IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 23 February 2024, accepted 23 March 2024, date of publication 1 April 2024, date of current version 9 April 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3383928

RESEARCH ARTICLE

A Compact Multi-Exposure File Format for Backward and Forward Compatible HDR Imaging

SELIN SEKMEN[1,2] **AND AHMET OĞUZ AKYÜZ**[1]
[1]Department of Computer Engineering, Middle East Technical University, 06800 Ankara, Turkey
[2]Aselsan Inc., 06750 Ankara, Turkey

Corresponding author: Ahmet Oğuz Akyüz (akyuz@ceng.metu.edu.tr)

ABSTRACT High dynamic range (HDR) imaging techniques offer photographers the ability to capture the full range of luminance in real-world scenes, overcoming the limitations of capture and display devices. One popular method for creating HDR images is the multiple exposures technique, which involves capturing multiple exposures with regular digital cameras and combining them later to generate an HDR image. In this work, we propose a method called Residual Compressed Exposure Sequences (ResCES) that aims to consolidate all the information from a bracketed sequence into a single JPEG file. Typically, the main image that is to be displayed by a standard image viewer is selected as the middle exposure of the sequence, although any other user-preferred exposure can be selected as well. When needed, the original exposures can be reconstructed from this single JPEG file, enabling their use in a standard HDR workflow. Our proposed approach utilizes a patch-based process, where we store under-exposed, over-exposed, and motion-detected patches while reconstructing other patches through the camera response function to minimize data loss. To further improve the fidelity of the reconstructed exposures, we employ a residual learning model in the last stage of our pipeline, effectively eliminating any artifacts that may occur in its earlier stages. The key innovation of ResCES is its ability to encapsulate the complete set of original exposures within a single JPEG file in an efficient manner, allowing for on-demand reconstruction – a feature that distinguishes it from existing HDR file formats in the literature. The experimental results demonstrate that ResCES achieves a high degree of similarity with respect to the original exposures, as shown by both quantitative and qualitative evaluations. The subjective visual evaluation conducted using 40 participants indicates that ResCES reconstruction results are statistically indistinguishable from the original exposures, while, on average, yielding a 4.5 times storage reduction. This, coupled with the ease of file maintenance, simplifies storing, sharing, and viewing of HDR images.

INDEX TERMS HDR, multi exposure, JPEG, metadata, residual learning.

I. INTRODUCTION

The pursuit of more accurate, visually compelling, and scene-referred representations of the real world has driven the development of HDR imaging [1], [2]. By expanding the range of luminance levels that can be faithfully represented, HDR imaging promises to deliver more realistic, immersive, and aesthetically pleasing visual content [3]. Today, the most commonly used method for generating HDR images remains to be the multiple exposures technique, in which a bracketed sequence of exposures are merged into a single HDR image. This process is typically accomplished by the

The associate editor coordinating the review of this manuscript and approving it for publication was Amin Zehtabian.

following equation:

$$E(x,y) = \frac{\sum_{i=1}^{N} w(Z_i(x,y)) \dfrac{f^{-1}(Z_i(x,y))}{\Delta t_i}}{\sum_{i=1}^{N} w(Z_i(x,y))}, \qquad (1)$$

where Z indicates the non-linear input exposures, N is their count, Δt are the exposure times, w is a weighting function, f is the camera response function (CRF), and E is the final irradiance estimate at pixel position (x, y) [4], [5], [6].

HDR images can be created from multiple exposures by other means as well. For instance, in exposure fusion, multiple exposures are fused into a single HDR image by using Laplacian blending using well-exposedness, contrast, and saturation as blending weights [7]. Recent work focuses on developing neural networks that can either learn blending weights or directly the final HDR image [8].

Regardless of which technique is used to create an HDR image, using multiple exposures is a key part of the process. In the following, we first motivate the need to maintain these multiple input exposures and not just the final HDR image. We then provide an overview of our algorithm. After summarizing the related literature we then elaborate two variants of our algorithm where the first one is the baseline method that does not utilize learning, and the second one is a more sophisticated, learning-based approach called ResCES. Finally, we share our quantitative/qualitative evaluation results and demonstrate a use-case that highlights the motivating principle behind our algorithm.

A. MOTIVATION

One of the primary challenges associated with multiple exposure techniques is the requirement to store multiple images for each captured scene. Typically, photographers capture a series of 3 to 9 exposures. Although it is possible to discard the individual exposures once the HDR image has been generated, there are several reasons for retaining them.

Firstly, most display devices used for viewing images have limited dynamic range, meaning that they are unable to accurately reproduce the full range of luminance captured in an HDR image. By storing the corresponding LDR versions of the exposures, users can quickly and conveniently view the captured scene without resorting to the process of tone mapping, which is required to prepare HDR images for display on LDR display devices [9], [10], [11], [12], [13].

Secondly, the process of capturing bracketed exposures may result in misalignment artifacts due to camera and/or object motion. Although these artifacts could be mitigated by the application of a deghosting algorithm [14], [15], [16], complete elimination of ghosting artifacts remains to be an elusive problem. As such, photographers may choose to preserve the input exposures and use them to create an HDR image with better quality as new and improved deghosting algorithms emerge. An illustrative example is shown in Figure 1, where a person has been captured while waving his hand in front of a window using a modern smartphone. The

FIGURE 1. An image produced by a modern smartphone camera. The hand area is affected by ghosting artifacts, and its background is saturated (compare the middle window to the left one). Had the individual exposures been saved efficiently in the produced camera output, a better offline reconstruction would have been possible, which is a motivating factor for our study.

"HDR merge" algorithm that runs on the phone ISP fails to produce a satisfactory result for the hand region, making the fingers totally disappear. It also fails to reproduce details in that window (compare the middle window to the left one). Had the individual exposures been saved efficiently in the produced image file, a better offline reconstruction would have been possible.

Similar arguments can be made for other components of the HDR imaging pipeline such as the recovery of the CRF and the selection of the weighting function used during the combination of the pixel values (Equation 1). Other decisions may include denoising of the input exposures [17], [18], [19], as well as whether exposure fusion should be employed instead of creating an HDR image [7], [20], [21], [22]. In each of these areas, new and improved algorithms appear on a regular basis from which photographers can benefit only if they retain the original input exposures [23], [24].

Thus, the main problem that we aim to solve in this paper is an efficient storage scheme for multiple exposures, which allows the photographer to retain the entirety of information present in the bracketed sequence in a single compact JPEG file.

B. OVERVIEW OF THE PROPOSED SCHEME

To address this challenge, this paper proposes a novel method to reduce the storage requirements of exposure sequences. To this end, we propose a new multi-exposure and backward compatible file format stored inside a single JPEG file. The primary image, which can be displayed by any image viewer, is a user-selected reference exposure. Rather than directly storing the remaining exposures, however, the technique divides them into equal sized patches for efficient storage.

Each patch is selectively stored or omitted based on its availability in the reference.

While the storage optimization achieved through patch-based selective storage significantly reduces the file size requirements, there is a potential drawback associated with the reconstruction of non-stored patches. The accuracy of the reconstruction heavily relies on the CRF, which characterizes the relationship between pixel values in the captured image and the corresponding scene radiance [25]. However, the inaccuracies in the CRF recovery may lead to undesirable artifacts in the reconstructed exposures. To overcome this limitation, the proposed method incorporates a deep residual network, which is designed to specifically address the reconstruction artifacts caused by the CRF and generate exposures that closely resemble the original ones.

II. RELATED WORK

In this section, we first review the HDR capture technologies that are commonly used today and argue that multiple-exposure based HDR imaging techniques still remain to be the norm rather than exception. We then summarize the existing HDR file formats that are mostly related to our proposed storage scheme.

A. HDR CAPTURE

While a multitude of approaches are available for the creation of HDR images and videos, one notable method involves the deployment of specialized HDR capture hardware [26], [27]. These devices equipped with HDR-capable sensors hold the potential to yield exceptional HDR content in future. Nevertheless, the widespread adoption of these advanced HDR capture tools remains elusive for most users due to their limited availability and high cost.

This disparity in access to high-quality HDR content creation tools has been evident in the work of Mukherjee et al. [28], who explored the feasibility of training object detectors directly with HDR images. Faced with the scarcity of HDR training data, they resorted to creating a pseudo-HDR dataset by applying a dynamic range expansion operator [29], [30] to a collection of low dynamic range (LDR) images, with a small subset of authentic HDR images employed for evaluation purposes. Indeed, while using inverse tone mapping approaches can be considered as an HDR content creation method, they hallucinate missing details rather than faithfully reconstructing them [31], [32], [33].

Other recent HDR capture methods involve creation of HDR images using neuromorphic (i.e. event) cameras [34], [35]. Due to high temporal resolution of these cameras, any high frequency change in the scene can be detected as events and this information can be used either in isolation or together with accompanying RGB data to create HDR images/videos [36], [37].

Despite the promise of these hardware solutions, their widespread adoption is expected to take time and therefore multiple-exposure based HDR capture solutions remains to be more widely used. Even if modern smartphone cameras are used in the ''HDR mode'', they internally resort to multiple exposures techniques and only show the combined image to the user [38]. The multiple-exposures techniques, which is pioneered by the works of [4], [5], and [6], is among one of the most highly studied topics of HDR imaging. Recent work focuses on reconstructing high quality HDR images/videos using learning-based approaches [39], [40], [41]. It is this rapid progress in multi-exposure HDR imaging that motivates our work: it is important to preserve an exposure bracketed sequence efficiently so that it can be used to create higher quality HDR images as new and improved multi-exposure based methods become available.

B. HDR STORAGE

HDR content is renowned for its ability to provide enhanced contrast and an expanded color gamut by increasing the range of luminance values it can represent [42]. However, the increased dynamic range necessitates higher bit depths to accurately encode the data, resulting in higher storage and transmission costs compared to LDR content. To address these challenges, several image and video encoding methods have been proposed in the literature.

One of the notable approaches in HDR image compression is the JPEG 2000 standard [43], which is a powerful image compression method that offers improved compression performance and supports higher bit depths compared to the original JPEG format. Similarly, the JPEG XR standard, presented by Dufaux et al. [44], is another approach that aims to overcome the bit-depth limitations of JPEG. JPEG XR, formerly known as HD Photo or Windows Media Photo, provides support for high dynamic range imaging and exhibits improved compression efficiency. However, despite their advantages, JPEG 2000 and JPEG XR have not gained widespread adoption in the realm of digital photography. The primary reason for their limited success is the lack of backward compatibility with the dominant legacy JPEG format, which is still widely used for storing digital images.

To address this compatibility issue, researchers have explored the utilization of existing JPEG standards and additional metadata for storing HDR content. The JPEG standards include marker segments that can be used to store application-specific data within the JPEG metadata sections [45], [46]. These application-specific markers enhance the functionality of JPEG images by providing essential data for image management, analysis, and interpretation [47], [48].

