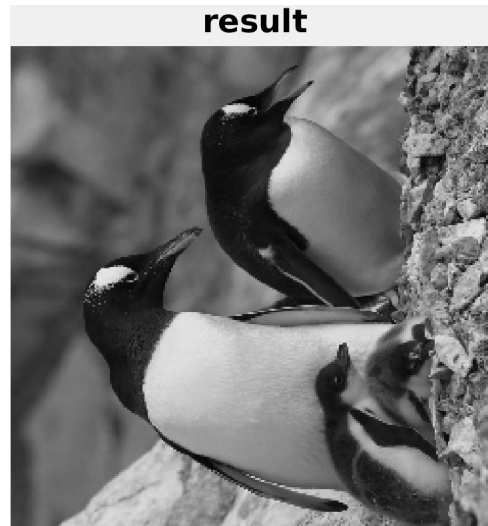


## DIP HW1

資工三 戴培倫 B04902105

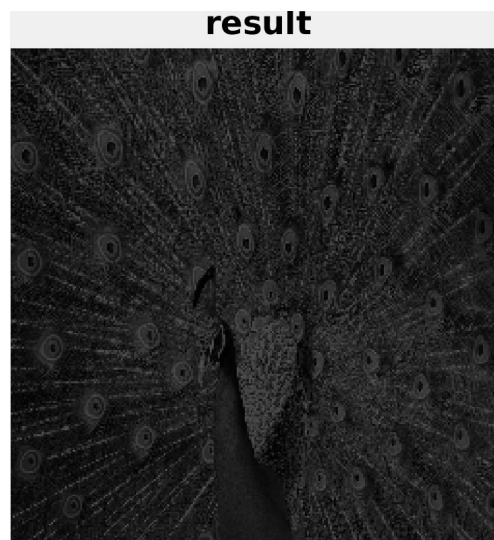
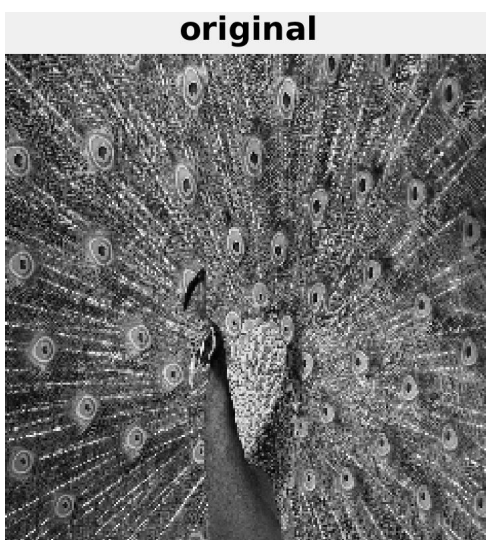
### WARM-UP



