Deep HDR Video from Sequences with Alternating Exposures

Nima Khademi Kalantari[1] and Ravi Ramamoorthi[2]

[1]Texas A&M University
[2]University of California, San Diego

Abstract

A practical way to generate a high dynamic range (HDR) video using off-the-shelf cameras is to capture a sequence with alternating exposures and reconstruct the missing content at each frame. Unfortunately, existing approaches are typically slow and are not able to handle challenging cases. In this paper, we propose a learning-based approach to address this difficult problem. To do this, we use two sequential convolutional neural networks (CNN) to model the entire HDR video reconstruction process. In the first step, we align the neighboring frames to the current frame by estimating the flows between them using a network, which is specifically designed for this application. We then combine the aligned and current images using another CNN to produce the final HDR frame. We perform an end-to-end training by minimizing the error between the reconstructed and ground truth HDR images on a set of training scenes. We produce our training data synthetically from existing HDR video datasets and simulate the imperfections of standard digital cameras using a simple approach. Experimental results demonstrate that our approach produces high-quality HDR videos and is an order of magnitude faster than the state-of-the-art techniques for sequences with two and three alternating exposures.

CCS Concepts
• ***Computing methodologies*** → ***Computational photography;***

1. Introduction

One of the major drawbacks of standard digital cameras is their inability to capture the full range of illumination in the scene. Extensive research has been done in the past decades to address this limitation and significant progress has been made in the case of still images. These approaches typically produce a high dynamic range (HDR) image through bracketed exposure imaging [DM97, SKY*12, HGPS13, OLTK15, MLY*17, KR17] or burst image denoising [LYT*14, HSG*16]. As a consequence, HDR imaging is now popular and available to the public through smartphone cameras and commercial software like Adobe Photoshop.

On the other hand, HDR video remains out of reach for the public as the majority of approaches focus on specialized cameras. These cameras are often bulky and expensive since they need, for example, complex optical systems [TKTS11] or sensors [ZSFC*15]. To generate HDR videos using inexpensive off-the-shelf cameras, we can capture the input low dynamic range (LDR) sequences by alternating the exposure of each frame. The HDR video can then be reconstructed by recovering the missing HDR details at each frame, from the neighboring images with different exposure.

Compared to bracketed exposure HDR imaging, the problem of HDR video reconstruction has received relatively less attention. Perhaps, the main reason is that this problem is more challenging than that of producing HDR images; in addition to producing artifact-free HDR images, a high-quality HDR video requires the frames to be temporally coherent. Although a few methods have been proposed to address this problem [KUWS03, MG11, KSB*13, LLM17], they are typically slow and not able to handle challenging cases (see Figs. 11, 13, and 14).

Our approach is inspired by the work of Kalantari and Ramamoorthi [KR17] on using deep learning for HDR image reconstruction. They divide the problem into two stages of alignment and HDR merge, use an existing optical flow method [Liu09] for the alignment, and model the merge process using a convolutional neural network (CNN). This approach takes three images with different exposures as the input and assumes that the medium exposure image is the reference. Since in our application the reference image at every frame has different exposure, this method cannot be directly used to generate HDR videos. Although we could potentially adapt this approach to our application, it shares the main drawback of existing HDR video techniques as it uses slow, optimization-based optical flow methods for alignment. Moreover, the existing optical flow methods are not optimized to produce high-quality HDR videos, and thus, this approach is suboptimal (Fig. 3).

We propose to address these problems by modeling *both* the alignment and HDR merge components using two sequential CNNs and train the two networks in an end-to-end fashion by minimizing the error between the estimated and ground truth HDR frames. In our system, we perform alignment by estimating the flow between the neighboring frames and the current frame (called reference hereafter) using a CNN (*flow network*), which is specifically designed for this application and performs better than existing learning-based optical flow methods [DFI*15, IMS*17] (see Fig. 4). These estimated flows are then used to warp the neighboring frames and produce a set of aligned images. We then produce the final HDR frames from the aligned images using a CNN (*merge network*), similar to Kalantari and Ramamoorthi [KR17], but with a few necessary changes that substantially improve the quality of the results (see Figs. 7 and 8).

As is common with deep learning systems, we need a large dataset of input LDR frames and their corresponding ground truth HDR frames to properly train our networks. We produce our training dataset by synthetically extracting the input LDR images from a set of HDR videos [FGE*14, KGB*14]. To avoid overfitting to this synthetically generated dataset, we simulate the imperfections of standard digital cameras by adding noise to the input LDR frames and perturbing their tone. Although simple, this process is essential for our trained network to generalize well and work on the input videos captured by off-the-shelf cameras, such as Basler acA2000-50gc. In summary, we make the following contributions:

- We propose the first deep learning approach to produce an HDR video from a sequence of alternating exposures. Our method is practical, produces high-quality results, and is 50 to 110 times faster than current techniques (Table 4).
- We present a flow network which is specifically designed for HDR video reconstruction application (Sec. 3.2) and performs better than the existing non-learning (Fig. 3) and learning-based (Fig. 4) optical flow approaches.
- We apply necessary changes to the input and architecture of the merge network by Kalantari et al. [KR17] (Sec. 3.3) to significantly improve the quality of the results (Figs. 7 and 8).

2. Related Work

Over the past decades, many powerful approaches have been developed to produce still HDR images from sequences with different exposures [DM97, SKY*12, HGPS13, OLTK15, MLY*17, KR17, WXTT18], burst images [LYT*14, HSG*16], or a single LDR image [EKD*17, EKM17, MBRHD18]. However, most of these approaches only demonstrate results for generating still HDR images and are not suitable for producing HDR videos [KSB*13]. The notable exception is the approach by Eilertsen et al. [EKD*17], which demonstrates HDR videos, hallucinated from input LDR videos with a single exposure. However, their method can only hallucinate *small* saturated regions and does not handle the noise in the dark areas. Moreover, this approach is also not designed to handle videos and produces results with flickering artifacts. For brevity, here we only focus on the algorithms that are designed to handle HDR video.

A large number of approaches propose to capture HDR images and videos through coded per-pixel [NM00, NN02, SHG*16] or scanline exposure/ISO [HST*14, HKU15, CBK17], and generate the results using appropriate reconstruction techniques. These methods can to work on input images with varying ISO, which can be captured with off-the-shelf cameras. Therefore, similar to our method, these techniques are also practical. However, while they do not need to handle motion, they often have difficulties generating high-quality results in the regions with high contrast. Moreover, changing the ISO is not as effective as changing the shutter time in reducing the noise. Zhao et al. [ZSFC*15] propose to reconstruct HDR images using modulus images, but their approach requires special sensors. Other methods capture images with different exposures simultaneously by splitting the light to different sensors using internal [TKTS11] or external [MMP*07] beam-splitters. However, these approaches require a specific optical design.

A category of approaches reconstruct HDR videos from input sequences that are captured by alternating the exposure of each frame. Kang et al. [KUWS03] propose the first HDR video reconstruction algorithm for sequences with alternating exposures by using optical flow to align neighboring frames to the reference frame. They then combine the aligned images with the reference frame using a weighting strategy to avoid ghosting. However, in cases with large motion, their approach typically introduces optical flow artifacts in the final results, as demonstrated in Figs. 11, 13 and 15.

Mangiat and Gibson [MG10] improve Kang et al.'s approach using a block-based motion estimation method coupled with a refinement stage. In a follow up work, they propose to filter the regions with large motion to reduce the blocking artifacts [MG11]. However, their approach still shows blocking artifacts in cases with large motion (Figs. 14 and 17). Moreover, their method is limited to handling sequences with only two alternating exposures.

Kalantari et al. [KSB*13] propose a patch-based optimization system to synthesize the missing exposures at each frame. These images are then combined to produce the final HDR frame. To increase the temporal coherency, they estimate an initial motion between the neighboring and reference frames. They then constrain the patch search to a small window around the predicted motion, where the size of the window is obtained by a greedy approach. This method produces results that are generally significantly better than the other approaches. However, it usually takes several minutes to solve the complex patch-based optimization and produce a single HDR frame. In contrast, our approach is generally 80 to 110 times faster than their method, taking only a few seconds to generate an HDR frame. Moreover, this approach is often not able to properly constrain the patch search and over/under-estimates the search window size. In these cases, it produces results with ghosting artifacts (Figs. 11, 13 and 15) or wobbly and unnatural motion (see supplementary video).

Gryaditskaya et al. [GPR*15] improve the method of Kalantari et al. [KSB*13] by adaptively adjusting the exposures. However, the idea of adaptive exposures can also be used to improve our system and is orthogonal to our contribution. Finally, the recent method of Li et al. [LLM17] poses the HDR video reconstruction problem as maximum a posteriori estimation. Specifically, they separate the problem of HDR frame reconstruction to finding the foreground and background in each frame. They propose to find the background using rank minimization and compute the foreground using a multiscale adaptive regression technique. Unfortunately, this approach is computationally expensive, taking roughly 2 hours to generate a frame with resolution of 1280 × 720. Moreover, as shown in Fig. 14, their method produces results with noise, ghosting, and discoloration in challenging cases.

