Exposure Fusion

Souhaiel Ben Salem*
MVA - Department of Mathematics
ENS Paris Saclay
souhaiel.ben_salem@ens-paris-saclay.fr

Abstract

In this project, we present the implementation and investigation of a technique for fusing a bracketed exposure sequence into a high-quality image, without the need for physically-based HDR assembly. This technique, called $ExposureFusion$, was introduced by T. Mertens, J. Kautz, F in (4). Their approach simplifies the image acquisition process by skipping the HDR assembly step, and is computationally efficient and versatile enough to include flash images in the sequence. The technique blends multiple exposures in a multi-resolution fashion, guided by quality measures such as saturation and contrast. The resulting image quality is comparable to existing tone mapping operators and demonstrates the potential of our proposed method.

1 Introduction

Real-life scenes often have a wider dynamic range compared to what camera sensors can capture. To preserve this range, photographers take multiple photos at different exposure times. A long exposure captures details in dark areas while overexposing the brighter parts, while a short exposure captures detail in bright areas. This sequence of photos is called a bracketed exposure. To create a high dynamic range (HDR) image, these photos must be combined. However, HDR images have more bits of information than can be displayed on typical screens, so they must be reduced in range through a process known as tone-mapping. This adjusts the colors to fit within the limited range of 8 bits.

In 2009, T. Mertens, J. Kautz, and F. Van Reeth introduced $ExposureFusion$ as a new approach to constructing a high dynamic range (HDR) image from a bracketed exposure sequence. Unlike other methods that first compute an intermediate HDR image, Exposure Fusion directly selects the best pixel values from the available photos to create a final image that blends the best features from each of the input images. This approach is based solely on straightforward quality criteria, such as saturation and contrast, which have proven to be highly effective. Additionally, the computations are proven to be performed almost in real-time due to the utilization of a pyramidal image decomposition. This technique answered important questions about identifying the best pixels and merging them seamlessly in the final image, making it a unique and effective solution compared to similar techniques that existed previously.

In this report we will be going through the algorithm of $ExposureFusion$ explaining it in detail and giving the intuition behind each used concept. Next, we will use our implementation of the algorithm and demonstrate its efficiency and superiority compared to other methods that the authors tried .

2 Related Work

HDR imaging combines multiple low dynamic range images taken with a normal camera to produce a single high dynamic range image. To linearize the intensities, the camera's specific response curve must be determined, which can be done by using the input sequence and exposure settings.
Displays typically have a restricted dynamic range and are unable to display HDR images as is. To overcome this limitation, tone mapping is employed to adapt the dynamic range of the image to the capabilities of the display device. This is done by compressing the dynamic range of the image to fit the display's limited dynamic range.

2.1 Global tone mapping operators

Global Tone Mapping Operators are used in computer graphics and image processing to reduce the dynamic range of an HDR image to a range that can be displayed on a typical computer screen or any other display device with a limited dynamic range. This is necessary because displays have a limited range of brightness and color levels, which are not capable of displaying the full range of brightness and color levels in an HDR image.
The main objective of a global TMO is to preserve as much of the original information in the HDR image as possible while still making it viewable on a display device. This is achieved by transforming the HDR image into a displayable range in a way that preserves important image features such as contrast, color, and details.
The tone mapping process is performed by mapping the high dynamic range of the HDR image to the limited dynamic range of the display device using a transfer function. This transfer function maps the high dynamic range intensities of the image to a new range of intensities that can be displayed on the target device. The function is designed to preserve important image features such as contrast and color while avoiding clipping and loss of detail in the highlights and shadows of the image. Their main advantage is speed, but sometimes fail to reproduce a pleasing image.

2.2 Local tone mapping operators

Local Tone Mapping Operators (LTMOs) (5) are techniques used in image processing to map the high dynamic range (HDR) values of an image to the lower dynamic range of a display device. Unlike global TMOs, which apply the same tone mapping function to all pixels in the image, local TMOs apply different tone mapping functions to different regions of the image.
The goal of local TMOs is to preserve the local image details, such as texture and contrast, while still reducing the dynamic range to fit the display device. This is achieved by dividing the image into small regions or windows, and applying a separate tone mapping function to each window. The tone mapping function is designed to preserve important image features such as contrast, color, and details within each region, while avoiding clipping and loss of information in the highlights and shadows.
Local TMOs are often preferred over global TMOs for high dynamic range images that have a large variation in brightness and color levels within the same image. By using a separate tone mapping function for each region, local TMOs can better preserve the local image details and avoid the loss of important image features.

