RESUME’s wavelet-based scalable video decoder

Hendrik Eeckhaut  Mark Christiaens  Harald Devos  Philippe Faes
Dirk Stroobandt
Hendrik.Eeckhaut@elis.UGent.be
Parallel information systems - ELIS - Ghent University - Belgium
http://www.elis.ugent.be/resume

Abstract

In the RESUME project we developed a real-time FPGA prototype of a fully scalable, wavelet based video decoder which overcomes the complexity and bandwidth issues associated with scalable video.

1 Introduction

The RESUME-project (Reconfigurable Embedded Systems for Use in Scalable Multimedia Environments) explored the benefits of using reconfigurable hardware for the implementation of scalable multimedia applications by building an FPGA implementation of a real-time, scalable, wavelet-based video decoder. The term *scalable video* refers to a coding scheme that can easily accommodate changes in quality of service (QoS) without the need for transcoding. A scalable video stream can be decoded at varying frame rates, resolutions and image qualities by simply skipping dispensable parts in the video stream, only decoding those parts that will contribute to the displayed video. The algorithmic structure of the RESUME scalable video coder and decoder (codec) is shown in Figure 1 and is described in [2].

Such a scalable video codec has advantages for both the server (the provider of the content) and the clients. On the one hand the server scales well since it has to produce only one encoded video stream that can be broadcasted to all clients, irrespective of their QoS requirements. On the other hand the client (or the network) can easily adapt the decoding parameters to its needs. This way it is possible to optimize the use of the network, display, the required processing power, the required memory, ...

Scalability has a lot of advantages but comes at a cost. The decoding algorithm is computationally complex and really stresses the system bandwidth as it replaces the block-based DCT-approach with frame-based wavelets. This has a tremendous impact on the hardware architecture.

![Figure 1. High-level overview of the codec.](image)

2 Hardware implementation

We implemented the complete decoding pipeline of our custom wavelet-based scalable video codec on a PCI development board equipped with a Stratix FPGA S60 and 256 MiB of DDR SDRAM memory. The FPGA board is plugged into a standard PC with two monitors, one dedicated to displaying the decoded video, the other to interact with the system (Figure 2). The design goals were real-time, lossless decoding of CIF-sequences (352 × 288 pixels) at 25 frames per second.

Implementing a complete video codec is a complex undertaking that requires careful planning. We applied the following methodology. First the entire software code base was cleaned-up and we made sure that the algorithms used were properly understood. We chose to use a HW/SW-codesign approach leaving as much of the algorithm as possible in SW running on a CPU while implementing the time-critical parts in reconfigurable HW. The hardware design was implemented using Altera’s SOPC (System-On-a-Programmable-Chip) Builder for component-based system integration. For each of the steps in the decoding pipeline we developed custom components to deliver the required hardware acceleration (Figure 3). Many compo-
Figure 2. Photograph of our wavelet-based scalable video decoder in action.

Figure 3. Simplified hardware overview.

...