Leveraging this feature, Ward et al. [49] introduced JPEG-HDR, an extension of the standard JPEG format specifically designed to store HDR images in a manner that allows accurate conversion to both HDR and LDR representations. This backward-compatible approach aims to facilitate the adoption of HDR imaging while ensuring compatibility with existing software and workflows. Similarly, for HDR video

streams, Mantiuk et al. proposed MPEG-HDR as an efficient backward-compatible video compression technique [50].

In addition to standardized approaches, various formats have been proposed specifically for HDR images such as, RGBE [51], LogLuv [52], and OpenEXR [53]. While these formats offer advanced capabilities and flexibility for HDR content, they often lack backward compatibility with existing software, limiting their practicality for widespread adoption.

The most recent advancements in HDR image and video technology are JPEG XT, Dolby Vision, HDR10, and HDR10+ standards. JPEG XT, as described by Artusi et al. [54], extensively utilizes metadata to store HDR images within a JPEG file. This approach offers backward compatibility with the widely recognized JPEG format while providing both lossy and lossless compression options for HDR images. Dolby Vision [55], on the other hand, is designed for a visually immersive HDR video experience by allowing the metadata to dynamically change each frame. HDR10 [56] and HDR10+ [57] are open standards with the former being limited to static metadata and the latter supporting dynamic metadata similar to Dolby Vision.

In addition to these, Hybrid Log-Gamma (HLG) [58], is a pioneering HDR technology developed by the BBC (British Broadcasting Corporation) and NHK (Japan Broadcasting Corporation), specifically tailored for live broadcasting and streaming. It stands out by being compatible with both LDR and HDR displays, eliminating the need for static metadata. Furthermore, SMPTE ST 2094 introduces the technology of Single Layer High Dynamic Range (SL-HDR) in three versions: SL-HDR1 [59], SL-HDR2 [60], and SL-HDR3 [61]. Each version plays a unique role in advancing HDR technology. SL-HDR1 enhances the HDR experience through scene-by-scene dynamic metadata, while SL-HDR2 builds upon this by incorporating advanced color grading techniques. Taking it to the next level, SL-HDR3 further improves HDR for displays with heightened brightness and contrast capabilities.

Overall, these various approaches and standards in HDR image and video compression demonstrate ongoing efforts to balance the increased demands of higher dynamic range content with backward compatibility and efficient storage solutions, aiming to facilitate the adoption and practicality of HDR technology in various domains.

The proposed method builds upon the existing backward-compatible storage schemes for HDR imaging. However, it differs from previous approaches that store a tone-mapped image as the primary representation along with auxiliary data for HDR image reconstruction. Instead, the proposed method stores the complete information captured in an exposure sequence by minimizing redundancy. By efficiently storing the original sequence, this approach simplifies the maintenance and usability of bracketed exposure sequences and enables the creation of an HDR image at any point in the future to benefit from ever-evolving HDR reconstruction algorithms.

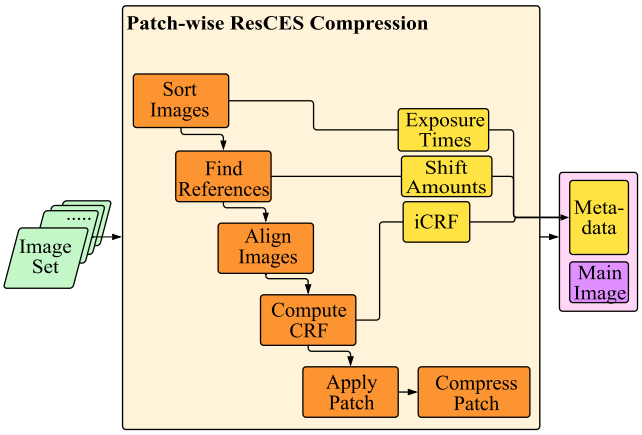

FIGURE 2. The compression pipeline of the patch-based CES algorithm.

III. COMPRESSED EXPOSURE SEQUENCES (CES)

The basic idea of the CES algorithm was outlined in our earlier work [62]. Here, we briefly review its key aspects and focus on the patch-based approach that makes this algorithm more useful and efficient. The core of the CES algorithm is comprised of two primary pipelines, namely compression and decompression. In the compression pipeline, we consolidate the information from multiple exposures into a single image to achieve a concise representation of the exposure sequence. Conversely, the decompression pipeline extracts the main image and its accompanying metadata from the JPEG file to subsequently reconstruct the original exposures.

A. COMPRESSION PIPELINE

A visual representation of the compression pipeline is shown in Figure 2. This pipeline is comprised of image sorting, reference determination, alignment, CRF computation, patchification, and patch compression stages. Our full workflow is detailed in Algorithm 1. Below, we explain these main stages.

1) IMAGE SORTING

Firstly, we load the image set and arrange all the exposures based on their respective exposure times. This results in a sequence denoted as $\langle I_1, I_2, \ldots, I_m, \ldots, I_n \rangle$, where the exposures are sorted in ascending order from the shortest to the longest exposure time. The middle (i.e., the median) exposure is represented by I_m. Each exposure in this set consists of 24 bits per pixel, allocating 8 bits per color channel. The corresponding exposure values are similarly arranged as $\langle E_1, E_2, \ldots, E_m, \ldots, E_n \rangle$.

2) REFERENCE IMAGE SELECTION

The ideal reference image would be the one that closely represents the scene's lighting conditions, colors, and overall visual characteristics. In this work, we select the middle exposure as the main reference exposure, assuming that it is the most balanced one in terms of pixel intensity distribution.

Algorithm 1 Compression Process of Patch-Based CES

procedure CESCompression(I_{List}, N) ▷ I_{List} is the set of exposures, N is the number of exposures in the set
  $\langle I_1, I_2, \ldots, I_m, \ldots I_N \rangle \leftarrow SortImageList(I_{List})$
  $\langle E_1, E_2, \ldots, E_m, \ldots E_N \rangle \leftarrow GetExposureTimes(I_{List})$
  $m \leftarrow median(1 \ldots N)$
  for each image x in I_{List} **do**
    $I_{ref_x} \leftarrow FindReference(I_x)$ ▷ Equation 2
    $Shift_x \leftarrow AlignImage(I_x, I_{ref_x})$
  $f \leftarrow ComputeCRF(I_{List})$
  for each image x in I_{List} **do**
    $L_x \leftarrow Linearize(I_x)$ ▷ Equation 3
  for each image x in I_{List} **do**
    $L_{dif_x} \leftarrow FindDifference(L_x, L_{ref_x})$ ▷ Equation 4
  for each image x in I_{List} except I_m **do**
    for each patch p in I_x **do**
      if $x < m$ **then**
        if $mean(p_{ref_x}) > 225$ **then**
          $P_{x_{oe}} \leftarrow Add(p_x)$ ▷ over-exposed
        else if $variance(L_{dif_{x_p}}) > 25$ **then**
          $P_{x_{md}} \leftarrow Add(p_x)$ ▷ motion-detected
        else
          $P_{x_s} \leftarrow Add(p_x)$ ▷ standard
      else
        if $mean(p_{ref_x}) < 25$ **then**
          $P_{x_{ue}} \leftarrow Add(p_x)$ ▷ under-exposed
        else if $variance(L_{dif_{x_p}}) > 25$ **then**
          $P_{x_{md}} \leftarrow Add(p_x)$ ▷ motion-detected
        else
          $P_{x_s} \leftarrow Add(p_x)$ ▷ standard
  $C_{data} \leftarrow Compress(P_{\langle 1..N \rangle_{oe}}, P_{\langle 1..N \rangle_{ue}}, P_{\langle 1..N \rangle_{md}})$
  $Metadata \leftarrow \{C_{data}, f, Shift_{\langle 1..N \rangle}, E_{\langle 1..N \rangle}, N\}$
  $JPEGFile \leftarrow \{I_m, Metadata\}$
  return $JPEGFile$

This exposure will be shown as the main image by a standard image viewer.

The reference image for a particular exposure, x, is determined by selecting an adjacent exposure in terms of exposure time in the direction of the median exposure, m, as illustrated in Figure 3 and shown by the following equation:

$$I_{ref_x} = \begin{cases} I_{x+1} & \text{if } x < m \\ I_{x-1} & \text{if } x > m, \end{cases} \tag{2}$$

where I_{x+1} denotes the subsequent image following the image with index x, I_{x-1} represents the preceding image before the image with index x, and I_{ref_x} designates the reference image for the image with index x.

By selecting exposures that are closely related in terms of their exposure time, we maximize the coherency between them, which ultimately results in better quality and compression efficiency.

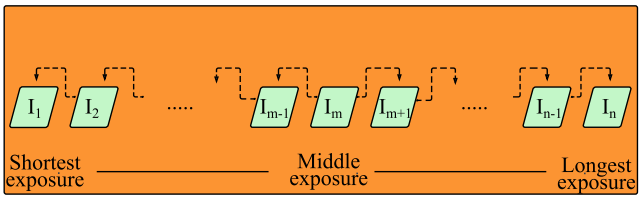

FIGURE 3. The reference of each exposure is the neighboring one in the direction of the middle exposure. This choice improves temporal and brightness coherency between them.

3) IMAGE ALIGNMENT

The aim of this stage is to align each exposure with its corresponding reference image in order to compensate for any camera movement that may have occurred during the image capture process. To accomplish this, we employ the enhanced correlation coefficient (ECC) image alignment algorithm [63]. Unlike conventional approaches that rely on pixel intensity differences as a similarity measure, ECC is capable of handling photometric distortions related to contrast and brightness, making it suitable to be used between images with different exposure values.

The ECC algorithm addresses the alignment task by formulating an objective function, which, although nonlinear in terms of the parameters, can be efficiently solved using an iterative scheme that is ultimately linear. This means that despite the apparent computational complexity of the problem, the algorithm discovers a simplified iterative solution. Consequently, we opted for the ECC algorithm in our pipeline due to its simplicity, efficiency, and ability to handle exposure differences while maintaining robustness.

4) CRF COMPUTATION

Once the alignment process is complete, we proceed to find the CRF, f, using the aligned exposures in order to linearize them. This transformation is accomplished using Equation 3, with I_x as the input images. The output of this linearization process is represented as L_x, denoting the resulting linear images:

$$L_x = 255\, f^{-1}\left(\frac{I_x}{255}\right). \tag{3}$$

To recover the camera response function, we adopt the classical algorithm by Robertson et al. [64], which provides a robust and accurate estimation according to our experimental results. The linearization process takes us from the integer to the floating point domain. We perform the subsequent operations in this domain to prevent data loss due to quantization.

