Single Exposure Fusion

Jiwon Chang
(Dated: December 14, 2021)

I. INTRODUCTION

A. HDR and Tone Mapping

Photographs often contain oversaturated or undersaturated areas that make it difficult to discern details. Photographers utilize various techniques to extend the perceptual dynamic range and ensure that details are visible in all areas of an image. If high dynamic range (HDR) cameras are available, tone-mapping an HDR image to standard dynamic range (SDR) is the desired method. If not, then photographers can use exposure fusion to combine several SDR photographs taken at different exposures [1]. However, exposure fusion produces motion blur from moving subjects and camera (Figure 1) [16].

A solution to the motion blur is to imitate an HDR-like look from a single exposure. Various approaches have been explored. Yun et al. uses a multiple-transformation based approach [16]. Several others train a neural network using a dataset of corresponding SDR and HDR images [11, 17]. Photographers often use Photoshop layer masks to fake the HDR look [6].

B. Single Exposure Fusion

This project explores an alternative approach that aims to produce a set of three images from a single exposure. Exposure fusion combines them into an HDR-like image. In other words, I attempt to restore what the scene would have looked like from a different exposure given a single photograph.

Let x_0 be an image taken at some medium exposure, and x_i be the image of the same scene taken i stops higher than x_0. Conventional exposure fusion, such as the Mertens' method [9], is a function

$$f_{\text{fusion}} : (x_{-k}, x_0, x_k) \to y$$

for some k, where y is the HDR-like image.[10] Machine learning methods are functions of the form $x_0 \to y$.

Single exposure fusion is a two-step process, where an image restoration function

$$f_{\text{restore}} : (x, i) \to x_i^*$$

is applied to obtain (x_{-k}^*, x_0, x_k^*) where $*$ denotes approximation. Then, we apply f_{fusion} to (x_{-k}^*, x_0, x_k^*) which gives us y^*.

Increasing the brightness, lowering the contrast, and increasing the saturation of the original image mimics a longer-exposure photograph. Similarly, the inverse process mimics a shorter-exposure photograph. This naive editing process causes noise and color distortion. I propose applying an image restoration algorithm to restore a natural-looking image. I compare the results of various existing resoration algorithms.

II. THEORY

I chose the following methods for comparison.

- Ground truth: Three multiple-exposure photographs, obtained from [1], combined with the Mertens' method [9].
- Naive edit: A medium-exposure image with its brightness, contrast, and saturation edited to mimic longer and shorter exposures.
- Gaussian filter: A simple local denoising algorithm.
- BM3D: State-of-the-art non-local denoising algorithm [4].
- Noise2Noise: ML-based denoising algorithm for gaussian and poisson noise [7].
- Deep image prior: General-purpose, no-training, ML algorithm [15].

This section provides a high-level theoretical overview of each method.

A. Exposure Fusion

Exposure fusion, first proposed by Mertens et al., combines several SDR images of the same scene taken at different exposures to produce a visually pleasing result [9]. It provided two main innovations. Not only does it skip the step of mapping the SDR images to an HDR color space, it also doesn't require the stop values of the images. The latter is crucial for single exposure fusion, since we only have one photograph. This limitation excludes other approaches such as Reinhard or Durand's methods [5, 12].

Mertens' method first calculates weights of how "important" a pixel is. The weight of pixel (i, j) for the k^{th} image is

$$W_{i,j,k} = (C_{i,j,k})^{\omega_C} \times (S_{i,j,k})^{\omega_S} \times (E_{i,j,k})^{\omega_E}$$

where C, S, E are the contrast, saturation, and well-exposedness of the pixel, and $\omega_C, \omega_S, \omega_E$ exponents are relative weights of each property.

Figure 1. HDR-like image produced with Mertens' method of exposure fusion. Without image alignment, motion blur is present.

1. Pixels with high contrast are more likely to contain details. Contrast is computed by calculating the second derivative of the image, a technique often used to detect edges.
2. Pixels with high saturation look more vivid. Saturation is computed as the standard deviation of the R, G, and B channels. Intuitively, gray pixels have the same value for each channel.
3. Selecting for pixels that are in the middle of brightness range helps avoid over-saturated or under-saturated pixels. A Gaussian curve weighs the well-exposed pixels the highest.

After all weights are calculated, they are normalized such that $\sum_k W_{i,j,k} = 1$. Finally, weighted blending combines all pixels into one.

B. Denoising

Local denoising methods take advantage of the property that a pixel value is most likely similar to its neighbors. A simple box filter averages the pixel values in an $n \times n$ kernel. Gaussian blur weighs closer pixels higher than those that are further away by using a Gaussian filter instead. Box filter and Gaussian filter, however, also tend to blur edges. Bilateral filter reduces noise in an edge-aware manner by incorporating the perceptual difference between two pixels' colors into the weights [13].

