Summary

Introduction

Computer Vision studies the analysis of digital images and videos. The corresponding research field offers an increasing range of possibilities for human computer interaction, artificial intelligence and the visual examination of complex processes.

Existing libraries such as OpenCV or Emgu CV offer access to a substantial number of solutions for image segmentation, object recognition and motion analysis. However, most algorithms demand a significant amount of computing resources and are documented and implemented for CPU based operations that do not benefit from contemporary hardware acceleration. This architectural drawback is responsible for unnecessary data bottlenecks, occupies computational resources and reduces the amount of data that can be processed.

This thesis proposes a hardware accelerated implementation of optical flow to discuss the benefits of highly parallel computing structures for image processing. Optical Flow is a motion analysis procedure to detect movement in camera images by comparing two consecutive frames. Image stabilization, feature tracking and the monitoring of assembly lines are some of its common applications.

In order to illustrate the contribution of our research, we will introduce our GPU implementations for two different optical flow solutions and compare them with CPU based procedures.

Methodology

This thesis will focus on two main solutions for Optical Flow analysis, one is based upon the scientific publication by Horn and Schunck [1], while the other is based upon Lucas and Kanade [2,3]. Our CPU based methods are written in C++, while the GPU algorithms are written on the GPU language HLSL.

Horn and Schunck

Horn and Schunck algorithms are the most popular solution for Optical Flow, and have been used for many applications and evaluation models. This method bases its solution on a differential technique, by including a constraint that assumes smooth flow in the whole image. The result is achieved by computing two steps, the first one obtains an estimation of the derivatives of the images, detecting the edges of the elements and the differences in their position over time, while the second one follows an iterative method to minimize the error in the final prediction of the velocity field.

Lucas Kanade

Lucas Kanade method is a widely used differential solution for Optical Flow. This method is mainly used as a technique to find the local Optical Flow, that obtains the velocity of few featured points. Lucas Kanade solution is achieved in two steps as the previously defined method. The first step obtains the spatial-temporal derivatives of the images, while the second step obtains the solution for the velocity field. This method assumes that the flow is constant in a neighbourhood of pixels, and obtains a solution for the x and y components of the velocity of every pixel by the least squares criterion.

Discussion

The development of the described methods for CPU and GPU allowed us to evaluate the performance of the algorithms in both processors. Considering different resolutions of input images that provide different levels of detail, the time of processing changed significantly as shown in Figure .

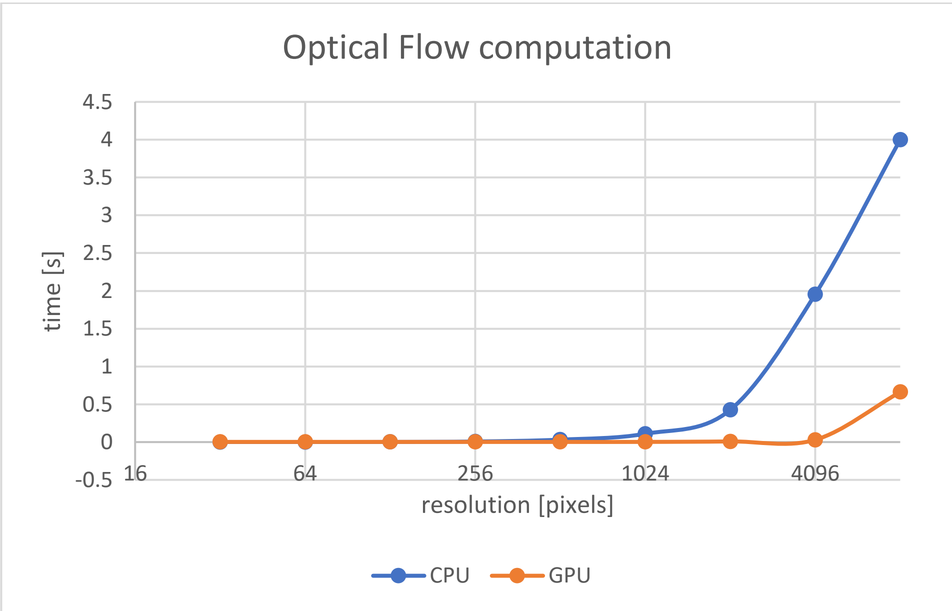

Figure 1. Optical Flow computation in CPU and GPU

Experiments

The acceleration of our GPU solution in reference to the CPU, offers the possibilities to use additional computing resources to process the resulting data and increase the accuracy of our prediction.

