Telecom Paris

Project IMA201

Increasing Depth of Field by Image Fusion

Elaborated by:

Kais ZAIRI

Haithem DAGHMOURA

Supervised by:

Saïd LADJAL

Academic year :
2023/2024

Abstract

In this project, we have implemented the guided fusion method from scratch using the steps described in the related paper [1] that suggests increasing the depth of field by combining several images acquired with different focus settings. The main idea is that at each point the information from the sharpest image at that point is retained without creating artefacts. To do this, we decomposed the image into two layers: detail and coarse content. This decomposition is carried out using anisotropic filtering. Then we explored an other use of this method for images with different exposures. The results of the test were consistent with those of the papers' [1] [2] and gave the desired effect of enhancing the depth of field by image by fusion. However, we noticed some of the limitation of this method such as the problems that occurs when the two images has a moving object in the scene or the problem that occurs from the zoom caused by a change of focus that we noticed in the data we took using our own pictures. Afterwards, we propose some solutions for this problems like cropping and searching for the zoom factor. Finally, we explored the fast guided filter method that enables a faster computation for larger images.

Contents

List of Figures

Introduction

Depth of field is an essential parameter in photography. It determines the range of distances over which objects appear sharp in an image. In portrait photography, a shallower depth of field is generally desirable to bring the subject into focus, while in landscape photography, a greater depth of field is required to capture all the details of the scene.

Depth of field is mainly influenced by three factors: the aperture of the optical system, its focal length, and the distance between the camera and the object being photographed.

$$DOF = \frac{2u^2Nc}{f^2} \tag{1}$$

where:

- u: distance of the object
- N: f-Number
- c: circle of confusion
- f: focal length

Increasing depth of field can be achieved using sophisticated cameras, but it is also possible to opt for a computerized approach. This involves merging several images taken with different depths of field into a single composite image that offers more complete information and encompasses all the desired details. Several methods can be employed to achieve this image fusion, but one of the most robust and relevant techniques is image fusion using guided filters. In this report, we will describe the guided filtering method and image fusion using guided filters. We will also criticize this method and provide some possible improvements.

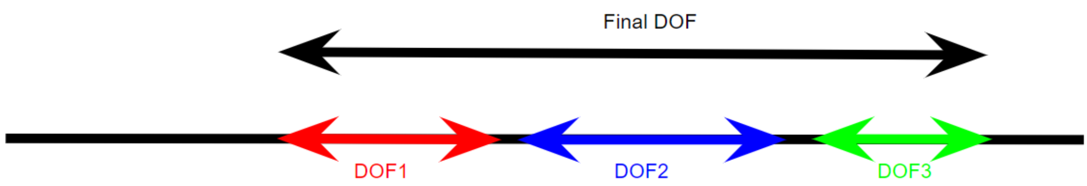


Figure 1: Depth of field augmentation by image fusion

1 Guided Filter

Guided filtering is a technique used to filter images. The result of filtering is locally a linear transformation of the guided image.[1] In addition, this filter has the property of edge-preserving smoothing. However, this is a more generic concept that can be applied to other applications beyond "smoothing".

1.1 Theory

The key assumption of the guided filter is a local linear model between the guidance I and the filter output q. We assume that q is a linear transform of I in a window w_k centered at the pixel k:

$$q_i = a_k I_i + b_k, \forall i \in w_k \tag{2}$$

where (a_k, b_k) are some linear coefficients assumed to be constant in w_k. We use a square window of a radius r. This local linear model ensures that q has an edge only if I has an edge, because $\Delta q = a\Delta I$.

To determine the linear coefficients, we seek a solution to (2) that minimizes the difference between q and the filter input p. Specifically, we minimize the following cost function in the window:

$$E(a_k, b_k) = \sum_{i \in w_k} ((a_k I_i + b_k - p_i)^2 + \epsilon a_k^2) \tag{3}$$

Here ϵ is a regularization parameter preventing a_k from being too large. The solution to (3) can be given by linear regression :

$$a_k = \frac{\frac{1}{|w|}\sum_{i \in w_k} I_i p_i - \mu_k \bar{p_k}}{\sigma_k^2 + \epsilon} \tag{4}$$

$$b_k = \bar{p_k} - a_k \mu_k \tag{5}$$

Here, μ_k and σ_k^2 are the mean and variance of I in w_k, $|w|$ is the number of pixels in w_k, and $\bar{p_k} = \frac{1}{|w|}\sum_{i \in w_k} p_i$ is the mean of p in w_k.
Next we apply the linear model to all local windows in the entire image. However, a pixel i is involved in all the windows w_k that contain i, so the value of q_i in (2) is not the same when it is computed in different windows. The strategy adopted is to average all the possible values of q_i. So after computing (a_k, b_k) for all patches w_k in the image, we compute the filter output by:

$$q_i = \frac{1}{|w|} \sum_{k : i \in w_k} (a_k I_i + b_k) = \bar{a_i} I_i + \bar{b_i} \tag{6}$$

where $\bar{a_i} = \frac{1}{|w|}\sum_{k \in w_i} a_k$ and $\bar{b_i} = \frac{1}{|w|}\sum_{k \in w_i} b_k$ With this modification Δq is no longer scaling of ΔI, because the linear coefficients $bara_i, \bar{b_i})$ vary spatially. But since $(\bar{a_i}, \bar{b_i})$ are the output of an average filter, their gradients should be much smaller than that of I near strong edges. In this situation we can still have $\Delta q \approx \bar{a}\Delta I$, meaning that abrupt intensity changes in I can be mostly maintained in q. This algorithm is also extended to the case of RGB color guidance images. Filtering using color guidance images is necessary when the edges or details are not discriminable in any single channel. To generalize to a color guidance image, the local linear model (2) as:

$$q_i = a_k^T I_i + b_k, \forall i \in w_k \tag{7}$$

Here I_i is a 3×1 color vector, a_k is a 3×1 coefficient vector, q_i and b_k are scalars. The guided filter for color guidance images becomes:

$$a_k = (\Sigma_k + \epsilon U)^{-1} \frac{1}{|w|} \sum_{i \in w_k} I_i p_i - \mu_k \bar{p_k} \tag{8}$$

$$b_k = \bar{p_k} - a_k^T \mu_k \tag{9}$$

$$q_i = \bar{a_k}^T I_i + \bar{b_k}, \tag{10}$$

Here Σ_k is the 3×3 covariance matrix of I in w_k, and U is a 3×3 identity matrix.

1.2 Resultats et discussion

1.2.1 Expriments

The first experiment consisted in reproducing the results of the artical named "Guided Image Filtering" for gray-scale image as guidance images. The idea is to guide the image with itself in order to get the same edges while smoothing with different kerenl sizes and epislon values.

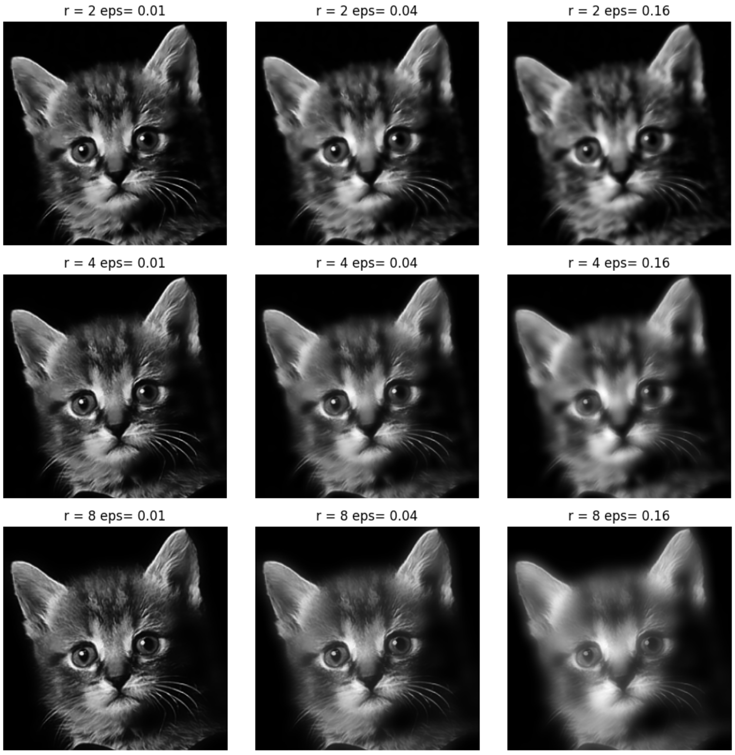


Figure 2: Guided gray image with itself with different parameters [7]

Then we experimented The case of filtering RGB color image with itself

Figure 3: Guided color image with itself with different parameters [8]

1.2.2 Discussion

Through the examples above we can conclude that the guided image filtering technique, when using as input (P) and guidance (I) the same image, offers the benefit of edge-preserving smoothing by leveraging information from the guidance image to control the filtering process. Unlike traditional smoothing techniques, which treat all image regions equally, guided filtering takes into account the local characteristics and structures within the image and this is especially clear for the case of the cat and Lena with r=8 and eps=0.04. In these two images we can see that the overall image remains the same and the defining edges of the cat and Lena are still there. In addition, we can notice the smoothing effect in the image.
In fact, this can be explained intuitively. If we consider the case of these examples (where $I = P$). We remark that if $\epsilon = 0$, then the solution to (3) is $a_k = 1$ and $b_k = 0$. If $\epsilon > 0$, we can consider two cases:
Case 1: "Flat patch". If the image I is constant in w_k, then (3) is solved by $a_k = 0$ and $b_k = \bar{p_k}$;
Case 2: "High variance". If the image I changes a lot within w_k, then a_k becomes close to 1 while b_k is close to 0. When a_k and b_k are averaged to get $\bar{a_i}$ and $\bar{b_i}$, combined in (7) to get the output, we have that if a pixel is in the middle of a "high variance" area, then its value is unchanged, whereas if it is in the middle of a "flat patch" area, its value becomes the average of the pixels nearby. More specifically, the criterion of a "flat patch" or a "high variance" is given by the parameter ϵ . The patches with variance σ^2 much smaller than ϵ are smoothed, whereas those with variance much larger than ϵ are preserved. This parameter determine "what is an edge/a high variance patch that should be preserved".

As a result, the guided filter effectively smooths out noise and unwanted variations while preserving the sharpness of edges and boundaries. This edge-preserving property is particularly advantageous in applications where maintaining image details, such as fine textures, edges, or object boundaries, is crucial. Whether used for denoising, enhancing image quality, or image manipulation, guided filtering ensures that the important structural elements of an image remain intact, resulting in visually pleasing and visually accurate results.

2 Image fusion with guided filtering

Image fusion is a technique used to combine information from several images into a single, more informative image. It aims to improve image quality and extract important features, and is applied in a number of fields. [2]

2.1 Theory

The figure below shows a descriptive diagram of image fusion with guided filters.