5) PATCH PROCESSING

Different from our earlier work that stored image differences directly [62] and therefore incurred significant storage costs, we divide each exposure into multiple patches, with each patch comprised of 64 × 64 pixels. These patches are then classified into four distinct categories as under-exposed,

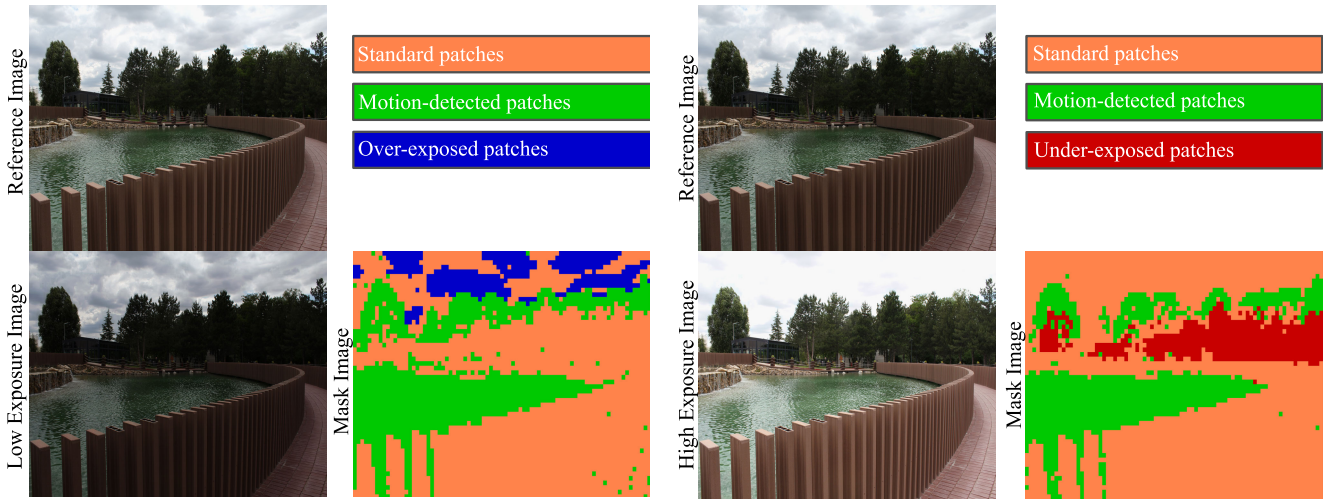

FIGURE 4. Patch-wise analysis of image exposures. Left-half: Reference exposure, low exposure, and the mask image indicating over-exposed (blue), and motion-detected (green) patches. A similar result is shown for the high exposure in the right-half of the figure with under-exposed patches of the reference shown in red. Orange patches are the standard patches that can be reconstructed from the reference. The reference image of a low and high exposure can be different, although it was shown as the same image in this example.

over-exposed, motion-detected, and standard patches. The main criteria in this categorization is the average luminance value of the reference patch and the difference of a patch from its reference. We compute this difference in the following manner:

$$L_{dif_x} = \left| L_{ref_x} - k_x L_x \right|, \quad \text{where } k_x = \frac{E_{ref_x}}{E_x}. \tag{4}$$

In this equation, E_x and E_{ref_x} stand for the exposure times of image x and its reference image, respectively. Additionally, L_x and L_{ref_x} represent the linearized versions of image x and its reference, and L_{dif_x} signifies the difference between them in the linearized domain.

By employing this approach, we are able to categorize and store the relevant patches for each exposure, avoiding redundant storage of patches which can be recovered from their reference. We use the following rules for patch classification:

- **Over-exposed patch:** An over-exposed patch occurs when its reference patch's average luminance surpasses a specified threshold (e.g., 225 in 8-bit data). Such patches are excessively bright, lacking the necessary details for accurate reconstruction during decompression.
- **Under-exposed patch:** An under-exposed patch is identified when the average luminance of its reference patch falls below a specified threshold (e.g., 25 in 8-bit data). These patches tend to be overly dark, devoid of the requisite details for accurate reconstruction during decompression.
- **Motion-detected patch:** If any significant deviation (e.g., 25 in 8-bit data) is detected between a patch and its corresponding reference after normalization (Equation 4), it is labeled as a motion-detected patch.
- **Standard patch**: These patches do not exhibit notable motion or exposure problems, allowing them to be reconstructed from their reference patches.

Of these categories, standard patches of a given exposure are not stored in the metadata section of the JPEG file, whereas all other categories are directly stored in their original JPEG compressed form (Figure 4).

B. DECOMPRESSION PIPELINE

The decompression pipeline as described in Algorithm 2 involves a series of steps designed to reconstruct the original exposures (Figure 5). The reconstruction process consists of individually reconstructing each exposure, starting with those

Algorithm 2 Decompression Process of Patch-Based CES

procedure CESDecompression(*JPEGFile*)
  $\{I_{rec_m}, Metadata\} \leftarrow JPEGFile$
  $\{C_{data}, f, Shift_{\langle 1..N \rangle}, E_{\langle 1..N \rangle}, N\} \leftarrow Read(Metadata)$
  $\{P_{\langle 1..N \rangle_{oe}}, P_{\langle 1..N \rangle_{ue}}, P_{\langle 1..N \rangle_{md}}\} \leftarrow Decompress(C_{data})$
  for each index x in $\langle 1 \ldots m-1 \rangle$ and $\langle m+1 \ldots N \rangle$ **do**
    $I_{ref_x} \leftarrow FindReference(I_x)$ ▷ Equation 2
    $L_{ref_x} \leftarrow Linearize(I_{ref_x})$ ▷ Equation 3
    $k_x = \frac{E_x}{E_{ref_x}}$ ▷ Equation 5
    $L_x = k_x L_{ref_x}$ ▷ Equation 5
    $I_x = ApplyCRF(L_x)$ ▷ Equation 6
    for each patch p in I_x **do**
      if p is not in $P_{x_{oe}}, P_{x_{ue}}, P_{x_{md}}$ **then**
        $P_{x_s} \leftarrow Add(p)$ ▷ Standard patch
    $I_x \leftarrow P_{x_s} + P_{x_{oe}} + P_{x_{ue}} + P_{x_{md}}$ ▷ Combine patches
    $I_{rec_x} \leftarrow ShiftImage(I_x, Shift_x)$ ▷ Reconst. image
  return $I_{rec_{\langle 1...N \rangle}}$

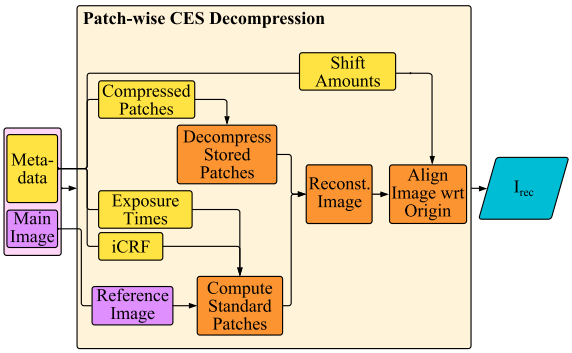

FIGURE 5. The decompression pipeline of the patch-based CES algorithm.

closest to the main reference exposure, which is typically the middle one. This is achieved by applying Equation 5 to the non-stored patches. Here, L_{ref_x} represents the linearized version of the reference image for exposure x. This is initially the main reference, but the process involves creating each reference one-by-one until all images are reconstructed. During this stage, it is important to note that certain patches are already stored within the metadata, which do not undergo the reconstruction process.

$$L_x = k_x L_{ref_x}, \text{ where } k_x = \frac{E_x}{E_{ref_x}}. \tag{5}$$

In this equation, E_x and E_{ref_x} denote the exposure times for image x and its corresponding reference image, respectively, and L_x is the estimated image x in linear domain.

It is important to note that the reconstructed pixels are initially in the linear domain. In order to go back to their original non-linear domain, we transform them using the recorded CRF:

$$I_x = 255\, f\left(\frac{L_x}{255}\right), \tag{6}$$

where f denotes the CRF.

In the final step, we use the pre-recorded shift amounts to disalign the exposures to make them closely resemble their original versions.

C. EFFECT OF CRF ON RECONSTRUCTION PROCESS

The importance of accurately recovering the CRF as close as possible to the actual response of the camera cannot be overstated. Any discrepancies or errors in the camera response function can lead to noticeable artifacts and inconsistencies in the reconstructed images [65]. For this purpose, we conducted a controlled evaluation of three CRF recovery algorithms that are commonly used for HDR imaging [4], [5], [6]. The process involved starting with an HDR image and creating 5 exposures from this image, similar to how multiple exposures are obtained from a real camera. However, we specifically enforced our "virtual camera" to have sRGB gamma [66]. We then fed the resulting exposures to each of the aforementioned CRF recovery algorithms and compared the similarity of the resulting curves with the ground-truth. As can be seen from Figure 6, Robertson et al.'s [6] curve better aligns with the true sRGB response.

In addition to this synthetic test, we evaluated the performance of these three algorithms on real exposure sequences. Figure 7 shows a sample result where we show the PSNR values of the reconstructed exposures using the original exposures as references. As can be seen from this figure, images reconstructed using Robertson et al.'s CRF have a higher PSNR value than the other two methods. Based on this evaluation, we opted to use the Robertson et al.'s method.

D. LIMITATIONS OF THE PATCH-BASED CES

Despite Robertson et al.'s approach was found to outperform the other CRF recovery algorithms, it does not completely prevent undesirable color distortions as shown in Figure 8. This stems from the inherent limitations of the CRF algorithms to perfectly estimate the camera response. Due to the imperfections in the manufacturing process, it is also possible that each pixel exhibits a slightly different response and neighborhood effects may further cause deviations in individual pixel values. To overcome this inherent limitation in the patch-based approach, we leverage a learning based algorithm as explained in the following section.

IV. RESIDUAL CES (ResCES)

ResCES uses a residual deep learning based neural network to reconstruct the original exposures with improved fidelity. While its operation is identical to patch-based CES for the compression stage, it uses a modified decompression workflow, as depicted in Figure 9 and detailed in Algorithm 3. Following the primary patch-based reconstruction, each individual patch undergoes a subsequent enhancement process through integration with the ResCES network model.

