GEOMETRY-ADAPTIVE BLOCK PARTITIONING FOR VIDEO CODING

Òscar Divorra Escoda, Peng Yin

Thomson
Corporate Research
Princeton, NJ, USA
{oscar.divorra,peng.yin}@thomson.net

*Congxia Dai**, *Xin Li*

West Virginia University
Department of CS&EE
Morgantown, WV, USA
{xin.li,congxiad}@csee.wvu.edu

ABSTRACT

Frame partitioning is a process of key importance in efficient video coding. Most recent video compression technologies, like H.264/AVC, use tree based frame partition. This reveals to be more efficient than simple uniform block partition, typically used in older video coding standards like MPEG-2 or H.263. However, tree based frame partition still does not code efficiently enough video information, as is unable to capture the geometric structure of 2D data. During last years, several works have been developed, mainly in the domain of still image representation and coding, in order to solve such limitations. An example is the use of wedge partitions. Based on these, in this paper, we study a way to better represent and code 2D video data by taking its 2D geometry into account. Our study is developed as an extension of H.264/AVC. Geometry-adaptive partitions are used to improve Intra and Inter prediction modes. Results obtained with the investigated method show that both better R-D and visual performance can be achieved.

Index Terms— Video Coding, Geometry-Adaptive partitions, Wedges, Motion compensation, Intra/Inter Prediction

1. INTRODUCTION

Tree-structured macroblock partition is a common and useful tool in current major video coding standards such as H.264/AVC [1]. Introduced long ago, tree-structured block partitions allow for better R-D coding performances as they exploit the multi-scale nature of video data. This is the case for motion data [2], as well as intra data [1]. Indeed, H.264/AVC has reported a very good coding performance by supporting 16x16, 16x8, 8x16, 8x8, 8x4, 4x8 and 4x4 tree-based block partitions, where a *uniform* prediction mode is used for each leaf. It is a well known fact that bigger blocks will be used in picture areas where pixels can share the same type of prediction information (e.g. same motion vector). However, if we consider Fig. 1, where two regions $S(x, y)$ and $R(x, y)$ are separated by a contour with different motions and textures, smaller size blocks will tend to cumulate around the boundary, i.e. it is generally more difficult to predict both regions with a single motion model. Sometimes, near region edges, such a tree based partition leads to separately code in different sub-blocks similar data, leading to an unnecessary overhead.

Based on the observation of tree-based partitioning limitations, some works have been conducted in order to supply more flexible block partitioning for efficient video coding. This is the case for [3, 4] and most recently [5][1]. They focus on the definition of alternative macroblock (MB) partitions for Inter data prediction. [3] partitions lack of real edge structure, while [4] uses a partition description unable to deal with the different statistical properties of local geometric information such as location and orientation.

This paper presents a general framework for the introduction of geometry-adaptive partitions within blocks for efficient Intra and Inter video data coding. Our work builds on the concepts proposed by recent theoretical studies for efficient coding of piecewise-smooth data [6, 7]. Indeed, we assume that video data such as motion (and, of course, Intra data) have a piecewise-smooth structure (e.g. regions $S(x, y)$ and $R(x, y)$ in Fig. 1 have homogeneous characteristics separated by an edge). Based on these concepts we discuss and propose the use of additional coding modes with geometric partitions in H.264/AVC. These modes are intended for coding Inter as well as Intra data. We discuss how these modes interact with the rest of H.264/AVC coding structure as well as examples of efficiency gains.

The paper is structured as follows: Sec. 2 briefly discusses the rate-distortion motivations for introducing geometry-adaptive block partitioning in video coding. Then, Sec. 3 presents a way of extending H.264/AVC for the use of geometric block partitions. Results of the proposed framework are discussed in Sec. 4. Finally, conclusions are drawn in Sec. 5.

