Evaluation of Dynamic Range Reconstruction Approaches and a Mobile Application for HDR Photo Capture

Martin Mirbauer*
Supervised by: Jaroslav Křivánek

Faculty of Mathematics and Physics
Charles University
Prague / Czech Republic

Abstract

Digital photography has become widespread with the global use of smartphones. However, most of the captured images do not fully use the camera capabilities and only store the captured photos in a format with limited dynamic range.

Recently several new dynamic range reconstruction methods using convolutional neural networks (CNNs) were published and their performance has not yet been comprehensively compared.

In this paper we present a framework for comparing dynamic range reconstruction approaches and use it to compare the reconstruction quality of the recently published CNN-based approaches. We also implement a mobile HDR camera application and evaluate the feasibility of running the best-performing reconstruction method directly on a mobile device.

Keywords: high dynamic range, inverse tone mapping, deep learning, mobile application development

1 Introduction

Digital cameras have become ubiquitous during the past few years, yet the majority of taken photographs do not fully use the camera capabilities. While the resolution and color accuracy reproduction have been improving, the captured image brightness still lacks the dynamic range of the original scene. Apart from physical sensor sensitivity and saturation limits, further dynamic range reduction is caused by the camera software usually storing the captured photo in a lossy 8-bit format such as JPEG File Interchange Format.

Recently several advanced approaches for HDR image reconstruction from a single exposure LDR image have been published ([6], [7], [26], [13] and [16]). These approaches use deep convolutional neural network to estimate the image areas, which were distorted when capturing the LDR photo. The relative performance of these approaches is unknown and suitability for practical applications needs to be evaluated.

The main contributions of this paper are:

1. A framework for comparing dynamic range reconstruction approaches, which take a single LDR image as an input.
2. A comparison of recently published CNN-based dynamic range reconstruction approaches.
3. Evaluation of feasibility of running inference using a large neural network on a mobile device.
4. Adaptation of HDR capture image processing pipeline to include a dynamic range reconstruction step.
5. An open-source implementation of a smartphone HDR camera application.

2 Background and Related work

2.1 HDR and dynamic range reconstruction

High dynamic range (or HDR) images contain more per-pixel information – usually the incoming luminance is stored in a floating-point number. This additional information can be used in image post-production allowing exposure fine-tuning or emphasizing details in shadowy areas, and even for viewing as HDR displays become widely available. Apart from normal photography producing “framed rectangle” images, the need for HDR images extends to spherical panoramas, which capture the whole 360deg of azimuth and 180deg of elevation, used as environment maps for computer games or image-based lighting for offline rendering.

The HDR to LDR transformation, also called tone mapping (TM), performed by camera software loses image data by reducing the range and precision of pixel values, therefore there is no “correct” inverse transformation. There are approaches to approximate the original luminance values, essentially attempting to find an inverse TM function or attempting to recover lost details. Dynamic range reconstruction methods can be grouped into following categories: image-based and machine-learning-based.

*martinm at cgg.mff.cuni.cz

2.2 Image-based methods

Image-based dynamic range expansion methods make assumptions about the original per-pixel TM function or use heuristics when attempting to invert the TM function.

Global expansion methods do not attempt to reconstruct any missing data, they only expand the luminance range by applying a spatially constant TM function. A simple global dynamic range expansion introduces/amplifies banding or quantization artifacts, which some of the approaches try to reduce. Landis et al. [10] expand the dynamic range using exponentiation: raising luminance values in selected range to a constant power. Akyüz et al. [2] γ-correct the pixel values then linearly scale them to match the required output range – Masia et al. [14] show this approach yields good results when correct γ is used and propose an automatic method to find such a value. Meylan et al. [15] present a piecewise linear mapping function to achieve greater expansion of specular highlights and blur the image around the highlights to reduce the amplified non-linearity artifacts caused by an original TM function.

Local expansion methods assume a spatially varying TM function, whose parameters change based on pixel values of the input image to preserve contrast in both highlights and shadows, e.g. the TM function by Reinhard et al. [17]. Banterle et al. [3] boost highlights by using density estimation of high-value pixels as interpolation weights between the original image and an image expanded using a global method. Rempel et al. [18] aim to prevent highlights "bleeding" over steep gradients introduced by the previous approach – an edge stopping function and flood-fill algorithm is used to preserve borders of the "near-saturated" areas, which need to be expanded. Huo et al. [8] propose to model human retina response, locally adapting to a certain luminance level and perceiving values in limited range around this level, and invert the modeled function.

Clipping-aware reconstruction methods are another family of HDR reconstruction methods, which detect clipped (over- or underexposed) areas and attempt to generate plausible contents. Wang et al. [23] use inpainting to replace clipped areas with patches from the input image. While this method convincingly restores lost details, it requires manual annotation to select areas to use as source of the texture. Jain et al. [9] and Savoy et al. [21] use Internet image retrieval to find similar images, which are then used for inpainting the clipped areas. Rouf et al. [20] attempt to recover lost color information in highlights caused by pixel value saturation in individual color channels. The hue and luminance gradient of the clipped area are extrapolated from the un-clipped border of the clipped region.

2.3 Machine-learning-based methods

Recently published machine learning-based approaches use deep convolutional neural networks (or CNNs) to learn the inverse TM function and to generate details in the clipped areas. CNNs are able to learn to recognize high-level objects in images, which can be internally used for estimating relative luminance of objects, e.g. when an object is classified as a light source such as the Sun or a street lamp, the luminance can be estimated accordingly.

