

# Homework Assignment 3

15-463/663/862, Computational Photography, Fall 2019  
Carnegie Mellon University

Due Friday, Oct. 11, at 11:59pm

The purpose of this assignment is to explore photography with flash/no-flash pairs. In general, getting the lighting in your scene right is one of the most important considerations one needs to take care of when taking a photograph. This can be particularly challenging when the scene you are trying to photograph has a large dynamic range, including both very dark and very bright objects. Previously, we saw a *passive* technique for photographing such difficult scenes, using HDR imaging. Here we will investigate an alternative *active* technique, where we insert new light into the scene using our camera’s flash.

As we discussed in class, using flash has advantages and disadvantages. The extra light reduces noise, allows us to use larger shutter speeds to reduce motion blur, and can help illuminate the scene more uniformly. On the flip side, the use of the harsh flash lighting can ruin the scene’s ambiance, produce highlights, and overall result in unappealing appearance. These issues commonly arise when using simple point flash lights, e.g., the one on the class DLSR or at the back of your cell phone. A photographer with access to some more advanced equipment would use either a pointable flash that they can bounce off of the ceiling, or they would use a diffuse display to make the light smoother. However, for the purposes of this assignment, we can assume we do not have access to such equipment.

Instead, we can attempt to produce images that combine the advantages of flash and ambient illumination by capturing two images, one with each illumination, and then *fusing* them into one. In class we saw two classes of techniques for this kind of *flash/no-flash photography*, one based on bilateral filtering, and another based on gradient-domain processing. In this homework, you will implement both of these techniques, and apply them to photos provided by us, as well as photos you capture on your own.

Throughout the assignment, we refer to a number of key papers that were also discussed in class. While the assignment and class slides describe most of the steps you need to perform, it is highly recommended that you read the associated papers. As always, there is a “Hints and Information” section at the end of this document that is likely to help.

## 1. Bilateral filtering (100 points)



Figure 1: From left to right: Photos taken under ambient lighting and flash lighting; denoising result using basic bilateral filtering; and final denoising result.

In the first part of the homework, you will use the lamp flash/no-flash images from the paper by Petschnigg et al. [5], available in the `./data/lamp` directory of the homework ZIP archive. Your focus will be on *denoising* the ambient image using bilateral filtering. Figure 1 shows representative results for this section. As we discussed in class, bilateral filtering is a form of “blurring” that attempts to respect sharp edges in the image. This is achieved by weighting neighboring pixels not just based on their distance from the pixel that is being filtered, but also based on their similarity to it. This can help reduce noise in an image, without overly blurring details. Further improvements can be obtained by using bilateral filtering techniques that combine information from both the flash and ambient illumination images, as proposed by Petschnigg et al. [5]. Figure 3 of that paper is particularly helpful for navigating this homework.

**Implement bilateral filtering (40 points).** As a first attempt at denoising, you can simply apply the bilateral filter on the ambient image. Using the notation of Petschnigg et al. [5], we will denote the ambient image  $A$  and the denoised image  $A^{\text{Base}}$ . For this, you will implement bilateral filter as described in the paper by Durand and Dorsey [3]. We leave it up to you to decide whether to implement the vanilla algorithm, or any of the efficient algorithms described in Section 5 of that paper (the “PiecewiseBilateral” algorithm is generally about as complicated to implement as the vanilla algorithm, and significantly faster; the “FastBilateral” algorithm can be tricky to implement correctly due to the downsampling and upsampling steps).

**Implement joint-bilateral filtering (30 points).** You will now attempt to get a better result by using *joint bilateral filtering*, as described in Section 4.1 of Petschnigg et al. [5], to producing the image  $A^{\text{NR}}$  in the paper’s notation. As we discussed in class, joint bilateral filtering involves computing the intensity kernel using the flash image—see also Equation (4) of Petschnigg et al. [5].

