Signal Processing: Image Communication 63 (2018) 113–124

Contents lists available at ScienceDirect

Signal Processing: Image Communication

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

# Optimal Lagrange multipliers for dependent rate allocation in video coding

Ana De Abreu [a,*], Gene Cheung [b], Pascal Frossard [c], Fernando Pereira [d]

[a] Trinity College Dublin, Dublin, Ireland
[b] National Institute of Informatics, 2-1-2, Hitotsubashi, Chiyoda-ku, Tokyo, 101–8430, Japan
[c] Signal Processing Laboratory (LTS4), Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland
[d] Instituto Superior Técnico - Instituto de Telecomunicações (IST-IT), Lisbon, Portugal

ARTICLE INFO

*Keywords:*
Lagrangian optimization
Video and multiview image coding
Rate–distortion (RD) optimization

ABSTRACT

In a typical video rate allocation problem, the objective is to optimally distribute a source rate budget among a set of (in)dependently coded data units to minimize the total distortion. Conventional Lagrangian approaches convert the lone rate constraint to a linear rate penalty scaled by a multiplier in the objective, resulting in a simpler unconstrained formulation. However, the search for the "optimal" multiplier – one that results in a distortion-minimizing solution among all Lagrangian solutions that satisfy the original rate constraint – remains an elusive open problem in the general setting. To address this problem, we are the first in the literature to construct a computation-efficient search strategy to identify this optimal multiplier numerically in the general dependent coding scenario. Specifically, we first formulate a general rate allocation problem where each data unit can be dependently coded at different quantization parameters (QP) using a previous unit as predictor, or left uncoded at the encoder and subsequently interpolated at the decoder using neighboring coded units. After converting the original rate-constrained problem to the unconstrained Lagrangian counterpart, we design an efficient dynamic programming (DP) algorithm that finds the optimal Lagrangian solution for a fixed multiplier. In extensive monoview and multiview video coding experiments, we show that for fixed target rate constraints, our algorithm is able to find the optimal multipliers in a distortion minimum sense among all Lagrangian solutions. Moreover, we show that our simple solution is able to compete with complex rate control (RC) solutions used in video compression standards such as HEVC and 3D-HEVC, which outlines the importance of the proper choice of the Lagrangian multipliers.

## 1. Introduction

In video coding, rate allocation is the problem of distributing a source bit budget $B$ to a set of (in)dependently coded data units, $v \in \mathcal{V}$, in order to minimize the distortion of all units. For example, a data unit $v$ can be a video frame predictively coded at a quantization parameter (QP) $q_v$, using a previous frame as a predictor. Coding at a larger (coarser) QP requires fewer bits in general but results in a higher quantization distortion. In some cases, leaving a unit uncoded at the encoder may be a better rate–distortion (RD) decision; the unit is subsequently interpolated at the decoder using neighboring coded units via techniques such as motion compensated interpolation (MCI) [1] in monoview video or depth-image based rendering (DIBR) [2,3] in multiview video when color and depth maps are available. In these cases, the more general rate allocation problem is to first select data units $\mathbf{v} \subseteq \mathcal{V}$ for coding, and then select a set of QPs $\mathbf{q} = \{q_1, \ldots, q_{|\mathbf{v}|}\}$ at which to code the selected units $\mathbf{v}$ to minimize the total distortion $D(\mathbf{v}, \mathbf{q})$ subject to a rate constraint $R(\mathbf{v}, \mathbf{q})$:

$$\min_{\mathbf{v}\subseteq\mathcal{V},\mathbf{q}} D(\mathbf{v}, \mathbf{q}) \quad \text{s.t. } R(\mathbf{v}, \mathbf{q}) \leq B. \tag{1}$$

