Project: Image Fusion with Guided Filtering

Amélie PESNEC AMELIE.PESNEC@ELEVES.ENPC.FR
Ecole des Ponts Paristech

Theïlo TERRISSE THEILO.TERRISSE@ELEVES.ENPC.FR
Ecole des Ponts Paristech

1. Introduction

The problem of fusing multiple images arises in several applications:

1. **Multi-focus images:** due to the limited depth-of-field of cameras, getting a detailed view of an entire scene might necessitate taking several photographs focused on different elements of the scene;
2. **Multi-exposure images (HDR):** to deal with the limited dynamic range of cameras, one option to picture scenes involving regions of various illuminations is to use various exposition times;
3. **Multi-source images:** different acquisition systems may reveal different details of an object, such as in medical imaging where computed tomography and magnetic resonance imaging respectively reveal hard and soft tissues of an organ.

In all of these instances, each image brings details for certain regions of the scene but not for others, and we wish to extract these details and combine them into a fused picture. The literature comprises multiple methods to tackle this problem, such as stationary wavelet transform (SWT) which uses wavelet transform to decompose images into different frequency components and then fuses the coefficients. Another existing algorithm is Laplacian pyramid (LAP) which decomposes the images into different levels of details.

This project focuses on the article Image Fusion with Guided Filtering [6] published in 2013 which proposes a new method for image fusion. This method extracts salient features from each input image, compares them and smoothes the comparison using Guided Filtering [4]; the resulting weight maps are used to fuse base and detail layers of the input pictures, and the resulting layers are finally recomposed.

In our work, we re-implemented the method proposed in the article, and we analyzed its performance. We tested it on various sets of images and tried to highlight the limitations of the method. We also evaluated the impact of the different hyperparameters in order to understand their role.

In section 2, we explain the proposed method, recall the different steps of the algorithm and the equations involved, as well as clarify the intuitions behind the method. In section 3, we show the results obtained with our re-implementation of the method and display some artifacts that appeared in some results. We also produced several experiments on sets of two or more input images, to emphasize the impacts of the different parameters. Finally, we conclude in Section 4. The code for this project is available at https://github.com/TheiloT/MVA-Image-Fusion-with-Guided-Filtering.

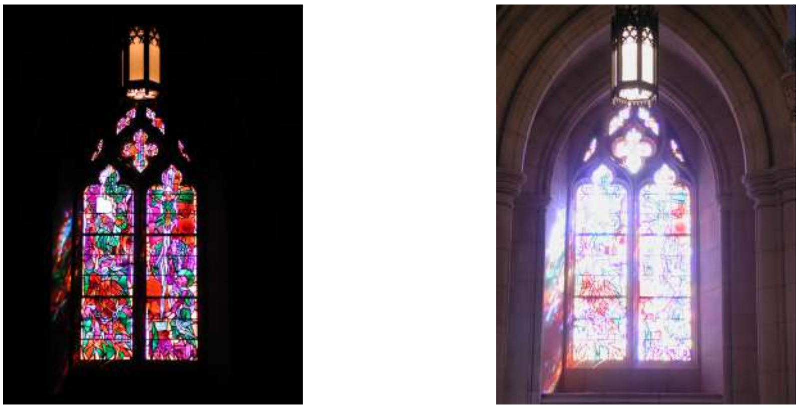
2. Method and implementation

The idea behind image fusion can be summarized as follows: if $N \in \mathbb{N}$ images I_n are given as input, if $W \in \mathbb{N}$ is their width and $H \in \mathbb{N}$ their height, the main idea of image fusion is to select for each pixel position $(i, j) \in [\![1, H]\!] \times [\![1, W]\!]$ the pixel $P_n^{(i,j)}$ from the image $n \in N$ which has the most detail in this position. The fused image is composed of all such selected pixels.

Now, this definition requires disambiguating the notion of detail in a given image region. In [6], the authors resort to the *saliency map*, defined in Section 2.2. One may expect to extract the important pixels from each image by comparing these saliency maps; however, raw saliency map comparisons cannot be used directly, as they are often noisy and misaligned with the structure of the scene (*e.g.* object boundaries): we need to enforce *spatial consistency.* To this aim, instead of resorting to costly regularization-based optimization techniques, the authors make use of an anisotropic filter: the *Guided Filter* [4], which blurs the saliency map comparisons while respecting the edges of objects in the original images.

One last issue encountered in this process is that some image regions may have different dimensions and levels of details; therefore, we need a way to apply different smoothings to different detail levels. In [6], this is done by using two guided filters with different parameters to derive two sets of weight maps from the saliency map comparisons. These weight maps are applied on base and detail layers of the input images to get fused base and detail layers, and the final fused image is obtained by summing these up.

In this section we detail each step as well as choices made for its re-implementation, and use the same pair of stained-glass window images used by the authors for illustration (Figure 1). Before that, we start with a refresher on the guided filter.



(a) Image 1

(b) Image 2

Figure 1: Stained-glass windows

2.1 Guided filter

Image filters are used in many image processing tasks, for instance when denoising images by blurring out noise in uniform areas, while preserving object edges. In this case, an anisotropic, edge-preserving filter is required, like the bilateral filter [3]. More generally, guided filters allow to filter out details of an image while selectively extracting base information.

The *Guided Filter* [4] used by the authors of [6] takes as input an image P to filter and a guide image I and outputs an image O. The idea behind guided filtering is that O can be modeled as

$$O = P - n \tag{1}$$

where n is the detail (noise, texture...) of image P. O is then searched as an approximate linear transform of I, with the idea that ∇O will be proportional to ∇I and thus will share the same edges. More precisely, the linear coefficients are determined on local patches of size $(2r+1) \times (2r+1)$ with r the radius of the patch. For a patch ω_k centered in pixel k,

$$O_{/\omega_k} = a_k I_{/\omega_k} + b_k \tag{2}$$

where (a_k, b_k) are constant on ω_k. These coefficients are chosen by minimizing the convex energy

$$E(a_k, b_k) = \sum_{i \in \omega_k} ((a_k I_i + b_k - P_i)^2 + \varepsilon a_k^2). \tag{3}$$

In this cost function, ε is a regularizing factor that sets the level of variance that has to be filtered out: if $\delta_k << \varepsilon$, then $a_k \approx 0$ and $b_k \approx \mu_k$, resulting in a uniform patch; and if $\delta_k >> \varepsilon$, then $a_k \approx 1$ and $b_k \approx 0$, thus preserving edges. The solution to the minimization of (3) has the following analytical expression:

$$\begin{cases} a_k = \frac{\frac{1}{|\omega|}\sum_{i \in \omega_k} I_i P_i - \mu_k \bar{P}_k}{\delta_k + \varepsilon} \\ b_k = \bar{P}_k - a_k \mu_k \end{cases} \tag{4}$$

where μ_k and δ_k are the mean and variance of I in ω_k and $\bar{P}_k = \frac{1}{|\omega|}\sum_{i \in \omega_k} P_i$ is the mean of P in ω_k.

To get the final output, for each pixel i, the coefficient to apply is the average of coefficients for all k such that $i \in \omega_k$:

$$O_i = \bar{a}_i I_i + \bar{b}_i \tag{5}$$

where $\bar{a}_i = \frac{1}{|\omega|}\sum_{k\in\omega_i} a_k$ and $\bar{b}_i = \frac{1}{|\omega|}\sum_{k\in\omega_i} b_k$. As a result, ∇O will not be proportional to ∇I on patches anymore as $(\bar{a}_i, \bar{b}_i)$ varies spatially, but assuming small gradients for these coefficients allows to consider that $\nabla O \approx \bar{a}\nabla I$ locally.

In the case where the input image P is colored, the filtering can be performed as above on each separate channel. However, if the guide is colored, $\mathbf{a}_k$ and $\mathbf{I}_i$ become 3−vectors in (2) and (3), so that:

$$\begin{cases} \boldsymbol{a}_k = (\Sigma_k + \varepsilon J)^{-1}\left(\frac{1}{|\omega|}\sum_{i\in\omega_k} \boldsymbol{I}_i P_i - \boldsymbol{\mu}_k \bar{P}_k\right) \\ b_k = \bar{P}_k - \boldsymbol{a}_k \boldsymbol{\mu}_k \end{cases} \tag{6}$$

where Σ_k is the 3×3 covariance matrix of $\boldsymbol{I}$ in ω_k and J is the 3×3 identity matrix. Finally:

$$O_i = \bar{\boldsymbol{a}}_i^T \boldsymbol{I}_i + \bar{b}_i. \tag{7}$$

To implement this filter, we started from a code delivered by Julie Delon, Lucía Bouza and Joan Alexis Glaunès [2], which uses integral images to speed computations when applying average filters in (4) and (5). We then adapted it for coloured guides, which slows down computations due to the construction of the covariance matrices on each window ω_k in (6). As a preliminary study of coloured guides, in appendix A we apply Guided Filtering to an artificial denoising problem, in which we apply random Gaussian noise to a painting of Kandinsky (Figure 18) and denoise it using Guided Filtering with $r = 20$ and $\varepsilon = 10^{-3}$. For the guide, we use either the original coloured painting (Figure 19c), its grayscale version (Figure 19a), or we simply filter one colour channel at at time, each time using the corresponding channel of the original painting as guide (Figure 19b). Note that while the grayscale guide clearly results in degraded colours, both the coloured Guided Filter and the per-channel filter bring good results in this instance.