Algorithm 3 Decompression Process of ResCES

procedure ResCESDecompression(*JPEGFile*)
  $\{I_{rec_m}, Metadata\} \leftarrow JPEGFile$
  $\{C_{data}, f, Shift_{\langle 1..N\rangle}, E_{\langle 1..N\rangle}, N\} \leftarrow Read(Metadata)$
  $\{P_{\langle 1..N\rangle_{oe}}, P_{\langle 1..N\rangle_{ue}}, P_{\langle 1..N\rangle_{md}}\} \leftarrow Decompress(C_{data})$
  for each index x in $\langle 1 \ldots m-1\rangle$ and $\langle m+1 \ldots N\rangle$ **do**
    $I_{ref_x} \leftarrow FindReference(I_x)$ ▷ Equation 2
    $L_{ref_x} \leftarrow Linearize(I_{ref_x})$ ▷ Equation 3
    $k_x = \frac{E_x}{E_{ref_x}}$ ▷ Equation 5
    $L_x = k_x L_{ref_x}$ ▷ Equation 5
    $I_x = ApplyCRF(L_x)$ ▷ Equation 6
    for each patch p in I_x **do**
      if p is not in $P_{x_{oe}}, P_{x_{ue}}, P_{x_{md}}$ **then**
        $p' \leftarrow ApplyResCESModel(p)$
        $P_{x_s} \leftarrow Add(p')$ ▷ Standard patch
    $I_x \leftarrow P_{x_s} + P_{x_{oe}} + P_{x_{ue}} + P_{x_{md}}$ ▷ Combine patches
    $I_{rec_x} \leftarrow ShiftImage(I_x, Shift_x)$ ▷ Reconst. image
  return $I_{rec_{\langle 1..N\rangle}}$

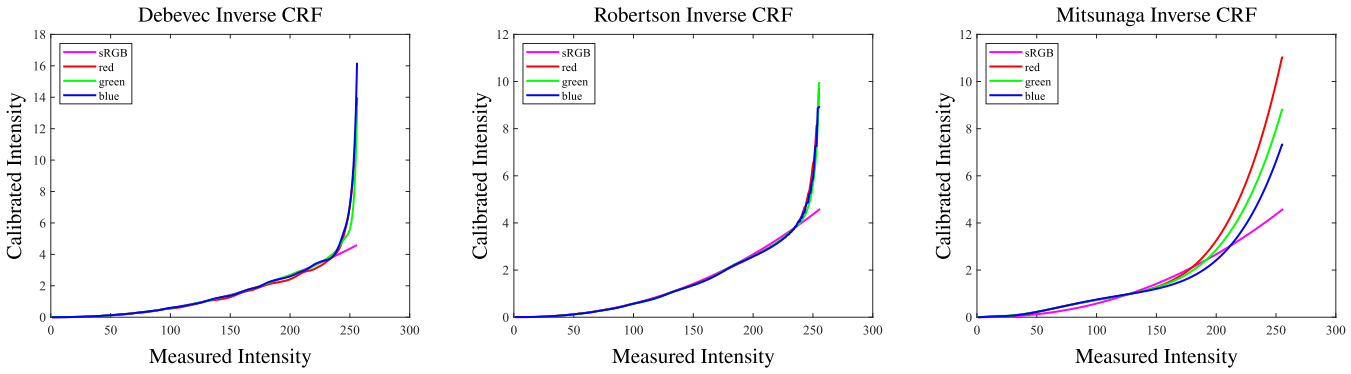

FIGURE 6. The results of Debevec and Malik's [4], Robertson et al.'s [6], and Mitsunaga and Nayar's [5] inverse CRF reconstructions. The response of each color channel is shown in red, green, and blue. The reference sRGB curve is shown in pink.

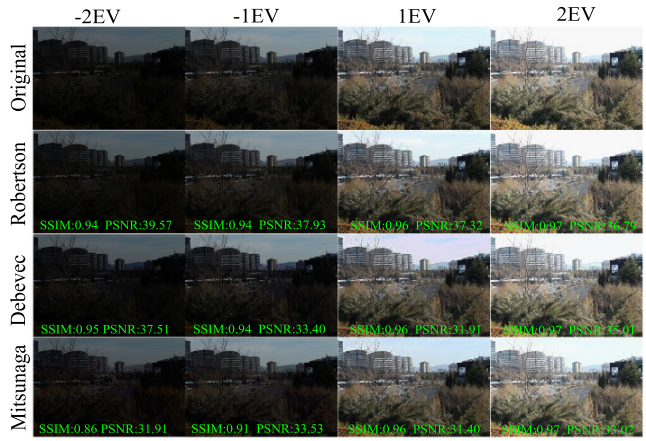

FIGURE 7. The results of patch-based CES reconstructed with the evaluated CRF recovery algorithms.

FIGURE 8. Block artifacts and color distortions may appear due to imperfections in the CRF recovery. Left: Patch-based CES reconstruction. Right: Original.

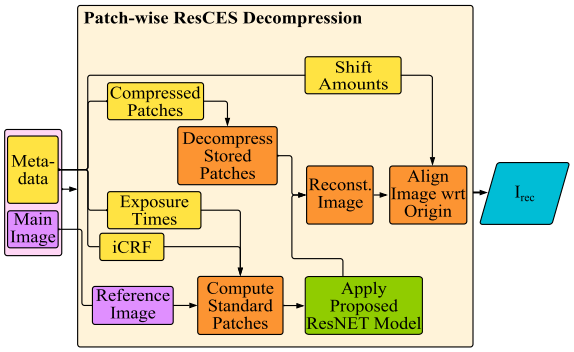

FIGURE 9. The decompression pipeline of the ResCES algorithm.

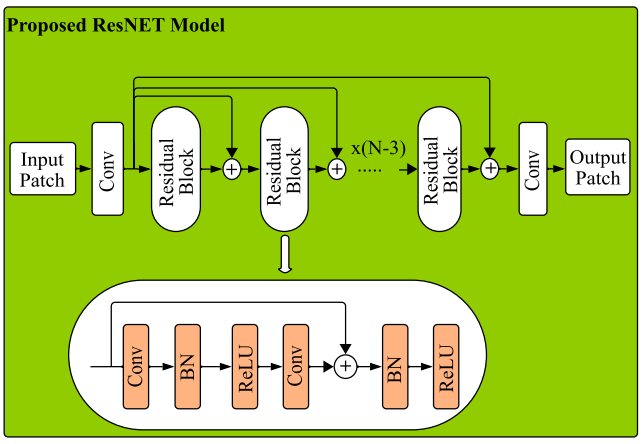

FIGURE 10. The proposed residual deep learning model.

A. NETWORK ARCHITECTURE

The diagram presented in Figure 10 illustrates the architecture of the proposed network model. This model follows a residual learning approach, as introduced by He et al. [67], and is primarily composed of residual blocks. These blocks facilitate the addition of the input from one convolutional layer to the output of the subsequent convolutional layer. By incorporating residual blocks, the network ensures the injection of information from the preceding layer to the subsequent layers, allowing for the training of deeper networks.

The proposed network model consists of three main components: the head, body, and tail. Notably, skip connections are employed, connecting the outputs from the head section to the outputs of each residual block. This technique, inspired from two well-known super-resolution

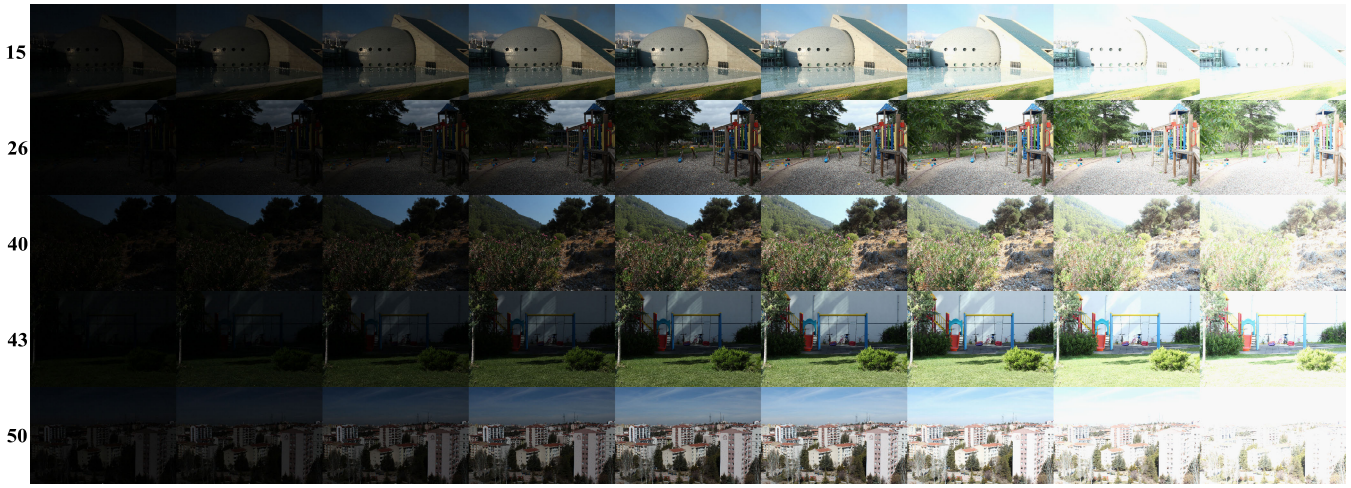

FIGURE 11. A subset of our dataset. Each image sequence is composed of nine exposures from the shortest exposure ($-4EV$) to the longest one ($+4EV$) with $1EV$ difference between each exposure.

reconstruction models, namely VDSR [68] and EDSR [69], enables the integration of feature information from the input layer to the output layer of each residual block. Consequently, the long-distance skip connections coerce the residual block modules to learn the disparity between the ground-truth and input images.

The head section initiates with a single convolutional layer. Considering the time-consuming nature of training, the body section is comprised of N stacked residual blocks (we take N as 12 in our case), arranged in the following sequence: [Conv-BN-ReLU-Conv-BN-ReLU]. Subsequently, the tail section encompasses another convolution layer. In total, the network comprises of $2N + 2$ convolution layers.

The original VDSR model and some other super-resolution reconstruction models, such as EDSR, do not use batch normalization (BN) layers. Since the BN layer normalizes the features, it diminishes the network's range flexibility. In the image super-resolution reconstruction task, the output image needs to be consistent with the input in color, contrast, and brightness. When the image passes through the BN layer, its color distribution is normalized, which amounts to contrast stretching. This affects the original contrast information of the image, so the BN layers reduce the quality of the output image in the image super-resolution reconstruction task. However, image enhancement, as we do in the current study, is different from the super-resolution reconstruction task. Our patch-based reconstructed images have color, brightness, and contrast deviations that need to be corrected. Therefore, the addition of the BN layers in our case improves convergence and improves the overall quality of the reconstructed patches.

B. DATASET

While there are pre-existing multi-exposure image stacks for HDR imaging, they are generally tailored for specific research goals such as deghosting [8], [14], [70], [71], [72] and have a limited number of bracketed exposures. In recognition of this limitation, we decided to create a new multi-exposure image dataset that encapsulates a diverse array of scenes and lighting conditions. Our dataset contains images of natural, urban, and indoor environments with both natural and artificial lighting. We tried to meticulously balance the inclusion of both static and dynamic scenes with varying degrees of motion, simultaneously paying attention to the visual appeal of the captured environments. This resulted in a total of 50 scenes, 5 of which are depicted in Figure 11. For capturing the bracketed sequences, we used a Canon EOS 600d dSLR camera with Magic Lantern firmware.[1] The camera was configured to capture images in the sRGB color space at a resolution of 5184×3456 pixels (18 MPs). Each scene was captured using 9 exposures with $1EV$ increments from $-4EV$ to $+4EV$.

