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

Answer:
Android Studio Codes

```
Log.e(TAG, "clicked flash/noflash Capture. ");
//TODO:hw3
//Set the black color code
//Basic Steps:
//1. set the foreground of whole screen to black, snap a picture
//2. set the foreground of whole screen to white, snap a picture after 200 ms delay
//3. set the foreground to be transparent (null) after 400 ms delay

//TODO:set the foreground to be black
final int black_color = 0xFF000000;
final Drawable f_black_color = new ColorDrawable(black_color);
main_frame.setForeground(f_black_color);
Log.e(TAG, "setting black color");
captureJPEG();
//TODO:take picture as jpg


final int white_color = 0xFFFFFFFF;
final Drawable f_white_color = new ColorDrawable(white_color);


//TODO: set the foreground to be white with 200 ms delay
Handler handler = new Handler();
handler.postDelayed(new Runnable() {
    public void run() {
        Log.e(TAG, "setting white color");
        main_frame.setForeground(f_white_color);
        captureJPEG();
        //TODO: take picture as jpg
    }
}, 200);
//set the foreground to be transparent with 400 ms delay
handler.postDelayed(new Runnable() {
    public void run() {
        main_frame.setForeground(null);
        //TODO: take picture as jpg
    }
}, 400);
//same thing as repaint
invalidateOptionsMenu();
```

In the original codes, there is a sentence of notation “//TODO: take picture as jpg” when the foreground is transparent. But I didn`t add capturing a photo order there, because on the course website it says “All you will need to do is to change the background of the screen and then capture a white background image and a black background image.” Figure 1. and Figure 2. are images of a flash image and no-flash image respectively.

