**Problem Statement/Motivation**

The programmability of modern graphics hardware and its steadily increasing computational power makes the implementation of highly complex image processing techniques in real-time rendering viable. Especially the process of blurring an image is of interest for a lot of effects. This work examined different image blurring techniques when realised using a shading language as well as general-purpose computing on graphics processing units (GPGPU). This is motivated by the increasing popularity for implementing special tasks like physics simulations using GPGPU.

**Contributions**

- Comparison of a broad range of different filtering methods in terms of quality and performance
- Comparison of CUDA versus GLSL regarding image filtering
- Creating a guideline for graphics developers who are interested in integrating image filtering in their application

**Results**

- Discussion of the performance differences between GLSL and CUDA when varying image and filter size
- Determined effective filter size through automated process
- Overall performance charts based upon these results
- Image quality comparison using a visual metric (hdr-vdp2)

**Conclusion**

- Pyramid methods fastest but lowest control over filter size
- Quasi-convolution offers the best trade-off between quality and performance amongst the tested pyramid filters
- No clear winner between CUDA and GLSL
- Considerable speed-up of some methods through the usage of CUDA shared memory.