HDR Techniques with Dynamic Scenes for Mobile Imaging

Nicholas Gaudio
Department of Electrical Engineering,
Stanford University

Suzannah Osekowsky
Department of Electrical Engineering,
Stanford University

Abstract

Cameras sensors suffer from low dynamic ranges compared to the dynamic range of the human eye. These low dynamic range sensors result in a loss of detail in the brightest (overexposed) and darkest (underexposed) regions of the scene. There are various well performing high dynamic range algorithms that are employed to mitigate camera sensor dynamic range shortcomings. Traditional high dynamic range algorithms require capturing and fuzzing multiple images of differing exposure times. The bane of these methods is that they require a long overall capture process that leads to poor performance for dynamic scenes with ghosting and tearing artifacts arising. Optical flow methods are also unable to remove these motion artifacts due to the lack of correspondence between images. Resultantly, it is proposed to the learn complex process of motion artifact removal through the use of convolutional neural networks. In this paper, we present an HDR algorithm for dynamic scenes for mobile imaging.

1. Introduction

In situations such as when one takes a picture indoors with sunlight coming in through a window, parts of the image are very bright (the window) and others may be very dark (an unlit corner). Many captured scenes by cameras are of a high dynamic range. This often times leads to images with under- and overexposed regions, as many camera sensors have a limited dynamic range that is often lower than the scene's dynamic range. Resultantly, high dynamic range (HDR) imaging techniques maintain detail in photographs of scenes from a high dynamic range (an image with very overexposed and underexposed areas).

Many techniques involve capturing multiple low dynamic range (LDR) images, taken at different exposures, which are then merged to form a high dynamic range image. High dynamic range imaging performs extremely well on static scenes as the LDR images are able to merge without suffering from motion artifacts. Well perfected optical flow algorithms [1][2] have been developed to perform frame alignment between the series of LDR images to successfully reduce the effects of hand motion between captures.

Figure 1: The result of MATLAB's native HDR and tone mapping algorithms on dynamic scenes.

Optical flow algorithms however suffer when the scene itself is dynamic as the dynamic areas of the scene results in a lack of correspondence between the LDR images upon merging. This lack of correspondence in dynamic scenes results in motion artifacts such as ghosting and tearing as seen in Figure 1 which employed the naïve, native MATLAB [3] HDR and tone mapping algorithms. The work done by Kalantari et al. [4] suggests employing a convolutional neural network to learn the complex process of removing the artifacts that result from scene motion between and during the three LDR captures that could not be corrected for through methods such as optical flow.

In this paper, we present a modification to this existing HDR technique for dynamic scenes, which was trained on and performed with the Canon EOS-5D Mark III camera, for the Google Pixel 2. Standalone camera sensors such as SLRs have higher dynamic ranges than mobile device sensors such as the Google Pixel 2. Resultantly, it is even more imperative for mobile devices to leverage proper HDR algorithms.

2. Related Work

The Google Pixel 2 has an existing HDR framework (HDR+) [5] that takes in a set of underexposed frames at the same exposure and fuses them into an image. While they were able to reduce motion artefacts and implement their HDR techniques efficiently, their dynamic range improvements were not as good as exposure-varied methods. Reducing the exposure, and taking the same exposure for all the frames, means they are collecting less data from the scene than they could be. If part of the camera

Figure 2: A scene example of the dynamic set, static set, input LDR set of images, and HDR ground truth image.

sensor is saturating light or dark at that exposure, taking multiple frames will not recover the missing data. Varying the exposure time should allow us to capture more data at the underexposed and overexposed areas of any given image.

There are many techniques for HDR imaging on mobile, some of which solve the ghosting problem well, and others with more radical improvements in level of detail. One of the latter was developed in 2008 by the Media Laboratory/Nokia Research Center [6]. Their method was reasonably successful, but was limited by memory availability on mobile platforms at that time. The technology improvements since 2008 (1 year after the release of the first iPhone) will allow us to store 3 images in parallel and combine all 3 at once, rather than the stacking approach utilized by the Nokia system. They also have an exposure determination system that requires the camera to be still as it captures many incremental exposures of the scene in order to determine which ones are required to cover the desired dynamic range, which is problematic for many use cases of mobile imaging.