2.2 Weight map construction

The saliency maps used by the authors are obtained by applying a Laplacian filter L of size 3×3 on the image to extract the high-frequencies of the picture:

$$\forall n \in \mathbb{N},\ H_n = I_n * L. \tag{8}$$

This high-pass filter is then locally averaged using a Gaussian low-pass filter of radius 5 and standard deviation 5 to get the saliency maps:

$$\forall n \in \mathbb{N},\ S_n = |H_n| * G. \tag{9}$$

This map quantifies the amount of details in the image, such as the sharpness of edges. In our implementation, we used Laplacian and Gaussian filters available in *OpenCV*. As Laplcian filters are known to be sensitive to noise, we also apply a prior Gaussian filter of radius 1 and standard deviation 3 prior to the application of L. The resulting saliency for the stained glass windows maps are illustrated on Figure 2.

These saliency maps are then compared between all images to get raw weight maps P_n indicating for each image n the pixels for which it has the highest saliency. That is, for all pixel k:

$$P_n^k = \begin{cases} 1 \text{ if } S_n^k = \max\left(S_1^k, S_2^k, \ldots, S_N^k\right) \\ 0 \text{ otherwise} \end{cases} \tag{10}$$

The resulting raw weight maps are illustrated on Figure 2.

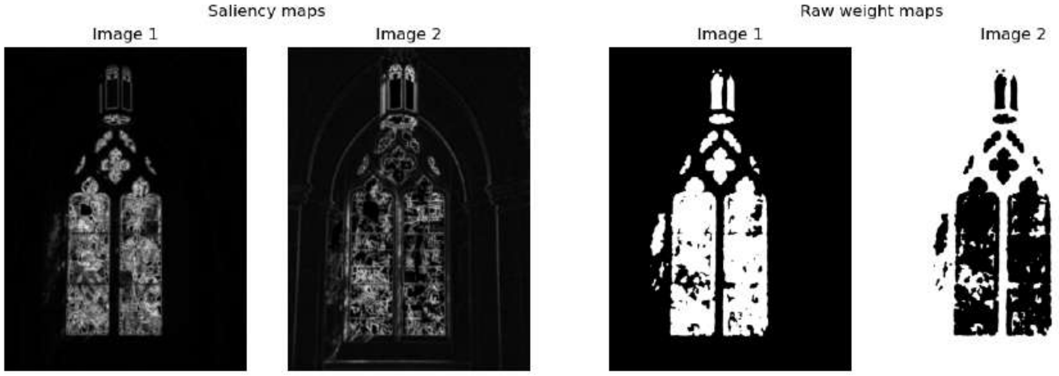


Figure 2: Saliency and raw weight maps

To produce the final weight maps, two guided filters are used to smooth the raw weight maps at two levels of details:

$$W_n^B = G_{r_1,\varepsilon_1}(P_n, I_n) \quad \text{and} \quad W_n^D = G_{r_2,\varepsilon_2}(P_n, I_n) \tag{11}$$

where $r_1,\ \varepsilon_1,\ r_2$ and ε_2 are the parameters of the Guided Filters, set by default to 45, 0.3, 7 and 10^{-6} respectively. The role played by these parameters and how to choose them is extensively discussed in Sections 3.2 and 3.3. However, as weight maps W_n^B and W_n^D will be used to fuse base and detail layers of the input images respectively, we can already give the following intuition: r_1 and ε_1 should be taken rather large so as to blur out small details, while r_2 and ε_2 should be taken smaller, so as to retain finer variations in the weight map. W_n^B and W_n^D are finally normalized over all N images so that their values sum to 1 at each pixel k. For the stained glass windows, the final weight maps are presented in Figure 3.

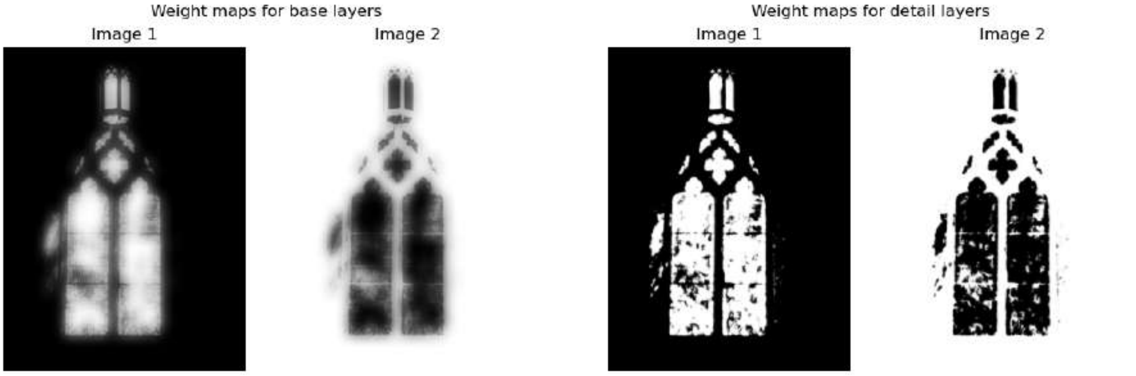


Figure 3: Weight masks for base and detail layers

2.3 Images decomposition and fusion

All input images are decomposed into base and detail layers using an average filter Z with a kernel of size 31×31:

$$\forall n \in [\![1, N]\!], \quad B_n = I_n \star Z \quad \text{and} \quad D_n = I_n - B_n \tag{12}$$

Intuitively, B_n contains large-scale features of the image, while D_n contains low-scale details. These layers are illustrated on Figure 4. Using the weight maps obtained in Section 2.2, we can combine these layers into a base and detail layer for the fused image:

$$\bar{B} = \sum_{n=1}^{N} W_n^B B_n \quad \text{and} \quad \bar{D} = \sum_{n=1}^{N} W_n^D D_n. \tag{13}$$

The final fused image is then easily built by recomposing these layers:

$$F = \bar{B} + \bar{D}. \tag{14}$$

The fused base and detail layers and the final result are presented on Figure 5.

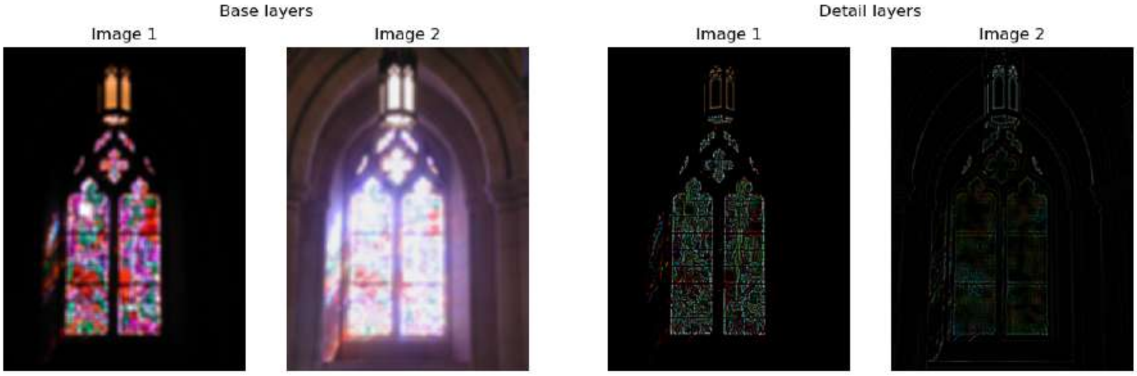


Figure 4: Base and detail layers

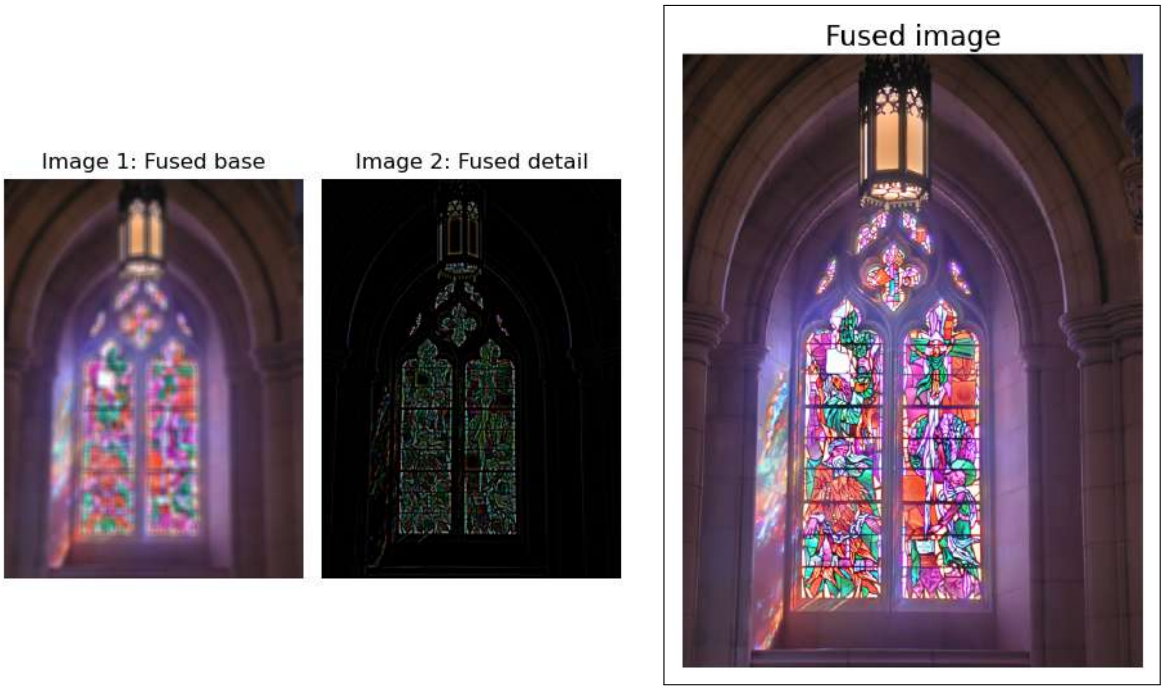


Figure 5: Fused layers and final fusion

3. Experiments

In this section, we perform several sets of experiments. In Section 3.1, we apply the method on a set of images from the Lytro dataset [5] composed of multi-focus grayscale and coloured pairs of images and make general comments on the behaviour of the method and on generated artifacts. An ablation is also performed on the decomposition of the image into base and detail layers. In Section 3.2, we make qualitative and quantitative in-depth analyses of the role of the parameters of the guided filters. Finally, in Section 3.3, we apply the method on a set of more than two images, for high-dynamic range imagery.