3 Exposure Fusion

In this section we detail the main idea introduced in the paper as well as the theoretical details in play. We will start by introducing the quality measures that will guide the fusion process and then move on to explain the different fusion approaches that the authors tried before finally opting for the final, superior algorithm.

3.1 Quality Measures

For the exposure algorithm to work, the weighting of the images in the stack should prioritize interesting areas with bright colors and details, and reduce the weight of areas that are flat and lacking in color due to under- or overexposure. To this end, the authors propose to evaluate the quality of each pixel in the input images using three metrics: contrast (C), saturation (S), and well-exposedness (E). These metrics measure the perceptual quality of the pixels.

- **Contrast (C)**: A Laplacian filter is applied to the grayscale version of each image in the sequence in order to assess its contrast. We take the absolute value of the filter response (3) which constitutes a measure of contrast which tends to give a high weight to significant features like edges and texture.

$$C_{ij,k} =\mid \frac{1}{3}\sum_{c=1}^{3} I_{c,k} * K_L \mid (i,j)$$

  Where $I_{c,k}$ is the $k-th$ image of the $c-th$ channel in the sequence and K_L is the Laplacian kernel.

- **Saturation (S)**: Since saturated colors are considered desirable and make the image appear more vivid, the authors proposed to include a saturation measure S that is calculated at each pixel by computing the standard deviation within the Red, Green, and Blue channels.

$$S_{ij,k} = \sqrt{\frac{1}{3}\sum_{c=0}^{2}\Big(I_{c,k}(i,j) - \frac{1}{3}\sum_{c=0}^{2} I_{c,k}(i,j)\Big)^2}$$

- **Well-Exposedness (E)**: The well-exposedness of a pixel is determined by examining the raw intensities within each color channel. To preserve intensities that are not too low (underexposed) or too high (overexposed), the authors propose to use a Gaussian curve to weight each intensity based on its proximity to 0.5. The weight for each intensity pixel intensity i is computed as $exp(-\frac{(i_c-0.5)^2}{2\sigma^2})$, where σ is set to 0.2. The Gaussian curve is applied to each color channel separately, and the results are multiplied to give the well-exposedness measure E.

$$E = \prod_{c=1}^{C} exp(-\frac{(i_c-0.5)^2}{2\sigma^2})$$

The final assessment of each pixel's value is determined by combining these three metrics in convex combination. This combination is a power function that ensures that only pixels which meet all three quality standards are preserved and allows to have control over the influence of each measure. As such , the weight of a pixel at position (i,j) belonging to the $k-th$ image is computed as:

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

For consistency reasons, the resulting weights are normalized during the blending process,such that :

$$\hat{W}ij,k = [\sum_{k'=1}^{N} W_{ij,k'}]^{-1} W_{ij,k} \qquad (1)$$

3.2 Naive Fusion

As a first attempt, the authors tries a naive (the simplest) fusion process where the resulting image R can then be obtained by a weighted blending of the input images:

$$R_{ij} = \sum_{k=1}^{N} \hat{W}_{ij,k} I_{ij,k} \qquad (2)$$

To illustrate this first idea, we use a sequence of images of St.Louis' old courthouse taken at night using different exposure times. We use our $naive\_fusion$ function to plot the weight maps associate to each image in the input sequence and the final result.

Figure 1: The input sequence (Top) and the corresponding weight maps (bottom) computed using equation (1). Image credits : Wikepedia

This scene represents a challenging situation for HRD algorithms since it was taking at night and since there a lot of light sources to handle as opposed to day-light scenes.
Before examining the final result of this fusion method, we must establish that a high weight (brighter pixel) means that a pixel should appear in the final image. These weights reflect desired image qualities, such as high contrast and saturation. We can also notice the presence of noise (seams) in the weight maps especially in the sky which is the darkest region in the input sequence.
The final result of this fusion process is as follows:

Figure 2: The output of our naive fusion method

The result we get after naive fusion is unsatisfactory result. This is because wherever weights vary quickly, disturbing seams will appear as a result of the fact that the images we are combining contain different absolute intensities due to their different exposure times.