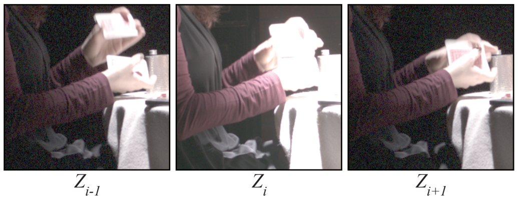

Figure 1: *We show a cropped version of three consecutive frames of the* POKER FULLSHOT *scene with two alternating exposures. Each frame of the LDR input video is missing some contents. For example, Z_{i-1} and Z_{i+1} are captured with low exposure and contain noise on the lady's dress, while the high exposure frame, Z_i, is missing the details on the lady's hand. To reconstruct an HDR image at each frame, the missing content needs to be reconstructed from the neighboring frames of different exposure.*

$\tilde{Z}_i$	input LDR frames with alternating exposures
Z_i	input LDR frames after alignment and CRF replacement
H_i	the HDR image at frame i
T_i	the HDR image at frame i in the log domain
t_i	exposure time at frame i
$h(Z_i)$	takes image Z_i from the LDR to the linear HDR domain: $h(Z_i) = Z_i^{\gamma}/t_i$
I_i	result of taking image Z_i to the linear HDR domain, i.e., $I_i = h(Z_i)$
$l_i(I_j)$	takes image I_j from the linear domain to the LDR domain at exposure i: $l_i(I_j) = \text{clip}[(I_j t_i)^{1/\gamma}]$
$g_i(Z_j)$	adjust the exposure of image Z_j to that of frame i, i.e., $g_i(Z_j) = l_i(h(Z_j))$
$Z_{i-1,i}$	the result of aligning image Z_{i-1} to Z_i.

Table 1: *The complete list of notations used in the paper.*

3. Deep HDR Video Reconstruction

The goal of our algorithm is to produce a high-quality HDR video from an input LDR video with alternating exposures. For simplicity, we explain our method for the case with *two* alternating exposures and discuss the extension to three exposures later in Sec. 3.4. In this case, as shown in Fig. 1, the input LDR video consists of a set of frames, Z_i, alternating between low and high exposures. The frames with low exposure are usually noisy in the dark regions, while the high exposure frames lack content in the bright areas because of the sensor saturation.

To produce an HDR frame, H_i, we need to reconstruct the missing content at frame i (reference) using the neighboring frames with different exposures (Z_{i-1} and Z_{i+1}). This is a challenging problem as it requires reconstructing high-quality and temporally coherent HDR frames. Existing approaches typically first align the neighboring images to the reference frame and then merge them into an HDR image. However, they often require solving complex optimization systems [KSB*13, LLM17], which makes them slow. Moreover, they usually use logical, but heuristically designed components (e.g., the HDR merge component of Kang et al. [KUWS03]), and thus, fail to produce satisfactory results in challenging cases.

We address the drawbacks of previous approaches by proposing to use convolutional neural networks (CNN) to learn the HDR video reconstruction process from a set of training scenes. Specifically, our approach builds upon the recent HDR image reconstruction method of Kalantari and Ramamoorthi [KR17], which breaks down the process into alignment and HDR merge stages and uses a CNN to model the merge process.

In our system, in addition to modeling the merge process using the merge network (Sec. 3.3), we propose a flow network (Sec. 3.2) to perform the alignment process. We train these two networks in an end-to-and fashion by minimizing the error between the reconstructed and ground truth HDR frames on a set of training scenes. Our learned flow network is designed for the HDR video reconstruction application and performs better (Fig. 3) than the traditional optical flow methods, as used by Kalantari and Ramamoorthi, and learning-based flow estimation approaches. We also substantially improve their merge network by proposing two critical changes (Figs. 7 and 8). During training, our system learns to produce HDR videos that are close to the ground truth based on an ℓ_1 metric (Eq. 6) and generalizes well to the test sequences, as shown in Sec. 5. Since each individual frame is of high quality, the resulting videos have reasonable temporal coherency. We show the overview of our algorithm in Fig. 2.

Note that, while it is possible to model the HDR video reconstruction process using a single CNN, training such a system on limited training scenes would be significantly difficult. By dividing the entire process into two stages, we provide a simpler and physically meaningful task to each network, and thus, make the training more tractable. Moreover, as discussed later in Sec. 4.2, the two-stage architecture is essential for generalizing our system to work with cameras that it has not been trained on.

3.1. Preprocessing

To reduce the complexity of the process for our learning system, we first globally align the neighboring frames to the reference frame using a similarity transform (rotation, translation, and isometric scale). We do so by finding the corresponding corner features in the reference and each neighboring image and then using RANSAC to find the dominant similarity model from the calculated correspondences. Furthermore, we replace the original camera response function (CRF) of the input images with a gamma curve. Specifically, we first transform all the frames into the linear HDR domain by applying inverse CRF, i.e., $\tilde{I}_i = f^{-1}(\tilde{Z}_i)/t_i$, where f is the CRF and t_i is the exposure time of frame i. We then use a gamma curve with $\gamma = 2.2$ to transfer the images from HDR to LDR domain $l_i(\tilde{I}_i)$:

$$Z_i = l_i(\tilde{I}_i) = \text{clip}[(\tilde{I}_i t_i)^{1/\gamma}], \tag{1}$$

where clip is a function that keeps the output in the range [0, 1] and l_i is a function that transfers the image $\tilde{I}_i$ from the linear HDR domain into LDR domain at exposure i (see Table 1).

Overall, the preprocessing step globally aligns $\tilde{Z}_{i-1}$ and $\tilde{Z}_{i+1}$ to the reference image, $\tilde{Z}_i$, and replaces the original CRF with a gamma curve to produce Z_{i-1}, Z_{i+1}, and Z_i. We use these processed images as the input to our system. We note that even if we omit the CRF replacement step, our system would require estimating the original CRF to transform the images from the LDR to the HDR domain in the merge step (Sec. 3.3). This is a requirement for almost all the previous approaches [KUWS03, MG10, KSB*13, LLM17]

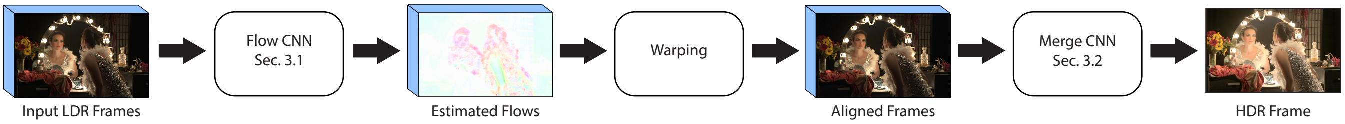

Figure 2: *We break down the HDR video reconstruction into two stages of alignment and HDR merge. To perform the alignment, we use the flow CNN to estimate a set of flows from the input frames. We then use the estimated flows to warp the neighboring frames and produce a set of aligned images. These images are then used by the merge CNN to produce the final HDR image.*

and is not a major limitation as the CRF can be easily estimated using Debevec and Malik's approach [DM97] from a series of images with different exposures. In the next sections, we discuss different components of our algorithm by starting with the flow network.

3.2. Flow Network

To reconstruct the missing content at frame i, we first need to align the neighboring frames to the reference frame. This requires estimating the flows from the frames, $i-1$ and $i+1$, to the reference frame, i. The estimated flows can then be used to warp the neighboring images, Z_{i-1} and Z_{i+1}, and produce a set of aligned images, $Z_{i-1,i}$ and $Z_{i+1,i}$. Note that, the neighboring images, Z_{i-1} and Z_{i+1}, are globally aligned to the reference image, Z_i, and thus, this process handles the non-rigid motion, possible parallax, and the potential inaccuracies of the global alignment.

Although there are powerful non-learning optical flow techniques [Liu09, XJM12, RWHS15, HLS17], we use CNNs to model the flow estimation process for a couple of reasons. First, CNNs are efficient and can be implemented on the GPU, and thus, they are significantly faster than the non-learning optimization-based optical flow methods. Second, the flow estimation is only one component of our system with the overall goal of producing high-quality HDR videos. By training our system in an end-to-end fashion, the flow estimation is optimized to maximize the quality of the HDR videos. Therefore, our flow estimation network is better suited for the HDR video reconstruction application than the existing flow estimation techniques, as shown in Fig. 3.

Recently, learning-based image transformation has been proposed for a variety of applications like image classification [JSZK15] and single image view synthesis [ZTS*16]. Specifically, several methods have proposed to perform optical flow estimation using deep networks [DFI*15, IMS*17, RB17]. These approaches are fast and can be optimized in combination with our merge network (Sec. 3.3) to minimize the error between ground truth and estimated HDR frames, and thus, do not have the aforementioned problems of the non-learning optical flow approaches. However, they use two input images to estimate the flow between them, and thus, are not suitable for HDR Video reconstruction, as shown in Fig. 4. In our application, the reference image often has missing content (e.g., because of noise in Fig. 4), and thus, estimating an accurate flow from each neighboring frame to the reference frame using only two input images is difficult.

To avoid this problem, we use the reference, Z_i, and the neighboring frames, Z_{i-1} and Z_{i+1}, as the input to our system. In this case, in regions where the reference image has missing content, the neighboring images can be used to estimate the appropriate flows. However, since the input frames are captured with alternating exposures, the reference and neighboring frames have different exposure times and, consequently, different brightness. We address this issue by adjusting the exposure of the reference frame to match that of the neighboring frames $g_{i+1}(Z_i)$:

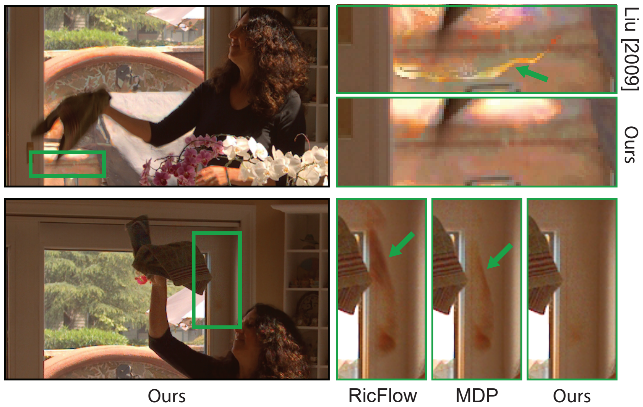

Figure 3: *We compare our flow network against the optical flow methods of Liu [Liu09] (top), Xu et al. [XJM12] (MDP), and Hu et al. [HLS17] (RicFlow) (bottom) by generating an HDR frame from the* THROWING TOWEL 2EXP *scene. We use the aligned images generated by the three optical flow approaches as the input to our merge network (Sec. 3.3) to produce the final HDR images and compare their results to our full approach. These methods are not designed for HDR video reconstruction, often producing significant alignment artifacts that cannot be masked by our merge network. Therefore, their final HDR frames usually contain tearing and other artifacts as indicated by the green arrows. On the other hand, our flow network has been trained to maximize the quality of the final HDR videos, and thus, our method produces HDR frames with higher-quality. Note that, our flow estimation network is faster than these traditional approaches.*

$$g_{i+1}(Z_i) = l_{i+1}\left(h(Z_i)\right) \tag{2}$$

where $h(Z_i)$ is a function that takes the image Z_i from the LDR domain to the linear HDR domain and is defined as:

$$h(Z_i) = Z_i^{\gamma}/t_i. \tag{3}$$

The input is then obtained by concatenating the exposure adjusted reference image as well as the two neighboring frames (9 channels), i.e., $\{g_{i+1}(Z_i), Z_{i-1}, Z_{i+1}\}$. The network takes this input and produces an output with 4 channels, consisting of two sets of flows from the previous, $i-1$, and next, $i+1$, frames to the reference frame, i, in x and y directions. These flows are then used to warp the neighboring images to produce a set of aligned images. Note that the inputs and outputs of the flow network are slightly different for the cases with three exposures, as discussed in Sec. 3.4.

