Exposure fusion

Final project: Introduction to numerical imagery

Raphaël Chekroun
Department of computer science
École normale supérieure
Paris, France
raphael.chekroun@ens.fr

Yoann Coudert–Osmont
Department of computer science
École normale supérieure de Lyon
Lyon, France
yoann.coudert-osmont@ens-lyon.fr

Abstract

This project is based on *Exposure Fusion*, by T. Mertens, J. Kautz, F. Van Reeth [1]. This article aims to present a method they developed to fuse a set of images of the exact same scene with different exposures into a high quality image, without any physically-based means. For the project, we implemented all their ideas, from the first naive one to the most elaborate one. When then compared the obtained fused images, with all those methods, in order to check on their efficiency.

1 Introduction

A limitation of modern digital cameras are their un-ability to capture an as large dynamic range as then one we can see in reality. When a scene have a high dynamic range, it often happens that a picture of it is overexposed (too much light on the receptors, hence getting saturated and making the scene look too clear - or even white) or underexposed (not enough caught light on the receptors, making the scene look too dark - or even black).

The studied article proposed a way to fuse images of the exact same scene taken with different exposure in one high quality, low dynamic range image. To do so, a perceptual quality measure, based on contrast, saturation and exposness, is computed for every pixel of every image. The fused image then is an image taking, for every coordinate, a combination of the "best" pixels in all the input images of the same scene.

Previous methods to compute such an image was to compute first an high dynamic range (HDR) image from the set of input images [2], and then remap the colors in order to display something with a lower dynamic range [3]. Thus, the Exposure Fusion method skip the step of the computation of the HDR image to fuse directly the input images, hence simplifying the acquisition pipeline.

Another strength of this method is that it allows to easily redefine the image qualities to incorporate to the fused image. Indeed, the algorithm starts by computing, for each image, its weight map which is a weighted geometric mean between some relevant characteristics of the image, detailed in the next part. Thus, this method is very adaptive, and allows to easily redefine the metrics grading each pixel of each image, in order to chose the "best one" for the fusing phase.

In this report, we will detail the construction of the weight map, and then use our own implementation[1] of the code to compare exposure fusion with other ideas the authors had, in order to highlight the efficiency of the exposure fusion method.

[1] https://github.com/raphychek/project-numerical-imaging-mva

2 Quality measure and weight map

The input set of image might contain images with overexposed or underexposed areas, whereas those same areas are of good quality in other images. The goal of this method is to advantage pixels of better quality, so that the color they bear will be more impacting on the fused image.

In order to advantage the best pixels, authors came with an idea of defining a weight map for every input image. This weight map is defined such that each coordinate of every weight map correspond to the weight that the corresponding pixel of the corresponding image will have on the pixel of the same coordinate on the fused image.

Hence, each weight can be seen as a metric measuring the quality of a pixel. This quality metric takes into account three measures:

- The constrast: the higher the contrast is on some point, the higher will be its weight. This measure tends to advantage edges and texture. We build a contrast matrix, computed as the absolute value of a Laplacian filtered grayscale version of an image.
- The saturation: as saturated colors are desirable in a photo, pixels with a higher saturation have more weight. A saturation matrix is computed as the standard deviation within the R, G, and B channel at each pixel of an image.
- The exposition: we want to advantage pixels that are neither overexposed (with intensities close to 1) nor underexposed (with intensities close to 0). In order to do so, we build an exposition matrix where each pixel will take the value of a Gauss curve centered in 0.5 (as we want pixel of intensity 0.5 to be more heavy weighted) with a standard deviation of 0.2 (value chosen by the article) evaluated in the intensity of each pixel.

2.1 Computation of the weight map

Let's suppose we are dealing with the k-th image of our dataset.

Hence, once we have computed it's contrast matrix C_k, its saturation matrix S_k and it's exposition matrix E_k, its weight map is defined as:

$$W_k[i,j] = (C_k[i,j])^{w_c} (S_k[i,j])^{w_s} (E_k[i,j])^{w_e}$$

with $W_k[i,j]$ is the weight of the pixel at position (i,j) in the k-th image and w_c, w_s and w_c the importance given to respectively contrast, saturation and exposition measures in the final weight.