3.3 Gaussian smoothing of the weigh maps

In a first to solve the problem of seams induced by sharp sharp weight map transitions, the authors proposed to smooth the weight maps using a Gaussian filter. In our implementation, we used a Gaussian blur with a kernel size of R and a standard deviation of 5.
The following figure shows the fusion result obtained after Gaussian smoothing of the weight maps:

Figure 3: The output of our Gaussian smoothing fusion method

The first thing we notice is that the seams have been eliminated, or rather have been smeared to a certain degree. However, the result is still somewhat cartoonish. Besides, as mentioned by the authors, we can still notice a ghosting effect (halos) around edges, and spills of information across object boundaries.

Figure 4: The output of our Gaussian smoothing fusion method

3.4 Cross-Bilateral smoothing of the weigh maps

In an attempt to mitigate the undesirable halos and improve the overall quality of the resulting image, the authors proposed to use an edge-aware smoothing operation using the cross-bilateral filter (2) which seems like a better alternative.
A cross bilateral filter is a variant of the bilateral filter, which is used for edge-preserving smoothing in image processing. Unlike the traditional bilateral filter, the cross bilateral filter uses two different functions for smoothing the intensity values and the gradient information. This can result in improved performance in terms of preserving edge information and reducing noise compared to traditional bilateral filters.

The cross bilateral filter is a better choice than a Gaussian filter for HDR exposure fusion because it provides better edge preservation. It employs two functions, an intensity-based smoothing function and a gradient-based smoothing function that are defined in terms of spatial and intensity domains.

The weight maps computed by the cross-bilater filter are as follows:

$$W_k(p) = \sum_{q \in_p} G_{\sigma_s}(\|q-p\|) G_{\sigma_r}(|I(q)-I(p)|) W_k(q)$$

Where S_p is the spatial neighborhood of pixel p, $G_{\sigma_s}(\|q-p\|)$ is the spatial smoothing function with a standard deviation of σ_s and $G_{\sigma_r}(|I(q)-I(p)|)$ is the intensity smoothing function with a standard deviation of σ_r.

The following figure shows the fusion result obtained after the smoothing of the weight maps using the cross-bilateral filter:

Figure 5: The output of our cross-bilateral-based fusion method

We can notice that uing a cross-bilateral filter instead of a Gaussian one definitely helps mitigate the ghosting effect and the spilling of information (the ghosting effect noticeable on the statue is caused by a slight misalignment between the fused images) . However, the end result is still not satisfactory.

Also, as stated by the authors, it is unclear how to determine the control image to determine where to end the smoothing process. Additionally, finding suitable parameters for the cross-bilateral filter to control spatial and intensity influence is challenging. We had to try a lot of different parameters (spatial and intensity standard deviations and patch size) to finally find an acceptable result.

3.5 Fusion using Laplacian Pyramids

In order to resolve the issue of seams, the authors propose to employ a technique that has been inspired by the work of Burt and Adelson (1).
The methodology of constructing the Gaussian Pyramid of images I as described in (1)involves a hierarchical process of successive reduction of the dimensions of the image through repeated downsampling by a factor of two, until the size of the image has been reduced to a single pixel, which is referred to as the maximum level of coarseness ($lmax$). The Laplacian Pyramid, on the other hand, is formulated as the resultant difference between two consecutive levels (l and $l+1$) of the Gaussian Pyramid, with the second level being augmented through upsampling by a factor of two. The final scale of the Laplacian Pyramid is designated as the residual, and is equivalent to the coarsest scale of the Gaussian Pyramid.
In our specific scenario, we have N distinct images and N corresponding normalized weight maps that serve as alpha masks. The $l-th$ level in a Laplacian Pyramid decomposition of an image A is represented by $L\{A\}^l$, while the corresponding $l-th$ level in the Gaussian Pyramid decomposition of image B is represented by $G\{B\}^l$. Subsequently, the process of blending is executed on the coefficients, which in this context refer to the pixel intensities at various levels within the pyramid as follows:

$$L\{R\}^l_{ij} = \sum_{k=1}^{N} G\{\hat{W}\}^l_{ij,k} L\{I\}^l_{ij,k}$$

Having discussed the theoretical foundations, let us now proceed to present the full algorithm outlining the step-by-step procedure of the entire process of exposure fusion, from its initiation to its conclusion.