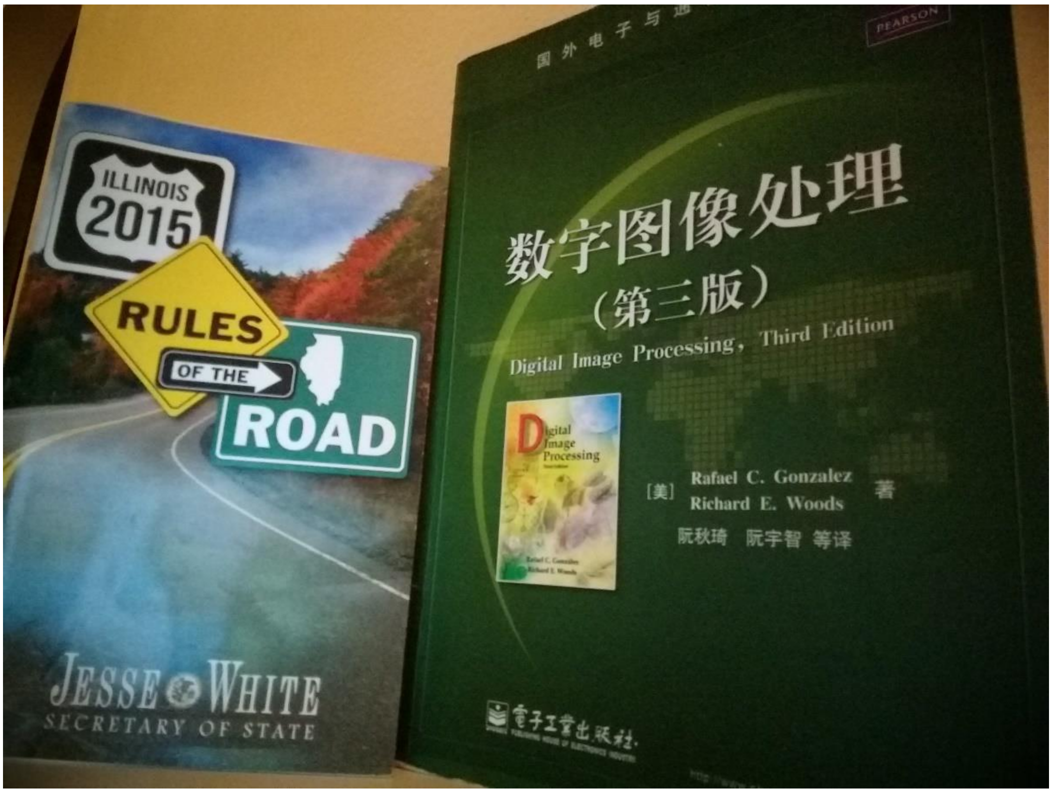

Figure 1. flash image

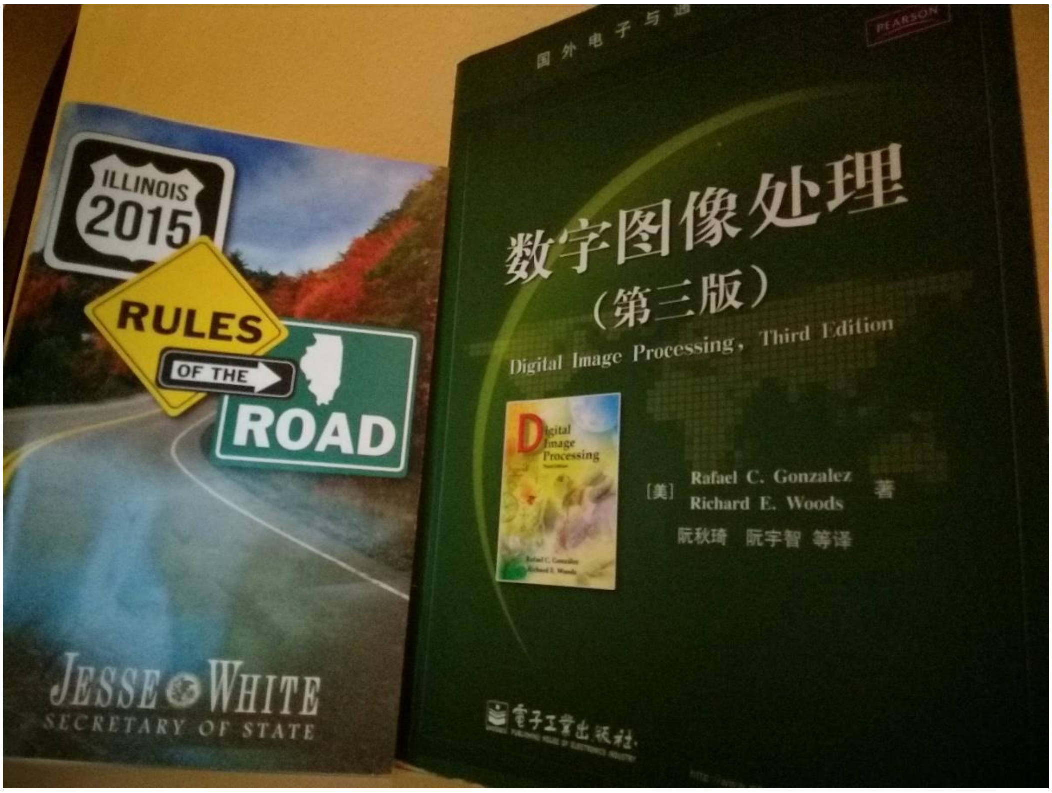

Figure 2. no-flash image

Question 2 &3. Denoise the no-flash image you captured in part 1 and extract the details from the flash image and fuse the images together.

Answer:

Matlab Codes

```
%%%%%% 1st step comparing different settings %%%%%%
clear;
clc;
image=imread('D:\Courses Files\Introduction to Computational Photography\HW3\original photos\flash.jpg');

image=im2double(image);
imager=image(:,:,1);
imageg=image(:,:,2);
imageb=image(:,:,3);

sigmas=[1 4 16 32 64 128];
sigmar=[0.05 0.1 0.2];

outr=bilateralFilter(imager,sigmas(1),sigemar(1));
outg=bilateralFilter(imageg,sigmas(1),sigemar(1));
outb=bilateralFilter(imageb,sigmas(1),sigemar(1));
out(:,:,1)=outr;
out(:,:,2)=outg;
out(:,:,3)=outb;
figure(1),imshow(out);

%%%%%% 2nd step fusing images %%%%%%

clear;
clc;
Fd=imread('D:\Courses Files\Introduction to Computational Photography\HW3\Denoised_flash\f 128 0.05.jpg');
F=imread('D:\Courses Files\Introduction to Computational Photography\HW3\original photos\flash.jpg');
Ad=imread('D:\Courses Files\Introduction to Computational Photography\HW3\Denoised_noflash\nf 128 0.05.jpg');

Fr=(F(:,:,1)+0.02)./(Fd(:,:,1)+0.02);
Fg=(F(:,:,2)+0.02)./(Fd(:,:,2)+0.02);
Fb=(F(:,:,3)+0.02)./(Fd(:,:,3)+0.02);

Af(:,:,1)=Ad(:,:,1).*Fr;
Af(:,:,2)=Ad(:,:,2).*Fg;
Af(:,:,3)=Ad(:,:,3).*Fb;
```

```
imshow(Af);
imwrite(Af,'Af.jpg','jpeg');
```

In this part, I tried $\sigma_s = [1,4,16,32,64,128]$ and for set $\sigma_r = [0.05,0.1,0.2]$ for each σ_s. From images of different experiments, it`s obvious that the quality of image decreases with σ_r increasing and increases with σ_s increasing. Some results of different setting experiments are shown below to verify the conclusion.

