Self-Gated Memory Recurrent Network for Efficient Scalable HDR Deghosting

K Ram Prabhakar, Susmit Agrawal, and R Venkatesh Babu, *Senior Member, IEEE*

***Abstract*—We propose a novel recurrent network-based HDR deghosting method for fusing arbitrary length dynamic sequences. The proposed method uses convolutional and recurrent architectures to generate visually pleasing, ghosting-free HDR images. We introduce a new recurrent cell architecture, namely Self-Gated Memory (SGM) cell, that outperforms the standard LSTM cell while containing fewer parameters and having faster running times. In the SGM cell, the information flow through a gate is controlled by multiplying the gate's output by a function of itself. Additionally, we use two SGM cells in a bidirectional setting to improve output quality. The proposed approach achieves state-of-the-art performance compared to existing HDR deghosting methods quantitatively across three publicly available datasets while simultaneously achieving scalability to fuse variable length input sequence without necessitating re-training. Through extensive ablations, we demonstrate the importance of individual components in our proposed approach. The code is available at https://val.cds.iisc.ac.in/HDR/HDRRNN/index.html.**

***Index Terms*—High dynamic range image fusion, exposure fusion, deghosting, computational photography, convolutional neural networks.**

I. Introduction

STANDARD digital cameras have limited sensor capability that can capture only a part of the natural scene illumination. This results in the captured images being too bright or too dark in certain regions, causing loss of textural and structural details in those regions. Specialized camera sensors capable of capturing natural high dynamic ranges are prohibitively expensive for day-to-day use cases [1], [2]. A cheaper and more practical substitute is to provide a software alternative to advanced hardware. One commonly used software solution is to capture multiple Low Dynamic Range (LDR) images, each with a different exposure value, and combine them to generate an image containing a High Dynamic Range (HDR) of illumination. The generated HDR image has enriched details and textures in both highlight and shadow regions.

In the absence of camera or object motion across the captured LDR images, HDR generation is a simple process [3]. However, if the images contain relative object or camera motion, fusing them with a standard HDR merging algorithm will result in ghost-like artifacts in the final result. The process of obtaining ghost-free HDR images even in the presence of movement among captured images is known as *HDR deghosting*. The HDR deghosting problem has a rich history spanning two decades, and several methods have been proposed in the literature to address it.

Rejection based methods attempt to remove pixels affected by motion (or dynamic pixels) from largely static exposure stacks. These dynamic pixels are replaced by LDR content from one of the captured images [4]–[9]. While these methods offer good quality results for mostly static scenes, they suffer from low HDR content in moving regions. Alignment based methods transform the input stack to resemble a chosen reference image, enforcing similar structure across all inputs [10]–[14]. The aligned images are then fused using HDR merging methods such as [3]. The drawback is that most alignment methods are time and compute-intensive and are often unable to remove all artifacts. Patch-based optimization methods synthesize an HDR output that resembles a chosen reference image in well-exposed regions of reference and borrow details from other LDR images in regions where reference is poorly exposed [15]–[17]. Despite their better performance, they are complex algorithms that suffer from high computational costs.

Data-driven methods use learning algorithms, primarily Convolutional Neural Networks (CNN), to learn fusion as well as deghosting [18]–[22]. Once trained, they can generate high-quality HDR images faster than most other non-deep methods. A potential drawback, however, is that most existing learning-based methods today are not *scalable* [18], [19], [21], [22]. We refer to scalability as a trait of a method to fuse arbitrary length LDR image sequences during inference. Most existing CNN-based methods are trained to operate on three images, with the middle image being the default reference. Hence, they can only fuse three LDR images during inference, not more, and not less than three.

In practical use cases, some challenging natural scenes may contain a very high illumination range that requires more than three LDR images to cover completely. In such cases, most existing CNN based HDR deghosting methods cannot be used to merge arbitrary length LDR images. These methods need to be re-trained with a new number of input images. However, such a solution poses two challenges: 1) A new and separate model must be trained for each possible sequence length. Hence, it increases the memory and storage footprint on the device. 2) New datasets have to be captured for different sequence lengths. Additionally,

Manuscript received February 19, 2021; revised July 9, 2021; accepted August 22, 2021. Date of publication September 16, 2021; date of current version December 2, 2021. This work was supported by MeitY under Project Grant No. 4(16)/2019-ITEA and Uchhatar Avishkar Yojana (UAY) Project IISC_10, MHRD, Government of India. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Min H. Kim. *(K Ram Prabhakar and Susmit Agrawal contributed equally to this work.) (Corresponding author: K Ram Prabhakar.)*

The authors are with the Department of Computational and Data Sciences, Indian Institute of Science, Bengaluru, KA 560012, India (e-mail: kramprabhakar@gmail.com; susmit600@gmail.com; venky@cds.iisc.ac.in).

This article has supplementary downloadable material available at https://doi.org/10.1109/TCI.2021.3112920, provided by the authors.

Digital Object Identifier 10.1109/TCI.2021.3112920

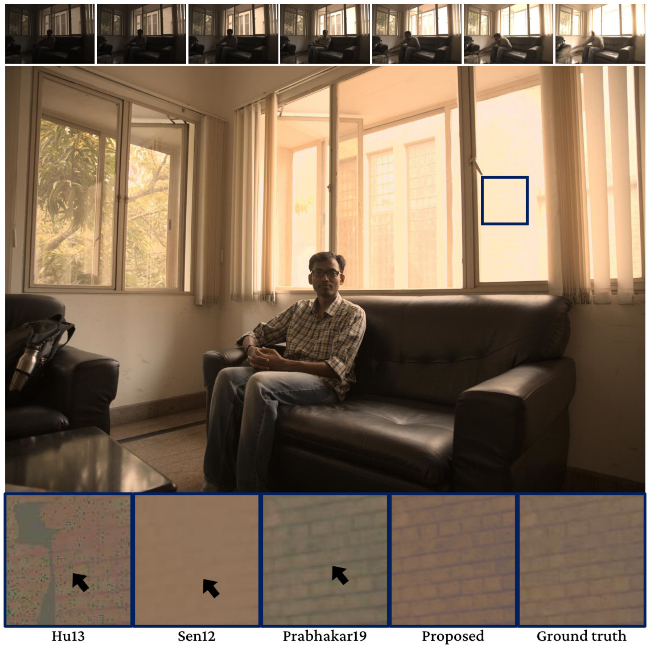

Fig. 1. A qualitative example to highlight results by various scalable HDR deghosting methods on a sequence with seven dynamic LDR images (top row). Existing scalable HDR deghosting methods are either too slow or introduce visible artifacts (highlighted by black arrow). Our proposed method generates high-quality artifact-free HDR images with less computational cost.

it is tedious to capture ground truth HDR, as it involves capturing static and dynamic sequences of the same scene with controlled motion. This shows the importance and need of having a method than can handle arbitrary exposure brackets and varying input lengths without necessitating re-training.

Among state-of-the-art non-deep methods, Sen *et al.* [15] and Hu *et al.* [16] methods are scalable. However, these methods have significant computation overhead and generate images with artifacts (see Fig. 1). The CNN-based method proposed by Prabhakar *et al.* [20] is capable of fusing a variable number of images, but at a considerable cost in terms of quality due to their internal aggregation strategy of features (see Fig. 7(c) and 10). This indicates that there is still room for developing a better and efficient scalable deep HDR deghosting method.

We address the problem as mentioned above by exploiting the internal memory capabilities of recurrent architectures. In our proposed approach, we introduce a novel recurrent cell called Self-Gated Memory (SGM) cell for efficient HDR deghosting. The SGM cell's internal vector representations make an ideal way to encode pixel color and intensity data while simultaneously filtering out structural information of non-reference images. Since the misaligned structures in non-reference images are the major cause of ghosting artifacts, removing them results in higher quality images. Also, we improve the accuracy of our model by using two SGM cells in a bidirectional setting. In summary, the main contributions of our work are as follows:

- We propose a novel network that uses recurrent architectures for efficient and scalable HDR deghosting. Our network can fuse an arbitrary number of LDR images into a ghost-free HDR image without re-training. To the best of our knowledge, our model is the first RNN-based approach proposed for multi-shot HDR deghosting.
- We present a novel Self-Gated Memory recurrent cell that uses self-gating to control data flow. The proposed SGM cell outperforms existing state-of-the-art HDR deghosting methods and other standard recurrent architectures.
- We perform extensive experiments to demonstrate the superiority of the proposed approach. Additionally, through rigorous ablation experiments, we justify the importance of each module in our proposed method.

The rest of the paper is organized as follows. Section II presents a brief review of existing HDR deghosting methods in the literature. In Section III, we present our novel recurrent architecture for HDR deghosting and describe the methodology in detail. In Section IV, we compare the proposed approach with existing state-of-the-art deghosting methods. In Section IV-C, we present the discussion on several ablations and running time comparison. Finally, we conclude the paper in Section V.

II. Related Works

Naively fusing images with camera and object motion will result in visible ghosting artifacts. Several methods have been proposed in the past to avoid such ghosting artifacts[1].

Alignment Methods: This class of algorithms aligns all input images with camera motion to a chosen reference image. Thus, the generated image sequence is mostly static and can be fused using standard HDR fusion algorithms with minimal ghosting artifacts. These methods are not primarily deghosting approaches suitable to handle object motion, hence they are used as a pre-processing step for such algorithms to eliminate global camera or scene motion. The methods within this category mostly differ by the choice of alignment strategy such as frequency domain matching and cross correlation [26]–[28], elastic registration [29], [30], SIFT [10], CIFT [31], neighbourhood similarity [32], etc. The early work by Mann *et al.* [33] proposed to jointly estimate CRF and register images using global homography. Candocia *et al.* [34] register inputs by optimizing to reduce pixel intensity variance. An alignment method proposed by Tomaszewska and Mantiuk [10] use SIFT to perform a key-point search in consecutive images of the input stack, and then use RANSAC to improve the matching. The images are warped using a homography computed from the matched key-points.