**Implement detail transfer (20 points).** While joint bilateral filtering can help avoid overblurring the ambient image, it cannot *add* new detail to it. You will enhance the detail of your result by implementing the flash-to-ambient detail transfer procedure described in Section 4.2 of Petschnigg et al. [5]. For this, you will need to apply basic bilateral filtering to the flash image. Following Petschnigg et al. [5], we denote the flash image  $F$  and the result of filtering  $F^{\text{Base}}$ . Then, you can form a new estimate for the ambient image by combining  $F$ ,  $F^{\text{Base}}$ , and the image  $A^{\text{NR}}$  from before, as:

$$A^{\text{Detail}} = A^{\text{NR}} \frac{F + \epsilon}{F^{\text{Base}} + \epsilon}, \quad (1)$$

where  $\epsilon$  is some small value you can select on your own.

**Implement shadow and specular masking (10 points).** Finally, it is time to create the mask  $M$  described in Section 4.3 of Petschnigg et al. [5]. This is used to detect regions in the flash image that have shadows or specularities not present in the ambient image. The mask can be generated using the simple thresholding operations described in the paper. Once you have the mask, form a final estimate  $A^{\text{Final}}$  of the ambient image as

$$A^{\text{Final}} = (1 - M) \cdot A^{\text{Detail}} + M \cdot A^{\text{Base}}. \quad (2)$$

**Implementation details.** These details apply to all four of the previous steps. As we discussed in class, the bilateral filter takes as input two parameters, the standard deviation  $\sigma_s$  of the spatial Gaussian kernel, and the standard deviation  $\sigma_r$  of the intensity Gaussian kernel. You will need to experiment with different values of these parameters to find which ones give the most aesthetically pleasing results. A good range to experiment with is  $\sigma_s \in [1, 64]$  and  $\sigma_r \in [0.05, 0.25]$ . Note that these values assume that the maximum value in your image is 1, so make sure to normalize first.

In general, the best parameters will be different for each of the four types of filtering you will implement: basic bilateral filtering, joint bilateral filtering, denoising with detail transfer, and mask-based merging. In your report, you should compare the results of the four algorithms for different sets of parameter values. In these comparisons, you should make sure to show difference images  $\text{image}_1 - \text{image}_2$ , which should help highlight the places in the image where the techniques perform differently. You should report the parameter values you chose as best for the four types of filtering, and discuss the advantages and disadvantages of each type.

## 2. Gradient-domain processing (100 points)

In the second part of the homework, you will use the museum flash/no-flash images from the paper by Agrawal et al. [1], available in the `./data/museum` directory of the homework ZIP archive. Figure 2(a-b) shows the two images. As you can see in the ambient image, the background painting is nicely illuminated from the ambient lighting. However, the person in the foreground is very dimly lit. We can attempt to address this using the camera’s flash, which results in the flash photograph. Now the person in the foreground is well-illuminated, but our photograph suffers from a different problem: The sharp lighting of our flash has created a specular highlight on the painting. In this part of the homework, you will focus on fusing the flash and



Figure 2: Combining photographs taken under ambient lighting (left) and flash lighting (middle), to produce a new image that combines the best of both worlds (right).

ambient images, in order to produce uniform illumination without the specularities and other artifacts. The end result of the fusion is shown in Figure 2(c). You will do this fusion using gradient-domain processing, as proposed by Agrawal et al. [1]. As usual, you are strongly encouraged to read through the paper, to find many implementation details and insights about the various parameters.

**Create the fused gradient field (25 points).** Following Agrawal et al. [1], we will denote the ambient image as  $a$ , the flash image as  $\Phi'$ , and the fused image as  $\Phi^*$ . The first step in the fusion process requires forming the gradient field that will be used to create  $\Phi^*$ . In turn, this gradient field will be created using the gradient fields  $\nabla a$  and  $\nabla \Phi'$  of the ambient and flash image, respectively.

First, compute  $\nabla a$  and  $\nabla \Phi'$ , then use them to implement the following steps:

1. Compute the *gradient orientation coherency map*  $M$  using Equation (5) of Agrawal et al. [1],

$$M = \frac{|\nabla \Phi' \cdot \nabla a|}{\|\nabla \Phi'\| \|\nabla a\|}. \quad (3)$$

2. Compute the pixel-wise *saturation weight map*  $w_s$  from the flash image  $\Phi'$  using Equation (11) of Agrawal et al. [1],

$$w_s = \tanh(\sigma \cdot (\Phi' - \tau_s)). \quad (4)$$

Normalize the saturation map to be in the range  $[0, 1]$ .

