image shown in the bottom-right. The insets show that pixels are warped while the objects maintain their overall integrity. Following the warping process, which is performed in the HDR domain, we generate high, low, and ground-truth low exposures using the same strategy explained in the previous section.

C. Synthetic Dataset

In order to augment the learning ability of the network to cope with fully saturated high patches while being able to produce smoothly varying outputs, we generated synthetic gradient images of various colors. To create these patterns, we generated an HDR image based on the Cornsweet profile [79]:

$$\begin{aligned}
G(y) &= 0.70\frac{1 - e^{y}}{1 - e}, \\
H(x) &= 0.176\frac{1 - e^{-x^2}}{1 - e^{-1}}, \\
K(y) &= 0.5 + G(y), \\
R(x, y) &= \begin{cases} H(4\,(x - 0.25))\;K(y) & x \in [0.25, 0.50), \\ -H(4\,(0.75 - x))\;K(y) & x \in [0.50, 0.75), \\ 0 & \text{otherwise}, \end{cases} \\
L(x, y) &= [10^{-1} + 10^{6}\,(G(y) + R(x, y) + 0.414)]\mathbf{C}, \qquad (9)
\end{aligned}$$



(a) Low: -1 EV (b) Low: -2 EV (c) Low: -3 EV (d) Low: -4 EV

Fig. 7. Synthetically generated exposures.

where (x, y) are the normalized pixel coordinates and $\mathbf{C}$ represents the desired base color of the profile. The output of this profile is an HDR image, which can be scaled to produce high and low exposures similar to real HDR images. We choose the scaling factors such that the high exposure is fully saturated while the low exposures contain varying degrees of details. An example set of 4 low exposures is shown in Fig. 7. We generate 1000 such patterns by varying the proportion of R, G, and B components between 10% and 100%, independently for each color channel.

V. Experimental Evaluation

We evaluate the performance of our algorithm under several test conditions corresponding to different application scenarios. First, we show our results for the Stuttgart and the HDR photographic survey datasets, for which we have ground-truth frames and therefore can report PSNR results. We then present the results for the smartphone dataset, in which the HDR frames are captured using the DOL-HDR functionality of the underlying sensor [80]. For the final scenario, we train and test our network on the Kalantari dataset [10] as this dataset is heavily used by the state-of-the-art studies. Here, we report our PSNR, HDR-VDP [81], and unified deghosting quality metric (UDQM) [82] results in addition to providing visual comparisons with the literature works.

A. Training Details

The ground-truth patches, l_{gt}, are kept as exposure-aligned to the low patches. We apply (3) to the network's predicted RGB patch during training to bring its exposure to the exposure of the low patch. This also enables the predicted pixel values to be in $[0, 1]$ range, allowing for the use of the differentiable μ-law tone-mapping prior to computing the loss function [10]:

$$T(E) = \frac{\log(1 + \mu E)}{\log(1 + \mu)}, \tag{10}$$

where E and T indicate the HDR and tone-mapped pixel values, respectively, and μ is a parameter. Following common practice, we set $\mu = 5000$. This function, while not designed to produce visually appealing tone-mapping results, prevents the high irradiance pixels from dominating the network loss. We apply (10) to both the network output and the ground-truth patch and then calculate the element-wise l_1 loss:

$$\mathcal{L} = ||T(\hat{l}_{\text{low}}) - T(l_{\text{gt}})||_1. \tag{11}$$

We implemented our network in PyTorch using the Lightning library [83]. We use a batch size of 512 composed of

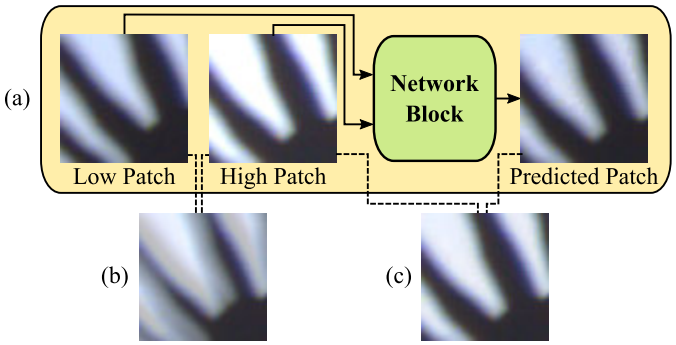


Fig. 8. (a) The inference process of our network. Low and high input patches are given as input and the reconstructed low patch is produced. (b) A simple blend of the original high and low patches demonstrates noticeable ghosting artifacts. (c) Blending the original high patch with the reconstructed low patch of our network output eliminates visible ghosting.

mixed EV-difference exposure pairs along with the ground-truths (low, high, low_{gt}) and train our network in parallel on 8 NVIDIA A100 80GB GPUs for up to 1000 epochs. We use ADAM optimizer with a learning rate of 5e-4 and keep the lowest validation loss checkpoint during the training. We employ random rotations of 90, 180, and 270 degrees, random horizontal and vertical flips, color channel swapping, and adding a small amount of Gaussian noise as data augmentation strategies.

B. Single Patch Analysis

To illustrate what our network learns, we begin by focusing on a single pair of low and high patches as input to our network block (Fig. 8). In the top row of this figure (a), the input pair of exposures (fingers of a person waving hand against a light background) and the output of our network are shown, demonstrating that our network learns to reconstruct a new low exposure as if it is taken at the high exposure's time instance. In (b), we show a simple blending of the input exposures to demonstrate the visibility of ghosting artifacts. In (c), we blend the original high and the reconstructed low patches. It can be seen that the ghosting artifacts are visually eliminated while the blue tint of the background is preserved. Note that in this example, we simply blend the exposures by assigning equal weights to each (rather than a true HDR merge) in order to solely illustrate the deghosting behavior of our network.

C. Scenario One: Stuttgart HDR Video Dataset

In our first scenario, we evaluate the performance of our deghosting algorithm on a challenging and cinematographic real-world HDR video dataset, known as the Stuttgart dataset [7]. To evaluate our network on this dataset, we selected 3 test and 27 training scenes. The training scenes themselves were split into 24 training and 3 validation scenes. Training data was prepared as explained in Section IV-A and saved in HDF5 files [84]. We performed training for 1000 epochs and chose the epoch that led to the lowest validation loss. During evaluation, we used an exposure difference of 2 EV (a factor of 4) between the high and low exposures. These exposures were generated from pairs of subsequent

TABLE II

AVERAGE PSNR (dB) SCORES OF OUR NETWORK ON 3 TEST SCENES OF STUTTGART HDR VIDEO DATASET. 30 FRAME PAIRS PER TEST SCENE THAT EVENLY COVER THE SCENE'S FRAME SEQUENCE ARE USED. PSNR SCORES ARE CALCULATED USING OUR PRODUCED LDR IMAGE AND THE GROUND-TRUTH LDR IMAGE. NONE OF THE TEST SCENES WERE SEEN BY THE NETWORK DURING TRAINING

Test Scenes	Average PSNR Per Scene	Mean Average PSNR
fishing_longshot,	49.257	
carousel_fireworks_03,	44.833	43.963
beerfest_lightshow_04	37.799	

TABLE III

AVERAGE PSNR SCORES ON THE 10 TEST SCENES OF HDR PHOTOGRAPHIC SURVEY DATASET BEFORE AND AFTER FINE-TUNING THE NETWORK ON SEPARATE TRAINING SCENES OF THAT DATASET

Test Scenes	Avg. PSNR Pre-finetune	Avg. PSNR Post-finetune
KingsCanyon, MammothHotSprings, PaulBunyan, BigfootPass, HallofFame, MiddlePond, OtterPoint, GoldenGate(1), SunsetPoint(2), TheNarrows(1)	37.826	40.242

frames: low/high from $(i-1)^{\text{th}}$ and i^{th} frames, respectively. We selected such 30 pairs of frames that are equally distributed across each scene (e.g., frames $(1, 2)$; frames $(31, 32)$; etc.). The PSNR results are summarized in Table II. Here it can be seen that, on average, we achieve PSNR rates that are well over $40dB$.