HDRCNN by Eilertsen et al. [6] uses an encoder-decoder network to hallucinate details in over-exposed parts of the image. The encoder consists of convolutional layers from the VGG-16 architecture [22], a then-state-of-the-art classification CNN. By using a verified working, already pre-trained network as a part of the network architecture, only the decoder weights need to be fully trained. To reduce the amount of information the network needs to learn, only the saturated areas in the input image are reconstructed, keeping the well-exposed pixels unmodified.

DrTMO by Endo et al. [7] aims to recover the inverse tone mapping of an image, reconstructing lost details in both highlights and shadows, by using an encoder-decoder architecture. To make training of the network feasible, the network outputs only 8-bit images, predicting increased and decreased exposures of the input image. The resulting HDR image is produced by merging these outputs.

Zhang et al. [26] use an encoder-decoder-based network to increase the dynamic range of an outdoor spherical panorama and to recover the elevation of the Sun from the latent representation. The Sun is expected to be located horizontally in the center of the input image. As the network outputs an HDR image, increasing the complexity of the training, the authors opted to limit the size of the network by fixing the resolution to 128 x 64 pixels, making it unsuitable for high-resolution HDR image reconstruction.

Marnerides et al. [13] attempt to avoid occasional checkerboard artifacts caused by "deconvolution" layers used in the up-scaling decoding stage in encoder-decoder-based networks. The authors propose the **ExpandNet** network architecture without any up-scaling steps – a custom multiscale CNN with 3 branches, each focusing on specific level of detail, which are then merged producing an HDR image. The authors claim this approach handles large clipped areas better than other CNN-based dynamic range reconstruction approaches, producing fewer artifacts.

Ning et al. [16] aim to solve the issue of limited availability of HDR images[1] by using a generative adversarial network, which can be trained on LDR images. The architecture consists of two networks trained simultaneously: an inverse tone mapping network and a discriminator network evaluating the reconstruction quality by distinguishing generated HDR images from real HDR images.

2.4 LDR-to-HDR reconstruction evaluation

There are comparisons of the machine-learning-based reconstruction approaches with traditional dynamic range expansion methods in the ML-based approaches' papers

[1] needed for supervised learning: training on LDR-HDR image pairs, as used in previous approaches

([6], [7], [26], [13] and [16]) showing superior results of the proposed methods. Two of the papers also compare the proposed network model with previously published ML-based reconstruction methods ([13] and [16]). However, none of the comparisons include all the approaches and a numeric evaluation on a larger dataset.

3 HDR reconstruction evaluation framework

The goal of the evaluation framework is to compare the relative reconstruction quality of several dynamic range reconstruction algorithms, which take a single LDR image as their input. The framework should take one or more HDR images – equirectangular-projected spherical panoramas and output error metric results for each input image and each reconstruction approach, needed for numeric comparison. Furthermore it should provide a graphical representation of reconstruction accuracy to simplify visual detection of systematic errors in reconstruction methods' outputs, and should provide a way to easily compare the reconstructed output images to other reconstructions of the same image and to the original HDR image. The proposed evaluation process consists of the stages described in the following paragraphs.

The aim of the **Import** stage is to convert the input HDR image into a common format and resolution. The resolution of the input images is unified by resampling them to prevent unfair error metric results caused by different image pixel counts.

The **Degrade** stage simulates dynamic range and luminance precision degradation by converting the input image into a lossy LDR format. The JPEG compression and sRGB color space is used, as it represents the majority of taken photos. The compression quality (DCT coefficient pre-quantization multiplier) is set to maximum to reduce the compression artifacts effect on the evaluation metrics.

The **Reconstruct** stage handles dynamic range reconstruction by running each reconstruction algorithm on a degraded image. Along with each reconstruction result, a non-reconstructed image is passed to the next stage to serve as a baseline for metrics evaluation.

The **Evaluate** stage compares the reconstruction result to the ground truth image by computing error metrics and generates *luminance* and *color mapping* plots.

To measure the similarity of two HDR images we need to define a set of metrics to use. We use two error metrics: SSIM [24] to measure loss of local texture details and RMSE to measure non-perceptual difference. The HDR-VDP-2 metric by Mantiuk et al. [12], used in some of the ML-based approaches' papers, is the appropriate similarity metric, because it computes perceptual quality difference of HDR images. However using the provided implementation would require us to use proprietary Matlab software, limiting the framework portability and results reproducibility. As we have not found an independent implementation of the metric, we decided not to include the metric in the evaluation but rather to design the framework to make adding new metrics easy.

Apart from numeric evaluation of reconstruction results, two graphs representing luminance and color transformation are generated. A *luminance mapping plot* is a scatter plot showing the dependency of reconstructed per-pixel luminance on the the ground truth luminance (each pixel is drawn as a dot at corresponding position). Along with the image data, a function $f(x) = x$ is plotted to help spot any non-linearity of the reconstruction algorithm – an optimal reconstruction algorithm should produce points only on this line. Similarly by drawing each color channel separately instead of combining the color channel values into luminance, drawing red, green and blue dot for each pixel, a *color mapping plot* is generated. It shows the distortion of individual color channels allowing the user to inspect whether reconstruction errors are color-dependent. These plots may help the user spot any global pixel value distortion or e.g. in a specific case of a photo of a landscape, the color mapping plot may help discriminate whether a reconstruction algorithm fails to recover the dynamic range in the sky or grass area. An example of the mapping graphs is shown in Figure 4 in the Results section.