Our method is based on a technique from Kalantari, et al. at UCSD that applies a fully convolutional neural network to dynamic scenes in order to create HDR images with reduced motion artifacts. Their codebase was developed in MATLAB. They accept RAW images from a Canon EOS-5D Mark III DSLR and create an HDR image of a dynamic scene. They constructed a small, four layer CNN with MatConvNet that takes in set of brightness-adjusted, frame aligned LDR images and their corresponding HDR images (obtained through simple gamma correction, $\gamma = 2.2$) and to produce pixelwise blending weights for each of the source images to use in an HDR formation algorithm which is detailed in Methods.

3. Our Method

We have adapted the Kalantari, et al. *Deep High Dynamic Range Imaging of Dynamic Scenes* algorithm approach which is trained on the Canon EOS-5D Mark III distribution for the Google Pixel 2's distribution. In order to accomplish this, we needed to collect an analogous HDR dataset as the only HDR dataset of dynamic scenes of was captured by Kalantari, et al. but was of the distribution of the Canon EOS-5D Mark III DSLR and not of a mobile imaging device.

3.1. Build and Preprocess Dataset

The inputs to the Canon EOS-5D Mark III trained network were 74 scenes, each of which are made up of three dynamic LDR images separated by two or three stops and a "ground truth" HDR image made from three similar, but static scenes. The middle exposed dynamic scene image was then replaced with the middle exposed static scene image as seen in Figure 2. Optical flow non-rigid frame alignment was performed to align the under- and overexposed images to the middle-exposed image for both the dynamic and static sets of images, for each scene, to reduce the effects of things such as hand motion between and during captures. Since exposure constancy is required

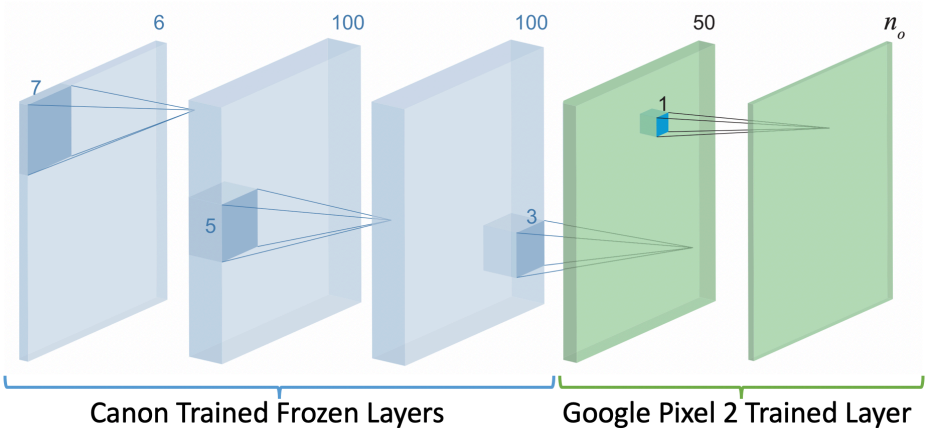


Figure 3: The transfer learned convolutional neural network architecture.

for the optical flow algorithm, the darker images were raised to the brighter ones though the following equation:

$$I_{low,high} = clip(I_{low}\frac{t_{high}}{t_{low}})$$

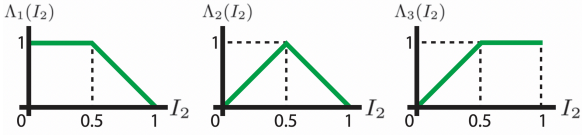
where I is the image and t is the exposure time. Lastly, the images were gamma corrected ($\gamma = 2.2$) to map the images into a more perceptual domain for our eyes.

We captured 16 scene a dataset with these characteristics (three differently exposed dynamic images and three differently exposed static images in each scene) using a Google Pixel 2 and an app called Camera FV-5. The app allowed us to capture raw images (.DNG files) and capture exposure-bracketed images all of which were separated by two stops. That is, the app allowed us to capture three sequential images at different exposure levels that we could customize from the in-app settings. The now preprocessed set of dynamic images in the LDR and HDR domains (six images total) serve as the input to our neural network. It is to be noted that all images that were trained and tested on were of dimensions 1500x1000x3 and of the .tif image format but since the neural network is fully convolutional, the network can accept images of any width and height.

3.2. HDR Generation Algorithm

The HDR generation method used in Kalantari, et al. was adopted for our purposes. This HDR algorithm was used to create the ground truth HDR image from the three static images to train the CNN on and to create the estimated HDR image. Thus, we processed the static images to create a ground truth HDR image for each scene.