We show representative visual results for this evaluation in Fig. 9, where each row shows a different scene. In reading order, the columns show the low-exposure frame, high-exposure frame, the results of directly merging these frames (no deghosting), our processed patches mask, and our HDR-merge result. Closer inspection shows that our algorithm almost completely eliminates ghosting artifacts despite the challenging nature of these scenes. Especially the second and third scenes which contain flashing lights on fast-moving objects under low illumination pose among the most difficult challenges for deghosting algorithms. The processed patches masks indicate that we only process the regions of the high exposure frame that are saturated, avoiding unnecessary computation for well-exposed regions. The tone-mapped HDR videos that we created for these three scenes can be found in the electronic supplementary materials.

D. Scenario Two: HDR Photographic Survey Dataset

This dataset contains 106 static HDR images for which we simulate motion using Perlin noise as explained in Section IV-B. The characteristics of this dataset are markedly different from the Stuttgart dataset, as it contains many bright outdoor images compared to predominantly darker scenes of the latter. As such, we hypothesized that some amount of fine-tuning would be necessary to produce high quality results. However, first we wanted to find out the results with direct application of the best network from the Stuttgart dataset. To this end, we used 10 images for testing purposes.

As can be seen in the left column of Fig. 10, direct application of the Stuttgart network leads to some noticeable artifacts. Closer inspection shows that although deghosting is

(a) Low Exposure (b) High Exposure (c) No Deghosting (d) Processed Patches (e) Ours

Fig. 9. Stuttgart HDR video dataset results. Each row shows a different test scene.

(a) Pre-finetune (b) Post-finetune

Fig. 10. HDR photographic survey dataset images visualizing the pre-finetune and post-finetune difference.

successful, patch artifacts are introduced. Some halo artifacts around the tree branches can be seen in the second row of this figure. These problems can be expected as the Stuttgart dataset on which the network was trained does not contain scenes with similar characteristics.

To improve our results, we performed fine-tuning by setting the initial learning rate to 10^{-4} (compared to 5.10^{-4} of the base network) and reduced it by half if there is no significant drop in the validation loss for 20 epochs. For this experiment, we split the training dataset into 86 training images and 10 validation images. We then applied the best network that yields the lowest validation loss on the previously selected test images. The results improved significantly as can be seen in the right column of Fig. 10. The mean PSNR results for these test scenes also confirm this improvement (Table III). A detailed visualization of the post-fine tuning results together with the input images, processed patch masks, and a comparison with the no deghosting scenario are depicted in Fig. 11.

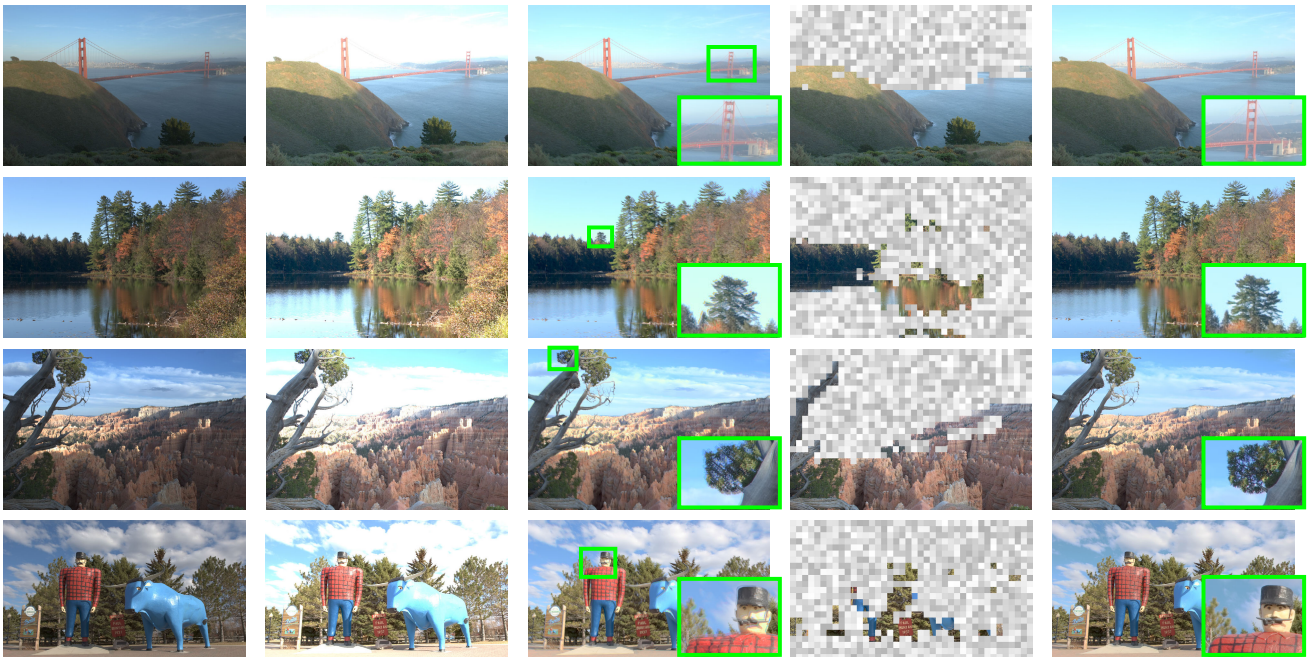
E. Scenario Three: Smartphone Dataset

This scenario is aimed to assess the suitability of our algorithm to be used in a realistic real-world setting. The modern trend in smartphone HDR image/video capture is to use an advanced sensor design that can overlap short and long exposures at every pixel position, known as the DOL-HDR technology [80].

However, rapid object and camera motion can still lead to ghosting artifacts and therefore deghosting remains an important challenge. To this end, we used 6 RAW images that represent the direct sensor output of the smartphone camera. The top half of the images represents the high exposure and bottom half the low exposure (the vertical resolution of the image is twice the vertical resolution of the sensor). The exposure variation is achieved by both exposure time and ISO modulation. We then applied a color processing pipeline that contains black-level subtraction and a simple demosaicking algorithm to obtain two RGB exposures.

