



Published in Image Processing On Line on 2019-12-29.
Submitted on 2017-09-17, accepted on 2019-12-28.
ISSN 2105-1232 © 2019 IPOL & the authors CC-BY-NC-SA
This article is available online with supplementary materials,
software, datasets and online demo at
<https://doi.org/10.5201/ipol.2019.278>

Extended Exposure Fusion

Charles Hessel

CMLA, ENS Cachan, CNRS, Université Paris-Saclay, 94235 Cachan, France

Communicated by Jose-Luis Lisani and Jean-Michel Morel *Demo edited by* Charles Hessel

Abstract

Extended Exposure Fusion (EEF) is a high dynamic range imaging technique. It was recently proposed as an improvement of an earlier technique called Exposure Fusion (EF), itself widely used thanks to its very good results. But EF has two well-known artifacts: an out-of-range artifact and a low-frequency halo. The extended version solves them, and delivers fused results with enhanced contrast everywhere in the image. We give in this paper a precise description, analysis and implementation of the extended exposure fusion. We notably verify that the artifacts are removed and that the local contrast is improved compared to the original exposure fusion.

Source Code

The Matlab/Octave source code, the code documentation, and the online demo are accessible at [the web page of this article](#)¹. Usage instructions are included in the `README.txt` file of the archive.

Keywords: high dynamic range; image fusion; tone-mapping; Laplacian pyramids

1 Introduction

Extended Exposure Fusion (EEF) is a high dynamic range imaging technique to fuse a bracketed exposure sequence into a high quality image, introduced in [3]. It is one of the numerous offsprings of the method called Exposure Fusion (EF), introduced in 2007 by Mertens et al. [9, 10].

The EF algorithm, contrarily to most HDR imaging techniques, does not create an intermediate HDR image but directly constructs the final LDR image by seamlessly fusing the best regions of the input sequence, using the Laplacian pyramid. A precise description of the method along with its two main limitations, namely, an out-of-range artifact and a low-frequency halo, can be found in [2].

In this paper, following the lines of the method introduced in [3], we describe the Extended Exposure Fusion. The key modification of the initial method consists in creating a series of images with reduced dynamic range, from each input. This is done through *remapping functions*. It appears

¹<https://doi.org/10.5201/ipol.2019.278>

that this simple correction solves the two known artifacts of Exposure Fusion, described by Mertens et al. themselves [10].

The remainder of this paper is organized as follows. In Section 2, the Extended Exposure Fusion proposed in [3] is accurately described. Section 3 confronts the initial method with the improved one and verifies carefully that the artifacts are solved. The effect of the two parameters on the fusion result, namely the dynamic reduction and the pyramid’s depth, are studied. The experiments give evidence that the images obtained with Extended Exposure Fusion indeed have more local contrast than when using the original method, as was stated in [3].

2 Extended Exposure Fusion (EEF)

Extended Exposure Fusion has three main steps. First, the initial bracketed exposure sequence is extended. Then, the extended sequence is fused with exposure fusion. Finally, a normalization step is added to stretch the values to the output dynamic range. These three points are described in Sections 2.1, 2.2, and 2.3.

2.1 Sequence Extension

The correction of the out-of-range artifact proposed in [3] simply consists in separating each input image in a series of images with reduced dynamic range. To do so, the dynamic range of each input image is distributed in several new images created from it. To this aim, the authors use the remapping function g defined in Equation (2). Its only parameter is the restrained range width $\beta \in (0, 1]$. The number of generated images per input one is $M = \lceil 1/\beta \rceil$, where $\lceil \cdot \rceil$ is the closest superior integer operator. This ensures that the full initial range is covered by the generated images. The image sequence is then extended from N to $M \times N$ images.

Let us call $\rho(k)$ the center of the reduced range, where $k \in \{0, \dots, M - 1\}$ is the index of the generated image in the series. The range is then limited to $[\rho(k) - \frac{\beta}{2}, \rho(k) + \frac{\beta}{2}]$, with

$$\rho(k) = 1 - \frac{\beta}{2} - k \frac{1 - \beta}{M - 1}. \tag{1}$$

Rather than brutally clipping the values outside this restrained range, the authors designed a function g that progressively reduces the contrast until it becomes zero.

$$g(t; k) = \begin{cases} t, & \text{if } |t - \rho(k)| \leq \frac{\beta}{2} \\ \text{sign}(t - \rho(k)) \left(a - \frac{\lambda^2}{|t - \rho(k)| - b} \right) + \rho(k), & \text{otherwise,} \end{cases} \tag{2}$$

where t denotes an intensity, $a = \frac{\beta}{2} + \lambda$ and $b = \frac{\beta}{2} - \lambda$, and λ is a parameter controlling the speed of the decay in the function outside the valid part of the range. Its value is fixed to $\lambda = 0.125$ in [3]. The decay behaves like $1/x^2$. As stated in [3], the particular shape of g is not important; any function with a sufficiently fast decay and a smooth transition to outside the valid range can be used. Smooth transitions avoid the creation of false edges that could be transmitted to the final image. An example of remapping functions for $\beta = 0.4$ is presented in Figure 1. This remapping is applied to the color images. Once the sequence is extended, it is simply fused with the Mertens et al. algorithm, which we recall in Section 2.2.

2.2 Exposure Fusion

Exposure fusion [9, 10] first measures the perceptual quality of the input sequence’s images using three pixel-wise metrics: contrast C , saturation S and well-exposedness E . We will denote in the

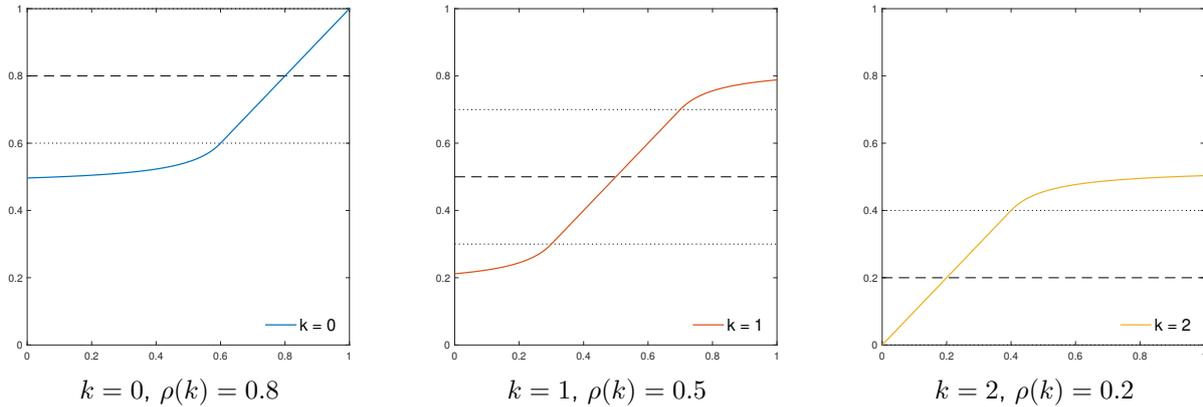


Figure 1: Set of functions g for $\beta = 0.4$. In this case $M = \lceil 1/\beta \rceil = 3$, thus $k \in \{0, 1, 2\}$. The value $\rho(k)$ is displayed with the dashed line, and the restrained interval is delimited by the dotted lines. Outside this restrained range (linear part of the remapping function), the contrast is progressively cancelled. These functions are three different mappings of the input range to a restrained dynamic range.

following by u the image, by $\mathbf{x} = (x_1, x_2)$ the position of the pixel, by c the color channel, and by k the position of the image in the sequence. The *contrast metric* uses the absolute value of a discrete Laplacian filter applied to the grayscale version of the image,

$$C_k(\mathbf{x}) = \left| \left(\frac{1}{3} \sum_{c=1}^3 u_{c,k} \right) * K_{\text{Laplacian}} \right|(\mathbf{x}), \quad (3)$$

where $K_{\text{Laplacian}}$ is a Laplacian kernel that computes the sum of differences over the four nearest neighbors. The *saturation metric* is the standard-deviation of the pixel's color,

$$S_k(\mathbf{x}) = \sqrt{\frac{1}{3} \sum_{c'=1}^3 \left(u_{c',k}(\mathbf{x}) - \frac{1}{3} \sum_{c=1}^3 u_{c,k}(\mathbf{x}) \right)^2}. \quad (4)$$

