Parallelization of the Canny Edge Detection Algorithm using CUDA

Team Members:

Alex Varvel: varvel@pdx.edu

Daniel Collins: daniel.e.collins1@gmail.com

Anil Reddy-Peddolla: anil2@pdx.edu

Topic Description:

Edge detection is a mathematical method of determining the "edges" within an image. An "edge" is defined as a point which the brightness of the image has discontinuities. Sharp changes in brightness allow the edge detection algorithm to map an outline of the image being observed, and these edges can correspond to changes in depth, orientation, materials, or lighting. Mapping the edges of an image is a non-trivial task, as many false edges can be mapped, or segments of a real edge can be left off if the change is not drastic enough for the algorithm to pick up.

The algorithm we chose to implement is the Canny Edge Detection algorithm. This algorithm is a first-order image processing algorithm that was developed by John F. Canny in 1986 and follows a 5-step detection algorithm: Apply a Gaussian filter, find the intensity gradient, apply non-maximum suppression, apply a double threshold, track the edges using hysteresis. The size of the Gaussian filter and the double threshold can be manipulated in order to affect both the computation time and the effectiveness of the algorithm. Based on the image being processed, different combinations of the two may yield more accurate results.

We plan to implement this algorithm using the parallel computing platform CUDA. We chose CUDA as our platform because it has been designed/optimized for graphics and image processing falls into that category. Additionally, CUDA is a very common platform in industry, so we feel having experience in it will benefit us greatly. In order to use CUDA, an nVidia graphics card must be present in the system. However, this won't pose any issues for our group as all of our computers use nVidia graphics.

Importance of Topic:

This topic is of particular importance for machine/computer vision and, in general, image processing. In image processing, an edge is of particular importance because it allows distinct characteristics of 3-D space to be mapped. Mapping edges allows the device processing the image to significantly reduce the amount of data being processed/stored which becomes particularly important for computer vision.

The Canny Edge Detection algorithm is an important algorithm within the edge detection field because it drastically reduces the amount of data that needs to be processed when compared to other first-order image processing algorithms. This algorithm can also be parallelized using CUDA, which falls in line with our project interests.

Method of Evaluation

Validation of an edge detector is an admittedly difficult problem in academia due to the subjectivity of the results. It requires establishment of what is called the *ground truth,* or a baseline of results. Our validation methodology will rely running two classes of images through our implementation as well as through a high-quality reference implementation (TBD). The classes of images will be 1) synthetic hand-crafted images with varying types of edges and 2) a corpus of stock images commonly used in literature. The output of each edge detector will be evaluated both by statistical analysis of the binaries as well as qualitative human analysis.

References

The original Canny edge detection paper:

Canny, John. "A computational approach to edge detection." *IEEE Transactions on pattern analysis and machine intelligence* 6 (1986): 679-698.

CUDA implementations of Canny edge detection:

Ogawa, Kohei, Yasuaki Ito, and Koji Nakano. "Efficient Canny edge detection using a GPU." *Networking and Computing (ICNC), 2010 First International Conference on*. IEEE, 2010.

Luo, Yuancheng, and Ramani Duraiswami. "Canny edge detection on NVIDIA CUDA." *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on*. IEEE, 2008.