In order to evaluate the improved quality of our implementation, we developed a simulator to generate motion and its corresponding Optical Flow data (ground truth). By comparing the ground truth with our estimation, we are able to determine a significant decrease in prediction error as visualised in Figure and Figure .

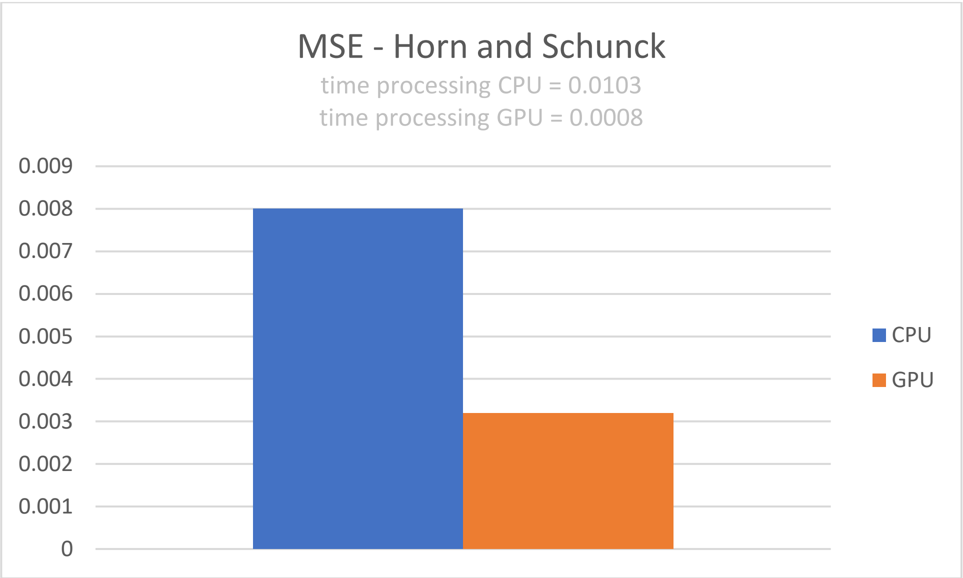

Figure 2. MSE evaluation between original and improved HS method

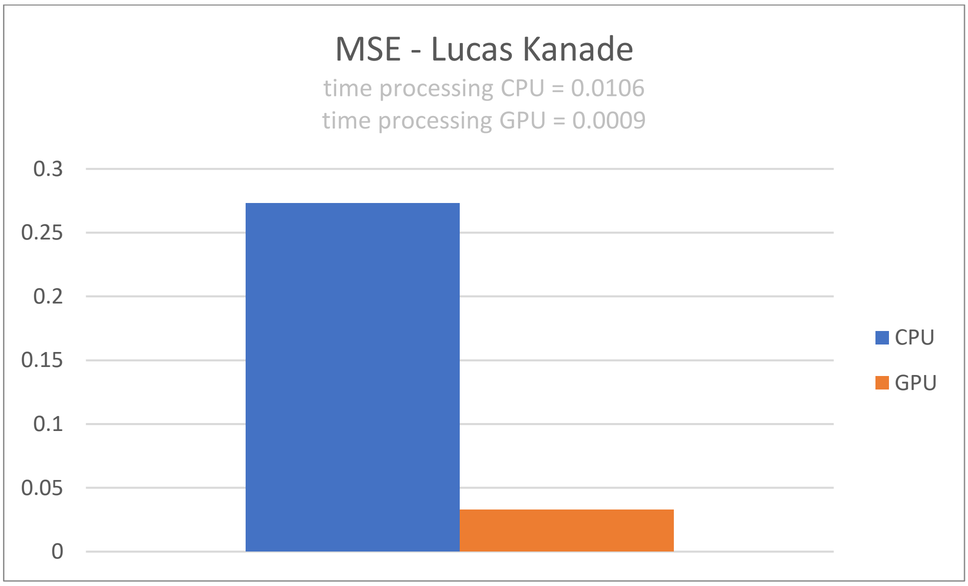

Figure 3. MSE evaluation between original and improved LK method

Conclusions

The necessity of managing a massive amount of image data, represented as pixels, makes the highly parallel structure of GPUs more efficient than other processing units. The increase of performance and quality of the proposed methods, represent a valuable advantage for real time applications. Our solution offers the evaluation of optical flow for each pixel even in high resolution. The corresponding processing time is fast enough to process live feeds from high speed cameras and post process the resulting data.

Therefore, we are able to efficiently reduce computing overloads and bottlenecks. The resulting methods could potentially reduce costs in several fields, such as machine vision, human computer interaction or accelerate the ongoing trend of device miniaturization.