J. Vis. Commun. Image R. 24 (2013) 12–22

Contents lists available at SciVerse ScienceDirect

J. Vis. Commun. Image R.

journal homepage: www.elsevier.com/locate/jvci

A novel enhancement for hierarchical image coding

Shuyuan Zhu [b], Bing Zeng [a,*]

[a] Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Clearwater Bay, Kowloon, Hong Kong, China
[b] School of Electronic Engineering, University of Electronic Science and Technology of China, Chengdu, Sichuan, China

ARTICLE INFO

Article history:
Received 17 February 2012
Accepted 12 October 2012
Available online 23 October 2012

Keywords:
Image coding
Hierarchical image coding
R-D coding performance
Constrained quantization

ABSTRACT

Hierarchical image coding usually codes a down-sampled version of an original image and then the difference between the original image and a reconstructed version that is interpolated from the down-sampled layer. In this paper, we demonstrate, for the first time, that when the bit-rate used to code the residual layer falls into a critical region (which covers almost all typical bit-rates used in practice), it often happens that all pixels in the down-sampled layer would be deteriorated if the corresponding coded residuals are added into them. To avoid this problem, we first propose a "naive" solution: no coded residuals will be added back into the down-sampled layer; whereas coded residuals will be added only into the interpolated pixels. Then, we propose to apply a constrained quantization technique during the coding of the residual layer so that all residual pixels at the interpolated positions will end up with an improved quality. To verify its effectiveness, we conduct extensive tests to show that the gap between the hierarchical coding scheme and its single-level counterpart (which is typically around 2–3 dB in the 2-level hierarchy) will be filled up by a rather big percentage.

1. Introduction

With the widespread deployment of various high-speed networking technologies, people can now enjoy numerous multimedia services. Under the multimedia umbrella, image and video signals are obviously the main trunk, thanks to their incompetent capability of offering pleasant interpretations to human eyes. In many practical applications, an image or video signal is always compressed and stored at a central server. From the server to its end-users, however, the paradigm is very heterogeneous in two aspects: (1) the transmission channel to each end-user is quite different one from the others and this channel is also time-varying and (2) end-users are always using different devices to receive the service.

To accommodate this heterogeneity, an image or video data can be compressed into the so-called hierarchical format or a more sophisticated one – the scalable format. The original idea of hierarchical coding came from the so-called Gaussian–Laplacian pyramids by Burt and Adelson [1]. It had since then been modified into different versions via different decompositions or by applying a lossy coding on the residual layers [2–7], and was finally formalized in 1993 in the JPEG image coding standard as the hierarchical encoding mode [8]. Afterwards, the research interest on the hierarchical image coding was shifted to approaches that are based on the wavelet decomposition or filter banks. Several elegant algorithms have been developed to this end, e.g., Shapiro's EZW (embedded zerotrees of wavelets) [9], Said and Pearlman's SPIHT (set partitioning into hierarchical trees) [10], along with many others [11–16]. More recently, hierarchical/multiresolution image coding based on some advanced decomposition techniques (such as curvelets, contourlets, and directional wavelets) has also been developed [17–20]. Meanwhile, many scalable image and video coding techniques have been developed, e.g., EBCOT (embedded block coding with optimized truncation) for image coding [21], FGS (fine granularity scalable) [22] and PFGS (progressive fine granularity scalable) [23] for video coding, SVC (scalable video coding) adopted in H.264/AVC [24], etc. This present work focuses on the hierarchical coding of image signals.

Although many wavelet or filter-bank algorithms provide a very fine scalability to fit the varying channel bandwidth and the spatial resolution of the user's end-device, it usually takes time to select the bits that are necessary to decode a specified hierarchical layer (with a fixed spatial resolution). For instance, in the SPIHT algorithm (and many other wavelet-based algorithms), bits are organized into individual hierarchical trees (or coded blocks), while each tree only represents one portion of the whole image in different resolutions. Therefore, the server needs to visit the first tree-set, select the necessary bits from it, jump out and go to the second tree-set, etc., and continue over all tree-sets. This visit-and-break fashion of selecting bits not only is time-consuming, but also makes it unrealistic to pre-packetize the bits in each tree. On the contrary, the hierarchy-based schemes are much simpler. This is because all bits representing

* Corresponding author. Fax: +852 2358 1485.
E-mail address: eezeng@ece.ust.hk (B. Zeng).