3.1 Overview

3.1.1 Testing the method on colored and grayscale images

A first test was carried out on the stained-glass windows of Figure 1, already used for illustration in Section 2. For this 2048×1536 image, our implementation takes about $32s$ to run, where the longest part is the computation of the weight maps which involves applying the Guided Filter. As the authors of [6] already delivered intermediate results of their algorithm for this pair of images, we could use it as a baseline to compare our implementation with. Therefore, we could demonstrate that the proposed method is relatively straightforward to implement and that the results are mostly reproducible. Note, however, that when zooming on some details, it seems that there are some color artifacts with our implementation, which are not present on the article's results. Figure 6 shows that the colors of the stained glass "fade" on the edge, an effect not observed on the original results. We suspect the difference notably comes from the decomposition step, as the base layers for the input images used in the article seem to differ slightly from ours; we could not find where our approaches differ.

Then, we tested the method on other images from the Lytro dataset [5] with standard parameters from the article: $r_1 = 45$, $\varepsilon_1 = 0.3$, $r_2 = 7$ and $\varepsilon_2 = 10^{-6}$. The method seems to be very effective for fusion of both colored and gray-scale multi-focus images, as shown in Figures 7a and 7b. To illustrate the application of the algorithm to various types of problems, we also marginally tested it on multi-source images. In Appendix B, we show three fusions of optic/infra-red pairs of images from the TNO dataset [7]. For the first pair of images, the algorithm manages to retain most of the details of the two persons from the IR image. However, it struggles to reveal the man in the doorway on the second example, and it only partially reconstructs the pales of the helicopter for the last example.

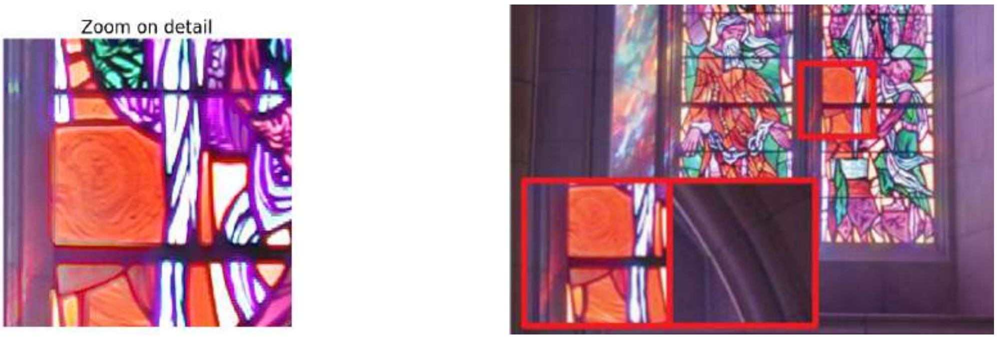
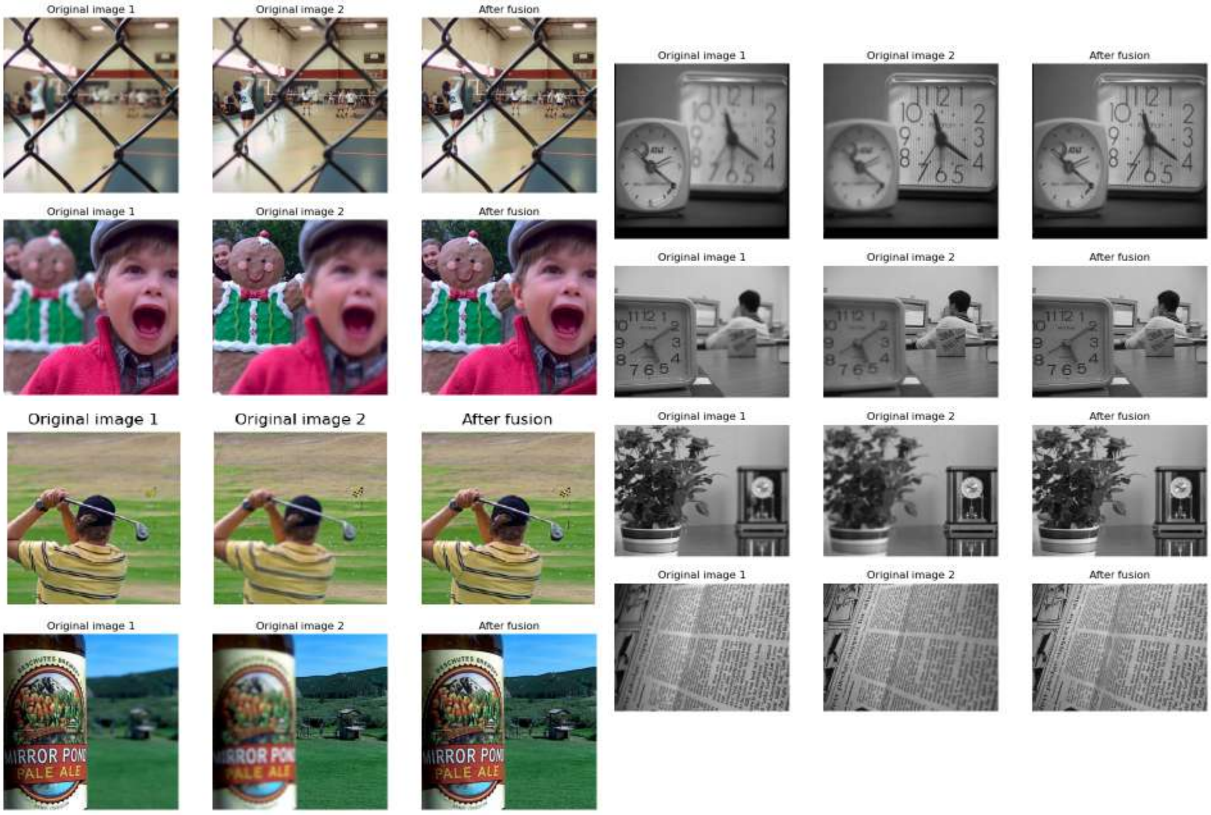


Figure 6: Zoom on our version and comparison with the article's result. Right figure is taken from [6]

3.1.2 Observations and artifacts

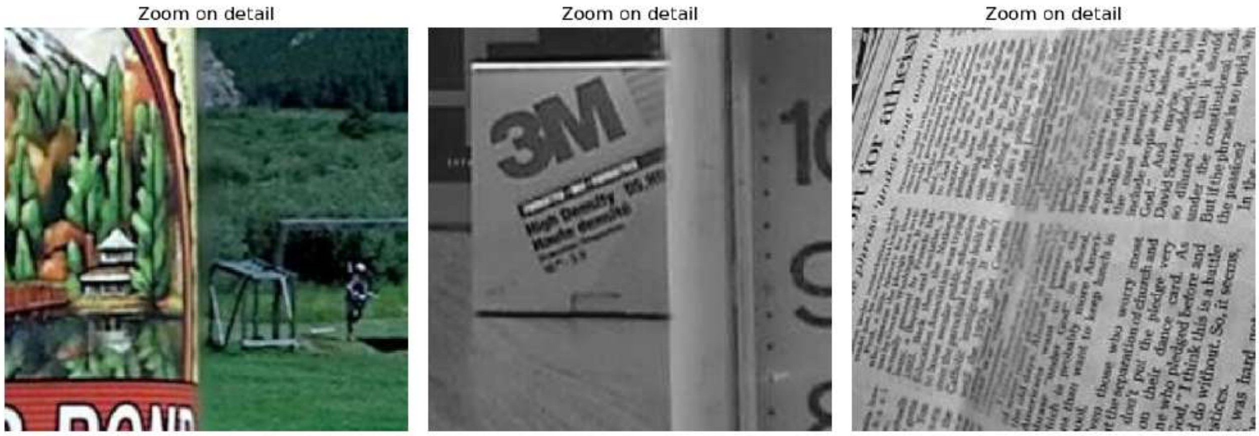
When testing on multiple images, we noticed that some artifacts can arise when the original images contain strong sharp edges that separate foreground and background elements. For instance, on Figure 8a, the close surrounding of the bottle remains blurry, whereas the rest of the image is sharp. This creates a blurred line which is quite visible when zooming on the image. We did not notice similar artifacts around sharp edges on the dataset of gray-scale images, see Figure 8b. Nevertheless, the fusion result of the newspaper images contains a blurry zone (Figure 8c), which correspond to the joining area between what used to be the foreground and the background. As the area was slightly blurry in both source images, it seems like the fusion algorithm struggles to reconstruct a sharp version.