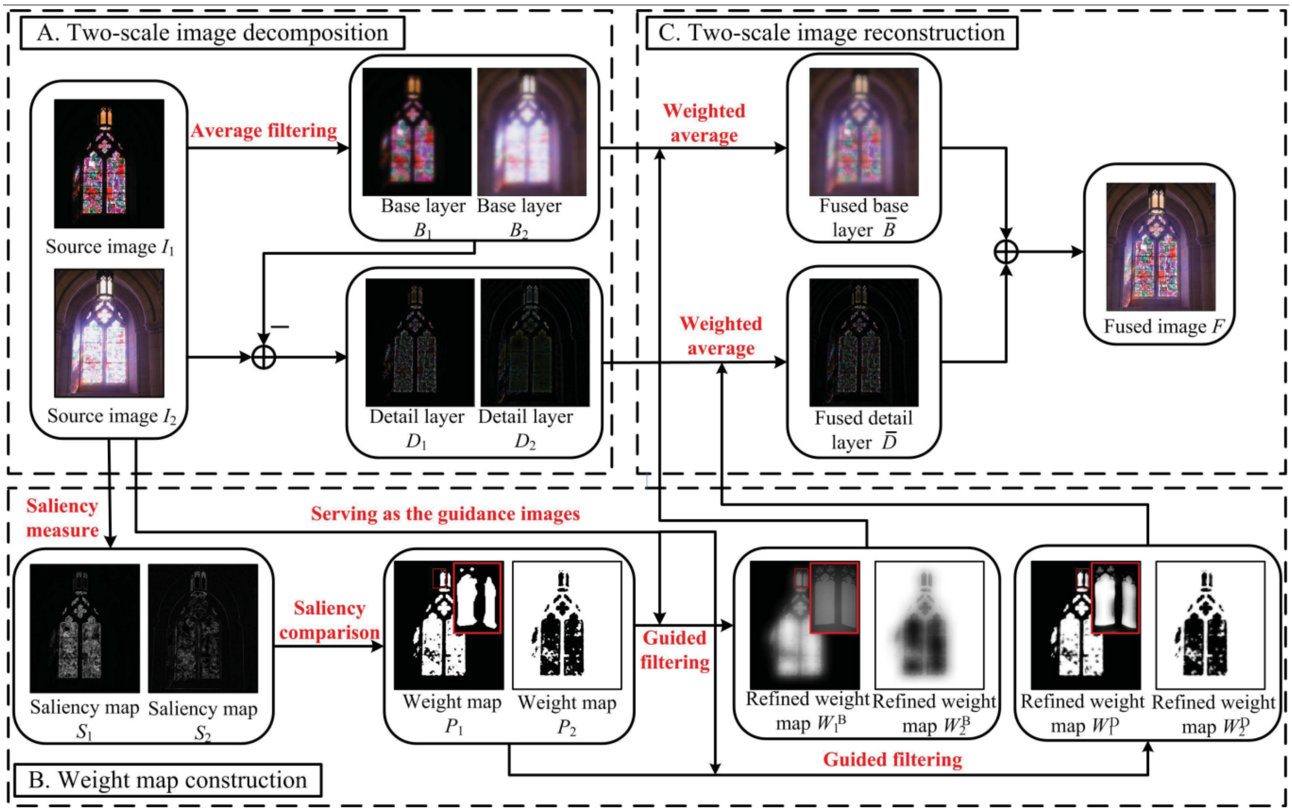


Figure 4: Schematic diagram of image fusion method based on guided filtering [2]

A. Two-Scale Image Decomposition

The first step consists of a frequency decomposition of the source images using an average filter Z into Base layers images and Detail layers images according to the following equations:

$$B_n = I_n * Z \tag{11}$$

$$D_n = I_n - B_n \tag{12}$$

where I_n is the n-th source image and Z is an average filter of size 31×31.

B. Weight Map Construction

As shown in Fig. 3, the weight map is constructed as follows.

- First, Laplacian filtering is applied to each source image to obtain the high-pass image H_n:

$$H_n = I_n * L \tag{13}$$

where L is a 3×3 Laplacian filter.
- Then, the local average of the absolute value of H_n is used to construct the Saliency maps S_n:

$$S_n = |H_n| * g_{r_g,\sigma_g} \tag{14}$$

where g is a Gaussian low-pass filter of size $(2r_g + 1) \times (2r_g + 1)$, and the parameters r_g and σ_g are set to 5.
- Next, the saliency maps are compared to determine the weight maps as follows:

$$P_n^k = \begin{cases} 1 \text{ if } S_n^k = max(S_1^k, ..., S_N^k) \\ 0 \text{ otherwise} \end{cases} \tag{15}$$

where N is number of source images, S_n^k is the saliency value of the pixel k in the nth image.
- Finally, guided image filtering is performed on each weight map P_n with the corresponding source image I_n serving as the guidance image.

$$W_n^B = G_{r_1,\epsilon_1}(P_n, I_n) \tag{16}$$

$$W_n^D = G_{r_2,\epsilon_2}(P_n, I_n) \tag{17}$$

where r_1, ϵ_1, r_2, and ϵ_2 are the parameters of the guided filter, W_n^B and W_n^D are the resulting weight maps of the base and detail layers.
Finally, the values of the N weight maps are normalized such that they sum to one at each pixel k.