In the **Render** stage the reconstructed HDR image from the previous stage is used as an environment map when rendering a simple 3D scene using *pbrt* by Pharr et al. [1] to visualize the effects of using the reconstructed image for image-based lighting and to find potential discontinuities at the spherical panorama image edges. The 3D scene consists of three glossy spheres with different surface roughness placed on a checkerboard-textured plane. The scene rendering is depicted in Figure 1. High intensity light sources, which occupy a small area on the environment map image, such as Sun, cause crisper shadows and larger highlights on glossy surfaces. The relative intensity of such sources can therefore be estimated by visually inspecting the size of the highlights. Comparing the rendering outputs (at a fixed exposure level) produced from a reconstructed image and the corresponding ground truth image can be used to assess the quality of the reconstruction algorithm. The scene is rendered and stored with high dynamic range to allow simulation of different exposures when viewing.

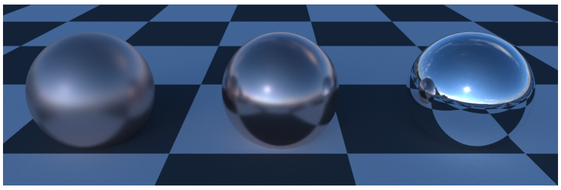


Figure 1: Rendering of the testing scene using the reconstructed image applied as an environment map for image-based lighting.

The **View** stage prepares a user-viewable version of the visual reconstruction results along with the ground truth image for comparison. The user can set an exposure value and switch between images (individual reconstructions and the ground truth image). Switching between images maintains the exposure value allowing the user to compare specific parts of the image. The HDR image viewer is based on PFSTools by Mantiuk et al. [11] is based on web technologies (HTML5 and JavaScript/ECMAScript) for platform independence and simple result publication.

Finally the numeric evaluation results are presented to the user. A web-page is generated containing an interactive table containing numeric evaluation results along with links to open the HDR image viewer with a given set of images and color/luminance mapping graphs. The table allows sorting[2] by clicking on a column header; secondary sorting criteria can be set by holding the shift key when clicking a header. This allows the user to quickly find best and worst reconstruction results, which can be inspected in more detail. Multi-criteria sorting simplifies the process of finding the ranking of individual reconstruction algorithms by choosing the Ground truth image column as a primary key and an error metric as a secondary key.

4 HDR camera application

We have created a camera application, which is able to capture the highest possible dynamic range of a scene and evaluated the feasibility of recovering details that could not be captured. The application can capture a set of HDR images with the same exposure parameters suitable for stitching to produce a spherical panorama usable for image-based lighting used in rendering.

The dynamic range of an image is limited by the sensor sensitivity and saturation and by the storage format, which can be overcome by bracketing – taking multiple LDR images of the same scene with different settings to capture details in both highlights and shadows. Dedicated cameras usually allow manual setting of exposure time, ISO (sensitivity) and aperture. In case of smartphone cameras the aperture is fixed. Bracketed photos can be merged by linearization[3], multiplication by their individual exposures and computing the weighted average prioritizing well-exposed pixel values. The process is described by Debevec et al. [4] or Robertson et al. [19] along with an algorithm to find the inverse camera response function. Dynamic range can be further improved beyond the physical sensor limits (e.g. the shortest exposure time and the lowest supported ISO) by hallucinating the clipped pixels using a CNN-based approach. Apple iOS was chosen as the target **platform** because of author's previous experience with development for this platform, lower number of different devices to support and because the author already owned a compatible smartphone. The application has been developed and tested on an iPhone SE with the iOS 10.2 operating system installed. The device has a dual core CPU clocked at 1.8 GHz and 2 GiB of RAM and a back camera[4] with 4032-by-3024 pixels sensor with supported exposure time range 0.000013 s – 0.333333 s and ISO range 23 – 1840. The camera API offers two modes of operation influencing what kind of exposure parameters is used: "Auto" mode[5], where the application specifies the requested list of EV offsets from the automatically determined $+0\,EV$ level, and "Manual" mode[6], which takes list of arbitrary (ISO, exposure time) tuples.

HDR images can be captured using various **camera applications**. Although some applications on both iOS and Android support taking an "HDR" image using multiple exposures, the software usually stores the merged image tone-mapped in an LDR format. This is the case of the built-in camera application, which automatically activates the "HDR" mode if the scene dynamic range significantly exceeds the sensor limits. True HDR images can be captured using third-party camera applications available in the App Store that allow manual exposure parameters settings (e.g. Musemage), automatic exposure bracketing (ProCam) or bracketed capture with integrated merging of the bracketed photos (Adobe Lightroom CC). Adobe Lightroom CC takes 3 RAW photos with different exposures, merges them and saves the result to a 32-bit DNG file[7]. However, all found applications are closed-source so they can not be modified to incorporate a dynamic range reconstruction step into the pipeline.

4.1 HDR reconstruction approach selection

The most important criterion when choosing an HDR reconstruction approach was the performance and capabilities of each approach. As shown below in the Results section, HDRCNN is the best-performing of the compared networks. However, as it only restores clipped highlights without modifying the un-saturated areas with shadows, DrTMO and ExpandNET also need to be considered.