Finally, the *well-exposedness* measures how close each pixel's color is to the median value 0.5 using a Gauss curve

$$E_k(\mathbf{x}) = \prod_{c=1}^3 \exp \frac{-(u_{c,k}(\mathbf{x}) - 0.5)^2}{2\sigma^2}, \quad (5)$$

with $\sigma = 0.2$. We follow the convention that the images' displayable range is $[0, 1]$. The quality measure of each pixel is finally obtained as a product of the above three metrics

$$w_k(\mathbf{x}) = C_k(\mathbf{x})^{\omega_c} \cdot S_k(\mathbf{x})^{\omega_s} \cdot E_k(\mathbf{x})^{\omega_e}. \quad (6)$$

The initial method [9] adds the exponents ω_c , ω_s and ω_e (each in $[0, 1]$) to the different metrics to allow the user to balance their importance, but the authors in [3] keep them constantly fixed to 1, which is equivalent to removing them. For the blending process, the resulting weights need to be normalized as

$$\hat{w}_k(\mathbf{x}) = \frac{w_k(\mathbf{x})}{\sum_{k'=1}^N w_{k'}(\mathbf{x})}. \quad (7)$$

The input images are then fused according to their normalized weight maps using the Ogden et al. multiscale fusion [11]. This technique builds the Laplacian pyramid [1] of the output image by blending the Laplacian pyramids of the input images according to the Gaussian pyramid of the weight maps. The fused image is obtained by collapsing the constructed pyramid. We will denote

$L\{u\}$ the Laplacian pyramid of u , $G\{w\}_k$ the Gaussian pyramid of the weights w_k , and l the scale. The blending operation is then

$$L\{v\}^l(\mathbf{x}) = \sum_{k=1}^N G\{\hat{w}\}_k^l(\mathbf{x})L\{u\}_k^l(\mathbf{x}). \quad (8)$$

Algorithm 1 describes the whole process, from the quality measurements to the multiscale fusion. We refer to [2] for pseudo-codes of the downsampling and upsampling procedures.

Algorithm 1: Exposure Fusion

```

input : sequence of images  $\{u_0, \dots, u_k, \dots, u_{N-1}\}$ 
output: fused image  $v$ 
1 foreach  $k \in \{0, \dots, N - 1\}$  do
2    $C_k \leftarrow \left| \left( \frac{1}{3} \sum_{c=1}^3 u_{c,k} \right) * K_{\text{Laplacian}} \right|$  // Contrast; Equation (3)
3    $S_k \leftarrow \left( \frac{1}{3} \sum_{c=1}^3 (u_{c,k} - \frac{1}{3} \sum_{c=1}^3 u_{c,k})^2 \right)^{1/2}$  // Saturation; Equation (4)
4    $E_k \leftarrow \prod_{c=1}^3 \exp -0.5 (u_{c,k} - 0.5)^2 / \sigma^2$  // Well-exposedness; Equation (5)
5    $w_k \leftarrow C_k \cdot S_k \cdot E_k$  // weight maps; Equation (6)
6    $\hat{w}_k \leftarrow w_k / \sum_{k=0}^{N-1} w_k$  // Final normalized weight map; Equation (7)
7 Initialize output pyramid  $L\{v\}$  with zeros
8 foreach  $k \in \{0, \dots, N - 1\}$  do
9   Compute  $G\{\hat{w}\}_k$  // Gaussian pyramid of  $\hat{w}_k$ 
10  Compute  $L\{u\}_k$  // Laplacian pyramid of  $u_k$ 
11  foreach scale  $l$  do
12   $L\{v\}^l \leftarrow L\{v\}^l + G\{\hat{w}\}_k^l L\{u\}_k^l$  // Update output pyramid: Equation (8)
13  $v \leftarrow$  collapse Laplacian pyramid  $L\{v\}$ 

```

2.3 Robust Normalization

After fusion, the fused image’s intensities need to be rescaled to the unit range. Depending on β and the content of the image, this typically amounts to stretch the intensity range (as opposed to EF). This final step over-stretches the colors to $[0, 1]$ by allowing a percentage of clipping (typically 1%) in both sides of the histogram. This operation is called Robust Normalization [2, 3]. We refer to [2] for a pseudo-code of this algorithm.

The pseudo-code of the method is presented in Algorithm 2. At Line 4 the input sequence is extended: for each image of the initial bracketed exposure sequence, M images are created with a restrained dynamic range. At line 5 the extended sequence is fused. An example of sequence extended using this method is presented in Figure 2. The associated weight maps are displayed in the bottom right corner of each image.

3 Results and Comparisons

While this correction of Mertens’ et al. method seems to be targeted to the out-of-range artifact, the authors in [3] argue that it has two important consequences:

1. A gain in the contrast of the fused images: not only because it avoids the need to compress the final image to make it fit the standard output dynamic range, but also because using the

Algorithm 2: Extended Exposure Fusion (EEF)

```

input : sequence  $\{u_0, \dots, u_{N-1}\}$ 
input : reduced range  $\beta$ 
output:  $z$ : fused image
1  $M \leftarrow \lceil 1/\beta \rceil$ 
2 for  $n \in \{0, \dots, N-1\}$  do
3   for  $k \in \{0, \dots, M-1\}$  do
4      $v_{k+n \times M} \leftarrow g(u_n; k)$  // Equation (2)
5  $z' \leftarrow \text{EXPOSUREFUSION}(v)$  // Algorithm 1
6  $z \leftarrow \text{ROBUSTNORMALIZATION}(z')$ 