Algorithm 1 Exposure Fusion

Input: Input sequence of Bracketed images I taken at different exposures
Input: Weights ω_s, ω_c, ω_e (quality measures)
Output: Fused image with improved HDR R

for each image located at the $k-th$ position within the input sequence of size N **do**
    Compute contrast measure C , saturation S and well-exposedness E
    Compute weights $W_{ij,k}$ of the $k-th$ image
    Normalize weights using
end for
Initialize output pyramid $L\{R\}$
for each image I located at the $k-th$ position within the input sequence **do**
    Compute Gaussian pyramid of weights $G\{\hat{W}\}_k$
    Compute Laplacian pyramid of input image $L\{I\}_k$
    for each coefficient at position (i, j) and scale l **do**
        Update Laplacian pyramid of the output image: $L\{R\}^l_{ij} \leftarrow \sum_{k=1}^{N} G\{\hat{W}\}^l_{ij,k} L\{I\}^l_{ij,k}$
    end for
end for
$R \leftarrow$ collapse Laplacian pyramid $L\{R\}$

The following figure shows the fusion result obtained using our final exposure fusion algorithm:

Figure 6: The output of our Laplacian-Pyramid-based fusion method

As show in our latest result, the technique of Multiresolution Blending has been proven to effectively prevent seam formation . This is because it blends image features rather than intensities. The blending equation is computed at each scale individually, allowing sharp transitions in the weight map to only impact sharp transitions in the original images (such as edges). However, flat regions in the original images will not be impacted by variations in the weight function, even if the absolute intensities among the inputs may differ.
More resultus are on the appendix.

3.6 Computational Complexity:

Our algorithm performs reasonably well especially for short input sequences of 3 or 4 images given it is implemented in python which notoriously slow compared to compiled languages like C_{++}. The following table shows the computation time for some of the examples presented in the appendix.

$w \times h \times N$	total time (s)
$580 \times 870 \times 3$	0.384
$558 \times 870 \times 3$	0.405
$960 \times 1280 \times 4$	1.45
$653 \times 870 \times 4$	0.591
$580 \times 870 \times 5$	0.624
$535 \times 870 \times 11$	1.240
$960 \times 1280 \times 13$	4.505

4 Discussion

n conclusion, the paper "Exposure Fusion" by Tom et al.(4) presents a simple technique for performing HDR image fusion. The technique is based on a multi-scale image blending approach, which blends differently exposed photos into one image, while preserving image details and dynamic range.
The final algorithm performs reasonably well and yield an image with a quality that is comparable to existing tone mapping operators, while eliminating all the problem we faced with the first three approaches. Additionally, our algorithm is computationally efficient as it does not require any physical-based calibration step. However if, there is still room for improvement as we can see from the comparison of our result with the locally mapped image (see Figure 7 below). This improvement can be achieved by exploring more the parameters of our algorithm and refining our quality measures.

Figure 7: local tone mapped image (left) vs our result (right)

It is important however to keep in mind that the results of our algorithm depend heavily on the input sequence. Hence, another way to improve the algorithm is to incorporate additional information as current technique relies only on the exposure levels of the input images. Incorporating additional information (camera noise for instance) means designing a better quality measure function that could potentially improve the quality of the resulting HDR image

References

[1] P. Burt and E. Adelson. The laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, 31(4):532–540, 1983.

[2] F. Durand and J. Dorsey. Fast bilateral filtering for the display of high-dynamic-range images. In *ACM Trans. Graph*, page 21(3):257–266, 2002.

[3] Jitendra Malik and Pietro Perona. Preattentive texture discrimination with early vision mechanisms. *J. Opt. Soc. Am. A*, 7(5):923–932, May 1990.

