EECS 395/495: Introduction to Computational Photography

Homework 3: Flash/No Flash Photography

Student Name: Haikun Liu

Student Number: 2903021

NetID: hlg483

Objective: To fuse together images captured with and without a flash

1. Write an android program to capture a flash/ no flash pair

Using the following code, we can configure the screen color of the tablet. When *color* is valued with the codeword of 'white', we can use the bright screen as a flash to capture the image with flash. When *color* and the codeword of 'black' are identical, we can take photos in a dim scene.

```
final int color = 0xXXXXXXXX;
final Drawable face_color = new ColorDrawable(color);
main_frame.setForeground(face_color);
```

We can use `captureJPEG()` to capture .jpg images with or without flash.

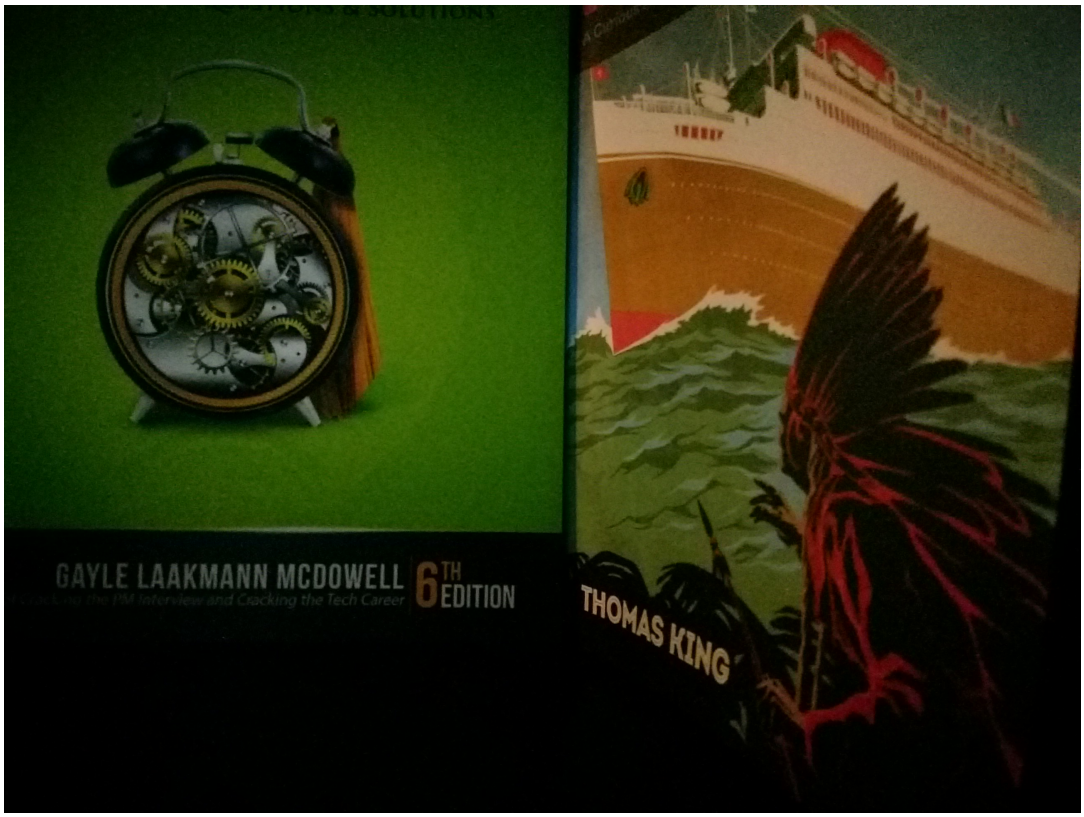

Figure 1. No-flash image: noisy but with correct color

Figure 2. Flash image: low noise but biased color

2. Denoise the flash and no-flash images

Using a bilateral filter, we can denoise the flash and no-flash image with proper parameter settings. There are two parameters of the Gaussian kernel affect the quality of denoised result: sigma_s is the standard deviation in spatial domain and sigma_r is the standard deviation in intensity domain. Each picture is cropped into a 1024*768 image for faster processing; and each color channel is denoised separately.

```
%Load images
[Im_w,map] = imread('/Users/HKLHK/Desktop/2015 Fall
Quarter (NU)/EECS495 Introduction to Computational
Photography/HW/HW3/Images/white.jpg');
Im_w = double(Im_w);
whiteIm = imcrop(Im_w,map,[95 145 1023 767]);%crop the photo
center at (1400,700) with size 1024*768

[Im_b,map] = imread('/Users/HKLHK/Desktop/2015 Fall
Quarter (NU)/EECS495 Introduction to Computational
Photography/HW/HW3/Images/black.jpg');
Im_b = double(Im_b);
blackIm = imcrop(Im_b,map,[95 145 1023 767]);%crop the photo
center at (1400,700) with size 1024*768
```