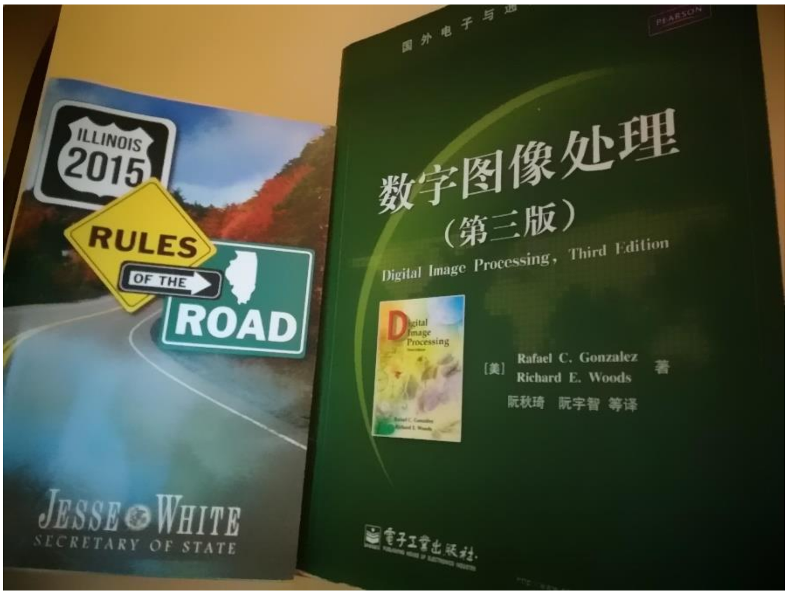

Figure 3. $\sigma_s = 32$ $\sigma_r = 0.05$ flash

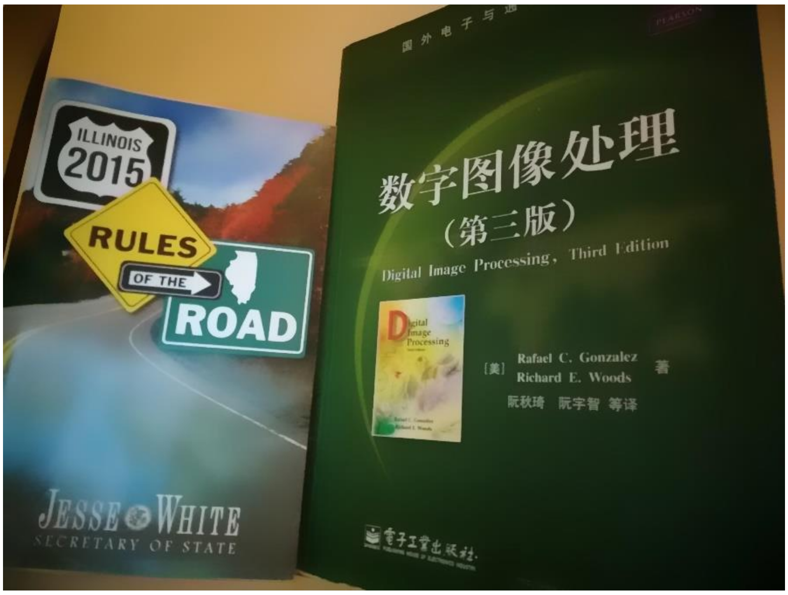

Figure 4. $\sigma_s = 32$ $\sigma_r = 0.1$ flash

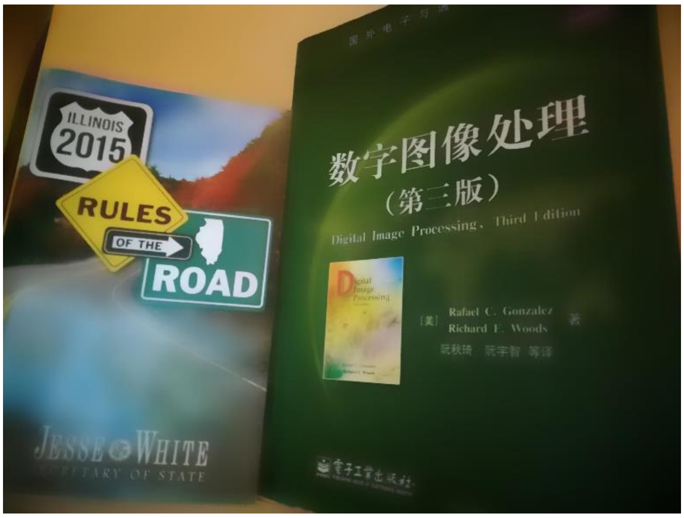

Figure 5. $\sigma_s = 32$ $\sigma_r = 0.2$ flash

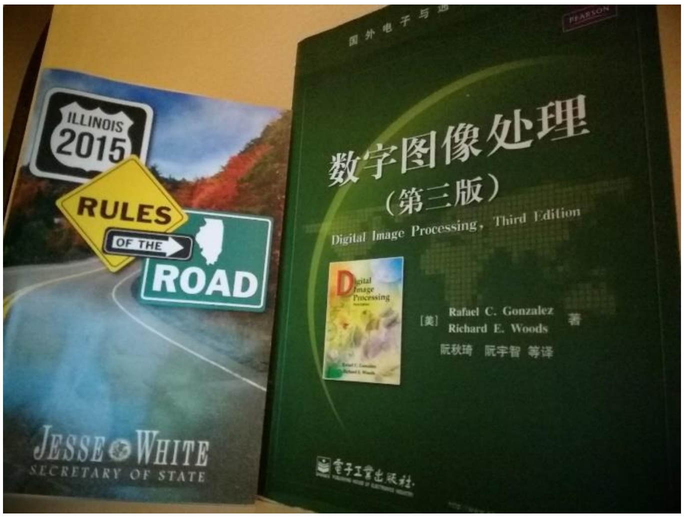

Figure 6. $\sigma_s = 1$ $\sigma_r = 0.05$ flash