The implementations of the DrTMO and ExpandNet CNNs use the Chainer neural network framework and PyTorch, respectively and are not compatible with iOS, requiring network model conversion to a supported framework such as CoreML[8], leaving TensorFlow-based HDR-CNN the only option. We exported the HDRCNN model for use in a natively-compiled application with network

[2] http://tablesorter.com/docs/

[3] applying inverse camera response function

[4] Camera specifications: https://developer.apple.com/library/archive/documentation/DeviceInformation/Reference/iOSDeviceCompatibility/Cameras/Cameras.html#//apple_ref/doc/uid/TP40013599-CH107-SW26

[5] AVCaptureAutoExposureBracketedStillImageSettings

[6] AVCaptureManualExposureBracketedStillImageSettings

[7] http://blogs.adobe.com/lightroomjournal/2017/03/lr-mobile-update-raw-hdr-capture-mode-for-ios-and-android.html\#comment-216145

[8] https://developer.apple.com/documentation/coreml

inputs and outputs mapping being the only modification.

System requirements need to be also considered due to limited memory and processing power of mobile devices. Both ExpandNet and DrTMO have higher memory footprint than HDRCNN[9]. This would require modifications to the network architectures[10] to reduce the requirements. Because of these reasons we selected HDRCNN as the most applicable approach.

As running the inference on the full-resolution image on the mobile device is infeasible[11], we reduced the inference resolution by splitting the image into tiles of 320 by 320 pixels, reducing the memory consumption to approx. 750 MiB, each tile inference taking approx. 2.5 s. The inference of the image split into tiles takes approx. 5 minutes, making it a non-interactive process, however, as only a few tiles usually contain clipped pixels, the rest can be skipped. The remaining drawbacks can be fixed[12] in future development of the application.

4.2 Capture pipeline

The HDR capture pipeline is based on the "Stack-Based Algorithms for HDR Capture and Reconstruction" chapter from a book by Dufaux et al. [5] adapted to fit the use-case of a mobile application, which simplifies taking environment maps, with an added dynamic range reconstruction step. The pipeline phases and data passed between individual phases are outlined in the Figure 2.

The aim of the **Metering** phase is to find parameters of the base exposure, where most of the pixels values are not clipped and optionally to measure the dynamic range of the scene. In our application the base exposure parameters can be found automatically[13] or set by the user when setting the number of down-/up-exposure steps and the EV step size, describing the dynamic range of the scene.

The **Exposure preparation** phase generates a set of exposure parameters to capture, covering the requested dynamic range. The exposure time and ISO are dependent on each other in a way, that when the exposure time is divided by n, ISO needs to be multiplied by n to keep the same exposure value. Increasing the EV by 1 can be achieved by doubling the exposure time or ISO.

We use the "Manual" mode for capturing images so we can reuse parameters for multiple shoots[14]. In addition to the base exposure parameters we compute additional (exposure time, ISO) tuples to capture the dynamic range requested by the user: up-exposure parameters are computed by increasing time up to 1/17 s, then by increasing the ISO[15]; for down-exposure we decrease the ISO until reaching the lower limit to reduce the noise, then we decrease the exposure time.

The **Capture** phase captures the required exposures of the scene, producing a set of images, also called a stack.

While the built-in camera supports RAW capture with 12-bit precision, we opted to use a pre-processed Y'CbCr format with only 8-bit precision due to the simpler processing[16] and due to the fact dynamic range reconstruction approaches accept 8-bit input images. This decision does not limit the achievable dynamic range, it only increases the required number of bracketed captures to achieve it.

The **Image alignment** phase compensates for camera movement between captures of individual images. We use the alignment approach by Ward et al. [25] assuming translation is sufficient as the majority of hand-held captured sequences do not require rotational alignment.

After image stack alignment the **dynamic range reconstruction** phase can occur. The easiest-to-implement approach would be to execute the dynamic range reconstruction remotely on a server to work around speed and compatibility issues. However, there would be trade-offs in form of hosting costs and relatively high bandwidth requirements[17] also impacting users of the application. Therefore we decided to run the dynamic range reconstruction directly on a mobile device.

First we need to determine **the position of HDR reconstruction in the pipeline**. The dynamic range can be reconstructed either before or after merging individual exposures. However, neither of these options is straightforward: as both the reconstruction and merging expect or require LDR inputs, either needs to be adapted.

One option would be to merge the exposures first and then feed the resulting HDR image into a neural network for further dynamic range expansion. Although this approach may theoretically produce better results[18], training the CNN to process such inputs would be necessary.

The option we chose is to recover the clipped highlights of the lowest-exposure-value image[19] followed by merging the resulting image with the captured images to produce one HDR image using a modified merging algorithm.

In the **Merge** phase we merge the reconstructed HDR image with LDR images captured with different exposures. We have modified the merging algorithm by Robertson et al. [19] to also accept HDR inputs. The modification consists of special handling of HDR inputs: we generate multiple LDR images with simulated exposures covering

[9] DrTMO requies up to 13.48 GiB of RAM for the provided 1536-by-1024 pixels image Forest.png.

[10] E.g. DrTMO may be modified to output less exposures at the same time, needing the network evaluation to be run multiple times.

[11] HDRCNN uses more than 14 GiB of RAM for 4032x3024 images.

[12] Tiling artifacts can be removed by overlapping the tiles and interpolating the inference results; speed can be improved by re-implementing the model in a framework supporting GPU/TPU acceleration.

[13] We use the "Auto" mode to find base exposure parameters: we let the system libraries find the "correct" exposure: take a metering image with the "+0 EV" exposure then access the image metadata.