Non-local denoising methods utilize the self-similarity present in most natural images to improve denoising performance. The non-local means method [3] divides the image into numerous pixel neighborhoods, then weighs statistically similar neighborhoods higher. Box matching 3D (BM3D) is a state-of-the-art extension of non-local-means algorithm [4].

C. ML-Based Denoising

Machine learning denoising algorithms are generally trained on a dataset of pairs of noisy and clean images. The learner's goal is to minimize empirical risk which is typically measured by mean squared error (MSE), peak signal-to-noise ratio (PSNR), or structural similarity index (SSIM).

Let us define a learner in the hypothesis space H with some loss function L. It is given a training set $S = ((x_1, y_1) \ldots, (x_n, y_n))$, an ordered list of pairs of SDR and HDR images. Then the learner chooses the best hypothesis h_S s.t.

$$h_S = \underset{h \in H}{\text{argmin}} L_S(h)$$

I chose Noise2Noise as a representative ML-based denoising algorithm with a competitive performance [7].

D. Deep Prior

An ideal image restoration algorithm that can faithfully approximate x_k must do more than remove noise. It also needs to correct "unnatu-

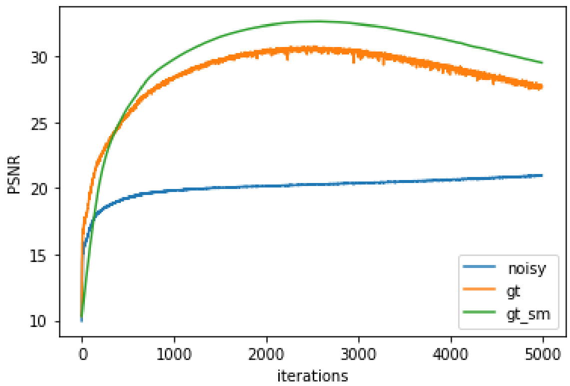

Figure 2. PSNR of a randomly initialized deep convolutional generative adversarial network over iterations. PSNR is calculated with respect to the ground truth (orange) and noisy image (blue). The network best approximates the ground truth at around 2500 iterations, then starts overfitting to the noise.

ral" coloration produced by naive editing method. I chose Deep prior as a general-purpose image restoration method that may be able to achieve such a goal.

Deep Prior utilizes a randomly initialized generative adversarial network (GAN) which is tasked to approximate a degraded image [15]. Its output approaches, by gradient descent, the degraded image and eventually overfits to the noisy details. Remarkably, if the training is stopped midway through, the outcome more closely approximates the ground truth (clean image) than the degraded image (Figure 2).

The restoration is due to an inherent inductive bias in deep convolutional neural networks to exploit self-similarity. It is therefore able to learn semantically meaningful features of an image faster than it learns random noise or degradation. Deep prior performs well for a wide range of restoration tasks such as denoising, deblurring, and inpainting.