C. NETWORK TRAINING

1) DATA PARTITIONING

For training our proposed network model, we adopted a 10-fold cross-validation strategy. We divided the 50 images into 10 subsets, each consisting of 5 scenes. Then, in an iterative manner, 8 subsets were used for training the model, while the remaining two were used for validation and testing. By utilizing this k-fold cross-validation approach, we aimed to ensure that our model is trainable and robust under a diverse set of training and testing scenes.

2) DATA AUGMENTATION

Our training data consists of pairs of RGB input patches of size 64×64 and their corresponding ground-truth patches. The RGB input patches are obtained from patch-based reconstructed exposures, which are generated using the patch-based CES algorithm (Section III). The ground-truth

[1] https://magiclantern.fm/

patches are the corresponding patches in the original exposures.

To simulate the inherent noise present in real-world images and improve the generalization capabilities of the network, we intentionally corrupted the RGB input patches by adding Gaussian noise, which is a common choice due to its statistical properties that mimic natural image noise. Specifically, we add Gaussian noise with a zero mean and random standard deviation uniformly sampled from the interval $[0, 0.02]$ assuming that the input pixel value range is $[0, 1]$. This range yields a controlled level of noise that is perceptually realistic while also allowing the network to learn patterns and features that are robust to noise variations. In addition to noise, we augment our dataset by randomly rotating the input and ground-truth patches by 0°, 90°, 180°, and 270°. This yields a total of approximately 6.3 million patches for each cross-validation iteration.

3) LOSS FUNCTION

In this study, we used the L_1 loss function defined by the following formula:

$$L(\Theta) = \frac{1}{N} \sum_{p=1}^{N} \left\| I'_p - I_p \right\|. \tag{7}$$

In this equation, I'_p represents the reconstructed patch obtained through the ResCES algorithm, while I_p corresponds to the ground-truth patch. The variable Θ represents the network parameters that need to be learned. Thus, the objective of our training process is to minimize the average difference between the reconstructed patches and the corresponding ground-truth patches.

4) IMPLEMENTATION DETAILS

For the implementation of our proposed networks, we used Keras [73] as the deep learning framework, with TensorFlow as the underlying backend. The training process was conducted on a single NVIDIA Tesla P100 GPU. As hyper-parameter optimization plays a crucial role in fine-tuning the performance of deep learning models, we used Keras Tuner [74] to automate this process. These experiments explored various factors such as the effects of different training epochs, batch sizes, number of layers, and learning rates. We aimed to find the combination of these factors that resulted in the best performance. Our experiments with this tuner revealed the following hyper-parameter settings as the ones with the smallest validation loss:

- Initial learning rate: 10^{-4} (explored the $[10^{-5}, 10^{-3}]$ range)
- Batch size: 64 (explored {32, 64, 128})
- Convolutional layer count: 26 (explored 20 to 30 layers)

It is important to note that the tuning process incorporated all variable parameters collectively, rather than individually. This implies that the tuner simultaneously explored various combinations of learning rates, batch sizes, and layers.

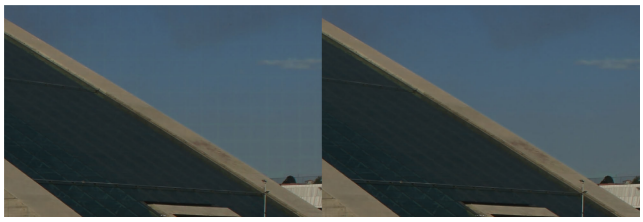

FIGURE 12. In the case of a 64 × 64 input patch, only the central 48 × 48 region is deemed a valid output, as shown in the right image. This overlap among the patches alleviates potential tiling artifacts, which could occur had the patches been disjoint, as illustrated in the left image.

As for the optimizer we used ADAM [75] for faster convergence. ADAM blends Momentum and RMSProp optimizers, adapting learning rates for each parameter, enhancing stability, and accommodating various gradient scales. Key hyper-parameters include the learning rate, β_1, β_2, and ε. We set them to the following values that were experimentally found to be good default settings [75]: $\beta_1 = 0.9$, $\beta_2 = 0.99$, and $\varepsilon = 10^{-7}$.

The initial learning rate of 10^{-4} was automatically halved after 20 epochs if the validation loss stagnated. If there was no improvement after 40 epochs, the training was early-terminated instead of waiting for the previously fixed count of 100 epochs. With this chosen setup and training configuration, the total training time for the proposed network took approximately 30 hours per fold. This time includes the processing of all the training samples and the adjustment of the network parameters to minimize the objective function.

During inference, we applied the ResCES algorithm to 64×64 patches. However, to avoid patch artifacts, we used the center 48×48 region in each patch. This allows neighboring patches to share a common border and produce coherent results, eliminating tiling artifacts (Figure 12).

V. RESULTS

In this section, we provide a comprehensive analysis of our experimental results, aiming to assess the effectiveness of our methodology using a diverse test dataset including both dynamic and static scenes. A sample selection from the test scenes for which we provide per-image results is shown in Figure 11. These scenes correspond to the test scenes of fold-1 in our 10-fold cross-validation.

In evaluating the reconstructed image quality, we utilized two widely used objective image quality metrics, namely the structural similarity index (SSIM) [76] and the peak signal-to-noise ratio (PSNR) [77], enabling us to make both perceptual and numerical assessments of the similarity between the recovered and original images. We also conducted a user study to investigate whether our reconstructions are visually distinguishable from the originals.

A. HYPER-PARAMETERS AND CONVERGENCE

As explained in Section IV-C, our first aim was to discover the network hyper-parameters that yield the best reconstruction

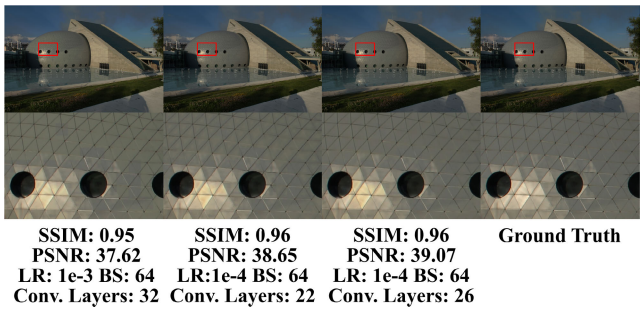

FIGURE 13. By comparing model outputs with different hyper-parameters we aimed to find the optimal configuration for the network. Three examples are shown above, together with the ground-truth image. Batch size (BS) was 64 in all cases. The insets show a detailed view of the marked region in the main image.

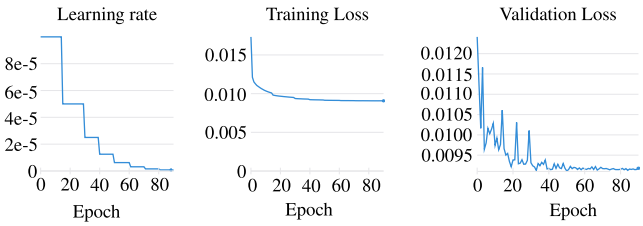

FIGURE 14. Convergence performance of our network: Learning rate, training, and validation loss as a function of epoch.

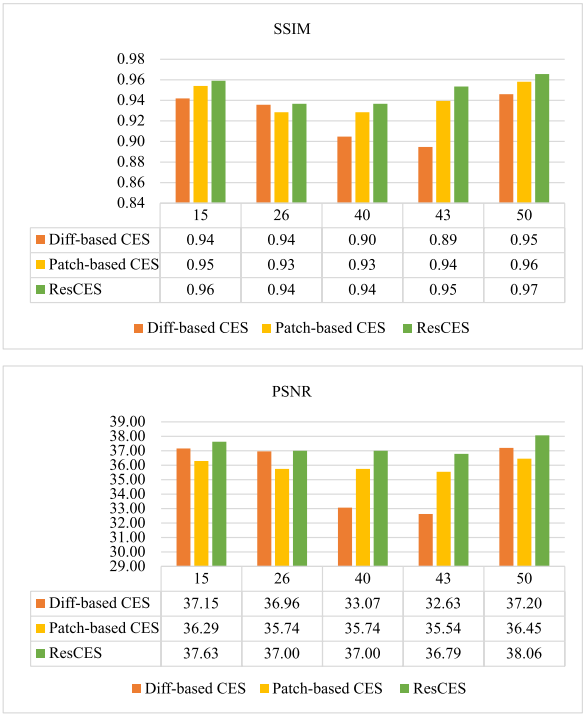

FIGURE 15. The difference-based CES, patch-based CES, and the ResCES outcomes are evaluated across the fold-1 test set (15, 26, 40, 43, and 50) using the SSIM and PSNR metrics. The proposed ResCES algorithm shows noticeable improvement in all cases.

results. The search space comprised of the learning rate, batch size, and the number of convolutional layers. Three examples that are obtained during the process are depicted in Figure 13. As can be seen from this figure, the final network with an initial learning rate of 10^{-4}, a batch size of 64, and a convolutional layer count of 26 provided better results compared to the other alternatives that were explored.

An important criterion that determines the quality of the trained network is its convergence behaviour. We show this result in Figure 14, where from left-to-right the plots show the learning rate, training loss, and validation loss as a function of epoch. These plots, which are obtained from fold-1 of the training process, show that a plausible convergence behavior is achieved.

B. IMAGE QUALITY VS. COMPRESSION RATIO

The image quality metrics for the five fold-1 scenes are shown in Figure 15. Here, we compare our ResCES results with two versions of the CES algorithm, namely the difference-based CES [62] and patch-based CES (Section III). It can be seen that the ResCES algorithm consistently outperforms the other algorithms with respect to both PSNR and SSIM metrics.