[4] Tom Mertens, Jan Kautz, and Frank Van Reeth. Exposure fusion. In *15th Pacific Conference on Computer Graphics and Applications (PG'07)*, pages 382–390, 2007.

[5] Erik Reinhard, Greg Ward, Sumanta Pattanaik, and Paul Debevec. *High dynamic range imaging : acquisition, display, and image-based lighting*. 01 2006.

A More Results of the final Blending algorithm

All image sequences are taken from either EasyHDR or Wikipedia.

Blending using a sequence of 3 images:

Figure 8: The input sequence (three first images) and the result of the multi-resolution exposure fusion (left).

Blending using a sequence of 4 images:

Figure 9: The input sequence (top) and the result of the multi-resolution exposure fusion (bottom).

Blending using a sequence of 5 images:

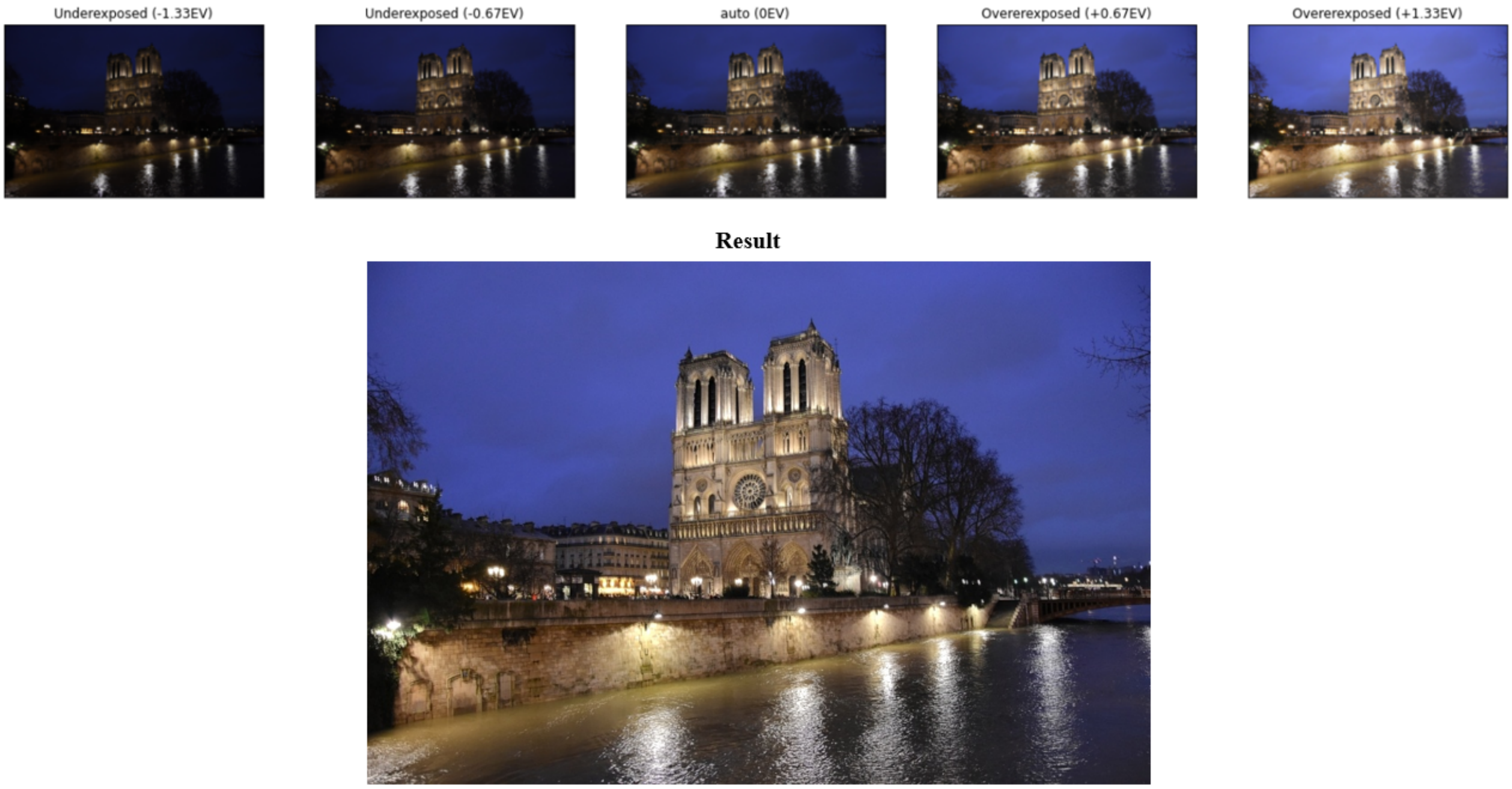

Figure 10: The input sequence (top) and the result of the multi-resolution exposure fusion (bottom) of NOtre Dame de Paris

Blending using a sequence of 13 images:

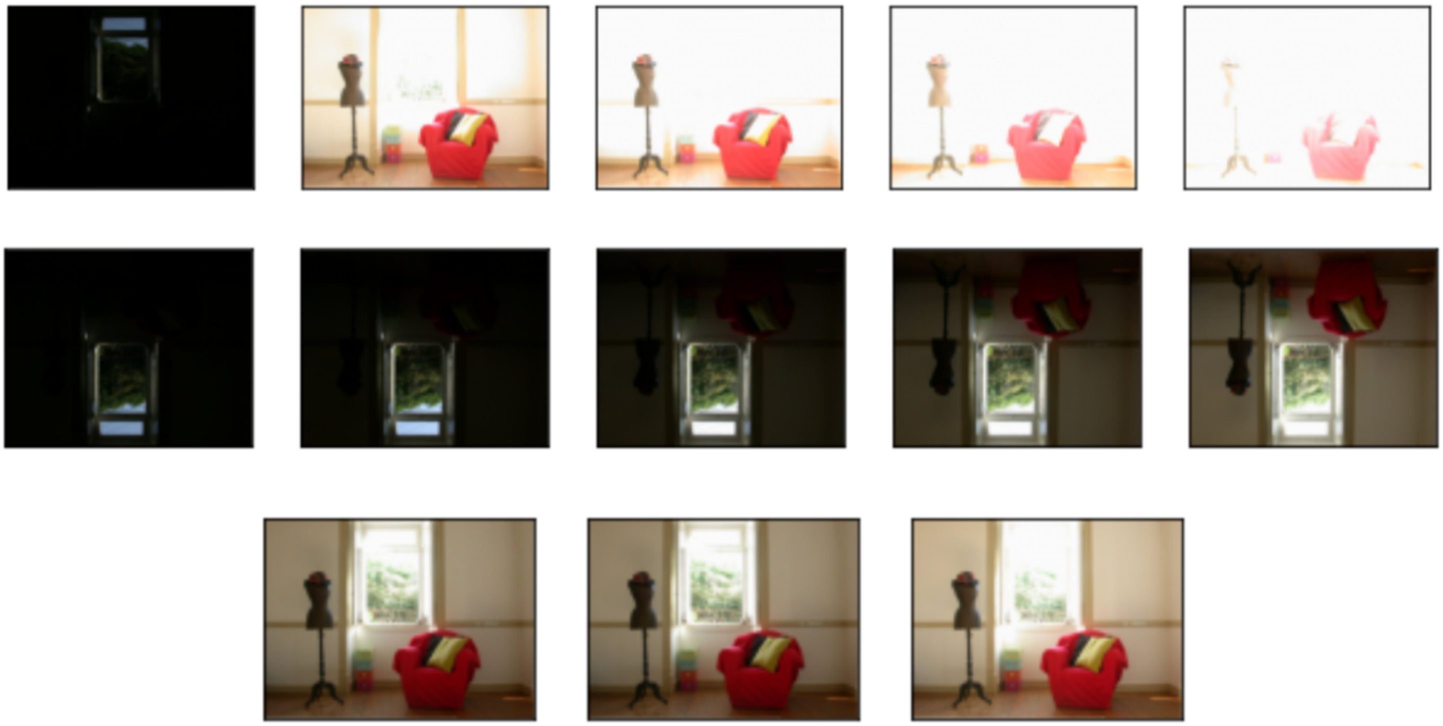

Figure 11: Input Sequence

Figure 12: Fusion result