Aleksandr Rozenman
Jovan Ruano
Steven Wirsz
Nov 12, 2013
Group A2

Image Cleanup Tool

SUMMARY

For our pragmatic experience, we created an automatic image correction tool which utilized three filters. Our software project has expanded this to a fully featured image cleanup tool designed around a graphical user interface. The program can load images, save images, and actively apply 8 different video filters as the user moves slider bars around. Three of the filters were custom built by our team. Although OpenCV filers have no "undo feature", we've designed the interface to partially and proportionally apply all of these filters to the image in real time allowing the user to see or reverse changes instantly.

After implementing this software tool using Microsoft Visual C++ 2010 and OpenCV version 2.4.6.0, we then experimented with a large number of images discovering how the filters could be applied to repair certain types of damaged or distorted images. Although many types of issues are simply unresolvable, our program was surprisingly effective at resolving a large number of problems and the flexibility of the user interface we designed made it very easy to experiment to find the most effective combination of filters to resolve certain types of issues.

FILTERS

- Bilateral Filter - "Oil painting"

This filter acts both as a sharpening and smoothing function. It enhances high-frequency edges and or blurs low-frequency changes in the image. It is a very effective tool when used in combination with other blurring or smoothing filters to remove noise. We implemented the slider bar control to affect the diameter of the area around each pixel sampled. The larger the diameter, the greater the blurring effect. Colors are generally simplified so at high levels or multiple repetitions this filter makes images look somewhat like an oil painting. (OpenCV: Smoothing)

- Gaussian Blur - "Out of focus"

This filter simply reduces image detail directly or gradually removes all details, high frequency and low-frequency alike. Its effect is similar to looking through a autofocus lens or a thick piece of glass. At very small levels, this blurring can remove noise or gross deformities in an image. Similar to the bilateral filter, the slider bar we implemented affects the radius of the Gaussian function applied to each pixel. (OpenCV: Smoothing)

- Smoothing Function (custom) - "Smeared colors"

This is a custom built filter from our pragmatic project, it performs a 5x5 mean averaging function on the color value of each individual pixel. Three nested for loops run through every row, column, and color of the image matrix and make the resulting image proportionally drawn from the original pixel's color and a certain percentage of the surrounding pixel's color. The slider bar determines which percentage of the pixel value is from surrounding pixels. At 0%, the pixel remains at its original value, and at 100% the pixel color is determined entirely by the mean average of its surrounding pixels. Just like Gaussian blurring, this is a smoothing function which for most images is most effective if only applied a little and in combination with other filters. It works best to slightly soften or "smear" rough or uneven color gradients in an image. In our pragmatic experience, this filter was implemented at the fixed value of 75% of the original / 25% surrounding pixels.

- Proportional Histogram Stretching (custom) – "Contrast Enhancing"

This is a custom-built filter that first converts the image to black and white in order to discard chrominance, then searches for the highest and lowest value luminance values before stretching and scaling the color histogram over the entire luminance range. After these values are extracted, they are applied to all three color matrices equally, so the color balance of the resulting image is not changed. The slider bar we created determines the percentage that the histogram is stretched from

their original values. Images with a narrow range of luminance values will be stretched by a greater degree but if the image already covers the entire luminance range from 0 - 255 this filter will do nothing. If thresholding filters are applied to image before this filter is applied, this filter is made more effective. This filter introduces small gaps in the histogram, so applying a small degree of smoothing afterward is recommended.

- Color and Contrast Histogram Stretching (custom) - “Color Correction”

This is a custom-built filter similar to the proportional contrast histogram stretching filter we wrote, but instead of scaling all colors equally, the histogram stretching is applied to the 3 RGB matrices separately. This has the result of equalizing any color imbalance in the image if a particular color is unrepresented or correcting for situations with poor lighting, overexposure, underexposure, and many forms of color image distortion. The slider bar is also implemented as a percentage of stretching from their original values. Just like proportional stretching, if the image covers the entire luminance range from 0 - 255 it will do nothing. Also like proportional stretching, this filter is most effective if applied after threshold filters, and a small degree of smoothing will remove slight color issues afterwards.