The HDR algorithm calculates the (alpha) weights for the three bracketed images being inputted to the algorithm according to the following "triangle" transfer functions:



$$\alpha_1 = 1 - \Lambda_1(I_2), \quad \alpha_2 = \Lambda_2(I_2), \quad \alpha_3 = 1 - \Lambda_3(I_2)$$

It is to be noted that all the alpha weights are calculated with respect to the middle-exposed image and of the same dimensions as the input image (i.e. 1500x1000x3).

Although α_1, α_2, and α_3 were all calculated from the middle-exposed image, α_1, α_2, and α_3 are applied to the under-, middle-, and overexposed gamma corrected HDR images respectively by the following weighting function:

$$\hat{H}(p) = \frac{\sum_{j=1}^{3} \alpha_j(p) H_j(p)}{\sum_{j=1}^{3} \alpha_j(p)}, \quad \text{where} \quad H_j(p) = \frac{I_j^{\gamma}}{t_j}$$

This HDR generation algorithm was used to generate the ground truth HDR image for training from the three static images and to calculate the estimated HDR image form the alpha weights outputted by the CNN.

3.3. Transfer Learned Convolutional Neural Network

The images we collected and processed per sections 3.1 and 3.2 were used to retrain the last layer of the Kalantari, et al. network. A four layer convolutional neural network, as seen in Figure 3, was used with decreasing filter sizes from 7 to 1 across the convolutional layers (7, 5, 3, 1). Each convolutional layer was followed by a rectified linear unit (ReLU), with the final layer being followed by a sigmoid activation to ensure the network outputs values [0, 1] for the alpha weights. Further, all layers have a stride of 1 and no downsampling or upsampling was performed. There are nine output channels to the CNN corresponding to the α_1, α_2, and α_3 weights each of which have the dimensions of the three-channel input image.

The Kalantari, et al. network was trained to the distribution of the Canon EOS-5D Mark III camera and not that of the Google Pixel 2. Resultantly, we leveraged transfer learning to update the network to the distribution of the Google Pixel 2.

All the weights and biases of the four layer neural network were initialized to the values of the Canon trained network. Then, layers one two and three were frozen during

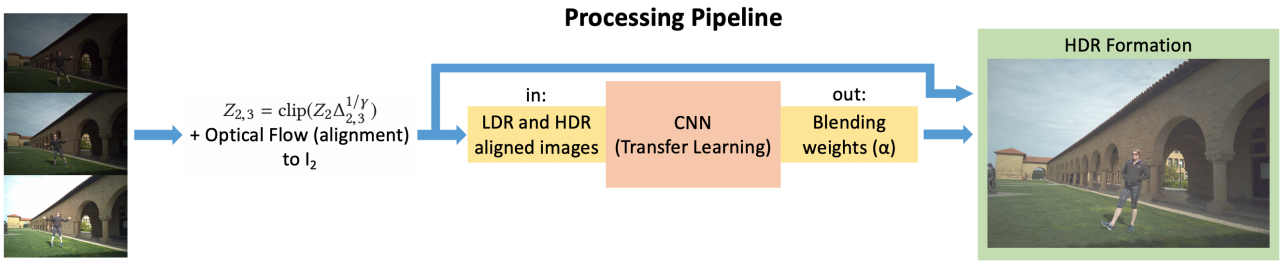


Figure 4: The time processing pipeline.

training (the weights and biases were held constant). The final (fourth) layer of the CNN, although initialized to the values of the Canon network was trained so that the network would be of the distribution of the Google Pixel 2, resulting in more domain specific results.

Since only the last layer was unfrozen the forward propagation through the network happened across all layers but backpropagation only occurred through the last layer. The idea behind exploiting transfer learning is that the low-level features, learned in the beginning layers of the CNN would be the same between the Canon EOS-5D Mark III and Google Pixel 2 distributions. Also, since only one layer had to be back propagated, it ensured that we would be able to perform a reasonable amount of training on a MacBook Pro CPU.

Since the output of the network is the alpha weights, to backpropagate the loss, the alpha weights were sent though the weighting function to compute the estimated HDR image *H'(p)*. Then, since HDR images are usually displayed after tone mapping, the loss function was computed between the estimated and ground truth tone mapped images. Since gamma encoding

$$H^{\frac{1}{\gamma}}, \gamma > 1$$

is not differentiable around zero (which is required for backpropagation), instead tone mapping was performed using the μ-law range compressor:

$$T = \frac{\log(1 + \mu H)}{\log(1 + \mu)}, \mu = 5{,}000$$

Then, the Euclidean distance between the estimated and ground truth tone mapped images is computed

$$E = \sum_{k=1}^{3} (\hat{T}_k - T_k)^2$$

as the loss and then backpropagated though the equation:

$$\frac{\partial E}{\partial w} = \frac{\partial E}{\partial \hat{T}} \frac{\partial \hat{T}}{\partial \hat{H}} \frac{\partial \hat{H}}{\partial \boldsymbol{\alpha}} \frac{\partial \boldsymbol{\alpha}}{\partial w}$$

using the ADAM optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ with a learning rate of 0.0001.