C. Two-Scale Image Reconstruction Two-scale image reconstruction consists of the following equations:

$$\bar{B} = \sum_{n=1}^{N} W_n^B B_n \tag{18}$$

$$\bar{D} = \sum_{n=1}^{N} W_n^D D_n \tag{19}$$

Then, the fused image F is obtained by combining the fused base layer $\bar{B}$ and the fused detail layer $\bar{D}$:

$$F = \bar{B} + \bar{D} \tag{20}$$

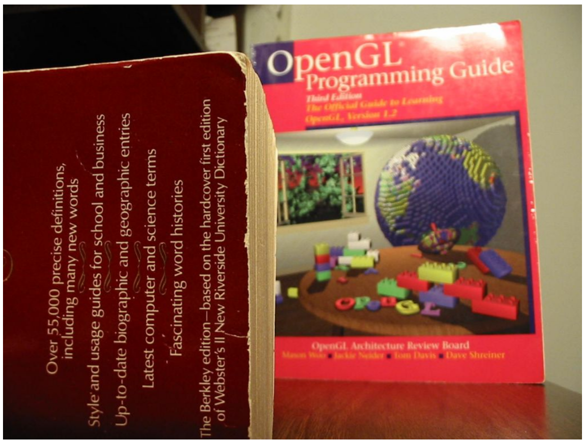
To sum up the above detailed approach, Guided Filter Fusion (GFF) involves the separation of each source image into a base layer that captures large-scale intensity variations and a detail layer containing small-scale details. A novel weight construction method is introduced to combine pixel saliency and spatial context for image fusion. Instead of relying on optimization-based methods, guided filtering is adopted as a local filtering technique for image fusion. The fusion process concludes with image construction through weighted averaging, giving higher weights to focus areas to preserve critical information.

2.2 Experiments and discussion

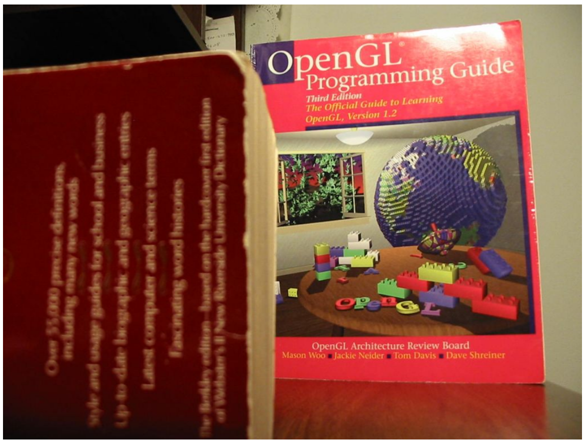
Our experiments were focused mainly on two types of application of guided fusion method: multi-focus fusion which is the main objective of our project and multi-exposure fusion for which the guided fusion method performs very well and was sited in the paper.

2.2.1 Multi-focus image fusion experiments

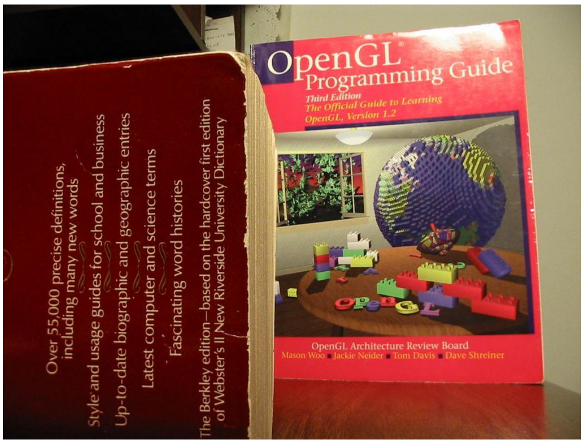
The following figures illustrate the image fusion performed on a multi-focus data set. This data set is available on GitHub and several other drives [3][5][4][6] that contain multiple images used in the paper.



(a) Image 1



(b) Image 2



(c) Fused image

Figure 5: Example 1 of image fusion in multi-focus dataset (two books)[5]

(a) Image 1

(b) Image 2

(c) Fused image

Figure 6: Example 2 of image fusion in multi-focus dataset (a flower and a tree) [5]

(a) Image 1

(b) Image 2

(c) Fused image

Figure 7: Example 3 of image fusion in multi-focus dataset (a doll and a painting) [5]

(a) Image 1

(b) Image 2

(c) Image 3

(d) Image 4

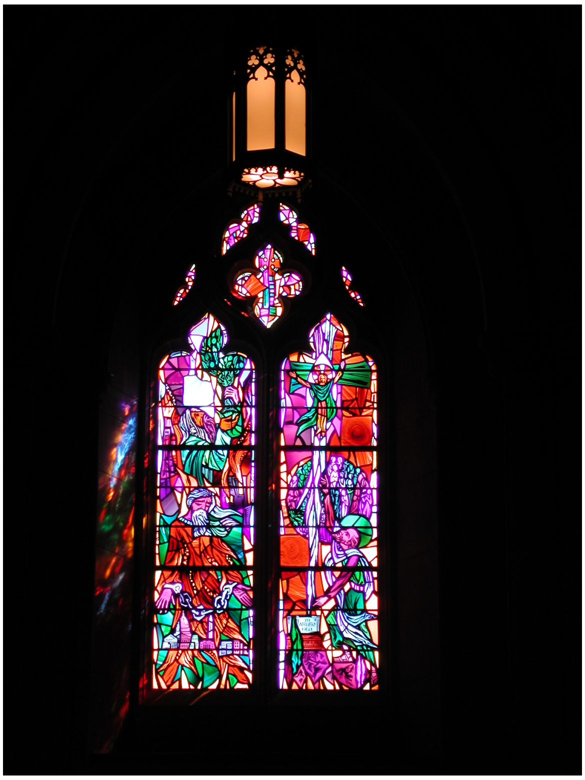
(e) Fused image

Figure 8: Example 4 of image fusion in multi-focus dataset (Télécom promo 2026)