Ward [13] proposed an algorithm that runs in linear time and uses grayscale images to create image pyramids. Bogoni [11] perform both global and local alignment in two steps using affine transforms and optical flow correction. Zimmer *et al.* [12] propose a method involving the optimization of a two-term energy function, containing a data term to promote alignment to reference and a regularizer term to ensure a smooth flow in poorly exposed regions. This optimization is applied after aligning images with optical flow. A fast alignment method introduced by Gallo *et al.* [14] computes sparse correspondences between an input and the reference image and propagates the sparse flow in an edge-aware fashion to approximate the dense flow map. Kang *et al.* [35] globally align frames in a video sequence and then use gradient-based optical flow to correct local regions to generate HDR video.

[1] Please refer to [23]–[25] for more detailed literature review.

Rejection Based Methods: This class of algorithms works well with mostly static image stacks [8], [36], [37]. The static pixels that are void of motion are fused using traditional static HDR merging techniques such as [3]. The algorithms in this category can be further divided into two subgroups, depending on how they identify and handle dynamic pixels. The first subclass of these algorithms replaces moving regions with content from the chosen reference image. Grosch [4] generate an error map based on thresholded color differences between pixels. The generated error map represents moving regions, which can then be filled from the reference image. An *et al.* [38] use zero-mean normalized cross-correlation to obtain dynamic pixels between input images. Wu *et al.* [5] find pixels that violate brightness consistency to detect dynamic regions. Gallo *et al.* [6] threshold the difference in the logarithmic domain instead of directly using pixel values to identify regions affected by motion.

Lee *et al.* [39] detect moving pixels by thresholding the difference of rank normalized input images. Li *et al.* [40] compute bi-directional pixel similarity metric between input and reference to identify motion region. Min *et al.* [7] identify moving regions by performing multi-level thresholding of intensity histograms and use these regions to construct a radiance map. Raman *et al.* [9] group superpixels to identify dynamic regions. The result is used to create a largely static version of the input stack. Heo *et al.* [8] locate the ghosted pixels by thresholding the joint probability computed between the reference and the other source images. Lin *et al.* [41] threshold the difference between normalized input images to find motion regions.

The second subclass of rejection based methods contain algorithms that work without selecting a reference image from the inputs [42]–[44]. They ignore images affected by the motion for the dynamic pixels and only use the rest of the stack images. The final result contains static regions of the whole scene. Khan *et al.* [45] use a kernel density estimator function to compute pixel weights and iteratively optimize the function. Their approach does not need explicit object detection or motion estimation. Pedone *et al.* [46] improve deghosting performance of [45] with morphological operations on the estimated motion bitmaps. Pece *et al.* [47] detect clusters of pixels affected by motion using binary operations and pick the least saturated clusters. Eden *et al.* [48] use a graph-cut method to address object motion within the input stack.

Granados *et al.* [49] iteratively reconstruct the HDR image by estimating a mean radiance map with minimum variance. An earlier work by Reinhard *et al.* [50] utilizes normalized variance to identify motion affected pixels. Building on top of [50]'s approach, Jacobs *et al.* [51] proposed to use an entropy measure instead of variance. Sidibe *et al.* [52] increase the exposure and locate dynamic pixels that does not proportionally increase its intensity value. Oh *et al.* [53] proposed a rank minimization approach to generate final HDR output. In their approach, authors calculate rank-1 matrix of stacked input images. The residual noise of the estimate is used to locate the moving dynamic pixels.

Non-Rigid Registration Methods: The following set of methods use optical flow to handle both global camera motion as well as local object motion. These algorithms warp moving regions in images of the input stack, such that these regions align with a chosen reference image. The final HDR result is obtained by merging the aligned static sequence. Ferradans *et al.* [54] use GMM to model the difference between optical flow warped images. The pixels that do not fit to estimated modes within pre-defined variance receives less weight in the fusion process. Jinno *et al.* [55] jointly estimate flow, occlusion and saturation maps by minimizing an energy function, which are later used to fuse the images. Hafner *et al.* [56] jointly estimate fused HDR and displacement fields in a optimization framework enforced with spatial smoothness constraints. This category methods suffer from erroneous dense correspondence or optical flow due to complex non-rigid motions and occlusion.

Patch Based Optimization Methods: This class of methods generates a final HDR result similar to the reference image in regions where the reference image is properly exposed [57], [58]. Sen *et al.* [15] propose a method to optimize both structure and content of the HDR image. They achieve it by parameterizing these quantities in an image synthesis equation and optimize for the same. Hu *et al.* [16] generate latent images from the input stack that are similar in structure but vary in an exposure. While this class of methods addresses the shortcomings of rejection based methods and registration methods, they suffer from high computational complexity.

Data Driven Techniques: These approaches make use of Convolutional Neural Networks trained on data that captures the essence of the HDR deghosting problem. Fusion and Deghosting rules are approximated by learning from examples. Kalantari *et al.* [18] use an input stack that has been aligned using optical flow and train the model to correct warping artifacts. Wu *et al.* [19] directly use neural networks to learn both alignment and fusion, thus attempting to remove the overhead of optical flow. Yan *et al.* [59] propose attention mechanisms to focus only on the relevant information from the input stack.

Yan *et al.* [60] use a network with a non-local module to identify matching neighbor features to fill in ill-exposed regions of the reference image. Prabhakar *et al.* [20] aggregate features derived by shared CNN modules to create a scalable architecture that can fuse an arbitrary number of images without retraining. Prabhakar *et al.* [22] propose a method to minimize the consumption of memory and enable the fusion of high-resolution images. They perform alignment on low-resolution inputs and upsample intermediate features while generating the final HDR using a Bilateral Grid Upsampler [61]. In spite of their better performance, most of the existing deep-learning methods [18], [19], [21], [22], [59], [60] are not scalable to fuse arbitrary number of images without re-training.

Recurrent Neural Networks have been extensively explored for processing temporal data such as speech [62]–[64], natural language processing [65]–[67], video captioning [68], [69], video segmentation [70], [71], low-level video processing [72], [73] and in so many other applications. The closest matching related application to HDR deghosting that utilizes recurrent architecture is burst denoising approach by Godard *et al.* [72] and the video denoising method by Chen *et al.* [73]. To the best of our knowledge, there has not been any previous attempt to use recurrent networks for HDR deghosting.

III. Proposed Method

A. Method Overview

Given a list of N Low Dynamic Range (LDR) images $S = \{I_1, I_2, \ldots, I_N\}$, the goal of our approach is to merge them into a single HDR image $\mathbf{Y}$ without any ghosting artifacts. Assuming

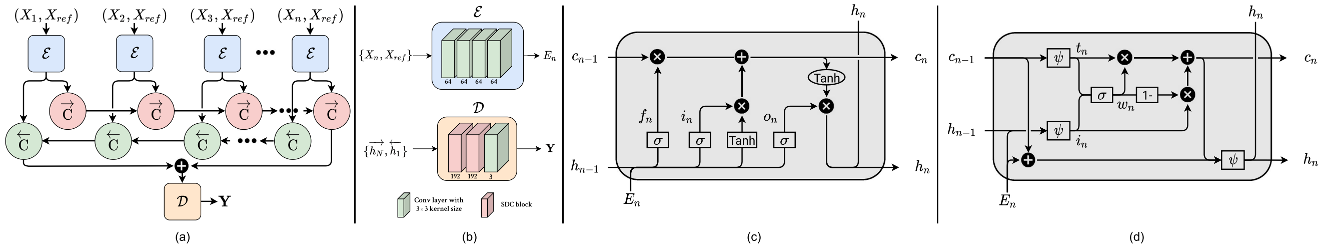

Fig. 2. (a) The architecture of the proposed approach. The current input data X_n concatenated with the reference data X_{ref} is passed as input to the encoder ($\mathcal{E}$). The encoded features are fed to the proposed bi-directional SGM recurrent cells ($\overrightarrow{\mathrm{C}}$ and $\overleftarrow{\mathrm{C}}$). The final output of the recurrent cells in both directions are concatenated and passed to the decoder ($\mathcal{D}$) to generate fused HDR ($\mathbf{Y}$). (b) Architecture of encoder ($\mathcal{E}$) and decoder ($\mathcal{D}$). The number below each block denotes the number of output channels of the corresponding block. Architecture of (c) standard LSTM cell and (d) proposed Self-Gated Memory (SGM) recurrent cell.

that the input sequence images are camera motion aligned, the challenge is to combine these images while accounting for object motion. Our proposed method is a reference-based method, where the final result will be structurally similar to one of the input images chosen as reference (I_{ref}). In particular, for static regions (regions without any object motion), the result contains HDR content fused from all input images. For dynamic regions (regions affected by motion), the result will have the same structure as the chosen reference image but with HDR content added from other images. Typically, the image with the least saturated pixels is chosen as a reference.

Motivation: Most deep learning-based approaches ([18], [19], [22], [59]) are trained to fuse fixed number of LDR images, usually 3 images. These methods require re-training with the corresponding number of images for fusing a sequence with the different number of input images. Another shortcoming is the lack of a training dataset for different sequence lengths. It is very tedious to collect large labeled datasets for different sequence lengths, as it requires capturing both static and dynamic sequences of same scene. Prabhakar *et al.* [20] addressed scalability in their work by concatenating the mean and max of input features. However, their method still suffers from artifacts in challenging scenarios (shown in Fig. 1).

Our proposed approach consists of a single neural network that uses convolutional and recurrent architectures. As shown in Fig. 2(a), the overall network can be logically divided into three sub-modules: an encoder, two recurrent cells in a bidirectional setting, and a decoder.