For the flow network, we build upon the hierarchical coarse-to-fine architecture, concurrently proposed by Ranjan and Black [RB17] and Wang et al. [WZK*17], and incorporate the three

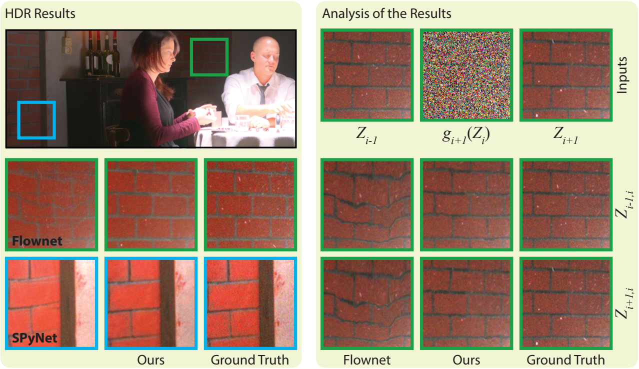

Figure 4: *On the left, we compare our flow network against the FlowNet [IMS*17] and SPyNet [RB17] by producing an HDR frame from the* Poker Fullshot *scene. Note that, we trained both the FlowNet and SPyNet networks in combination with our merge network (Sec. 3.3) to have a fair comparison. The other two networks are not able to register the bricks in the background producing ghosting artifacts, while our approach generates comparable results to the ground truth, as shown on the left. On the right, we analyze the results for the FlowNet (green) inset by showing the inputs to both FlowNet and our flow network on the top. The analysis for SPyNet is similar, but we omit it for brevity. FlowNet takes two images (e.g., Z_{i-1} and $g_{i+1}(Z_i)$) as the input and obtains the flow between them. In this case, since the reference image, $g_{i+1}(Z_i)$, has severe noise, obtaining an accurate flow is difficult. Therefore, as shown on the bottom two rows, the aligned images ($Z_{i-1,i}$ and $Z_{i+1,i}$) using the FlowNet contain artifacts. On the other hand, our flow network takes all the three images as the input, and thus, can use the information in the previous and next frames to produce more accurate flows. As a result, our method produces aligned images with higher quality that resemble the ground truth aligned images. Note that the brightness of the insets are adjusted for the best visibility. See the full videos in supplementary video.*

inputs into the architecture, as shown in Fig. 5. Our system consists of a series of flow estimator CNNs working at different resolutions. The estimated flows at the coarser scales capture the large motions and are used to initialize the inputs for the CNN in the finer scales, which are responsible for estimating the smaller motions.

In our system, we first generate a pyramid of our three input images by downsampling them using factors of 16, 8, 4, and 2. The three images at different resolutions are used as the input to their corresponding scale. At the coarsest scale (five in Fig. 5), we simply use the input images at that resolution to produce two sets of flows. These flows are then upsampled and used to warp the two neighboring images. The warped neighboring images as well as the reference image are then used as the CNN's input to produce two sets of flow at this finer resolution. Note that, the estimated flows are computed between the warped neighboring images and the reference image. Therefore, the full flow is obtained by adding the upsampled flow from the previous scale and the estimated flows at this scale. This process is repeated until reaching the finest scale and producing the final flows. The calculated flows are then used to warp the neighboring images and produce a set of aligned images, $Z_{i-1,i}$ and $Z_{i+1,i}$. These images are used by the merge network to produce the final result, as discussed in the next section.

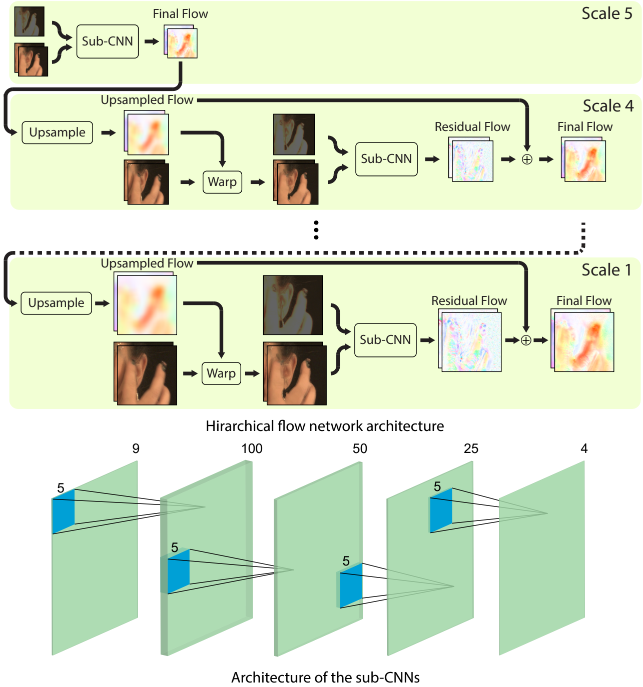

Figure 5: *We show our hierarchical coarse-to-fine flow network architecture on the top. At the coarsest level, the sub-CNN simply takes the input images and estimates two sets of flows. In all the other scales, we first upsample the flow from previous scale and then use it to warp the neighboring images. The warped images as well as the reference image are used to estimate the flows. These estimated flows are then added to the upsampled flows from the previous scale to produce the final flows in that scale. Note that, the figure is only for illustration of the architecture and the resolution of the images in different scales is not accurate. On the bottom, we demonstrate the architecture of the sub-CNNs used in our flow network. Our network consists of four convolutional layers with kernel size of 5. Each layer is followed by a rectified linear unit (ReLU), except for the last one, which has a linear activation function.*

Note that the flow network is essential for producing high-quality results by correcting non-rigid motions in the neighboring frames. Without this component, the regions with motion in the neighboring frames cannot be properly used to reconstruct the final HDR frame. In these areas, the merge network would either rely on the reference image or combine the misaligned images and produce noisy or ghosted results, as shown in Fig. 6.

3.3. Merge Network

The goal of this network is to produce a high-quality HDR frame from the aligned and reference images. Since the registered images contain residual alignment artifacts, this network should basically detect these artifacts and exclude them from the final HDR image. Recently, Kalantari and Ramamoorthi [KR17] demonstrated that this challenging problem can be effectively addressed by CNNs. Here, we also use a CNN to produce HDR images from a set of LDR inputs, but propose two simple, yet necessary changes in terms of input and architecture to improve the quality of the results.

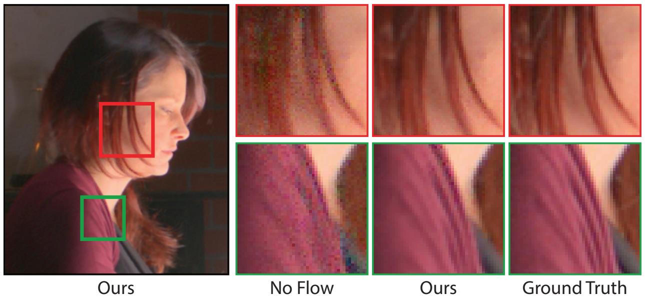

Figure 6: *We compare our full approach to our method without the flow network on the* POKER FULLSHOT *scene. The result without the flow network has artifacts in regions with non-rigid motion.*

Input/Output: Kalantari and Ramamoorthi only used the aligned images, including the reference image, as the input to the network. By adapting this strategy to HDR video, we can provide the two aligned neighboring images, $Z_{i-1,i}$ and $Z_{i+1,i}$, as well as the reference image, Z_i, to the network to produce the final HDR image. However, in some cases both aligned images contain artifacts around the motion boundaries, which would appear in the resulting HDR image (see Fig. 7).

We observe that these artifacts in most cases happen on the background regions. However, these areas are usually well-aligned in the original neighboring images. Therefore, in addition to the three images, we also use the neighboring images in our system, i.e., $\{Z_i, Z_{i-1,i}, Z_{i+1,i}, Z_{i-1}, Z_{i+1}\}$. These additional inputs greatly help the merge network to produce high-quality results, as shown in Fig. 7. Note that in some cases the artifacts appear on the moving subjects. However, these areas have complex motions, and thus, the artifacts are usually not noticeable in a video.

We provide the five images in both the LDR and linear HDR domains as the input to the network (30 channels). Our network then estimates the blending weights for these five images (15 channels output). We estimate a blending weight for each color channel, similar to the existing techniques [DM97, KSB*13], to properly utilize the information in each channel. The final HDR image at frame i, H_i, is computed as a weighted average of the five input images using their blending weights as:

$$H_i = \frac{w_1 I_i + w_2 I_{i-1,i} + w_3 I_{i+1,i} + w_4 I_{i-1} + w_5 I_{i+1}}{\sum_{k=1}^{5} w_k}. \qquad (4)$$

Here, w_k is the estimated blending weight for each image and $I_i = h(Z_i)$, where $h(Z_i)$ is the function that takes the image Z_i from the LDR to the linear HDR domain. Note that our system increases the dynamic range by directly combining the pixel values of the input and warped images and does not hallucinate content.