Furthermore, upon visual comparison of the results, it was observed that the reconstructed exposures obtained through the ResCES technique display significant improvement over the patch-based CES method (Figure 16). As mentioned earlier, the primary limitation of the patch-based approach is the occurrence of color shifts and block artifacts along the borders where the originally stored patches and the reconstructed patches from the reference exposure are

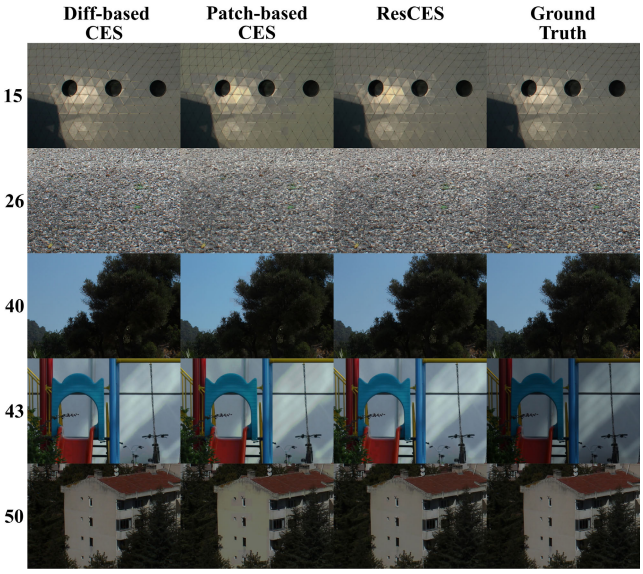

FIGURE 16. ResCES decreases the color shifts and block artifacts that appear in the patch-based CES algorithm. In these examples, the difference-based CES also produces good results, albeit at a much higher storage cost as can be seen in Table 1. The insets are taken from the corresponding test images.

merged. However, the ResCES approach effectively mitigates these problems, leading to visually improved reconstructed images. Although the difference-based CES results also look plausible in this figure, its total storage cost is significantly higher than the ResCES method as discussed below.

TABLE 1. Compression ratios for five test scenes. The original size is the total storage size required for the 9 bracketed exposures. The compressed size is the size of the single JPEG file created by using our algorithm.

Image Name (Test Set)	Original Size (MB)	Diff-based CES Compressed Size (MB)	Diff-based CES Compression Ratio (X)	Patch-based CES/ResCES Compressed Size (MB)	Patch-based CES/ResCES Compression Ratio (X)
15	44.4	34.6	1.28	9.18	4.84
26	63.1	40.2	1.57	12	5.26
40	71.2	49.1	1.45	20.3	3.51
43	55.9	43.1	1.30	12.1	4.62
50	45.4	36.2	1.25	9.6	4.73

TABLE 2. Stored and non-stored patch percentages for each exposure of the test image set. Note that the majority of the patches are not stored yielding a significant reduction in the overall file size.

Image Name (Test Set)		Stored Patches Over-exposed Patches	Stored Patches Under-exposed Patches	Stored Patches Motion-detected Patches	Non-stored Patches Standard Patches	Non-stored Patches %
15	**Exp. 1**	57	0	213	7506	96.53
	Exp. 2	84	0	103	7589	97.60
	Exp. 3	192	0	575	7009	90.14
	Exp. 4	246	0	520	7010	90.15
	Exp. 6	0	285	73	7418	95.40
	Exp. 7	0	16	25	7735	99.47
	Exp. 8	0	0	70	7706	99.10
	Exp. 9	0	0	359	7417	95.38
26	**Exp. 1**	0	0	21	7755	99.73
	Exp. 2	100	0	61	7615	97.93
	Exp. 3	429	0	558	6789	87.31
	Exp. 4	523	0	328	6925	89.06
	Exp. 6	0	2160	161	5455	70.15
	Exp. 7	0	883	249	6644	85.44
	Exp. 8	0	164	421	7,191	92.48
	Exp. 9	0	109	742	6925	89.06
40	**Exp. 1**	0	0	1275	6501	83.60
	Exp. 2	256	0	1493	6027	77.51
	Exp. 3	581	0	2982	4213	54.18
	Exp. 4	812	0	2485	4479	57.60
	Exp. 6	0	744	2412	4620	59.41
	Exp. 7	0	376	2339	5061	65.08
	Exp. 8	0	255	2594	4,927	63.36
	Exp. 9	0	68	2782	4926	63.35
43	**Exp. 1**	0	0	1537	6239	80.23
	Exp. 2	59	0	1135	6582	84.65
	Exp. 3	184	0	2491	5101	65.60
	Exp. 4	287	0	1640	5849	75.22
	Exp. 6	0	1606	329	5841	75.12
	Exp. 7	0	764	158	6854	88.14
	Exp. 8	0	296	118	7,362	94.68
	Exp. 9	0	14	237	7525	96.77
50	**Exp. 1**	4	0	24	7748	99.64
	Exp. 2	45	0	49	7682	98.79
	Exp. 3	542	0	330	6904	88.79
	Exp. 4	850	0	651	6275	80.70
	Exp. 6	0	169	39	7568	97.33
	Exp. 7	0	23	25	7728	99.38
	Exp. 8	0	13	342	7,421	95.43
	Exp. 9	0	0	648	7128	91.67

Table 1 reports the original and compressed sizes of the exposure sequences used for testing purposes. The original size corresponds to the total size of 9 exposures in each sequence, whereas the compressed size is the size of the single JPEG file that contains the reference image and the necessary reconstructive information. It is worth noting that the ResCES method achieved compression ratios ranging from 3.51 to 5.26. This ratio largely depends on how many patches from each exposure are stored due to being a dynamic, under-, or over-exposed patch. We provide this analysis in Table 2 in which we show the percentage of patch categories for each exposure of the test sequences. It can be seen from this table that the large majority of the patches are not stored; that is they are reconstructed from the reference image and the patches stored in its metadata.

Finally, in Table 3 we share our results for all folds of the 10-fold cross validation evaluation. As can be seen from this table, the proposed ResCES algorithm consistently outperforms the other approaches, yields high SSIM and PSNR scores, and on average achieves a storage reduction ratio of approximately 4.5 times.

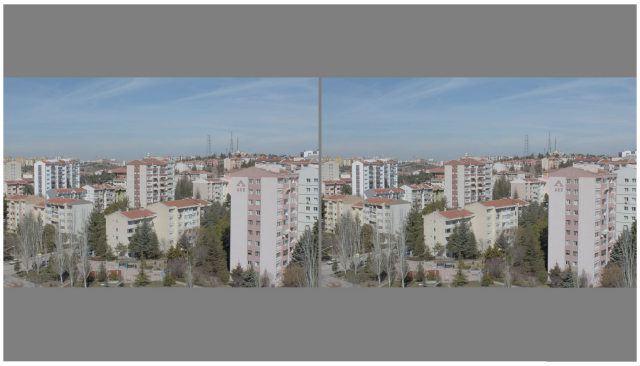

FIGURE 17. A sample test stimulus from the subjective experiment. In this case, the left image is obtained from our reconstructed exposures and the right image from the original exposures. This information was withheld from the participants.

C. SUBJECTIVE EVALUATION

To validate whether our algorithm produces visually indistinguishable results compared to the original exposures, we conducted a subjective evaluation. To this end, we selected the images from the fold-1 of our experiment and created tone-mapped HDR images [4], [78]. This process was repeated using both the original exposures and our ResCES-based reconstructed exposures after they were stored in a compressed form in the single JPEG file. The resulting images are placed side-by-side on a neutral gray background with full-HD resolution (see Figure 17). The positions of the original and reconstructed images were randomized. The participants' task was to indicate whether the left or the right image was better. We also allowed for the visually indistinguishable option to avoid forcing users to make decision when they could not notice a difference. In total, 40 voluntary participants with informed consent between the ages of 20-50 attended the experiment. The results are summarized in Table 4.

According to these results, 34% of the participants found our reconstructions better, 37.5% found the originals better, and 28.5% found them to be indistinguishable. To understand whether these differences are statistically significant, we conducted a two-tailed z-test [79] by equally distributing the indistinguishable votes between the other two groups. This resulted in a z-value of 0.22, which is significantly lower than the critical z-value of 1.96 at a significance level of $\alpha = 0.05$. This indicates that the preference between the original and reconstructed images is not statistically significant.

D. RUN-TIME ANALYSIS

A detailed run-time analysis of our compression and decompression pipelines is shared in Table 5. To obtain these results, we used a system with an Intel i7-11850H processor running at 2.50 GHz, 64 GBs of system memory,

TABLE 3. Our overall cross-validation results that include all folds. Each row shows the mean result of the 5 exposure sequences within each fold. The last row shows the overall average across all folds.

Fold	Diff-based CES		Patch-based CES		ResCES			Diff-based CES		Patch-based CES/ResCES	
	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	Ori. File Size (MB)	Comp. File Size (MB)	Comp. Ratio (X)	Comp. File Size (MB)	Comp. Ratio (X)
1	0.92	35.41	0.94	35.95	0.95	37.30	56.00	40.64	1.37	12.62	4.60
2	0.93	36.75	0.95	35.79	0.95	37.19	45.38	36.12	1.27	10.23	4.47
3	0.93	36.43	0.94	35.90	0.95	36.51	43.36	32.30	1.34	9.84	4.48
4	0.94	36.79	0.94	35.51	0.95	36.90	45.90	35.08	1.31	10.16	4.56
5	0.93	36.17	0.94	36.09	0.95	36.61	47.18	33.54	1.37	10.93	4.35
6	0.92	34.86	0.95	36.34	0.96	36.67	48.22	38.28	1.29	12.80	3.96
7	0.95	38.45	0.94	36.07	0.95	37.37	35.88	26.46	1.35	7.65	4.68
8	0.93	35.38	0.94	35.11	0.95	36.16	53.08	37.42	1.41	13.10	4.06
9	0.92	35.20	0.94	36.45	0.95	36.58	48.50	35.68	1.39	13.31	4.02
10	0.94	38.80	0.94	36.50	0.95	36.56	39.20	28.34	1.38	8.85	4.46
Avg	**0.93**	**36.42**	**0.94**	**35.97**	**0.95**	**36.78**	**46.27**	**34.39**	**1.35**	**10.95**	**4.36**

TABLE 4. Summary of the subjective evaluation, which was conducted for the first fold of our image set. The figures indicate the number of times the participants preferred the original image and our ResCES-based reconstruction or found them to be indistinguishable.

