Available online at www.sciencedirect.com
ScienceDirect
Signal Processing: *Image Communication* 22 (2007) 252–265
SIGNAL PROCESSING: IMAGE COMMUNICATION
www.elsevier.com/locate/image

Multiple description video coding using coefficients ordering and interpolation

Nicola Conci*, Francesco G.B. De Natale

Department of Information and Communication Technology, University of Trento, Italy

Received 18 December 2006; accepted 21 December 2006

Abstract

A Multiple Description Video Codec is presented, suitable for high-quality digital video transmission. The method works in the DCT domain, and associates to each descriptor two data sources: a subset of the original transform coefficients, and side information concerning the position and magnitude of the remaining coefficients. When one or more descriptors are (partially) lost, the side information is used at the decoder to interpolate the missing coefficients, thus achieving a better reconstruction. The effectiveness of the proposed approach is greatly enhanced by the use of a JPEG-like syntax and an efficient 3D-VLC in the encoding of the overhead information. This choice makes also easier and natural the integration of the MDC scheme in most DCT-based standard encoders such as M-JPEG and MPEG-1/2. With respect to previous approaches to MDC, the proposed strategy is characterized by a limited overhead and a reduced impact in terms of computational complexity. Furthermore, the coefficient selection strategy allows splitting the video stream into an arbitrary number of descriptors, overcoming a major limitation of most MDC schemes. These features make the proposed method suitable for real-time applications of video delivery, especially in wireless environments where the computational complexity is usually performed by low-power devices. As far as the reconstruction quality is concerned, extensive experimental sessions demonstrated that the loss of one descriptor produces a slight degradation in a two-descriptors scheme, while it is almost seamless in a three-descriptors scheme.

Keywords: Multiple description coding; Video streaming; Error resilience

1. Introduction

The diffusion of internet video-streaming services is nowadays one of the most challenging applications and this is confirmed by many concurrent reasons, including the increasing diffusion of broadband networks in all countries, the reduction of connection costs both for end-users and service providers, the growing awareness of those users on the available technologies, together with the industrial push on the market of audio-visual systems [8]. In this evolving scenario, technologies for video compression and delivery play a key role, being responsible for the quality of service (QoS) of the whole system. Nevertheless, the performance of the underlying data networks may pose hard limitations to QoS, due to channel and network problems (bit-errors, delay jitter, packet losses, node congestion, link failures, server faults, etc.). Typical

*Corresponding author. Tel.: +39 0461 883917; fax: +39 0461 882093.
E-mail addresses: conci@dit.unitn.it (N. Conci), denatale@ing.unitn.it (F.G.B. De Natale).

0923-5965/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.image.2006.12.009

application-layer protocols used for real-time video streaming, such as RTP/RTCP over UDP [19], may be affected by such problems, resulting in a dramatic reduction of the quality experienced at the end-user's side, due to reconstruction errors, de-synchronization, system crashes.

There is no ultimate solution that can cope with all the above-mentioned problems. The common aim is to allow a seamless decoding of the stream even in the presence of critical errors in the received data, and this can be achieved by introducing error resilience features. Such features usually improve the reliability of the transmission, at the cost of some kind of overhead (delay, complexity, redundancy, etc.). It is possible to distinguish two main classes of approaches, namely *source*- and *receiver based*. The formers are very effective, for they can strongly limit the errors by applying an efficient rate control, as well as error detection and correction using ARQ mechanisms. Unfortunately, retransmission strategies imply the availability of a return channel, and are, therefore, suitable mainly for point-to-point transmission over networks characterized by a limited round-trip-time [3]. In the case of multicast or broadcast applications, or in the presence of networks with high latency (e.g., satellite connections), receiver-based approaches turn out to be more realistic. In this case, the decoder is responsible of selecting the amount of information to retrieve, according to the (possibly time-varying) network conditions. This solution implies a suitable structuring of the video stream, to be performed at the encoder. Two possible approaches achieving such a goal are layered coding and multiple-description coding (MDC) [4,10].

Layered coding is very efficient for many applications, in particular when the video streaming server has to deal with heterogeneous networks and terminals, where the end-user can be provided with a wireless mobile device such as a smartphone, or with a powerful multimedia workstation connected to ADSL or fibreoptics. In this case, the video is structured in a basic stream, called base layer, which represents a rough version of the stream containing the fundamental information for understanding the scene (usually at low quality and low rate). In addition, a certain number of enhancement layers are provided, containing the necessary information to achieve a progressive increase of the video quality. The base layer makes available to every user a baseline quality, which is progressively improved by the successive enhancement layers if the bandwidth is sufficiently large. One of the most efficient methods of layered coding is represented by the streaming profile of the ISO-IEC MPEG-4 standard (see [15]), Fine Granularity Scalability (FGS), that has been recently adopted even in the ISO-ITU H.264 (also known as MPEG-4 part 10) [18]. In FGS, the mechanism for achieving a continuum of quality enhancements is constructed on a bit-plane basis, scanning the data to be delivered from the most significant bit to the less significant one. Layered coding is often associated to unequal error protection (UEP) to ensure a higher protection to the base layer, being the most important information to be delivered [6,9]. It is worth mentioning that in general scalable video coding is more appropriate for the delivery of pre-encoded sequences. In practice, some scalability methods (such as FGS) are applicable even for real-time streaming, thanks to their flexibility and pleasant transitions in quality variations. Nevertheless, if the base layer is not received or contains errors, the additional information coming from the enhancement layers is almost useless.

MDC, aims at solving these problems by structuring the stream into two or more equivalent representations of the stream (called descriptors), which are individually decodable at lower quality, while providing the highest quality if suitably merged. It is worth noting that the advantages of such method go beyond the concept of scalability, because the generated video stream is structured in such a manner that each descriptor owns the fundamental property of independence. Independence is a common term to generalize the concept of diversity, which becomes one of the most appropriate ways for transmitting a video sequence in an error-prone environment. The combination of MDC and diversity (both in terms of path and server diversity) was first presented in [1,26,27], where the authors demonstrated the significant improvements in terms of robustness in real-time applications.

Usually, MDC applications send data over n ($n = 2$ in the example of Fig. 1) independent routes (path diversity) and the decoder must be able to reconstruct the stream at acceptable quality for every subset of received descriptors. The more the descriptors correctly received, the higher the quality of the decoded stream. The extension of this concept to server diversity comes very easy. In this second option, the only difference consists in the download strategy of the client application.