

# **Tone Adjustment / Smoothing filter tool**

Diliprao Boinapally and Princy Razafimanantsoa  
Portland State University

## **Abstract**

This paper presents a tone mapping and filtering tool. It allows the user to concurrently apply several tone mapping and filters to a given image. The user adjusts each one of them with a slider. Since this project's original goal was to implement an interactive local tone adjustment tool our deliverable contains a brush allowing the user to paint a color on a designated region of the image. The main contribution of this project is to gather several tone mapping and filters in one tool.

## **Introduction**

On the market of digital photography software Adobe Photoshop is king. This tool has a feature named Camera Raw. It allows the user to apply several tone mapping to an inputted image. The first goal of this class project was to implement a tool similar to Camera raw while including the interactive local tone adjustment found in the paper: "Interactive local adjustment of tonal values. Lischinski, Dani and Farbman, Zeev and Uyttendaele, Matt and Szeliski, Richard. ACM SIGGRAPH 06". Due to time constraint we switched to a global tone adjustment and filtering tool. However, we included a brush stroke feature for a future development. This brush implements the region selection of the previous paper using a basic brush.

First we are to talk briefly about the above-mentioned paper. Second we will discuss about the method used to build the tool before showing experiments. Finally we will conclude with some future work.

## **Related Work**

This work was based on "Interactive local adjustment of tonal values. Lischinski, Dani and Farbman, Zeev and Uyttendaele, Matt and Szeliski, Richard. ACM SIGGRAPH 06". In this paper the authors presents an interactive local tone adjustment. This tool allows a user to adjust the tone of a particular region of the image. This is performed by a roughly selecting the desired region with a brush. The user applies the tone mapping, which is spread smoothly in the image. This means the user don't have to manually select a region of the image to apply a local tone adjustment. Furthermore, this can be applied not only to tonal correction but also to filters. Our work is global instead of local and includes more filters.

## **Method**

The first part of our method was to implement the tone mapping and the filters. To do so we used OpenCV. This set of library provides useful methods to perform the

desired tone mapping and filters. In our case we implemented the solution in java using Eclipse. These are the tone mapping / filters included:

### Brightness and Contrast

The math formula for brightness and contrast is:

$$g(x) = \alpha f(x) + \beta$$

$g(x)$  is the outputted image

$f(x)$  is the original image

$\alpha$  is the variable for the contrast

$\beta$  is the variable for the brightness

We use sliders to adjust the values of  $\alpha$  and  $\beta$ . In order to set up only one of the parameter we have to assign a fixed value to the other one. If we want to adjust the brightness and not the contrast we assign  $\alpha$  to "1". To adjust the contrast while keeping the same brightness  $\beta$  is set to "0".

### Box filter

The box filter allows the image to be smoothed/blurred. It applies a box kernel to every pixel of the image:

$$K = \frac{1}{K_{\text{width}} \cdot K_{\text{height}}} \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & 1 & 1 & \dots & 1 \\ \cdot & \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \cdot & \dots & \cdot \\ 1 & 1 & 1 & \dots & 1 \end{bmatrix}$$

The user controls the size of this kernel with a slider.

### Gaussian filter

The Gaussian filter uses a kernel with a Gaussian "shape". Like the previous one, a slider controls the kernel size. This filter has to more parameters: standard deviation in X and Y. These are computed using the kernel. We chose that option to simplify the user input.

### Median filter

Likewise this filter is controlled over its kernel size using one slider. It denoises the image by setting each pixel with the median value of its neighborhood.

### Bilateral filter

Like the Gaussian filter the bilateral filter has two extra parameters: the standard deviation in color and in space. The user input a single standard deviation, which is applied for both of the extra parameter. The kernel size is always set to 9 as recommended by the javadoc.

We included a part of the Interactive local tone adjustment tool in our deliverable: the basic brush. Basically the user can draw a brush stroke on top of the image. In the paper [1] this specifies the region selected by the user. Internally it assigns a

weight to each pixel. In the case of the basic brush the weight is “1”. The paper provides three more kind of brush.

### **Experiment**

To test our prototype we used several images (see Appendix 1). For the small images (512x512 and 440x595) the tool is doing pretty well. For a bigger image (3624 × 2407) the filters are a little bit slowed down.

### **Conclusion**

To conclude we did a tone adjustment and smoothing tool. We implemented 2 tone mapping and 4 filters. As a future work we may implement the full interactive tone adjustment tool. We already have the region selection part. The missing piece is the constraint propagation, which involves a resolution of a linear equation. This may be done in OpenCV.

### **Bibliography**

[1] Lischinski, Dani and Farbman, Zeev and Uyttendaele, Matt and Szeliski, Richard. ACM SIGGRAPH 06

## Appendix 1



Figure 1 small image (512x512)



Figure 2 small image (440x595)



Figure 3 big image (3624 × 2407)