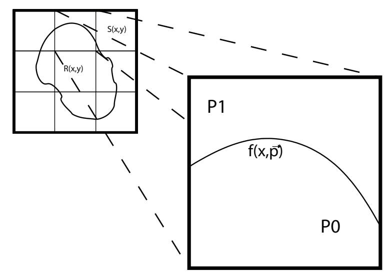

Fig. 1. 2D data where regions $S(x, y)$ and $R(x, y)$ are separated by a boundary. Each region can be associate with a different model, and region boundary can be approximated by means of a parametric boundary model.

2. EFFICIENT CODING OF 2-D DATA WITH PIECEWISE-SMOOTH CHARACTERISTICS

Recent studies in still image coding, and multi-dimensional data approximation, have shown that tree structures tend to be R-D suboptimal when used to code data with piecewise-smooth characteristics [6, 7]. This is because tree partitioning, with homogeneous approximation models within leaves, are often unable to exploit the existing redundancy along region boundaries. As shown in Fig. 1, one may intuitively consider that a good compression approach should exploit region shapes and code, with no further splitting, each one

*Congxia Dai contributed to this work while at Thomson, Corporate Research, Princeton, NJ, USA

[1]The parallel work [5] was still unpublished at the time of submission but the authors wanted to cite it as it had just been posted available on the web. Despite some similarities, our work presents a wider framework.

of the disjoint regions with homogeneous characteristics: $P0$ and $P1$. This is what has been actually proved in terms of R-D for the case of piecewise-smooth images [6]. In [6], the authors show that distortion reduces with the rate, in a way close to the optimal, when wedge-like partitions (see Fig. 2) are used within tree leaves when piecewise-smooth images, having smooth contours[2], are to be coded.

In video coding, and in particular for intra coding, similar assumptions to those of [6] may be considered, as the same kind of data is to be encoded (in Sec. 3 we briefly discuss how directional prediction in H.264 fits into this). A slightly different framework may be inter data coding, where motion compensation (MC) is used. However, if we assume that motion fields may have also a piecewise-smooth structure where there are different regions with different homogeneous motions, then the use of a partition strategy that better encodes motion edges can lead to more efficient coding. This may be specially the case for occlusions, motion bendings, etc.

Based on the above considerations, R-D asymptotic behavior of encoded prediction video frames can be argued to be analogous to the study developed in [6]. Indeed, for a classic quadtree partition for MC, asymptotic dependency of distortion with the rate can be found to be:

$$D_{Tree}\left(R\right) \sim \frac{\log R}{R}. \qquad (1)$$

Very roughly, this can be justified from the fact that, for a tree of J levels, the number N_J of quadtree leaves that cross the smooth motion contour would be such that $N_J \sim 2^J$, then $D_{Tree}(J) \sim N_J 2^{-2J} \sim 2^{-J}$ as distortion, in an ideal situation, may be assumed to mainly come from the leaves crossing the motion boundary and distortion in each of such leaves may be considered, in average, proportional to its area ($\sim 2^{-2J}$). In what concerns rate, $R(J) \sim J2^J$, as the total number of leaves grows also like $\sim 2^J$ and the number of bits per leaf can be assumed to be of the order of J if the search range is assumed to be of the order of the size of the biggest of possible tree partitions. See [6, 7] for further detail in this kind of analysis. If we compare Eq. 1 to the result obtained when wedge-like partition are used within tree leaves [6]:

$$D_{Wedge-Tree}\left(R\right) \sim \frac{\log R}{R^2}, \qquad (2)$$

we can observe that decay of distortion with the rate is faster in the second case, leading to more efficient coding.

Theoretical results for piecewise-smooth data approximation with higher order region boundaries may be found in [7].

3. GEOMETRY-ADAPTIVE BLOCK PARTITION FOR VIDEO CODING: AN H.264/AVC EXTENSION

In this section, based on previous discussions, we propose a framework for the use of local geometry parametric models to extend state of the art video codecs such as H.264/AVC [1].