Encoder ($\mathcal{E}$): As shown in Fig. 2(b), the encoder is a block of 3×3 convolutional layers. It acts as a basic feature extractor and processes the input images for fusion by the recurrent module. Since the number of input images can vary, we use a single encoder to process all N images in the input sequence. For the rest of the paper, let I_n represent the n^{th} image in the input sequence. The gamma-corrected HDR version (H_n) of I_n is obtained by $H_n = I_n^{2.2}/t_n$, where t_n is the exposure time of I_n. Let X_n be the concatenation of I_n and H_n. $\mathcal{E}$ then takes concatenated X_n and X_{ref} as input,

$$X_n = I_n \oplus H_n \tag{1}$$

$$X_{ref} = I_{ref} \oplus H_{ref} \tag{2}$$

$$E_n = \mathcal{E}\left(\{X_n, X_{ref}\}\right) \tag{3}$$

where, $\oplus$ denotes the feature concatenation operation along the channel axis.

Recurrent Module: This is the core module in the proposed architecture. The recurrent module takes as input the list of encoded images $E_1, E_2, \ldots, E_n$ and iteratively fuses them into output vectors $h_1, h_2, \ldots, h_N$, such that an output vector h_n at the n^{th} timestep contains information from all input vectors E_1 to E_n. The final output h_N thus contains fused information from the entire input sequence.

The recurrent module may contain a single recurrent cell in a unidirectional setting, or it may use two cells in a bidirectional arrangement. In the bidirectional setting, one of the cells processes the input sequence starting from E_1 to E_N, while the second cell processes inputs starting from E_N to E_1. The outputs for the cells are $\overrightarrow{h_N}$ and $\overleftarrow{h_1}$ respectively, with the arrows indicating the direction in which the input sequence is processed. The module's output is then given by concatenating the two vectors $\overrightarrow{h_N}$ and $\overleftarrow{h_1}$.

In the forward direction, h_0 and c_0 are zero vectors of the same height and width as each input image, named $\overrightarrow{h_{init}}$ and $\overrightarrow{c_{init}}$. Then, in the reverse direction, h_{N+1} and c_{N+1} are $\overleftarrow{h_{init}}$ and $\overleftarrow{c_{init}}$ respectively. We use cell $\overrightarrow{\mathrm{C}}$ in forward and a separate cell $\overleftarrow{\mathrm{C}}$ in reverse direction for images $i = (1, \ldots, N)$,

$$(\overrightarrow{h_i}, \overrightarrow{c_i}) = \overrightarrow{\mathrm{C}}\left(E_i, \overrightarrow{h_{i-1}}, \overrightarrow{c_{i-1}}\right), \tag{4}$$

$$(\overleftarrow{h_{N-i+1}}, \overleftarrow{c_{N-i+1}}) = \overleftarrow{\mathrm{C}}\left(E_{N-i+1}, \overleftarrow{h_{N-i+2}}, \overleftarrow{c_{N-i+2}}\right) \tag{5}$$

For N=3, the forward cell unrolls as,

$$(\overrightarrow{h_1}, \overrightarrow{c_1}) = \overrightarrow{\mathrm{C}}\left(E_1, \overrightarrow{h_{init}}, \overrightarrow{c_{init}}\right), \tag{6}$$

$$(\overrightarrow{h_2}, \overrightarrow{c_2}) = \overrightarrow{\mathrm{C}}\left(E_2, \overrightarrow{h_1}, \overrightarrow{c_1}\right), \tag{7}$$

$$(\overrightarrow{h_3}, \overrightarrow{c_3}) = \overrightarrow{\mathrm{C}}\left(E_3, \overrightarrow{h_2}, \overrightarrow{c_2}\right) \tag{8}$$

Whereas the reverse cell unrolls as,

$$(\overleftarrow{h_3}, \overleftarrow{c_3}) = \overleftarrow{\mathrm{C}}\left(E_3, \overleftarrow{h_{init}}, \overleftarrow{c_{init}}\right), \tag{9}$$

$$(\overleftarrow{h_2}, \overleftarrow{c_2}) = \overleftarrow{\mathrm{C}}\left(E_2, \overleftarrow{h_3}, \overleftarrow{c_3}\right), \tag{10}$$

$$(\overleftarrow{h_1}, \overleftarrow{c_1}) = \overleftarrow{\mathrm{C}}\left(E_1, \overleftarrow{h_2}, \overleftarrow{c_2}\right) \tag{11}$$

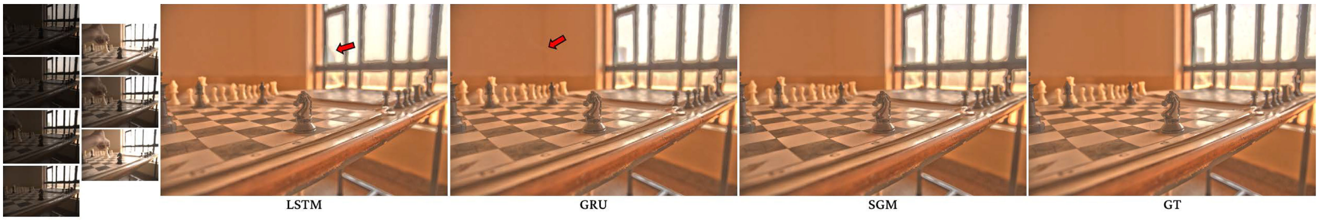

Fig. 3. Qualitative comparison between proposed SGM cell against baseline recurrent cell types: LSTM and GRU.

Feature decoder ($\mathcal{D}$): After processing the complete input sequence, the output features produced by the recurrent module are passed to a decoder block that generates the final HDR image. In bidirectional architecture, the concatenation of the final outputs from both recurrent cells serves as the input to this module (Fig. 2(b)),

$$\mathbf{Y} = \mathcal{D}(\overrightarrow{h_N} \oplus \overleftarrow{h_1}) \tag{12}$$

In our implementation, we use two stacked SDC blocks [74] to capture details across different receptive fields, followed by a single convolutional layer to generate the final image.

Loss: The model produces an HDR image $\mathbf{Y}$ as its output. This image is tonemapped using the μ-law tonemapping function defined as,

$$T(\mathbf{Y}) = \frac{log(1 + \mu\mathbf{Y})}{log(1 + \mu)} \tag{13}$$

where $\mu = 5000$. The loss $\mathcal{L}$ between the ground truth HDR $\hat{\mathbf{Y}}$ and the predicted image $\mathbf{Y}$ is obtained by computing the ℓ_2 loss between their tonemapped representations:

$$\mathcal{L} = \ell_2(T(\mathbf{Y}), T(\hat{\mathbf{Y}})) \tag{14}$$

B. Self-Gated Memory (SGM) Recurrent Cell

Before we present the details of the proposed SGM cell, we first discuss existing standard recurrent cells, their shortcomings and finally on how they are addressed in the SGM cell. We start with the three standard recurrent cell types, the standard RNN cell, the Long Short-Term Memory (LSTM), and the Gated Recurrent Unit (GRU) cells. Out of all three, LSTM performed better than other two cell types in terms of quantitative metrics (Table III). Qualitatively, images generated by LSTM has less ghosting artifacts compared to GRU (Fig. 3). However, we identify three challenges in LSTM cell design that can be addressed to improve the performance specific to HDR deghosting task.

The problems encountered with LSTM cell are discussed below, along with the corresponding change in the proposed SGM cell architecture to address the issue:

i) The forgetting of information from the internal state in both LSTM and SGM can be represented in the form: $c_n = \alpha \cdot c_{n-1} + \beta$. It is evident that α plays a large role in deciding what information is forgotten. In case of LSTM, α is obtained from the forget gate as shown below:

$$\begin{array}{c|c} \text{LSTM} & \text{SGM} \\ \alpha = \sigma((h_{n-1} \oplus E_n) \odot k + b) & \alpha = \sigma((t_n \oplus i_n) \odot k + b) \\ & t_n = \psi(c_{n-1} \odot k + b) \end{array}$$

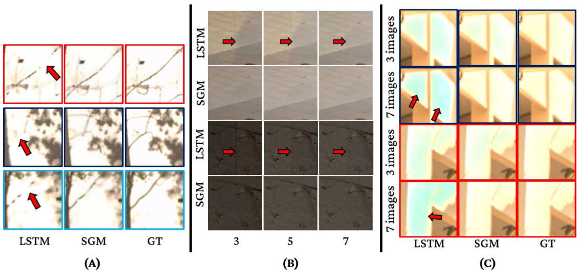

Fig. 4. Qualitative comparison between LSTM and SGM cells.

TABLE I
COMPARISON OF VARIOUS SGM CELL ARCHITECTURES

	Metrics (→)	PSNR-L			PSNR-T		
	Images (→) / Cell types (↓)	3	5	7	3	5	7
	LSTM	43.10	43.90	50.30	47.02	47.63	48.28
Proposed	Type1	43.21	43.69	49.25	46.61	47.37	47.06
	Type2	42.68	42.99	49.02	46.35	46.44	46.82
	Type3	42.13	43.02	49.49	45.76	46.38	46.35
	Type4	43.01	43.25	48.12	46.05	45.89	45.05
	Type5	42.29	43.34	49.19	46.09	46.82	47.21
	Type6	43.08	43.73	49.98	46.95	47.20	47.43
	Type7	43.16	43.55	49.28	46.19	46.70	46.50
	Final SGM	**43.88**	**44.37**	**51.04**	**47.68**	**48.05**	**48.69**

It is obvious through the equations that c_{n-1} is not directly involved in deciding what information is to be "forgotten" (Fig. 2(c)). Thus, the important exposure information may be forgotten without much importance given to c_{n-1}. The forgotten information is then overwritten with information from the current image, resulting in a large contribution of final images to the output. In some cases, this manifests as saturated regions in the output, even though intermediate images may have the required detail to generate saturation-free results.