Architecture: Kalantari and Ramamoorthi [KR17] use a simple architecture with four convolutional layers for the merge network. Although their system is able to produce high-quality results, it is not able to mask the alignment artifacts in challenging cases (see Fig. 8). This is mainly because the receptive field of their network is small, and thus, their system detects the alignment artifacts by observing a small local region. However, in some cases the network needs to see a bigger region to properly distinguish the alignment artifacts from structures. Therefore, we propose to use an encoder-decoder architecture for modeling the HDR merge process. Specifically, we use a fully convolutional architecture with three downsampling (encoder) and upsampling (decoder) units, as shown in Fig. 9. Each downsampling unit consists of a *convolution* layer with stride of two, followed by another convolution layer with stride of one. The upsampling units consist of a *deconvolution* layer with stride of two, followed by a convolution layer with stride of one. We use a sigmoid as the activation function of the last layer, but all the other layers are followed by a ReLU.

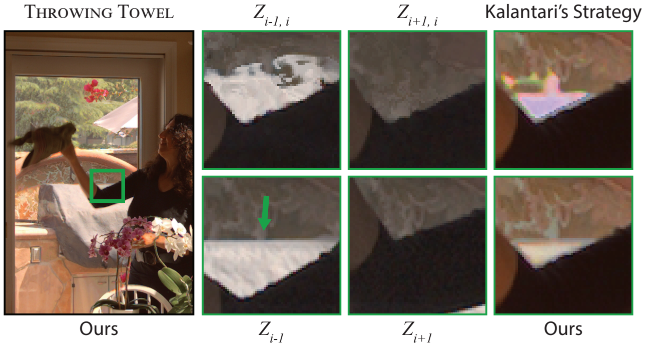

Figure 7: *Here, we compare our approach using five images as the input to the merge network against Kalantari's strategy using only three images. In both cases, we train the merge network in combination with the flow network on the HDR video data. The grill cover is saturated in the reference image and should be reconstructed from the neighboring images. As shown in the insets, both aligned neighboring images, $Z_{i-1,i}$ and $Z_{i+1,i}$, have registration artifacts on the grill cover. Therefore, using only the aligned images, as proposed by Kalantari and Ramamoorthi [KR17], we are not able to properly reconstruct the missing content. However, as indicated by the green arrow, the grill cover is artifact-free in one of the neighboring images, Z_{i-1}. Since we also pass the neighboring images as the input to our merge network, we are able to produce results with higher quality. Note that, the insets in the first two columns are in the LDR domain, while the last column shows the tonemapped HDR images.*

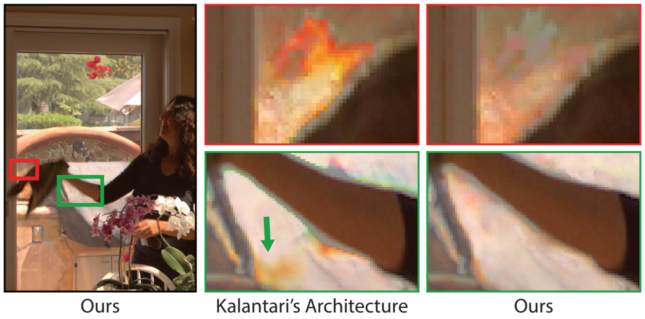

Figure 8: *Comparing to the network architecture, as proposed by Kalantari and Ramamoorthi [KR17], our encoder-decoder architecture produces results with fewer discoloration and objectionable artifacts. Note that, Kalantari's network is retrained on the HDR video data to have a fair comparison.*

3.4. Extension to Three Exposures

In this case, the input video alternates between three (low, medium, and high) exposures. For example, a sequence of Z_{i-2}, Z_{i-1}, Z_i, Z_{i+1}, and Z_{i+2} frames can have low, medium, high, low, and

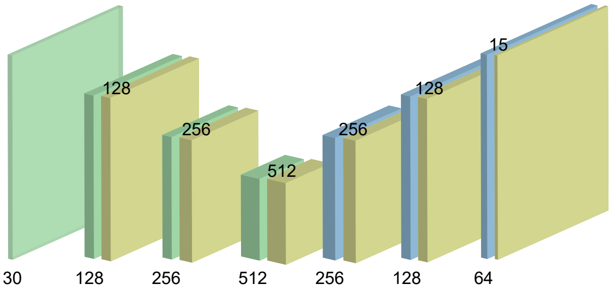

Figure 9: *We demonstrate the architecture of our merge network. The green and blue boxes refer to the convolution and deconvolution layers with stride of two and kernel size of four. These layers basically downsample (green) or upsample (blue) the feature maps by a factor of two. The layers indicated by yellow are simple convolutions with stride of one and kernel size of three. With the exception of the last layer, which has a sigmoid activation function, all the other layers are followed by a ReLU. The merge network takes five images (Sec. 3.3) in the LDR and linear HDR domains (30 channels) as the input and produces blending weights for these five images (15 channels).*

medium exposures, respectively. Here, our system utilizes four neighboring images in addition to the reference image to reconstruct a single HDR frame.

To adapt our system to this case, we simply adjust the inputs and outputs of the flow and merge CNNs. Specifically, our flow CNN takes Z_{i-2}, Z_{i+1}, and $g_{i+1}(Z_i)$, as well as Z_{i-1}, Z_{i+2}, and $g_{i+2}(Z_i)$ as the input. Here, $g_{i+1}(Z_i)$ and $g_{i+2}(Z_i)$ refer to the exposure adjusted versions of the reference image. Therefore, in total our flow network takes six images as the input (18 channels). The flow network then outputs four flows (8 channels), which are used to warp the four neighboring images to the reference image. These four aligned images $(Z_{i-2,i}, Z_{i-1,i}, Z_{i+1,i}, Z_{i+2,i})$ along with the original neighboring $(Z_{i-2}, Z_{i-1}, Z_{i+1}, Z_{i+2})$ and the reference image (Z_i) in both LDR and linear HDR domains (54 channels) are used as the input to the merge network to produce the final HDR frame.

4. Training

As with most machine learning approaches, our system consists of two main stages of training and testing. During training, which is an offline process, we find optimal weights of the networks through an optimization process. This requires 1) an appropriate metric to compare the estimated and ground truth HDR images and 2) a large number of training scenes. Once the training is done, we can use our trained networks to generate results on new test scenes. In the next sections, we discuss our choice of loss function and the dataset.

4.1. Loss Function

HDR images and videos are typically displayed after tonemapping, a process that generally boosts the pixel values in the dark regions. Therefore, defining the loss function directly in the linear HDR domain, underestimates the error in the dark areas. To avoid this problem we transfer the HDR images into the log domain, which is a common approach used by several recent algorithms [ZL17, EKD*17, BVM*17, KR17]. Specifically, we use the differentiable μ-law function for transferring the HDR images into the log domain:

$$T_i = \frac{\log(1+\mu H_i)}{\log(1+\mu)}, \tag{5}$$

where H_i is the HDR frame and is always in the range [0, 1] and μ (set to 5000) is a parameter controlling the rate of range compression. To train our system, we minimize the ℓ_1 distance between the estimated, $\hat{T}_i$, and ground truth, T_i, HDR frames in the log domain:

$$E = \|\hat{T}_i - T_i\|_1. \tag{6}$$

We chose ℓ_1 as we found it produces slightly sharper images than ℓ_2. Note that, we directly minimize this error to train both our flow and merge networks, and thus, do not need the ground truth flows for training. Since all the components of our system, including the warping, are differentiable, we can easily compute all the required gradients using the chain rule. These gradients are used to update the networks' weights iteratively until convergence.

4.2. Dataset

In order to train our system, we need a large number of training scenes consisting of three input LDR frames with alternating exposures (a reference frame and two neighboring frames) and their corresponding ground truth HDR frame. We construct our training set by selecting 21 scenes from two publicly available HDR video datasets by Froehlich et al. [FGE*14] (13 scenes) and Kronander et al. [KGB*14] (8 scenes). These datasets consists of hundreds of HDR frames for each scene, captured using cameras with specific optical designs containing external [FGE*14] or internal [KGB*14] beam-splitters.

To generate the training set from these datasets, we first select three consecutive frames from a scene and transform them to the LDR domain (see Eq. 1), using two different exposure times. In our system, we use these three images as the input and select the middle HDR frame to be used as the ground truth. We generate our datasets with exposures separated by one, two, and three stops, where the low exposure time is randomly selected around a base exposure. We augment the data by applying geometric transformations (rotating 90 degrees and flipping) on the training data.

Since this dataset is produced synthetically, a system trained on it would not work properly on scenes captured with off-the-shelf cameras. In practice, real world cameras capture noisy images and are also hard to calibrate. However, our synthetic dataset lacks these imperfections. To address this issue, we simulate the imperfections of standard cameras by adding noise and adjusting tone of the synthetic images. This simple approach is critical for making sure our system generalizes well to images, captured with standard cameras, that it has not been trained on. In the next sections, we discuss the noise and tone adjustment strategies as well as our mechanism for patch generation.