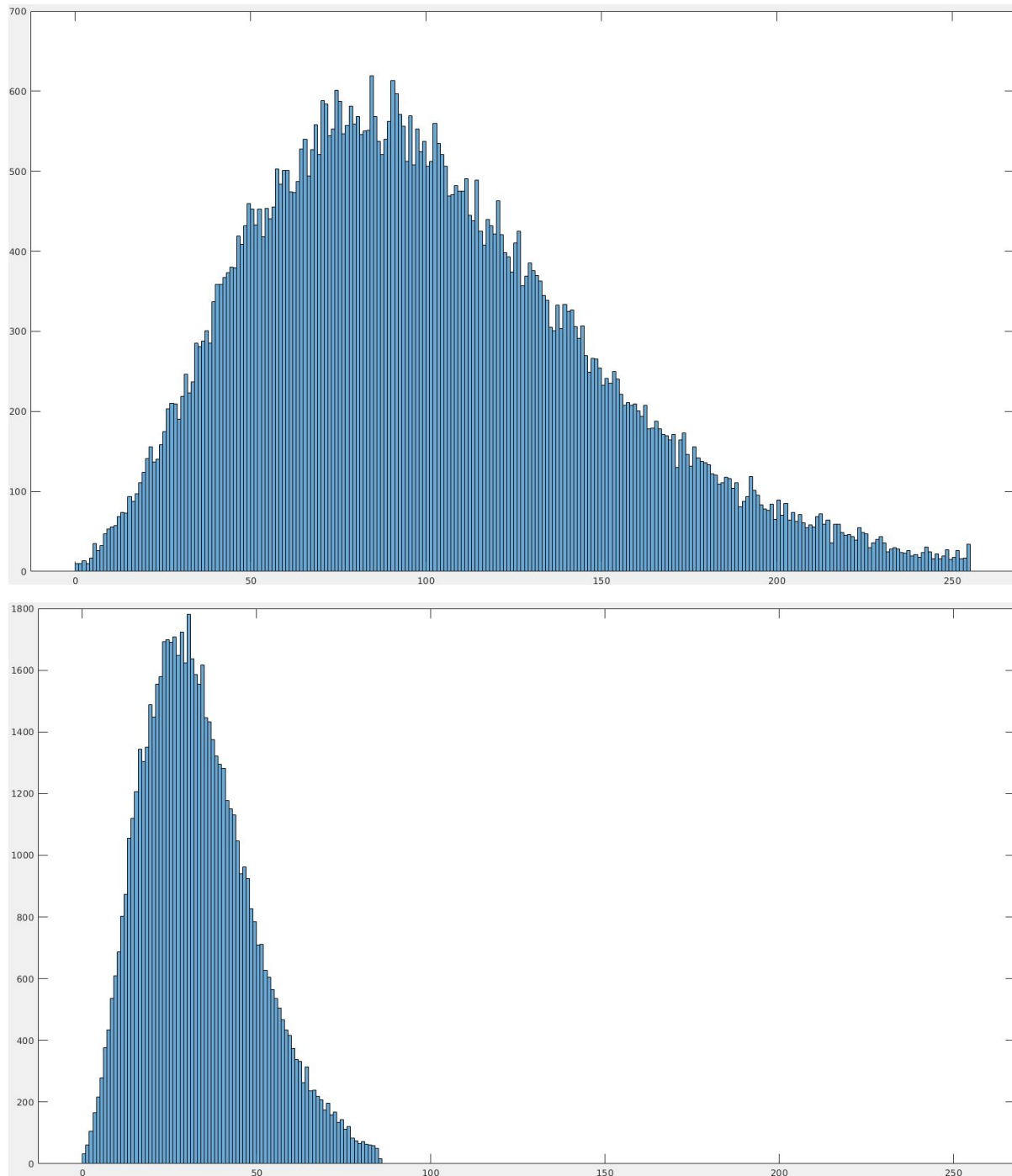
Pixel value =  $(0.3 R) + (0.59 G) + (0.11 B)$   
Flipped along the  $(0,0),(255,255)$  axis.

### Problem 1

(a)