Similarly to [3, 4], given a block of a frame to be predicted, a new mode is tested in addition to those based on classic tree partitioning. In our case, the concerned block or region is partitioned into two regions divided by a parametric boundary model $f(x, y, \vec{p})$ (see Fig. 1), where x, y represent the spatial coordinates, and $\vec{p}$ stands for the parameters. In particular, in our present implementation, a first order polynomial model is used to generate the splitting *wedges* in prediction blocks. Each one of the partitions generated is predicted by means of the most appropriate predictor in terms of R-D. In the following, we describe the details concerning the partition, the two tested settings (Inter and Intra) and how these are inserted within the H.264/AVC framework.

[2]In [6], smooth contours were considered to be curves two times continuously differentiable.

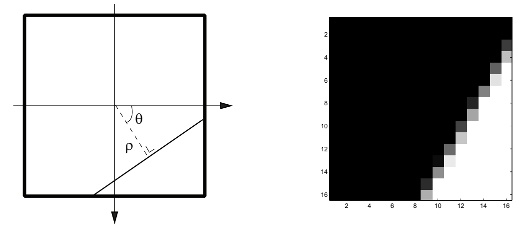

Fig. 2. Left: Line partition of a block based on geometric parameters θ and ρ. Right: Example of wedge-like partition with $\theta = 30$ and $\rho = 4$. White color indicates one of the partitions, black marks the complementary partition. Gray intermediate values show "partial-surface" pixels.

3.1. Defining Geometry-Adaptive Block Partitions

In this paper, geometric partitions within blocks are modeled by the implicit formulation of a line. Hence, partitions are defined such that given:

$$f(x, y) = x\cos\theta + y\sin\theta - \rho, \qquad (3)$$

each pixel (x,y) is classified such that:

$$GEO\_Partition(x, y) = \begin{cases} if\ f(x, y) > 0 & Partition\ 0 \\ if\ f(x, y) = 0 & Line\ Boundary \\ if\ f(x, y) < 0 & Partition\ 1 \end{cases}$$

The line generated by the zero level line of Eq. 3 is defined in terms of its angle θ and distance ρ (see Fig. 2). Unlike the formulation proposed in [4], the representation of the boundary in terms of θ and ρ allows to cover a block more uniformly with lines, it helps exploiting at the coding stage the different statistics that ρ and θ present, and to independently set the resolution of sampling for each parameter depending on the general quantization parameter (QP) (this necessary fact is disregarded in [5, 4]). Moreover, according to [7] higher order curve parameters can be more coarsely sampled than lower ones.

However, the digital adaptation of wedge partitioning must handle that some pixels can not be fully classified in either of the partitions. This is because the continuous version of the line crosses such pixels. Fig. 2 shows in black and white those pixels belonging to each partition, and in gray those "partial-surface" pixels that fall in between both partitions. In order to supply visually natural predicted blocks, such "partial-surface" pixels require their prediction value to be computed as a linear combination of their corresponding value if they were fully attached to each partition.

3.2. Geometry-Adaptive Inter and Intra Prediction Modes

The use of geometry-adaptive block partitions for inter prediction, relies on the assumption that motion fields in video coding may present regular boundaries. As an example, this is a reasonable assumption in the case where two textured regions, having different motion, are occluding one to the other. For this purpose, we extend H.264/AVC by introducing, at each tree partitioning scale, an additional mode where motion is compensated according to the geometric partitioning of a block (see Fig. 3, right). In this additional coding mode, the encoded information is an index to describe θ, an index to describe ρ and the necessary motion information for each of the partitions. Within the geometry-adaptive motion compensation mode, a search on θ, ρ and motion vectors is performed in order

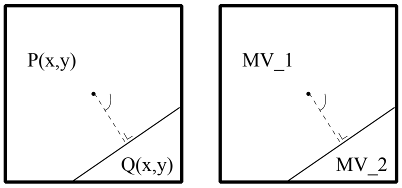

Fig. 3. Example of two blocks with a geometric partition. In the left block, spatial prediction or model fitting are used to model intra data in each geometric partition. In the right, motion compensation based prediction is used to model inter data geometric partition.

to find the best configuration in an R-D sense. If we consider a full search strategy, this can be done in two stages, for every θ and ρ pair, the best motion vectors in a R-D sense are searched.