To prevent this, we use c_{n-1} in computation of the weight maps used for forgetting in the SGM cell. In the SGM cell, as shown in above table and in Fig. 2(d), c_{n-1} contributes significantly in computation of the new state c_n. Visually, we see that SGM cell has retained structural information in locations where LSTM fails to, as shown in Fig. 4(a). Quantitatively, cell architecture ablations types 6 and 7 in Table I show that not using c_{n-1} actively for computing the new state results in degraded performance.

ii) LSTM performs minimal computation before deleting information from the long-term memory. The bulk of computation is performed in the input gate, and the result is unconditionally integrated with c. If any structural artifacts are not removed by the input gate, they will propagate through the rest of the sequence with high probability. These show up as warping artifacts in the final output of the LSTM cell, as shown in Fig. 4(b).

To address this issue in the SGM, we propose a set of architectural changes. First, we have individual layers for both h and c, resulting in preliminary filtering from both long-term and short-term memory before state update. Next, we use both states to compute the weight map that ultimately updates c. It can be seen in Fig. 4(b) that this design helps to eliminate warping artifacts. The cell architecture ablations for these changes are Type 4, Type 6 and Type 7.

iii) Finally, LSTM maps high values generated internally as well as in its output to $[-1, 1]$ range, due to the Tanh activations. This means that if two pixel values from two different images are high, they will be mapped close together, losing relative intensity information between them. This is especially problematic if the number of images during evaluation is higher than the number of training images since they will probably have more oversaturated regions than what the LSTM has seen during training. In such cases, having additional images will not result in higher quality output, as shown in Fig. 4(c). Both models were trained with only 3 images and tested on 3 and 7 length sequences. As highlighted by the arrows, an LSTM model trained with 3 images, generates oversaturation when tested with 7 images. In contrast, SGM cell can generate output without such artifacts.

SGM implicitly addresses this due to the Swish activation, which has an unbounded upper limit. Fig. 4(c) shows that SGM is able to extract information from long sequences, where LSTM saturates in quality.

With these major considerations, we perform a series of cell architecture ablations to empirically get the optimal architecture for our application. The final cell contains the following structures:

Input gate: This gate is used to extract relevant features from the input E_n to the cell, and its previous output h_{n-1}. It uses a convolution block followed by self-gating:

$$\overrightarrow{i_n} = \psi((\overrightarrow{h_{n-1}} \oplus E_n) \odot \overrightarrow{k_i} + \overrightarrow{b_i}) \tag{15}$$

where k_i is the kernel used to convolve the input, b_i is the bias, $\oplus$ is the concatenation operation, and $\odot$ is the convolution operation.

Transform gate: This gate is used to extract relevant features from the previous internal state c_{n-1} of the cell. It uses a convolution block followed by self-gating:

$$\overrightarrow{t_n} = \psi(\overrightarrow{c_{n-1}} \odot \overrightarrow{k_t} + \overrightarrow{b_t}) \tag{16}$$

where k_t is the kernel used to convolve the previous state, and b_t is the bias.

Update gate: The update gate is used to update the internal state c_n of the cell, using the features extracted from the previous state c_{n-1}, previous output h_{n-1} and the input E_n. It performs a weighted sum of the two features to derive the new state c_n. The weights are determined using a convolutional block having sigmoid activation:

$$\overrightarrow{w_n} = \sigma((\overrightarrow{i_n} \oplus \overrightarrow{t_n}) \odot \overrightarrow{k_w} + \overrightarrow{b_w}) \tag{17}$$

$$\overrightarrow{c_n} = \overrightarrow{w_n} \cdot \overrightarrow{t_n} + [1 - \overrightarrow{w_n}] \cdot \overrightarrow{i_n} \tag{18}$$

where k_w is the kernel used to convolve the input, and b_w is the bias.

Output Gate: This gate produces the final output, h_n. It uses a convolution block followed by self-gating. The inputs to this gate are the old cell state (c_{n-1}), the new state (c_n) and the current input (E_n):

$$\overrightarrow{h_n} = \psi((\overrightarrow{c_{n-1}} \oplus (\overrightarrow{c_n} + E_n)) \odot \overrightarrow{k_o} + \overrightarrow{b_o}) \tag{19}$$

where k_o is the kernel used to convolve the previous state, and b_o is the bias.

We use two cells in a bidirectional arrangement in our implementation, each having 64 filters with 3 × 3 kernel size for all its convolutional layers. The equations for the cell operating on the sequence $\mathcal{S}$ in the reverse direction are identical to the equations mentioned above, applied from I_N to I_1:

$$\overleftarrow{i_n} = \psi\left((\overleftarrow{h_{n+1}} \oplus E_n) \odot \overleftarrow{k_i} + \overleftarrow{b_i}\right) \tag{20}$$

$$\overleftarrow{t_n} = \psi(\overleftarrow{c_{n+1}} \odot \overleftarrow{k_t} + \overleftarrow{b_t}) \tag{21}$$

$$\overleftarrow{w_n} = \sigma((\overleftarrow{i_n} \oplus \overleftarrow{t_n}) \odot \overleftarrow{k_w} + \overleftarrow{b_w}) \tag{22}$$

$$\overleftarrow{c_n} = (\overleftarrow{w_n} * \overleftarrow{t_n}) + ((1 - \overleftarrow{w_n}) * \overleftarrow{i_n}) \tag{23}$$

$$\overleftarrow{h_n} = \psi((\overleftarrow{c_n} \oplus (\overleftarrow{c_{n+1}} + E_n)) \odot \overleftarrow{k_o} + \overleftarrow{b_o}) \tag{24}$$

To show the contribution of each structure and connection within the cell, we perform a series of architecture experiments with the particular component removed. The following ablations are performed:

- Type 1: We remove the connection feeding c_{n-1} to the remaining inputs of the output gate. The output of the cell now becomes a function of the input of the current internal state and input, with the information of previous inputs present as a nonlinear transformation within the new state. We observe that removing this connection results in a minor dip in overall performance of the network.
- Type 2: We do not concatenate E_n to the other inputs of the output gate. The output is thus derived only from the new state that integrates information from the current input, and previous state. Important information about the specific frame may be lost during this computation. As such, the drop in performance after removing this connection is significant.
- Type 3: The output gate only takes c_n as its input; c_{n-1} and E_n are not used. As expected, drop in overall performance is greater than removing any of the individual connections.
- Type 4: We remove the transform gate completely from the cell. The remaining parts are same as the final version. The drop in performance indicates that it is advantageous to have some of the local information in c_{n-1} forgotten before the state update, without interference from the new input.
- Type 5: We replace self-gating with two separate layers having sigmoid and Tanh as their activations, and multiply

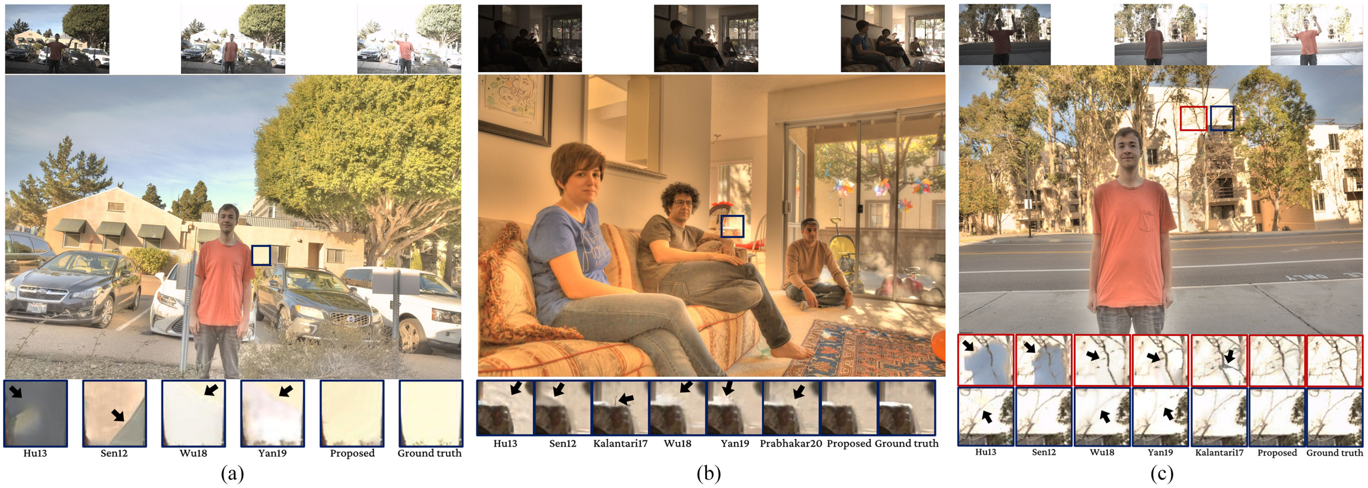

Fig. 5. Qualitative comparison between state-of-the-art approaches and the proposed approach on three validation sequences from Kalantari17 dataset [18].

them to get the output of the gate. We follow the LSTM-style connections in the output gate. This shows that it is harder to optimize gates having two separate components.

- Type 6: We perform LSTM-style connections to “forget” parts of c_{n-1}. That is, c_{n-1} is not used to compute the weightmaps to perform state update.
- Type 7: We evaluate whether the order of processing $\{E_n, h_{n-1}\}$ and forgetting from c_{n-1} impacts performance. We modify the cell structure from Type 6 ablation to perform this experiment.

IV. Evaluation and Results

A. Implementation

Our models and pipelines were implemented using Tensorflow. The network was trained on a machine with Intel core i7-8700 CPU and a NVIDIA RTX 2080ti 11 GB GPU. We use the Adam optimizer [77] with 2×10^{-4} learning rate and a batch size of four to train the model for 200 epochs. The learning rate is halved after every 25 epochs. We train the model using Kalantari *et al.* [18] and Prabhakar *et al.* [20] datasets for 3-image fusion. [18] dataset consists of 74 training and 15 validation sequences, while [20] dataset consists of 466 training and 116 validation sequences. In both [18] and [20] datasets, each sequence consists of three varying exposure images with exposure bias $\{-2, 0, +2\}$ or $\{-3, 0, +3\}$. Additionally, we trained on Prabhakar *et al.* [78] dataset for validating scalability performance. Dataset given by [78] consists of 70 training sequences with 7 LDR images, and the sequence length (3, 5 or 7) is selected randomly during training. It also contains 14 sequences with 7 LDR images for testing. The sequence $\{I_2, I_4, I_6\}$ is used for validating 3-image performance and sequence $\{I_2, I_3, I_4, I_5, I_6\}$ is used for performing 5-image validation. Images I_1 through I_7 are all used for 7-image validation. Apart from [18], [20] and [78], we have validated our model on publicly available datasets like Sen *et al.* [15], Cambridge [79] and Tursun *et al.* [80].