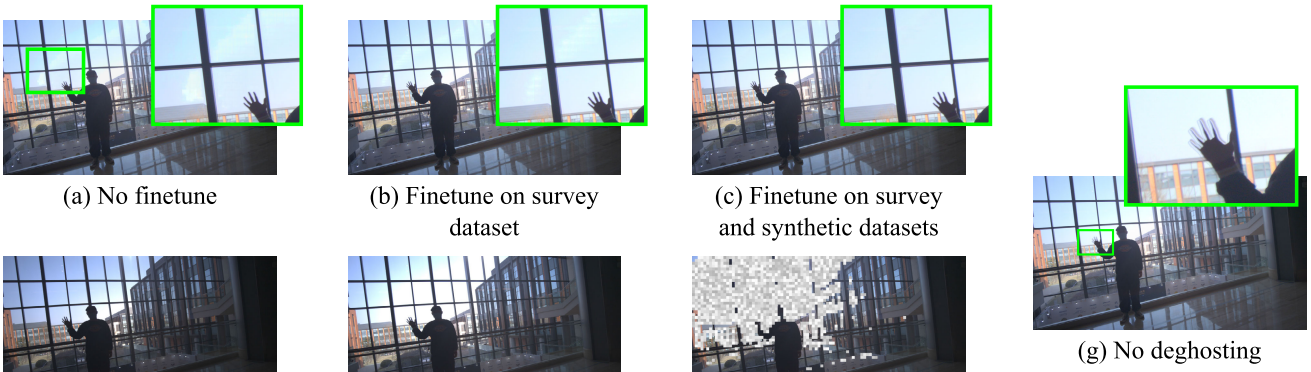
Similar to the HDR photographic survey dataset, we anticipated that fine-tuning would be necessary to produce artifact-free HDR images from these inputs which are collected from an entirely different sensor. As this was not possible due to the lack of ground-truth data for this sensor, we decided to use the HDR photographic survey fine-tuning dataset but also augment it with synthetic images (Section IV-C).

We show the impact of fine-tuning in Fig. 12, where (a) denotes the results of the original network trained on the Stuttgart dataset, (b) the results of fine tuning using only the HDR photographic survey training data, and (c) the results with augmenting this with the synthetic data. It can be seen that each step leads to an improvement in the visual quality of the results. A visual summary of our results for the other images of the smartphone dataset is presented in Fig. 13, where we also show that our algorithm avoids ghosting for the non-saturated regions of the high exposure as the pixels of those regions are not processed by the network, but directly used after exposure scaling in the final image. This optimization is particularly important for smartphones where computational complexity and battery consumption are key factors.



(a) Low Exposure (b) High Exposure (c) No Deghosting (d) Processed Patches (e) Ours

Fig. 11. Four of the HDR photographic survey dataset images.



(a) No finetune (b) Finetune on survey dataset (c) Finetune on survey and synthetic datasets (g) No deghosting

(d) Low exposure (e) High exposure (f) Processed patches

Fig. 12. The effect of fine-tuning in eliminating visual artifacts.

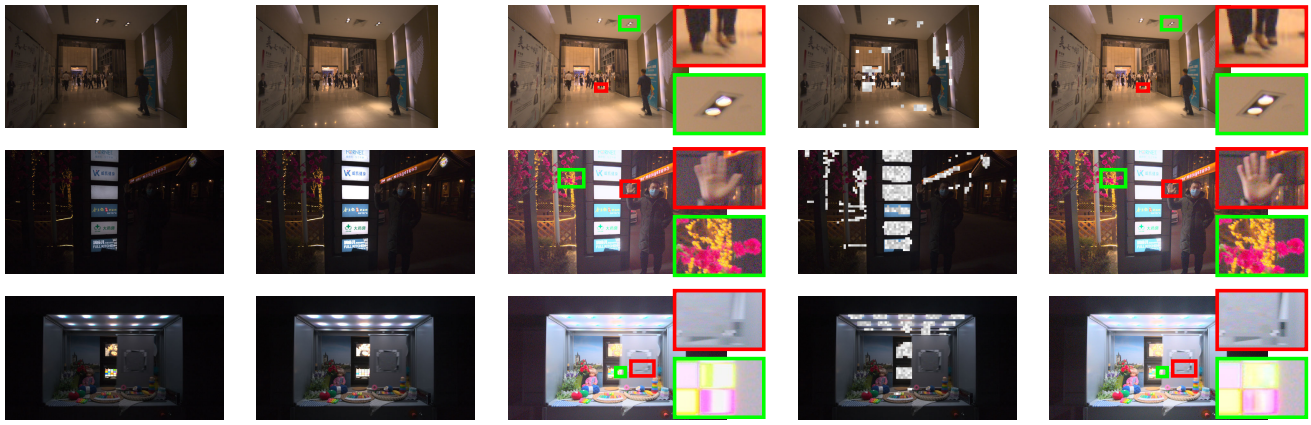
F. Scenario Four: Kalantari Dataset

This dataset is separated into 74 training and 15 testing scenes by the original authors and therefore is commonly used for comparison of deghosting algorithms [10]. For each scene, it contains 3 exposures (low, medium, high) and a ground-truth HDR image aligned to the medium exposure LDR image. This characteristic poses a difficulty for our algorithm in that we use two exposures and produce a result aligned to the high exposure. We therefore decided to use only the low and medium exposures for producing our results, whereas we used all three exposures for the compared studies. Note that this poses a disadvantage for our algorithm as we cannot reconstruct details in the dark regions of the medium exposure, which could be well-exposed in the high exposure.

Also, we split the dataset's training scenes into 66 training and 8 validation scenes to find the network with the minimum validation loss. During training we employ the following additional augmentation strategy in addition to those discussed earlier: with 50% probability low, medium, and ground-truth images are scaled with a random factor in $[1.0, 4.0]$. This is done to generalize the network performance to a more diverse range of luminance values. Similar to the Stuttgart dataset, for training we only used the patches that meet the saturation criteria in the high exposure as explained in Section III-C.

The HDR-VDP and the overall PSNR results that are directly computed in the linear HDR domain (PSNR-L) and after tone-mapping with the μ-law (PSNR-μ) show that our results are comparable with the state-of-the-art algorithms in the literature despite we use a fewer number of exposures (Table IV).

The HDR-VDP scores are computed using the following parameters: 24″ display size, full-HD resolution, and a viewing distance of 0.5 meters. Our method is among the top two



(a) Low Exposure (b) High Exposure (c) No Deghosting (d) Processed Patches (e) Ours

Fig. 13. Smartphone image results. The first row demonstrates a hand-held camera setup with global motion, the second row local object motion with a static camera, and the third row occlusion of a colored region by a moving screen. Red (top) insets show visuals from the dynamic but unprocessed regions of the images due to being well-exposed. Green (bottom) insets show visuals from the processed regions demonstrating our reconstruction quality.