3. Compute a new gradient field for the fused image  $\Phi^*$  using Equation (12) of Agrawal et al. [1],

$$\nabla \Phi^* = w_s \nabla a + (1 - w_s) (M \cdot \nabla \Phi' + (1 - M) \cdot \nabla a). \quad (5)$$

You should experiment with different values for  $\sigma$  and  $\tau_s$ , starting with the values recommended in the paper. In your report, you should show the gradient fields  $\nabla a$ ,  $\nabla \Phi'$ , and  $\nabla \Phi^*$ , and mention the parameter values you used.

**Integrate the fused gradient field (75 points).** You should now integrate  $\nabla \Phi^*$  to form the final fused image  $\Phi^*$ . For this, you will implement the Poisson solver based on conjugate gradient descent, as discussed in class. This is based on the iteration, for  $i = 0, \dots, N$ ,

$$\eta^i = \frac{d^i \cdot r^i}{d^i \cdot \text{LaplacianFiltering}(d^i)}, \quad (6)$$

$$\Phi^{*,i+1} = \Phi^{*,i} + \eta^i d^i, \quad (7)$$

$$r^i = \nabla \cdot \nabla \Phi^* - \text{LaplacianFiltering}(\Phi^{*,i}), \quad (8)$$

$$\beta^{i+1} = \frac{r^{i+1} \cdot r^{i+1}}{r^i \cdot r^i}, \quad (9)$$

$$d^{i+1} = r^{i+1} + \beta^{i+1} \cdot d^i, \quad (10)$$

with  $d^0 = r^0$ , and where LaplacianFiltering corresponds to convolution with the Laplacian kernel. For initializing the iteration, you can set  $\Phi^{*,0}$  equal to the ambient image. Instead of using a fixed  $N$ , you should implement one of the standard convergence criteria, e.g., the norm of the residual  $r^i$  becoming smaller than some threshold.

When doing integration, you need to select and be mindful of boundary conditions. We recommend using Dirichlet boundary conditions: Set the outermost pixels of  $\Phi^*$ , at all four image edges, to equal the corresponding pixels of either the ambient image, or the flash image, or their average. Experiment with different boundary conditions, and show the final merge you obtain.

### 3. Capture your own flash/noflash pairs (100 points)

Now it is time to apply what you implemented above to your own pictures. You should capture two sets of flash/no-flash pairs. The first one should be suitable for applying the denoising techniques based on bilateral filtering. Good examples include dimly lit environments, e.g., a room illuminated by just a desk lamp, or an outdoors scene at night illuminated by only a street lamp. You can look at the paper by Petschnigg et al. [5] for inspiration.

The second set should be suitable for applying the fusion algorithm based on gradient-domain processing. To create results which are clearly better than any single exposure, you should take pictures of a scene where flash affects some parts of the image, while leaving others relatively unaffected. Good examples include dark scenes that have both matte and specular objects. You can look at the paper by Agrawal et al. [1] for inspiration.

The total number of points you will get for this part will depend on how visually compelling the final fused images you create are.

### 4. Bonus: Reflection removal in flash/no-flash photography (50 points)

A different potential problem when using flash is that it can create reflections off of glass surfaces. See Figure 6 in Agrawal et al. [1]. Implement the fusion algorithm described in Section 4.1 (“Observation Deck”) of the paper, and then use it on a flash/no-flash pair you capture with your own camera. We suggest using a scene with a window for demonstrating this part, as in the paper.

### 5. Bonus: Other types of gradient-domain processing (up to 50 points)

Gradient-domain processing is a general image processing paradigm that can be used for a broad set of applications, including blending, tone-mapping, colorization, converting to grayscale, edge enhancement, image abstraction and non-photorealistic rendering, and so on.

GradientShop [2] is a paper and associated app that provide a unifying framework for expressing all these tasks as gradient-domain processing algorithms. Read this paper, and implement whichever of the applications described within appeal to you the most (the instructor’s personal favorite is non-photorealistic rendering, but it won’t reflect badly on your grade if you go for something else). For additional inspiration, you can take a look at the results in the Poisson image editing paper [4].