B. Quantitative Results

In Table II, we compare the proposed method trained for 3 images on [18] and [20] datasets against nine other HDR deghosting approaches. The approaches against which we compare the proposed approach are: 1) Hu13 - Hu *et al.* [16], 2) Sen12 - Sen *et al.* [15], 3) Endo17 - Endo *et al.* [81], 4) Eilertsen17 - Eilertsen *et al.* [82], 5) Kalantari17 - Kalantari *et al.* [18], 6) Wu18 - Wu *et al.* [19], 7) Prabhakar19 - Prabhakar *et al.* [20], 8) Prabhakar20 - Prabhakar *et al.* [22], 9) Yan19 - Yan *et al.* [59]. We evaluate the performance of different methods using three popular metrics: HDR-VDP-2 [83], SSIM [84] and PSNR. HDR-VDP-2 is a full reference HDR image quality assessment metric for evaluating prediction quality in all luminance ranges.

To get an overall view of quality, we compute SSIM and PSNR using four different tonemapping functions, in addition to comparison in the linear domain. The tonemapping functions used are: Log tonemapper (Eqn. (13)), Krawczyk *et al.* [75], Reinhard *et al.* [85] and Durand *et al.* [76]. In total, we compare against nine existing methods on eleven different metrics. Table II(a) shows quantitative comparison on Kalantari17 dataset with 15 test sequences. Our proposed method outperforms all other existing methods on ten out of eleven metrics. Similarly, in Table II(b), we present a quantitative comparison on Prabhakar19 dataset with 116 test images. We have trained all existing methods and our method on [20] for a fair comparison. The proposed approach outperforms all approaches in every metric except for PSNR on the [76] tonemapping function.

In Table III, we present quantitative comparison against existing scalable HDR deghosting methods on Prabhakar *et al.* dataset [78] with sequences of length 3,5 and 7. We also compare against different recurrent cells to demonstrate the proposed SGM cell’s effectiveness.

C. Qualitative Results

Among the classical non-deep HDR deghosting methods, Hu13 [16] and Sen12 [15] are popular patch-based optimization approaches.[2] As both Hu13 and Sen12 methods synthesize the

[2] Please refer to Supplementary material for more results.

TABLE II

QUANTITATIVE COMPARISON BETWEEN THE PROPOSED METHOD AGAINST NINE STATE-OF-THE-ART METHODS. THE BEST SCORE IS HIGHLIGHTED BY BLUE CELL COLOR, AND THE SECOND-BEST SCORE IS HIGHLIGHTED BY GRAY CELL COLOR. THE ABBREVATIONS USED ARE AS FOLLOWS: T1 - LINEAR DOMAIN, T2 - LOG TONEMAPPER, T3 - KRAWCZYK *ET AL.* [75], T4 - REINHARD *ET AL.* [50], AND T5 - DURAND *ET AL.* [76] TONEMAPPERS

	PSNR					SSIM					HDR-VDP-2
	T1	T2	T3	T4	T5	T1	T2	T3	T4	T5	
Hu13	31.25	35.75	30.18	28.87	31.69	0.941	0.963	0.940	0.942	0.944	62.07
Sen12	38.57	40.94	27.81	30.33	31.35	0.971	0.978	0.955	0.966	0.969	64.74
Endo17	8.846	21.33	12.26	12.10	20.65	0.107	0.622	0.715	0.787	0.738	54.00
Eilertsen17	14.21	14.13	25.23	20.26	26.82	0.350	0.882	0.925	0.923	0.935	57.95
Kalantari17	41.27	42.74	34.12	33.70	32.99	0.981	0.987	0.980	0.979	0.974	66.10
Wu18	40.91	41.65	34.98	34.54	33.69	0.986	0.986	0.982	0.982	0.977	67.44
Prabhakar19	39.68	40.47	33.56	34.08	32.60	0.980	0.975	0.966	0.975	0.961	66.50
Prabhakar20	41.33	**42.82**	35.18	36.94	36.29	0.986	0.989	0.984	0.985	0.982	67.15
Yan19	41.08	41.21	29.83	33.28	29.51	0.989	0.989	0.962	0.978	0.974	67.53
Proposed	**41.68**	42.07	**36.09**	**37.85**	**36.29**	**0.990**	**0.990**	**0.986**	**0.988**	**0.983**	**67.59**

(a) Quantitative evaluation on [18] dataset.

	PSNR					SSIM					HDR-VDP-2
	T1	T2	T3	T4	T5	T1	T2	T3	T4	T5	
Hu13	29.47	32.58	27.61	26.84	26.20	0.954	0.949	0.912	0.919	0.917	63.50
Sen12	32.93	33.43	30.22	29.15	30.91	0.972	0.964	0.950	0.948	0.950	65.47
Endo17	9.760	8.980	11.18	13.07	20.60	0.132	0.641	0.675	0.763	0.711	55.76
Eilertsen17	14.19	15.66	22.47	22.04	24.97	0.442	0.869	0.879	0.897	0.904	58.74
Kalantari17	32.50	35.63	30.08	28.81	30.45	0.969	0.961	0.938	0.943	0.940	65.40
Wu18	34.40	38.03	33.35	31.82	32.66	0.977	0.971	0.962	0.957	0.956	66.59
Prabhakar19	32.74	36.08	30.66	29.83	30.54	0.967	0.959	0.942	0.935	0.939	66.10
Prabhakar20	34.98	38.30	32.99	31.76	32.53	0.978	0.970	0.960	0.952	0.953	66.25
Yan19	35.28	38.65	33.82	32.08	**33.31**	0.980	0.973	0.963	0.961	0.957	66.88
Proposed	**36.38**	**39.03**	**34.45**	**32.43**	32.85	**0.983**	**0.975**	**0.967**	**0.963**	**0.960**	**67.95**

(b) Quantitative evaluation on [20] dataset.

TABLE III

COMPARISON OF THE PROPOSED METHOD AGAINST EXISTING METHODS TO FUSE VARIABLE NUMBER OF IMAGES FROM [78] DATASET

	Metrics (→)	PSNR-L				PSNR-T			
	Images (→) / Methods (↓)	3	5	7	Mean	3	5	7	Mean
	Sen12	40.61	41.71	47.08	43.13	46.26	47.51	47.27	47.01
	Hu13	35.71	36.38	39.31	37.13	41.85	42.65	40.81	41.77
	Prabhakar19	39.53	39.47	45.43	41.47	39.79	40.02	40.10	39.97
Proposed	Bi-Vanilla	43.26	43.88	50.07	45.74	46.15	46.52	47.16	46.61
	Bi-LSTM	43.10	43.90	50.30	45.77	47.02	47.63	48.28	47.64
	Bi-GRU	42.46	43.02	49.23	44.90	45.69	46.01	46.45	46.05
	Bi-SGM	**43.88**	**44.37**	**51.04**	**46.43**	**47.68**	**48.05**	**48.69**	**48.14**

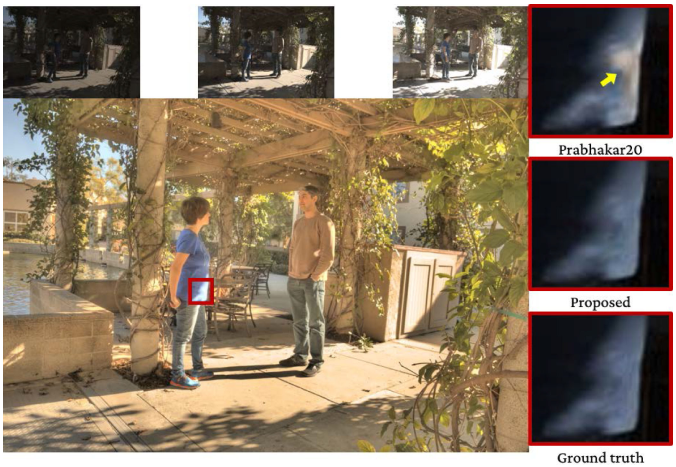

Fig. 6. A qualitative example from Kalantari17 dataset [18]. The red zoomed box highlight the artifact introduced by the Prabhakar20 method on saturated dynamic regions. Comparatively, the proposed method can generate plausible textures, even in saturated areas. Image best viewed electronically.

result with reference images, these methods' success depends on carefully choosing a reference image with the least amount of saturation. In the presence of heavily saturated regions in the reference image, they tend to introduce artifacts. As shown in Fig. 1, the result by Hu13 method has structural artifacts in heavily saturated regions of the reference image. In the same region, the Sen12 method generates results without any texture information. Similarly, in 5(c), we present a qualitative comparison between different state-of-the-art methods and proposed method on a challenging sequence from the Kalantari17 dataset. The results in Fig. 5 and 10 show the artifacts introduced by both Hu13 and Sen12 methods on various sequences with arbitrary length sequences.

Fig. 5(c) shows a qualitative comparison between Kalantari17 and the proposed method on a difficult example from the Kalantari17 dataset. The Kalantari17 method has residual optical flow warping artifacts present in the result shown in the zoomed red box. Similarly, in Fig. 5(b), their result has warping artifact along the boundaries. In another example from the Prabhakar19 dataset shown in Fig. 7(c), Kalantari17 method fails to correct warping artifacts in the dynamic regions. In Fig. 7(a), Kalantari17 method's result has artifacts even in static regions.