Adding Noise: The images captured with standard digital cameras typically contain noise in the dark regions. Therefore, to produce a high-quality HDR image, the information in the dark areas should be taken from the image with the high exposure. Unfortunately, since we generate the input LDR images synthetically, the images with different exposures contain the same amount of noise as their HDR counterparts. Therefore, if we train our system on this dataset, our merge network is not able to utilize the content of

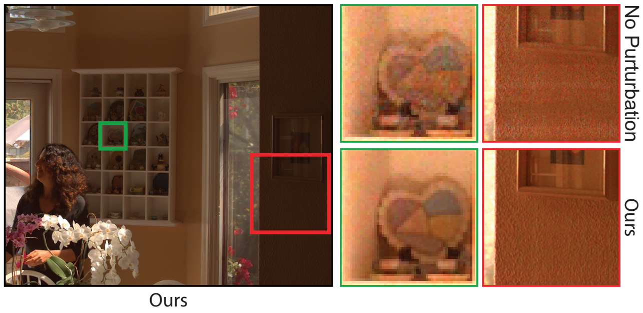

Figure 10: *We compare the result of our method without tone perturbation during training (Sec. 4.2) against our full approach on a frame from the* THROWING TOWEL 2EXP *scene. Our system without tone perturbation produces noisy results on real-world scenes. The brightness of the insets are adjusted for the best visibility.*

the high exposure image in the dark regions, often producing noisy results in real-world scenarios.

We address this problem by adding zero-mean Gaussian noise, a commonly-used image noise model [JD13, GCPD16], to the input LDR images with low exposure. This increases the robustness of the flow network and encourages the merge network to use the content of the clean high exposure image in the dark regions. Note that, we add the noise in the linear domain, and thus, the noise in the dark areas are typically magnified after transferring the image to the LDR domain. In our implementation, we randomly choose standard deviation between 10^{-3} and 3×10^{-3}, so our system learns to handle noise with different variances. Note that, while there are more complex noise models [HDF10, GKTT13, GAW*10], we found the simple Gaussian noise to be sufficient for our purpose.

Tone Perturbation: In practice, calibrating the cameras and finding the exact camera response function (CRF) is usually difficult. Therefore, the color and brightness of the neighboring images are often slightly different from those of the reference image even after exposure adjustment. However, our LDR images are extracted synthetically, and thus, are consistent. Therefore, training our system on this dataset limits the ability of both our flow and merge network to generalize to the scenes captured with standard cameras, as shown in Fig. 10.

To avoid this issue, we slightly perturb the tone of the reference image by independently applying a gamma function to its different color channels. Specifically, we apply gamma encoding with $\gamma = \exp(d)$, where d is randomly selected from the range [-0.7, 0.7]. We use this perturbed reference image as the input to our flow and merge networks, so they learn to handle the inconsistencies of the reference and neighboring images when estimating the flows and the blending weights. However, we use the original reference image (before tone perturbation) along with the neighboring images during the blending process (Eq. 4) to produce HDR images that match the ground truth. Note that since the neighboring frames have the same exposure, their color and brightness always match even when the estimated CRF is highly inaccurate. Therefore, we only apply the perturbation on the reference image. It is also worth noting that, we assume the CRF is known and this process only simulates the inaccuracies of the CRF estimation, which can be modeled using the simple gamma function. This is in contrast to the inverse tone mapping methods, such as the approach by Eilertsen et al. [EKD*17], that assume the CRF is unknown and, thus, need to properly model it.

As noted in Sec. 3, the two stage architecture is essential for this perturbation strategy to work. In the case of modeling the entire process with one network, the CNN takes the neighboring images, as well as the perturbed reference image and should produce the final HDR image. This requires the CNN to undo a random tone adjustment applied on the reference image, which is difficult. For the same reason, estimating the blending weights using the merge network is essential, and we cannot directly output the final HDR frame using this network.

Patch Generation: As is common with the deep learning systems, we break down the images into small overlapping patches of size 352×352 with a stride of 176 to be able to efficiently train the networks. Most patches in our dataset contain static backgrounds, which are not useful for training the flow network. Therefore, we only select a patch if the two neighboring images have more than 2000 pixels with absolute difference of 0.1 and more. Note that this strategy is not perfect, but it mostly selects the patches in the areas with motion and we found it to work well for our application. We select around 1000 patches for each scene, and thus, have a total of 22,000 patches in our training set.

5. Results

We implemented our approach in MATLAB and used MatConvNet [VL15] to efficiently implement our flow and merge networks. Although our flow architecture is similar to that of Wang et al. [WZK*17], since our inputs and outputs are different we are not able to use their pre-trained network. Therefore, we train both the flow and merge networks by initializing their weights using the Xavier approach [GB10]. We solve the optimization using ADAM with the default parameters, $\beta_1 = 0.9$ and $\beta_2 = 0.999$, and a learning rate of 10^{-4}. We use mini-batches of size 10 and perform training for 60,000 iterations, which takes roughly 5 days on a machine with an Intel Core i7, 64GB of memory, and a GeForce GTX 1080 GPU. All the results are tonemapped using the method of Reinhard et al. [RSSF02] with the modification to add temporal coherency, as proposed by Kang et al. [KUWS03]. Note that the same tonemapping approach was used by Kalantari et al. [KSB*13]. Here, we only show one or two frames from each video, but the full videos are available in the supplementary materials.

We compare our approach against the methods of Kang et al. [KUWS03], Mangiat and Gibson [MG11], Kalantari et al. [KSB*13], and Li et al. [LLM17]. We implemented the method of Kang et al. [KUWS03] and used the publicly available source code for the approaches by Kalantari et al. [KSB*13] and Li et al. [LLM17]. For Mangiat and Gibson's approach, the authors provided their results on only three scenes, which we compare against in Figs. 14 and 17. Moreover, Li et al.'s approach takes roughly 2 hours to produce a single frame with a resolution of 1280×720, and thus, producing the videos for all the scenes was difficult. Therefore, we only compare against this approach on two scenes in Fig. 14 and supplementary video. Finally, as discussed in Sec. 1, the approach by Kalantari and Ramamoorthi [KR17] always assumes the reference is the image with medium exposure, and thus,

Input	2 Exposures			3 Exposures		
Method	Kang	Kalantari	Ours	Kang	Kalantari	Ours
PSNR	38.06	38.77	**40.67**	35.20	35.93	**39.37**
HDR-VDP-2	65.88	62.12	**74.15**	61.09	62.64	**68.86**
HDR-VQM	79.95	83.41	**85.51**	73.99	74.15	**82.14**

Table 2: *Quantitative comparison of our system against the approaches by Kang et al. [KUWS03] and Kalantari et al. [KSB*13]. Note that, the PSNR (db) values are computed on the images that are tonemapped using Eq. 5, but the results shown throughout the paper are tonemapped using the method of Reinhard et al. [RSSF02]. The values are averaged over all the frames of the four videos.*

	PSNR	HDR-VDP-2	HDR-VQM
RicFlow	36.97	63.39	78.52
MDP	36.31	64.23	77.22
Liu	**38.06**	**65.88**	**79.95**

Table 3: *Quantitative comparison of the method of Kang et al. [KUWS03] using different optical flow approaches.*

a direct comparison to this method is not possible. Nonetheless, we adapt this method to our application and demonstrate that we significantly improve their flow estimation (see Fig. 3) and merge network (Figs. 7 and 8).

Comparison on Scenes with Ground Truth: We begin by quantitatively comparing our results against the methods of Kang et al. [KUWS03] and Kalantari et al. [KSB*13] on four videos from Froehlich et al. [FGE*14]. Specifically, we select three seconds of the CAROUSEL FIREWORKS, FISHING LONGSHOT, POKER FULLSHOT, and POKER TRAVELLING SLOWMOTION scenes, none of which were included in the training set. For each video, we generate input LDR videos with two and three alternating exposures using the approach described in Sec. 4.2 and by adding noise to represent the real videos more closely. The two exposure inputs have a three stop separation, while the inputs with three alternating exposures are separated by two stops. These videos have a resolution of 1920 × 1080, but have a 10 pixel wide black border around them, which we crop for quantitative comparison.

As shown in Table 2, we evaluate the results in terms of PSNR in the tonemapped domain (using Eq. 5). We also include HDR-VDP-2 [MKRH11] and HDR-VQM [NSC15] values, metrics specifically designed for evaluating the quality of HDR images and videos, respectively. All the values are computed for each individual frame and averaged over all the frames of the four video sequences. As seen, our method produces better results than the other methods even in the challenging cases with three alternating exposures.

Note that, the results of Kang et al. are obtained using Liu's optical flow method [Liu09]. We also use the approaches by Xu et al. [XJM12] (MDP) and Hu et al. [HLS17] (RicFlow) and report the results in Table 3. Overall, although MDP and RicFlow rank high on Middleburry and Sintel benchmarks, they are slightly worse than the approach of Liu [Liu09] for this application. Therefore, we use Liu's method to produce the results of Kang et al.'s approach.

We show individual frames for two of these scenes in Fig. 11. Here, the top row shows the POKER FULLSHOT scene with two exposures. This video demonstrates people playing cards on a table illuminated with a bright light. Kang et al.'s approach [KUWS03] uses optical flow to register the neighboring frames, and thus, produces artifacts in the regions with significant motion such as the lady's hands. The patch-based method of Kalantari et al. [KSB*13] underestimates the patch search window in the regions with small motion (e.g., the lady's arms and chest) and produces ghosting artifacts. Our method, on the other hand, produces high-quality results that closely resemble the ground truth.