```

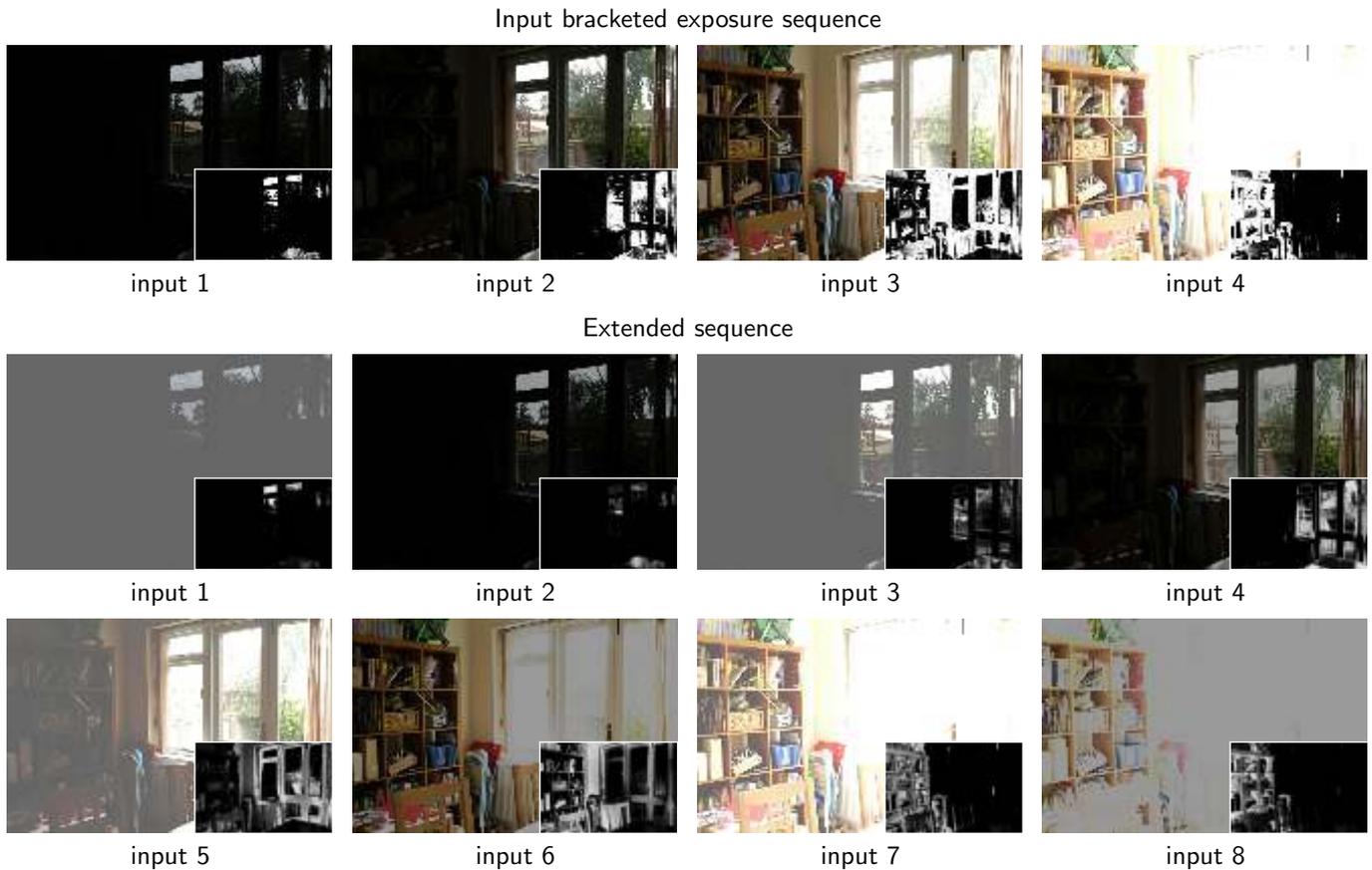


Figure 2: Example of simulated sequence in EEF with $\beta = 1/2$ (sequence “House”). The input bracketed sequence is displayed on the top row, along with its associated weight maps. The extended sequence is presented on the center and bottom row, with their associated weight maps too. Similarly to the images, the weight maps are also split in two.

same strategy, the output range can be forced to be smaller than $[0, 1]$ (using small β). In that configuration the final image is stretched rather than compressed and therefore gains local contrast. Results and comparisons with the initial method are provided in Section 3.1.

2. It allows to use deeper pyramids, and, in consequence, to avoid the low-frequency halo artifact. Indeed, this artifact of the method is sometimes observed when there are strong intensity differences between two distant areas of the input sequence [10, 2]. This can be solved using deeper pyramids as suggested by the authors themselves; yet deeper pyramids also yield stronger out-of-range artifacts and therefore are not adopted in the original method. Hence, by solving

the latter, the authors [3] also solved the low frequency halo. This is discussed in Section 3.3.

In brief, with this simple modification of the algorithm, the two identified shortcomings of the initial method are solved, and the fused image can be even more contrasted locally than the input sequence. The negative side of this correction is an increase of the computational time by a factor M (typically between 2 and 4), compared to the initial EF method.

The next Sections 3.1 and 3.3 compare EEF to the original EF. Section 3.4 will present related methods and compare them to EF and EEF.

3.1 Out-of-range Artifact Correction and Local Contrast

In Figure 2 we show the extended sequence generated from a four-images sequence. Using $\beta = 1/2$, each image's range is "split" in two. The weights attributed to each image by EF are displayed on the corner of their associated image.

In Figure 3 we compare the results obtained with EF (last column, equivalent to $\beta = 1$) and EEF, for several values of β . The remapping functions, used to distribute the initial intensities between the generated images, are displayed on the top row. The robust normalization was applied to all fused images. As a consequence, EF results are compressed and EEF results stretched. Furthermore, the smaller β , the stronger the final stretching in EEF. We observe, by comparing the results to the input sequences given in Figures 2 and 4, that the EEF results with $\beta = 0.5$ have the same local contrast as the input images (or slightly more), and more contrast than the EF outputs. The results obtained with smaller β are even more contrasted locally.

This can be easily verified in Figure 5, where we plotted an horizontal line going through the top left pane of the windows of the fused "House" image, taken in the EF result (in blue) and the EEF one, with $\beta = 1/2$ (in red) and with $\beta = 1/4$ (in yellow). The out-of-range values are indeed removed, while the local contrast is as well preserved as in EF. Only the strong edges are compressed, which is the aim of tone-mapping operators.

In Figure 6, we reproduced the experiment of [2] that clearly illustrated the out-of-range artifact of EF in the case of a simple test-pattern. This experiment is the same as in Figure 5 but with test patterns that are constant (apart from the noise) in the vertical axis, which makes the plots easier to interpret. The noise was used to simulate texture. The two top plots display the central horizontal lines of the three input images (left) and the six remapped ones (right). The central plot displays the same lines taken in the fused results. The full images are shown on the bottom row. The fused result with EF (purple line) has out-of-range artifacts for both bright and dark values, whereas the result with EEF (green line) has no artifact.

Arguably, in Figure 6 the local contrast in the bright and dark bands of the EEF result is somewhat reduced with respect to the EF result. This happens because of the combination of two effects. First, the contrast in the remapped images outside the restrained range is not completely canceled, due to the shape of function g . This is visible in the top right plot of Figure 6. Thus the contrast metric in these areas is close to zero but not null. Second, values outside the valid range can be assigned higher weights by the well-exposedness measure than values inside, if they are closer to 0.5. When these two effects are combined, the final weights assigned to remapped values outside the restrained range can become non-negligible. This means that in those rare situations the local contrast will be slightly reduced. This explains the slight loss of contrast in the bright and dark bands of the EEF result with respect to the EF one in the central plot of Figure 6. The issue can be alleviated by using a smaller weight ω_e for the well-exposedness measure, or a smaller parameter λ in the function g (which will cause a stronger contrast reduction outside the valid range). Remark that in Figure 6 the loss of local contrast in the same regions of the EF result after clipping or linear

