Real-time Vision Using FPGAs , GPUs and Multi-Core CPUs

Anders Kjær-Nielsen

Computer Vision has gone through a significant evolution in the last decade, whilst finding its way into more and more application domains, with robotics, video surveillance and automation being the most dominant. This trend is due to extensive basic research in the field of vision, however the introduction and evolution of a wide variety of powerful hardware architectures have made the developed theory more applicable in performance demanding and real-time[1] applications. Three different architectures have dominated the field due to their parallel capabilities that are often desired when dealing with massive amounts of visual input; the Field Programmable Gate Array (FPGA), the Graphics Processing Unit (GPU) and the multi-core Central Processing Unit (CPU).

This work presents implementations of low-level vision algorithms such as debayer mosaicing, undistortion and rectification, and linear- and non-linear filtering processes on FPGAs that has been used for preprocessing images in the context of a bigger Early Cognitive Vision (ECV) system. With the introduction of GPUs for general purpose computing the preprocessing was re-implemented on this architecture and used together with a multi-core CPU to form the basis for a heterogeneous platform. This platform combined with a developed framework, called the "Vision Frontend", has been used to increase the performance of the ECV system, such that this is now performing in real-time at more than 20 fps with a latency of less than 150 ms (a speed up of a factor 90 with the frame rate and a factor 26 with the latency compared to a single core CPU implementation). The framework, that offers both coarse and fine grained parallelism, is very general and therefore makes it applicable as a general purpose parallel processing scheme for various computer vision as well as other computationally demanding projects.

The work includes several applications where the developed parallel implementations have been utilized: the European project DrivSco where the "Vision Frontend" is used in a personalized driver assistance system, the Danish Handyman project "Ring on the Hook" where the "Vision Frontend" is used to enable the placing of rings onto moving hooks using a robot arm, and finally an institute research called RoBlood where large filtering kernels are implemented on an FPGA to achieve an embedded and small end product that consumes little power.

The "Vision Frontend" has become the cornerstone for parallel and real-time processing within our group and is therefore modified and extended for the use in various projects that requires some sort of real-time performance. One example is a project called Robo Packman, where the fast processing of the ECV system has been utilized to find the pose of pieces of meat for automatic handling. Another example is a second European project called PACO-PLUS, where the ECV system is used in an attempt to design a cognitive robot using visual input.

[1] A term that is application dependent, however generally refers to both the frame rate and latency of a system