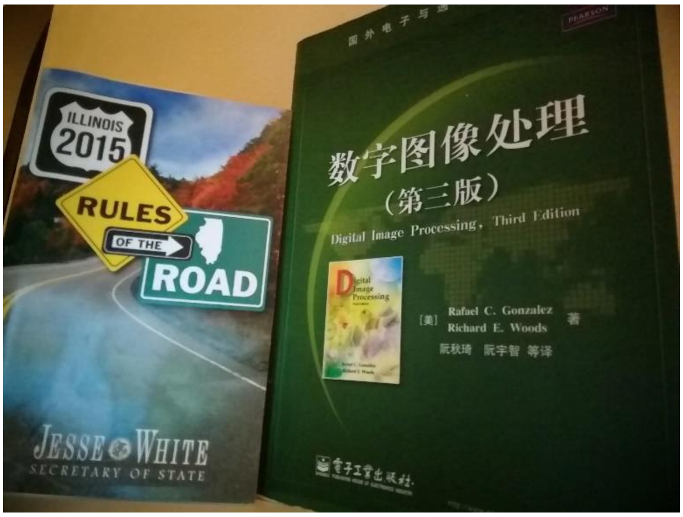

Figure 7. $\sigma_s = 4$ $\sigma_r = 0.05$ flash

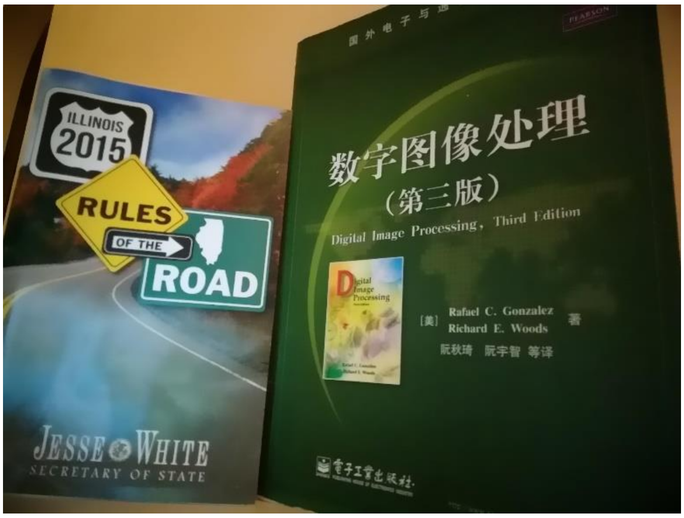

Figure 8. $\sigma_s = 16$ $\sigma_r = 0.05$ flash

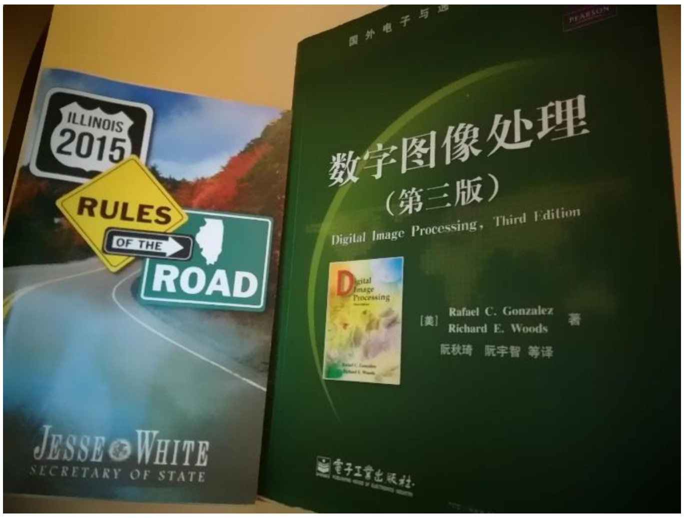

Figure 9. $\sigma_s = 32$ $\sigma_r = 0.05$ flash

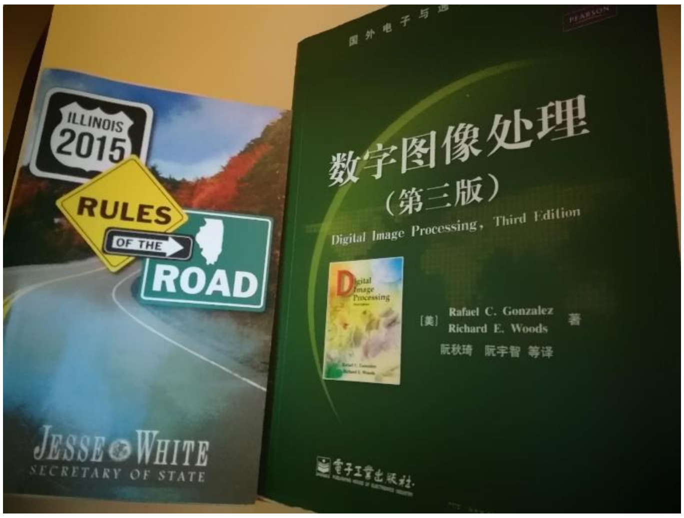

Figure 10. $\sigma_s = 64$ $\sigma_r = 0.05$ flash