EXTENDED EXPOSURE FUSION

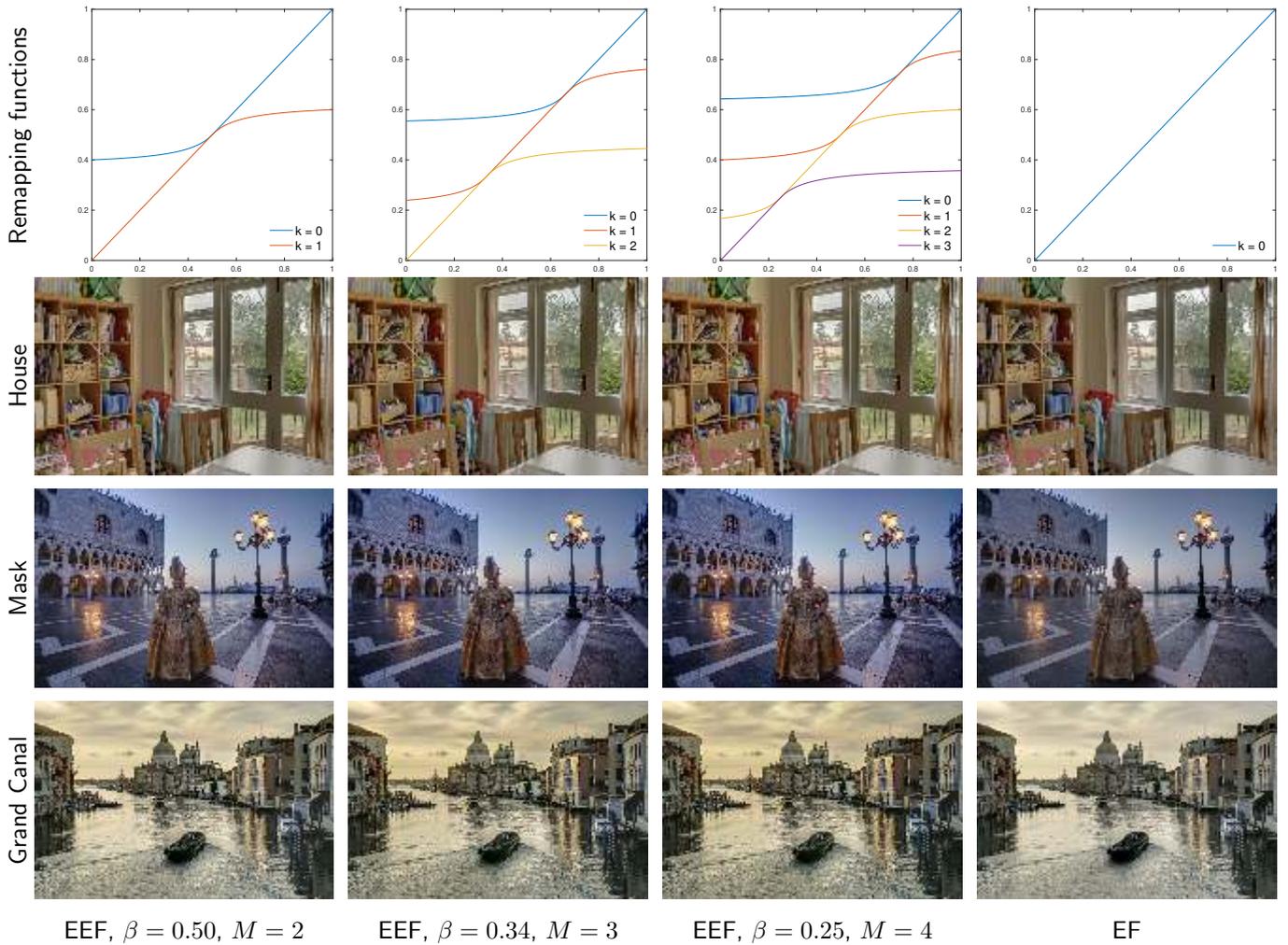


Figure 3: Comparison between Exposure Fusion and Extended Exposure Fusion. In this example all images underwent the robust normalization with 0.1% clipping in the white and 0.9% in the black. For EF, this results in a compression of the contrast, due to the out-of-range artifact, whereas for EEF the dynamic dynamic is stretched (the smaller β , the stronger the stretching). The top row displays the set of remapping functions $\{g(t; k) \mid k = 0, \dots, M-1\}$ used to extend the input bracketed sequence. In the case of the exposure fusion method [9] (right column), the sequence is not extended, which can be seen as a special case of EEF [3] where $\beta = 1$ and the remapping function is the identity.



Figure 4: Input bracketed exposure sequences “Mask” and “Grand Canal”, each composed of three images: under-exposed, correctly exposed, and over-exposed. The input sequence “House” is displayed in Figure 2.

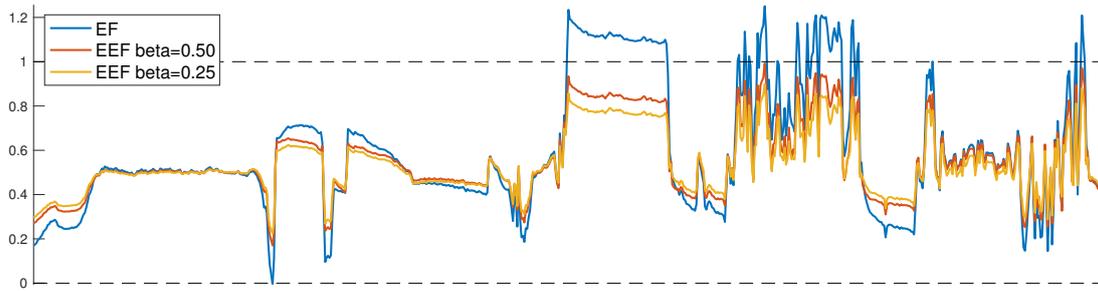


Figure 5: Comparison between EF (blue) and EEF (red and yellow) for a line of the House sequence. EEF is used with $\beta=1/2$ and $\beta=1/4$. This horizontal line goes through the top left pane of the windows. The EF method has clear out-of-range artifacts, whereas EEF has not. The fusion results are displayed as is, without affine rescaling. To fit the displayable range, the EF result needs compression, and thus loses contrast. On the contrary, the final dynamic range obtained with EEF is generally smaller than $[0, 1]$, thus its contrast is increased.

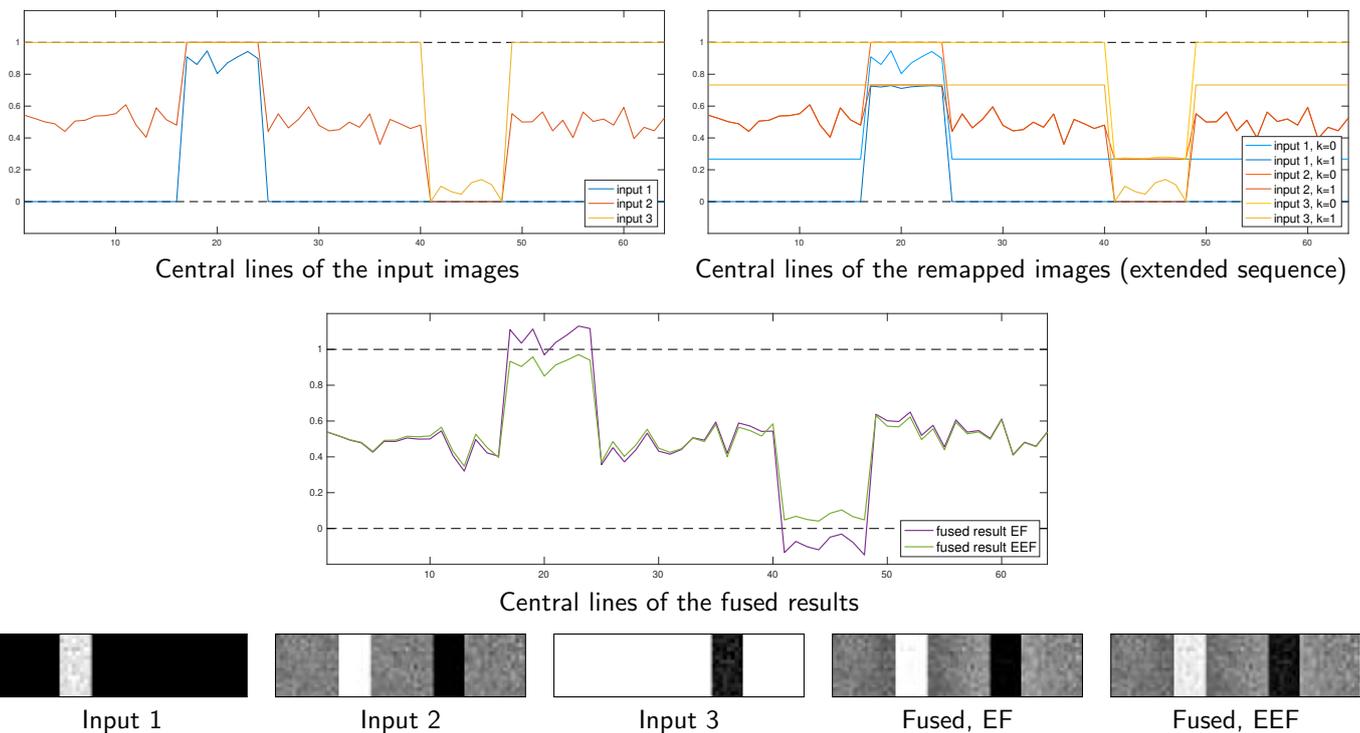


Figure 6: Verification with a test-pattern of the out-of-range artifact in EF and EEF. We present on the top left plot sections of the three input images, on the top right plot sections of the remapped input images (the extended sequence fused by EEF), and on the center plot the same sections taken in the fused results by EF and EEF. The bottom row shows the three input images and the two fused results. The fusion with EF [9] causes out-of-range artifacts, whereas the fusion with EEF [3] does not. The parameter in EEF was set to $\beta = 0.64$. This experiment is reproduced from [2].

rescaling would be much more serious. We nevertheless suggest a correction to avoid this issue, described in Section 3.2.