TABLE IV

PSNR-L, PSNR-μ AND HDR-VDP-3.0.6 [81] RESULTS ON THE KALANTARI DATASET ARE PRESENTED IN EACH METRIC SCORE CELL, RESPECTIVELY, SEPARATED WITH A PIPE SYMBOL. THE BOTTOM ROW SHOWS THE MEAN RUNTIME IN SECONDS, AVERAGED OVER 3 RUNS AND THE SCENE COUNT. DISK READ AND WRITE OPERATIONS ARE INCLUDED AS WE FOCUS ON THE TIMING OF THE COMPLETE ALGORITHM PIPELINE (SEE SECTION V-G FOR A MORE DETAILED ANALYSIS). LITERATURE STUDIES ARE RUN ON CPU ONLY DUE TO THE LIMITED GPU HARDWARE CAPACITY AND SOFTWARE CONFIGURATION OF THE TEST MACHINE. THE TIMINGS OF OUR STUDY ARE REPORTED FOR BOTH GPU AND CPU RUNS

Scene	PSNR-L | PSNR-μ | HDR-VDP					
	Sen [12]	Kalantari [10]	Wu [47]	Yan [45]	Prabhakar [37]	Ours
001	50.77 | 47.38 | 9.74	54.24 | 48.65 | 9.76	55.60 | 48.69 | 9.66	55.59 | 50.38 | 9.83	54.41 | 47.73 | 9.79	55.18 | 47.29 | 9.73
002	56.06 | 44.28 | 9.69	58.25 | 45.00 | 9.67	59.05 | 44.64 | 9.57	60.69 | 45.35 | 9.73	57.46 | 42.67 | 9.57	56.66 | 44.18 | 9.66
003	44.38 | 47.56 | 9.53	47.04 | 48.96 | 9.59	44.73 | 47.24 | 9.41	45.32 | 49.42 | 9.70	43.62 | 44.66 | 9.54	45.21 | 47.13 | 9.52
004	44.68 | 41.62 | 9.07	48.25 | 42.23 | 9.09	44.36 | 42.71 | 9.05	43.62 | 42.54 | 9.10	22.62 | 22.48 | 6.60	44.00 | 39.57 | 8.86
005	44.38 | 42.80 | 9.66	43.18 | 42.28 | 9.59	44.59 | 41.62 | 9.50	44.33 | 42.74 | 9.64	43.97 | 40.46 | 9.39	44.79 | 42.72 | 9.65
006	37.86 | 41.18 | 9.39	40.12 | 40.52 | 9.36	34.77 | 40.55 | 9.24	34.66 | 41.82 | 9.44	27.61 | 27.77 | 7.23	34.67 | 39.15 | 9.12
007	20.22 | 32.73 | 7.76	24.43 | 35.08 | 8.39	26.35 | 38.18 | 8.32	26.21 | 39.05 | 8.56	24.73 | 34.29 | 8.31	26.45 | 39.40 | 8.80
008	19.91 | 31.04 | 7.96	20.17 | 31.29 | 8.24	22.75 | 35.21 | 8.85	21.25 | 34.44 | 8.74	23.95 | 36.12 | 8.81	22.35 | 35.02 | 8.62
009	22.96 | 35.19 | 8.02	27.83 | 36.95 | 8.58	30.01 | 41.70 | 8.77	30.05 | 41.98 | 8.74	28.56 | 39.99 | 8.62	30.16 | 42.68 | 8.71
010	15.19 | 27.64 | 7.42	23.04 | 33.66 | 8.02	25.51 | 37.37 | 8.73	25.02 | 37.39 | 8.65	23.22 | 35.21 | 8.57	24.88 | 36.95 | 8.60
BarbequeDay	36.88 | 42.99 | 9.74	39.26 | 44.09 | 9.81	36.20 | 42.06 | 9.67	37.22 | 43.99 | 9.81	34.79 | 38.19 | 9.42	37.51 | 44.02 | 9.82
LadySitting	40.27 | 46.44 | 9.51	47.07 | 48.87 | 9.60	44.77 | 47.25 | 9.41	45.36 | 49.43 | 9.70	44.37 | 46.37 | 9.64	45.31 | 47.14 | 9.52
ManStanding	61.53 | 49.02 | 9.81	64.31 | 49.76 | 9.72	65.55 | 49.46 | 9.65	66.47 | 51.93 | 9.86	62.86 | 47.29 | 9.79	62.82 | 48.66 | 9.79
PeopleStanding	44.24 | 42.77 | 9.66	44.03 | 42.44 | 9.61	44.63 | 41.59 | 9.48	44.83 | 42.83 | 9.63	43.01 | 40.10 | 9.20	45.07 | 42.76 | 9.65
PeopleTalking	36.14 | 41.98 | 9.39	40.75 | 43.27 | 9.37	40.24 | 42.80 | 9.30	41.13 | 43.87 | 9.44	39.06 | 42.37 | 9.38	40.83 | 39.86 | 9.20
Means:	38.36 | 40.97 | 9.09	41.46 | 42.20 | 9.23	41.27 | 42.74 | 9.24	41.45 | 43.81 | 9.37	38.28 | 39.05 | 8.92	41.06 | 42.44 | 9.28
Mean Runtime (s):	64.06	26.35	7.24	13.58	46.97	0.50 | 2.30

methods according to this metric as well. A spatial map of the HDR-VDP results for the *BarbequeDay* scene is illustrated in Fig. 14 where "hotter" colors indicate visually more noticeable differences with respect to the ground-truth (the single HDR-VDP scores reported in Table IV are computed by a spatial pooling of these maps [81]).

In order to understand why the HDR-VDP maps indicate visible differences in certain parts of this scene, we share our results together with Yan et al.'s [45] and the ground-truth in Fig. 15. The closeup views in this figure show that our and Yan et al.'s results are almost indistinguishable from each other and they are both free of ghosting artifacts. The ground-truth image, on the other hand, has some ghosting artifacts due to the motion of the leaves.[1]

[1]Note that in this dataset ground-truth images were created by asking people to stay stationary, but the leaf motion was apparently unavoidable.

Finally, in Table V we share the UDQM results, which is a perceptually validated unified deghosting quality metric [82]. This metric assesses the visibility of various ghosting artifacts as well as the preservation of the dynamic range available in input exposures. Note that this metric can be used in scenarios where ground-truth images are not available – and is therefore unaffected by ghosting in the ground-truths themselves. Our method yields the highest mean value of this metric on the Kalantari test dataset. For validation we include a no deghosting scenario, which yields the lowest UDQM score.

As can be seen in the last row of Table IV, the run-time complexity of our algorithm is significantly lower than the other methods. This is shown in Fig. 16 where it can be seen that our algorithm outperforms the other methods in computation time while also having a favorable metric performance. The detailed timing analysis of our algorithm is further discussed in the following section.

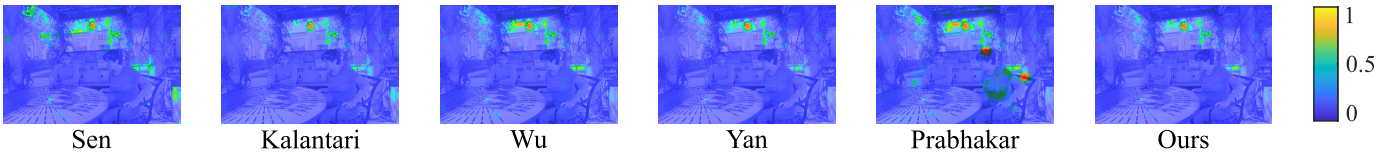


Fig. 14. HDR-VDP maps of the probability of detection of differences of the HDR outputs compared to the ground-truth on the *BarbequeDay* scene of the Kalantari dataset. Full-resolution versions are shared in supplementary materials.