Intra prediction modes in H.264/AVC are also based on a quadtree partition strategy of the signal: 16x16, 8x8 (FRext), 4x4 block sizes are used. This is a limiting feature in terms of R-D optimality. However, the fact that directional prediction is used, means that some geometric characteristics of intra pictures are already considered by H.264/AVC. This can be further improved by trying to capture geometry from the picture at the partition stage itself (in addition to the prediction stage). Moreover, given the constraint of causality that intra predictors have, appropriate partitioning in a geometric sense may help overcome H.264/AVC prediction errors like those shown in Fig. 5. Fig. 3 left, shows that given a partition described by θ and ρ, we may use different modeling strategies $P(x, y)$ and $Q(x, y)$ for each partition. Modeling strategies, among other, can be directional prediction as performed in H.264/AVC, DC prediction and/or plane prediction. Moreover, for 16x16 geometry-adaptive blocks, we may consider also the possibility of additional prediction directions to those proposed by H.264/AVC. Of course, this poses causality problems depending on the prediction direction. In such a case, non-predictable partitions (such as $Q(x, y)$ in Fig. 3) can be alternatively encoded by fitting some polynomial model and explicitly signaling its quantized parameters. A simple example of this is to signal the DC component of the partition.

3.3. Implementing Geometry-Adaptive Modes within H.264/AVC

To test our method, we introduce a new inter and intra mode based on the geometric division of a 16x16 MB, named as INTER16x16_GEO, and INTRA16x16_GEO respectively. INTER16x16_GEO is inserted after INTER8x16 and before INTER8x8. Though INTER16x8 and INTER8x16 can be seen as a special case of INTER16x16_GEO, the percentage of using these two modes is relatively high. So we stick to H.264/AVC design in order to save bits for coding the partition parameters. INTRA16x16_GEO is inserted before INTRA16x16.

For coding purposes, a dictionary of possible partitions is *a priori* defined for $\rho : \rho \in [0, \frac{\sqrt{2}MB_{Size}}{2})$, where $\rho \subset \{0, \Delta\rho, 2 \cdot \Delta\rho, \dots\}$, and $\theta : \{if\ \rho = 0\ then\ \theta \in [0, 180),\ else\ \theta \in [0, 360)\}$, where $\theta \subset \{0, \Delta\theta, 2 \cdot \Delta\theta, \dots\}$, being $\Delta\rho$ and $\Delta\theta$ the selected sampling steps. The sampling indices for ρ and θ are coded to represent the partition edge. We note that for $\rho = 0$, angles 0 and 90 are redundant with modes INTER16x8 and INTER8x16, so they are removed. For simplicity, we model ρ and θ as two independent components. Boundary parameters from upper block and left block are used for prediction, in case a geometric mode is used in such blocks, and the difference is entropy coded using variable length coding, to be more specific, Exp-Golomb entropy coding. If there is no boundary in neither upper or left block, the edge is coded without prediction. In such a case, θ is assumed to have uniform distribution (fix length code is