In order to finely tune the sigma_s and sigma_r, we use sigma_s = 1,4 and 16, and sigmal_r = 0.05, 0.10 and 0.20 for testing.

```
%Finding sigma_s and sigma_r for no-flash image


sigma_s_t = [1 1 1 4 4 4 16 16 16];
sigma_r_t = [0.05 0.10 0.20 0.05 0.10 0.20 0.05 0.10 0.20];
figure;
for j=1:9

for i=1:3
```

```
    blackIm_Denoise_t(:,:,i) =
bilateralFilter(blackIm(:,:,i),sigma_s_t(j),sigma_r_t(j
)*max(max(blackIm(:,:,i))));%denoise each color channel
separately
end
subplot(3,3,j);
image(uint8(blackIm_Denoise_t));
xlabel(['sigma_s = ' num2str(sigma_s_t(j)) ', sigma_r ='
num2str(sigma_r_t(j))]);
end

%choosing sigma_s = 16 and sigma_r = 0.1 for no-flase image

sigma_s_b = 16;
sigma_r_b = 0.1;

for i=1:3
    blackIm_Denoise(:,:,i) =
bilateralFilter(Im_b(:,:,i),sigma_s_b,sigma_r_b*max(max
(Im_b(:,:,i))));
end

%denoise parameter testing for flash image
sigma_s_t = [1 1 1 4 4 4 16 16 16];
sigma_r_t = [0.05 0.10 0.20 0.05 0.10 0.20 0.05 0.10 0.20];
figure;
for j=1:9

for i=1:3
    whiteIm_Denoise_t(:,:,i) =
bilateralFilter(whiteIm(:,:,i),sigma_s_t(j),sigma_r_t(j
)*max(max(whiteIm(:,:,i))));
end
subplot(3,3,j);
image(uint8(whiteIm_Denoise_t));
xlabel(['sigma_s = ' num2str(sigma_s_t(j)) ', sigma_r ='
num2str(sigma_r_t(j))]);
end
```