- Add Gaussian Noise

This filter simply adds random "static" noise to the image being examined. The position of the slider determines the amount of noise added. (m19404)

- Threshold Black to Zero

This filter performs the function of simply removing pixels with a dark luminance value. Based on the position of the sliding bar, every pixel with an intensity level below the slider bar’s threshold value is simply set to zero. This has the effect of taking a dirty black background in an image and making it pure black. Besides being useful to remove distortion at the dark end of the image color range, this filter can be used prior to histogram stretching to increase the range of the histogram stretching

performed. The most effective way to use this filter is simply to move the slider until noticeable details are “damaged” and then back off a bit. (OpenCV: Basic Thresholding)

- Threshold White to Zero

Not quite opposite of the black threshold filter, this filter simply creates a cap on the highest possible luminance value in the image. The brightest whites are simply capped at a certain top frequency chosen by the slider bar. This is very useful to remove or downplay any bright white distortion or glare in the image. It also can be used prior to histogram stretching to increase the contrast stretching range. (OpenCV: Basic Thresholding)

COMBINATION FIXES

Originally our goal was to make a more or less automated image repair tool. Our proposal had us originally planning to design a total of four different “1-Click button fixes” or "filter sequences" to remove noise, fix colors, etc. Implementing these sequences would not be difficult, but in the course of our testing and research we found that the level that these filters need to be applied is solely dependent on the picture itself. The degree which filters are applied such as adding blur, smoothing, thresholding the black and white values is extremely subjective for each image and not possible for a computer to automate until computers can subjectively understand what looks good in the image or not. To substitute for this work, we detailed examples of exactly how these filters could be used by a human operator to repair or enhance various types of images.

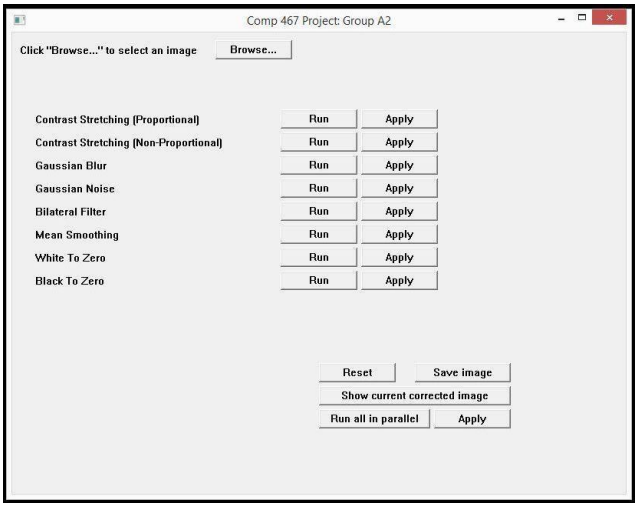
DEVELOPMENT NOTES

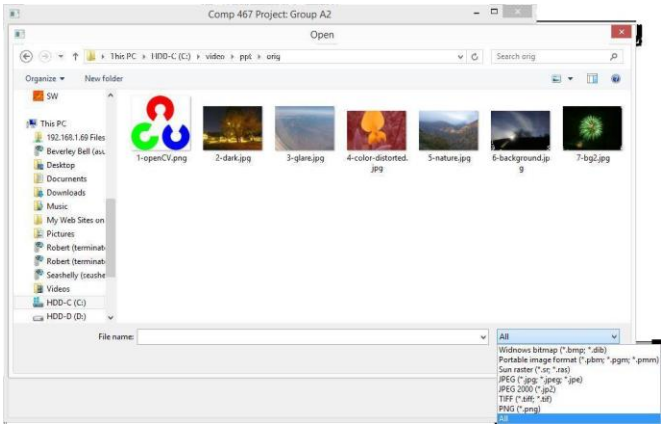
- Slider bars were created by constantly reprocessing to temporary working image matrices. In order to make all filters have some sort of variability to be used in a slider bar required us to mathematically scale many operations.

● Behind the scenes juggling: There is no such thing as undo filter or reapply filter with different settings. Every time the slider bar is moved, the source image is used in the filter applied to create a destination image. Instead of requiring the user to constantly click save changes, refresh, restore, etc. we update the temporary “source” matrix only under certain circumstances invisible to the user in the background. When the user goes to another filter, the destination image is then copied onto the source image making all previous modifications permanent.

● Many images with too little detail cannot be repaired to any degree. The detail has to be there in the surrounding pixels and the contrast range must exist somewhere or the resulting image will not improve from the source image no matter what filters are applied.