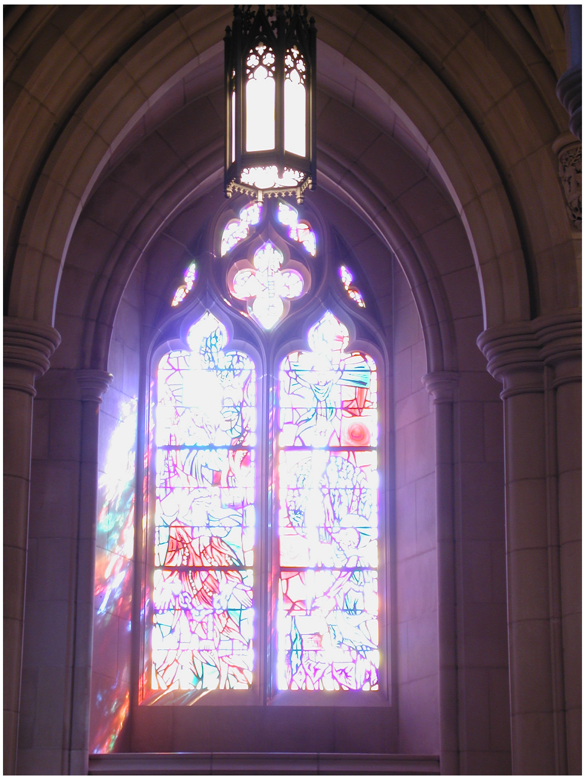
When applied to a dataset of multi-focus, Guided Filter Fusion produces high-quality, information-rich fused images. Notably, this method excels at preserving the features and details of the source images without introducing visible artifacts or brightness distortions. It also effectively preserves the focus areas of different source images without introducing artifacts. For example, this effect is visible in the book images. where we have two books in the image and only one of them has a clear writing on it. The result of the fusion gives us the two books with the best focus from the inputs. In this case the details were the letters which present high frequency zones and it was successfully passed to the result.

2.2.2 Multi-exposure fusion experiments

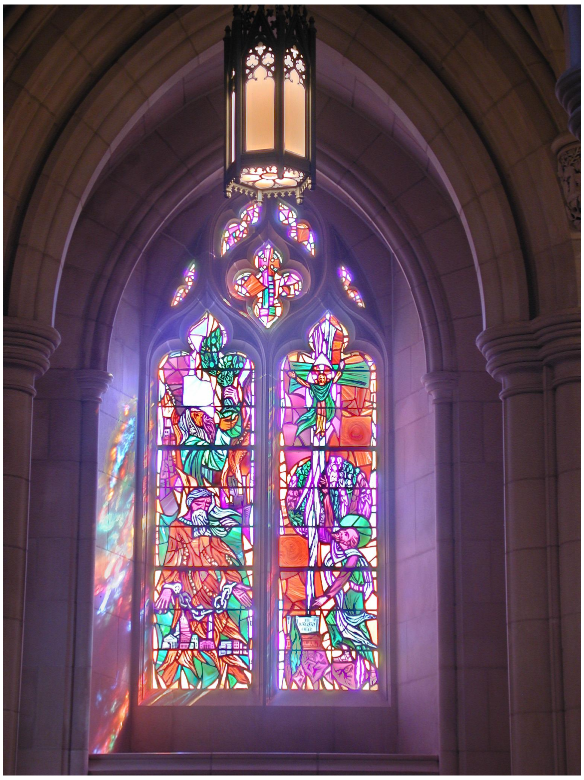
The following figures illustrate the image fusion performed on a multi-focus dataset.



(a) Image 1



(b) Image 2



(c) Fused image

Figure 9: Example 1 of image fusion in multi-exposure dataset (cathedral window) [3]

(a) Image 1

(b) Image 2

(c) Fused image

Figure 10: Example 2 of image fusion in multi-exposure dataset (waffle house)[6]

(a) Image 1

(b) Image 2

(c) Image 3

(d) Image 4

(e) Fused image

Figure 11: Example 3 of image fusion in multi-exposure dataset (library)[6]

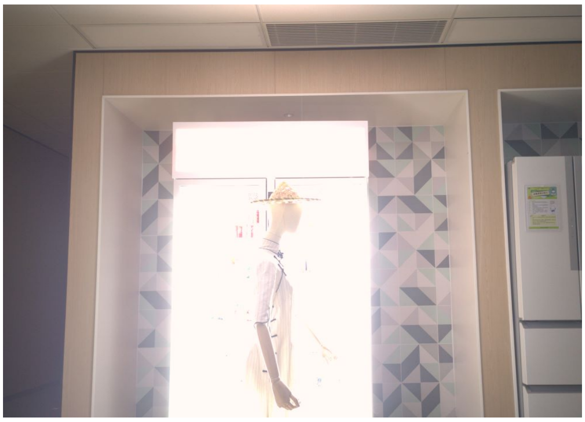
In a similar way to multi-focus dataset, Guided Filter Fusion produces high-quality and information-rich fused images with muti-exposure dataset. Notably, this method excels at preserving the features and details of the source images without introducing visible artifacts or brightness distortions. It also effectively preserves the focus areas of different source images without introducing artifacts. For example, the fused image of the cathedral window take all details of the from the first image and takes also all the details around the window from the second image.In the library example, we see that all the details are combined in the final image from inside and outside the room.