[14] This is crucial for stitching when taking panoramas.

[15] matching the built-in Camera application behavior

[16] There is no need for debayering and white-balance steps.

[17] Upload LDR: ˜2-5 MB (lossy JPEG LDR image), download HDR: ˜35-40 MB (Radiance HDR (RGBE) format, better compression may be achieved with wavelet transform used by OpenEXR).

[18] The network would have access to the full captured dynamic range.

[19] essentially predicting even lower-exposure-value images, allowing us to use the pre-trained model

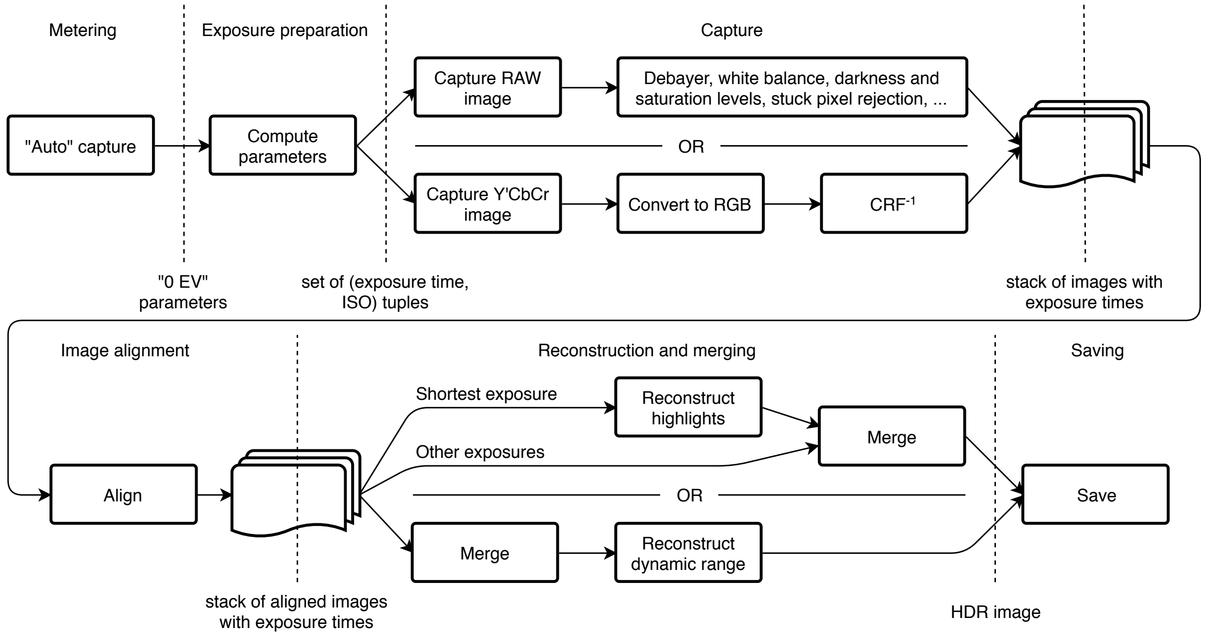


Figure 2: Schematic diagram of the HDR capture pipeline.

the input dynamic range, which are then merged without applying the found inverse camera response function.[20]

Finally we **Save** the merged image. As the built-in photo library is not suited for storing HDR images, we store the captured photos in an application directory accessible from the computer using iTunes. We chose the Radiance HDR (RGBE) image format[21] for its compatibility and relatively compact data storage, while the encoding not being too computationally demanding.

5 Results

5.1 Comparison of existing ML-based dynamic range reconstruction methods

Out of the five deep learning-based high dynamic range reconstruction approaches referenced in the *Related work* chapter only four approaches have a public implementation. As the neural network by Zhang et al. [26] only supports low resolution inputs and outputs, making the output images unsuitable for use as environment maps, it was excluded from the evaluation. The remaining methods, which provide pre-trained neural networks and support high resolution images, are: HDRCNN [6], DrTMO [7] and ExpandNET [13]. As the pre-trained models are used, the reconstruction quality depends on the training dataset size and image selection. This issue was disregarded due to too high computational demands to train all the networks and may be addressed in a future work.

The evaluation **dataset** consists of HDR spherical panoramas from the following sources: *NoEmotion HDRs – Dayhdr*[22] and *HDRLabs sIBL Archive*[23]. This set of HDR images contains a combination of outdoor scenes with direct sunlight and sky with and without clouds as well as indoor images.

The high dynamic range reconstruction evaluation framework was used to process (import, degrade, reconstruct using each approach and compute error metrics for) each image in the dataset to measure/compare the quality of reconstruction methods.

We computed **average metrics** to get an aggregated performance of individual dynamic range reconstruction methods and to spot any global issue with the networks. The results are shown in Table 1 with the best achieved value for each metric highlighted in bold text.

The *Average RMSE (unsaturated)* of *LDR (baseline)* is non-zero because of quantization of the HDR image[24]. All reconstruction methods introduce an error to well-exposed (unsaturated) pixels. HDRCNN is the only method, which globally outperforms the baseline (both based on RMSE and SSIM). The distortion is so large it negates any metric value improvements caused by the dynamic range reconstruction. An example of the distortion is shown in Figure 3. Before we proceed with further comparison, we need to “linearize” the output of reconstruction methods.

Linearization transforms the reconstructed pixel val-

[20] The modified code is in the MergeRobertsonUnlimited repository.