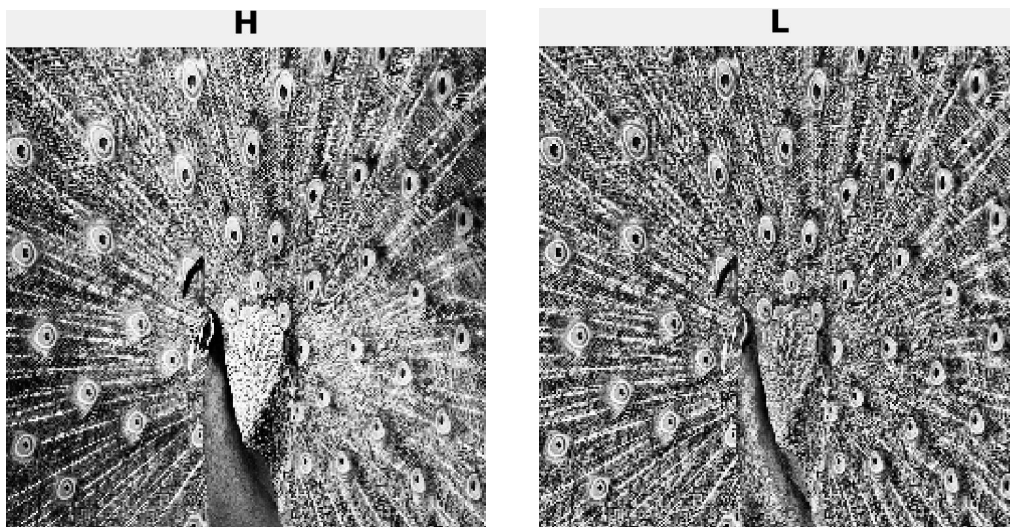


**(b)**



It's obvious that the whole histogram shifted left, the max pixel value dropped to  $255/3$ . The height of the histogram also increases.

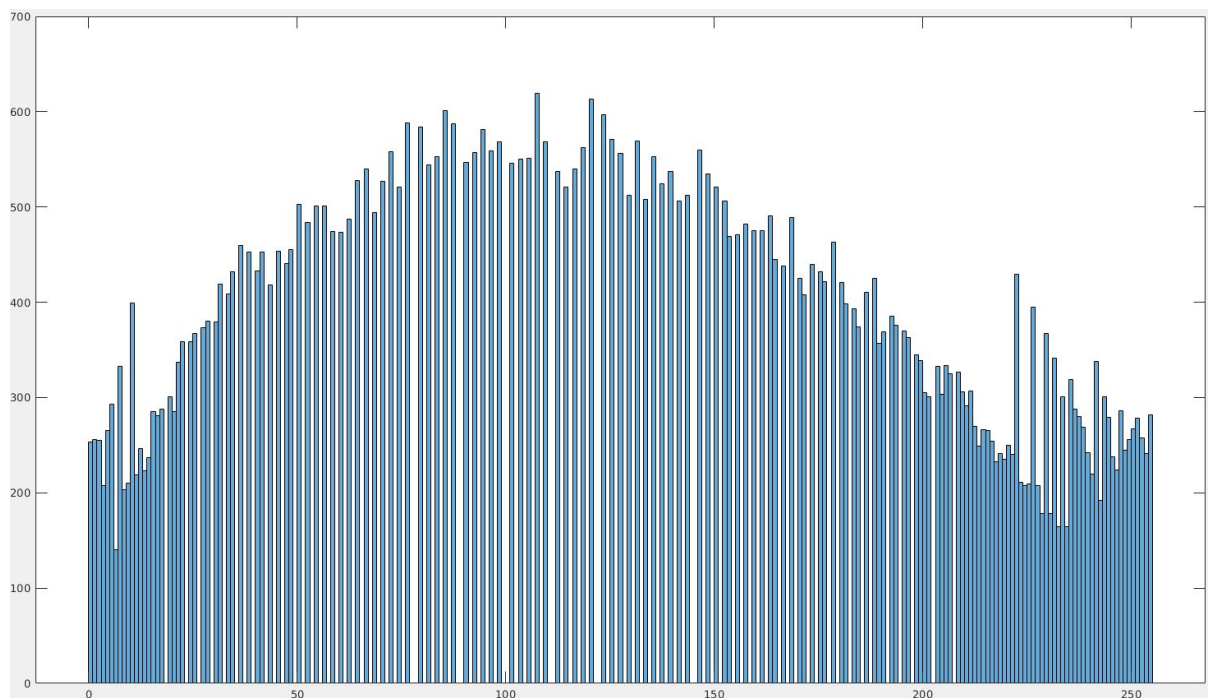
(c)(d)



Local histogram equalization (winsize=21)