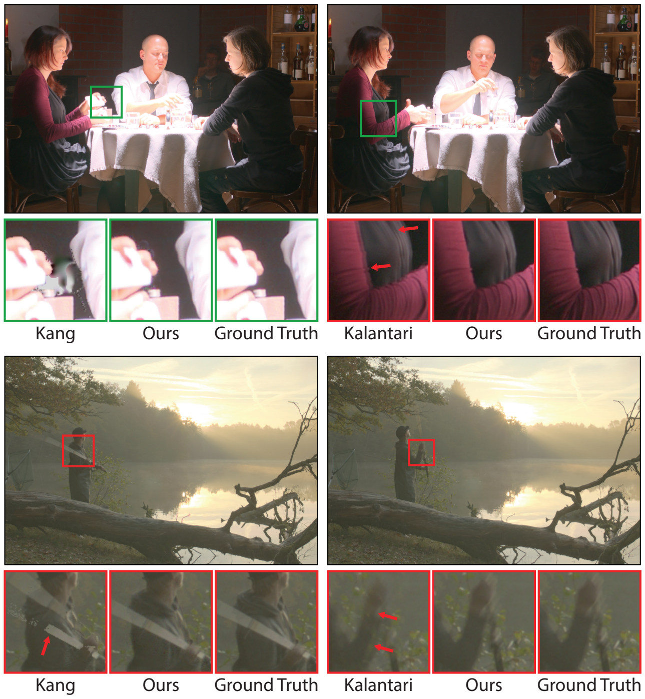

Figure 11: *Comparison against the methods of Kang et al. [KUWS03] and Kalantari et al. [KSB*13] on sequences with ground truth HDR video. The scene on the top has two alternating exposures with three stop separations and the one on the bottom has three exposures separated by two stops.*

The bottom row demonstrates the challenging FISHING LONGSHOT scene (bottom row) with three exposures, exhibiting significant motion on the man's hand and the fishing rod. Both Kang et al. and Kalantari et al.'s approaches are unable to produce satisfactory results on the fast moving areas, producing tearing and ghosting artifacts. Therefore, the videos generated by these two approaches contain jittery motion on the fast moving fishing rod. Moreover, Kalantari et al.'s approach is not able to properly constrain the patch search on the slow moving areas (tree leaves), generating wobbly and unnatural motion, which can be seen in the supplementary video. However, our approach properly handles both the fast and slow moving areas and generates a high-quality HDR video.

To demonstrate that our approach consistently produces better results than the other approaches, we plot the HDR-VDP-2 scores for all the frames of the POKER FULLSHOT scene in Fig. 12. As seen, our method has better scores than the two approaches by Kang et al. and Kalantari et al. in most cases, two of which are shown in Fig. 11 (top). However, in a few frames our method produces slightly worse results, as shown at the bottom insets of Fig. 12. Note that, all the approaches have different performance on the odd

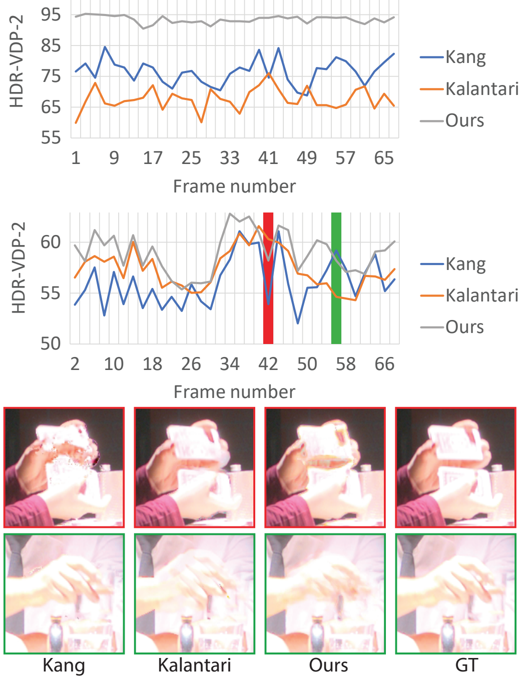

Figure 12: *We show the HDR-VDP-2 [MKRH11] scores for different frames of the* POKER FULLSHOT *scene with two alternating exposures on the top. We separate the odd and even frames, since their scores have different ranges. The reference image has low exposure in the odd frames, while it has high exposure in the even frames. Our method produces better results than other approaches in almost all the frames. On the bottom, we compare the results visually for two frames, indicated by red and green bars on the plots, in which our method is slightly worse than the other methods. Specifically, the red inset shows a frame where our method produces slightly lower HDR-VDP-2 values than the method of Kalantari et al. The green inset demonstrates a frame where our method is slightly worse than the method of Kang et al. and produces a blurry hand.*

and even frames, where the reference has low and high exposures, respectively. This is mainly because the frames with high exposure lack contents in the bright areas, while the frames with low exposure are noisy in the dark areas. Therefore, reconstructing the even frames is typically more challenging than the odd frames.

Comparison on Kalantari et al.'s Scenes: We compare our approach against other methods on several scenes from Kalantari et al. [KSB*13] with two (Figs. 13 and 14) and three (Fig. 15) exposures. Note that these scenes have been captured with the off-the-shelf Basler acA2000-50gc camera with the ability to alternate between different exposures. These results demonstrate the ability of our approach to generalize to real-world cases, since our approach has not been trained on the videos from this camera.

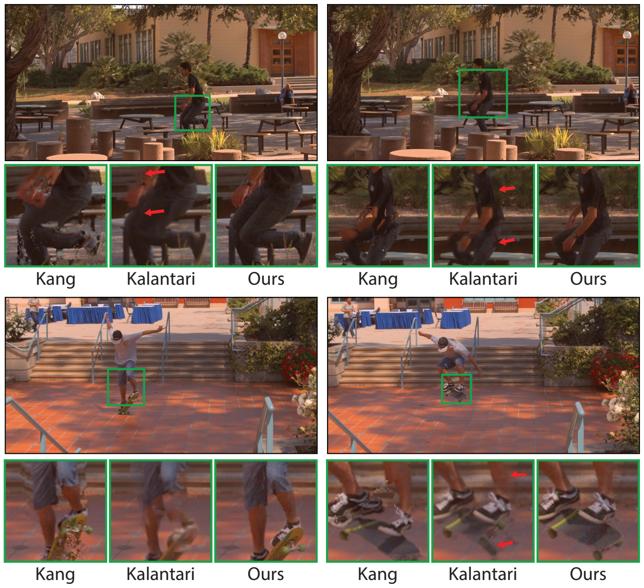

Figure 13: *Comparison against the methods of Kang et al. [KUWS03] and Kalantari et al. [KSB*13] on scenes with two alternating exposures separated by three stops.*

Figure 13 compares our approach against Kang et al. [KUWS03] and Kalantari et al.'s approaches [KSB*13] on the NINJA (top) and SKATEBOARDER (bottom) scenes. The fast moving person in the NINJA scene is challenging for the other approaches. The method of Kang et al. produces tearing artifacts on the arms and legs of the moving person, while Kalantari et al.'s approach produces results with ghosting and blurring artifacts. However, our approach produces high-quality results without objectionable artifacts.

Similarly, the SKATEBOARDER scene contains a fast moving person on a bright day. Again, Kang et al.'s approach produces results with tearing artifacts due to the inability of the optical flow to properly register the neighboring frames. The patch-based method of Kalantari et al. is not able to reconstruct the fast moving legs and skateboard, producing results with ghosting and blurring artifacts. Moreover, Kalantari et al.'s approach underestimates the patch window search on the moving lady (top left) in the bright regions, producing jittery motion, which can be seen in the supplementary video. However, our method produces high-quality results and is significantly faster than the other techniques (see Table 4).

We compare our method against the approaches by Li et al. [LLM17] and Mangiat and Gibson [MG11] in Fig. 14. Li et al.'s method produces results with significant noise and discoloration for the FIRE scene and ghosting artifacts on the fast moving areas for the THROWING TOWEL 2EXP scene. On the other hand, Mangiat and Gibson's method uses a block-based motion estimation approach, and thus, their results suffer from blocking artifacts. Moreover, in some cases they filter the image to hide the blocking artifacts and produce blurry results (blue inset in bottom row).

Finally, we compare our approach against the methods of Kang et al. and Kalantari et al. on challenging scenes with three exposures separated by two stops. Figure 15 shows the result of this comparison for the CLEANING (top) and THROWING TOWEL 3EXP (bottom) scenes. The CLEANING scene shows a lady cleaning a table while the camera rotates around her. Kang et al.'s approach is not able to properly handle the fast moving arms and produces results with tearing artifacts. Kalantari et al.'s method produces comparable results to ours in the fast moving regions, despite having artifacts on the lady's arm and producing blurry cleaning cloth. However, their approach is unable to perform well in the very

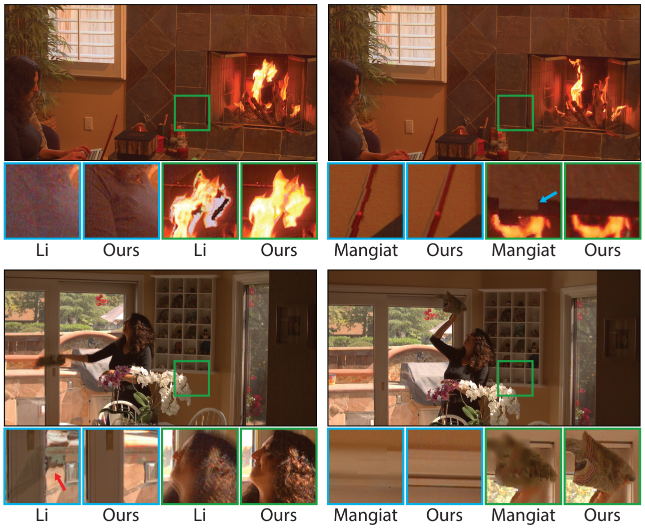

Figure 14: *Comparison with Li et al. [LLM17] and Mangiat and Gibson's approaches [MG11].*

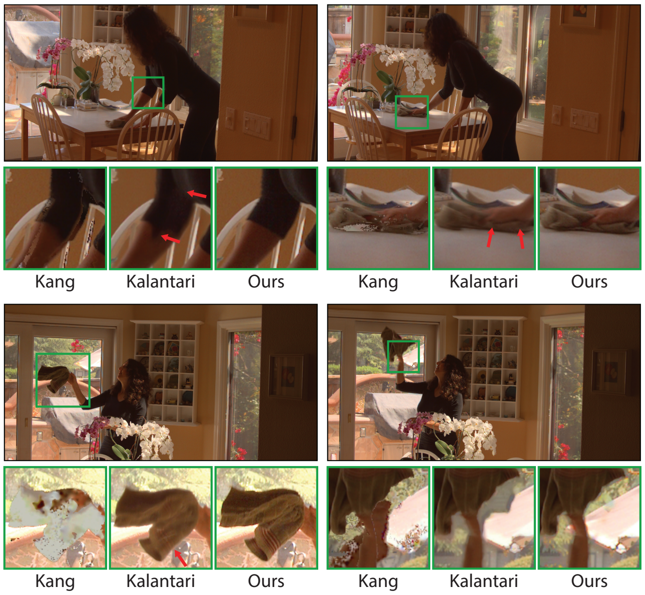

Figure 15: *Comparison against other approaches on scenes with three exposures separated by two stops.*

dark regions (lady's hair) producing unnatural motion (see supplementary video).