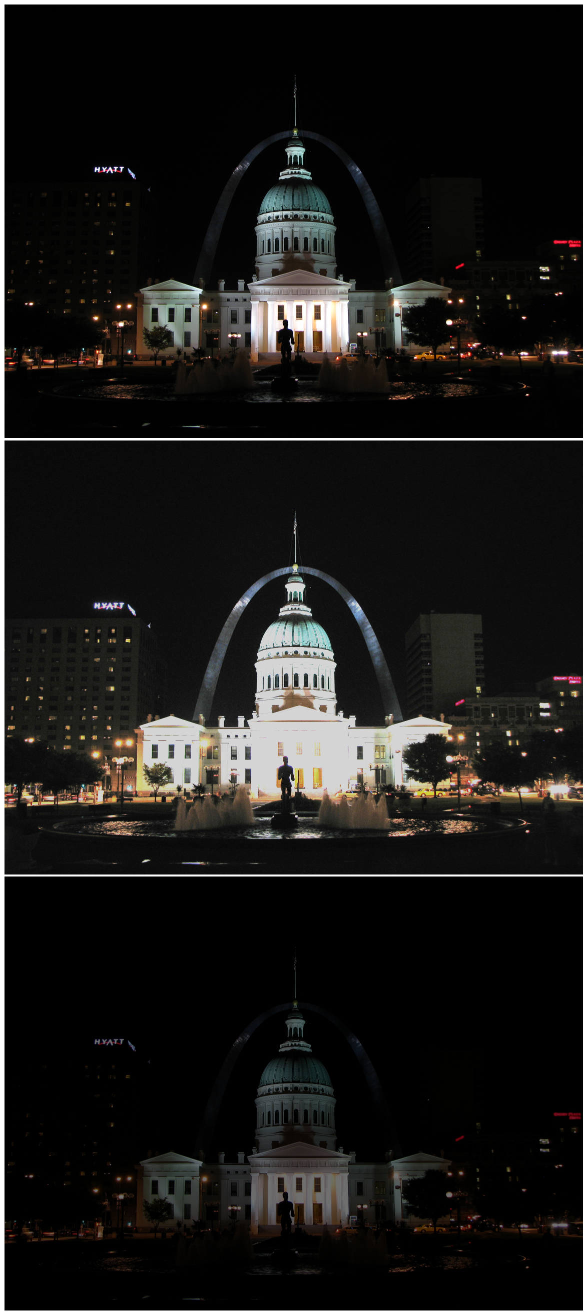

Figure 3. Medium-brightness image (top) was edited using simple brightness, contrast, and saturation adjustments to mimic the look of longer (middle) and shorter (bottom) exposures.

III. METHODS

I obtained a set of three images of the same scene at 0, -2, and +2 stops from [1]. Then, I used OpenCV's implementation of the Mertens' method to combine the three images to serve as ground truth (Figure 1).

I then edited the middle-exposure image x_0 using Python Imaging Library's ImageEnhance module by eye to mimic some x_k and x_{-k} (Figure 3).

I used the OpenCV implementation of Gaussian blur and the bm3d PyPI package [8]. I chose a Gaussian kernel size of 15×15 and BM3D $\sigma = 15$ by experimentation to produce subjectively pleasing results.

I reproduced Noise2Noise by cloning the Github repository [2] in Google Colab and using the pre-trained .pickle files for Gaussian and Poisson noise.

I modified Ulyanov et al.'s Github repository [14]. Using their denoising Jupyter notebook as the baseline, I implemented features to save output as file every 500 iterations and debugged it to run on Paperspace Gradient and on modern Python. I halved the naively edited images' dimensions such that the device doesn't run out of memory. The approximation process is shown in Figure 4.

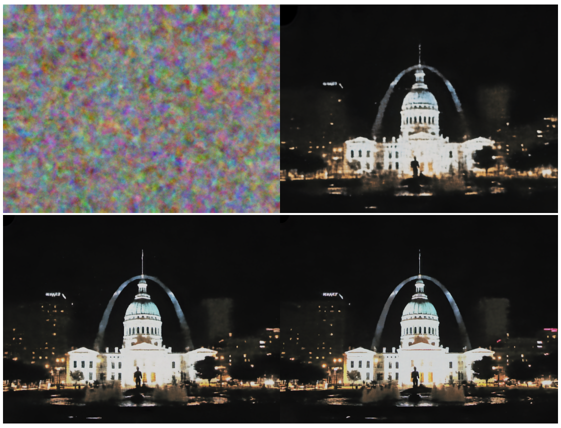

Figure 4. Top left to bottom right: Deep Prior learns to restore the naively-edited long-exposure image over 1500 iterations.

Finally, I applied the Mertens' method to the denoised images to produce HDR-like images.

IV. RESULTS

Figure 5 shows the resultant HDR-like photographs produced by the six methods outlined in Section II. As aforementioned, the naive edit doesn't necessarily approximate x_2 and x_{-2}, but rather some arbitrarily longer or shorter exposures. As such, it is rather difficult to objectively measure their performance. Since no-reference image quality assessment is still in its infancy, I instead rely on subjective quality assessment.

All single exposure fusion methods successfully extend the perceptual dynamic range compared to the medium-brightness image in Figure 3. In particular, the under-exposed buildings in the background are now visible. Furthermore, all methods contain no motion blur.

The naive editing process, however, produces noise that degrades perceptual visual quality. The various denoising algorithms vary in performance. Subjectively, BM3D does the best job of noise removal. Deep Prior, on the other hand, produce an image that has a paint-like texture.

Assuming that the well-exposed areas of the original image is in focus, and that the photographer desires the background to be blurred out, Deep Prior's result provides a suitable aesthetic quality.

The ground truth is the only image that portrays the background buildings in a warm color. None of the restoration algorithms are able to deduce that, since the background buildings are lit by warm lights, the longer-exposure photograph should have a yellow tint.

V. DISCUSSION

Ultimately, I am reporting a negative result. Single exposure fusion does not outperform other single-exposure, non-ML methods such as [16]. The most promising result, Deep Prior, takes up to two hours to produce, which is not feasible in practice. Furthermore, only ML-based approach such as [11, 17] can infer characteristics of the scene, such as the warm color of the high-exposure image in the chosen scene.