[21] `http://radsite.lbl.gov/radiance/refer/filefmts.pdf`

[22] `http://noemotionhdrs.net/hdrday.html`

[23] `http://hdrlabs.com/sibl/archive.html`

[24] The size of an 8-bit value step is 1/256 = 0.00390, because rounding to the nearest value is used, the upper bound for per-pixel error is half a step, which is 1/256/2 = 0.00195. The actual *Average RMSE (unsaturated)* is less, because not all pixel values are “exactly in the middle between steps” – the values are likely uniformly distributed.

Method	Avg. RMSE (all)	Avg. RMSE (saturated)	Avg. RMSE (unsat.)	Avg. SSIM
LDR	0.5533	2.2735	**0.0014**	0.9866
HDRCNN	**0.4964**	**2.0438**	0.0237	**0.9885**
DrTMO	0.6189	2.2613	0.1620	0.8249
ExpandNET	0.5986	2.4114	0.0771	0.9303

Table 1: **Average metrics** results: comparison of "raw" reconstruction methods (without linearization). LDR (baseline) is the degraded image without any reconstruction. RMSE is computed separately for all, only saturated, and only unsaturated pixels.

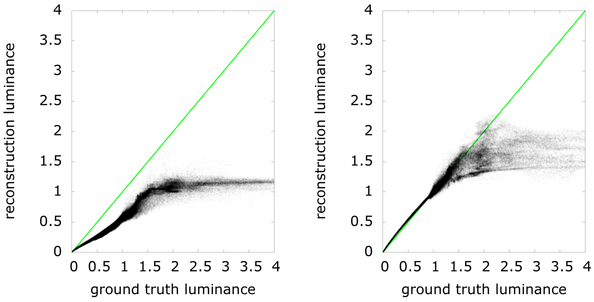


Figure 3: Luminance mapping graph for the image GravelPlaza_REF, reconstructed by DrTMO (left) and HDRCNN (right), showing non-linearity artifacts.

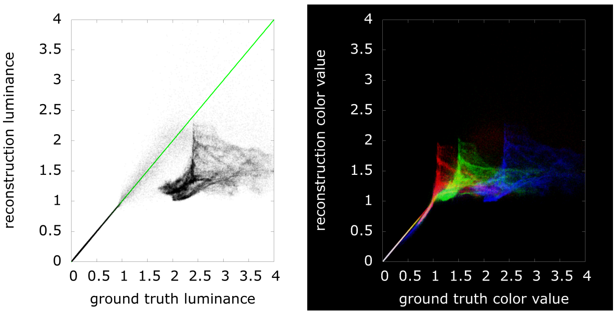


Figure 4: Luminance/color mapping graph showing a discontinuity artifact of the HDRCNN-reconstructed NarrowPath_3k image. The blue channel-sky is the most affected.

ues to "straighten" the luminance mapping curve to fit the $f(x) = x$ line, based on the values in the [0,1] range in the input LDR image, only affected by quantization. In case of HDRCNN the distortion in the [0,1] range is constant, caused by color space conversion[25]. Other approaches (DrTMO and ExpandNet) vary the distortion between pictures so we find regression parameters per-image. In case of DrTMO, which outputs multiple LDR exposures, we also calibrate camera response function for correct merging into a single HDR image.

The best-performing transformation function[26] over the dataset is $f(x) = a \cdot x$, fixing the curve slope, i.e. correcting the exposure. Although the luminance mapping curve of DrTMO nor ExpandNet are linear, using other functions introduced errors in other parts of the curve, yielding sub-baseline results. The improved metrics values after linearization are shown in the Table 2.

To compare the relative performance aggregated over all test cases we computed **average ranking** results. As seen in the Table 3, the HDRCNN approach performed the best for both metrics. Although DrTMO and ExpandNet RMSE rank is on average between second and third place, beating the baseline, they yield significantly worse results than HDRCNN. Based on the SSIM metric DrTMO and ExpandNet rank on average worse than LDR (baseline), meaning running no reconstruction at all yields better result. This is due to the pixel value distortion, although improved by our imperfect linearization of the $[0,1]$ area, causing a loss of details.

Using the HDR reconstruction evaluation framework we made following **observations**:

- All approaches managed to detect saturated areas and increase the luminance values.
- All approaches have a problem with large saturated areas, causing discontinuity in the luminance mapping graph, as shown in Figure 4.
- The first line of ExpandNet output image is almost black, exhibiting as points along the x axis of the luminance mapping plot. The issue has been reported to the authors.

The best-performing dynamic range reconstruction approach is HDRCNN, leading in both error metrics, outperforming other reconstruction methods for the vast majority of input images. The remaining two methods, while outperforming the baseline in the RMSE metric, reduce the SSIM metric results below the baseline level. HDRCNN produces the lowest distortion in unsaturated areas, desirable for merging the image with other exposures in our HDR camera application, as well as the best recovery of clipped highlights.

Method	Avg. RMSE (all)	Avg. RMSE (saturated)	Avg. RMSE (unsat.)	Avg. SSIM
LDR	0.5533	2.2735	**0.0014**	0.9866
HDRCNN	**0.5023**	**2.0741**	0.0030	**0.9933**
DrTMO	0.5487	2.2709	0.0195	0.9800
ExpandNet	0.5437	2.2603	0.0252	0.9819

Table 2: **Average metrics** results for each metric and reconstruction approach after output linearization.