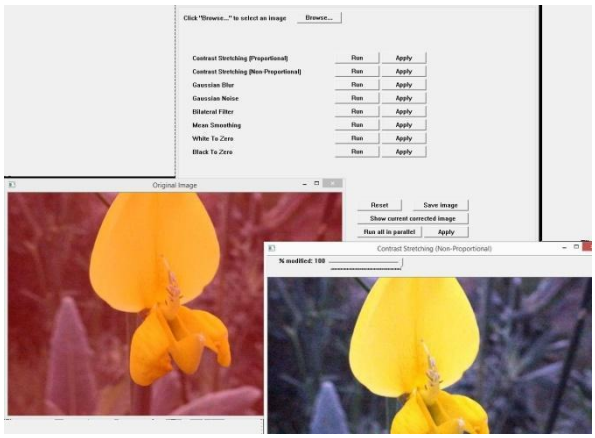
INTERFACE

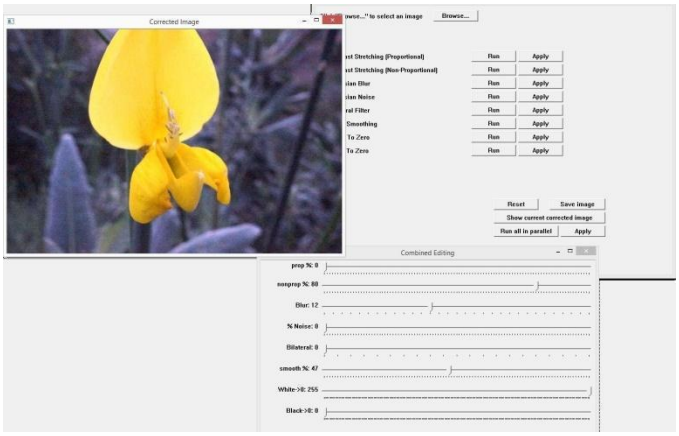




The GUI interface written for this project is very flexible and allows the user to browse an open any file in a number of different file formats to modify and also save the image back to disk when any modifications are complete. Images can modified with single filters, or a single interface can apply multiple filters to an image simultaneously.

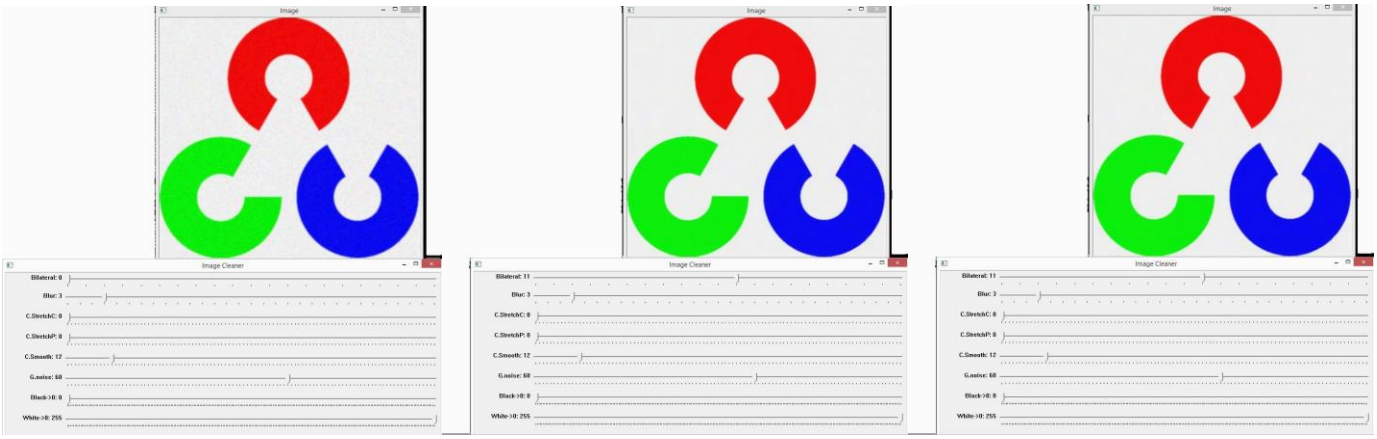
The GUI creates multiple windows that the user can relocate around the desktop in any positional order they want:



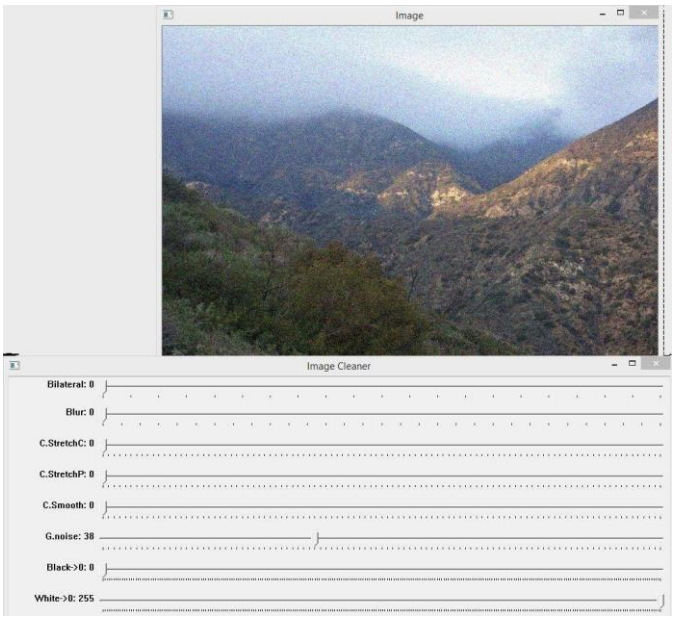


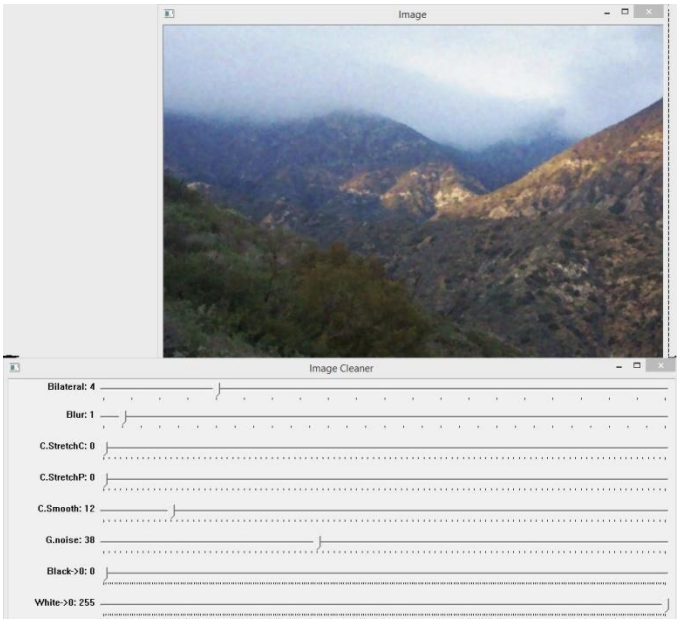
EXPERIMENTS: NOISE REMOVAL

Because this software tool can artificially create image distortion in terms of Gaussian noise, the first test to perform this on a simple image of the OpenCV logo. To remove this noise, we first apply a small degree of both smoothing filters. Because these individual pixels of noise have no correlation to the color of the image surrounding them, these functions reduce them to a far greater degree than other elements image, shifting the noise from high-frequency changes to low-frequency changes. We then then apply a bilateral filter which restores image sharpness somewhat and completely eliminates the lowest frequency distortions. As you can see from these images, this type of static noise can be almost completely removed.



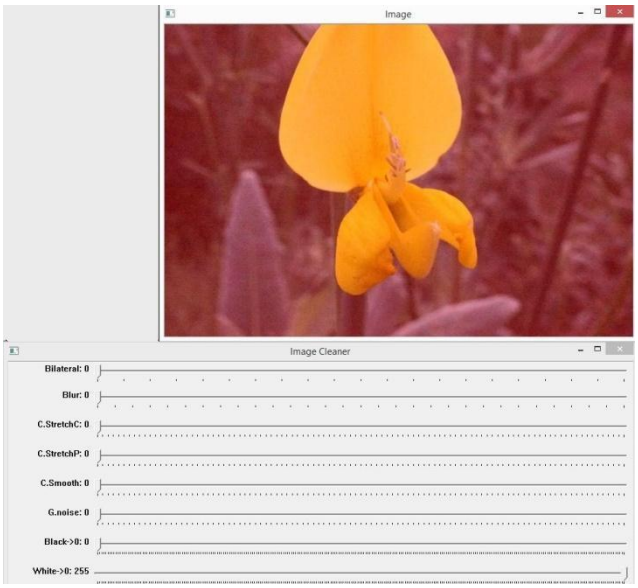
Here we have the same procedure demonstrated on a real image. The only difference here is that there is some loss of low-frequency details with some slight blurring in the final image.