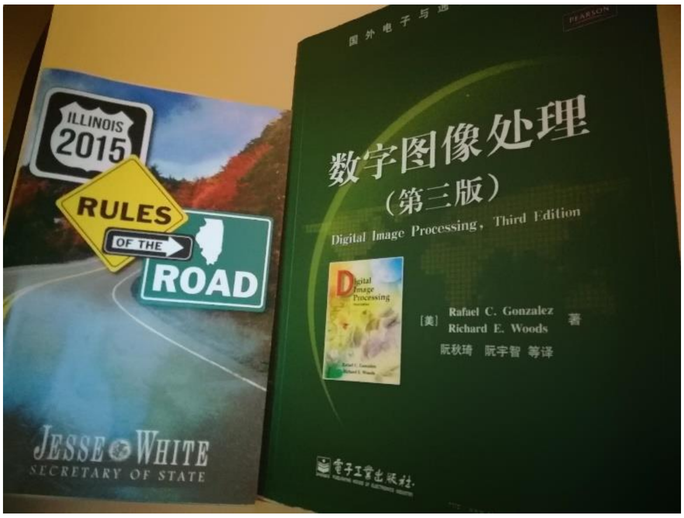

Figure 11. $\sigma_s = 128$ $\sigma_r = 0.05$ flash

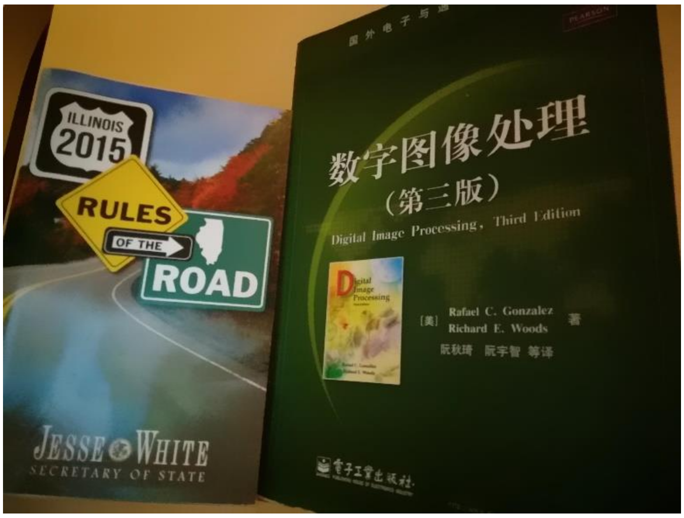

Figure 12. $\sigma_s = 32$ $\sigma_r = 0.05$ no flash

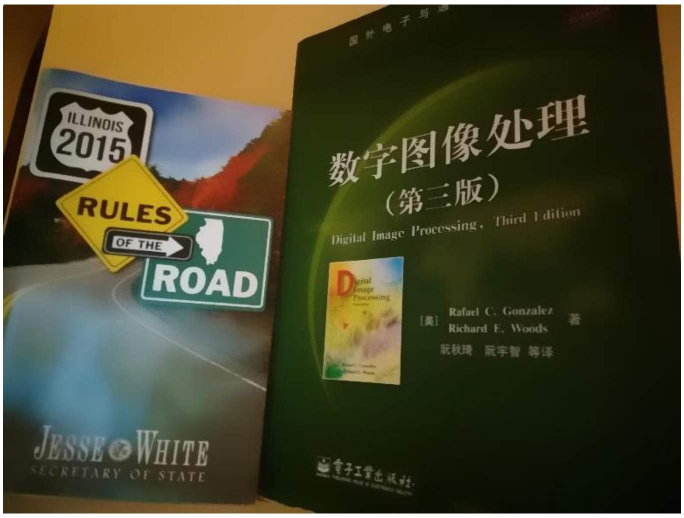

Figure 13. $\sigma_s = 32$ $\sigma_r = 0.1$ no flash

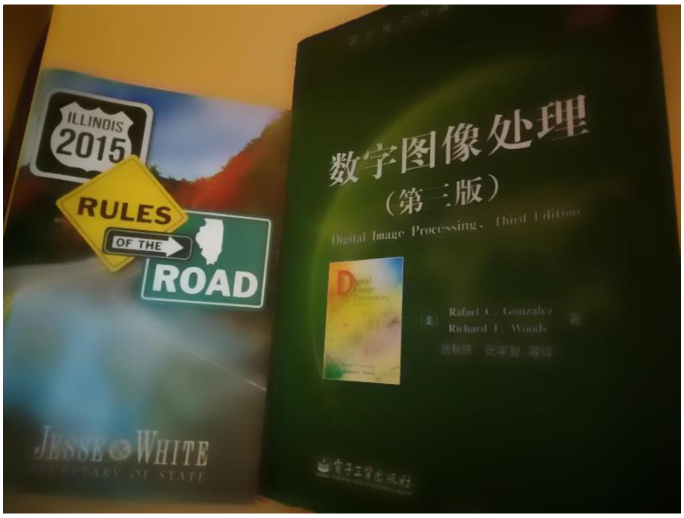

Figure 14. $\sigma_s = 32$ $\sigma_r = 0.2$ no flash

