
Available online: https://doi.org/10.1109/MMSP48831.2020.9287138

Copyright © 2020 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

See this record in Federation ResearchOnline at: http://researchonline.federation.edu.au/vital/access/HandleResolver/1959.17/184688
A Coarse Representation of Frames Oriented Video Coding By Leveraging Cuboidal Partitioning of Image Data

Ashek Ahmmed#, Manoranjan Paul#, Manzur Murshed*, and David Taubman†

# School of Computing and Mathematics, Charles Sturt University, Australia.

* School of Science, Engineering, and Information Technology, Federation University, Australia.

† School of Electrical Engineering and Telecommunications, University of New South Wales, Sydney, Australia.

Abstract—Video coding algorithms attempt to minimize the significant commonality that exists within a video sequence. Each new video coding standard contains tools that can perform this task more efficiently compared to its predecessors. In this work, we form a coarse representation of the current frame by minimizing commonality within that frame while preserving important structural properties of the frame. The building blocks of this coarse representation are rectangular regions called cuboids, which are computationally simple and has a compact description. Then we propose to employ the coarse frame as an additional source for predictive coding of the current frame. Experimental results show an improvement in bit rate savings over a reference codec for HEVC, with minor increase in the codec computational complexity.

Index Terms—Video coding, commonality, cuboids, HEVC.

I. INTRODUCTION

Nowadays multimedia contents like video are watched and shared using a variety of display devices and in majority of the web platforms. This relentless expansion of video-based applications has been facilitated by video coding. Video coding, which refers to the process of reducing the bit budget required for a given subjective reproduction quality of a video signal, still remains one of the most challenging problems in signal processing. With the aid of video coding, digital video signal is converted into a format suitable for transmission and storage, while minimizing the number of bits required to represent it. Video coding systems leverage on the fact that a video sequence consists of frames, sampled at a specific sampling frequency, and there exists significant commonality within a frame as well as between successive frames.

Modern video coding standards are block-based. The frame that needs be coded, known as the current frame, is artificially partitioned into square or rectangular shape blocks by grouping neighboring pixels. Then for each current frame block, an identical shape block is located in the set of already coded frame(s), known as reference frame(s), by minimizing a rate-distortion criteria. This newly obtained block is taken as the prediction for the current block and thus only motion vector and some residual information need be communicated from the encoder.
Partitioning the target frame into square or rectangular blocks, without taking into account the underlying motion vector field, can generate high prediction error, especially around the vicinity of moving object boundaries. To compensate for the prediction error, a popular approach, proposed by Chan et al. [2], is to partition the current block into smaller square or rectangular sub-blocks with the aim of matching the blocks to the moving objects in a video scene. For example, the widely used video coding standard H.264/AVC [3] supports several types of block partitions from $4 \times 4$ to $16 \times 16$ pixels and the current standard, the HEVC [4], provides a range of symmetric and asymmetric partitions with the maximum block size of up to $64 \times 64$ pixels, as can be seen from Fig. 1. The emerging video coding standard, the versatile video codec (VVC) allows maximum block size of $128 \times 128$ pixels with a multi-stage splitting structure where in the first stage, the target block is partitioned using HEVC like quad-tree structure and in the second stage, each resulting sub-block may further be split horizontally and vertically into binary or ternary blocks [5].

Another direction of works [6]–[15] also partition motion blocks into smaller sub-blocks. These approaches consider the actual underlying motion discontinuities, through the employment of slanted and arbitrary positioned partitioning of blocks. Segmentation-based video coding paradigms offer more flexible partitioning. Performing segmentation over the reference frame(s) [16]–[18], rather than in the current frame [19], [20], makes it possible to use information from additional frames or even other media types e.g. depth maps as evidence within the segmentation estimation step.