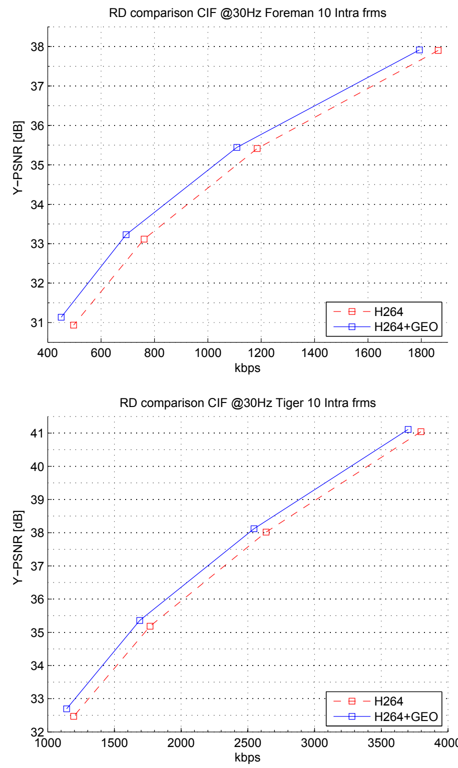

Fig. 4. Intra data R-D coding results using additional INTRA16x16_GEO mode in H.264/AVC. Above: R-D comparison results for Foreman sequence. Down: results for Tiger sequence.

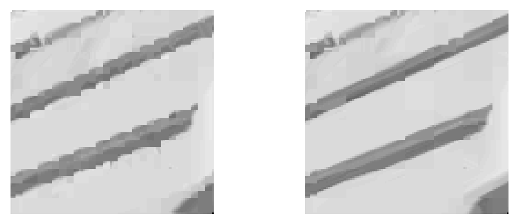

Fig. 5. Intra Predicted picture results (residual has been subtracted from pictures): Left: shows the H.264/AVC prediction. Right: prediction generated by our geometry extended H.264/AVC.

used) and ρ is assumed to have an exponential distribution (variable length coding is used).

For INTRA16x16_GEO mode, in each partition, we will consider to fill the prediction with two different modes: directional prediction from neighboring decoded pixels or the DC value of the data. A flag is used to indicate which mode is selected.

Concerning de-blocking filter, no change is introduced for INTRA16x16_GEO mode with respect to other Intra modes. However, de-blocking filter is permanently enabled on MB boundary 4x4 blocks for INTER16x16_GEO mode.

4. EXPERIMENTAL RESULTS

In this section, JSVM 6 reference code [8] has been used as H.264/AVC compliant codec for generating results. For the rest of the section, average compression gains are given according to the convention proposed in [9]. Encoder settings for H.264/AVC and for geometry

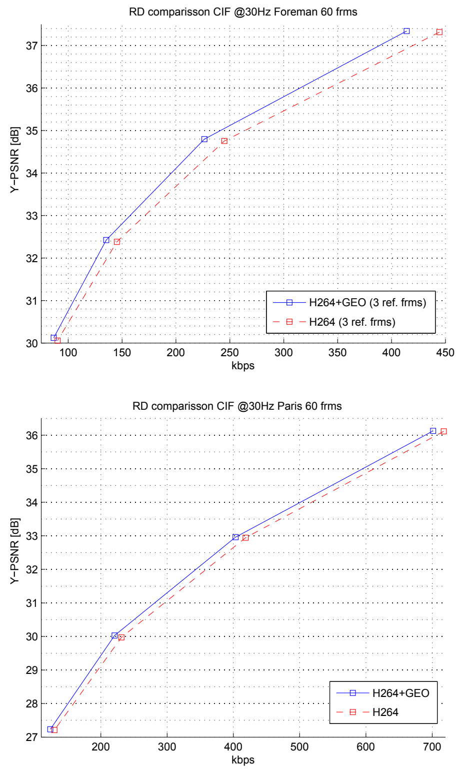

Fig. 6. Inter data R-D coding results using the additional INTER16x16_GEO mode in H.264/AVC. Above: R-D comparison for Foreman sequence. Down: R-D comparison for Paris sequence.

extended H.264/AVC are exactly the same with the exception of geometric extensions. All tests were performed in CIF resolution except for the case of Tiger which is 480x480. Settings such as de-blocking filter and FRext were left on. All sequences have been encoded using the following angle and radius resolutions: $\Delta\theta = 11.25$ degrees and $\Delta\rho = 1$ pixel.