(a) Results of fusion on multi-focus coloured images

(b) Results of fusion on multi-focus grayscale images

Figure 7: Sample results from the Lytro dataset



(a) Coloured guides (b) Gray-scale guides (c) Gray-scale guides

Figure 8: Zooming on fusion results

3.1.3 Ablation: fusion without base/detail decomposition

In principle, the fusion process with Guided Filtering can be carried out without decomposing the image into base and detail layers, as neither the computation of the saliency maps nor the application of the Guided Filters make use of these layers. To identify the importance of this decomposition, we carry out a simple ablation: starting from the raw weight maps obtained in (10), we only use one guided filter with parameters r and ε in (11) to get a single set of weight maps W. Then, we directly apply this weight map to the input images to get the final image:

$$F = \sum_{n=1}^{N} W_n I_n. \tag{15}$$

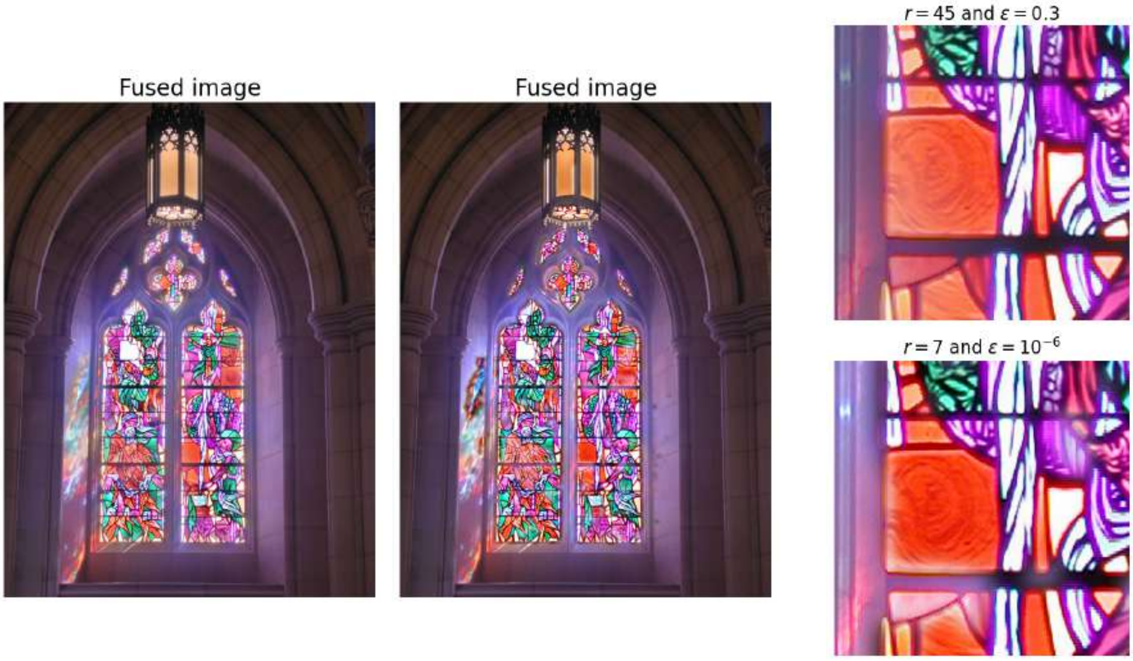
For the stained-glass windows, we use this ablated implementation with the parameters of Guided Filters 1 and 2 from (11) respectively on Figures 9a and 9b, along with zoom on details in Figure 9c. We observe that we already get quite satisfying results without decomposition and using the parameters of the large-scale Guided Filter $r = 45$ and $\varepsilon = 0.3$, and we also get less chromatic artifacts when compared with Figure 8. On the contrary, using the parameters of the small-scale filter $r = 7$ and $\varepsilon = 10^{-6}$ introduce multiple artifacts and a bad reconstruction. However, we cannot conclude to the superfluousness of the detail layer and associated filter, as the same ablation carried-out on a pair from the Lytro dataset leads to opposite conclusions, as illustrated on Figure 10 where the edge is sharper with the small-scale filter. Our conclusion is that while only one filter may be active at a time for a given set of images, the decomposition cannot be neglected as one filter or the other (or both) may be of importance depending on the nature of the details present in the set of images at hand.

3.2 Experiments on pairs of images

3.2.1 Impact of hyperparameters : Qualitative evaluation

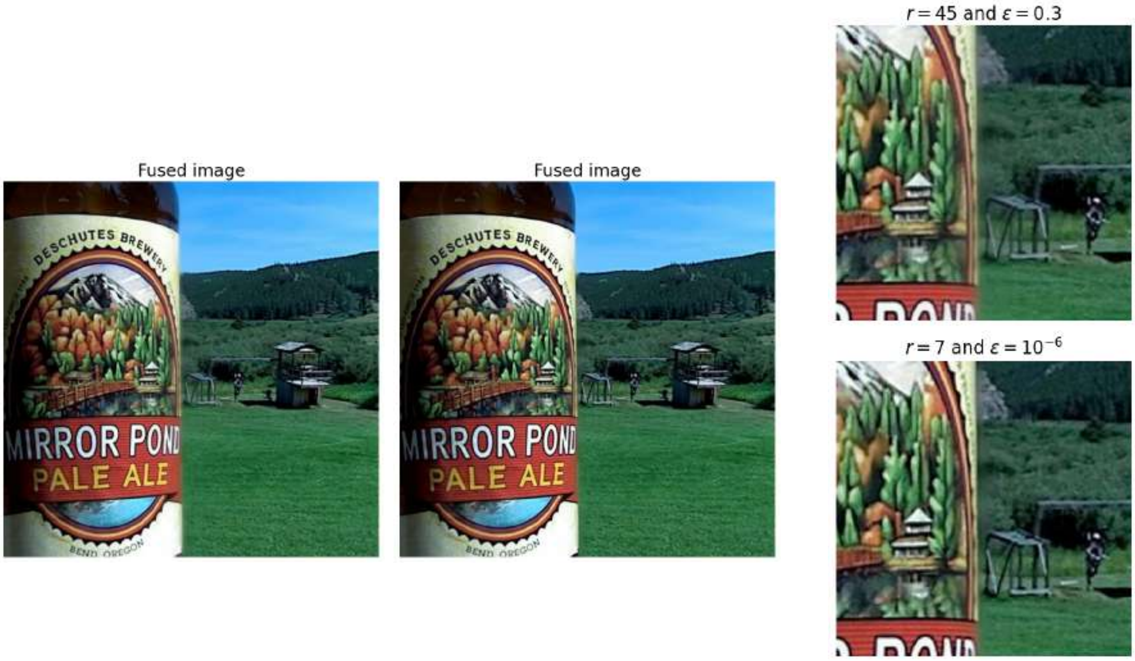
We studied the impacts of the hyperparameters r_1, ε_1, r_2 and ε_2 involved in the computation of the weight masks for both base and detail layers. As a first step, we qualitatively compared the fusion results obtained when changing one parameter to extreme values and keeping the other ones at their standard values, given in the article ($r_1 = 45$, $\varepsilon_1 = 0.3$, $r_2 = 7$ and $\varepsilon_2 = 10^{-6}$). For r_1 and r_2, the extreme values are 1 and 250. For ε_1, the tested values are 10^{-6} and 100, and for ε_2, they are 0.3 and 100. As displayed on Figure 11, there are very few visual impacts when changing the parameters, even when zooming on the images. With a very high r_2, we can see that the result becomes overall blurrier, but this is still a minor impact. This confirms the analysis of the article which states that "the [guided filtering based fusion method] does not depend much on the exact parameter choice" [6].

However, if we take a deeper look into the intermediate results of the algorithm, we can visually see the impacts of the parameters. Figure 12 shows the weight masks for different hyperparameter values,


Fused image
Fused image
$r = 45$ and $\varepsilon = 0.3$
$r = 7$ and $\varepsilon = 10^{-6}$

(a) Fusion for $r = 45$ and $\varepsilon = 0.3$ (b) Fusion for $r = 7$ and $\varepsilon = 10^{-6}$ (c) Zoom on details

Figure 9: Ablation: no decomposition into base and detail layers.


Fused image
Fused image
$r = 45$ and $\varepsilon = 0.3$
$r = 7$ and $\varepsilon = 10^{-6}$

(a) Fusion for $r = 45$ and $\varepsilon = 0.3$ (b) Fusion for $r = 7$ and $\varepsilon = 10^{-6}$ (c) Zoom on details

Figure 10: Ablation: no decomposition into base and detail layers.

as well as for the standard values. As r_1 and ε_1 have an effect on the weight mask of the base layer, we only display the latter. Likewise, for r_2 and ε_2, we only display the weight mask of the detail layer.

Figures 12a and 12b show that the weight mask of the base layer focuses on pixels that are in sharp areas and that the weights of adjacent pixels are close. This allows the weight mask of the base layer to create a spatially smooth fused base. On the contrary, the weight mask of the detail layers is aligned along the edges, which allows to have a neat fused detail image.