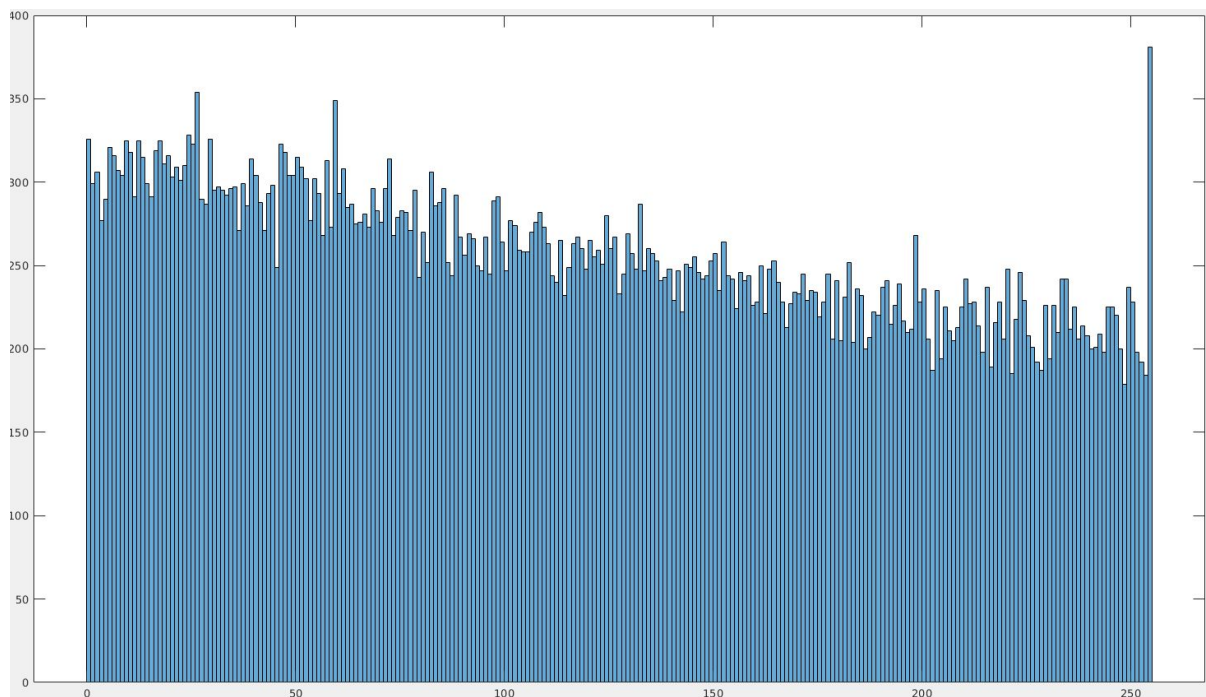
(e)

Histogram equalization



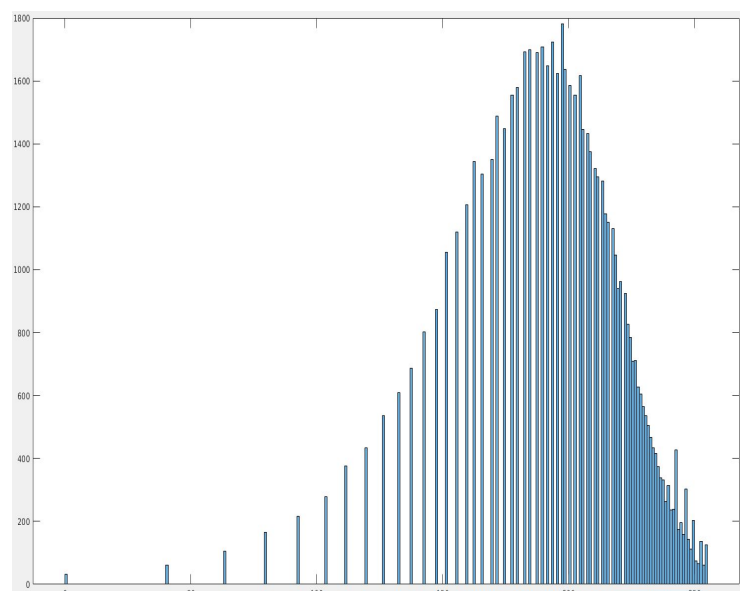
Since there are more pixels landing in between value 50~150, and that I equalize them by  $\text{cdf} \times 255$ , so there are more empty slots in the middle of the histogram.

