COMP4490 - Computer Graphics 2
Project Report
Investigation of Non-Photorealistic Filters and their Effects on Images and Video

Robin Swanson & Nico Richard
umswans5@myumanitoba.ca umrich84@myumanitoba.ca

April 2014

For our project we implemented various non-photo-realistic image filters resulting in a "cartoon-ish" effect. We were also able to implement a rudimentary pipeline for splitting an input video into frames, applying a filter to each frame, and recombining them. This was accomplished using the ffmpeg toolset as well as the OpenMP library to parallelize the operations.

Here we include some example output from the various filters we were able to implement as well as any additional details pertinent to their implementation.

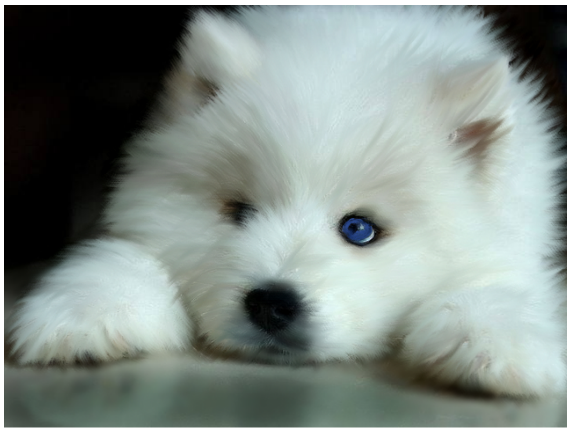
Examples of output [1] videos[2] can be found on our YouTube account.

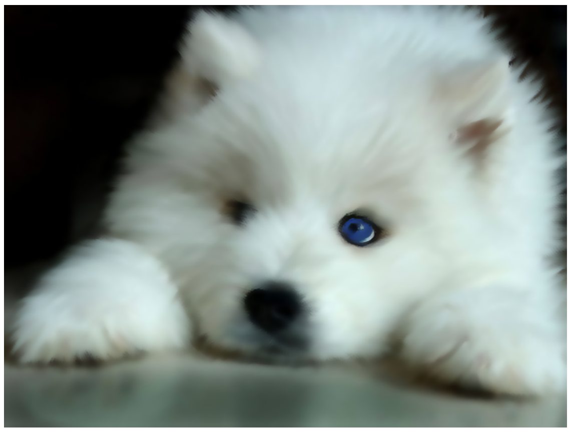
All source code can be found on our Github[3].

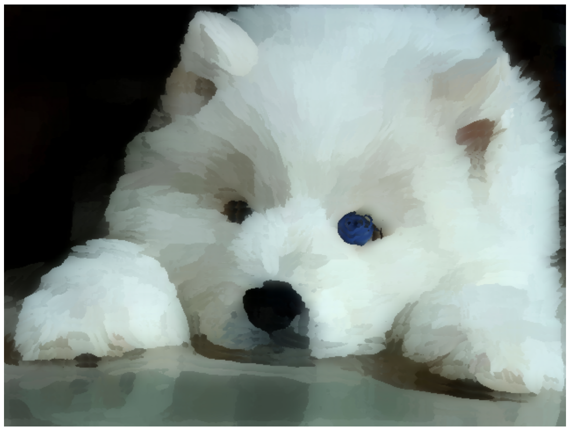
1 Filter Details

- Bilateral Filter: The Bilateral Filter is a simple gaussian (spacial, pixel value, and temporal) convolution. Smooths colours similar to a gaussian blur while preserving edge values. Currently implemented and working correctly.
- Kuwahara Filter: This filter explores regions nearby to the pixel and chooses the RGB value of the area with the smallest deviation of colour. Currently implemented and working correctly.
- Anisotropic Kuwahara Filter: Similar to the standard Kuwahara filter, but aims to preserve directionality of colour by modifying the regions explored based on a smoothed structure tensor. Currently implemented but *not* working correctly.
- Difference of Gaussian Filter: By convolving the image with two Gaussian kernels of different sizes we can take the difference of the two and use it as a band bass filter. This allows us to approximate the Laplacian of Gaussian equation. When the size of the second kernel is approximately 1.6 times the size of the first we can use the result as a cartoon-like effect. Currently implemented and working correctly.
- eXtended Difference of Gaussian Filter: By extending the standard Difference of Gaussian filter with a threshold function with a continuous ramp we can improve noise reduction and appearance. I've also done some re-parametrization of the filter to allow for strong customization. Currently implemented and working correctly.

[1] Long DoGs – `https://www.youtube.com/watch?v=jtmSWE7YlcA`
[2] Short DoGs – `https://www.youtube.com/watch?v=yUu6u1vjXO4`
[3] Soderberg on GitHub – `https://github.com/CityTransit/Soderberg`


(a) Our test input image before any modifications have been made.


(b) Test input image modified using the Bilateral Filter.


(c) Test input image modified using the Kuwahara Filter.

- Utility Filters: A Gaussian Filter for blurring and a Sharpening Filter to increase the definition of edges as well as a Kernel class capable of generating classical sobel, box blurring, and more! Currently implemented and working correctly.