Your implementation does not need to follow GradientShop exactly. For instance, instead of implementing the “long-edge detector” described in the paper, you can come up with your own detector of continuous edges. We also recommend *not* trying the video-based applications.

In your submitted report, you should make sure to describe in detail exactly what you did. This includes what task you chose to implement, where you followed and where you deviated from GradientShop, and in the latter case what you did differently. You should also show at least three examples showcasing the effects of your gradient-domain processing pipeline.

## Deliverables

As described on the course website, solutions are submitted through Canvas. Your solution should be an archive (e.g., a ZIP file) that includes the following:

- A PDF report explaining what you did for each problem, including answers to all questions asked throughout Problems 1 and 3, as well as any of the bonus problems you choose to do. The report should include any figures and intermediate results that you think may help. Make sure to include explanations of any issues that you may have run into that prevented you from fully solving the assignment, as this will help us determine partial credit. The report should also explain any additional image files you include in your solution (see below).
- All of your Matlab code, including code for the bonus problems, as well as a `README` file explaining how to use the code.
- Image files for Problems 1 and 2, showing the various fusion results you are asked to generate.
- Image files that you create in Problem 3, showing the original flash and no-flash pairs you capture with your camera, and the fused images you generate from them. You can also include additional image files for various experiments (e.g., fusion with different values) other than your final ones, if you think they show something important.
- If you do Bonus Problem 4: The original flash/no-flash images you captured and the corresponding fused image files you produced.
- If you do Bonus Problem 5: The original and corresponding processed image files you used.

Please organize your solution submission using the following file structure:

```
.zip
├── .pdf ..... The PDF report.
├── src/ ..... Contains all Matlab M-files and the README file explaining how to use the code.
└── data/ ..... Contains all image, video, and other data files.
```

## Hints and Information

- Whenever you compute image gradients in this homework, you should do it by either convolving with the forward-difference kernels, or using Matlab's `diff` function. Do *not* use the function `gradient`, as it implements central differences, which will result in inconsistencies with the Laplacian kernel used for the Poisson solver.
- During conjugate gradient descent, make sure to only compute the divergence  $\nabla \cdot \nabla \Phi^*$  once, instead of recomputing it in every iteration.
- When performing convolutions in the gradient-domain processing part of the homework, you should be careful about the various padding arguments Matlab provides. Given the type of boundary conditions you are using, you should always do convolution with the `'same'` argument.
- When dealing with color images, you should apply all of the algorithms you implement to each color channel separately.
- The algorithms in this homework can work with both RAW and rendered (e.g., PNG or JPEG) images. Typically, working with RAW will give you better looking results, but in that case you will also need to use `dcraw` and tonemapping. We leave it up to you to decide what images to use.

## Credits

Some inspiration for this assignment and the write-up came from Oliver Cossairt's computational photography course at Northwestern University. The lamp and museum flash/no-flash pairs came from Petschnigg et al. [5] and Agrawal et al. [1], respectively.

## References

- [1] A. Agrawal, R. Raskar, S. K. Nayar, and Y. Li. Removing photography artifacts using gradient projection and flash-exposure sampling. *ACM Transactions on Graphics (TOG)*, 24(3):828–835, 2005.
- [2] P. Bhat, C. L. Zitnick, M. Cohen, and B. Curless. Gradientshop: A gradient-domain optimization framework for image and video filtering. *ACM Transactions on Graphics (TOG)*, 29(2):10, 2010.
- [3] F. Durand and J. Dorsey. Fast bilateral filtering for the display of high-dynamic-range images. In *Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '02, pages 257–266, New York, NY, USA, 2002. ACM.
- [4] P. Pérez, M. Gangnet, and A. Blake. Poisson image editing. In *ACM Transactions on graphics (TOG)*, volume 22, pages 313–318. ACM, 2003.
- [5] G. Petschnigg, R. Szeliski, M. Agrawala, M. Cohen, H. Hoppe, and K. Toyama. Digital photography with flash and no-flash image pairs. *ACM transactions on graphics (TOG)*, 23(3):664–672, 2004.