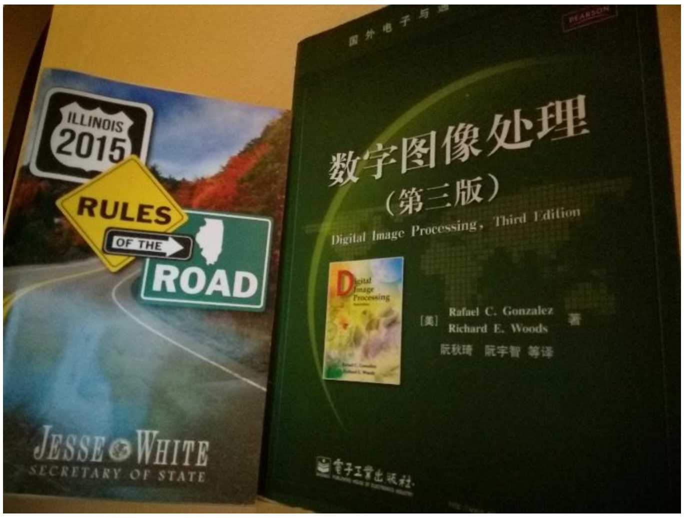

Figure 15. $\sigma_s = 1$ $\sigma_r = 0.05$ no flash

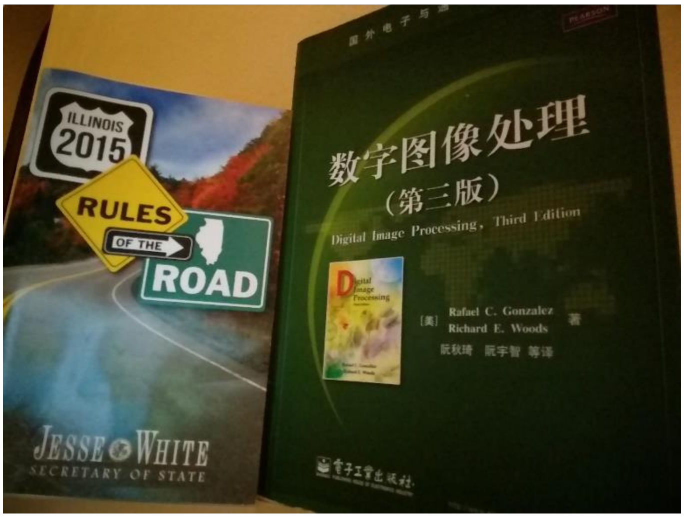

Figure 16. $\sigma_s = 4$ $\sigma_r = 0.05$ no flash

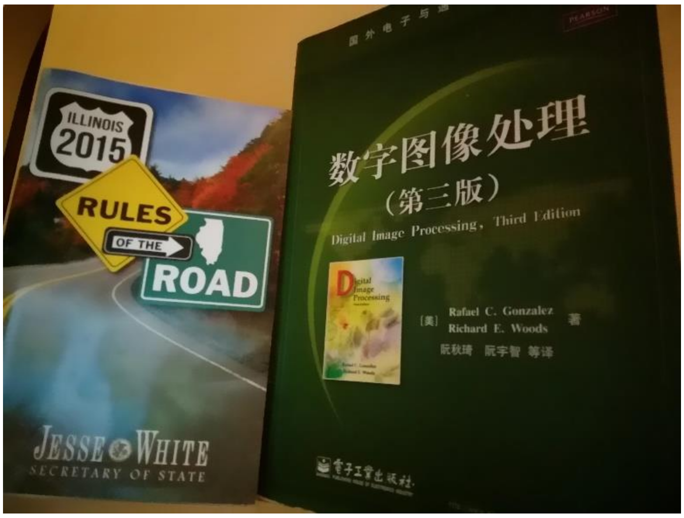

Figure 17. $\sigma_s = 16 \;\; \sigma_r = 0.05$ no flash

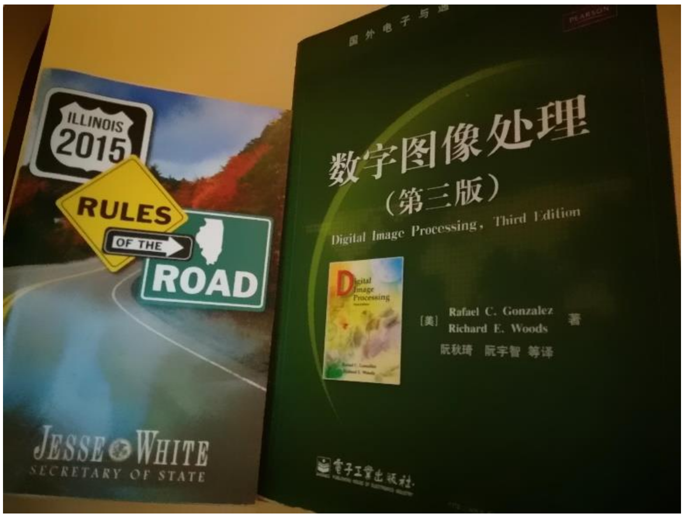

Figure 18. $\sigma_s = 32 \;\; \sigma_r = 0.05$ no flash