[25] HDRCNN expects linear LDR image input, while images in sRGB color space are more common, thus used in the *degrade* step.

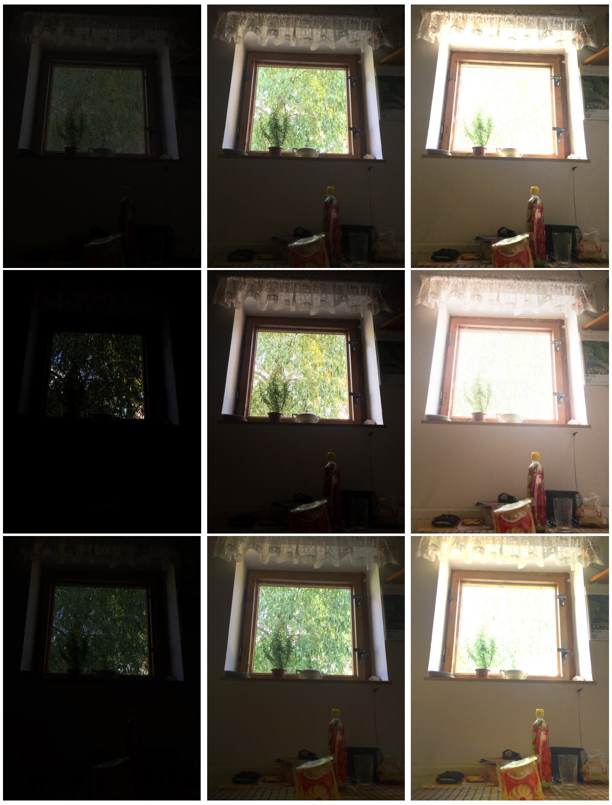
[26] See color_mapping/fix_mapping_auto.py for the list of attempted function fits

Method	Average RMSE rank	Average SSIM rank
LDR	3.3376	2.7532
HDRCNN	1.2337	1.0259
DrTMO	2.6493	2.9610
ExpandNet	2.7792	3.2597

Table 3: **Average ranking** of each reconstruction approach and metric, after output linearization.

5.2 HDR Camera Application

To compare our HDR camera application with other existing applications, we have captured a scene with high dynamic range – an interior with a window on a sunny day – using three applications: the built-in default Camera application, Lightroom CC and our HDR camera application[27]. The comparison of the captured images is shown in Figure 5.



Frst row: default Camera application using "HDR" mode. Second row: Lightroom CC in HDR mode. Third row: the image produced by our HDR camera application. Each row contains three columns with shifted exposure values. From left to right: $-5\ EV$, $+0\ EV$ and $+3\ EV$.

Figure 5: Comparison of HDR camera applications.

The default iPhone Camera application in "HDR" mode captures only a tone-mapped LDR image lacking highlights in the $-5\ EV$ image. The captured dynamic range is improved[28] preserving some details in both highlights and shadows – the window nor the interior is completely clipped. The shadows contain noise visible on the wall below the window when the exposure is shifted by $+3\ EV$.

In comparison, Lightroom CC produces a true HDR image. While Lightroom CC uses only 3 exposures, it manages to sufficiently cover the dynamic range of the scene, which is achieved thanks to the RAW image capture and processing. There is, however, visible noise in the shadows, observable in the $+3\ EV$ shifted exposure. This is caused by the lack of higher-exposure-value captures.

The image produced with our HDR camera application captures the whole dynamic range of the scene[29]. There is color inaccuracy when compared to the other two applications. Our HDR application also suffers from different exposure scaling reducing contrast. Both issues are probably caused by inaccurate camera response calibration.

Our HDR camera application is slower compared to the default camera application. Capturing a single image takes approximately 2 seconds due to merging being executed even when there is only one exposure. Capturing 4 exposures and merging them takes approximately 10 s. Optimization is left as a future work.

6 Conclusions

We have designed and implemented a framework for evaluating the reconstruction quality of multiple HDR reconstruction algorithms and used it to compare three recently published machine-learning-based HDR reconstruction approaches.

We then ported the best-performing neural network to a smartphone to evaluate its usability on a device with limited computational power. Reconstruction of full-resolution images is too computationally demanding for a mobile device, but running the inference after splitting the image to smaller tiles only on the parts which contain saturated pixels is feasible. We have also created a mobile camera application for capturing HDR images, fully using the built-in camera dynamic range.

The HDR reconstruction framework and HDR camera application source code is available online: `https://bitbucket.org/hdri/dp_root`.

[27] The HDR image produced by our HDR camera application was exposure-shifted by $-2\ EV$ in order to match the brightness of other applications' $+0\ EV$ brightness level.

[28] compared to the "HDR" mode being disabled

[29] The lowest-exposure-value image contained no clipped highlights and the highest-exposure-value contained no clipped shadows. This means there would be no benefit in running the dynamic range reconstruction algorithm on this scene. A scene with directly visible Sun, specular reflections or caustics would benefit from expansion of dynamic range beyond the capability of the camera sensor.

References

[1] pbrt, Version 3: Source code for pbrt, the renderer described in the third edition of ”Physically Based Rendering: From Theory To Implementation”, by Matt Pharr, Wenzel Jakob, and Greg Humphreys. `https://github.com/mmp/pbrt-v3`. Accessed: 2018-07-20.

[2] Ahmet Oğuz Akyüz, Roland Fleming, Bernhard E Riecke, Erik Reinhard, and Heinrich H Bülthoff. Do hdr displays support ldr content?: a psychophysical evaluation. *ACM Transactions on Graphics (TOG)*, 26(3):38, 2007.