The final pixel weight of the k-th image, $W_k[i,j]$, will then be used to guide the fusion process, as seen in part 3.

Let's see what W looks like and the kind of information it bears.

2.2 Illustration of weight maps

We chose to illustrate the weight maps with photos of a room with a window. This choice is motivated by the fact that having a window in the room allows a high difference of lighting between the room and the window (as long as the outside is sunny enough). Thus a photo with the correct exposure to observe the room will always tend to saturate the window (hence making it all white), while a photo with the correct exposure to observe through the windows will always tend to underexpose the room (hence making it all black).

The Figure 1 shows the images of the weight maps W obtained on these pictures.

In the first place, we notice that the images W_i are almost binary: pixels are either completely white or completely black. Nonetheless, some areas of gradient are noticeable. For instance there is a gradient on the table in images W_0 and W_1.

In a second place, we can observe some noise, especially on the wall. In fact, in W_2, the wall is almost entirely white but there are many black pixels on it. This problem is due to the contrast component of the weight maps, because if the input images contain some noise, the Laplacian will have some too. This might even cause problems for the fusion of input images.

Now that we have computed those weight maps, we will see how to use them in order to obtained the exposure fused, high quality, low dynamic range image.

Figure 1: Four photos of different exposures and their associated image of weight. White pixel in W_i means that the corresponding pixel in I_i will be used in the result.

3 Fusion

3.1 Naive method

The first naive idea would be to compute a convex combination of the N input images, $I_1 ... I_N$, with weigh maps $W_1 ... W_N$ as coefficients. To do so we first need to normalize W in order to obtain new normalized maps $\hat{W}$:

$$\hat{W}_k[i,j] = \left(\sum_{l=1}^{N} W_l[i,j] \right)^{-1} W_k[i,j]$$

Where $W_k[i,j]$ is the weight of the pixel at position (i,j) in the k-th image. Our fusion R is then defined by the following weighted sum:

$$R[i,j] = \sum_{k=1}^{N} \hat{W}_k[i,j] I_k[i,j] \tag{1}$$

Now let's see what the result looks like at Figure 2. The result is clearly unsatisfactory. Indeed as we said in 2.2 our images of weight have rough edges and contain some noise. A first solution that one could imagine is to smooth the weight maps.

Figure 2: Naive Fusion

3.2 Smoothing weight maps

In order to erase noise and remove sharp transitions in weight maps, using a Gaussian filter on the maps before computing R with (1) seems to be a good idea. A first interesting observation is that the noise disappear, but the quality of the image remains to be desired. Halos appears around edges, unwanted transition are created (flat surface doesn't seems to be flat anymore). In fact the weight maps may want two objects side by side to be taken in two different images, so smoothing will widen the border between the two objects in the weight maps.

A simple solution to erase these halos is to use a cross bilateral filter [4] so as not to mix the weights between two objects side by side. This filter works like the bilateral filter but uses a second images instead of the filtered image to ponder the weights. Given a neighborhood Ω_p of a pixel p, we define the new map W' as :

$$W'_k[p] = \sum_{q \in \Omega_p} f(\|q-p\|) g\left(\|I_k[q] - I_k[p]\|\right) W_k[q]$$

We use two centered gaussians for f and g. In this way, a gaussian blur is computed by taking only the pixels that have a similar color to the pixel p in the image I_k. In our experiments we use a standard deviation of 5 for f and a standard deviation of 0.18 for g (the colors being in $[0.1]^3$).

The results obtained are presented in the Figure 3. We effectively observe the disappearing of halos with the cross-bilateral filter, but other problems remain, such as the unwanted transitions. Furthermore in our implementation cross bilateral filter is very slow but it is certainly due to the fact that we implemented in Python, using loops (that are very slow).

Gaussian filter Cross-bilateral filter

Figure 3: Fusion with smoothing filters. A gaussian one on the right and a cross-bilateral one on the left

3.3 Laplacian pyramids

Since the simple smoothing of our weight maps does not work, we used more elaborate image fusion techniques than a simple weighted sum. Several previous works deal with this problem of images merging with masks. They use other representations of images. A very efficient technique is Poisson editing [5] which represents images by their gradient. We can manipulate the gradient and then reconstruct the new image solving a Poisson equation. In our case we could merge the gradients of the input images before reconstruction. But this method can be costly in terms of computing time.

