Télécom Paristech

IMA 201

Image Fusion with Guided Filtering

Work of :

Alaeddine Sabbagh

Ramzi Charradi

Supervised by

Mr.Said Ladjal
Lecturer at Télécom Paristech

15 November 2018

Contents

List of Figures

Abstract

The ultimate goal of this project is to implement the guided filtering method and check its limits: The idea is to merge two images with the latter method to get a new image which is more informative. The proposed method is based on a two-scale decomposition of an image into a base layer containing large scale variations in intensity, and a detail layer capturing small scale details. This guided filtering-based weighted average technique is proposed to make full use of spatial consistency for fusion of the base and detail layers. The guided filter is also a more generic concept beyond smoothing: It can transfer the structures of the guidance image to the filtering output,

Index Terms: Guided filter, image fusion, two-scale decomposition.

1 Introduction

This work is a review of (Li, Kang, and Hu 2013)[1]. It has been conducted as part of the validation process of the IMA 201 lecture at Télécom Paristech. This reports focuses on explaining the idea of the guided filter fusion as presented in the studied paper.We start by laying the fondation of guided filtering then we move to understanding the fusion of two images using this strategy on two levels: gray scale image fusion and color image fusion. Through image fusion,different images of the same scene can be combined into a single fused image . The fused image can provide more comprehensive information about the scene which is more useful for human and machine perception. A good fusion algorithm takes care of the following:

• It can preserve most of the useful information of different images.

• It does not produce artifacts.

• It is robust to imperfect conditions such as misregistration and noise

In this report we start by explaining the guided filtering principale then we see its application on the fusion of two images and eventually we discuss the limits of this implementation.

2 Guided filtering principal

2.1 Gray Scale Images

In theory, the guided filter assumes that the filtering output O is a linear transformation of the guidance image I in a local window w_k centered at pixel k.

The idea involves a filtering input image p, a guidance image I, and an output image O. Both I and p are given beforehand according to the application, The filtering output at a pixel i is expressed as a weighted average: Let w_k be a local window with size $(2r + 1) \times (2r + 1)$ centered at k then:

$\forall i \in w_k, O_i = a_k I_i + b_k$;

where $a_k and b_k$ are some linear coefficients assumed to be constant in w_k . This

local linear model ensures that O has an edge only if I has an edge, because $\nabla O = a\nabla I$. This model has been proven useful in image super-resolution [2]. To determine a_k and b_k We need to minimize the following cost function in the window w_k: $E(a_k, b_k) = \sum_{i\in w_k}((a_k I_i + b_k - P_i)^2 + \epsilon a_k^2)$.
where ϵ is a regularization parameter given by the user. The coefficients a_k and b_k can be directly solved by linear regression as follows:

$$a_k = \frac{\frac{1}{|w|}\sum_{i\in w_k} I_i P_i - \mu_k \overline{P_k}}{\sigma_k^2 + \epsilon}$$

$$b_k = \overline{P_k} - a_k \mu_k$$

where μ_k and σ_k are the mean and variance of I in w_k respectively, |w| is the number of pixels in w_k, and P_k is the mean of P in w_k.
Having obtained the linear coefficients a_k and b_k, However, a pixel i is involved in all the overlapping window w_k that covers i, so the value of the outpuet depends on the window w_k. A simple strategy is to average all the possible values of a_i and b_i Eventually, can compute the filtering output O_i:
$O_i = \overline{a_i} I + \overline{b_i}$ Where $\overline{a_i} = \frac{1}{|w|}\sum_{k\in w_i} a_k$ and $\overline{b_i} = \frac{1}{|w|}\sum_{k\in w_i} b_k$

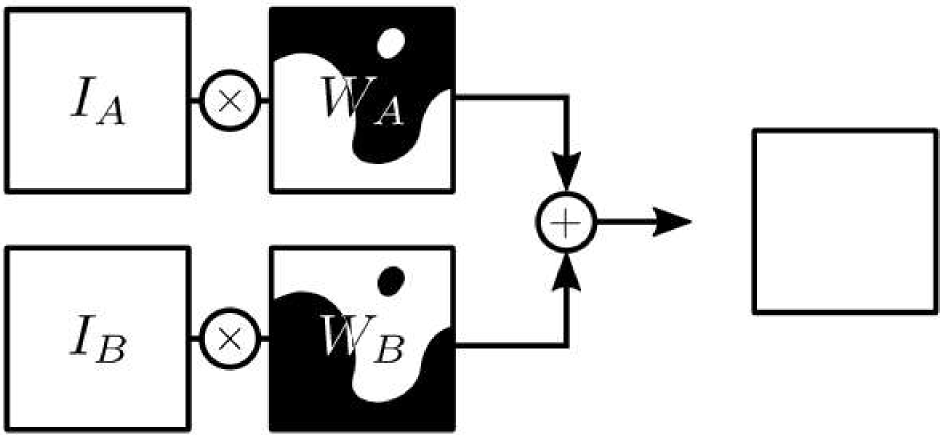
2.2 Extension to color Images

For color images we conduct the guided filtering on each of the channels: the Red, the blue and the green one.

3 Image fusion

3.1 Basic binary fusion

let's say we want to merge two images I_A and I_B, whatever there is on them: Roughly, merging will consist in picking each pixel of the output image either from IA or from IB. More generally, the output will be a linear combination of both inputs: $W_A \times I_A + W_B \times I_B$ where $\times$ stands for the pixel-wise product.



Weighted image merging

Figure 1: Binary fusion

As illustrated on the scheme, in most cases $W_B = 1 - W_A$ or almost.
This may be important for some energy conservation.