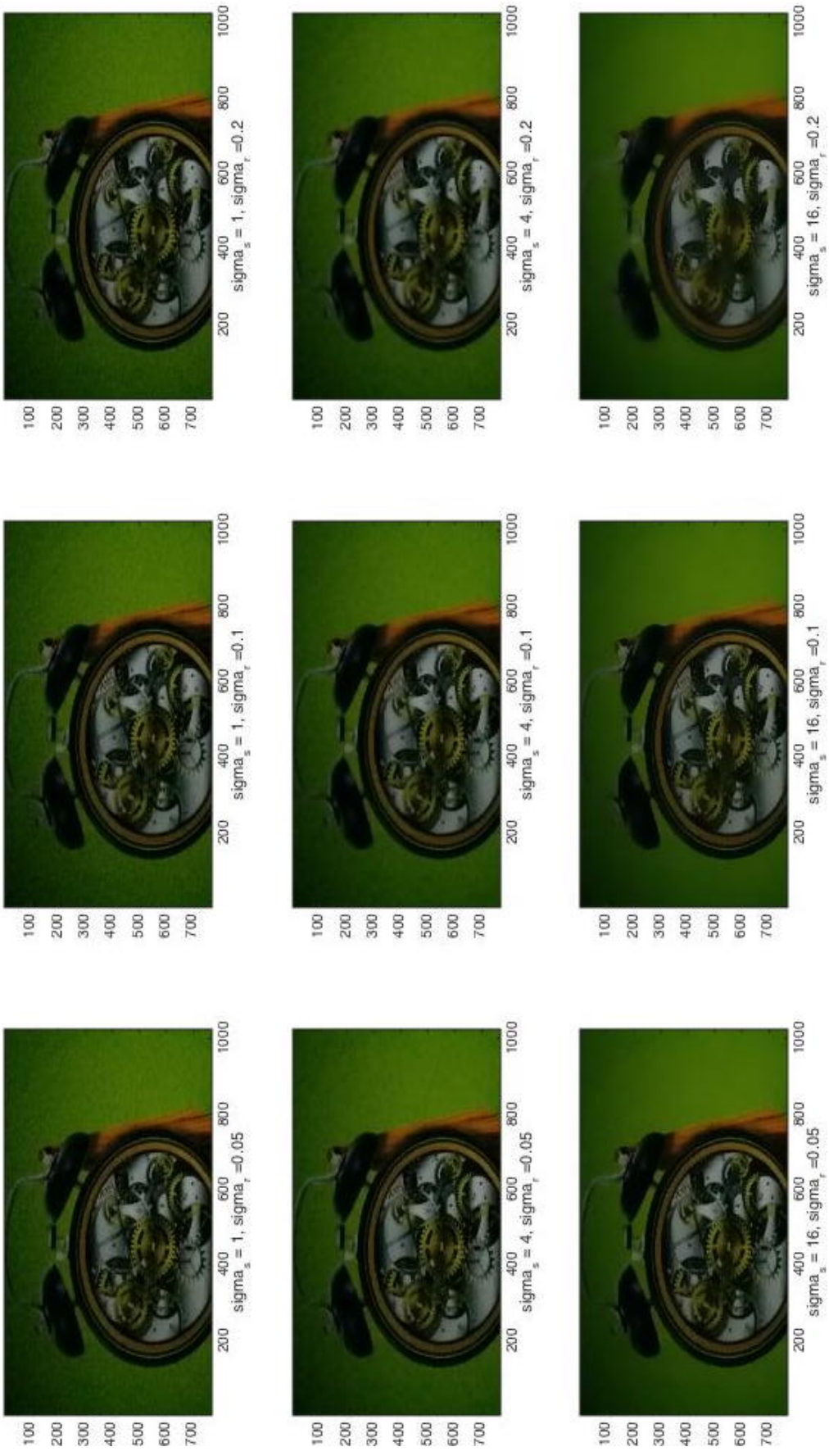

Figure 3. Denoised no-flash image using different parameter settings

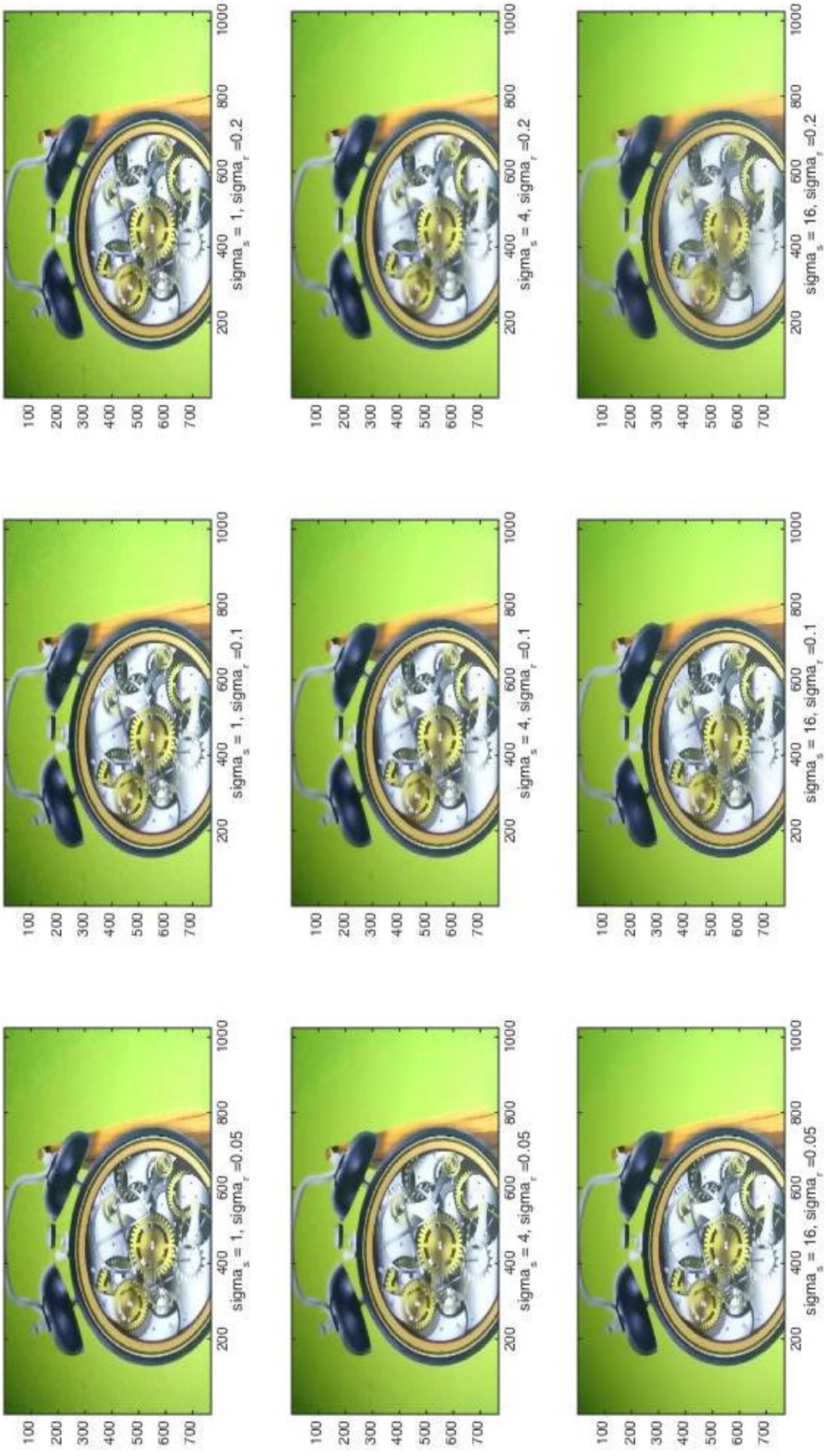

Figure 4. Denoised flash image using different parameter settings

Based on the results in Figure 4 and Figure 5, we choose sigma_s = 16 and sigma_r = 0.10 for denoising both no-flash and flash pictures.

```
%choosing sigma_s = 16 and sigma_r = 0.1 for no-flase image

sigma_s_b = 16;
sigma_r_b = 0.1;

for i=1:3
    blackIm_Denoise(:,:,i) =
bilateralFilter(Im_b(:,:,i),sigma_s_b,sigma_r_b*max(max
(Im_b(:,:,i))));
end

%choosing sigma_s = 16 and sigma_r = 0.1 for flase image

sigma_s_w = 16;
sigma_r_w = 0.1;

for i=1:3
    whiteIm_Denoise(:,:,i) =
bilateralFilter(Im_w(:,:,i),sigma_s_w,sigma_r_w*max(max
(Im_w(:,:,i))));
end
```

3. Fuse the images together

According to the formula below, we can fuse the flash image F, the denoised flash image Fd and the denoised no flash image Ad toghther into a detailed true color picture.

$$A_f = A_d * \frac{F+\epsilon}{F_d+\epsilon}, \quad \epsilon = .02$$

```
%fuse image

epsilon = 0.02;
```

```
fusedIm =
blackIm_Denoise.*((Im_w+epsilon)./(whiteIm_Denoise+eps
ilon));
```