A second, older solution seems to be a better candidate. We could use Laplacian pyramids as proposed in the section *Image Merging* of the paper Pyramid-based computer graphics [6] in which they merge seamlessly an image of apple and an image of orange. As they explained, when we only smooth the weight maps, two types of errors can occur along the edges of weight maps. If the size of the gaussian is too small, transitions will still be visible (just a little blured). This is due to a large difference between low frequencies around this edge. But if the size of the gaussian is too large, there will be undesirable transition zones due to mismatch in high frequencies. We can't avoid both defects at the same time. Then, we should just use large gaussian for low frequencies and small gaussian for high frequencies.

The idea is to construct a Laplacian pyramid $L\{I_k\}^0, ..., L\{I_k\}^{n-1}$ for each input image I_k. This way $L\{I_k\}^0$ will contain the higher frequency information in the image and $L\{I_k\}^{n-1}$ the lower frequency information. Then we construct a Gaussian pyramid $G\{\hat{W}_k\}^0, ..., G\{\hat{W}_k\}^{n-1}$ for each weight maps. As $L\{I_k\}^l$ will be weighted with $G\{\hat{W}_k\}^l$, the low frequencies are actually more smoothed than the high ones, as it was suggested in the previous paragraph. The Laplacian pyramid is then defined in a manner similar to equation (1):

$$\forall l = 0, ..., n-1 \quad L\{R\}^l[i,j] = \sum_{k=1}^{N} G\{\hat{W}_k\}^l[i,j] L\{I_k\}^l[i,j]$$

R is then rebuilt thanks to this pyramid. The result for the room images used so far is given in Figure 4. The image obtained is very pleasant to look at. The previous defects have totally disappeared.

Figure 4: Final result using Laplacian pyramids

4 Tests on our own photos

We tried the Laplacian pytamid method on our own photos. If the results is good for every set of images we tried it on, we used the adaptability of the model to finetune its paremeters the parameters of the model (w_c, w_s, w_e) in order to make every image look even better.

The first set of photo:

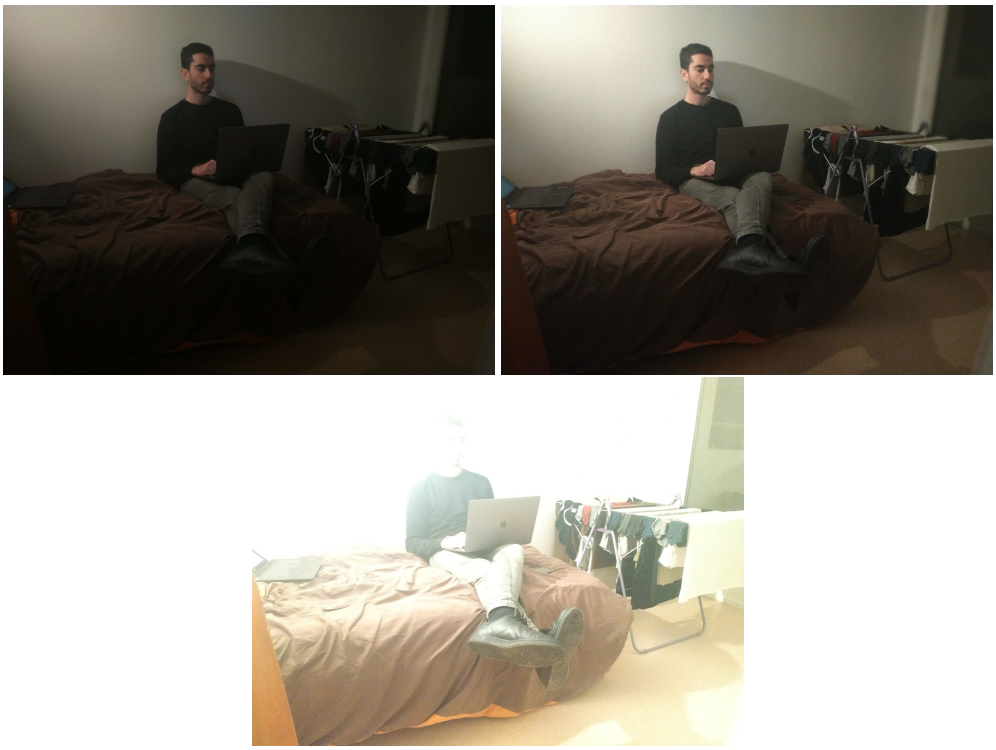

Figure 5: 'Beautiful Raphaël in the ENS Cachan campus' dataset

The obtained result the result with the default parameters $w_c = w_s = w_e = 1$ is good (see Figure 6 - no halos, no artifacts, edges are neats), but the colors seems pretty fade. In particular, the sweet wore by Raphaël seems to be dark grey, while it is, in fact, black.

Figure 6: With $w_c = w_s = w_e = 1$

This can be seen as either a lack of contrast or a too low exposition. To correct this, we tried a lot of parameters. In particular we increased the value of w_c and w_e relatively to w_s, in order to increase their relative importance in the phase of computation of the weight maps.

The best results (it term of our own perception) was obtained with $w_c = 50, w_s = 1, w_e = 20$:

Figure 7: With $w_c = 50, w_s = 1, w_e = 20$.
See in particular the color of the sweet and the color difference shoulder/wall which is more contrasted.

We made another test on personal photos, of a orange lamp on a desk, in front of a clearer orange band painted on the wall (see Figure 8).

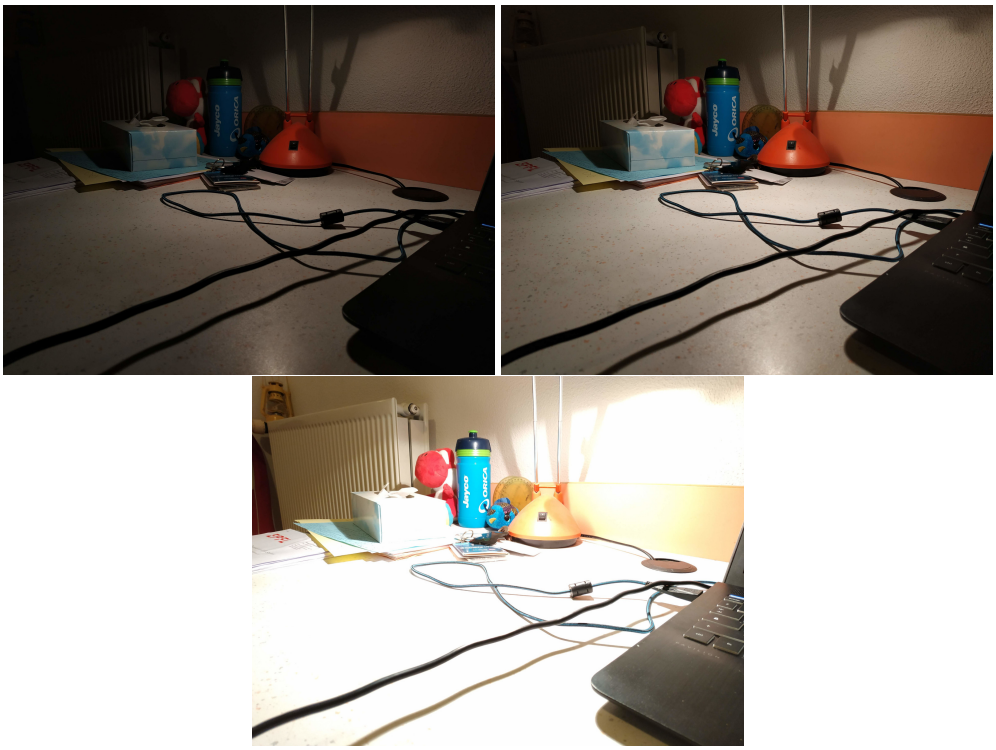

Figure 8: 'Beautiful desk in the ENS Cachan campus' dataset

Just as before, the colors seems to fade a little, with the orange of the lamp looking just as orange as the wall (see Figure 9).

Figure 9: With $w_c = w_s = w_e = 1$

This time, the exposition seems to be the problem. We indeed find the best result (in term of our own perception) to be for the parameters $w_c = 1, w_s = 1, w_e = 40$:

Figure 10: With $w_c = 1, w_s = 1, w_e = 40$.
See in particular the color of the lamp compared to the background orange, way more bright this time

This parts highlights a strength of this model: its adaptability. Indeed, we notice here that it is possible to finetune the model in order to even more optimize the perceived quality of the fused image.