3 Limitations and enhancements

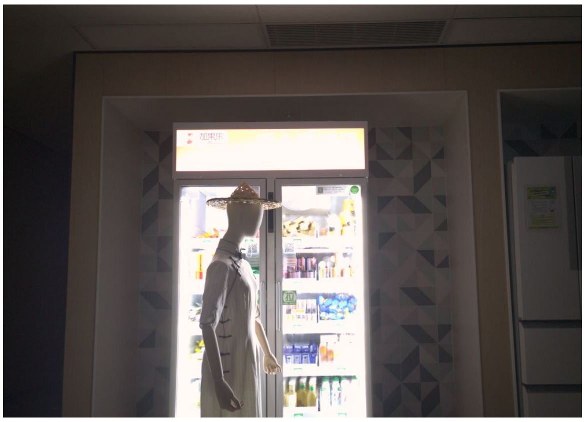
In the last section we showed that the guided fusion method performs well on several sets of images, however, in practical scenarios, achieving this ideal outcome can be challenging. Imperfections and inconsistencies within the inputs may lead to unwanted artifacts and blurring. Therefore, pre-processing steps are necessary to refine the dataset, especially when dealing with multi-focus images. We will explore these imperfection and our proposed amelioration in the next section.

3.1 Overlapping objects

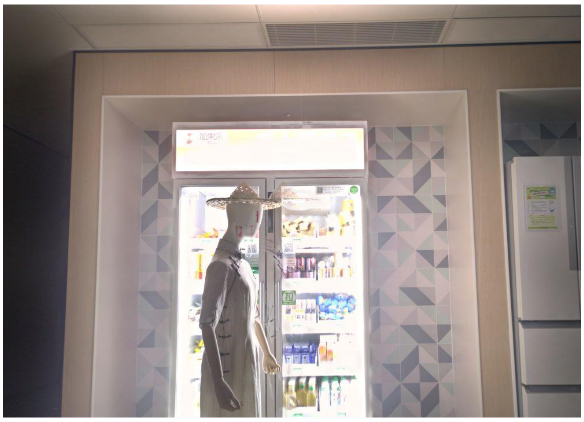
When experemienting with various datasets, we noticed fiew limitation to the fusion method. We noticed this limitation while trying to fuse the same scene but with an object having moved across both images.



(a) Image 1



(b) Image 2



(c) Fused image

Figure 12: GFF for multi-exposure images of a model [6]

In this images, the input consists of multi-exposure images with a model that has moved in the images. The fusion technique gives a good result in the outer space and we can see the contents of the vending machine and the details in the wall as well. However in the model case we see an overlapping of objects. For instance if we focus on the head we see some objects appearing. This is due to the fact that this method considers both the model and the red object to be parts of the picture. In fact, if we focus on this part in the input images we see that effectively the first input image contains only the red object in that part and in the second image we remark tht the head of the model is the one present. So the fusion can't decide throught the composition and decomposition which one contains the information to keep since both of them gives us 'details'. So the result will be a weighted average of both and thus the overalapping of these objects.

In situations involving movement, the guided filter assigns higher weights to critical areas within the scene. However, a challenge arises when there is an overlap of weights, making it evident when we set a certain threshold greater than 1 for the summation of these weights. To address this issue, we can mitigate the weights of one of the source images. The selection of which image to attenuate is a crucial decision. One approach is to introduce a metric that accumulates the weights within a predefined square window, for instance, a 50-pixel window. By doing so, we can then reduce the weights in the area with the lowest cumulative sum, effectively resolving the issue of weight overlap.

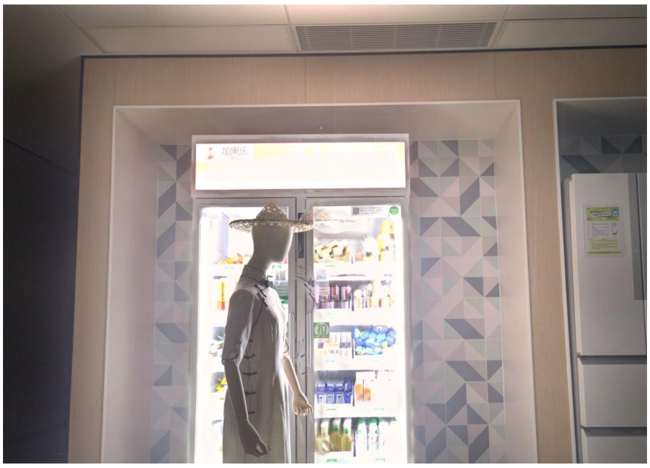
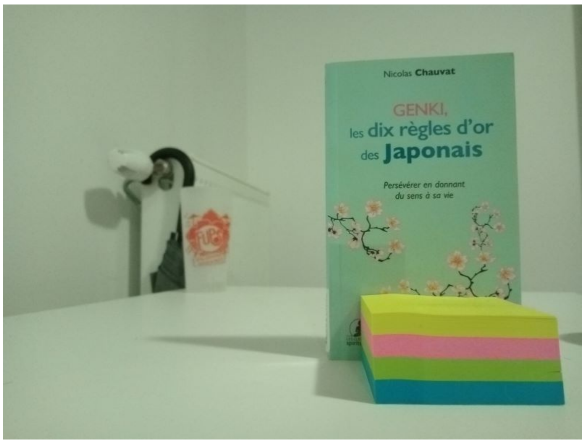