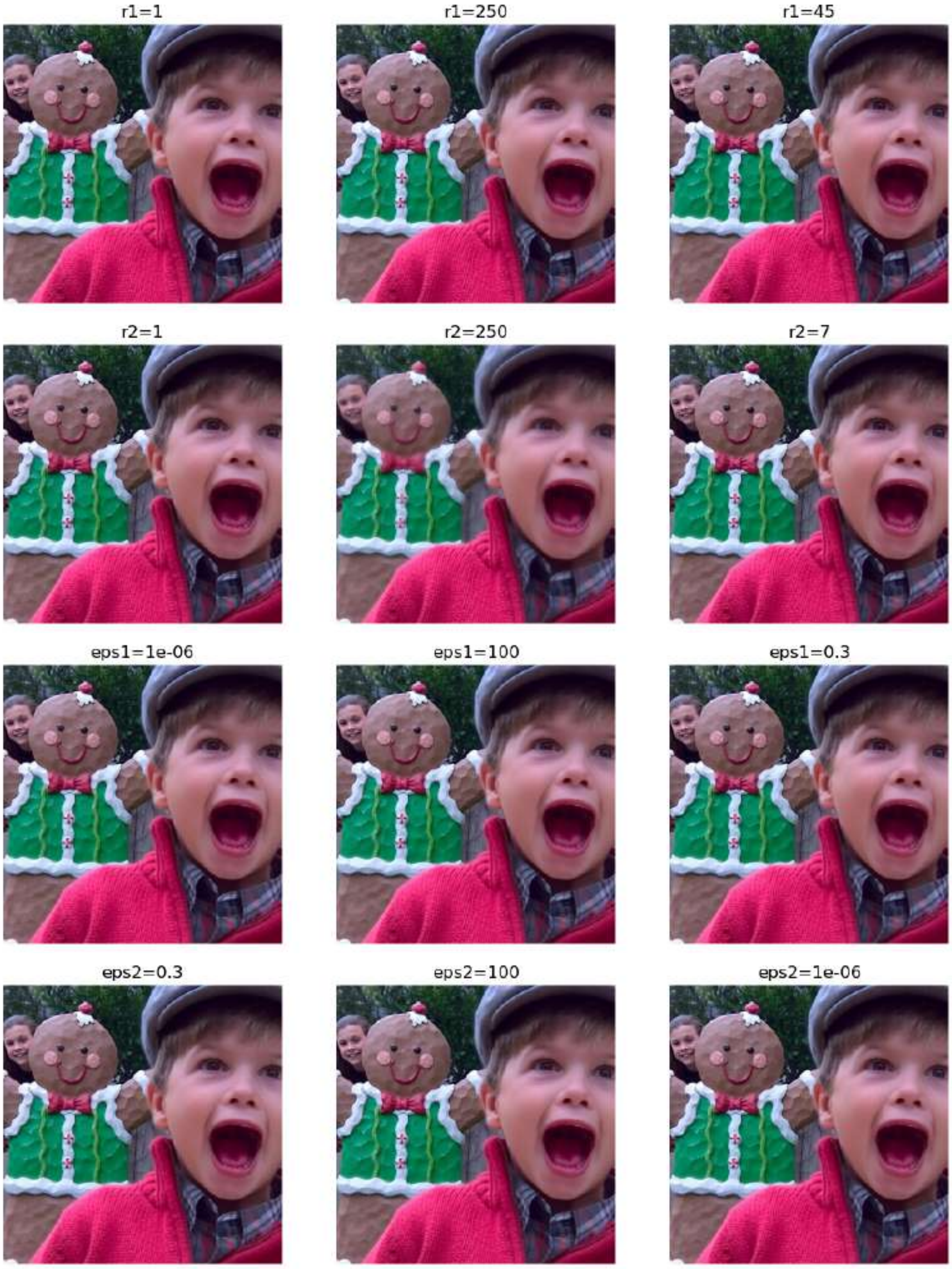
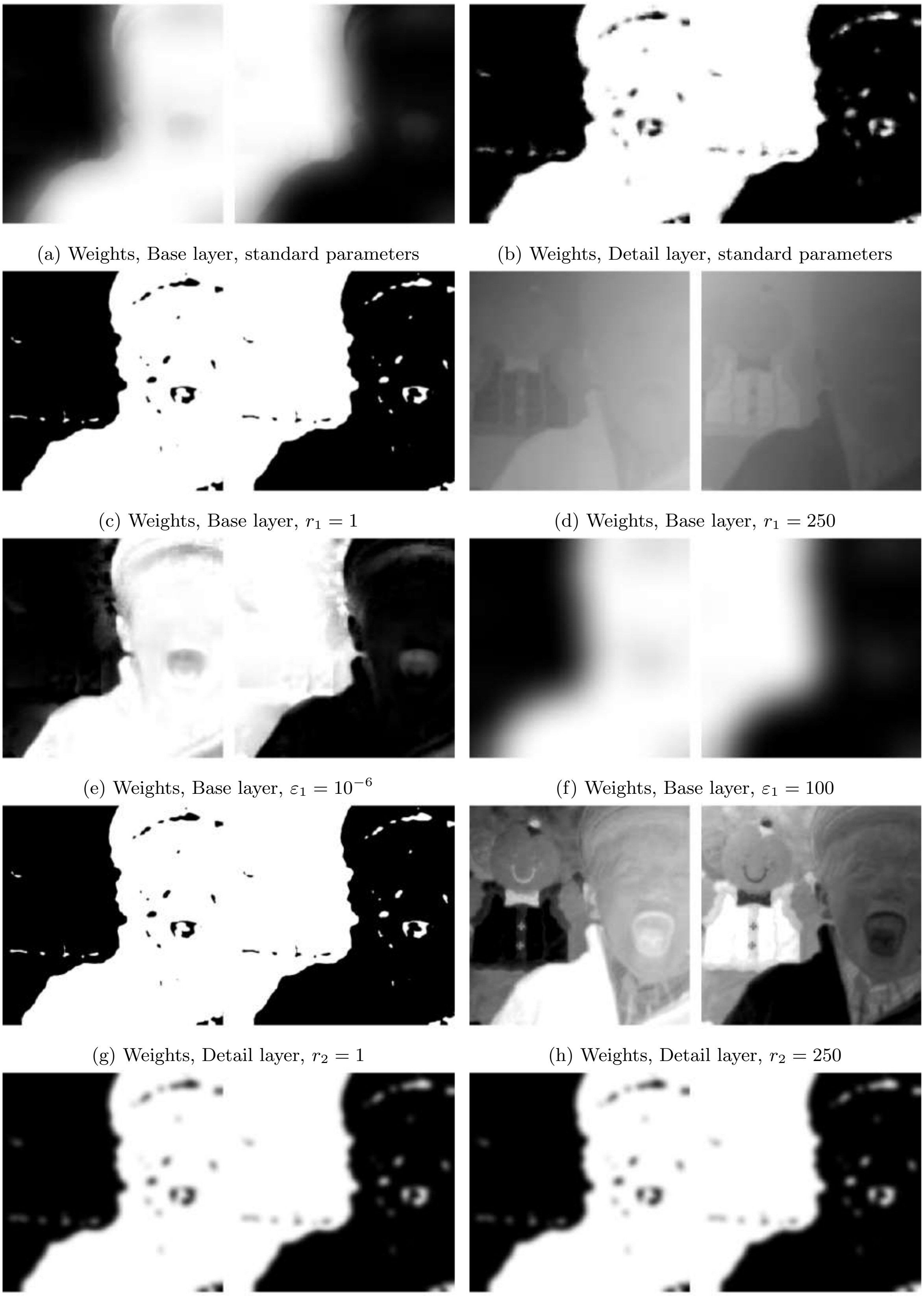


Figure 11: Visual impact of hyperparameters

Figure 12c and 12d display the effect of r_1 on the base weight mask. With $r_1 = 1$, the weight mask is very sharp and almost looks the same as the detail weight layer. This comes from the fact that coefficients a_k and b_k from (6) are averaged on very small windows. With $r_1 = 250$, the window of average is extremely large, making the weight mask an overall grey image. Figure 12e shows that a small ε tends to diminish spatial consistency, as weights start to follow details. This is explained by the fact that a small ε allows for big a_k coefficients, which increase the impact of the details of the guide image on the weight mask. On the contrary, a big ε forces a_k to be close to 0, therefore the weight mask is equal to the average of the input image P_i. This explains the blurriness of the weight mask. Because the weight mask of the base layer is used to create the spatially smooth areas of the fused image, it is best to have relatively big r_1 and ε_1.

Figures 12g and 12h display the impact of r_2 on the detail weight mask. $r_2 = 1$ is similar to the standard detail weight mask, with sharper weight difference along edges, whereas $r_2 = 250$ makes the weights focus on details that are both in the foreground and the background. Therefore, the fused image



(a) Weights, Base layer, standard parameters

(b) Weights, Detail layer, standard parameters

(c) Weights, Base layer, $r_1 = 1$

(d) Weights, Base layer, $r_1 = 250$

(e) Weights, Base layer, $\varepsilon_1 = 10^{-6}$

(f) Weights, Base layer, $\varepsilon_1 = 100$

(g) Weights, Detail layer, $r_2 = 1$

(h) Weights, Detail layer, $r_2 = 250$

(i) Weights, Detail layer, $\varepsilon_2 = 0.3$

(j) Weights, Detail layer, $\varepsilon_2 = 100$

Figure 12: Weight maps for base and detail layers, for various parameters

will contain “details” that are actually blurry elements, hence the fuzzy aspect of result with $r_2 = 250$ on Figure 11. Finally, Figures 12i and 12j show that bigger ε makes the weight mask blurrier, as explained earlier. This should be avoided because we want the detail weight mask to be edge-aligned.

3.2.2 Impact of hyperparameters : Quantitative evaluation

In order to further analyze the impact of the hyperparameters, we made a quantitative evaluation. To this aim, we implemented 3 metrics, which were used in the article [6] to evaluate the fusion performance. These metrics focus on different performance criteria and are defined in the following paragraph. We call A, B the 2 input images and F the fusion result. We implemented these metrics for gray-scale images, as the given definitions were adapted for gray-scale, and we converted coloured images to grayscale before applying these metrics.