Additionally, we applied EF and EEF on the bigger bracketed sequence dataset of Ma et al. [7] and measured the scaling factors applied by the robust normalization on the fused images. The resulting compression/stretching factors for the 17 sequences are displayed in Table 1. On average, the EF fused images are compressed by a factor 0.766. In contrast, EEF fused images, using $\beta = 0.64$ (i.e. $M = 2$) are only very slightly compressed, by a factor 0.970.

These experiments suggest that EEF does not require the simulation of many additional images to correct EF's out-of-range artifact: a division of each input in two distinct ones is enough. Yet the generation of more inputs, with smaller reduced ranges, leads to an even greater local contrast gain.

	Balloons	Belgium House	Lamp	Candle	Cave	Chinese garden	Farmhouse	House	Kluki	
EF	0.585	0.693	0.649	0.553	0.826	0.915	0.708	0.715	0.785	
EEF	0.785	0.858	0.827	0.719	1.066	1.128	0.936	0.881	1.038	...
		Lamp2	Landscape	Lighthouse	Madison capitol	Memorial	Office	Tower	Grand Canal	Average
		0.634	1.330	0.776	0.685	0.660	0.849	0.726	0.936	0.766
...		0.866	1.513	1.001	0.871	0.828	1.041	0.966	1.158	0.970

Table 1: Final robust normalization factors on Ma et al. dataset [7]. Sequences “Lamp”, “House”, “Memorial” and “Grand Canal” are the same (but downsampled) as the ones we used. Here, the EEF parameter is $\beta = 0.64$ ($M = 2$). A smaller β would lead to more dynamic reduction (thus a larger final normalization factor) but also requires longer extended sequences.

We display in Figure 7 and Figure 8 two additional results with long sequences.

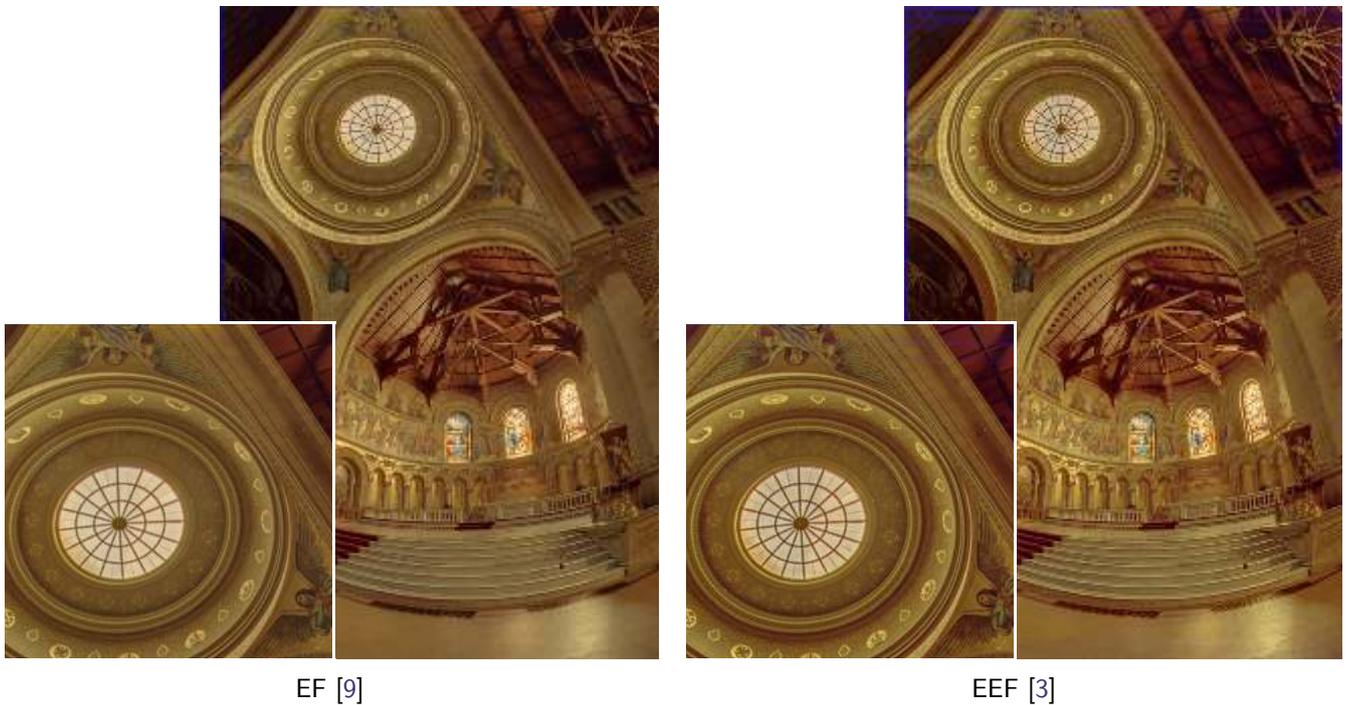


Figure 7: Comparing the EF and EEF method ($\beta = 1/4$) on the Memorial image (the input sequence has 16 images). Robust normalization was applied to both images, with 0.1% and 0.9% of clipping in the white and in the black respectively. In the EF case, this required to apply a factor 0.675, that is, a strong compression. In the EEF case however, the factor was 1.265, which entails a contrast increase everywhere in the image.

Effect of the parameter on the result. In the end, the effect of the parameter β on the result is quite simple: the smaller β , the stronger the range compression and, consequently, the stronger the final stretching of the dynamic. Independently, even with the restrained range strategy, the EEF method cannot guarantee that the output image does not contain out-of-range artifacts. This is manifest in Table 1: this issue remains for example in all images for which the robust normalization factor is inferior to one. However, it is guaranteed that EEF reduces this artifact with respect to EF, and this can be verified in the same table. It should also be noted that Table 1 uses $\beta = 0.64$. This means that there is still room for more dynamic compression (with smaller values). Since there is no way to predict the amount of out-of-range artifact, we recommend to use an intermediate value for β , e.g. between 0.25 and 0.75, and then adjust for a specific image if needed. We used 0.3 in the demo



Figure 8: Comparing EF and EEF ($\beta = 1/4$) on the Lamp image (the input sequence has 15 images). Robust normalization was applied to both images, with 0.1% and 0.9% clipping in the white and in the black respectively. For EF, this amounted to apply a 0.676 factor, namely a strong compression. For EEF however, the factor was 1.299, namely a contrast increase everywhere in the image.

associated with this paper. Too small values for β may not be recommendable, since they increase the computation time. They nonetheless yield results with strong local contrast and saturation, similar to the unnatural yet common “HDR rendition” often seen in consumer photography.

3.2 Improvement of EEF’s Weights

This section is a parenthesis in the description of the EEF algorithm [3] where we propose a small correction for the issue described in the previous Section 3.1.