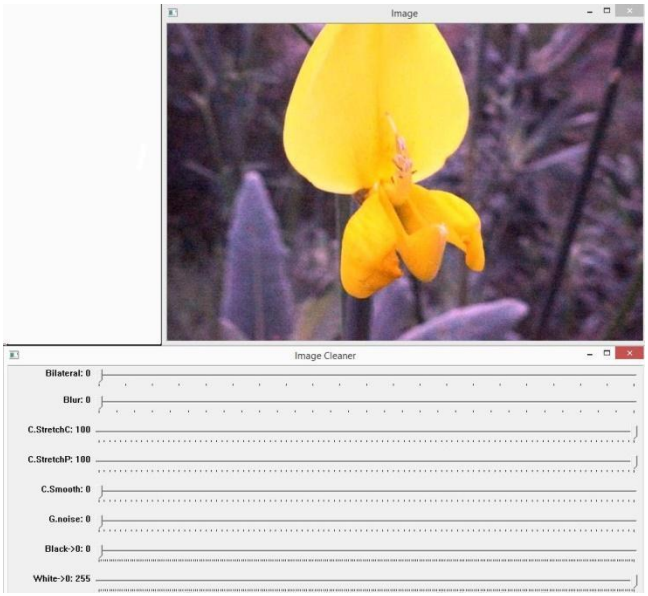




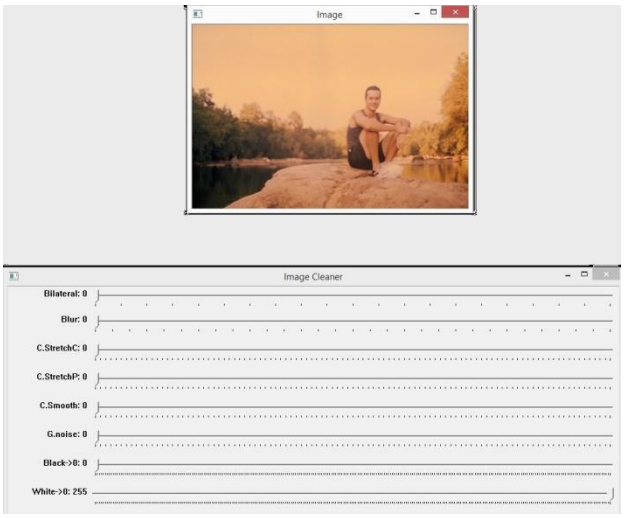
EXPERIMENTS: COLOR CORRECTION

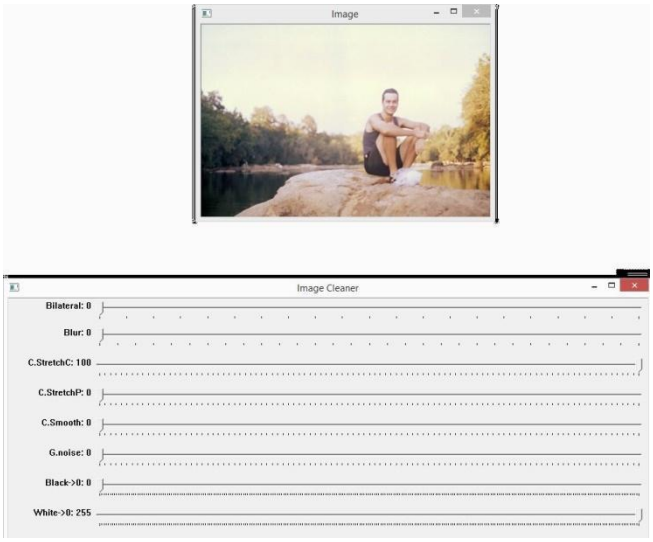
In this image of this distorted flower, the proportional histogram stretching not only brightens the image, but the color histogram correction filter is able to almost fully restore the natural color of the image