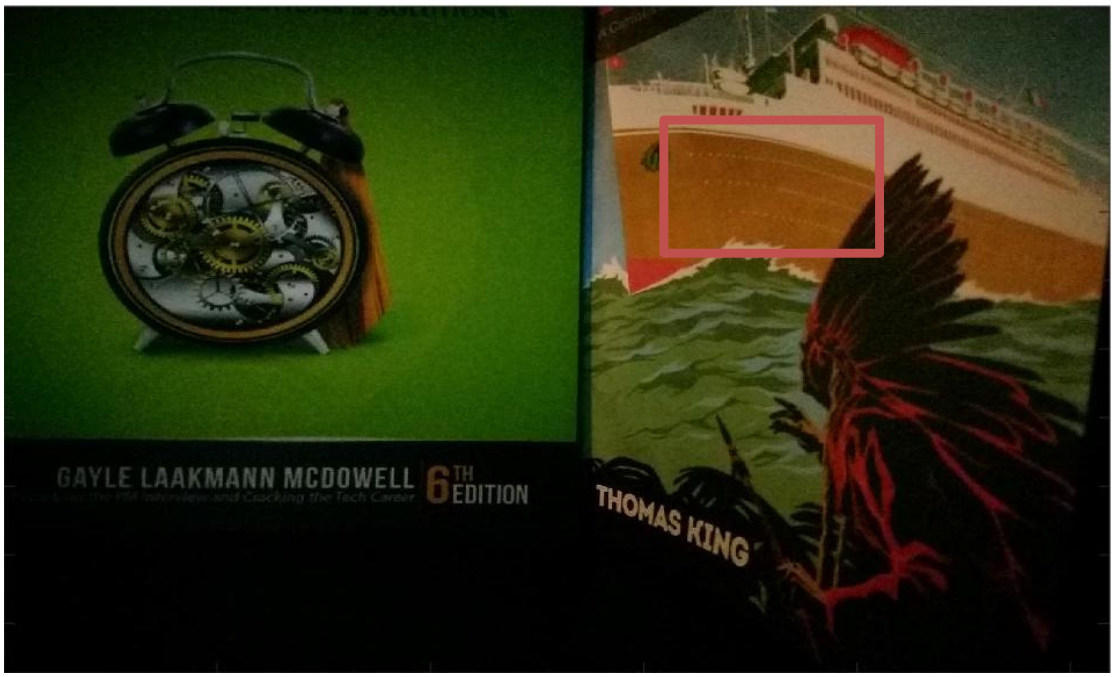

Figure 5. The original no flash image

Figure 6. Closeup images of the original no flash image

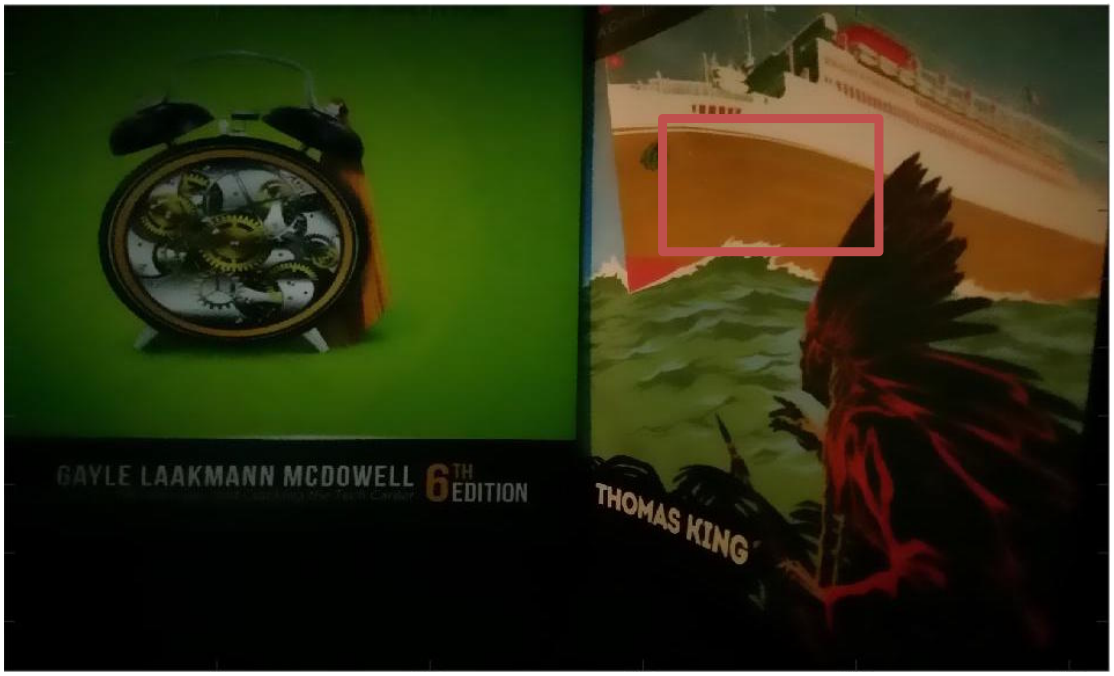

Figure 7: The denoised result of the no flash image

Figure 8. Closeup images of the denoised no flash image

Figure 9. The fused result combining the denoised image from

Figure 10. Closeup images of the fused image

Notice that the fused image has the same lighting and color characteristics as the original no-flash image with more details compared with denoised no-flash image.