Each training scene was augmented tenfold and processed into 40x40x3 patches with a stride of 20. The test scenes were not divided into patches nor augmented.

4. Results

Due to disk space limitations, we were constrained to a training set of 10 images. The final layer of the transfer learned network was trained on a MacBook Pro CPU for 31,000 iterations. Figure 5 shows that the PSNR increased across all iterations as expected, in accordance with our loss function. It can be seen that the PSNR converged but to a level of only 6.47 indicating that most likely more layers would have to be retrained to resulting in higher development set PSNR values.

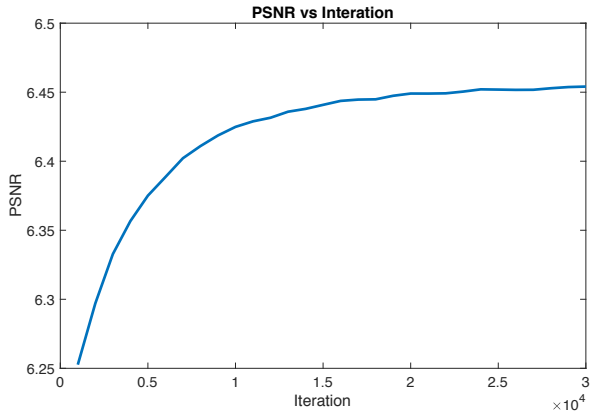


Figure 5: Peak signal to noise ratio by iteration during training on the development set.

Figure 6 shows the results of the Canon EOS-5D Mark III trained network and Google Pixel 2 transfer learned network on the Google Pixel 2 training, development, and test data. It can be seen that the transfer learned network was able to successfully create an HDR tone mapped image from the three bracketed images, similar to that of the Canon trained network. Having said that, significant motion artifacts remained in the transfer learned outputs even though the network was properly optimizing for the loss suggested function between the tone mapped expected and ground truth images.

Figure 6: The results and PSNR values of the Canon EOS-5D Mark III trained network and Google Pixel 2 transfer learned network to the ground truth HDR image.

Figure 7: The SSIM results and values of the Canon EOS-5D Mark III trained network and Google Pixel 2 transfer learned network to the ground truth HDR image.

5. Analysis

The results of the transfer learned network show that the network was able to capture high dynamic range of the scene, but unable to successfully remove the motion artifacts that arise in dynamic scenes. Figure 6 further shows that for all test scenes, the Google Pixel 2 transfer learned network was able to interestingly result in a higher PSNR than the Canon network. This is most likely due the differences in distributions of the Canon EOS-5D Mark III and Google Pixel 2, showing the need for the Kalantari et al. network to be retrained for mobile device distributions.

While the PSNR values were higher for the transfer learned network, qualitatively, the Canon network resulted very successful motion artifact removal and thus resulted in better HDR image reconstruction for dynamic scenes. It can thus be deduced that the motion artifacts were not effectively optimized for by solely retraining the last layer of the CNN with the current loss function as although the PSNR increased, the motion artifacts were not eliminated.

Although the PSNR was not able to show a strong correlation to artifact removal, calculating structural similarity index (SSIM) was able to accurately measure the presence or absence of motion artifacts. Figure 7 shows that for all test scenes, the Canon network was able to result in higher SSIM values compared to the transfer learned network. Further, the SSIM plots in Figure 7 show that the least structural similarities were in fact around the dynamic areas of the scenes. Since the SSIM index is a proper metric of motion artifact prevalence, it should be optimized for in the loss function during training.

5.1.1 Comparison to Previous Work

The previous work by Kalantari et al. was able to remove a significant amount of motion artefacts using a larger captured dataset and running training on all the network layers for two days on a GeFource GTX 1080 GPU for 2,000,000 iterations. We were able to capture high dynamic range and recover detail in bright and dark areas, but not able to replicate their motion artifact elimination given compute and memory limitations. With additional resources, however, we believe we can achieve similar results with our modifications.