Image Name	Original	ResCES	Indistinguishable
15	15	17	8
26	12	20	8
40	17	8	15
43	18	14	8
50	13	9	18
Total	**75**	**68**	**57**

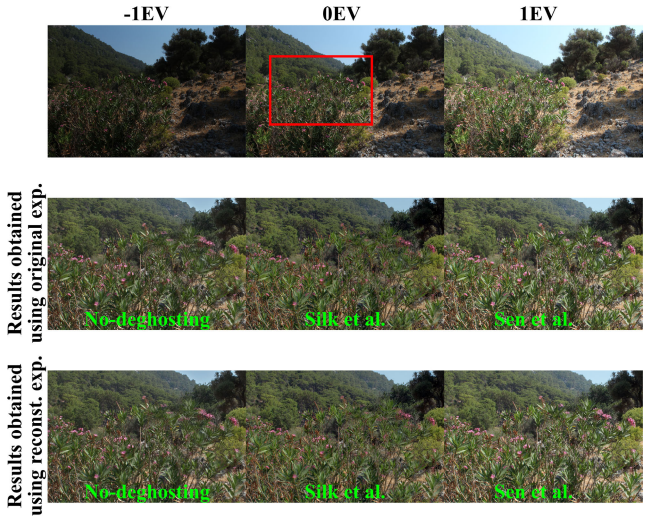

FIGURE 18. Visualizing the effects of deghosting algorithms on both the original and ResCES reconstructed exposures. The original exposures are shown at the top. The middle row shows the algorithm results obtained from the original exposures and the bottom row shows the same for the reconstructed ones using our algorithm. Note the high degree of similarity between the two. The figure also shows that the ability to reconstruct the original exposures from our compact form allows the application of different and possibly better deghosting algorithms as they emerge.

and an NVIDIA Geforce RTX 3070 GPU. Among the steps in the compression pipeline, the computation of the camera response function takes the longest time. In practice, the CRF for a camera can be recovered once and reused for subsequent image sequences unless critical camera parameters such as the color space or post-processing effects are changed. As for the decompression pipeline, the application of our residual network model takes the longest time. We note that these timing results are obtained for a bracketed exposure sequence of 9 images with each exposure captured at 5184×3456 resolution (18 MPs).

VI. USE-CASE: FORWARD COMPATIBLE DEGHOSTING

To illustrate one of the motivating features of our algorithm, we incorporate it into the HDR deghosting workflow, which deals with one of the main challenges of HDR photography: handling moving objects and/or camera viewpoint changes between exposures. Here, we showcase the usage of two distinct HDR deghosting algorithms, both of which leverage the ResCES reconstruction. The first algorithm follows the approach of Silk and Lang [80] by employing optimal exposures during the deghosting process. The second algorithm adopts a patch-based deghosting method by Sen et al. [81].

In particular, Silk and Lang's algorithm employs a technique termed pairwise down-weighting (PWD), which is effective when motion affects only a small portion of the input image stack. However, situations involving dynamic elements such as foliage, flags, and fluids can lead to certain super-pixels displaying motion across all input images. This specific motion, referred to as fluid motion (FM), proves challenging for the PWD approach's accuracy. In such cases, the algorithm generates an alternative output by relying exclusively on the best exposure, optimizing cumulative pixel weights within the motion-affected region. On the other hand, the approach presented by Sen et al. incorporates a patch-based deghosting method that employs image patches and matching strategies to mitigate ghosting artifacts. The patch-based algorithm has demonstrated robustness in accommodating scene content variations and diverse manifestations of ghosting artifacts.

Figure 18 illustrates the results achieved by employing deghosting algorithms in this context. These algorithms have been employed for both the original and reconstructed exposures generated through ResCES. The HDR images are then reconstructed using Equation 1 and the results are tone-mapped [78]. It can be noted that the results obtained from the

TABLE 5. Run-time analysis of ResCES. The reported results are obtained for a sequence of 9 exposures 18 megapixels each.

Image Name (Test Set)	ResCES Compression Time (sec)						ResCES Decompression Time (sec)				
	Sort Images & Find References	Align Image wrt. Ref. Image	Compute CRF	Apply Patch	Compress Metadata	Total Comp. Time	Decompress Metadata	Apply Proposed Network Model	Reconstruct Image	Align Image wrt. Original	Total Decomp. Time
15	1.41	4.67	12.05	5.26	1.40	24.80	1.13	90.76	2.98	1.59	96.46
26	1.57	4.63	12.22	5.35	1.52	25.29	1.25	90.51	2.91	1.79	96.45
40	1.62	4.71	12.36	5.42	2.03	26.13	1.35	90.48	2.89	1.78	96.49
43	1.51	4.66	12.21	5.34	1.62	25.35	1.21	90.73	2.91	1.64	96.50
50	1.44	4.62	12.19	5.29	1.42	24.95	1.16	90.87	3.00	1.62	96.65

original exposures and the reconstructed ones resemble each other. Upon scrutinizing the results, it becomes evident that the patch-based deghosting method surpasses Silk and Lang's algorithm in performance, highlighting that the dynamic nature of this process introduces the possibility of better algorithms emerging over time. Consequently, the forward compatibility of the proposed storage system may prove to be worthwhile in such evolving scenarios.

VII. CONCLUSION AND FUTURE WORK

In this paper, we proposed a novel multi-exposure file format based on JPEG to address the challenges associated with storing and managing exposure sequences commonly used in HDR photography. By employing a selective patch-based storage scheme and utilizing a deep residual network for minimizing reconstruction artifacts, the proposed method offers a significant reduction in storage requirements without compromising image quality. We believe that by allowing an entire bracketed sequence to be stored within a single file in a compact manner, our method simplifies the management of bracketed exposure sequences. Furthermore, it allows a compactly stored exposure sequence to benefit from emerging algorithms in the field, making it forward compatible.

Future work in this area can focus on refining the deep residual network, exploring alternative storage schemes, and incorporating additional optimizations to the overall workflow to further enhance the efficiency and usability of multi-exposure sequences for HDR photography.

ACKNOWLEDGMENT

The authors gratefully acknowledge the invaluable support of TUBITAK ULAKBIM, High Performance and Grid Computing Center (TRUBA resources), where the numerical calculations presented in this article were conducted.

REFERENCES

[1] E. Reinhard, G. Ward, S. Pattanaik, and P. Debevec, *High Dynamic Range Imaging: Acquisition, Display and Image-Based Lighting*, 2nd ed. San Francisco, CA, USA: Morgan Kaufmann, 2010.

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

[3] CNET. *What is HDR for TVs, and Why Should You Care?* Accessed: Oct. 4, 2023. [Online]. Available: https://www.cnet.com/news/what-is-hdr-for-tvs-and-why-should-you-care/

[4] P. E. Debevec and J. Malik, "Recovering high dynamic range maps from photographs," in *Proc. 24th Annu. Conf. Comput. Graph. Interact. Techn.*, New York, NY, USA. Reading, MA, USA: Addison-Wesley, 1997, pp. 369–378.

[5] T. Mitsunaga and S. K. Nayar, "Radiometric self calibration," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 1999, p. 380.

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

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

[8] N. K. Kalantari and R. Ramamoorthi, "Deep high dynamic range imaging of dynamic scenes," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 1–12, Aug. 2017.

[9] E. Reinhard, T. Kunkel, Y. Marion, J. Brouillat, R. Cozot, and K. Bouatouch, "Image display algorithms for high- and low-dynamic-range display devices," *J. Soc. Inf. Display*, vol. 15, no. 12, pp. 997–1014, Dec. 2007.

[10] K. Panetta, L. Kezebou, V. Oludare, S. Agaian, and Z. Xia, "TMO-net: A parameter-free tone mapping operator using generative adversarial network, and performance benchmarking on large scale HDR dataset," *IEEE Access*, vol. 9, pp. 39500–39517, 2021.

[11] I. R. Khan, W. Aziz, and Seong-O. Shim, "Tone-mapping using perceptual-quantizer and image histogram," *IEEE Access*, vol. 8, pp. 31350–31358, 2020.

[12] A. Rana, P. Singh, G. Valenzise, F. Dufaux, N. Komodakis, and A. Smolic, "Deep tone mapping operator for high dynamic range images," *IEEE Trans. Image Process.*, vol. 29, pp. 1285–1298, 2020.

[13] X. Huang, Q. Zhang, Y. Feng, H. Li, X. Wang, and Q. Wang, "HDR-NeRF: High dynamic range neural radiance fields," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 18377–18387.

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

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

[16] Q. Yan, D. Gong, J. Q. Shi, A. van den Hengel, C. Shen, I. Reid, and Y. Zhang, "Dual-attention-guided network for ghost-free high dynamic range imaging," *Int. J. Comput. Vis.*, vol. 130, no. 1, pp. 76–94, Jan. 2022.

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

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

[19] S. De Neve, B. Goossens, H. Luong, and W. Philips, "An improved HDR image synthesis algorithm," in *Proc. 16th IEEE Int. Conf. Image Process. (ICIP)*, Nov. 2009, pp. 1545–1548.

[20] H. Xu, J. Ma, J. Jiang, X. Guo, and H. Ling, "U2Fusion: A unified unsupervised image fusion network," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 1, pp. 502–518, Jan. 2022.

[21] H. Xu, J. Ma, and X.-P. Zhang, "MEF-GAN: Multi-exposure image fusion via generative adversarial networks," *IEEE Trans. Image Process.*, vol. 29, pp. 7203–7216, 2020.

[22] C. Yaqing and W. Huaming, "Multi-exposure fusion with guidance information: Night color image enhancement for roadside units," *IEEE Access*, vol. 11, pp. 64494–64506, 2023.

[23] S. Choi, O.-J. Kwon, and J. Lee, "A method for fast multi-exposure image fusion," *IEEE Access*, vol. 5, pp. 7371–7380, 2017.

[24] Z. Li, C. Zheng, J. Zheng, and S. Wu, "Neural augmented exposure interpolation for HDR imaging," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2023, pp. 171–175.