Fig. 15. Visual comparison of our output with Yan et al. [45] on *BarbequeDay* scene of the Kalantari dataset.

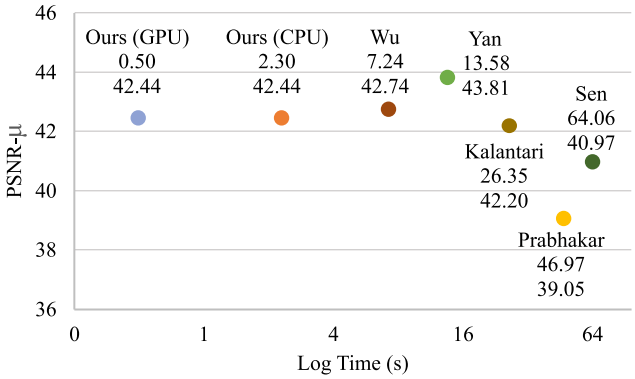


Fig. 16. Computation time and metric performance plot. Values separated as rows represent the total computation time in seconds and PSNR-μ values, respectively.

G. Computational Complexity

In this section, we provide a detailed timing analysis of the different stages of our algorithm. We note that our current implementation uses the Pytorch framework for the network block. In other parts, such as patch processing of the input images and HDR reconstruction, it uses the vector operations of the Numpy library, which is implemented on the CPU. The time taken by each step, whether it is implemented on the CPU or GPU as well as its description are reported in Table VI.[2]

To summarize, the cold-start timing of our overall algorithm in which the network execution happens in the GPU is 0.497 seconds. This includes loading of the model and the input exposures from the disk and writing of the final HDR result back to the disk. The timing for the fully CPU-based version is 2.303 seconds with the primary difference due to slower network inference on the CPU. The hot-start timing of our GPU-based algorithm where the model is pre-loaded, the input exposures are assumed to be picked from the live video frame flow circulating in the device memory, and the resulting HDR frame is written to the device memory for further processing takes only 0.204 seconds.

[2]The following configuration was used to produce the timings: *(i)* CPU: AMD Ryzen 5 5600X, *(ii)* RAM: 32 GB, *(iii)* GPU: NVIDIA GeForce RTX 3070 Ti, *(iv)* SSD: SAMSUNG 980 PRO 1TB NVMe, *(v)* OS: Ubuntu 22.04.

TABLE V

UNIFIED DEGHOSTING QUALITY METRIC (UDQM) SCORES FOR THE KALANTARI DATASET. THE NO DEGHOSTING CONDITION IS INCLUDED FOR VALIDATION PURPOSES. THE LAST ROW SHOWS THAT OUR MEAN UDQM SCORE IS THE BEST AMONG THE COMPARED STUDIES

Scene	No Deghost.	Sen	Kalantari	Wu	Yan	Prabhakar	Ours
001	0.369	0.418	0.407	0.402	0.405	0.403	0.420
002	0.379	0.426	0.414	0.410	0.412	0.409	0.426
003	0.406	0.442	0.435	0.430	0.431	0.435	0.440
004	0.316	0.415	0.401	0.399	0.402	0.413	0.433
005	0.279	0.405	0.387	0.372	0.377	0.365	0.404
006	0.320	0.417	0.410	0.387	0.392	0.386	0.429
007	0.354	0.369	0.364	0.359	0.361	0.359	0.364
008	0.362	0.381	0.386	0.358	0.358	0.341	0.363
009	0.379	0.374	0.380	0.378	0.377	0.374	0.379
010	0.357	0.358	0.366	0.363	0.361	0.361	0.367
B.Da.	0.317	0.400	0.392	0.382	0.386	0.379	0.400
L.Si.	0.402	0.441	0.435	0.429	0.430	0.427	0.440
M.St.	0.374	0.423	0.411	0.408	0.410	0.406	0.425
P.St.	0.269	0.403	0.385	0.368	0.375	0.359	0.403
P.Ta.	0.342	0.422	0.412	0.402	0.407	0.403	0.439
AVG	0.348	0.406	0.399	0.390	0.392	0.388	**0.409**

H. Network Ablation

We conduct network ablation studies using the Kalantari dataset and provide the results in Table VII. All ablation types except increasing the filter counts of layers are shown to reduce our metric performance. As increasing the filter count affects the complexity and inference time of our network negatively, we choose not to employ more number of filters in our network even though it improves metric scores by a small margin. Employment of feature extraction and attention modules in the pipeline improves our algorithm's performance, whereas decreasing and increasing the depth of our U-Net architecture by one level reduces it. We argue that the former might be due to the excessive feature compression at the bottom U-Net level, and the latter due to the insufficient capacity of the network to learn the task with fewer levels. A similar performance

TABLE VI

TIMING RESULTS FOR THE VARIOUS STAGES OF OUR ALGORITHM COMPUTED AS THE AVERAGE OF ALL TEST IMAGES IN THE KALANTARI DATASET. DASHES INDICATE THAT THE CORRESPONDING STAGE IS ONLY IMPLEMENTED ON THE CPU

Runtime Steps	GPU (s)	CPU (s)	Description
Preparation	0.089	0.017	Read and load model from disk and send it to GPU if needed.
LDR Read	–	0.022	Read a bracket of two exposures from disk to CPU memory.
Patching Block	–	0.027	Find the list of patches that need processing.
Network Block	0.131	2.009	Run inference. GPU timing also includes uploading/downloading patches/results to/from GPU.
Reconstruction & Merge Blocks	–	0.046	Linearize original high exposure and then merge the reconstructed patches to their locations.
HDR Write	–	0.182	Write the HDR image produced to disk.
Total: (overall)	**0.497**	**2.303**	Avg. time of processing 1 scene, with preparation and disk access.
Total: (on a device)	**0.204**	**2.082**	Avg. time of processing 1 scene, no preparation, no disk access.

TABLE VII

METRIC RESULTS OF THE ABLATION STUDIES

Ablation Type	PSNR-L	PSNR-μ
DeepDuoHDR	41.06	42.44
Without feature extractor	40.67	42.15
Without attention network	41.01	42.31
Increase U-Net depth (+1)	40.83	42.17
Decrease U-Net depth (-1)	41.00	42.28
More filters in layers (doubled)	41.26	42.46
Fewer filters in layers (halved)	40.76	42.35

reduction is also seen when we reduce the number of filters, which is arguably related to the decreased network capacity. Considering all of those ablation experiments and computational costs, we finalized our network structure as explained in Section III-D.3.

VI. DISCUSSION & CONCLUSION

In this paper, we proposed a low complexity two-exposure HDR deghosting algorithm, specifically focused towards adoption on mobile devices, that has high performance under diverse and challenging scenarios. We evaluated our algorithm's performance independently with Stuttgart, HDR Photographic Survey, and smartphone datasets. As re-training existing methods on these datasets would be impractical and possibly error-prone, we used the Kalantari dataset for literature comparisons using the authors' publicly available codes.