Murshed et al. proposed the cuboidal partitioning of image data (CuPID) algorithm in [21]. In CuPID framework, an image is partitioned into relatively-homogeneous rectangular regions, known as cuboids, by employing statistical features. These features are derived from the image pixel intensity distribution. The target image is recursively split into halves at a hyperplane orthogonal to one of the axes whereby a greedy optimization heuristic, equipped with sum of the information entropy in the split-pair halves as the objective function, is used to find the minimizing hyperplane. Cuboids attempt to preserve moving object boundaries with different objects present in the scene are described by separate cuboids, they are computationally efficient and can be described in a compact way [22]–[24].

Since the capacity of communications systems is currently being outpaced by the increasing demand for access to audiovisual services, techniques that can model the available mutual information within a video sequence more efficiently compared to the state of the art video coding systems demand attention. Paul et al. proposed an approach in [25] that generated a frame, known as the most common frame in a scene, by carefully selecting highly similar features that are present in the first 25 frames of a video sequence. This common frame was encoded as an I-frame and used as a reference frame for the P-frames along with their usual temporal reference(s).

In this paper, we propose to generate a coarse representation of the current frame, $C$, at cuboid level. This coarse frame, denoted by $R_f$ herein, can be considered a low-frequency version of the frame $C$. The frame $R_f$ can (i) provide significant structural information about $C$, (ii) costs fewer bits to encode due to its compact representation, and (iii) adds minor increment to the overall computational complexity in the reference codec even for 4K resolution sequences. An example of these frames are shown in Fig. 2. Experimental results show if the frame $R_f$ is employed as a reference frame for encoding the frame $C$, bit rate savings can be obtained due to the improved modeling of commonality.

The rest of the paper is organized as follows: In section II, we describe the generation process of the coarse frame $R_f$. Section III, describes the architecture of the employed codec. Experimental results are reported in section IV. Finally, in section V, we present our conclusions.

II. COARSE REPRESENTATION OF THE CURRENT FRAME USING CUPID

In this section, the generation process of cuboids, over the current frame, using the CuPID algorithm is discussed. Next, questions like how these estimated cuboids are employed to produce the coarse representation of the current frame both
Fig. 3. Block diagram of the coding/decoding framework that uses the coarse representation of the current frame, $C$ i.e. the frame $C_{lf}$ as a reference frame, along with the usual temporal reference(s).

Algorithm 1: coarseRepresentation($C_i, \{s^i\}_{i=1}^{n-1}$)

**Input**: the current frame; partitioning indices to describe the cuboids

**Output**: a coarse representation, $R_{lf}$, of the current frame

1. begin
   2. $\{C^{(i)}\}_{i=1}^{n} \leftarrow$ cuboidExtraction($C, \{s^i\}_{i=1}^{n-1}$);
   3. for $i = 1 : n$ do
      4. $m(i) \leftarrow$ meanIntensity($C^{(i)}$);
      5. $R_{lf}[C^{(i)}] := m(i)$;
   6. end
   7. return $C_{lf}$
   8. end

At that point, the set $\{s^i\}_{i=1}^{n-1}$ represents the cuboid map obtained over the current frame $C$. It is necessary to communicate the optimal partitioning indices $s^i$ values from the encoder to the decoder in order to reproduce the cuboid map at the decoder. The Exponential-Golomb coding technique [27] is used to encode the indices and then these encoded indices are augmented to the bitstream.

B. Generation of the coarse frame $R_{lf}$ for the current frame

Next, for each obtained cuboid, $C^{(i)}$, a representative intensity value, $m(i)$, is computed. This value is taken to be the mean pixel intensity, considering all the pixels $p$ and their intensities $p(x, y)$ within the coverage of the corresponding cuboid.

$$m(i) = \frac{\sum_{p \in C^{(i)}} p(x, y)}{|C^{(i)}|}$$

(3)

After this every encompassing pixel $p \in C^{(i)}$ intensity gets replaced by the value $m(i)$. This procedure is repeated for every cuboid of the obtained cuboid map which in turn generates a coarse representation of the current frame, $C$. Fig. 2 shows examples of this coarse representation frame $R_{lg}$. These low-frequency representations attempt to model the commonality that exist within the frame $C$ and the degree of detailed information they can communicate tend to increase with growing number of cuboids, $n$. To reproduce the frame $R_{lf}$ at the decoder, cuboid partitioning indices $\{s^i\}_{i=1}^{n-1}$ and the mean intensity vector $\mathbf{m} \in \mathbb{Z}_n^a$ are coded using the Exponential Golomb Coding technique [27] and then transmitted from the encoder. Each cuboid mean pixel intensity value is of same data type as an individual pixel’s.