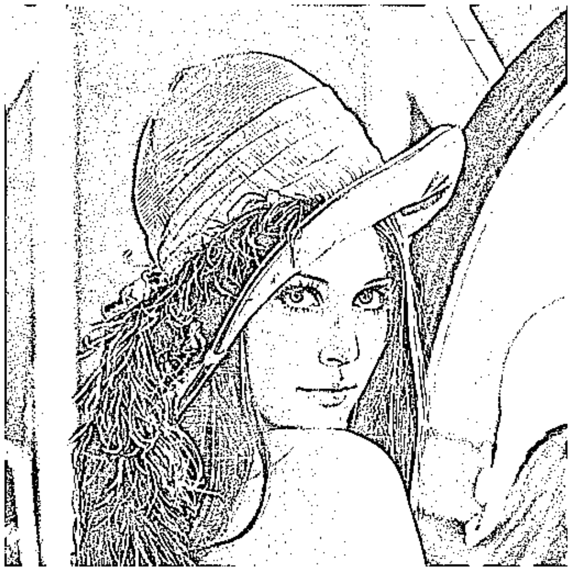
2 Implementation Details

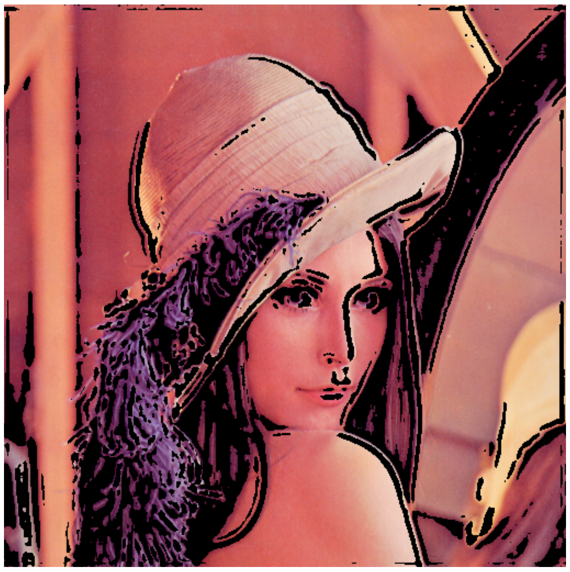
All of the code found in our work is original work written by us based on the reference papers with only two exceptions:

- The libpng code to read and write .png files was largely based on example code from libpng sources as there was no significant need or reason to write our own implementation. However, they were modified to suit our needs (e.g., removal of alpha channels, etc.).
- The current state of the Anisotropic Kuwahara filter (at the moment not working properly) is largely based on a GLSL implementation[4] in an effort to debug the filter and much of our original implementation was lost at some point. This filter proved to be quite difficult to implement, and continues to elude us. The principles behind it are fairly easy to understand, but modifying the circular search regions was a more interesting problem than anticipated. As it currently stands, the filter in no way affects the output image.

[4]polyakf on Google Code – `http://code.google.com/p/polyakf/`

(d) Our test input image before any modifications have been made.


(e) Test input image modified using only the Difference of Gaussian Filter.


(f) Test input image modified using the XDoG Filter with added colour.

3 Using The Code

We've included a simple make file to make compilation somewhat easier. The simplest way to get up and running may be to clone the git repo on Aviary and use the Aviary make flag.

```
git clone https://github.com/CityTransit/Soderberg.git
make aviary
```

We've also included a sample main file (batch.cpp) which uses OpenMP to provide some high level parallelism. This file can be used as a sample to create your own programs or modified to test different filters.

4 Future Work

Fixing the Anisotropic Kuwahara filter, obviously, would have been ideal. Unfortunately it proved to be quite difficult to implement.

More fine-grained parallelism would have been nice to exploit. Image filters are inherently embarrassingly parallel and it would not have been too difficult to implement. However, for this project we were much more interested in exploring a breadth of filters than perfecting any one filter.

References

[1] Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: Proc. ICCV, pp. 839846 (1998). doi: 10.1109/ICCV.1998.710815

[2] Kyprianidis, J.E., Kang, H., Dllner, J.: Image and video abstraction by anisotropic Kuwahara filtering. Comput. Graph. Forum 28 (7), 19551963 (2009). doi:10.1111/j.1467-8659.2009.01574.x

[3] Winnemller, H., Kyprianidis, J.E., Olsen, S.C.: XDoG: an extended difference-of-Gaussians compendium including advanced image stylization. Comput. Graph. 36 (6), 740753 (2012). doi:10.1016/j.cag.2012.03.004

[4] Kang, H., Lee, S., Chui, C.K.: Flow-based image abstraction. IEEE Trans. Vis. Comput. Graph. 15 (1), 6276 (2009). doi: 10.1109/TVCG.2008.81

[5] Kyprianidis, J.E., Dllner, J.: Image abstraction by structure adaptive filtering. In: Proc. EG UK TPCG, pp. 5158 (2008). doi: 10.2312/LocalChapterEvents/TPCG/TPCG08/051-058

[6] Torre, V., Poggio, T.A.: On edge detection. IEEE Trans. Pattern Anal. Mach. Intell. 8 (2), 147163 (1986). doi: 10.1109/TPAMI.1986.4767769

[7] "Image and Video-Based Artistic Stylisation"; Paul Rosin, John Collomosse; Computational Imaging and Vision, Volume 42 (2013)