GFF (Guided Filter Fusion) is no different from this simple merging and all the magic is in the definition of the weight matrix.

3.2 Saliency

In order to find the merging weights, also called masks when they are binary, GFF relies on the notion of saliency. In simple terms, the idea is to pick the most detailed pixel, regarding the input images. The saliency map explicits this alforith: It is a measure of "detailness" based on the laplacian of the image. The higher the magnitude of the laplacian, the more detailed the picture. Visually, the brighter the pixel is, the more detailed our pixel is. We can spread the notion of the neighborhood of a pixel, if we blur the saliency map.

The GFF algorithm compares the saliency of both input images and sets the output at pixel i as 1 if the first image has a higher salency at that pixel and 0 otherwise. This can easily be extended to more than two images.

At this point, the weight W_A and W_B are just binary masks
representing $\text{argmax}(S_A, S_B)$ where S stands for the saliency map.

3.3 Image fusion with guided filtering

The main problem of binary weights is that it introduces artifacts at the mask boundaries, moving from one input image to the other one. The first idea to tackle this is to blur the masks, but one quickly notices that it breaks the sharpness of the mask in some areas in which it is needed.

In order to blur the mask while keeping details, GFF uses a Guided Filter. This makes the merging, i.e the artifacts, much less visible.

3.3.1 Multiscale decomposition

We won't merge the two original images using GFF directly. Instead, we will decompose each image into two scales: a detailed level with high frequencies and a blurred level with low frequencies. By the end, We will merge the 4 obtained images back into the final output. This operation consistes of 2 steps:

- The base layer B_n can be obtained by convolving the source images with a simple averaging filter.
- The detail layer can be easily obtained by subtracting the base layer from the source image: $D_n = I_n - B_n$.

3.3.2 Multiscale reconstruction

Two-scale image reconstruction consists of the following two steps. First, the base and detail layers of different source images are fused together by weighted averaging:

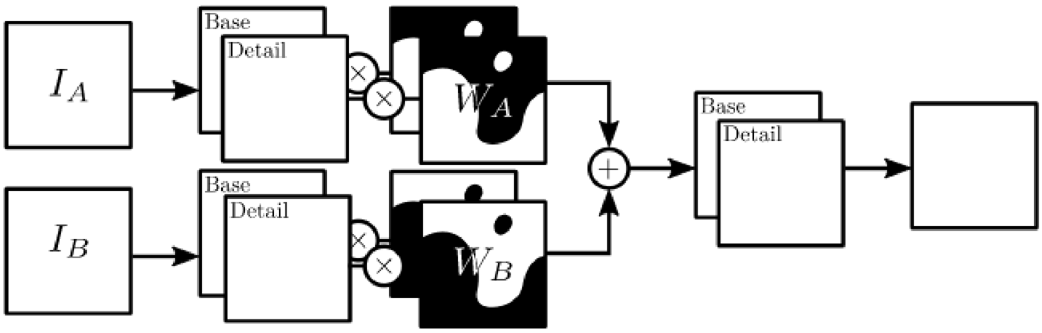
$$\overline{B} = \sum_{n=1}^{N} W_n^B B_n$$

$$\overline{D} = \sum_{n=1}^{N} W_n^D D_n$$

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

$$F = \overline{B} + \overline{D}$$

The following figure resumes the GFF for 2 images:



Multiscale weighted image merging

Figure 2: GFF

4 Results and discusssion of the GFF

To capture the following picture we have used Canon 5D Mark III.

4.1 Results

4.1.1 Example 1

(a) 1.a

(b) 1.b

Figure 3: The 2 input images

Figure 4: Output1

4.1.2 Example 2

(a) 2.a

(b) 2.b

Figure 5: The 2 input images

Figure 6: Output2

As expected the in both examples, final iamge is more informative than the two input images, and it doesn't contain any artifacts.

4.2 Discussion

In this section we will try to see the extent of the GFF method and see if it can be applied in other contexts.

4.2.1 Example 1

Here we merge two images where one of them is so bright and the other is locally dark.

(a) 3.a (b) 3.b

Figure 7: The 2 input images

Figure 8: Output4

The final image is more detailed that the second image which is so bright. We got the high frequencies in the 3.a. Moreoever, the word "acces" is so dark in 3.a that we couldn't read it, but in the final image we became able to. We can say that the GFF is also useful when merging two images with different contrast, we still get the best from each picture, even though some noise appeared in output4. We see the same effect in the next example:

Here one of the second pictures is overbright.

(a) 1.a

(b) 1.b

Figure 9: The 2 input images

Figure 10: Output3

The effect we see in the final picture is comparable to the effect we see when streching the histogramme.

4.2.2 Example 2

In this example we add a new object to the scene.

(a) 4.a

(b) 4.b

Figure 11: The 2 input images

Figure 12: Output4

The final image preserved the edges of the hand, which is expected since GFF preserves all edges in both images as shown in section 2. Because the pixels are almost uniformily distributed locally, in the zone of the hand, GFF picked the pixels who are in in 4.a since they contain more details.

4.2.3 Example 3

Here we added gaussian noise to the first image.

(a) 5.a (b) 5.b

Figure 13: The 2 input images

Figure 14: Output4

After adding noise to the first image, the fused image preserves the noise. Since GFF relies on saliency map, it has a tendency to highlight areas where pixel values tend to change quickly, which might be noisy areas.

5 References

[1] http://xudongkang.weebly.com/uploads/1/6/4/6/16465750/tip1.pdf.

[2] A. Zomet and S. Peleg, ‘Multi-sensor super resolution’, 2002.