Figure 13: Result of GFF after weights reducing

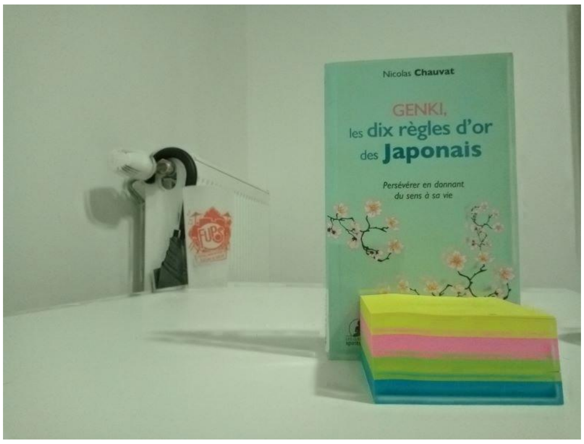
3.2 Focus zoom

In order to put this method into practice, for example if we want to implement the guided filter fusion method in our smartphones. There is an option in smartphones to change the focus, i.e., to keep the scene in focus so as to concentrate on one part of the scene. For the different focuses, we'll take photos that focus on different parts of the scene and then merge these images using our proposed method.



(a) Image 1

(b) Image 2



(c) Fused image

Figure 14: Example 1 of GFF on images taken by our smartphone

(a) Image 1

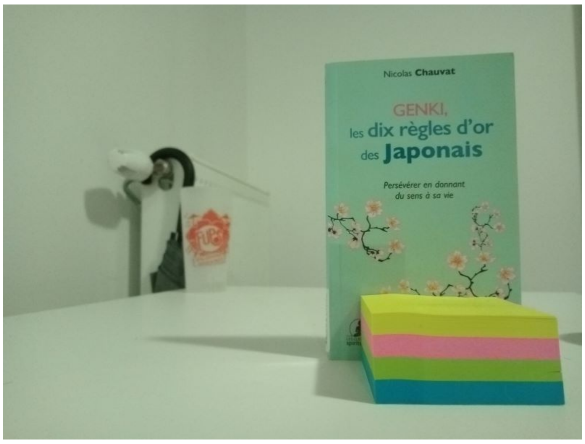
(b) Image 2

(c) Fused image

Figure 15: Example 2 of GFF on images taken by our smartphone

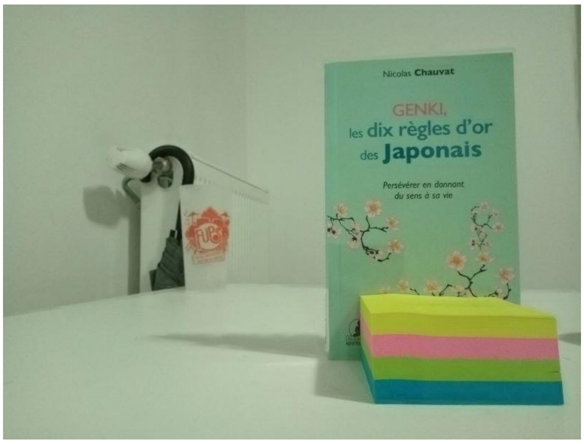
We note that there are artifacts in the fused image. It's important to mention that the fusion method works very well, but the images provided by the different focuses are not sharp, there is a focus zoom between the images. The problem is therefore to prepare the images correctly and transform them into states similar to multi-focus or multi-exposure dataset.

We can now assume that the two images have two parallel planes, so that there is a zoom factor f that allows us to move from the first image to the second. The focus option in smartphones only takes discrete values, and the same applies to possible zoom factors. By evaluating a few zoom factors, for example between 1 and 1.2 in steps of 0.01 or 0.02, we calculate the similarity index between the original image and the zoomed image, and select the image with the best index and the corresponding zoom factor.



(a) Image 1

(b) Image 2



(c) Fused image

Figure 16: GFF after finding the best zoom factor in example 1

(a) Image 1

(b) Image 2

(c) Fused image

Figure 17: GFF after finding the best zoom factor in example 2

We can now see that artifacts have all mostly disappeared, and so the guided filter fusion remains a good choice that can be implemented on our smartphones.
NB: In practice, it is not necessary to find the right zoom factor every time. Since the parameters have not changed, we can associate the right zoom factor with each focus. Even if we have some parameters of the optical system in the smartphone, we can determine the right zoom factor by simple geometric formulas.

3.3 Fast guided filter

To improve the efficiency of the guided filter, which is particularly useful when processing large images or when real-time performance is required, we make an assumption about the local linearity of the linear coefficients (a_k and b_k) within the filter. These coefficients are associated with each pixel "k" in the image. Leveraging the power of downscaling and upscaling techniques, the algorithm efficiently interpolates the coefficients of neighboring pixels, eliminating the need for separate filtering operations. This interpolation process is facilitated by a user-defined parameter "s", in which we subsample the input image and the guided image by a factor of 1/s before applying the guided filter to these subsampled images. To restore the output image to its original dimensions, we use interpolation, with bilinear interpolation being the recommended choice. In theory, this approach reduces computation time by a factor of 1/s. The choice of 's' becomes a compromise between algorithm speed and preservation of essential information. In practice, 's' is generally set at values such as 'r/4' or 's=r' to balance computational efficiency and fidelity to original image quality.