I also encountered several issues using the ML-based restoration algorithms. Poisson Noise2Noise produced color aberrations and was discarded. Gaussian Noise2Noise performed more favorably. This is unexpected, since the noise on a dark image should be dominated by poisson-distributed photon shot noise. Although Deep Prior successfully restored the long-exposure image, it slowed down considerably for short-exposure image for an unknown reason. Since it was only able to train 500 iterations after many hours, I had to discard it.

This project does explore an interesting image processing problem. Prior ML-based approaches have focused on $x_0 \to y$. That is, they produce an HDR-like image given a medium-exposure image. However, photographers may also often desire to approximate a different exposure. For instance, one may only have access an overexposed image x_2 and wish to recover x_0. Since zero-training approaches have been shown to be lackluster in this project, ML-based approach should be explored more.

[1] High-dynamic-range imaging. *Wikipedia* (Nov. 2021). https://en.wikipedia.org/w/index.php?title=High-dynamic-range_imaging&oldid=1056640545.

[2] Noise2Noise: Learning Image Restoration without Clean Data - Official TensorFlow implementation of the ICML 2018 paper. NVIDIA Research Projects, Dec. 2021. https://github.com/NVlabs/noise2noise.

[3] Buades, A., Coll, B., and Morel, J.-M. A Non-Local Algorithm for Image Denoising. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)* (San Diego, CA, USA, 2005), vol. 2, IEEE, pp. 60–65.

[4] Dabov, K., Foi, A., Katkovnik, V., and Egiazarian, K. Color Image Denoising via Sparse 3D Collaborative Filtering with Grouping Constraint in Luminance-Chrominance Space. In *2007 IEEE International Conference on Image Processing* (San Antonio, TX, USA, Sept. 2007), IEEE, pp. I – 313–I – 316.

[5] Frédo Durand, and Julie Dorsey. Fast bilateral filtering for the display of high-dynamic-range images — Proceedings of the 29th annual conference on Com-

Figure 5. Comparison of the six chosen techniques. Top left: ground truth. Top right: Naive edit. Middle left: Gaussian. Middle right: BM3D. Bottom left: Noise2Noise. Bottom right: Deep Prior. Although all approximation techniques increase perceptual dynamic range, only the ground truth correctly represents the warm-colored scene. BM3D and Deep Prior are best able to reduce noise in the background buildings.

puter graphics and interactive techniques. https://dl.acm.org/doi/abs/10.1145/566570.566574, 2002.

[6] Jimmy McIntyre. Photoshop Secrets 21: Create a Fake HDR Effect, Apr. 2016.

[7] Lehtinen, J., Munkberg, J., Hasselgren, J., Laine, S., Karras, T., Aittala, M., and Aila, T. Noise2Noise: Learning Image Restoration without Clean Data. *arXiv:1803.04189 [cs, stat]* (Oct. 2018).

[8] Mäkinen, Y. Bm3d: BM3D for correlated noise. https://pypi.org/project/bm3d/.

[9] Mertens, T., Kautz, J., and Van Reeth, F. Exposure Fusion: A Simple and Practical Alternative to High Dynamic Range Photography. *Computer Graphics Forum 28*, 1 (Mar. 2009), 161–171.

[10] Strictly speaking, the Mertens' method generalizes to any number of images.

[11] Raipurkar, P., Pal, R., and Raman, S. HDR-cGAN: Single LDR to HDR Image Translation using Conditional GAN. *arXiv:2110.01660 [cs, eess]* (Oct. 2021).

[12] Reinhard, E., Heidrich, W., Debevec, P., Pattanaik, S., Ward, G., and Myszkowski, K. *High Dynamic Range Imaging: Acquisition, Display, and Image-Based Lighting*. Morgan Kaufmann, May 2010.

[13] Tomasi, C., and Manduchi, R. Bilateral filtering for gray and color images. In *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)* (Jan. 1998), pp. 839–846.

[14] Ulyanov, D. Deep image prior, Dec. 2021. https://github.com/DmitryUlyanov/deep-image-prior.

[15] Ulyanov, D., Vedaldi, A., and Lempitsky, V. Deep image prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2018), pp. 9446–9454.

[16] Yun, S.-H., Kim, T.-C., and Kim, J. H. Single exposure-based image fusion using multi-transformation. In *The 1st IEEE Global Conference on Consumer Electronics 2012* (Oct. 2012), pp. 142–143.

[17] Zhen, G., Chen, L., Xu, S., and Ding, D. Reconstructing HDR image of high color fidelity using Generative Adversarial Networks. In *Twelfth International Conference on Graphics and Image Processing (ICGIP 2020)* (Xi'an, China, Jan. 2021), Z. Pan and X. Hei, Eds., SPIE, p. 53.