Normalized mutual information metric Q_{MI}. This metric, which is based on information theory, is defined in [6] as :

$$Q_{MI} = 2\left[\frac{MI(A,F)}{H(A)+H(F)} + \frac{MI(B,F)}{H(B)+H(F)}\right]$$

with $MI(A,F) = H(A) + H(F) - H(A,F)$, $H(A)$ the entropy of image A and $H(A,F)$ the joint entropy. This metric measures how well the information of both input images is preserved in the fusion result.

Structural similarity for fusion Q_Y. This metric is based on structural similarity metric SSIM and quantifies how well structural information is preserved. It is defined as :

$$Q_Y = \begin{cases} \lambda_w \,\mathrm{SSIM}\,(A_w, F_w) + (1-\lambda_w)\,\mathrm{SSIM}\,(B_w, F_w) \\ \quad \text{if } \mathrm{SSIM}\,(A_w, B_w \mid w) \geq 0.75 \\ \max\left\{\mathrm{SSIM}\,(A_w, F_w), \mathrm{SSIM}\,(B_w, F_w)\right\} \\ \quad \text{if } \mathrm{SSIM}\,(A_w, B_w \mid w) < 0.75 \end{cases}$$

where SSIM is computed in windows of size w and $\lambda_w = \frac{s(A_w)}{s(A_w)+s(B_w)}$, with $s(A_w)$ the variance of image A in the window. In practice, we used $w = 7$ as it is done in the article, and we used the implementation of SSIM metric from scikit-image.

Gradient based index Q_G. This index is introduced in [8]. It quantifies how much edge information is preserved in the fusion result. It is defined as :

$$Q_G^{AB/F} = \frac{\sum_{n=1}^{N}\sum_{m=1}^{M} Q^{AF}(n,m)w^A(n,m) + Q^{BF}(n,m)w^B(n,m)}{\sum_{i=1}^{N}\sum_{j=1}^{M}\left(w^A(i,j) + w^B(i,j)\right)}$$

If $Q^{AF}(n,m) = 1$, then there is no loss of edge information. $Q^{AF}(n,m)$ is the product of $Q_g^{AF}(n,m)$, the edge strength preservation value and $Q_\alpha^{AF}(n,m)$, the edge orientation preservation value. These are defined as :

$$Q_g^{AF}(n,m) = \frac{\Gamma_g}{1 + e^{\kappa_g(G^{AF}(n,m)-\sigma_g)}}$$
$$Q_\alpha^{AF}(n,m) = \frac{\Gamma_\alpha}{1 + e^{\kappa_\alpha(A^{AF}(n,m)-\sigma_\alpha)}}$$

$G^{AF}(n,m)$ is the relative strength value of image A with respect to F and $A^{AF}(n,m)$ the relative orientation value :

$$G^{AF}(n,m) = \begin{cases} \frac{g_F(n,m)}{g_A(n,m)}, & \text{if} \quad g_A(n,m) > g_F(n,m) \\ \frac{g_A(n,m)}{g_F(n,m)}, & \text{otherwise} \end{cases}$$
$$A^{AF}(n,m) = \frac{|\alpha_A(n,m) - \alpha_F(n,m)| - \pi/2\ |}{\pi/2}$$

The edge strength and orientation values $g_A(n,m)$ and $\alpha_A(n,m)$ are defined using $s_A^x(n,m)$ and $s_A^y(n,m)$ the outputs of horizontal and vertical Sobel operators centered on pixel of position (n,m).

$$g_A(n,m) = \sqrt{s_A^x(n,m)^2 + s_A^y(n,m)^2}$$
$$\alpha_A(n,m) = \tan^{-1}\left(\frac{s_A^y(n,m)}{s_A^x(n,m)}\right)$$

The weights $w^A(n,m)$ are defined as $g_A(n,m)^L$. The values of the parameters Γ_g, σ_g, κ_g, Γ_α, σ_α, κ_α, L are not mentioned in the studied article [6] so we used the values of the original article where this metric was defined [8] : $\Gamma_g = 0.9994$, $\sigma_g = 0.5$, $\kappa_g = -15$, $\Gamma_\alpha = 0.9879$, $\sigma_\alpha = 0.8$, $\kappa_\alpha = -22$, $L = 1.5$.

Using these three metrics, we evaluated the performance of fusion for different sets of hyperparameters r_1, ε_1, r_2, ε_2, by doing an average on 10 pairs of images. When varying the value of one parameters, all the other ones are kept at their standard value. We also display the standard deviation with the colored shadows around the average values. These graphics are shown on Figure 13.

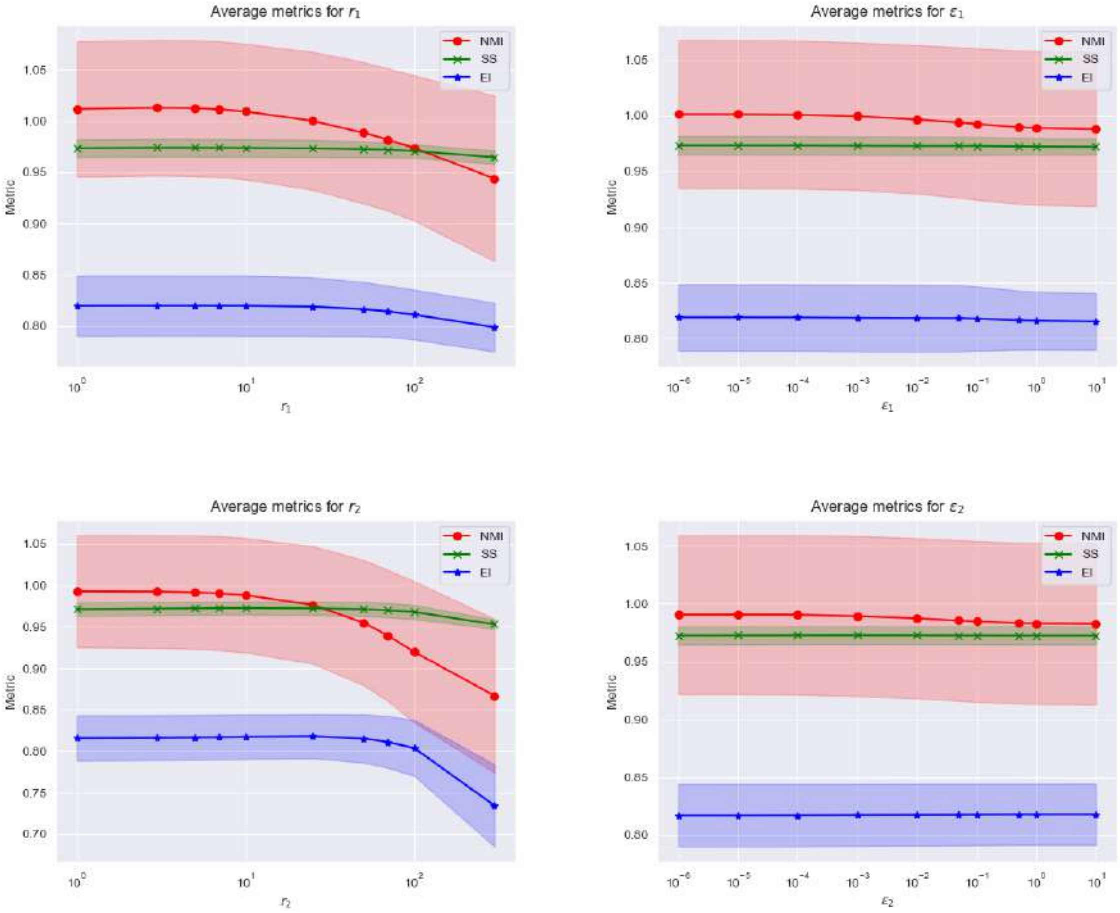


Figure 13: Quantitative evaluation of hyperparameters' impact on fusion performance

First, we can see that the average metric values are quite stable for all hyperparameters, and the performance starts to decrease only for extremely high hyperparameter values. Once again, this confirms the article's remark about the low impact of the parameter values, at least for multi-focus pairs of images. The graphics also attest that r_1 and r_2 values should not be too high, especially for r_2. They also highlight the importance of having multiple metrics when assessing fusion performance. In fact, we do not observe the same rate of decrease in the metric values when changing the hyperparameters. Moreover, we notice that the standard deviation with SSIM metric is much smaller than with normalized mutual information metric. This means that the NMI value changes a lot along the tested images. On the contrary, the value of SSIM metric is stable along images. As these metrics furthermore evaluate different performance criteria (structural information, original information, edge information), the fusion performance should be evaluated looking at these various metrics, as it is done in the article.

3.3 Experiment on multiple images: application to HDR

To evaluate the method on a more challenging problem, we turned to High Dynamic Range imagery, again reusing a dataset already tested by the authors in their article [6]: the Stanford Memorial Church radiance map dataset [1]. This dataset, shown on figure 14, is composed of 16 photographs of dimension 768 × 512 of the Stanford Memorial Church, taken with various exposure times.