C. Determination of the number of cuboids

The CuPID algorithm takes as inputs: the frame for which the cuboid map need be generated and the number of cuboids, $n$, that should belong to the map. With increasing number of $n$, both the subjective and objective quality of the coarse frame, $R_{lf}$, increase as can be seen from Fig. 2. However, these gains come at increased bit rate and computational time required to produce the frame $R_{lf}$. In particular, since the cuboid maps and the mean intensity values are estimated at the encoder, therefore computational complexity of the encoder increases while any increase in the decoder side complexity remains minor. Table I enlists the average computational time (in seconds) required to generate a $C_{lf}$ frame, its corresponding PSNR and bit requirements.
Fig. 4. (From left to right) Rate distortion performance of two different coding strategies for the \textit{ParkAndBuildings}, \textit{Vehicles} 4K video sequence. Bit rate savings are achieved by using the $R_f$ frames as reference for the frames with POCs \{0, 32, 64, 96\} and the rest of the frames get conventionally coded as P-frames.

### TABLE I

<table>
<thead>
<tr>
<th>$n$</th>
<th>500</th>
<th>2000</th>
<th>4000</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T = t_{\text{coarse}} / t_{e}$</td>
<td>0.88%</td>
<td>2.76%</td>
<td>9.14%</td>
</tr>
<tr>
<td>PSNR (in dB)</td>
<td>20.44</td>
<td>21.49</td>
<td>22.13</td>
</tr>
<tr>
<td>Bits</td>
<td>32454</td>
<td>48706</td>
<td>94310</td>
</tr>
</tbody>
</table>

### III. STRUCTURE OF THE CODING/DECODING ARCHITECTURE

In typical video coding systems, the first frame, with POC 0, is coded as an I-frame. The next frame, with POC 1, is inter-coded as a P-frame using the already coded frame 0 as a reference frame.

We propose to modify this architecture with the aim of providing the encoder an alternative coding tool that attempts to model the commonality within a frame in a more effective way. Figure 3 shows a simplified block diagram of the proposed codec. For the frame with POC 0, we generate a coarse representation of it using the procedure described in section II. This newly obtained frame, $R_f$, is then used as a reference frame for the current frame (POC 0). It means, in the proposed paradigm, the first frame is coded as a P-frame. The following frame, with POC 1, is coded conventionally as a P-frame. However, for motion compensated prediction, the encoder can now use an improved version of the coded frame 0 as the reference. Next, the frame with POC 2 is coded as a P-frame using the already coded and improved frame 1 as the reference frame.

In practice, the cuboid-based coarse representation reference frame, $R_f$, is added to the reference picture list for inter-prediction of the current frame with POC 0. The encoder makes a rate-distortion (RD) optimized decision on a prediction unit basis, adapting them with motion vectors and coded residuals where necessary.

### IV. EXPERIMENTAL ANALYSIS

The RD performance of the proposed coding paradigm is investigated, on 5 different 4K video sequences: 2 of which part of the JVET test sequences and the other 3 are part of the data set in [1].

The first $N$ frames of each 4K sequence, where the value of $N$ is either 100 or 120 depending on the frame rates, are coded by the SHM 12.4 reference software [28] for the scalable extension of HEVC using its single layer architecture. That means the each frame only gets to predict from temporal reference frame just like conventional HEVC encoder. Low delay P- GOP structure i.e. IPPP...P is employed as per the common test conditions [29]. Intra period of 32 is used. Four different quantization parameter values (QP = 27, 32, 37, 42) are tested.