As shown in Fig. 5(c), 5 and 7, Wu18 and Yan19 methods suffer from artifacts in reference saturated dynamic regions. In such regions, Wu18 and Yan19 methods fail to reconstruct complete texture from other LDR input images. In 5(c), both Wu18 and Yan19 generate results with missing branches. Also, in Fig. 5, both methods introduce artifacts in regions affection by saturation and motion. Similarly, in Fig. 7(a) and Fig. 7 b, Wu18 and Yan19 results fail to faithfully reconstruct window bars; instead, they smoothen out the details.

Prabhakar20 method generates output with artifacts as shown in Fig. 6. The zoomed highlighted region is affected by motion and saturation. As the Prabhakar20 method's input is optical flow corrected, their approach hallucinates incorrect textures in the saturated areas. In Fig. 5(b), a similar artifact is noticed in the saturated and moving region. In Fig. 7(c), both Prabhakar20 and Kalantari17 methods result in artifacts due to optical flow alignment. In Fig. 7(a) and Fig. 7(b), Prabhakar20 method has residual ghosting artifact left in the result.

Prabhakar19 is the only scalable deep learning-based method. In Fig. 1, we show an example with seven input images. Prabhakar19 method's output has color distortion in the region marked with a blue box. In Fig. 7(c), the Prabhakar19 method's output has an over-smoothing artifact in the saturated reference region. Similarly, in Fig. 10(a) the output of Prabhakar19 method contains inaccurate texture details. In Fig. 7(b), Prabhakar19 method over-smoothens the cloud region, thus losing the necessary details in them.

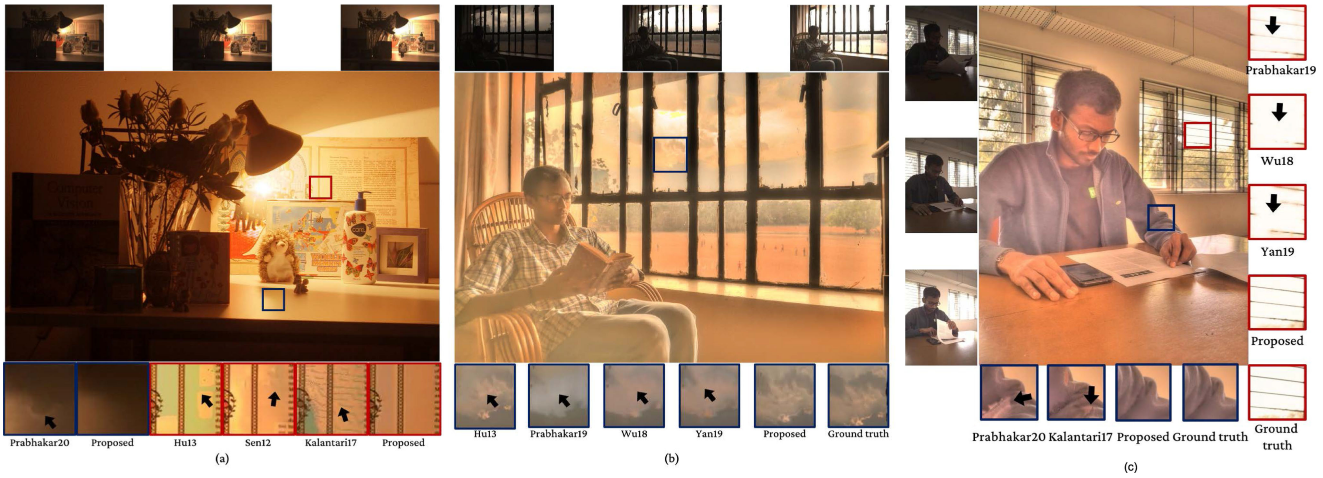

Fig. 7. Qualitative comparison on sequences from (a) Cambridge dataset [79], (b,c) Prabhakar19 dataset [20]. Image best viewed electronically.

TABLE IV
QUANTITATIVE COMPARISON BETWEEN SEVERAL BASELINE ABLATION EXPERIMENTS ON KALANTARI17 DATASET

Uni/Bi directional	Optical flow	PSNR-L	PSNR-T	HDR-VDP-2
→	✗	41.16	41.71	67.29
	✓	41.64	41.84	67.35
⇄	✗	41.26	41.93	67.33
	✓	**41.68**	**42.07**	**67.59**

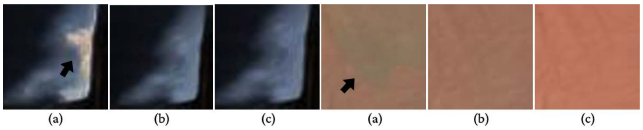

Fig. 8. Qualitative comparison between (a) unidirectional and (b) bidirectional SGM cells. As highlighted by the black arrow, the output by unidirectional cell has artifacts. The ground truth is shown in (c).

TABLE V
AN ABLATION STUDY ON THE IMPORTANCE OF SGM CELL COMPONENTS

Ablations	PSNR-L (dB)
Shuffled input sequence	41.53
Tanh at the output gate	40.74
Without Sigmoid in Eqn. (16), (18) and (19)	41.05
Proposed final architecture	**41.68**

In comparison, the proposed method generates results with accurate texture and vivid color details for variable-length sequences without re-training. The proposed method's result is void of ghosting artifacts and can generate plausible textures in regions affected by motion and saturation (Fig. 6 and 5). It should be noted that, as the existing state-of-the-art deep learning methods except Prabhakar19, were trained on three input images, we tested those methods only on sequences with three input images.

In Fig. 9, we present an example for adding more LDR images in a dynamic scene. As seen from the results, adding more images leads to increase in the dynamic range.

D. Ablations

Uni vs Bi-directional SGM: We run various experiments to show the effectiveness of our approach in a variety of baseline settings. To validate the importance of bi-directional setting, we train a model using only one SGM cell in one direction. The input to decoder consists of final time step feature maps. The results are provided in Table IV. Also, we observe that using optical flow provides slight boost across all metrics. We observe that bi-directional SGM cell offers better results compared to uni-directional cell in all three comparison metrics. The qualitative comparison in Fig. 8 shows that the output by uni-directional model has visible artifacts in saturated regions, which is corrected in bi-directional model.

Scalability: To verify the effectiveness of our model to fuse arbitrary number of images without retraining, we have tested a model trained with UCSD dataset [18] on [78] dataset without retraining for different input image lengths. We achieved a average PSNR-L of 43.64 dB, while the only other learnable scalable architecture [20] achieves 38.02 dB. Also, we show qualitative comparisons for HDR Deghosting on the METU dataset consisting of 9-image sequences without re-training (Fig. 10). Additionally, we also show qualitative results for the Multi-Exposure fusion problem using our method, where the models have been trained on sequence lengths ranging from 3 images to 9 images and tested on examples having 30 images in a sequence (Fig. 9 in supplementary file).

Image order: In order to understand the importance of image order, we train a model after randomly shuffling the low, medium and high exposure images in the [18] dataset. In this experiment, we have considered all permutations of the three sequences, including those where the reference image was displaced from the central spot. The medium exposure image is used as reference, however its location in the sequence is random. For validation, the same strategy is used. The setup with optical flow corrected images gives a PSNR-L of 41.53 dB, which is comparable to existing state-of-the-art methods (Table V). This shows that our model can provide high quality HDR images even if the input LDR images are not sorted by exposure time. We have also

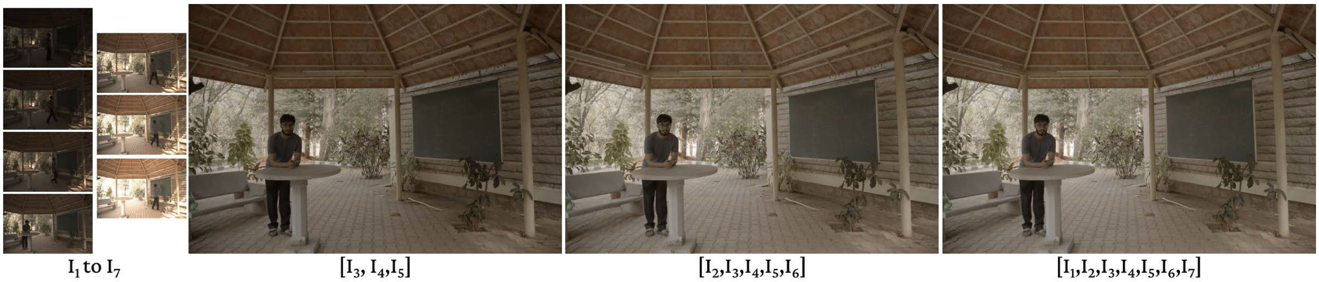
I_1 to I_7
$[I_3, I_4, I_5]$
$[I_2, I_3, I_4, I_5, I_6]$
$[I_1, I_2, I_3, I_4, I_5, I_6, I_7]$

Fig. 9. An qualitative example to highlight effect of adding more LDRs to the network. Left column: Seven input varying exposure images with foreground motion. Second column image: output by proposed method by fusing middle three images (I_3, I_4, I_5). Third column image: By fusing middle five images I_2 to I_6. Last image: By fusing all seven images.