1047-3203/$ - see front matter © 2012 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.jvcir.2012.10.002

each layer in the hierarchical structure are put into an independent set. Thus, one just needs to send the whole set if a layer is needed, without any break during the process of pumping bits into the buffer (to be transmitted). As evidence, H.264's SVC has adopted the hierarchical idea to implement its spatial scalability [24].

In this paper, we take a revisit to the hierarchical image coding scheme. Because of the hierarchical feature, it is known that a multi-layer hierarchical coding always yields a performance lower than that of the single-level (i.e., non-hierarchical or baseline) coding. For instance, a gap of 2–3 dB exists typically in the two-level hierarchy. This is pretty much the same as the gap between the FGS video coding and its non-scalable counterpart [22,23]. Our study in the present work shows that the contributing factor to this performance drop is two folded: (1) reduced efficiency in the coding of both the base-layer and residual signals in a hierarchical scheme and (2) adding the "badly" coded residuals into the upper-layer pixels that are already coded with a "good" quality. The first error-source is intrinsic and thus unavoidable: (1) the inter-pixel correlation in the base layer decreases so that the resulted coding efficiency is reduced and (2) a residual signal becomes harder to encode. However, our study will illustrate that the second error-source can be avoided by allowing the coded residuals to be added back into the interpolated pixels only. This result seems rather straightforward. Nevertheless, we believe that it is the first time, to our best knowledge, to discover this interesting and useful phenomenon. Based on this observation, we then propose to apply an advanced quantization technique so as to further improve the quality of those interpolated pixels. As a result, it will be found that the aforementioned performance gap can be filled up by a rather significant percentage (e.g., 30–60% in many practical coding scenarios).

The rest of this paper is organized as follows. In Section 2, we first review briefly the hierarchical encoding mode formalized in JPEG. Then, we focus on the performance gap between it and its non-hierarchical counterpart (called the baseline encoding mode in JPEG). In this aspect, we demonstrate that there exists a critical region for each image such that a part of the performance gap is avoidable and this region includes most typical bit-rates used in practice. In Section 3, we provide a "naive" solution to fill up the performance gap and demonstrate its effectiveness by experimental tests on practical image data. In Section 4, we first introduce the constrained quantization technique we developed recently in [25] and then illustrate how it can be used here to fill up the gap more efficiently and test its effectiveness. In Section 5, we modify the constrained quantization algorithm to control the resulted bit-rate tightly so as to achieve a further improved R-D performance. Some experimental results for the multi-layer hierarchy (more than 2 layers) are provided in Section 6, and, finally, some conclusions are drawn in Section 7.

2. Hierarchy-based multi-layered image coding

Referring to Fig. 1 which shows the K-layer hierarchical image coding, let's use $X_k(n,m)$ and $\ddot{X}_k(n,m)$, $k = 0,\ldots,K-1$, to denote the source image and the coded/decoded image at layer L_k, respectively; while $k = 0$ indicates the bottom layer (of the highest resolution). The up-sampled and interpolated image from the decoded one at layer L_k is denoted as $\text{Int}\{\ddot{X}_k(n,m)\} = X'_{k-1}(n,m)$, and the difference between it and the source image is defined as the residual signal $R_k(n,m) = X_k(n,m) - \dot{X}_k(n,m)$ – which is then encoded/decoded as $\hat{R}_k(n,m)$. Some analysis will first be performed in the following for the case $K = 2$, from which we will identify a critical problem that was ignored all the time in the past. Then, we will characterize this problem more precisely and validate it in the real image coding scenario. Finally, we will discuss the extension to the case $K > 2$.