To address different variants of the rate allocation problem, Lagrangian approaches – where the lone rate constraint is first converted to a linear rate penalty in the objective scaled by a multiplier $\lambda$ – are common in the literature [4–8]. This results in a simpler unconstrained problem:

$$(\mathbf{v}_\lambda, \mathbf{q}_\lambda) = \arg\min_{\mathbf{v}\subseteq\mathcal{V},\mathbf{q}} D(\mathbf{v}, \mathbf{q}) + \lambda\, R(\mathbf{v}, \mathbf{q}) \tag{2}$$

which in general is easier to solve for a fixed multiplier $\lambda$ [4–8]. However, the Lagrangian relaxed problem (2) is inherently not the same as the original rate-constrained problem (1); the difference in distortion between their respective optimal solutions is called a *duality gap* (see Appendix B and [4]).

To minimize this gap, it is imperative to find the "optimal" multiplier $\lambda^*$ — one that results in a distortion-minimizing solution $(\mathbf{v}_{\lambda^*}, \mathbf{q}_{\lambda^*})$ among all Lagrangian solutions $(\mathbf{v}_\lambda, \mathbf{q}_\lambda)$ to (2) for different $\lambda$ that satisfy

* Corresponding author.
*E-mail addresses:* deabreua@scss.tcd.ie (A. De Abreu), cheung@nii.ac.jp (G. Cheung), pascal.frossard@epfl.ch (P. Frossard), fp@lx.it.pt (F. Pereira).

https://doi.org/10.1016/j.image.2018.02.003
Received 4 June 2017; Received in revised form 11 December 2017; Accepted 4 February 2018
Available online 17 February 2018

$R(\mathbf{v}_\lambda, \mathbf{q}_\lambda) \leq B$. However, given empirical discrete rate and distortion functions $R(\mathbf{v}, \mathbf{q})$ and $D(\mathbf{v}, \mathbf{q})$, the search for this optimal multiplier numerically without resorting to continuous rate and distortion models [9,10] remains an open problem in the general setting.

*To address this problem, we are the first in the literature to construct a computation-efficient search strategy to find this optimal multiplier numerically in the general dependent coding scenario.* Specifically, we first formulate a general rate allocation problem, where each data unit can be dependently coded at different QPs using a previous coded unit as predictor, or left uncoded at the encoder for interpolation at the decoder using neighboring coded units as reference. After converting the original rate-constrained problem to the unconstrained Lagrangian counterpart (2), we design an efficient dynamic programming (DP) algorithm that finds the optimal solution to (2) for a fixed $\lambda$.

To find the optimal multiplier, we iteratively compute neighboring *singular multiplier values* [4] from the computed DP solution; each singular value $\lambda$ results in multiple simultaneously optimal solutions with different rates, *i.e.* $R(\mathbf{v}_\lambda^{(1)}, \mathbf{q}_\lambda^{(1)}) \neq R(\mathbf{v}_\lambda^{(2)}, \mathbf{q}_\lambda^{(2)})$. We show that singular values alone lead to all Lagrangian solutions to (2). When we obtain solutions $(\mathbf{v}_{\lambda^*}^{(1)}, \mathbf{q}_{\lambda^*}^{(1)})$ and $(\mathbf{v}_{\lambda^*}^{(2)}, \mathbf{q}_{\lambda^*}^{(2)})$ corresponding to singular value $\lambda^*$ with rates $R(\mathbf{v}_{\lambda^*}^{(1)}, \mathbf{q}_{\lambda^*}^{(1)}) \leq B \leq R(\mathbf{v}_{\lambda^*}^{(2)}, \mathbf{q}_{\lambda^*}^{(2)})$, we prove that $(\mathbf{v}_{\lambda^*}^{(1)}, \mathbf{q}_{\lambda^*}^{(1)})$ is the distortion-minimizing Lagrangian solution, and declare $\lambda^*$ as the optimal multiplier. To the best of our knowledge, no previously proposed Lagrangian multiplier searches [4–8] provide this theoretical claim in our general setting.[1]