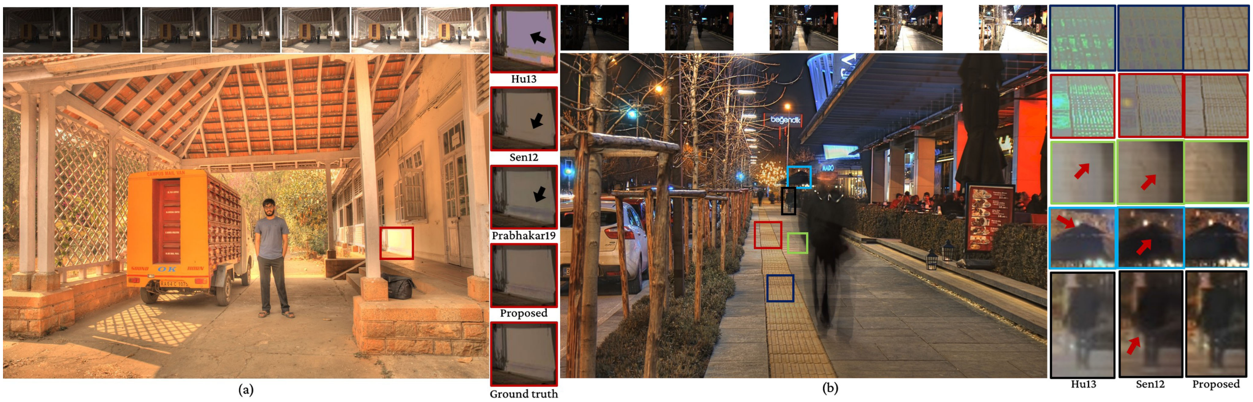

Fig. 10. A qualitative example from (a) Prabhakar19 dataset [20] and (b) METU dataset [25].

trained a model with 5 images in random order which achieved a PSNR-L of 43.62 dB versus 43.75 dB without shuffling.

SGM cell: We also conduct experiments regarding the structure of the proposed SGM Cell (Table V). They are not included as separate Type Ablations, since they are minimal changes on top of the final version of the cell. In the first experiment, we replace the swish activation at the output gate with the Tanh activation (change in equations 19 and 24), and observe a drop of 0.94 dB in PSNR-L. In another ablation experiment, we remove the Swish activations and use the outputs of the gates in equations (15), (16) and (19) (and corresponding reverse equations) directly without applying any nonlinearity. This experiment resulted in a drop of 0.63 dB in PSNR-L.

E. Running Times

In Table VI, we present the running time comparison between existing methods and the proposed method for variable length sequences. Our approach can fuse a sequence with three images of resolution 1000 × 1500 in 0.03 seconds on a NVIDIA Quadro RTX 6000 GPU and a Intel Core i7 3.00 GHz CPU. Also, our method is atleast 17 × faster than other scalable HDR deghosting approaches.

TABLE VI
RUNNING TIME (IN SECONDS) AND NUMBER OF PARAMETERS COMPARISON BETWEEN EXISTING METHODS AND PROPOSED METHOD

		Number of Images			Parameters (Millions)	
	Architecture	3	5	7	Network	Cell
	Sen12	146.0	496	1187	-	-
	Hu13	264.2	550	798	-	-
	Kalantari17	50.02	-	-	0.38	-
	Wu18	6.800	-	-	16.61	-
	Yan19	0.096	-	-	1.44	-
	Prabhakar19	0.663	1.274	1.873	12.21	-
Proposed	Bi-LSTM	0.042	0.058	0.075	1.30	0.33
	Bi-GRU	0.032	0.049	0.059	1.12	0.22
	Bi-SGM	0.037	0.052	0.072	1.19	0.26

V. CONCLUSION

We propose a efficient scalable HDR deghosting method based on recurrent neural networks. In our approach, we introduce a novel Self-Gated Memory recurrent cell that can control the information flow by gating the output with a function of itself. The SGM cell offers high accuracy than standard GRU and has less parameters than LSTM cell. We utilize SGM cell in bi-directional setting to achieve better performance and visually pleasing results. The major advantage of our proposed method compared to many existing deep learning based HDR deghosting methods, is that our approach can fuse arbitrary length sequence without a need for re-training. Further, we demonstrate the superiority of our method over existing state-of-the-art methods on three publicly available datasets.

REFERENCES

[1] 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, 2011.

[2] H. Zhao, B. Shi, C. Fernandez-Cull, S.-K. Yeung, and R. Raskar, "Unbounded high dynamic range photography using a modulo camera," in *Proc. IEEE Int. Conf. Comput. Photography*, 2015, pp. 1–10.

[3] P. E. Debevec and J. Malik, "Recovering high dynamic range radiance maps from photographs," in *Proc. ACM SIGGRAPH 2008 Classes*, New York, NY, USA, 2008, Art. no. 31.

[4] T. Grosch, "Fast and robust high dynamic range image generation with camera and object movement," *Vis., Model. Visualization, RWTH Aachen*, pp. 277–284, 2006.

[5] S. Wu, S. Xie, S. Rahardja, and Z. Li, "A robust and fast anti-ghosting algorithm for high dynamic range imaging," in *Proc. IEEE Int. Conf. Image Process.*, 2010, pp. 397–400.

[6] O. Gallo, N. Gelfandz, W.-C. Chen, M. Tico, and K. Pulli, "Artifact-free high dynamic range imaging," in *Proc. IEEE Int. Conf. Comput. Photography*, 2009, pp. 1–7.

[7] T.-H. Min, R.-H. Park, and S. Chang, "Histogram based ghost removal in high dynamic range images," in *Proc. IEEE Int. Conf. Multimedia Expo*, 2009, pp. 530–533.

[8] Y. S. Heo, K. M. Lee, S. U. Lee, Y. Moon, and J. Cha, "Ghost-free high dynamic range imaging," in *Proc. Asian Conf. Comput. Vis.*, 2010, pp. 486–500.

[9] S. Raman and S. Chaudhuri, "Reconstruction of high contrast images for dynamic scenes," *Vis. Comput.*, vol. 27, no. 12, pp. 1099–1114, 2011.

[10] A. Tomaszewska and R. Mantiuk, "Image registration for multiexposure high dynamic range image acquisition," in *Proc. Int. Conf. Comput. Graph., Visual. Comput. Vis.*, 2007, pp. 49–56.

[11] L. Bogoni, "Extending dynamic range of monochrome and color images through fusion," in *Proc. 15th Int. Conf. Pattern Recognit.*, 2000, pp. 7–12.

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

[13] 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, 2003.

[14] O. Gallo, A. Troccoli, J. Hu, K. Pulli, and J. Kautz, "Locally non-rigid registration for mobile HDR photography," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2015, pp. 49–56.

[15] 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. 203–214, 2012.

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

[17] K. Ma, H. Li, H. Yong, Z. Wang, D. Meng, and L. Zhang, "Robust multi-exposure image fusion: A structural patch decomposition approach," *IEEE Trans. Image Process.*, vol. 26, no. 5, pp. 2519–2532, May 2017.

[18] N. K. Kalantari and R. Ramamoorthi, "Deep high dynamic range imaging of dynamic scenes," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 144–156, 2017.

[19] 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.*, 2018, pp. 120–135.

[20] 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*, 2019, pp. 1–8.

[21] Q. Yan *et al.*, "Multi-scale dense networks for deep high dynamic range imaging," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2019, pp. 41–50.

[22] K. R. Prabhakar, S. Agrawal, D. Singh, B. Ashwath, and R. V. Babu, "Towards practical and efficient high-resolution HDR deghosting with CNN," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 497–513.

[23] P. Sen and C. Aguerrebere, "Practical high dynamic range imaging of everyday scenes: Photographing the world as we see it with our own eyes," *IEEE Signal Process. Mag.*, vol. 33, no. 5, pp. 36–44, Sep. 2016.

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

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

[26] L. Cerman and V. Hlavac, "Exposure time estimation for high dynamic range imaging with hand held camera," in *Proc. Comput. Vis. Winter Workshop*, 2006, pp. 76–81.

[27] A. A. Rad, L. Meylan, P. Vandewalle, and S. Süsstrunk, "Multidimensional image enhancement from a set of unregistered differently exposed images," *Proc. Comput. Imag. V*, International Society for Optics and Photonics, vol. 6498, 2007, Art. no. 649808.

[28] S. Yao, "Robust image registration for multiple exposure high dynamic range image synthesis," *Proc. Image Process., Algorithms Syst. IX*, International Society for Optics and Photonics, vol. 7870, 2011, Art. no. 78700Q.

[29] J. Im, S. Jang, S. Lee, and J. Paik, "Geometrical transformation-based ghost artifacts removing for high dynamic range image," in *Proc. 18th IEEE Int. Conf. Image Process.*, 2011, pp. 357–360.

[30] J. Im, S. Lee, and J. Paik, "Improved elastic registration for removing ghost artifacts in high dynamic imaging," *IEEE Trans. Consum. Electron.*, vol. 57, no. 2, pp. 932–935, May 2011.

[31] M. Gevrekci and B. K. Gunturk, "On geometric and photometric registration of images," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2007, pp. I- 1261.

[32] A. O. Akyüz, "Photographically guided alignment for HDR images," in *Proc. Eurograph.*, 2011, pp. 73–74, doi: 10.2312/EG2011/areas/073-074.

[33] S. Mann, C. Manders, and J. Fung, "Painting with looks: Photographic images from video using quantimetric processing," in *Proc. 10th ACM Int. Conf. Multimedia*, 2002, pp. 117–126.

[34] F. M. Candocia, "Simultaneous homographic and comparametric alignment of multiple exposure-adjusted pictures of the same scene," *IEEE Trans. Image Process.*, vol. 12, no. 12, pp. 1485–1494, Dec. 2003.

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

[36] A. Srikantha, D. Sidibé, and F. Mériaudeau, "An SVD-based approach for ghost detection and removal in high dynamic range images," in *Proc. Int. Conf. Pattern Recognit.*, 2012, pp. 380–383.

[37] H.-S. Sung, R.-H. Park, D.-K. Lee, and S. Chang, "Feature based ghost removal in high dynamic range imaging," *Int. J. Comput. Graph. Animation*, vol. 3, no. 4, pp. 23–39, 2013.

[38] J. An, S. H. Lee, J. G. Kuk, and N. I. Cho, "A multi-exposure image fusion algorithm without ghost effect," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2011, pp. 1565–1568.

[39] D.-K. Lee, R.-H. Park, and S. Chang, "Improved histogram based ghost removal in exposure fusion for high dynamic range images," in *Proc. IEEE 15th Int. Symp. Consum. Electron.*, 2011, pp. 586–591.