## Local Histogram Equalization



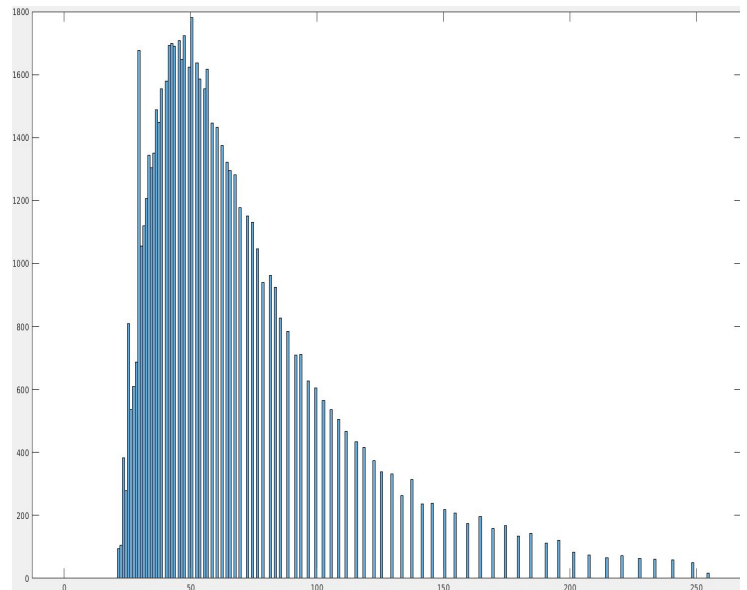
Since I took window size=21, so there are at most 441 pixels ranking between 0~255. That is why there is no empty slot, And it seems to be more union than the result of histogram equalization.

**(f)**

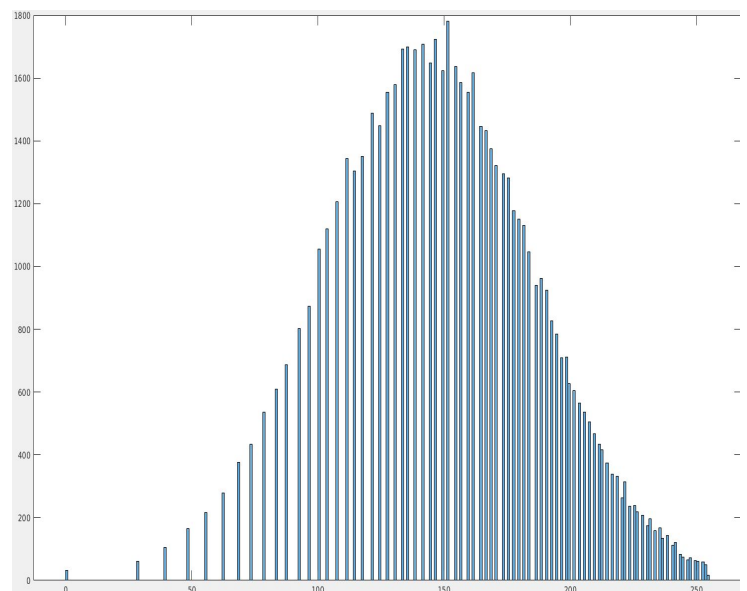
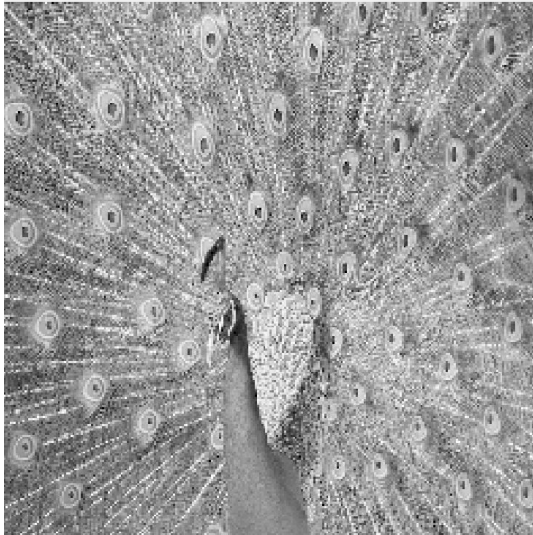




**inverse log transform**



**power-law transform**



Log Transform: take natural log of pixel value + 1 (prevent (log 0))  
Inverse Log Transform: take the exponential of base 1.03  
Power-law Transform: take the square root of all pixel values (gamma=0.5)  
Three of them then divide all pixels by the max value and times 255 to make the max value of the image 255.