As for the modified paradigm, for each of the 4 frames with POCs \{0, 32, 64, 96\} respectively, a coarse representation frame $R_f$ is generated. These 4 coarse frames are then grouped together and supplied to the SHM encoder, configured as multi-layer scalable codec, as a base layer to perform quality scalability. The QP values between the base layer and enhancement layer are kept the same and inter-layer motion estimation was allowed. These modifications ensured that frames with POCs \{0, 32, 64, 96\} are coded as P-frames, using their corresponding coarse frames as a reference, unlike the single layer case where these frames were coded as I-frames.

For our experimental analysis, each $R_f$ frame was generated using the number of cuboids, $n = 2000$. The rationale behind selecting this particular $n$ is that the maximum coding tree unit (CTU) size in HEVC is $64 \times 64$ pixels and the ratio of a picture's luminance component resolution to this maximum CTU size is closer to the number $n = 2000$. That means the estimated cuboid map does not contain any cuboid with size smaller than $64 \times 64$ pixels. The bits required to represent each coarse frame $R_f$ is calculated and added to the bit rate obtained from the multi-layer SHM encoder for calculating the rate-distortion (RD) results.
In this paper, we have proposed to generate a coarse representation of a target frame that is conventionally coded as an I-frame. This coarse representation is computationally efficient, has a compact representation, and can capture important structural information pertinent to the current frame. Afterwards, this coarse frame is employed as a reference frame for the current frame. Experimental results show a bit rate savings of up to 19% over an HEVC encoder.

TABLE II
THE BØJNTEGAARD DELTA GAINS OBTAINED FOR THE TEST SEQUENCES OVER HEVC WHEN THE COARSE REPRESENTATION-BASED REFERENCE IS EMPLOYED.

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Num_frames</th>
<th>Delta rate</th>
<th>Delta PSNR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vehicles@50Hz</td>
<td>100</td>
<td>−17.46%</td>
<td>+0.50 dB</td>
</tr>
<tr>
<td>ParkAndBuildings@50Hz</td>
<td>100</td>
<td>−19.11%</td>
<td>+0.56 dB</td>
</tr>
<tr>
<td>Book@50Hz</td>
<td>100</td>
<td>−10.37%</td>
<td>+0.35 dB</td>
</tr>
<tr>
<td>CatRobot@60Hz</td>
<td>120</td>
<td>−10.40%</td>
<td>+0.29 dB</td>
</tr>
<tr>
<td>DaylightRoad2@60Hz</td>
<td>120</td>
<td>−3.1%</td>
<td>+0.10 dB</td>
</tr>
</tbody>
</table>

Fig. 5. A coarse representation frame, R_	ext{g}, from the 4K DaylightRoad2 sequence. The R_	ext{g} frame is generated using the number of cuboids, n = 2000.

Fig. 4 shows the RD curve for the test sequences ParkAndBuildings, Vehicles, and CatRobot sequence while Table II tabulates the Bjøntegaard Deltas [30] for all test sequences under consideration.

The employed hybrid prediction paradigm generates a bit rate saving for all test sequences. The maximum gain in bit rate and PSNR is obtained in the ParkAndBuildings sequence. It can be observed from Fig. 4 that both in low bit rate and high bit rate cases, the proposed encoding scheme is outperforming the reference coder for HEVC. The minimum gain is observed over the DaylightRoad2 sequence. Fig. 5 depicts that frames from this sequence contain slanted straight lines which are difficult to describe using rectangular cuboids. Increasing the number of cuboids in such cases or the employment of arbitrary shaped cuboids demand attention.

V. CONCLUSIONS

In this paper, we have proposed to generate a coarse representation of a target frame that is conventionally coded as an I-frame. This coarse representation is computationally efficient, has a compact representation, and can capture important structural information pertinent to the current frame. Afterwards, this coarse frame is employed as a reference frame for the current frame. Experimental results show a bit rate savings of up to 19% over an HEVC encoder.

ACKNOWLEDGMENT

The authors would like to thank Dr. Vadim Seregin, of Qualcomm Technologies Inc., for sharing his knowledge and expertise about the SHM reference software for scalable HEVC.

REFERENCES


Figures and tables are attached.