We found that the run-time complexity of our algorithm is significantly lower than that of existing methods without compromising deghosting quality. According to the UDQM results, our algorithm yielded the best deghosting performance. In terms of HDR-VDP and PSNR metrics, our algorithm is also among the top ones. These results were achieved at a fraction of the computational cost of these algorithms. This reduced complexity can be attributed to three factors: *(i)* using two exposures instead of the commonly used three, *(ii)* avoiding the processing of the pixels that can be reconstructed from the high exposure only, and *(iii)* employing a simpler network architecture that predicts an aligned low patch instead of an HDR one.

TABLE VIII

PROCESSED PIXELS PERCENTAGE OF KALANTARI DATASET

Scene	Percentage	Scene	Percentage
001	10.60	009	80.12
002	10.18	010	79.68
003	5.84	BarbequeDay	27.80
004	11.06	LadySitting	5.84
005	30.41	ManStanding	3.07
006	36.83	PeopleStanding	30.41
007	60.71	PeopleTalking	11.02
008	55.99	**Average:**	30.64

In Table VIII, we show the percentage of pixels processed by our algorithm for the Kalantari dataset. It can be seen from this table that if a high exposure has only a few saturated regions, we also perform proportionally less processing. The degree of this processing can be controlled using the parameters of our algorithm that were described in Section III-C. If our algorithm is implemented on a device with a smaller processing power, one can use less aggressive values for the saturation (S_t) and saturated pixel count (C_s).

As for the complexity of our algorithm, we found it to have $630.739K$ parameters and a GMAC value of 1.354 for processing a pair of 64×64 patches. If we assume that 30% of the pixels will be processed for a full-HD exposure pair, our total processing cost would amount to 205.6 GMACs/frame. This compares very well against a state-of-the-art algorithm [45], which has a GMAC value of 2916.62 for processing a full-HD exposure triplet [6]. This reduced complexity of our algorithm allows it to be used for rapidly creating HDR videos comprised of more than 1000 frames as in the Stuttgart dataset.

It is worthwhile to emphasize that the use of attention based structures for HDR image deghosting it not the chief novelty of our study – many recent deghosting algorithms use similar attention mechanisms (see Table I). However, it should be noted that our overall deghosting framework is distinctly different from the existing studies. In our work, we show that significant computational advantages can be gained by not processing well-saturated parts of the reference image. We also show that seams between processed and non-processed regions, as well as between two processed regions can be avoided by making the receptive field of the network larger than the final output patch size. We also show that an HDR deghosting algorithm does not need to produce an HDR image: by only producing the low exposure that is consistent with the high exposure, we allow any HDR merge algorithm (classical or learning-based) to be used, providing modularity. Additionally, to further reduce the computational load, we use only two exposures instead of the commonly used three and show that similar results can be obtained with algorithms that use three exposures. Finally, we devise a synthetic data method which allows the network to perform well for a new sensor for which no training data is available. All of these computational improvements are introduced without compromising quality

as can be seen in the experimental evaluation section of our manuscript.

In this work, we focused on using the high exposure as the reference and reconstructing its saturated details from the low exposure. While this is desirable from a signal-to-noise ratio perspective, the mirror scenario in which the low is the reference may be preferred in some applications, for example for mitigating the motion blur. We leave the exploration of this alternative for future work. Another promising research direction can be to learn to reconstruct dense flow vectors instead of reconstructing images. Perlin noise-based flow simulation described in Section IV-B can be used to generate ground-truth flow vectors for this purpose. Finally, performing deghosting and HDR reconstruction directly in the Bayer domain may allow for a greater reduction in computational complexity, which we plan to address in future work.

REFERENCES

[1] P. Debevec, "Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography," in *Proc. 25th Annu. Conf. Comput. Graph. Interact. Techn.*, New York, NY, USA, 1998, pp. 189–198.

[2] F. Banterle, A. Artusi, K. Debattista, and A. Chalmers, *Advanced High Dynamic Range Imaging*. Boca Raton, FL, USA: CRC Press, 2017.

[3] O. T. Tursun, A. O. Akyüz, A. Erdem, and E. Erdem, "The state of the art in HDR deghosting: A survey and evaluation," *Comput. Graph. Forum*, vol. 34, no. 2, pp. 683–707, May 2015.

[4] A. Akyüz, O. Tursun, J. Hasić-Telalović, and K. Karađuzović-Hadžiabdić, "Chapter 1—Ghosting in HDR video," in *High Dynamic Range Video*, A. Chalmers, P. Campisi, P. Shirley, and I. G. Olaizola, Eds., Cambridge, MA, USA: Academic Press, 2017, ch. 1, pp. 3–44. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780128094778000017

[5] L. Wang and K.-J. Yoon, "Deep learning for HDR imaging: State-of-the-art and future trends," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 12, pp. 8874–8895, Dec. 2022.

[6] E. Pérez-Pellitero et al., "NTIRE 2022 challenge on high dynamic range imaging: Methods and results," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2022, pp. 1008–1022.

[7] J. Froehlich, S. Grandinetti, B. Eberhardt, S. Walter, A. Schilling, and H. Brendel, "Creating cinematic wide gamut HDR-video for the evaluation of tone mapping operators and HDR-displays," *Proc. SPIE*, vol. 9023, pp. 279–288, Mar. 2014.

[8] T. Mitsunaga and S. K. Nayar, "Radiometric self calibration," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2, Jun. 1999, pp. 374–380.

[9] M. Robertson, S. Borman, and R. Stevenson, "Estimation-theoretic approach to dynamic range enhancement using multiple exposures," *J. Electron. Imag.*, vol. 12, no. 2, pp. 219–228, 2003.

[10] N. K. Kalantari and R. Ramamoorthi, "Deep high dynamic range imaging of dynamic scenes," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 1–144, 2017.

[11] N. K. Kalantari and R. Ramamoorthi, "Deep HDR video from sequences with alternating exposures," *Comput. Graph. Forum*, vol. 38, no. 2, pp. 193–205, May 2019.

[12] P. Sen, N. K. Kalantari, M. Yaesoubi, S. Darabi, D. B. Goldman, and E. Shechtman, "Robust patch-based HDR reconstruction of dynamic scenes," *ACM Trans. Graph.*, vol. 31, no. 6, pp. 1–11, Nov. 2012.

[13] P. Sen, "Overview of state-of-the-art algorithms for stack-based high-dynamic range (HDR) imaging," *Electron. Imag.*, vol. 30, no. 5, pp. 311-1–311-8, Jan. 2018.

[14] J. Hu, O. Gallo, K. Pulli, and X. Sun, "HDR deghosting: How to deal with saturation?" in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 1163–1170.

[15] H. Zimmer, A. Bruhn, and J. Weickert, "Freehand HDR imaging of moving scenes with simultaneous resolution enhancement," *Comput. Graph. Forum*, vol. 30, no. 2, pp. 405–414, Apr. 2011.