The THROWING TOWEL 3EXP scene contains a lady throwing a towel in front of a bright window. This scene is particularly challenging because of the significantly fast and non-rigid motion of the towel. Since optical flow is not able to handle fast moving objects, Kang et al.'s approach produces results with severe artifacts. Although Kalantari et al.'s method slightly overblurs the towel and produces results with artifacts on the umbrella, it is overall comparable to our method in the fast moving areas. However, their method produces ghosting artifacts on the lady's shoulder and wobbliness on the white flowers, which can be seen in the supplementary video.

Comparing to Burst Imaging: We also compare our approach using sequences with varying exposures against the alternative method of denoising a sequence of frames with the same short exposure in the supplementary video. We use the V-BM4D method [MBFE12], which is specifically designed for denoising videos. However, this approach is not able to properly remove the significant noise in the sequences with low exposures. In comparison, our method can utilize the detail in the frames with high exposure to produce a high-quality HDR video.

Input	2 Exposures			3 Exposures		
Method	Kang	Kalantari	Ours	Kang	Kalantari	Ours
1920 × 1080	195s	300s	**3.1**s	370s	520s	**4.6**s
1280 × 720	70s	125s	**1.4**s	135s	185s	**2.2**s

Table 4: *Timing comparison with the methods of Kang et al. [KUWS03] and Kalantari et al. [KSB*13] on inputs with two and three exposures and different resolutions. Overall, our approach is between 50 to 110 times faster than the other techniques.*

Timing: We provide timing comparison in Table 4. On average, our approach produces a single frame with resolution of 1920 × 1080 in 3.1 and 4.6 seconds for the inputs with two and three exposures, respectively. Comparing to the methods of Kang et al. and Kalantari et al. for inputs with two exposures, our approach is roughly 60 and 100 times faster, respectively. The speed up increases to roughly 80 and 110 times comparing to the approaches by Kang et al. and Kalantari et al. for the inputs with three exposures. Our approach is also significantly faster than the two other methods for inputs with resolutions of 1280 × 720. For input images with two exposures at this resolution, our method takes roughly 1.4 seconds to generate a single frame, spending 0.4 second for global alignment, 0.57 second to generate the flows, and 0.43 second to merge the aligned images into the final HDR frame.

Note that, we use the optical flow method of Liu [Liu09] for Kang et al.'s approach to achieve the best quality (see Table 3). Using faster optical flow methods, like RicFlow [HLS17], the timing reduces to roughly 50 seconds for two exposure inputs with resolution of 1920 × 1080 at the cost of sacrificing the quality. However, even in this case, our approach is an order of magnitude faster than Kang et al.'s method. Moreover, Mangiat and Gibson's approach [MG11] takes roughly 40 seconds to produce a single frame with resolution of 1280 × 720 for the inputs with two exposures, which is almost 30 times slower than our technique.

Limitations: HDR video reconstruction from sequences with alternating exposure is a notoriously challenging problem. Although our method produces results with better quality than the state-of-the-art approaches, in some cases it is not able to produce satisfactory results. For example, in cases where the reference image is over-exposed and there is significant parallax and occlusion, our flow network is not able to properly register the neighboring frames and our method produces results with ghosting and other artifacts, as shown in Fig. 16. However, these areas are challenging for other approaches as well and they produce results with similar artifacts.

Moreover, in cases where the reference image has low exposure and there is complex motion, as shown in Fig. 17 (left), the flow network is not able to properly align the images. Therefore, in these regions, our merge network relies on the reference and produces a slightly noisy image. However, our result is considerably better than the other techniques. Furthermore, in rare cases,

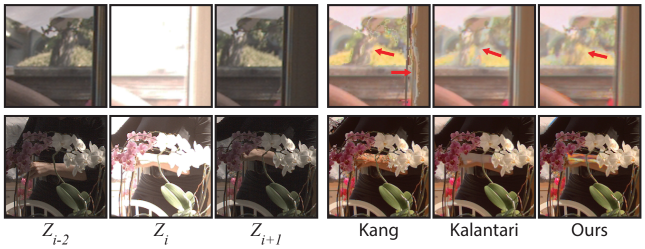

Figure 16: *The top and bottom rows show an inset from the* Cleaning *and* Throwing Towel 3Exp *scenes, respectively. In both cases, the reference frame, Z_i, is over-exposed and the missing content should be recovered from the neighboring frames with low exposure, Z_{i-2} and Z_{i+1}. Note that since these sequences have three alternating exposures, the previous neighboring frame with low exposure is Z_{i-2}. Moreover, we have excluded the frames with medium exposure for clarity of exposition, but they are used in our system to generate the final results. Because of significant parallax in the top inset, none of the methods are able to properly register the images, producing results with ghosting artifacts. Moreover, the lady's hand at the bottom inset has significant motion and is being occluded by the flower. Therefore, our method similar to other approaches contains artifacts in this region. Zoom in to the electronic version to see the differences.*

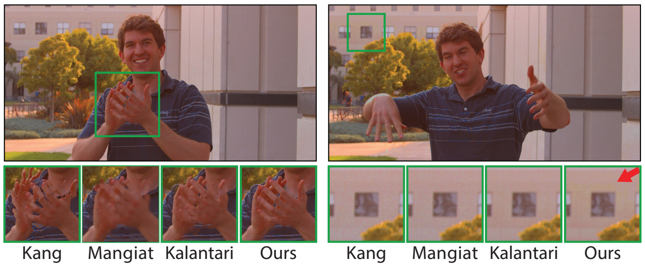

Figure 17: *On the left, we demonstrate a case with complex motion. Our approach fails to align the neighboring images, and thus, heavily relies on the reference image, producing a slightly noisy image. However, our result is significantly better relative to other approaches. On the right, we show a case where our system produces a result with slight discoloration. The differences are best seen by zooming into the electronic version.*

our approach produces results with slight discoloration, as shown in Fig. 17 (right). This discoloration, which is not noticeable on the still image, results in slight flickering in the video (see supplementary video). This is due to the fact that we define our loss function on *individual* frames. We leave the investigation of the possibility of using perceptual error metrics on videos to future work.

Finally, our system is limited to work with a fixed number of exposures and requires re-training to handle a different number of exposures. However, this is not a major limitation as we demonstrate our results using sequences with two and three exposures, covering the majority of the cases.

6. Conclusion and Future Work

We have presented the first learning-based technique for reconstructing HDR videos from sequences with alternating exposures. We divide the entire process into two stages of alignment and HDR merge and model them with two sequential CNNs. We then train both networks in an end-to-end fashion by minimizing the ℓ_1 distance between the estimated and ground truth HDR images. We produce our training set from publicly available HDR video datasets by simulating the imperfections of standard digital cameras. We demonstrate that our method produces better results than state-of-the-art approaches, while it is an order of magnitude faster.

In the future, we would like to investigate the possibility of designing a system that is independent of the number of exposures. Moreover, it would be interesting to adapt our system to other capturing configurations, e.g., stereo cameras with different exposures. We would also like to experiment with the architecture of the networks to increase the efficiency of our approach and reduce timings to interactive or real-time rates.

Acknowledgements

Funding for this work was provided in part by ONR grant N000141712687, NSF grant 1617234, and the UC San Diego Center for Visual Computing.

References

[BVM*17] Bako S., Vogels T., Mcwilliams B., Meyer M., Novák J., Harvill A., Sen P., Derose T., Rousselle F.: Kernel-predicting convolutional networks for denoising monte carlo renderings. *ACM TOG 36*, 4 (July 2017), 97:1–97:14. 7

[CBK17] Choi I., Baek S., Kim M. H.: Reconstructing interlaced high-dynamic-range video using joint learning. *TIP 26*, 11 (Nov 2017), 5353–5366. 2

[DFI*15] Dosovitskiy A., Fischery P., Ilg E., Häusser P., Hazirbas C., Golkov V., v. d. Smagt P., Cremers D., Brox T.: Flownet: Learning optical flow with convolutional networks. In *IEEE ICCV* (Dec 2015), pp. 2758–2766. 1, 4

[DM97] Debevec P. E., Malik J.: Recovering high dynamic range radiance maps from photographs. In *ACM SIGGRAPH* (1997), pp. 369–378. 1, 2, 3, 6

[EKD*17] Eilertsen G., Kronander J., Denes G., Mantiuk R. K., Unger J.: Hdr image reconstruction from a single exposure using deep cnns. *ACM Trans. Graph. 36*, 6 (Nov. 2017), 178:1–178:15. 2, 7, 8

[EKM17] Endo Y., Kanamori Y., Mitani J.: Deep reverse tone mapping. *ACM TOG 36*, 6 (Nov. 2017), 177:1–177:10. 2

[FGE*14] Froehlich J., Grandinetti S., Eberhardt B., Walter S., Schilling A., Brendel H.: Creating cinematic wide gamut HDR-video for the evaluation of tone mapping operators and HDR-displays. *SPIE 9023* (2014), 90230X–90230X–10. 2, 7, 8

[GAW*10] Granados M., Ajdin B., Wand M., Theobalt C., Seidel H., Lensch H. P. A.: Optimal hdr reconstruction with linear digital cameras. In *CVPR* (June 2010), pp. 215–222. 8