[25] M. D. Grossberg and S. K. Nayar, "Modeling the space of camera response functions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 10, pp. 1272–1282, Oct. 2004.
[26] A. Chalmers, P. Campisi, P. Shirley, and I. G. Olaizola, *High Dynamic Range Video: Concepts, Technologies, and Applications*. London, U.K.: Academic Press, 2016.
[27] I. Takayanagi and R. Kuroda, "HDR CMOS image sensors for automotive applications," *IEEE Trans. Electron Devices*, vol. 69, no. 6, pp. 2815–2823, Jun. 2022.
[28] R. Mukherjee, M. Bessa, P. Melo-Pinto, and A. Chalmers, "Object detection under challenging lighting conditions using high dynamic range imagery," *IEEE Access*, vol. 9, pp. 77771–77783, 2021.
[29] F. Banterle, P. Ledda, K. Debattista, and A. Chalmers, "Inverse tone mapping," in *Proc. 4th Int. Conf. Comput. Graph. Interact. Techn. Australasia Southeast Asia*, 2006, pp. 349–356.
[30] Y. Kinoshita and H. Kiya, "ITM-net: Deep inverse tone mapping using novel loss function considering tone mapping operator," *IEEE Access*, vol. 7, pp. 73555–73563, 2019.
[31] L. Wang, L.-Y. Wei, K. Zhou, B. Guo, and H.-Y. Shum, "High dynamic range image hallucination," *Rendering Techn.*, vol. 321, no. 326, p. 3, 2007.
[32] Y.-L. Liu, W.-S. Lai, Y.-S. Chen, Y.-L. Kao, M.-H. Yang, Y.-Y. Chuang, and J.-B. Huang, "Single-image HDR reconstruction by learning to reverse the camera pipeline," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 1651–1660.
[33] D. Marnerides, T. Bashford-Rogers, and K. Debattista, "Deep HDR hallucination for inverse tone mapping," *Sensors*, vol. 21, no. 12, p. 4032, Jun. 2021.
[34] N. Messikommer, S. Georgoulis, D. Gehrig, S. Tulyakov, J. Erbach, A. Bochicchio, Y. Li, and D. Scaramuzza, "Multi-bracket high dynamic range imaging with event cameras," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2022, pp. 547–557.
[35] Z. A. El Shair, A. Hassani, and S. A. Rawashdeh, "CSTR: A compact spatio-temporal representation for event-based vision," *IEEE Access*, vol. 11, pp. 102899–102916, 2023.
[36] J. Han, C. Zhou, P. Duan, Y. Tang, C. Xu, C. Xu, T. Huang, and B. Shi, "Neuromorphic camera guided high dynamic range imaging," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 1730–1739.
[37] Y. Yang, J. Han, J. Liang, I. Sato, and B. Shi, "Learning event guided high dynamic range video reconstruction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2023, pp. 13924–13934.
[38] S. Dabral, "Image data processing for digital overlap wide dynamic range sensors," U.S. Patent 17 168 224, May 27, 2021.
[39] L. Wang and K.-J. Yoon, "Deep learning for HDR imaging: State-of-the-art and future trends," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 12, pp. 8874–8895, Dec. 2022.
[40] K. Lee, J. Park, Y. I. Jang, and N. I. Cho, "Frequency-domain multi-exposure HDR imaging network with representative image features," *IEEE Access*, vol. 11, pp. 124899–124910, 2023.
[41] Q. Yan, W. Chen, S. Zhang, Y. Zhu, J. Sun, and Y. Zhang, "A unified HDR imaging method with pixel and patch level," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2023, pp. 22211–22220.
[42] E. Sikudová, T. Pouli, A. Artusi, A. O. Akyüz, F. Banterle, Z. M. Mazlumoglu, and E. Reinhard, "A gamut-mapping framework for color-accurate reproduction of HDR images," *IEEE Comput. Graph. Appl.*, vol. 36, no. 4, pp. 78–90, Jul. 2016.
[43] D. S. Taubman and M. W. Marcellin, *JPEG2000: Image Compression Fundamentals, Standards and Practice*. Boston, MA, USA: Kluwer Academic, 2002.
[44] F. Dufaux, G. J. Sullivan, and T. Ebrahimi, "The JPEG XR image coding standard [standards in a nutshell]," *IEEE Signal Process. Mag.*, vol. 26, no. 6, pp. 195–204, Nov. 2009.
[45] W. B. Pennebaker and J. L. Mitchell, *JPEG: Still Image Data Compression Standard*. Berlin, Germany: Springer, 1992.
[46] G. K. Wallace, "The JPEG still picture compression standard," *IEEE Trans. Consum. Electron.*, vol. 38, no. 1, pp. 18–34, Feb. 1992.
[47] J. Tesic, "Metadata practices for consumer photos," *IEEE Multimedia-Mag.*, vol. 12, no. 3, pp. 86–92, Jul. 2005.
[48] S. Çiftçi, A. O. Akyuz, and T. Ebrahimi, "A reliable and reversible image privacy protection based on false colors," *IEEE Trans. Multimedia*, vol. 20, no. 1, pp. 68–81, Jan. 2018.
[49] W. Greg, "JPEG-HDR: A backwards-compatible, high dynamic range extension to JPEG," in *Proc. 13th Color Imag. Conf.*, 2005, p. 3.
[50] R. Mantiuk, A. Efremov, K. Myszkowski, and H.-P. Seidel, "Backward compatible high dynamic range MPEG video compression," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 713–723, Jul. 2006.
[51] G. J. Ward, "The RADIANCE lighting simulation and rendering system," in *Proc. 21st Annu. Conf. Comput. Graph. Interact. Techn.*, New York, NY, USA, 1994, pp. 459–472.
[52] G. W. Larson, "Overcoming gamut and dynamic range limitations in digital images," in *Proc. Color Imag. Conf.*, vol. 1998, pp. 214–219.
[53] F. Kains, R. Bogart, D. Hess, P. Schneider, and B. Anderson. *OpenEXR*. Accessed: Nov. 26, 2017. [Online]. Available: https://www.openexr.org/
[54] A. Artusi, R. K. Mantiuk, T. Richter, P. Hanhart, P. Korshunov, M. Agostinelli, A. Ten, and T. Ebrahimi, "Overview and evaluation of the JPEG XT HDR image compression standard," *J. Real-Time Image Process.*, vol. 16, no. 2, pp. 413–428, Apr. 2019.
[55] *Dolby Vision*. Accessed: Sep. 4, 2023. [Online]. Available: https://www.dolby.com/technologies/dolby-vision/#gref
[56] P. Topiwala, W. Dai, and M. Krishnan, "Improvements on HDR10," in *Proc. Digital Media Industry Academic Forum (DMIAF)*, 2016, pp. 17–22.
[57] D. Vo, C. Liu, M. Nelson, B. Mandel, and S. Hyun, "HDR10+ adaptive ambient compensation using creative intent metadata," *IEEE Trans. Consum. Electron.*, vol. 68, no. 2, pp. 149–160, May 2022.
[58] R. Mignot, F. Dufaux, and T. Ebrahimi. (2016). *Hybrid Log-Gamma: A New HDR Electro-Optical Transfer Function*. ITU-R Study Group 6 Contribution. [Online]. Available: https://www.itu.int/md/T17-SG06-C-0131/en
[59] ETSI. (Aug. 2021). *High-Performance Single Layer High Dynamic Range (HDR) System for Use in Consumer Electronics Devices; Part 1: Directly Standard Dynamic Range (SDR) Compatible HDR System (SL-HDR1)*. [Online]. Available: https://www.etsi.org/deliver/etsi_ts/103400_103499/10343301/01.04.01_60/ts_10343301v010401p.pdf
[60] ETSI. (Aug. 2021). *High-Performance Single Layer High Dynamic Range (HDR) System for Use in Consumer Electronics Devices; Part 2: Enhancements for Perceptual Quantization (PQ) Transfer Function Based High Dynamic Range (HDR) Systems (SL-HDR2)*. [Online]. Available: https://www.etsi.org/deliver/etsi_ts/103400_103499/10343302/01.03.01_60/ts_10343302v010301p.pdf
[61] ETSI. (Aug. 2021). *High-Performance Single Layer High Dynamic Range (HDR) System for Use in Consumer Electronics Devices; Part 3: Enhancements for Hybrid Log Gamma (HLG) Transfer Function Based High Dynamic Range (HDR) Systems (SL-HDR3)*. [Online]. Available: https://www.etsi.org/deliver/etsi_ts/103400_103499/10343303/01.02.01_60/ts_10343303v010201p.pdf
[62] S. Sekmen and A. O. Akyüz, "Compressed exposure sequences for HDR imaging," in *Proc. 27th Intl. Conf. Central Eur. Comput. Graph., Visualizat. Comput. Vis.*, 2019, pp. 143–151.
[63] G. D. Evangelidis and E. Z. Psarakis, "Parametric image alignment using enhanced correlation coefficient maximization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 10, pp. 1858–1865, Oct. 2008.
[64] M. A. Robertson, S. Borman, and R. L. Stevenson, "Dynamic range improvement through multiple exposures," in *Proc. Int. Conf. Image Process.*, 1999, pp. 159–163.
[65] A. O. Akyüz and A. Gençtav, "A reality check for radiometric camera response recovery algorithms," *Comput. Graph.*, vol. 37, no. 7, pp. 935–943, Nov. 2013.
[66] S. Süsstrunk, R. Buckley, and S. Swen, "Standard RGB color spaces," in *Proc. Final Program IS T/SID Color Imag. Conf.*, 1999, pp. 127–134.
[67] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE CVPR*, Jun. 2016, pp. 770–778.
[68] J. Kim, J. Kwon Lee, and K. Mu Lee, "Accurate image super-resolution using very deep convolutional networks," 2015, *arXiv:1511.04587*.
[69] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced deep residual networks for single image super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jul. 2017, pp. 136–144.
[70] O. T. Tursun, A. O. Akyüz, A. Erdem, and E. Erdem, "An objective deghosting quality metric for HDR images," *Comput. Graph. Forum*, vol. 35, no. 2, pp. 139–152, 2016.
[71] K. Karaduzovic-Hadziabdic, J. H. Telalovic, and R. Mantiuk. (2017). *Multi-Exposure Image Stacks for Testing HDR Deghosting Methods*. [Online]. Available: https://www.repository.cam.ac.uk/handle/1810/261766

[72] U. çogalan and A. O. Akyuz, "Deep joint deinterlacing and denoising for single shot dual-ISO HDR reconstruction," *IEEE Trans. Image Process.*, vol. 29, pp. 7511–7524, 2020.
[73] F. Chollet. (2015). *Keras*. [Online]. Available: https://github.com/fchollet/keras
[74] T. O'Malley, E. Bursztein, J. Long, F. Chollet, H. Jin, and L. Invernizzi. (2019). *Keras Tuner*. [Online]. Available: https://github.com/keras-team/keras-tuner
[75] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
[76] 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.
[77] P. C. Teo and D. J. Heeger, "Perceptual image distortion," in *Proc. 1st Int. Conf. Image Process.*, vol. 2, Nov. 1994, pp. 982–986.
[78] 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, Jul. 2002.
[79] R. V. Hogg, E. A. Tanis, and D. L. Zimmerman, *Probability and Statistical Inference*, vol. 993. New York, NY, USA: Macmillan, 1977.
[80] S. Silk and J. Lang, "Fast high dynamic range image deghosting for arbitrary scene motion," in *Proc. Graph. Interface*. Toronto, ONT, Canada: Canadian Human-Computer Communications Society, 2012, pp. 85–92.
[81] P. Sen, N. K. Kalantari, M. Yaesoubi, S. Darabi, D. B. Goldman, and E. Shechtman, "Robust patch-based HDR reconstruction of dynamic scenes," *ACM Trans. Graph.*, vol. 31, no. 6, pp. 1–11, Nov. 2012.

SELIN SEKMEN received the B.Sc. degree in computer science from Bilkent University, Ankara, Turkey, in 2011, and the M.S. degree in computer engineering from Middle East Technical University (METU), Ankara, in 2014, where she is currently pursuing the Ph.D. degree. In addition to her academic pursuits, she has held the position of a Senior Team Leader with Aselsan, a company associated with the Turkish Armed Forces Foundation. Her primary areas of research interest encompass image processing and computer vision.

AHMET OĞUZ AKYÜZ received the Ph.D. degree in computer science from the University of Central Florida. He is a Professor with the Computer Engineering Department, Middle East Technical University, working primarily in the field of computer graphics and image processing. His research interests include high dynamic range imaging, color, visual perception, image processing, and computer graphics.