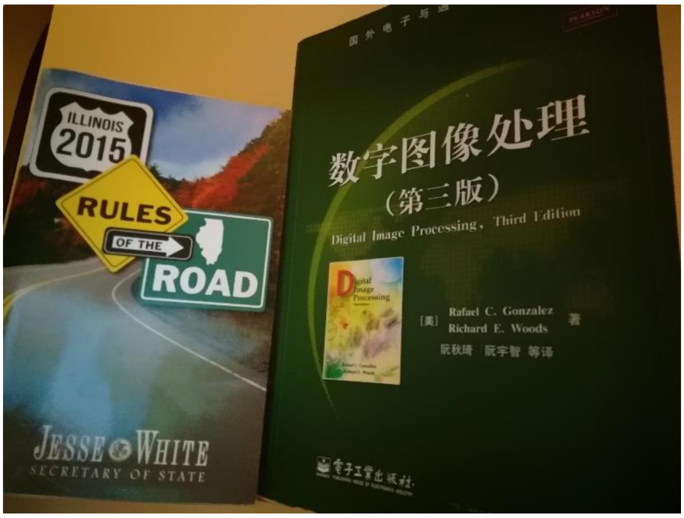

Figure 19. $\sigma_s = 64 \;\; \sigma_r = 0.05$ no flash

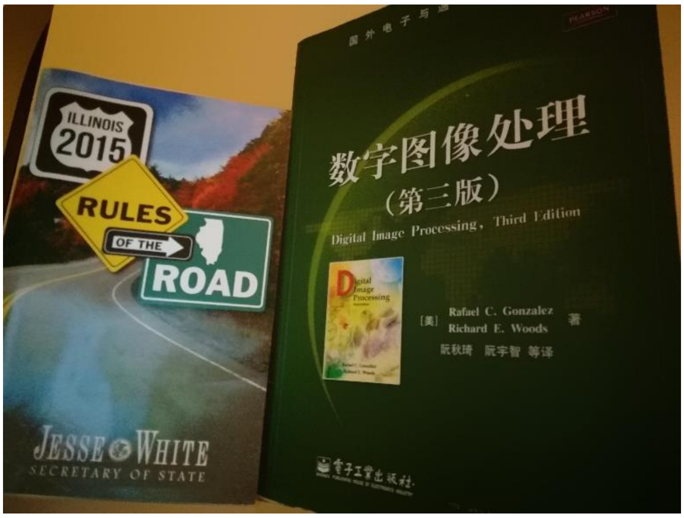

Figure 20. $\sigma_s = 128$ $\sigma_r = 0.05$ no flash

According to the compared result, I chose $\sigma_s = 128$ $\sigma_r = 0.05$ for both flash image and no-flash image. The final fused image is Figure 21.