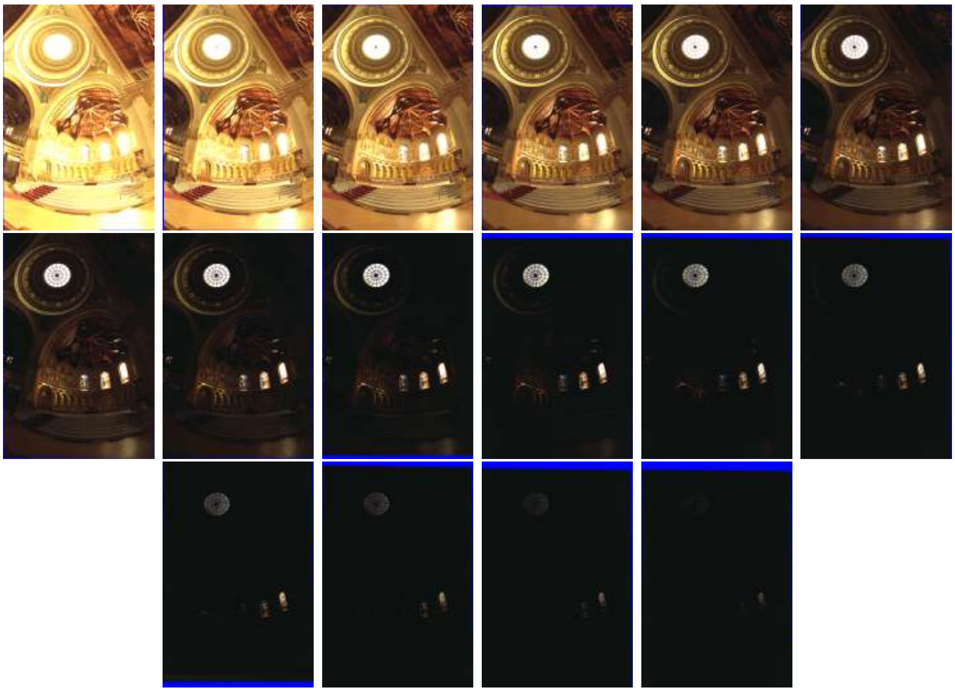


Figure 14: Stanford Memorial Church HDR dataset

Running the algorithm on this dataset takes about 32 seconds. Searching for parameters close to the standard parameters of the article, we easily get very satisfying results with $r_1 = 42$, $\varepsilon_1 = 0.1$, $r_2 = 15$ and $\varepsilon_2 = 5 \times 10^{-2}$; the fused image is presented in Figure 15. The algorithm gives a well-balanced fusion with only minor deformations, observed on the boundary of some windows and on the dome. It also handles quite well the blue boundaries visible on some of the input images of Figure 14. However, we also notice an increased sensibility to the parameters of the Guided Filters when comparing with the results obtained on pairs of multi-focus images. In particular:

- Figure 16 shows reconstructions for fixed $r_2 = 15$ and $\varepsilon_2 = 5 \times 10^{-2}$ and variable r_1 and ε_1. For smaller values of these parameters, the stained-glass windows show more details, at the expanse of shadowy regions appearing at their boundaries. On the contrary, increasing these parameters gives a lighter and more uniform result, but brushes out the details of the windows.
- Figure 17 shows reconstructions for fixed $r_1 = 42$ and $\varepsilon_1 = 0.1$ and variable r_2 and ε_2. The impact of changing these parameters is similar to that of the previous case, but more subtle. Here, decreasing the parameters blurs out the boundaries of the windows, while increasing them lightens the windows without completely saturating them as for the first filter.

Overall, for this application the setting of the parameters is a more challenging compromise to find, although the standard values already give good results. It also seems that the filter for the base layer shares a greater responsibility in the conservation of the edges of objects and the uniformity of the fused image, although the second filter remains necessary to fine-tune the result. This is in line with the interpretation of the parameters made in previous sections.

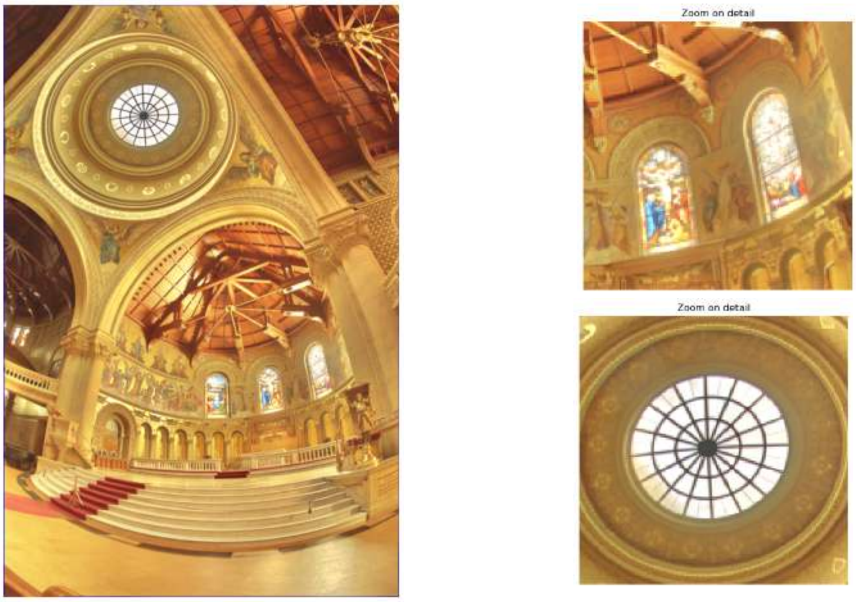


Figure 15: HDR reconstruction and details ($r_1 = 42, \varepsilon_1 = 0.1, r_2 = 15, \varepsilon_2 = 5 \times 10^{-2}$)

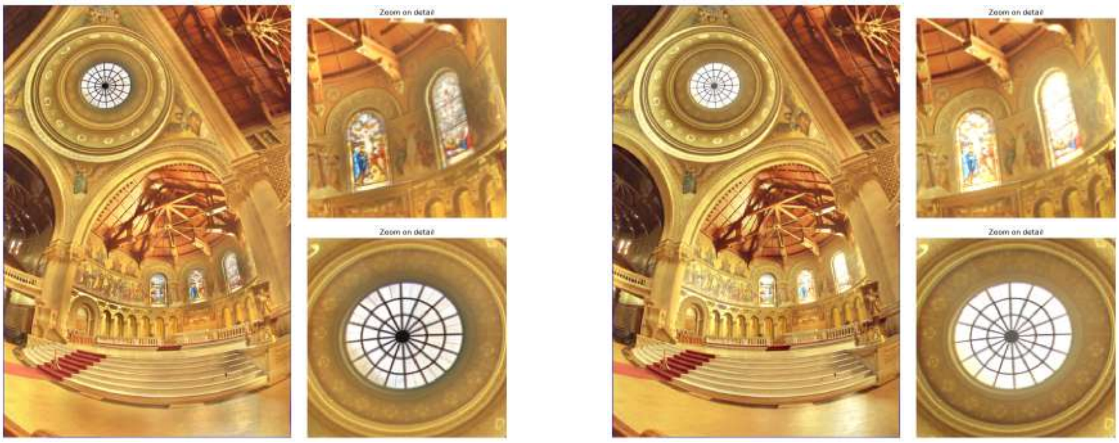


Figure 16: Reconstruction for ($r_1 = 15$, $\varepsilon_1 = 0.1$) (Left) and ($r_1 = 90$, $\varepsilon_1 = 0.6$) (Right)

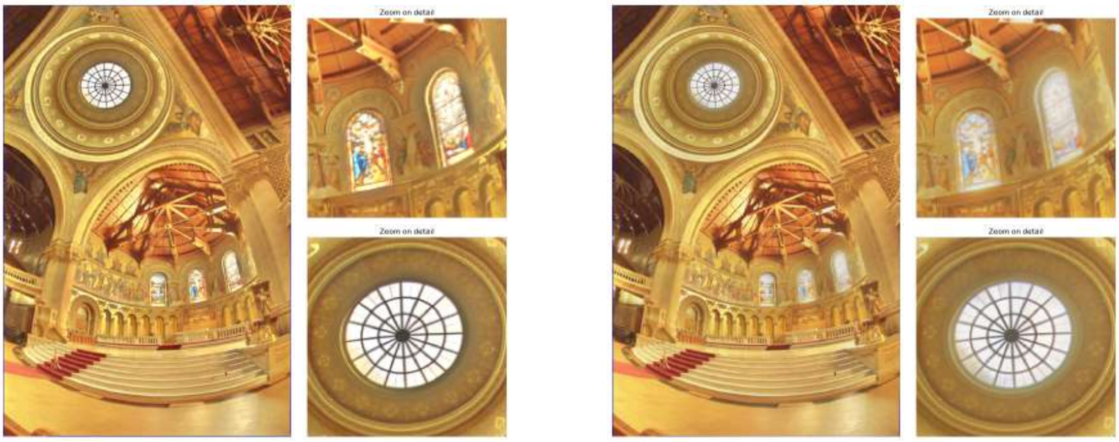


Figure 17: Reconstruction for ($r_2 = 7$, $\varepsilon_2 = 10^{-4}$) (Left) and ($r_2 = 20$, $\varepsilon_2 = 0.5$) (Right)

4. Conclusion

The fusion method based on Guided Filtering proposed in article [6] is rather straightforward to implement, as it combines conventional components such as Laplacian, Gaussian and Guided Filters in a rather linear way. It also relies on widespread ideas, such as the decomposition into base and detail layers, although it also comes with a share of originality, such as applying a filter not on the original images but on their saliency maps comparison. Furthermore, the results are mostly reproducible. Overall, this fusion algorithm is efficient and the results are realistic and refined for all tested applications: multi-focus images, multi-exposure images and, marginally, multi-source images.

In some cases, we observed some artifacts, such as a blurry halo between an object at the foreground and the background for multi-focus images. We also noticed that the fusion can be less sharp in transition areas between the 2 source images, meaning the areas which join the blurry and sharp parts of the two source pictures. That is why we can say that the method can sometimes lack some precision around edges and requires high-quality source images. However, these artifacts were minority and we can say that the proposed method produces very convincing fusions.

