Implementing a Hardware-Friendly Wavelet Entropy Codec for Scalable Video

Hendrik Eeckhaut, Mark Christiaens, Harald Devos and Dirk Stroobandt

ELIS-PARIS, Ghent University, St. Pietersnieuwstraat 41, 9000 Gent, Belgium

ABSTRACT

In the RESUME project (Reconfigurable Embedded Systems for Use in Multimedia Environments) we explore the benefits of an implementation of scalable multimedia applications using reconfigurable hardware by building an FPGA implementation of a scalable wavelet-based video decoder. The term 'scalable' refers to a design that can easily accommodate changes in quality of service with minimal computational overhead. This is important for portable devices that have different Quality of Service (QoS) requirements and have varying power restrictions.

The scalable video decoder consists of three major blocks: a Wavelet Entropy Decoder (WED), an Inverse Discrete Wavelet Transformer (IDWT) and a Motion Compensator (MC). The WED decodes entropy encoded parts of the video stream into wavelet transformed frames. These frames are decoded bitlayer per bitlayer. The more bitlayers are decoded the higher the image quality (scalability in image quality). Resolution scalability is obtained as an inherent property of the IDWT. Finally framerate scalability is achieved through hierarchical motion compensation.

In this article we present the results of our investigation into the hardware implementation of such a scalable video codec. In particular we found that the implementation of the entropy codec is a significant bottleneck. We present an alternative, hardware-friendly algorithm for entropy coding with excellent data locality (both temporal and spatial), streaming capabilities, a high degree of parallelism, a smaller memory footprint and state-of-the-art compression while maintaining all required scalability properties. These claims are supported by an effective hardware implementation on an FPGA.

Keywords: Wavelet, entropy coding, FPGA, scalable video

1. INTRODUCTION

"Scalable video" is encoded in such a way that it allows to easily change the Quality of Service (QoS) i.e. the frame rate, resolution, color depth and image quality of the decoded video, without having to change the video stream used by the decoder (except for skipping unnecessary blocks of data without decoding) or without having to decode the whole video stream if only a part of it is required.

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 video stream that can be broadcast to all clients, irrespective of their QoS requirements. On the other hand the client can easily adapt the decoding parameters to its needs. A home cinema system can decode the stream at full quality, while a small portable client can decode the stream at low resolution and frame rate without needing the processing power of the larger clients. This way the decoder can optimize the use of the display, the required processing power, the required memory, . . .

The internal structure of one implementation of a scalable encoder is shown in Figure 1 and was described in Ref. 1–5. It consists of the following parts:

Further author information:
Hendrik Eeckhaut: Hendrik.Eeckhaut@ELIS.UGent.be, Telephone: +32 9 264 89 65; http://www.elis.ugent.be/~heeckhau

The remainder of this paper is not included as this paper is copyrighted material. If you wish to obtain an electronic version of this paper, please send an email to bib@elis.UGent.be with a request for publication P105.189.pdf.