# Image Processing Demo
—
# User Guide

Frank Jargstorff
fjargstorff@nvidia.com

March 15, 2004

# 1 Introduction

This user guide explains the five different filter-pipelines integrated in the `cg_scotopic` demo:

1. Simple image display. Strictly spoken this is not a filter-pipeline since it contains no filter. The load operator is hooked directly into the ImageView class.

2. A Gauss filter. This filter is the based on an naïve implementation that convolutes the complete $n \times n$ filter kernel.

3. A night filter. This is the first stage filter for the complete scotopic-vision filter. It basically is an advanced version of the blue filters used for “Hollywood Night” effects. For details see [1].

4. The scotopic vision filter. This one is an implementation of the night-scene filter described in [1]. Beyond the reduction in color vision this filter applies a smart combination of blurring and sharpening that simulates physiological processes of the human visual system to achieve the most realistic effect of a night scene.

5. A fast two-pass Gauss filter.

# 2 Controlls and Features

There are two basic kinds of controlls for the application:

- Those that can be activated through a menu or keyboard shortcuts. Right-clicking on the application window pops-up a menu with all the options. The keyboard shortcuts are given with each menu option.

- Interacting with the application by moving the mouse with the left or middle mouse-button down.

  **Left Mouse Button** moves the image.

  **Middle Mouse Button** zooms in and out on the image.

## 2.1 General Navigation and Controll

Common to all experiments is the

- mouse navigation capability,
- the **h**-key to bring up the parameter-control sliders, and
- the **q**-key to terminate the application.

The controll sliders intercept the mouse events. Therefore it is impossible to move or resize the image while the controll sliders are displayed.

## 2.2 Controlls for specific experiments

Controlls that are for specific experiments are:

- The **d**-key to configure the pipeline to display mode.
- The **g**-key to activate the naïve Gauss filter.
- The **n**-key to switch into simple night scene mode.
- The **o**-key to enable the full sc**o**topic filter pipeline.
- The **1**-key to activate a simple one-dimensional Gauss filter.
- The **2**-key to configure the pipeline to use two one-dimensional Gauss filters to perform a complete Gaussian two-dimensional blur.
- The **f**-key will save the current image as `Scotopic.dds` into the current directory.

# 3 Experiments

On startup the application loads the example image and is in display mode. The image can be intuitively moved around using the left mouse button.

## 3.1 Gauss Filter

Enabling the parameter slider using the **h**-key exposes the only tweakable for the Gauss filter: the standard deviation **Sigma** ($\sigma$). $\sigma$ basically determines how much the image gets blurred. Changing the value clicking on the slider with the left-mouse button and moving it around will trigger the image to be re-filtered and displayed. With this primitive implementation this might be less than smooth on most current graphics cards.

## 3.2 Night Filter

The tweakable for the night filter is the overall brightness.

## 3.3 Scotopic Filter

The scotopic filter combines the night-filter as first stage with two fast Gauss filters to achieve blurring and sharpening in one step. The variable parameters are **Sigma** for the less blurry of the two Gauss filters. The second Gauss filter is always set to operator with a $\sigma' := \sigma \cdot 1.3$, i.e. creates a slightly blurrier image. The next tweakable **Gamma** determines how much sharpening is applied to the blurred image (this is basically subtracting the blurrier image from the blurred image). The **Brightness** parameter is the same as in the night filter example.

## 3.4 One-Dimensional Gauss Filter

Just like the 2D version this filter takes a **Sigma** to determine blurriness.

Using the **t**-key it is possible to toggle the one-dimensional filter's filter direction. Default is horizontal blurring.

## 3.5 Two-Pass Gauss Filter

Despite the fact that this pipeline has an additional stage for the second 1D Gauss filter it is obvious in comparison with the naïve Gauss implementation that it is much faster.

# 4 Hardware Requirements

Since all the images are processed as 16-bit floating point and the complex filter graphs, especially the Scotopic filter use several images simultaneously some cards tend to run out of video memory to allocate the pixel-buffers (invisible frame-buffers used by the filters).

If the application exits on you with “out-of-resource” exceptions or even crashes without an error message it is most likely this happened because the OpenGL driver ran out of resources.

There are certain configurations that can be set in the driver’s control-panel that can greatly reduce resource consumption:

- Turn off full-screen anti-aliasing.
- If running in DualView mode set nView to single-display mode. In this case 3D graphics acceleration is only available on one of the two screens.

The app runs without issues on any GeForceFX with at least 128MByte of video memory.

# References

[1] Thompson, W.B., et al., *A Spatial Post-Processing Algorithm for Images of Night Scenes*, Journal of Graphics Tools 7(1), 2002, pp.1–12.