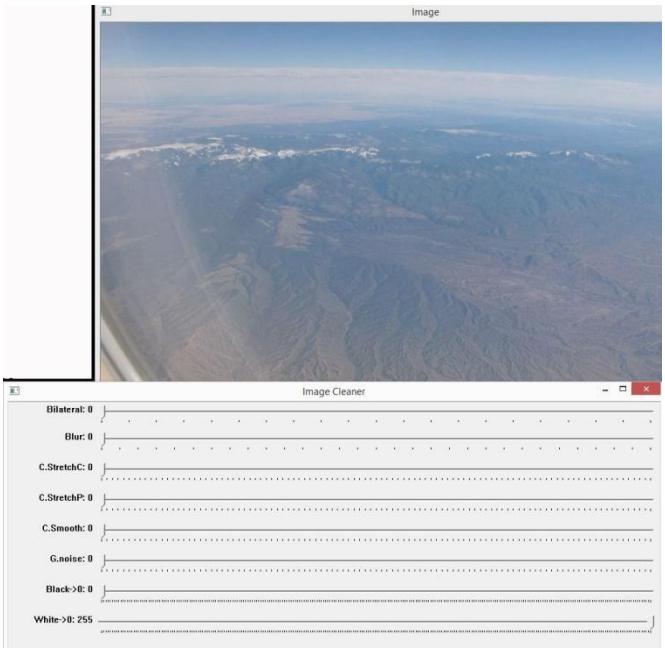


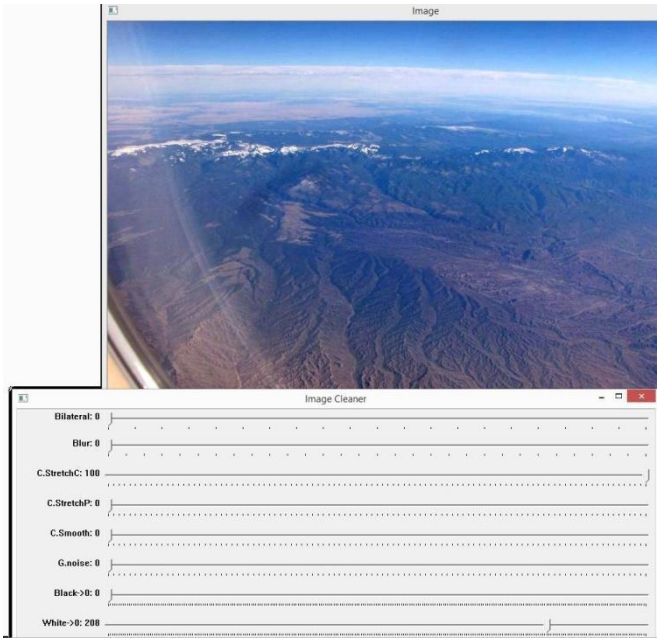
The color histogram correction filter also can almost completely restore the original colors within this photograph:



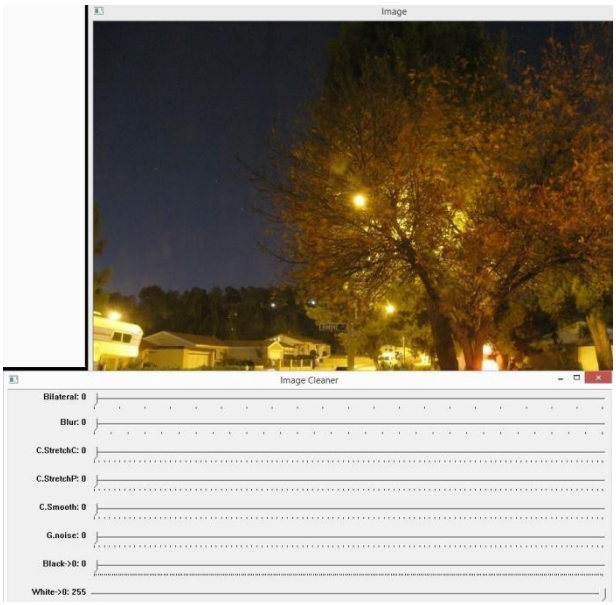


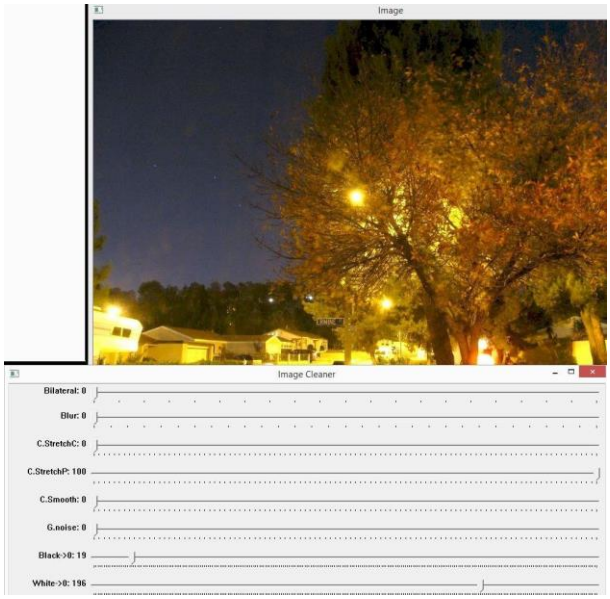
Jet window photo: glare and overexposure can also be removed by the color histogram correction filter. However if you first run a threshold filter to get rid of even more of the glare before applying the contrast filter, the resulting image is even brighter and clearer





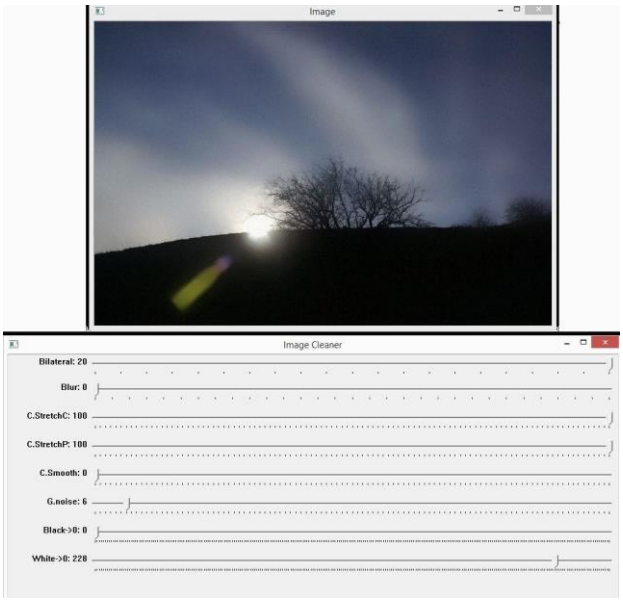
Night image: with the full range of black to white, even moving the color histogram and proportional histogram filters to full will make no change, but the image will be brightened if you first apply both black and white thresholding before applying histogram filter





EXPERIMENTS: BACKGROUND REMOVAL

Shadowed tree: Final example of the bilateral filter removing background details. If the bilateral filter is set to a high setting, it still does not distort the sharp high-frequency edges of the tree but downplays the sky details tremendously. Applying slight amounts of Gaussian noise and performing histogram stretches further improves the image.



CONCLUSION: REFLECTION OF LEARNING EXPERIENCE

PARTICIPATION

- Aleksandr Rozenman created the GUI interface, file handling, implemented OpenCV
- Jovan Ruano contributed with testing and test images
- Steven Wirsz implemented image filters, slider bars, additional testing and test images, PowerPoint, and documentation

REFERENCES

"OpenCV: Smoothing Images" Nov 08, 2013 *Opencv dev team.* 11 Nov 2013

<http://docs.opencv.org/doc/tutorials/imgproc/gausian_median_blur_bilateral_filter/gausian_median_blur_bilateral_filter.html#smoothing>

"OpenCV: Basic Thresholding Operations" Nov 08, 2013 *Opencv dev team.* 11 Nov 2013

<http://docs.opencv.org/doc/tutorials/imgproc/threshold/threshold.html>

"m19404 / ContrastStretching / ContrastStretching / Noise.hpp" March, 2013, 11 Nov 2013

<https://github.com/pi19404/m19404/blob/master/ContrastStretching/ContrastStretching/Noise.hpp>