[GB10] Glorot X., Bengio Y.: Understanding the difficulty of training deep feedforward neural networks. In *AISTATS* (May 2010), vol. 9, pp. 249–256. 8

[GCPD16] Gharbi M., Chaurasia G., Paris S., Durand F.: Deep joint demosaicking and denoising. *ACM TOG 35*, 6 (Nov. 2016), 191:1–191:12. 7

[GKTT13] Granados M., Kim K. I., Tompkin J., Theobalt C.: Automatic noise modeling for ghost-free HDR reconstruction. *ACM TOG 32*, 6 (2013), 201:1–201:10. 8

[GPR*15] Gryaditskaya Y., Pouli T., Reinhard E., Myszkowski K., Seidel H.-P.: Motion Aware Exposure Bracketing for HDR Video. *CGF* (2015). 2

[HDF10] Hasinoff S. W., Durand F., Freeman W. T.: Noise-optimal capture for high dynamic range photography. In *CVPR* (June 2010), pp. 553–560. 8

[HGPS13] Hu J., Gallo O., Pulli K., Sun X.: HDR deghosting: How to deal with saturation? In *IEEE CVPR* (June 2013), pp. 1163–1170. 1, 2

[HKU15] Hajisharif S., Kronander J., Unger J.: Adaptive dualiso hdr reconstruction. *EURASIP Journal on Image and Video Processing 2015*, 1 (2015), 41. 2

[HLS17] Hu Y., Li Y., Song R.: Robust interpolation of correspondences for large displacement optical flow. In *CVPR* (July 2017), pp. 4791–4799. 4, 9, 11

[HSG*16] Hasinoff S. W., Sharlet D., Geiss R., Adams A., Barron J. T., Kainz F., Chen J., Levoy M.: Burst photography for high dynamic range and low-light imaging on mobile cameras. *ACM TOG 35*, 6 (Nov. 2016), 192:1–192:12. 1, 2

[HST*14] Heide F., Steinberger M., Tsai Y.-T., Rouf M., Pająk D., Reddy D., Gallo O., Liu J., Heidrich W., Egiazarian K., Kautz J., Pulli K.: Flexisp: A flexible camera image processing framework. *ACM TOG 33*, 6 (Nov. 2014), 231:1–231:13. 2

[IMS*17] Ilg E., Mayer N., Saikia T., Keuper M., Dosovitskiy A., Brox T.: Flownet 2.0: Evolution of optical flow estimation with deep networks. In *IEEE CVPR* (Jul 2017). 1, 4

[JD13] Jeon G., Dubois E.: Demosaicking of noisy bayer-sampled color images with least-squares luma-chroma demultiplexing and noise level estimation. *IEEE Transactions on Image Processing 22*, 1 (Jan 2013), 146–156. 7

[JSZK15] Jaderberg M., Simonyan K., Zisserman A., Kavukcuoglu K.: Spatial transformer networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2* (Cambridge, MA, USA, 2015), NIPS, MIT Press, pp. 2017–2025. 4

[KGB*14] Kronander J., Gustavson S., Bonnet G., Ynnerman A., Unger J.: A unified framework for multi-sensor hdr video reconstruction. *Signal Processing: Image Communication 29*, 2 (2014), 203 – 215. Special Issue on Advances in High Dynamic Range Video Research. 2, 7

[KR17] Kalantari N. K., Ramamoorthi R.: Deep high dynamic range imaging of dynamic scenes. *ACM TOG 36*, 4 (2017). 1, 2, 3, 5, 6, 7, 8

[KSB*13] Kalantari N. K., Shechtman E., Barnes C., Darabi S., Goldman D. B., Sen P.: Patch-based high dynamic range video. *ACM TOG 32*, 6 (Nov. 2013), 202:1–202:8. 1, 2, 3, 6, 8, 9, 10, 11

[KUWS03] Kang S. B., Uyttendaele M., Winder S., Szeliski R.: High dynamic range video. *ACM TOG 22*, 3 (2003), 319–325. 1, 2, 3, 8, 9, 10, 11

[Liu09] Liu C.: *Beyond Pixels: Exploring New Representations and Applications for Motion Analysis*. Doctoral thesis, Massachusetts Institute of Technology, May 2009. 1, 4, 9, 11

[LLM17] Li Y., Lee C., Monga V.: A maximum a posteriori estimation framework for robust high dynamic range video synthesis. *IEEE Transactions on Image Processing 26*, 3 (March 2017), 1143–1157. 1, 2, 3, 8, 10

[LYT*14] Liu Z., Yuan L., Tang X., Uyttendaele M., Sun J.: Fast burst images denoising. *ACM TOG 33*, 6 (Nov. 2014), 232:1–232:9. 1, 2

[MBFE12] Maggioni M., Boracchi G., Foi A., Egiazarian K.: Video denoising, deblocking, and enhancement through separable 4-d nonlocal spatiotemporal transforms. *IEEE TIP 21*, 9 (Sept 2012), 3952–3966. 11

[MBRHD18] Marnerides D., Bashford-Rogers T., Hatchett J., Debattista K.: Expandnet: A deep convolutional neural network for high dynamic range expansion from low dynamic range content. *Computer Graphics Forum 37*, 2 (2018), 37–49. 2

[MG10] Mangiat S., Gibson J.: High dynamic range video with ghost removal. In *Proc. SPIE 7798* (2010), no. 779812, pp. 1–8. 2, 3

[MG11] Mangiat S., Gibson J.: Spatially adaptive filtering for registration artifact removal in HDR video. In *ICIP 2011* (Sept 2011), pp. 1317–1320. 1, 2, 8, 10, 11

[MKRH11] Mantiuk R., Kim K. J., Rempel A. G., Heidrich W.: Hdr-vdp-2: A calibrated visual metric for visibility and quality predictions in all luminance conditions. *ACM TOG 30*, 4 (July 2011), 40:1–40:14. 9, 10

[MLY*17] Ma K., Li H., Yong H., Wang Z., Meng D., Zhang L.: Robust multi-exposure image fusion: A structural patch decomposition approach. *IEEE TIP 26*, 5 (May 2017), 2519–2532. 1, 2

[MMP*07] McGuire M., Matusik W., Pfister H., Chen B., Hughes J., Nayar S.: Optical splitting trees for high-precision monocular imaging. *IEEE Computer Graphics and Applications 27*, 2 (march-april 2007), 32–42. 2

[NM00] Nayar S., Mitsunaga T.: High dynamic range imaging: spatially varying pixel exposures. In *CVPR* (2000), pp. 472 – 479. 2

[NN02] Nayar S. K., Narasimhan S. G.: *Assorted Pixels: Multi-sampled Imaging with Structural Models*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2002, pp. 636–652. 2

[NSC15] Narwaria M., Silva M. P. D., Callet P. L.: Hdr-vqm: An objective quality measure for high dynamic range video. *Signal Processing: Image Communication 35* (2015), 46 – 60. 9

[OLTK15] Oh T. H., Lee J. Y., Tai Y. W., Kweon I. S.: Robust high dynamic range imaging by rank minimization. *IEEE PAMI 37*, 6 (2015), 1219–1232. 1, 2

[RB17] Ranjan A., Black M. J.: Optical flow estimation using a spatial pyramid network. In *CVPR* (July 2017). 4, 5

[RSSF02] Reinhard E., Stark M., Shirley P., Ferwerda J.: Photographic tone reproduction for digital images. *ACM TOG 21*, 3 (July 2002), 267–276. 8, 9

[RWHS15] Revaud J., Weinzaepfel P., Harchaoui Z., Schmid C.: Epicflow: Edge-preserving interpolation of correspondences for optical flow. In *IEEE CVPR* (June 2015), pp. 1164–1172. 4

[SHG*16] Serrano A., Heide F., Gutierrez D., Wetzstein G., Masia B.: Convolutional sparse coding for high dynamic range imaging. *CGF 35*, 2 (2016), 153–163. 2

[SKY*12] Sen P., Kalantari N. K., Yaesoubi M., Darabi S., Goldman D. B., Shechtman E.: Robust patch-based HDR reconstruction of dynamic scenes. *ACM TOG 31*, 6 (Nov. 2012), 203:1–203:11. 1, 2

[TKTS11] Tocci M. D., Kiser C., Tocci N., Sen P.: A versatile HDR video production system. *ACM TOG 30*, 4 (July 2011), 41:1–41:10. 1, 2

[VL15] Vedaldi A., Lenc K.: MatConvNet: Convolutional neural networks for Matlab. In *ACMMM* (2015), pp. 689–692. 8

[WXTT18] Wu S., Xu J., Tai Y.-W., Tang C.-K.: Deep high dynamic range imaging with large foreground motions. In *The European Conference on Computer Vision (ECCV)* (September 2018). 2

[WZK*17] Wang T.-C., Zhu J.-Y., Kalantari N. K., Efros A. A., Ramamoorthi R.: Light field video capture using a learning-based hybrid imaging system. *ACM TOG 36*, 4 (2017). 5, 8

[XJM12] Xu L., Jia J., Matsushita Y.: Motion detail preserving optical flow estimation. *IEEE PAMI 34*, 9 (Sept 2012), 1744–1757. 4, 9

[ZL17] Zhang J., Lalonde J. F.: Learning high dynamic range from outdoor panoramas. In *ICCV* (Oct 2017), pp. 4529–4538. 7

[ZSFC*15] Zhao H., Shi B., Fernandez-Cull C., Yeung S. K., Raskar R.: Unbounded high dynamic range photography using a modulo camera. In *ICCP 2015* (April 2015), pp. 1–10. 1, 2

[ZTS*16] Zhou T., Tulsiani S., Sun W., Malik J., Efros A. A.: *View Synthesis by Appearance Flow*. Springer International Publishing, Cham, 2016, pp. 286–301. 4