For simplicity, at this point, we just consider, as possible coding modes, for geometric partitions, the DC value or directional prediction with the same direction as the partition edge. This coding mode is tested in concurrence with other H.264/AVC Intra coding modes. R-D coding results over 10 frames of sequence Foreman and Tiger can be found in Fig. 4. As shown by the results, Intra geometry-adaptive modes can better exploit strong geometric content of pictures, with observed compression gains of 8.69% and 5.65% for Foreman and Tiger Intra data respectively. More graphically, one of the effects of such a mode is observed in Intra predicted data when using geometry-adapted coding. Standard H.264/AVC directional prediction fails when picture structures are non-causal given a scanning order. Akin to Intra data, Inter data can also profit from geometry-adapted block partitions to increase coding efficiency. For the present results, we only use geometric partitions within 16x16 coding blocks. As reflected in Fig. 6, coding gains of 7.6% and 4.46% are obtained for the first 60 frames of Foreman and Paris respectively. In the case of Foreman, multiple reference frame MC was tested (3 ref. frames), Paris just uses single reference frame MC.

Finally, Fig. 7 shows the visual impact of geometry-adaptive partitions for QP=36 in Foreman sequence (frame 7th). H.264/AVC with geometric extensions has a sharper look with higher detail around contours. Moreover, higher detail on the neck is observed as well as

Fig. 7. Foreman sequence: 7th Reconstructed picture visual detail. Left: standard H.264/AVC result. Right: H.264/AVC with geometric extensions.

Foreman upper lib is better reconstructed when geometry-adaptive modes are used.

5. CONCLUSIONS

In this paper, geometry-adaptive block partition for video coding is presented as an improvement over classic quadtree partition for inter and intra coding. Despite only 16x16 block geometric partitions have been tested, and a limited choice of prediction modes within geometric partitions have been used, interesting efficiency improvements have been achieved compared to H.264/AVC. Additional prediction modes, further explicit coding models (other than DC), and the use of Intra and Inter 8x8 geometry-adapted modes should further improve achieved performances. Also, geometric parameter coding may be further improved. An important issue to address, though, is the complexity requirements for geometric partition selection at the encoder side. To the contrary, at the decoder side, the discussed framework does not require a significant complexity increase. Future work involves research on efficient and accurate fast selection strategies at the encoder for geometry-adaptive modes.

REFERENCES

[1] ITU-T, *ITU-T Recommendation H.264*, Mar 2005.

[2] G. J. Sullivan and R. L. Baker, “Rate-distortion optimized motion compensation for video compression using fixed or variable size blocks,” in *GLOBECOM*, Phoenix, USA, Dec 1991, IEEE.

[3] S. Kato, S. Sekiguchi, S. Adachi, and M. Etoh, “Performance evaluation on h.26l-based motion compensation with segmented multiple reference frames,” in *ICIP*, Rochester, USA, Sep 2002.

[4] S. Kondo and H. Sasai, “A motion compensation technique using sliced blocks in hybrid video coding,” in *ICIP*, Genoa, Italy, Sep 2005.

[5] E. M. Hung, R. L. De Queiroz, and D. Mukherjee, “On macroblock partition for motion compensation,” in *ICIP*, Atlanta, USA, Oct 2006, IEEE.

[6] M. N. Do, P. L. Dragotti, R. Shukla, and M. Vetterli, “On the compression of two-dimensional piecewise smooth functions,” in *ICIP*, Thessaloniki, Greece, Oct 2001.

[7] V. Chandrasekaran, M. Wakin, D. Baron, and R. Baraniuk, “Surflets: A sparse representation for multidimensional functions containing smooth discontinuities,” in *ISIT*, Chicago, USA, Jul 2004, IEEE.

[8] ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q. 6, *JSVM 6 Software*, Apr 2006.

[9] G. Bjontegaard, “Calculation of average psnr differences between rd-curves,” Tech. Rep., VCEG-M33, ITU-T Q.6/SG16, Apr 2001.