As explained in Section 3.1, it can happen that the EEF method slightly reduces the local contrast of the very bright or dark regions. To correct this, we propose the following modification: to reduce the weights attributed to the values outside the restrained range. This is done through a new metric, equal to one in the restrained range and inferior to one outside. We use for this purpose the derivative of g , given by

$$g'(t; k) = \begin{cases} 1, & \text{if } |t - \rho(k)| \leq \frac{\beta}{2} \\ \frac{\lambda^2}{(|t - \rho(k)| - b)^2}, & \text{otherwise.} \end{cases} \quad (9)$$

The metric is called R for restrained range, and computed as follows

$$R_k(\mathbf{x}) = \prod_{c=1}^3 g'(u_c(\mathbf{x}); k), \quad (10)$$

where u is an image from the initial sequence, before remapping. The weights are computed as

$$w_k(\mathbf{x}) = C_k(\mathbf{x}) \cdot S_k(\mathbf{x}) \cdot S_k(\mathbf{x}) \cdot R_k(\mathbf{x}), \quad (11)$$

and then normalized as usual. There are no other changes. This modification of the EEF method is reported in Algorithm 3. Only line 5 and 6 are modified with respect to Algorithm 2. Figure 9 shows on the test-pattern that the local contrast is indeed better preserved by including this supplementary metric. Figure 10 shows that some regions which lack contrast in the EEF result are improved thanks to these modified weights. Observe for example the boat and the bush on the right.

This correction is included in the online demo and activated by default, with an option to deactivate it. Apart from this section, all the discussions in this paper are based on the version published in [3], i.e. without the improved weights.

Algorithm 3: Extended Exposure Fusion (EEF), with improved weights

```

input : sequence  $\{u_0, \dots, u_{N-1}\}$ 
input : reduced range  $\beta$ 
output:  $z$ : fused image
1  $M \leftarrow \lceil 1/\beta \rceil$ 
2 for  $n \in \{0, \dots, N-1\}$  do
3   for  $k \in \{0, \dots, M-1\}$  do
4      $v_{k+n \times M} \leftarrow g(u_n; k)$  // Equation (2)
5      $R_{k+n \times M} \leftarrow \prod_{c=1}^3 g'(u_{c,n}; k)$  // Equation (10)
6  $z' \leftarrow \text{EXPOSUREFUSION}(v, R)$  // Algorithm 1 with Equation (6) replaced by (11)
7  $z \leftarrow \text{ROBUSTNORMALIZATION}(z')$ 

```

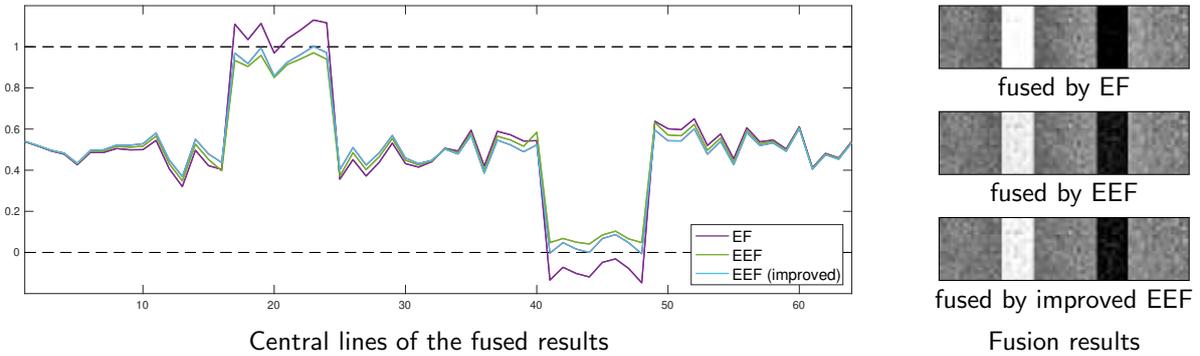


Figure 9: Comparing the EEF method with improved weights (as described in Section 3.2) with the original [3]. The improvement simply consists in reducing the weights when they are outside the restrained range. This helps preserving local contrast in very bright and dark regions, like the white and black bands in this test-pattern.

3.3 Low-frequency Halo and Pyramid Depth

The low-frequency halo was described by the authors of Exposure Fusion in [10]. It is explained in [2]. Its solution consists in using deeper pyramids [10, 2]. However, this also increases the out-of-range artifact [2], so this solution is not practical with the original method. As suggested for EEF in [3] however, since the out-of-range artifacts are reduced, pyramids can go deeper without artifact and then solve the low-frequency halo.

3.3.1 Pyramid Depth

In Mertens’ implementation of exposure fusion, the depth D of the pyramid is computed as

$$l_{\max}^{\text{EF}} = \lfloor \log(\min\{H, W\}) / \log 2 \rfloor, \tag{12}$$

where H and W are respectively the height and width of the input images. Since these dimensions are rarely powers of two, the residual image of the pyramid is often a few pixels wide in its smallest dimension. Nevertheless, there is no practical objection in continuing the downsampling when one

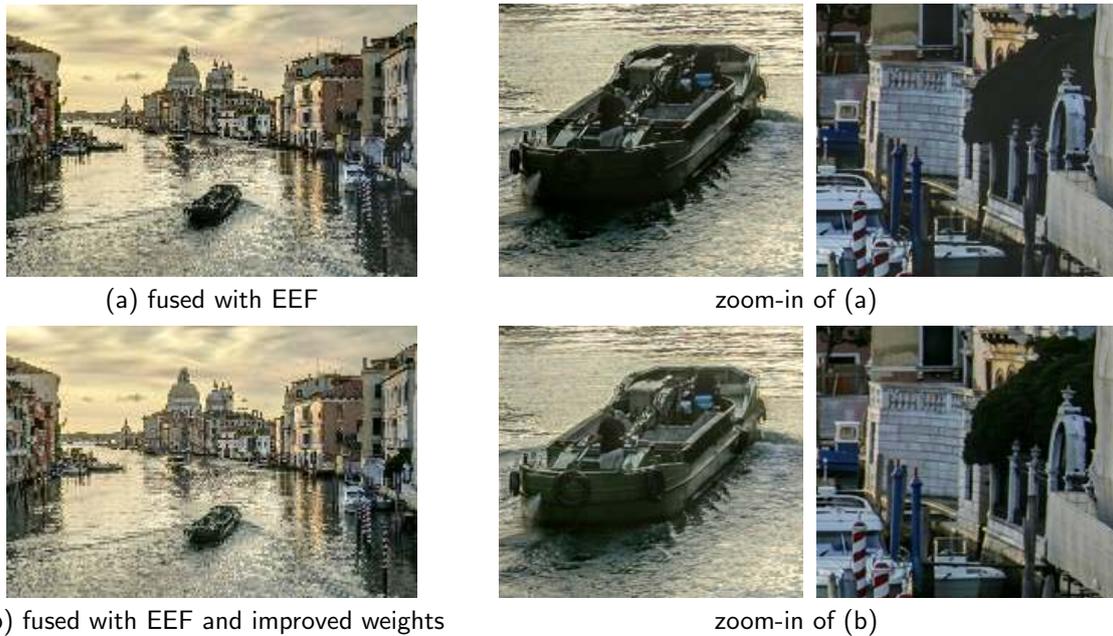


Figure 10: Comparing the EEF method in [3] and the EEF method with the improved weights proposed in Section 3.2. The contrast in the dark regions is better with the improved weights. $\beta = 0.3$.

dimension has only one pixel left; indeed, the borders are replicated and it can be downsampled anyway: the Laplacian coefficients simply become zeroes. Hence, the authors suggest to use the max rather than the min in Equation (12), so as to handle images with one dimension considerably smaller than the other. Yet, the residual size might still be a few pixels wide.

In the implementation of the Gaussian and Laplacian pyramids, a new line/column is added when the corresponding dimension is odd. Denoting D^l a dimension (H or W) of an image at scale l , its value is then

$$D^l = \lceil D^{l-1}/2 \rceil, \tag{13}$$

with D^0 the dimension at scale zero, i.e. the initial one. The genuine deepest scale of a pyramid is then the smallest l such that $D^l = 1$, with $D^0 = \max\{H, W\}$. We call it l_{\max} . Alternatively, one can stop when the smallest dimension of the image reaches 1, that is, by starting at $D^0 = \min\{H, W\}$; we call it l'_{\max} . Both are proposed in the online demo, l_{\max} is denoted “deepest” and l'_{\max} “deeper”.