Our work did well in resolving detail throughout a high dynamic range image, which is a challenge for the HDR+ implementation already available through the Google Pixel 2 due to the fact that HDR+ captures three images at the same exposure rather than at different exposures, so it can lose detail in "blown out" regions. The HDR+ algorithm, however, is very straightforward and non-learned, so it requires very little processing power on a mobile GPU.

5.2. Discussion

The transfer learned algorithms was able to resolve high dynamic range scenes similar to those of state-of-the-art tactics but suffered from similar motion artifacts in dynamic scenes. We have high confidence in the ability of this method to overcome those issues given sufficient compute.

5.2.1 Limitations

Applying this work to more data requires input of dynamic exposure-bracketed images captured on a specific platform. This is not readily available. While there are many exposure-bracketed HDR datasets, few contain motion, which produces significant problems in most HDR techniques.

Additionally, the techniques here discussed cannot resolve detail that is not captured in any of the input images. As you can see in the test images, if the input is blown out, then so is the output. This is likewise true for dark environments. We are unable to reconstruct data we don't have or compensate for poorly chosen exposures.

5.3. Future Work

5.3.1 Full Retraining with Additional Compute

The largest issue we faced with this project was a lack of compute resources. Given access to a GPU we could still initialize all layers of the network to the Canon parameters but re-train all layers, which was infeasible to do with a CPU as backpropagation is very computationally expensive. Likewise, with increased disk space we would be able to train on our entire captured training set. It is believed retraining all layers of the CNN with a GPU would improve on removing the motion artifacts still prevalent due to re-training only last layer.

5.3.2 SSIM + PSNR Bi-Objective loss function

The analysis of the SSIM index test results show that SSIM is a great metric of measuring motion artifacts. Resultantly, it is proposed that maximizing SSIM between the tone mapped estimated and ground truth images should be included in the loss function in addition to the Euclidean distances between the two images. The relative weighting of the two objective functions can also be optimized for.

5.3.3 Improving Exposure Bracketing

The technology that we used to capture dynamic exposure-bracketed scenes was difficult to adjust on a scene-to-scene basis, which resulted in some of our dataset scenes being over- or underexposed for all of the exposures. Applying similar HDR techniques to images captured using an automatic exposure for the center image would allow for better quality input data and therefore a better output and more uniformly similar to the best of our results.

6. Conclusion

By performing transfer learning on the Kalantari et.al. network, we were able to successfully fuse three different exposures into a high dynamic range scene for the distribution of the Google Pixel 2. Although our results were of a higher PSNR compared to the Canon EOS-5D Mark III trained network, the Google Pixel 2 transfer learned network was unable to successfully remove all

motion artifacts that were effectively quantified using the SSIM index. This suggests that SSIM should be included in the loss function as future work for this project with access to a more powerful processor and with more memory. With more time and resources for training, and with a more tightly customized dataset, transfer learning would be an effective solution for optimizing the Kalantari et.al. network for high dynamic range imaging of dynamic scenes for mobile imaging.

References

[1] C. Liu. Beyond Pixels: Exploring New Representations and Applications for Motion Analysis**.** *Doctoral Thesis*. 2009. Massachusetts Institute of Technology.

[2] Z. Chen, H. Jin, Z. Lin, S. Cohen, and Y. Wu, ‘‘Large displacement optical flow from nearest neighbor fields,’’ in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2013, pp. 2443–2450

[3] Banterle, F., Artusi, A., Debattista, K. and Chalmers, A. (2017). Advanced high dynamic range imaging. 2nd ed. Natick, MA, USA: AK Peters (CRC Press).

[4] N. K. Kalantari and R. Ramamoorthi, “Deep High Dynamic Range Imaging of Dynamic Scenes,” ACM Transactions on Graphics (Proceedings of SIGGRAPH 2017), vol. 36, no. 4, 2017.

[5] S.W. Hasinoff, D. Sharlet, R. Geiss, A. Adams, J.T. Barron, F. Kainz, J. Chen, and M. Levoy, “Burst photography for high dynamic range and low-light imaging on mobile cameras,” ACM Transactions on Graphics (Proc. SIGGRAPH Asia), vol. 35, no. 6, 2016.

[6] R. C. Bilcu, A. Burian, A. Knuutila, and M. Vehvil, “High dynamic range imaging on mobile devices,” 2008 15th IEEE International Conference on Electronics, Circuits and Systems, 2008.

[7] Authors. The frobnicatable foo filter, 2006. ECCV06 submission ID 324. Supplied as additional material eccv06.pdf.

[8] Authors. Frobnication tutorial, 2006. Supplied as additional material tr.pdf.