Nonetheless, we can still raise two limits. First, the fact that we have to fine-tune those parameters for every image sets. Indeed, as every photo is different in term of light, exposition, saturation, contrast, those parameters are very dependant of the original set of images we have to fuse. Indirectly, this method hence depends on the characteristics of the camera, and of the the scene and conditions of the original photos. And second, even after fine-tuning, the results can be 'good' technically but the way we perceive it might not follow.

A good example is on the 'Beautiful Raphaël in the ENS Cachan campus': even if technically the best photo is the one generated with fine-tuned parameters (no halo, no shades too dark, everything on the scene seems clear), we prefer the second original photo of the set, whose colors are closer to reality (same sweet color, same bed color) and a more beautiful and coherent lighting (in the generated photo, there's a strong shadow on the wall on the right of Raphaël, while the light source seems not to be that bright).

5 Conclusion

As a conclusion, this article proposed a method to fuse images of the exact same scene with different exposure in order to obtain a new high definition, low dynamic range with a good exposure image, without to have to convert into HDR first. This merging works in two simple steps : (i) a weight map is computed for each image, based on simple quality measures (contrast, saturation and exposeness), (ii) from the weight maps and the original images, the fused imaged is generated.

As seen in part 3, different method of fusion were tried. The most efficient, and by far on the example as its the only method giving results looking 'real' and 'good' $i.e.$ without artifacts, halos or blurs, is the one based on Laplacian pyramids.

A strength of this method is its adaptability: if a fused image is not satisfactory, in term of contrast, saturation or exposeness, modifying the coefficient of the weight maps component measure might fix this problem. It's also very easy to add a new quality measure, by just changing the computation of the weight map.

Nonetheless, this fine-tuning having to be hand-made everytime, this method lose its interest. Moreover, the color might be impacted by the exposure fusion algorithm, as the examples shows us, and worst: the whole coherence of the image might suffer of it. In particular -as seen in examples- the shadows on the fused image might not be coherent with the apparent lighting.

An interesting future research might be to quantify the coherence of the photo, by establishing a metric able to catch this information, in order to automatically fine-tune the weight parameters. Another interesting research might be trying to do exposure fusion using convolutional neural networks. If such a neural network is well designed, it might be able to take into account the coherence issue without having to fine-tune the model for every input set of images.

References

[1] Tom Mertens, Jan Kautz, and Frank Van Reeth. "Exposure Fusion". In: *Proceedings of the Pacific Conference on Computer Graphics and Applications, Pacific Graphics 2007, Maui, Hawaii, USA, October 29 - November 2, 2007*. 2007, pp. 382–390. DOI: 10.1109/PG.2007.17. URL: https://doi.org/10.1109/PG.2007.17.

[2] Paul E. Debevec and Jitendra Malik. "Recovering high dynamic range radiance maps from photographs". In: *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1997, Los Angeles, CA, USA, August 3-8, 1997*. 1997, pp. 369–378. DOI: 10.1145/258734.258884. URL: https://doi.org/10.1145/258734.258884.

[3] Erik Reinhard et al. *High Dynamic Range Imaging - Acquisition, Display, and Image-Based Lighting (2. ed.)* Academic Press, 2010. ISBN: 978-0-12-374914-7. URL: http://www.elsevierdirect.com/product.jsp?isbn=9780123749147%5C&dmnum=CWS1.

[4] Elmar Eisemann and Frédo Durand. "Flash photography enhancement via intrinsic relighting". In: *ACM Trans. Graph.* 23.3 (2004), pp. 673–678. DOI: 10.1145/1015706.1015778. URL: https://doi.org/10.1145/1015706.1015778.

[5] Patrick Pérez, Michel Gangnet, and Andrew Blake. "Poisson image editing". In: *ACM Trans. Graph.* 22.3 (2003), pp. 313–318. DOI: 10.1145/882262.882269. URL: https://doi.org/10.1145/882262.882269.

[6] J. M. Ogden et al. *Pyramid-Based Computer Graphics*. 1985.