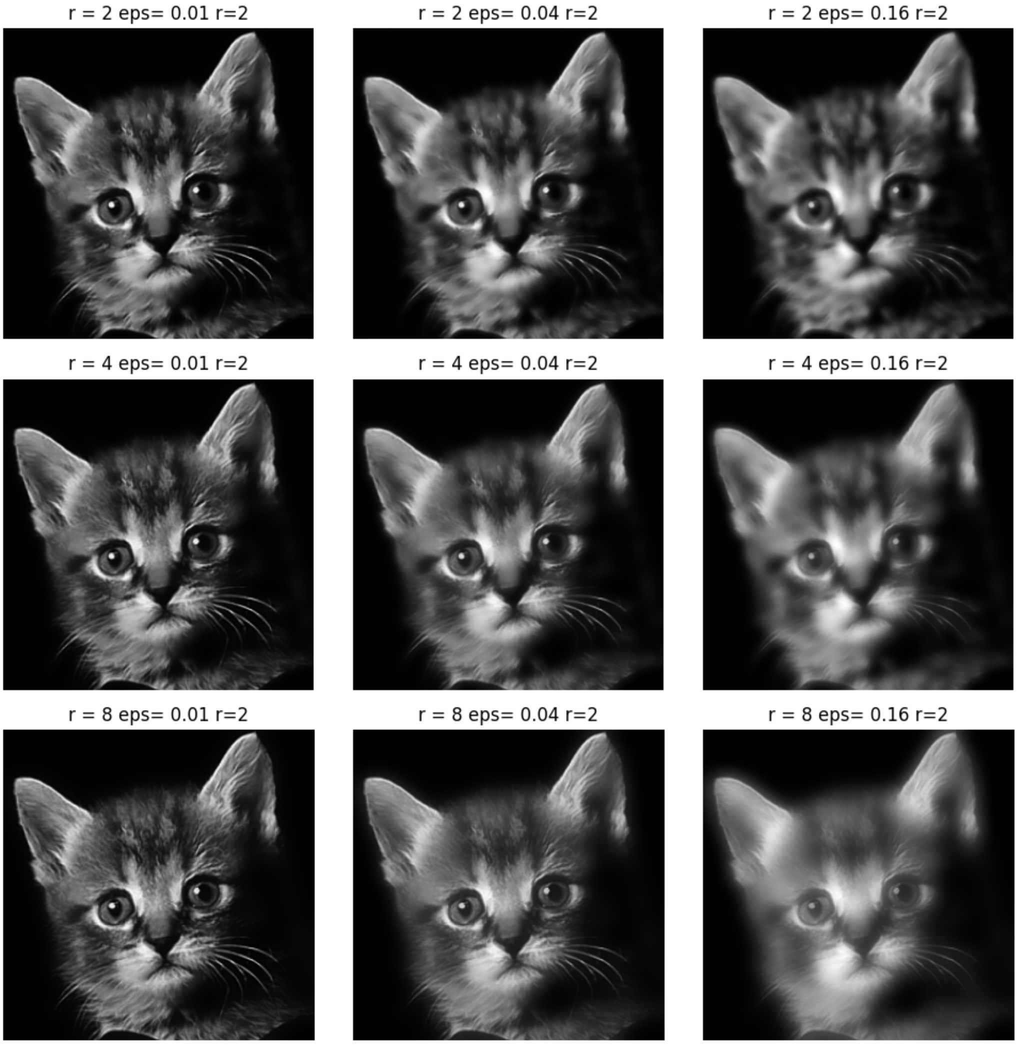


Figure 18: Fast guided gray image with itself with different parameters [7]

Figure 19: Fast guided color image with itself with different parameters [8]

Comparing these results with those obtained with my original guided filter method, we find that they are the same and in a faster time.

Conclusion

We have presented the image fusion method based on guided filtering. This method utilizes the average filter to get the two-scale representations, which is simple and effective. Moreover the guided filter is used to make use of the correlations between neighberhood pixels for weight optimization. Experiments show that this method can well preserve the original and complementary information of multiple input images in order to increase the depth of focus and make use of multiple exposure timing in order to get more details in the resulting image. Furthermore, the proposed method is computationally efficient, making it quite qualified for real applications. At last, we showed some limitations of the fusion method and how to improve the performance of this technique throught cropping, adapting the zoom between two images and using fast guided filter in order to speed it.

References

[1] K. He, J. Sun, and X. Tang, “Guided image filtering,” in Proc. Eur. Conf. Comput. Vis., Heraklion, Greece, Sep. 2010,

[2] Shutao Li, Xudong Kang, and Jianwen Hu "Image Fusion with Guided Filtering" in IEEE Transactions on image processing, Vol. 22, No. 7, JULY 2013

[3] https://drive.google.com/drive/folders/0BzXT0LnoyRqlRDJRZ0VnUG9pUk0?resourcekey=0-Ofj0Njyzo4muONUHdedSqA

[4] https://drive.google.com/drive/folders/0BzXT0LnoyRqlVjht0EhiUzU5a2M?resourcekey=0-qV1X6ELUBEoYLK_Eva-Qog

[5] https://www.researchgate.net/publication/273000238_multi-focus_image_dataset

[6] https://github.com/HaoZhang1018/MFI-WHU/blob/master/MFI-WHU.rar

[7] https://github.com/lisabug/guided-filter/blob/master/data/cat.png?raw=true

[8] https://raw.githubusercontent.com/lisabug/guided-filter/master/data/Lenna.png