[16] J. Zheng, Z. Li, Z. Zhu, S. Wu, and S. Rahardja, "Hybrid patching for a sequence of differently exposed images with moving objects," *IEEE Trans. Image Process.*, vol. 22, no. 12, pp. 5190–5201, Dec. 2013.

[17] G. Ward, "Fast, robust image registration for compositing high dynamic range photographs from hand-held exposures," *J. Graph. Tools*, vol. 8, no. 2, pp. 17–30, Jan. 2003.

[18] S. Wu, S. Xie, and Z. Li, "An edge-based method for aligning differently exposed images," in *Proc. 7th IEEE Conf. Ind. Electron. Appl. (ICIEA)*, Jul. 2012, pp. 813–818.

[19] S. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, "High dynamic range video," *ACM Trans. Graph.*, vol. 22, no. 3, pp. 319–325, Jul. 2003.

[20] S. Mangiat and J. Gibson, "High dynamic range video with ghost removal," *Proc. SPIE*, vol. 7798, pp. 307–314, Sep. 2010.

[21] S. Mangiat and J. Gibson, "Spatially adaptive filtering for registration artifact removal in HDR video," in *Proc. 18th IEEE Int. Conf. Image Process.*, Sep. 2011, pp. 1317–1320.

[22] N. K. Kalantari, E. Shechtman, C. Barnes, S. Darabi, D. B. Goldman, and P. Sen, "Patch-based high dynamic range video," *ACM Trans. Graph.*, vol. 32, no. 6, pp. 1–202, 2013.

[23] X. Zhang, Q. Zhu, T. Hu, and Q. Yan, "EiffHDR: An efficient network for multi-exposure high dynamic range imaging," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2024, pp. 6560–6564.

[24] J. Xiao, Q. Ye, T. Liu, C. Zhang, and K.-M. Lam, "Deep progressive feature aggregation network for multi-frame high dynamic range imaging," *Neurocomputing*, vol. 594, Aug. 2024, Art. no. 127804. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231224005757

[25] H. Li, Z. Yang, Y. Zhang, D. Tao, and Z. Yu, "Single-image HDR reconstruction assisted ghost suppression and detail preservation network for multi-exposure HDR imaging," *IEEE Trans. Comput. Imag.*, vol. 10, pp. 429–445, 2024.

[26] Q. Yan et al., "SMAE: Few-shot learning for HDR deghosting with saturation-aware masked autoencoders," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 5775–5784.

[27] Q. Yan, W. Chen, S. Zhang, Y. Zhu, J. Sun, and Y. Zhang, "A unified HDR imaging method with pixel and patch level," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 22211–22220.

[28] F. Zhou, Z. Fu, and D. Zhang, "High dynamic range imaging with context-aware transformer," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jun. 2023, pp. 1–8.

[29] Q. Yan et al., "Dual-attention-guided network for ghost-free high dynamic range imaging," *Int. J. Comput. Vis.*, vol. 130, no. 1, pp. 76–94, Jan. 2022.

[30] S. Catley-Chandar, T. Tanay, L. Vandroux, A. Leonardis, G. Slabaugh, and E. Pérez-Pellitero, "FlexHDR: Modelling alignment and exposure uncertainties for flexible HDR imaging," 2022, *arXiv:2201.02625*.

[31] U. Cogalan, M. Bemana, K. Myszkowski, H.-P. Seidel, and T. Ritschel, "Learning HDR video reconstruction for dual-exposure sensors with temporally-alternating exposures," *Comput. Graph.*, vol. 105, pp. 57–72, Jun. 2022.

[32] Y. Jiang, I. Choi, J. Jiang, and J. Gu, "HDR video reconstruction with tri-exposure quad-bayer sensors," 2021, *arXiv:2103.10982*.

[33] Z. Liu et al., "ADNet: Attention-guided deformable convolutional network for high dynamic range imaging," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2021, pp. 463–470.

[34] Y. Niu, J. Wu, W. Liu, W. Guo, and R. W. H. Lau, "HDR-GAN: HDR image reconstruction from multi-exposed LDR images with large motions," *IEEE Trans. Image Process.*, vol. 30, pp. 3885–3896, 2021.

[35] M. Anand, N. Harilal, C. Kumar, and S. Raman, "HDRVideo-GAN: Deep generative HDR video reconstruction," in *Proc. 12th Indian Conf. Comput. Vis., Graph. Image Process.*, Dec. 2021, pp. 1–9.

[36] G. Chen, C. Chen, S. Guo, Z. Liang, K. K. Wong, and L. Zhang, "HDR video reconstruction: A coarse-to-fine network and a real-world benchmark dataset," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 2482–2491.

[37] K. R. Prabhakar, S. Agrawal, and R. V. Babu, "Self-gated memory recurrent network for efficient scalable HDR deghosting," *IEEE Trans. Comput. Imag.*, vol. 7, pp. 1228–1239, 2021.

[38] K. R. Prabhakar, S. Agrawal, D. K. Singh, B. Ashwath, and R. V. Babu, "Towards practical and efficient high-resolution HDR deghosting with CNN," in *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXI 16*. Springer, 2020, pp. 497–513.

[39] Q. Yan et al., "Ghost removal via channel attention in exposure fusion," *Comput. Vis. Image Understand.*, vol. 201, Dec. 2020, Art. no. 103079.

[40] Q. Yan et al., "Deep HDR imaging via a non-local network," *IEEE Trans. Image Process.*, vol. 29, pp. 4308–4322, 2020.

[41] T. V. Vo and C. Lee, "High dynamic range video synthesis using superpixel-based illuminance-invariant motion estimation," *IEEE Access*, vol. 8, pp. 24576–24587, 2020.

[42] T. Suda, M. Tanaka, Y. Monno, and M. Okutomi, "Deep snapshot HDR imaging using multi-exposure color filter array," in *Proc. Asian Conf. Comput. Vis.*, 2020, pp. 1–18.

[43] S. Choi, J. Cho, W. Song, J. Choe, J. Yoo, and K. Sohn, "Pyramid inter-attention for high dynamic range imaging," *Sensors*, vol. 20, no. 18, p. 5102, Sep. 2020.

[44] K. R. Prabhakar, R. Arora, A. Swaminathan, K. P. Singh, and R. V. Babu, "A fast, scalable, and reliable deghosting method for extreme exposure fusion," in *Proc. IEEE Int. Conf. Comput. Photography (ICCP)*, May 2019, pp. 1–8.

[45] Q. Yan et al., "Attention-guided network for ghost-free high dynamic range imaging," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1751–1760.

[46] O. Martorell, C. Sbert, and A. Buades, "Ghosting-free DCT based multi-exposure image fusion," *Signal Process., Image Commun.*, vol. 78, pp. 409–425, Oct. 2019.

[47] S. Wu, J. Xu, Y. W. Tai, and C. K. Tang, "Deep high dynamic range imaging with large foreground motions," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 117–132.

[48] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11.

[49] T. Mertens, J. Kautz, and F. Van Reeth, "Exposure fusion," in *Proc. 15th Pacific Conf. Comput. Graph. Appl. (PG)*, Oct. 2007, pp. 382–390.

[50] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015, *arXiv:1503.02531*.