2.1. The potential problem

Set $K = 2$ in this part as well as the next two parts. Let's denote the support domain of image $X_0(n,m)$ as $\Sigma = \{(n,m),\ n = 0,\ldots,N-1,\ m = 0,\ldots,M-1\}$. The 2×2 sub-sampled set of Σ is denoted as $\Sigma_S = \{(n,m),\ n = 0,\ldots,N/2-1,\ m = 0,\ldots,M/2-1\}$, which represents the support domain for the down-sized image $X_1(n,m)$. Let $\Sigma = \Pi \cup \Pi^{\perp}$, where $\Pi = \{(2i,2j),\ i = 0,\ldots,N/2-1,\ j = 0,\ldots,M/2-1\}$ and $\Pi^{\perp}$ is the complementary set. Let's assume (temporarily) that no pre-filtering is used in the 2×2 down-sampling process. According to Fig. 1, $X_1(n,m)$ is coded/decoded as $\ddot{X}_1(n,m)$, and $\ddot{X}_1(n,m)$ is then interpolated to produce $X'_0(n,m)$. Clearly, $X'_0(n,m) = \ddot{X}_1(n/2,m/2),\ \ \forall (n,m)\in\Pi$ (interpolation does not change these pixels); whereas other pixels $X'_0(n,m)$, $(n,m)\in\Pi^{\perp}$, represent the interpolated pixels. Similarly, we can partition the pixels of $\ddot{X}_0(n,m)$ into two sets according to Π and $\Pi^{\perp}$. Now, we start our analysis to illustrate that a quality deterioration would possibly happen to $\ddot{X}_0(n,m)$, $(n,m)\in\Pi$, in a very generic scenario where a typical bit-rate is employed at each layer.

To facilitate a quick understanding of this potential problem, let's first consider three extremes (as the special cases).

Case 1: Layer L_0 is coded *losslessly*. In this case, we have $\hat{R}_0(n,m)\equiv R_0(n,m),\ \forall (n,m)\in\Sigma$. Then, it is easy to understand that, no matter how layer L_1 is coded, it always helps improve the quality to add $\hat{R}_0(n,m)$ back into $X'_0(n,m)$; and in fact doing this yields an overall lossless coding: $\ddot{X}_0(n,m)\equiv X_0(n,m)$, $\forall (n,m)\in\Sigma$.

Case 2: Layer L_0 is *not* coded at all. In this case, we have $\hat{R}_0(n,m)\equiv 0,\ \forall (n,m)\in\Sigma$. Therefore, layer L_1 alone will provide the ultimate quality.

Case 3: Layer L_1 is coded *losslessly*. In this case, we have $X'_0(n,m)(n,m)\equiv X_0(n,m)$ and thus $R_0(n,m)\equiv 0$, $\forall (n,m)\in\Pi$. After a lossy coding at layer L_0, however, it is very likely to happen that $\hat{R}_0(n,m)\neq 0$ for many or even most $(n,m)\in\Sigma$. Thus, any position $(n,m)\in\Pi$ s.t. $\hat{R}_0(n,m)\neq 0$ will surely experience a quality drop if $\hat{R}_0(n,m)$ is added back into $X'_0(n,m)$.

In reality, *Case 1* would rarely occur (too costing), whereas *Case 2* destroys the hierarchical feature completely. On the other hand, *Case 3* will usually not occur either in practice. Nevertheless, it often happens that the upper layer is coded with a good quality while the residual layer is coded with a lower quality. In such a case, the quality on pixels $\ddot{X}_0(n,m)$, $\forall (n,m)\in\Pi$, may get decreased if $\hat{R}_0(n,m)$ is added back into $X'_0(n,m)$.

2.2. Characterization of coding errors

To reflect the reality, we assume from now on that a low-pass filtering is always applied before a 2×2 down-sampling in the hierarchical structure. At layer L_1, the coding happens on $X_1(n,m)$ directly, leading to the coding error $_1(n,m) = \ddot{X}_1(n,m) - X_1(n,m)$, $(n,m)\in\Sigma_S$. However, because $X_1(n,m)\neq X_0(2n,2m)$ (due to the low-pass filtering), a small adjustment is needed when one calculate the error between the reconstructed $\ddot{X}_1(n,m)$ and its ground-truth value $X_0(2n,2m)$: $_1(n,m)\leftarrow {}_1(n,m)+\sigma$. On the other hand, the coding at layer L_0 is not performed on $X_0(n,m)$, but $R_0(n,m)$ instead. Let's assume that the coding error at layer L_0 is $\Delta R_0(n,m) = \hat{R}_0(n,m) - R_0(n,m)$. After the error-compensation at the decoder side, one can get

$$\begin{aligned}\ddot{X}_0(n,m) &= X'_0(n,m) + \hat{R}_0(n,m)\\ &= X'_0(n,m) + R_0(n,m) + \Delta R_0(n,m)\end{aligned} \tag{1}$$

so that the reconstructed error at this layer is