## Problem 2

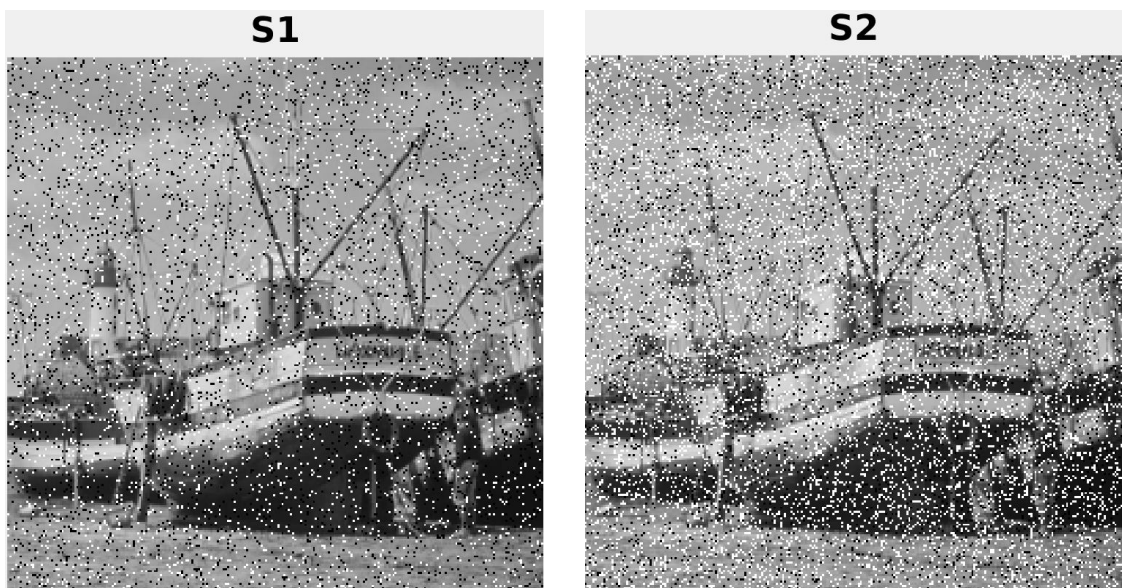
(I)

(a)



I add some gaussian noise with (mean=0, sigma=10) and (mean=0, sigma=30), then clip the value to (0,255). The image with sigma=10 seems a little blurred, but the color doesn't have much difference than the original one. However, we can see significant difference between the image with sigma=30 and the original one since the large variance allows the noise to have large value.

(b)



S1: salt 5%, pepper 5%

S2: salt 15%, pepper 5%

In S1, we can see there are about the same amount of salt and pepper.

In S2, there are obviously more salt than pepper.

(c)



I use low pass filtering on G1 to try to moderate the noise. I use a 3 by 3 filter with value 15 at the center and value 1 for the rest. I found that setting the center value between 10 and 30 has the best result. PSNR even drops when the center value is between 2 and 5. From the result of denoising, we can see that the noise seems to merge into the background, however, the edges of the objects are slightly blurred. I use median filtering on S1 to denoise the image. The result seems really good. And that filter with size 3 leads to best result.

(d)

**G1: PSNR = 28.138959db**

**RG: PSNR = 29.988778db**

**S1: PSNR = 15.296708db**

**RS: PSNR = 28.173759db**

(II)

**Original**

**Result**



I tried the outlier method with  $13 \times 13$  window and  $\epsilon=5$ . While implementing, I skip the pixels who have more than 50 neighbors whose pixel value is less than 100. That's because I want to avoid the eye, which is the darkest part in the image. After that, I use the low-pass filter method with filter  $\{\{1, 1, 1\}, \{1, 5, 1\}, \{1, 1, 1\}\}$  in order to smoothen the wrinkles.