[51] J. Liu, J. Tang, and G. Wu, "Residual feature distillation network for lightweight image super-resolution," in *Computer Vision—ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*. Springer, 2020, pp. 41–55.

[52] M. Aggarwal and N. Ahuja, "Split aperture imaging for high dynamic range," *Int. J. Comput. Vis.*, vol. 58, no. 1, pp. 7–17, Jun. 2004.

[53] M. D. Tocci, C. Kiser, N. Tocci, and P. Sen, "A versatile HDR video production system," *ACM Trans. Graph.*, vol. 30, no. 4, pp. 1–10, Jul. 2011.

[54] S. K. Nayar and T. Mitsunaga, "High dynamic range imaging: Spatially varying pixel exposures," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2000, pp. 472–479.

[55] M. Schöberl, A. Belz, J. Seiler, S. Foessel, and A. Kaup, "High dynamic range video by spatially non-regular optical filtering," in *Proc. 19th IEEE Int. Conf. Image Process.*, Sep. 2012, pp. 2757–2760.

[56] A. Serrano, F. Heide, D. Gutierrez, G. Wetzstein, and B. Masia, "Convolutional sparse coding for high dynamic range imaging," *Comput. Graph. Forum*, vol. 35, no. 2, pp. 153–163, May 2016.

[57] S. Hajisharif, J. Kronander, and J. Unger, "Adaptive dualISO HDR reconstruction," *EURASIP J. Image Video Process.*, vol. 2015, no. 1, pp. 1–13, Dec. 2015.

[58] R. G. Rodríguez and M. Rodríguez, "High quality video in high dynamic range scenes from interlaced dual-ISO footage," in *Proc. IST Intl. Symp. Electron. Imaging Sci. Technol.*, San Francisco, CA, USA, 2016, p. 7.

[59] I. Choi, S.-H. Baek, and M. H. Kim, "Reconstructing interlaced high-dynamic-range video using joint learning," *IEEE Trans. Image Process.*, vol. 26, no. 11, pp. 5353–5366, Nov. 2017.

[60] U. çogalan and A. O. Akyuz, "Deep joint deinterlacing and denoising for single shot dual-ISO HDR reconstruction," *IEEE Trans. Image Process.*, vol. 29, pp. 7511–7524, 2020.

[61] J. Gu, Y. Hitomi, T. Mitsunaga, and S. Nayar, "Coded rolling shutter photography: Flexible space-time sampling," in *Proc. IEEE Int. Conf. Comput. Photography (ICCP)*, Mar. 2010, pp. 1–8.

[62] H. Cho, S. J. Kim, and S. Lee, "Single-shot high dynamic range imaging using coded electronic shutter," *Comput. Graph. Forum*, vol. 33, no. 7, pp. 329–338, Oct. 2014.

[63] V. G. An and C. Lee, "Single-shot high dynamic range imaging via deep convolutional neural network," in *Proc. Asia–Pacific Signal Inf. Process. Assoc. Annu. Summit Conf. (APSIPA ASC)*, Dec. 2017, pp. 1768–1772.

[64] A. G. Vien and C. Lee, "Single-shot high dynamic range imaging via multiscale convolutional neural network," *IEEE Access*, vol. 9, pp. 70369–70381, 2021.

[65] Y. Xu, Z. Liu, X. Wu, W. Chen, C. Wen, and Z. Li, "Deep joint demosaicing and high dynamic range imaging within a single shot," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 7, pp. 4255–4270, Jul. 2022.

[66] F. Heide et al., "FlexISP: A flexible camera image processing framework," *ACM Trans. Graph.*, vol. 33, no. 6, pp. 1–13, 2014.

[67] L. Wang, L.-Y. Wei, K. Zhou, B. Guo, and H.-Y. Shum, "High dynamic range image hallucination," *Rendering Techn.*, vol. 321, p. 3, Jun. 2007.

[68] F. Banterle, P. Ledda, K. Debattista, and A. Chalmers, "Inverse tone mapping," in *Proc. 4th Int. Conf. Comput. Graph. Interact. Techn. Australasia Southeast Asia*, 2006, pp. 349–356.

[69] A. G. Rempel et al., "LDR2HDR: On-the-fly reverse tone mapping of legacy video and photographs," *ACM Trans. Graph.*, vol. 26, no. 3, p. 39, 2007.

[70] Y. Endo, Y. Kanamori, and J. Mitani, "Deep reverse tone mapping," *ACM Trans. Graph.*, vol. 36, no. 6, pp. 1–10, Dec. 2017.

[71] D. Marnerides, T. Bashford-Rogers, and K. Debattista, "Deep HDR hallucination for inverse tone mapping," *Sensors*, vol. 21, no. 12, p. 4032, Jun. 2021.

[72] M. Santana Santos, T. Ing Ren, and N. Khademi Kalantari, "Single image HDR reconstruction using a CNN with masked features and perceptual loss," 2020, *arXiv:2005.07335*.

[73] S. Lee, G. Hwan An, and S.-J. Kang, "Deep recursive HDRI: Inverse tone mapping using generative adversarial networks," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 596–611.

[74] S. Hasinoff et al., "Burst photography for high dynamic range and low-light imaging on mobile cameras," *ACM Trans. Graph.*, vol. 35, no. 6, pp. 1–12, 2016.

[75] B. Lecouat, T. Eboli, J. Ponce, and J. Mairal, "High dynamic range and super-resolution from raw image bursts," *ACM Trans. Graph.*, vol. 41, no. 4, pp. 1–21, Jul. 2022.

[76] A. O. Akyüz and E. Reinhard, "Noise reduction in high dynamic range imaging," *J. Vis. Commun. Image Represent.*, vol. 18, no. 5, pp. 366–376, Oct. 2007.

[77] D. M. Fairchild, "The HDR photographic survey," in *Proc. 15th Color Imag. Conf. (CIC)*, Nov. 2007, pp. 233–238.

[78] K. Perlin, "Improving noise," in *Proc. 29th Annu. Conf. Comput. Graph. Interact. Techn.*, Jul. 2002, pp. 681–682.

[79] F. Kingdom and B. Moulden, "Border effects on brightness: A review of findings, models and issues," *Spatial Vis.*, vol. 3, no. 4, pp. 225–262, 1988.

[80] S. Dabral, "Image data processing for digital overlap wide dynamic range sensors," U.S. Patent 17 168 224, May 27, 2021.

[81] R. Mantiuk, K. J. Kim, A. G. Rempel, and W. Heidrich, "HDR-VDP-2: A calibrated visual metric for visibility and quality predictions in all luminance conditions," *ACM Trans. Graph.*, vol. 30, no. 4, pp. 1–14, Jul. 2011.

[82] O. T. Tursun, A. O. Akyüz, A. Erdem, and E. Erdem, "An objective deghosting quality metric for HDR images," *Comput. Graph. Forum*, vol. 35, no. 2, pp. 139–152, May 2016.

[83] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 1–12.

[84] M. Folk, G. Heber, Q. Koziol, E. Pourmal, and D. Robinson, "An overview of the HDF5 technology suite and its applications," in *Proc. EDBT/ICDT Workshop Array Databases*, Mar. 2011, pp. 36–47.