3.3.2 Two Experiments

In Figure 11 we present an example of low-frequency artifact and its resolution with deeper pyramid. For this experiment we created an image made of plateaus and a small noise to simulate texture. Two other images were generated using an affine rescaling of the first one (and keeping the part in $[0, 1]$ only). Lines of the input sequence are displayed in the top plot, and fused results with EF and EEF and different pyramid depths in the center and bottom plots. The low-frequency artifact of the exposure fusion method can be easily spotted on the left and right parts of the center blue line. This is the standard result given by Mertens et al. method. These halos can be removed by using deeper pyramids [10, 2]. But the out-of-range artifact is amplified by this modification [2], as can be seen on the bottom plot. In the EEF method however, no low-frequency nor out-of-range artifacts are visible in the results.

Figure 12 brings yet another empirical proof of this. A real bracketed exposure sequence is fused. A low-frequency halo is visible above the roof in EF’s result, using a pyramid of depth l_{\max}^{EF} . The fused result in EEF with the same depth has the same halo. However, when using l'_{\max} or l_{\max} , the

halo disappears.

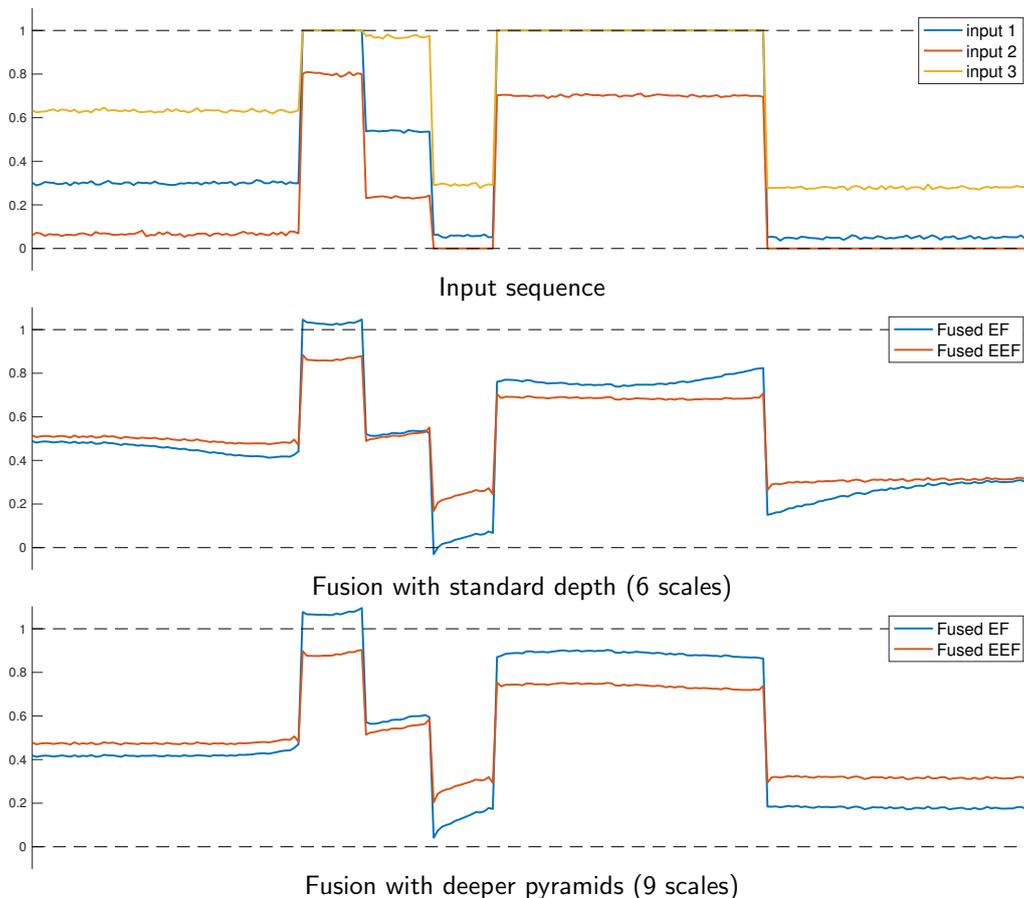


Figure 11: Low frequency halo in exposure fusion and its resolution with deeper pyramids. The top plot displays the input sequence. The center plot shows the standard fusion result of EF and compares it to the EEF result. Low-frequency halos, typical of EF, are visible on the left and right parts of the EF result (blue line). As shown in the bottom plot, these spurious low-frequency halos can be removed using deeper pyramids, yet this also increases the out-of-range artifact, so this solution cannot be used in practice. Our result, however, displays no low-frequency artifact nor out-of-range artifacts, independently of the pyramid’s depth.

3.4 Comparison with Other Exposure Fusion Methods

We now compare EF and EEF with three other state-of-the-art exposure fusion based methods: the Kou et al. gradient domain guided exposure fusion [4] (GGMEF), the Ma et al. patch-wise exposure fusion [8] (PWMEF), and the Li et al. detail-enhanced exposure fusion [5] (DEEF).

- Kou et al. [4] circumvented the out-of-range artifact by reducing the depth of the pyramid. But this increased the low-frequency halo, which is partly avoided with the introduction of an edge-aware smoothing pyramid. The approach in [3] is simpler and does not rely on an additional edge-aware smoothing step, but on mere remapping functions. Unexpectedly, this simpler solution also gives better results, since the Kou et al. method does not succeed in removing the halos. Rather than reducing the depth of the pyramid as in [4], EEF increases it. This improves the methods efficiency, as a deeper pyramid gives more natural results, with a better preserved relative brightness.
- The Ma et al. method [8] (PWMEF) is a patch-based method that uses the “luminance”, “contrast” and “structure” decomposition defined in [7] and is based on SSIM [12]. The luminance