The experiments conducted on the impact of the parameters of the Guided Filter enabled to understand the role of parameters r and ε in the construction of the weight masks. Although changing these parameters had a visual impact on the weight masks, it had little effect on the final fusion result for multi-focus pictures. This was confirmed by the quantitative analysis which showed that various metric values were stable when changing parameters. The 3 metrics that we re-implemented assessed different criteria of performance which are all equally important for a global performance evaluation. On the other hand, for more challenging applications involving a greater number of input pictures with starker difference between them, the result starts being more sensitive to the choice of parameters, in similar but subtly different manners for the two filters.

All in all, it is hard to conclude on obvious roles played by the parameters for the two filters, as our experiments have shown a stronger dependency of the result on one filter or the other depending on the situation. However, the ablation carried out in Section 3.1.3, the study performed in Section 3.2.1 and the application to multiple images of Section 3.3 seem to indicate that the base filter is usually responsible for most of the spatial consistency and uniformity of the image as well as the conservation of the most prominent features, while the detail filter is useful for fine-tuning the fusion result.

References

[1] P. E. Debevec and J. Malik. Recovering high dynamic range radiance maps from photographs. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '97, pages 369–378, USA, Aug. 1997. ACM Press/Addison-Wesley Publishing Co.

[2] J. Delon. TP Transfert de Couleur. `https://github.com/storimaging/Notebooks/blob/main/ContrastAndColor/Solutions/TP_color_transfer.py`, 2022.

[3] F. Durand and J. Dorsey. Fast bilateral filtering for the display of high-dynamic-range images. *ACM Transactions on Graphics*, 21(3):257–266, July 2002.

[4] K. He, J. Sun, and X. Tang. Guided Image Filtering. In K. Daniilidis, P. Maragos, and N. Paragios, editors, *Computer Vision – ECCV 2010*, Lecture Notes in Computer Science, pages 1–14, Berlin, Heidelberg, 2010. Springer.

[5] M. Nejati, S. Samavi, and S. Shirani. Multi-focus image fusion using dictionary-based sparse representation. *Information Fusion*, 25(C):72–84, Sept. 2015.

[6] Shutao Li, Xudong Kang, and Jianwen Hu. Image Fusion With Guided Filtering. *IEEE Transactions on Image Processing*, 22(7):2864–2875, July 2013.

[7] A. Toet. The TNO Multiband Image Data Collection. *Data in Brief*, 15:249–251, Dec. 2017.

[8] C. Xydeas and V. Petrovic. Objective image fusion performance measure. *Electronics Letters*, 36:308–309, Mar. 2000.

Appendix A. Guided filter illustration

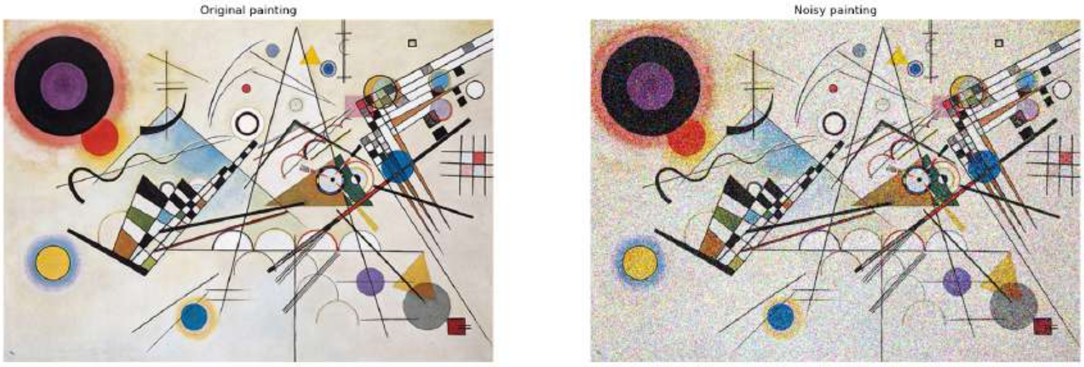
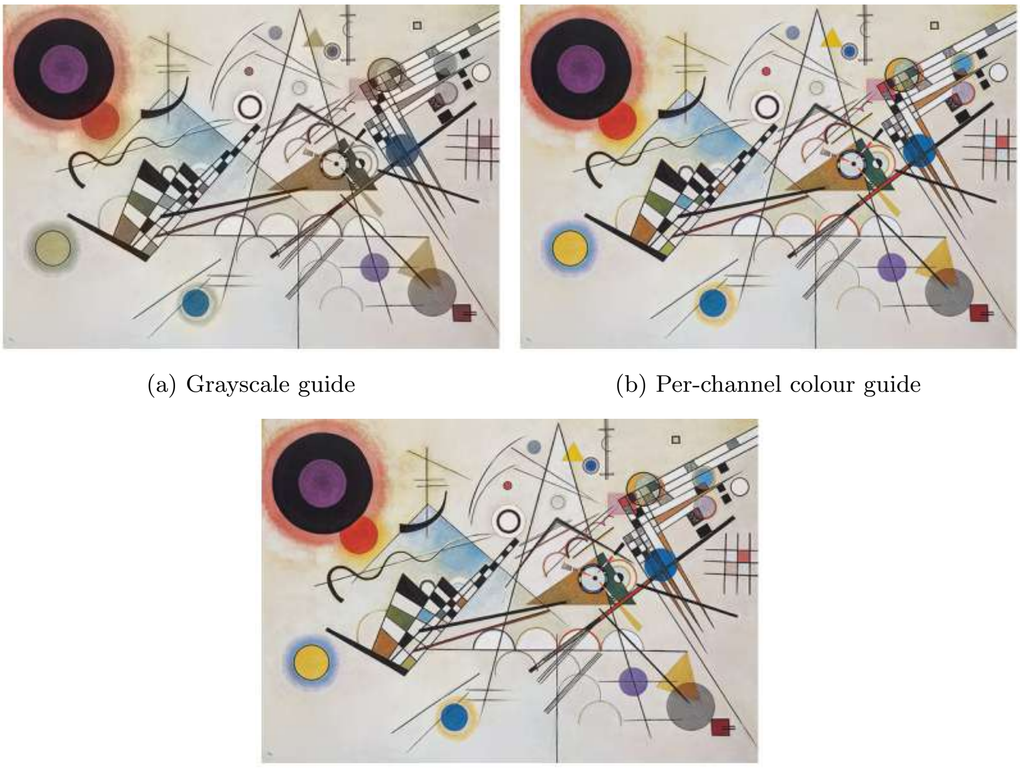


Figure 18: Original painting by Kandinsky and noisy painting



(a) Grayscale guide

(b) Per-channel colour guide

(c) Colour guide

Figure 19: Filtered painting for various guides

Appendix B. Application to multi-source images

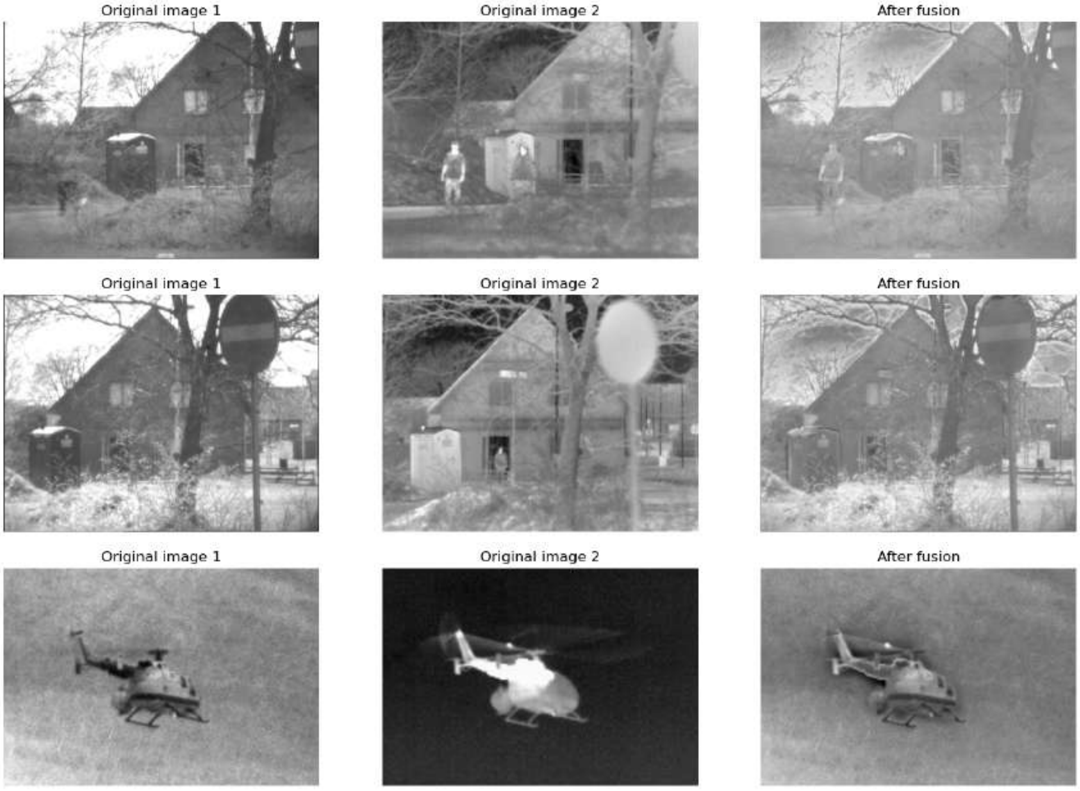


Figure 20: Results of fusion on multi-source images (Left: optic; Right: infra-red)