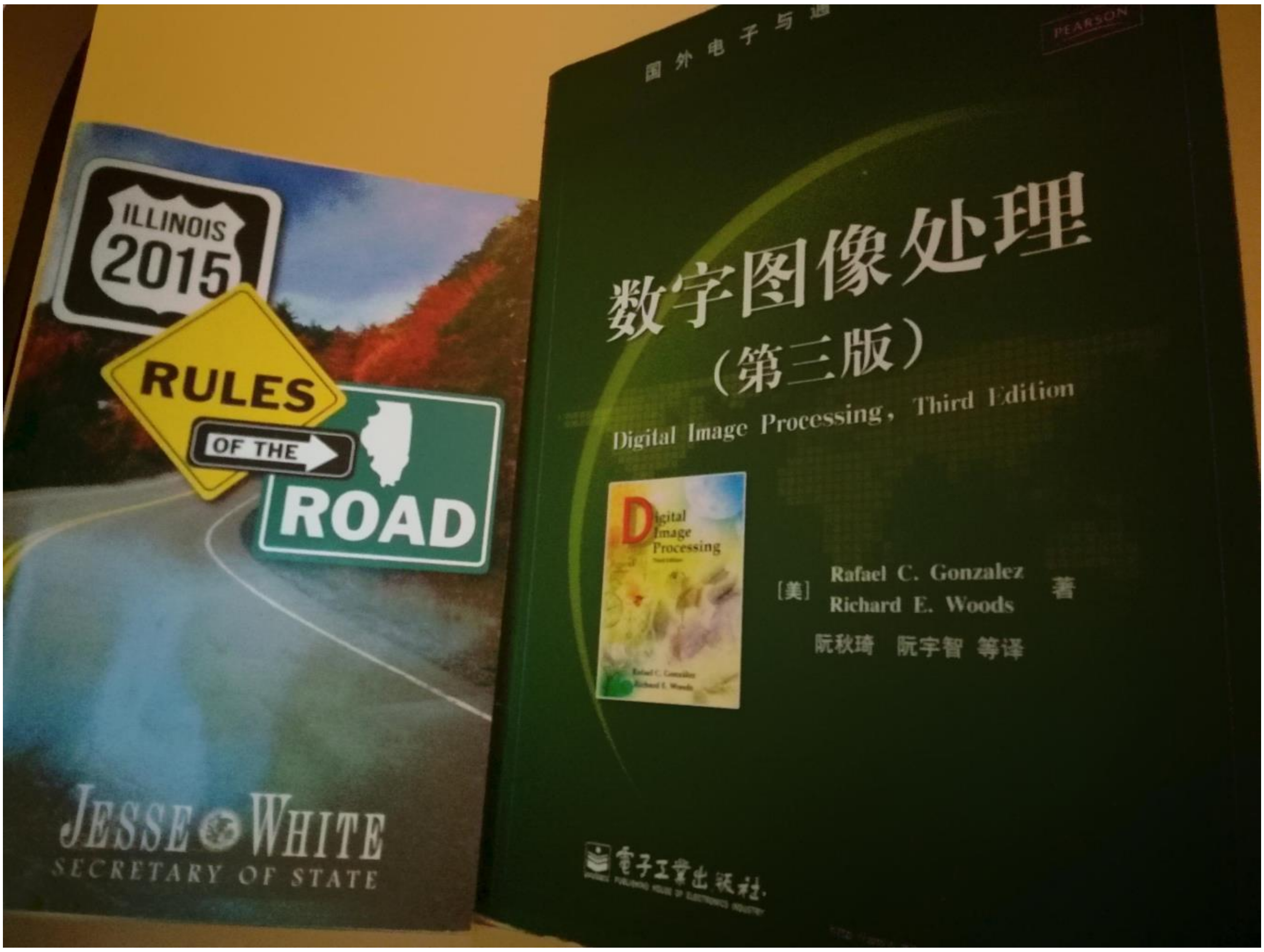

Figure 20. final fused image

To compare the setting I chose and the default setting(which means image is the only input parameter for the function bilateralFilter,so sigmaRange and samplingSpatial are determined by the function), Figure 21 is the fused image by images of flash and no-flash processed with default sigmaRange and samplingSpatial.

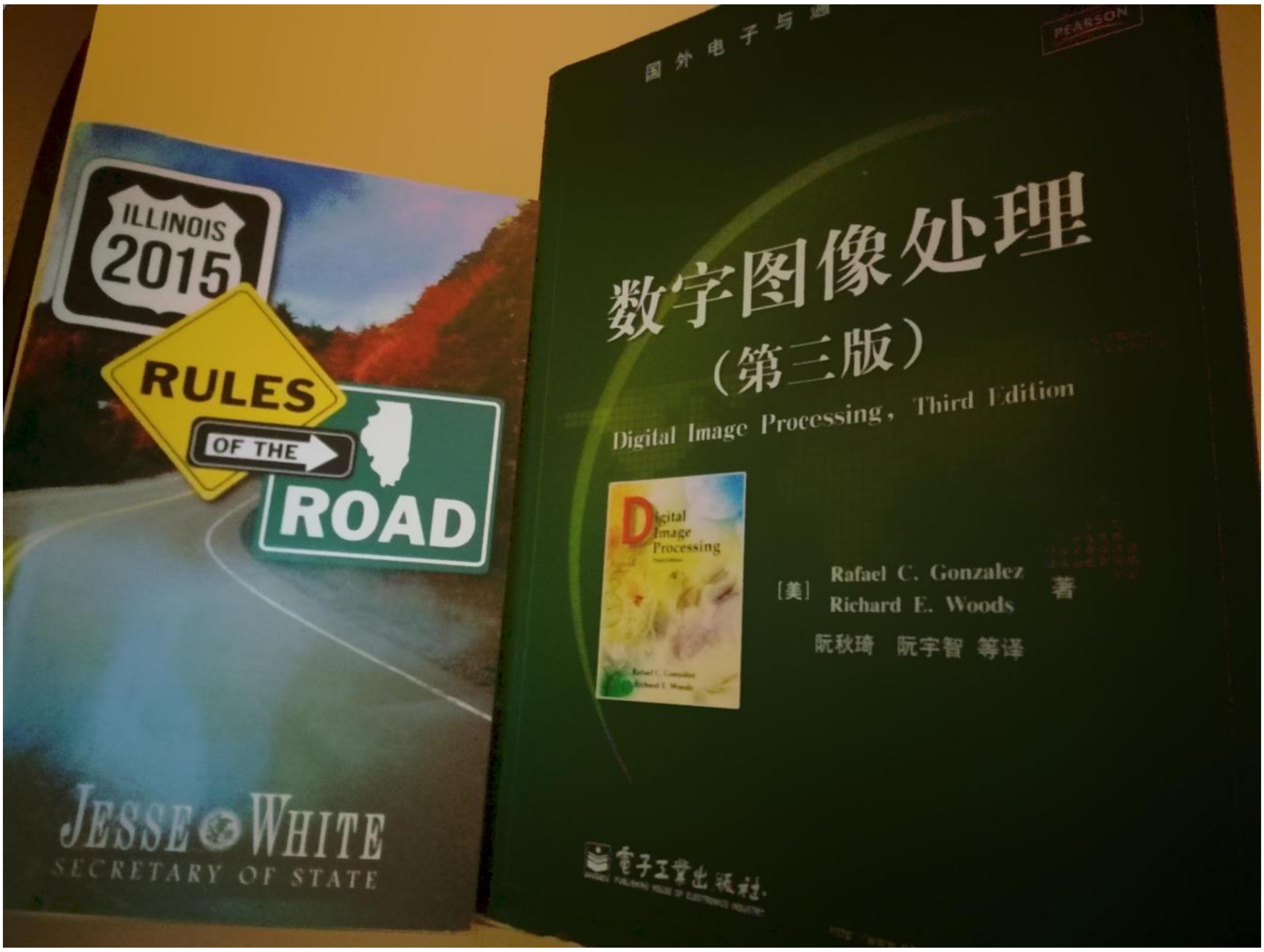

Figure 21. fused image of default setting

From Figure 21 we can observe some errors of color in some parts of the image, so I think the setting I chose is better than the default one.