[40] Z. Li, S. Rahardja, Z. Zhu, S. Xie, and S. Wu, "Movement detection for the synthesis of high dynamic range images," in *Proc. IEEE Int. Conf. Image Process.*, 2010, pp. 3133–3136.

[41] H.-Y. Lin and W.-Z. Chang, "High dynamic range imaging for stereoscopic scene representation," in *Proc. 16th IEEE Int. Conf. Image Process.*, 2009, pp. 4305–4308.

[42] W. Zhang and W.-K. Cham, "Gradient-directed multiexposure composition," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 2318–2323, Apr. 2012.

[43] S. Silk and J. Lang, "Fast high dynamic range image deghosting for arbitrary scene motion," in *Proc. Graph. Interface*, 2012, pp. 85–92.

[44] W. Zhang and W.-K. Cham, "Gradient-directed composition of multi-exposure images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 530–536.

[45] E. A. Khan, A. Akyiiz, and E. Reinhard, "Ghost removal in high dynamic range images," in *Proc. IEEE Int. Conf. Image Process.*, 2006, pp. 2005–2008.

[46] M. Pedone and J. Heikkilä, "Constrain propagation for ghost removal in high dynamic range images," *Proc. 3rd Int. Conf. Comput. Vision Theory Appl.*, vol. 1, pp. 36–41, 2008.

[47] F. Pece and J. Kautz, "Bitmap movement detection: HDR for dynamic scenes," in *Proc. Conf. Vis. Media Prod.*, 2010, pp. 1–8.

[48] A. Eden, M. Uyttendaele, and R. Szeliski, "Seamless image stitching of scenes with large motions and exposure differences," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2006, pp. 2498–2505.

[49] M. Granados, B. Ajdin, M. Wand, C. Theobalt, H.-P. Seidel, and H. P. Lensch, "Optimal HDR reconstruction with linear digital cameras," in *Proc. Comput. Vis. Pattern Recognit.*, 2010, pp. 215–222.

[50] E. Reinhard and K. Devlin, "Dynamic range reduction inspired by photoreceptor physiology," *IEEE Trans. Vis. Comput. Graphics*, vol. 11, no. 1, pp. 13–24, Jan./Feb. 2005.

[51] K. Jacobs, C. Loscos, and G. Ward, "Automatic high-dynamic range image generation for dynamic scenes," *IEEE Comput. Graph. Appl.*, vol. 28, no. 2, pp. 84–93, Mar./Apr. 2008.

[52] D. Sidibé, W. Puech, and O. Strauss, "Ghost detection and removal in high dynamic range images," in *Proc. 17th Eur. Signal Process. Conf.*, 2009, pp. 2240–2244.

[53] T.-H. Oh, J.-Y. Lee, Y.-W. Tai, and I. S. Kweon, "Robust high dynamic range imaging by rank minimization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 6, pp. 1219–1232, Jun. 2015.

[54] S. Ferradans, M. Bertalmío, E. Provenzi, and V. Caselles, "Generation of HDR images in non-static conditions based on gradient fusion," *Proc. Int. Conf. Comput. Vision Theory Appl.*, vol. 2, pp. 31–37, 2012.

[55] T. Jinno and M. Okuda, "Multiple exposure fusion for high dynamic range image acquisition," *IEEE Trans. Image Process.*, vol. 21, no. 1, pp. 358–365, Jan. 2012.

[56] D. Hafner, O. Demetz, and J. Weickert, "Simultaneous HDR and optic flow computation," in *Proc. 22nd Int. Conf. Pattern Recognit.*, 2014, pp. 2065–2070.

[57] N. Menzel and M. Guthe, "Freehand HDR photography with motion compensation," in *Proc. Vision, Model., Visualization Conf., Saarbrücken*, 2007, pp. 127–134.

[58] S.-C. Park, H.-H. Oh, J.-H. Kwon, W. Choe, and S.-D. Lee, "Motion artifact-free HDR imaging under dynamic environments," in *Proc. 18th IEEE Int. Conf. Image Process.*, 2011, pp. 353–356.

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

[60] Q. Yan *et al.*, "Deep HDR imaging via a non-local network," *IEEE Trans. Image Process.*, vol. 29, pp. 4308–4322, Feb. 2020, doi: 10.1109/TIP.2020.2971346.

[61] M. Gharbi, J. Chen, J. T. Barron, S. W. Hasinoff, and F. Durand, "Deep bilateral learning for real-time image enhancement," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 1–12, 2017.

[62] Y. Zhao, X. Jin, and X. Hu, "Recurrent convolutional neural network for speech processing," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2017, pp. 5300–5304.

[63] S. Pascual and A. Bonafonte, "Multi-output RNN-LSTM for multiple speaker speech synthesis and adaptation," in *Proc. 24th Eur. Signal Process. Conf.*, 2016, pp. 2325–2329.

[64] G. Gelly and J.-L. Gauvain, "Optimization of RNN-based speech activity detection," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 26, no. 3, pp. 646–656, Mar. 2018.

[65] T. Mikolov and G. Zweig, "Context dependent recurrent neural network language model," in *Proc. IEEE Spoken Lang. Technol. Workshop*, 2012, pp. 234–239.

[66] M. Sundermeyer, T. Alkhouli, J. Wuebker, and H. Ney, "Translation modeling with bidirectional recurrent neural networks," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2014, pp. 14–25.

[67] S. Ghosh, O. Vinyals, B. Strope, S. Roy, T. Dean, and L. Heck, "Contextual LSTM: A step towards hierarchical language modeling," in *Proc. KDD Workshop Large-Scale Deep Learn. Data Mining (DL-KDD)*, 2016, pp. 31–41.

[68] W. Pei, J. Zhang, X. Wang, L. Ke, X. Shen, and Y.-W. Tai, "Memory-attended recurrent network for video captioning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 8347–8356.

[69] H. Yu, J. Wang, Z. Huang, Y. Yang, and W. Xu, "Video paragraph captioning using hierarchical recurrent neural networks," in *Proc. IEEE Conf. Comput. Vis. pattern Recognit.*, 2016, pp. 4584–4593.

[70] Y.-T. Hu, J.-B. Huang, and A. G. Schwing, "MaskRNN: Instance level video object segmentation," *Advances Neural Inf. Process. Syst.*, vol. 2017, pp. 325–334, 2017, *arXiv:1803.11187*.

[71] S. Valipour, M. Siam, M. Jagersand, and N. Ray, "Recurrent fully convolutional networks for video segmentation," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2017, pp. 29–36.

[72] C. Godard, K. Matzen, and M. Uyttendaele, "Deep burst denoising," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 538–554.

[73] X. Chen, L. Song, and X. Yang, "Deep RNNs for video denoising," *Proc. Appl. Digit. Image Process. XXXIX*, International Society for Optics and Photonics, vol. 9971, 2016, Art. no. 99711T.

[74] R. Schuster, O. Wasenmuller, C. Unger, and D. Stricker, "SDC-stacked dilated convolution: A unified descriptor network for dense matching tasks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2556–2565.

[75] G. Krawczyk, K. Myszkowski, and H.-P. Seidel, "Lightness perception in tone reproduction for high dynamic range images," *Proc. Comput. Graph. Forum*, vol. 24, no. 3, pp. 635–646, 2005.

[76] F. Durand and J. Dorsey, "Fast bilateral filtering for the display of high-dynamic-range images," *ACM Trans. Graph.*, vol. 21, no. 3, pp. 257–266, 2002.

[77] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Representations*, May 2015. [Online]. Available: http://arxiv.org/abs/1412.6980

[78] K. R. Prabhakar and R. V. Babu, "High dynamic range deghosting dataset," *Data Retrieved From IISc VAL Website*. 2020, [Online]. Available: https://val.cds.iisc.ac.in/HDR/HDRD/

[79] K. Karaduzovic-Hadziabdic, T. J. Hasic, and R. K. Mantiuk, "Multi-exposure image stacks for testing HDR deghosting methods," 2017. [Online]. Available: https://www.repository.cam.ac.uk/bitstream/handle/1810/261766/pairwise_comparison_results.zip?sequence=7

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

[81] Y. Endo, Y. Kanamori, and J. Mitani, "Deep reverse tone mapping," *ACM Trans. Graph.*, vol. 36, no. 6, pp. 177–1, 2017.

[82] G. Eilertsen, J. Kronander, G. Denes, R. K. Mantiuk, and J. Unger, "HDR image reconstruction from a single exposure using deep CNNs," *ACM Trans. Graph.*, vol. 36, no. 6, pp. 178–193, 2017.

[83] 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. 40–54, 2011.

[84] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[85] E. Reinhard, M. Stark, P. Shirley, and J. Ferwerda, "Photographic tone reproduction for digital images," *ACM Trans. Graph.*, vol. 21, no. 3, pp. 267–276, 2002.

K. Ram Prabhakar received the bachelor's degree from Anna University in 2010, the master's degree from NIT Rourkela in 2014, and the Ph.D. degree from the Department of Computational and Data Sciences, Indian Institute of Science, Bangalore, India. He has broad-ranging research interests and likes to work on problems involving computational photography, machine learning, and multi-modal multimedia data processing (e.g., images, videos, text, audio/speech).

Susmit Agrawal received the B.E. degree in computer science and engineering from Visvesvaraya Technological University, Karnataka, India, in 2020. He is currently a Researcher with Video Analytics Lab, Indian Institute of Science. His research interests include computational photography and generative models.

R. Venkatesh Babu (Senior Member, IEEE) received the Ph.D. degree from the Department of Electrical Engineering, Indian Institute of Science (IISc), Bengaluru, India. He held Postdoctoral positions with the NTNU, Norway, and IRISA/INRIA, France. He was a Research Fellow with Nanyang Technological University, Singapore. He is currently a Professor with the Department of Computational and Data Sciences, IISc. His research interests include span computer vision, machine learning and image/video processing. He is the recipient of SERB-STAR Award (2020).