[3] Francesco Banterle, Patrick Ledda, Kurt Debattista, and Alan Chalmers. Inverse tone mapping. In *Proceedings of the 4th international conference on Computer graphics and interactive techniques in Australasia and Southeast Asia*, pages 349–356. ACM, 2006.

[4] Paul E Debevec and Jitendra Malik. Recovering high dynamic range radiance maps from photographs. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 369–378. ACM Press/Addison-Wesley Publishing Co., 1997.

[5] Frédéric Dufaux, Patrick Le Callet, Rafal Mantiuk, and Marta Mrak. *High dynamic range video: from acquisition, to display and applications*. Academic Press, 2016.

[6] Gabriel Eilertsen, Joel Kronander, Gyorgy Denes, Rafał K Mantiuk, and Jonas Unger. Hdr image reconstruction from a single exposure using deep cnns. *ACM Transactions on Graphics (TOG)*, 36(6):178, 2017.

[7] Yuki Endo, Yoshihiro Kanamori, and Jun Mitani. Deep reverse tone mapping. *ACM Trans. Graph*, 36(6), 2017.

[8] Yongqing Huo, Fan Yang, Le Dong, and Vincent Brost. Physiological inverse tone mapping based on retina response. *The Visual Computer*, 30(5):507–517, 2014.

[9] Garima Jain, Anand Plappally, and Shanmuganathan Raman. Internethdr: Enhancing an ldr image using visually similar internet images. In *Communications (NCC), 2014 Twentieth National Conference on*, pages 1–6. IEEE, 2014.

[10] Hayden Landis. Production-ready global illumination. *Siggraph course notes*, 16(2002):11, 2002.

[11] Rafał Mantiuk, Grzegorz Krawczyk, Radosław Mantiuk, and Hans-Peter Seidel. High dynamic range imaging pipeline: Perception-motivated representation of visual content. In Bernice E. Rogowitz, Thrasyvoulos N. Pappas, and Scott J. Daly, editors, *Human Vision and Electronic Imaging XII*, volume 6492 of *Proceedings of SPIE*, San Jose, USA, February 2007. SPIE.

[12] Rafat Mantiuk, Kil Joong Kim, Allan G Rempel, and Wolfgang Heidrich. Hdr-vdp-2: a calibrated visual metric for visibility and quality predictions in all luminance conditions. In *ACM Transactions on graphics (TOG)*, volume 30, page 40. ACM, 2011.

[13] Demetris Marnerides, Thomas Bashford-Rogers, Jonathan Hatchett, and Kurt Debattista. Expandnet: A deep convolutional neural network for high dynamic range expansion from low dynamic range content. In *Computer Graphics Forum*, volume 37, pages 37–49. Wiley Online Library, 2018.

[14] Belen Masia, Ana Serrano, and Diego Gutierrez. Dynamic range expansion based on image statistics. *Multimedia Tools and Applications*, 76(1):631–648, 2017.

[15] Laurence Meylan, Scott Daly, and Sabine Süsstrunk. The reproduction of specular highlights on high dynamic range displays. In *Color and Imaging Conference*, volume 2006, pages 333–338. Society for Imaging Science and Technology, 2006.

[16] Shiyu Ning, Hongteng Xu, Li Song, Rong Xie, and Wenjun Zhang. Learning an inverse tone mapping network with a generative adversarial regularizer. *arXiv preprint arXiv:1804.07677*, 2018.

[17] Erik Reinhard, Michael Stark, Peter Shirley, and James Ferwerda. Photographic tone reproduction for digital images. *ACM transactions on graphics (TOG)*, 21(3):267–276, 2002.

[18] Allan G Rempel, Matthew Trentacoste, Helge Seetzen, H David Young, Wolfgang Heidrich, Lorne Whitehead, and Greg Ward. Ldr2hdr: on-the-fly reverse tone mapping of legacy video and photographs. In *ACM transactions on graphics (TOG)*, volume 26, page 39. ACM, 2007.

[19] Mark A Robertson, Sean Borman, and Robert L Stevenson. Dynamic range improvement through multiple exposures. In *Image Processing, 1999. ICIP 99. Proceedings. 1999 International Conference on*, volume 3, pages 159–163. IEEE, 1999.

[20] Mushfiqur Rouf, Cheryl Lau, and Wolfgang Heidrich. Gradient domain color restoration of clipped highlights. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 7–14. IEEE, 2012.

[21] Florian M Savoy, Vassilios Vonikakis, Stefan Winkler, and Sabine Süsstrunk. Recovering badly exposed objects from digital photos using internet images. In *Digital Photography X*, volume 9023, page 90230W. International Society for Optics and Photonics, 2014.

[22] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[23] Lvdi Wang, Li-Yi Wei, Kun Zhou, Baining Guo, and Heung-Yeung Shum. High dynamic range image hallucination. In *Proceedings of the 18th Eurographics conference on Rendering Techniques*, pages 321–326. Eurographics Association, 2007.

[24] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[25] Greg Ward. Fast, robust image registration for compositing high dynamic range photographs from handheld exposures. *Journal of graphics tools*, 8(2):17–30, 2003.

[26] Jinsong Zhang and Jean-François Lalonde. Learning high dynamic range from outdoor panoramas. *arXiv preprint arXiv:1703.10200*, 2017.