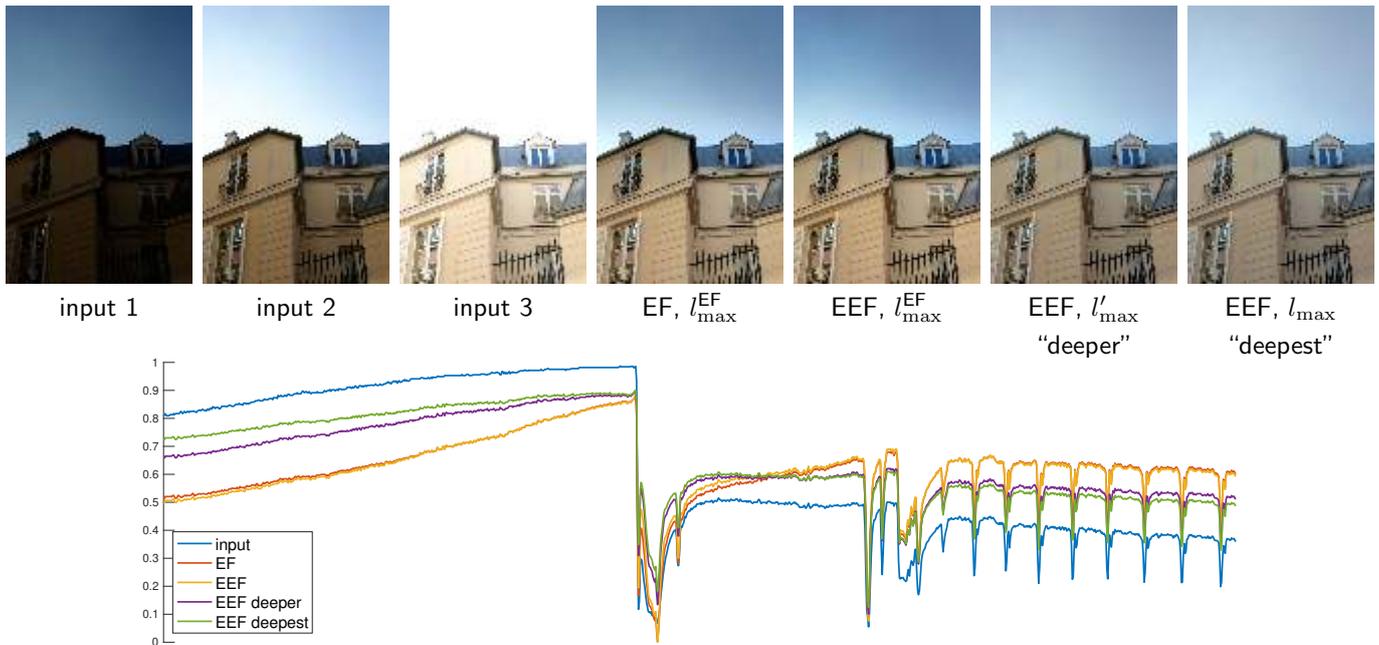


Figure 12: Resolution of the low-frequency halo using deeper pyramids in EEF. The input bracketed sequence is composed of the three first images labeled input 1, 2 and 3. Follows the fusion result of EF, then three different results obtained with EEF and different pyramid depths l_{\max}^{EF} , l'_{\max} and l_{\max} (from shallow to deep). A vertical line of these images is plotted on the graph below (the images were converted in gray beforehand). The image labeled input (blue line) is the second one of the bracketed exposure sequence. This illustrates how a deeper pyramid avoids the creation of halos.

l is the mean value of the patch, the contrast c is its ℓ^2 norm, and the structure \mathbf{s} is the patch after removal of its mean value and division by its ℓ^2 norm, so that $\mathbf{x} = c \cdot \mathbf{s} + l$, where \mathbf{x} is the patch and the bold font denotes vectors. The Ma et al. method then simply consists in constructing the best luminance, contrast and structure values from the input sequence patches, and finally to invert the decomposition to obtain a new patch, optimal according to these three components. The structure and luminance are obtained by a weighted combination of the input patches values, using metrics similar to Mertens et al. [9]. As for the final contrast value, the max is simply taken. After reconstruction, overlapping patches are averaged.

- The Li et al. [5] detail-enhanced exposure fusion (DEEF) is a modified version of exposure fusion with an additional and final contrast enhancement step. The modification of exposure fusion consists in a supplementary edge-aware smoothing step for the weight maps: the Gaussian pyramid associated with each input image is smoothed using the weighted guided image filter [6]. The guide is the Gaussian pyramid of the luminance of the input image. The fusion is then carried out like in exposure fusion, but the obtained image is not the final result: A detail layer is computed from the input sequence using a structure tensor and obtained by solving a quadratic optimization problem. The final result is obtained by adding this detail layer to the fused image.

We present in Figure 13 fusion results with the methods of Kou et al. [4] (GGMEF), Ma et al. [8] (PWMEF) and Li et al. [5] (DEEF). The GGMEF method produces halos, that can easily be spotted in the sky of the Grand canal image. The PWMEF method produces excessively bright results. It is visually pleasing, but one can notice out-of-range artifacts in the house image. Furthermore, the bookshelf is exaggeratedly saturated. Finally, the DEEF results are dark and lack local contrast. In fact, they look very similar to those of EF. The EEF results show a larger local contrast. This is visible in the zoom-in on the mask result. The method delivers fused images without halo and without out-of-range artifacts.



Figure 13: Comparison between the original EF, EEf (with $\beta = 1/4$), and other fusion methods. All images underwent robust normalization with 0.1% of clipping in the white and 0.9% in the black.

4 Conclusion

We have described and analyzed the Extended Exposure Fusion, recently introduced in [3] to fuse a bracketed exposure sequence into a single high quality image. This method is similar to the previous and widely used Exposure Fusion [9, 10] but solves its out-of-range artifact and low-frequency halo, thanks to a simple modification consisting in reducing the dynamic range of the input images. To preserve all information, the range from each input image is distributed into two or more new images, thus fabricating an extended sequence which is then simply fused by exposure fusion. Not only this correction solves the initial method’s issues, but it also delivers fused images with increased local contrast with respect to the initial sequence. This corrected version of Exposure Fusion comes at the price of an increased computational time (the input sequence length is increased by an integer factor at least equal to two – and rarely superior to four). Yet it corrects all known artifacts of the Mertens et al. method. We therefore would recommend to use it rather than the original one.

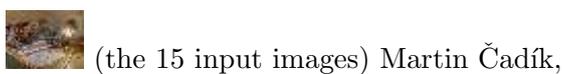
Image Credits



Min H. Kim



Jacques Joffre



(the 15 input images) Martin Čadík,



(the 16 input images) Paul Debevec



Charles Hessel.

Acknowledgment

We thank Jean-Michel Morel for fruitful comments and discussions. Work partly financed by Office of Naval research grant N00014-17-1-2552, DGA Astrid project “filmer la Terre” n°ANR-17-ASTR-0013-01, and ANRT CIFRE Ph.D. scholarship n°2014/1323 of the French Ministry for Higher Studies, Research and Innovation.

References

- [1] P. BURT AND E. ADELSON, *The Laplacian pyramid as a compact image code*, IEEE Transactions on Communications, 31 (1983), pp. 532–540. <https://doi.org/10.1109/TCOM.1983.1095851>.
- [2] C. HESSEL, *An Implementation of the Exposure Fusion Algorithm*, Image Processing On Line, 8 (2018), pp. 369–387. <https://doi.org/10.5201/ipol.2018.230>.
- [3] C. HESSEL AND J.-M. MOREL, *An extended exposure fusion and its application to single image contrast enhancement*, in IEEE Winter Conference on Applications of Computer Vision (WACV), IEEE, 2020.
- [4] F. KOU, Z. LI, C. WEN, AND W. CHEN, *Multi-scale exposure fusion via gradient domain guided image filtering*, in IEEE International Conference on Multimedia and Expo (ICME), IEEE, 2017. <https://doi.org/10.1109/ICME.2017.8019529>.
- [5] Z. LI, Z. WEI, C. WEN, AND J. ZHENG, *Detail-enhanced multi-scale exposure fusion*, IEEE Transactions on Image processing, 26 (2017), pp. 1243–1252. <https://doi.org/10.1109/TIP.2017.2651366>.
- [6] Z. LI, J. ZHENG, Z. ZHU, W. YAO, AND S. WU, *Weighted guided image filtering*, IEEE Transactions on Image processing, 24 (2015), pp. 120–129. <https://doi.org/10.1109/TIP.2014.2371234>.
- [7] K. MA, K. ZENG, AND Z. WANG, *Perceptual quality assessment for multi-exposure image fusion*, IEEE Transactions on Image processing, 24 (2015), pp. 3345–3356. <https://doi.org/10.1109/TIP.2015.2442920>.
- [8] K. MA AND Z. WANG, *Multi-exposure image fusion: A patch-wise approach*, in IEEE International Conference on Image Processing (ICIP), IEEE, 2015. <https://doi.org/10.1109/ICIP.2015.7351094>.
- [9] T. MERTENS, J. KAUTZ, AND F. VAN REETH, *Exposure fusion*, in Pacific Conference on Computer Graphics and Applications, 2007. <https://doi.org/10.1109/PG.2007.17>.
- [10] —, *Exposure fusion: A simple and practical alternative to high dynamic range photography*, Computer Graphics Forum, 28 (2009), pp. 161–171. <https://doi.org/10.1111/j.1467-8659.2008.01171.x>.
- [11] J. M. OGDEN, E. H. ADELSON, J. R. BERGEN, AND P. J. BURT, *Pyramid-based computer graphics*, RCA Engineer, 30 (1985), pp. 4–15.
- [12] Z. WANG, A. C. BOVIK, H. R. SHEIKH, AND E. P. SIMONCELLI, *Image quality assessment: From error visibility to structural similarity*, IEEE Transactions on Image processing, 13 (2004), pp. 600–612. <https://doi.org/10.1109/TIP.2003.819861>.