Experimental results illustrate the good performance of our proposed rate allocation algorithm with optimal Lagrange multiplier selection when data units are independently or predictively coded in both monoview and multiview video sequence compression. Specifically, we show that our algorithm is able to find the closest rate values given a fixed bit budget constraint, driven by the optimal selection of Lagrangian multipliers. Moreover, we show that our bit allocation strategy is able to compete with complex rate control (RC) solutions adopted in the reference softwares of current monoview and multiview video standards, namely HEVC [11] and 3D-HEVC [12], that do not skip frames in Y-PSNR. These illustrative results show that our novel and generic algorithm for optimal selection of Lagrange multiplier value can bring large benefits in complex rate allocation problems.

The paper is organized as follows. We first review related work in Section 2 and formulate our dependent rate allocation problem in Section 3. We then describe a DP algorithm that solves the rate-constrained problem optimally but in exponential time in Section 4. To reduce complexity, we convert the problem to the Lagrangian relaxed version and propose a corresponding polynomial-time DP algorithm for a fixed multiplier. In Section 5, we discuss an efficient search methodology to identify the optimal multiplier based on the computation of neighboring singular multiplier values. Finally, we present experimental results and conclusion in Sections 6 and 7, respectively.

## 2. Related work

### 2.1. Continuous RD function modeling

In order to obtain rate and distortion functions for different data units, one can simply apply different QPs to encode each data unit and observe the resulting rate and distortion values. Then, these empirical RD points are fitted into mathematical functions for each particular sequence to derive an RD model [13,14]. However, in these types of models some parameters have to be estimated for each given video sequence and therefore they cannot be easily generalized.

Alternatively, it is possible to theoretically derive the different parameters of the RD model under simplifying assumptions. Several RD models using well understood exponential functions have been proposed [9,10]. In [9], both a Laplacian and a Generalized Gaussian (GG) distribution have been considered in their RD model for a wavelet video coding. In [10], the Bernoulli Generalized Gaussian (BGG) model have been adopted for both rate and distortion functions. The work in [15] proposed a gradient based R-lambda (GRL) model for intra frame rate control and a new bit allocation scheme for the coding tree unit (CTU) rate control. Specifically, bit per pixel (BPP), gradient per pixel (GPP) and $\lambda$ are modeled using a hyperbolic function. The authors then proposed a bit allocation scheme at three levels: GOP, frame and CTU. The authors in [16] proposed a pixel-wise unified rate-quantization (URQ) model working at the multi-level regardless of block sizes, where texture complexity is estimated using a linear MAD-based measurement. For the R–Q model, the authors assume that the ratio of distortion over bits is constant, while the parameter $\lambda$ used to trade off distortion and rate is chosen using an empirical formula. The work in [17] proposed a rate control scheme for HEVC based on new rate models (Laplacian distributions) for texture and non-texture bits, taking into account different statistical characteristics at different depths of coding units (CU). Because there is no differential coding among the CU, coding parameters for each CU are selected independently. Bits for each frame are allocated ignoring inter-frame dependency (*i.e.*, a poorly coded frame $F_i$ can negatively affect the coding performance of the following frame $F_{i+1}$ that uses $F_i$ as predictor during motion prediction.) In general, these approaches suffer from: (i) modeling errors due to idealized model inaccuracy; and (ii) continuous approximation error since the problem (selecting QPs from a finite set) is inherently discrete. In contrast, we take an empirical approach and solve the inherent discrete problem of selecting data units and QPs for coding directly, and thus do not suffer from modeling errors.

### 2.2. Constrained formulation via dynamic programming

Addressing directly the discrete rate-constrained bit allocation problem, one common approach is Dynamic Programming (DP). For example, assuming independently coded data units, [18] constructed a tree to represent all possible solutions: a node at a stage $i$ of the tree represents a particular selection of QPs $\{q_1, \ldots, q_i\}$ for data unit 1 to $i$. If two different nodes at the same stage $i$ have the same accumulated rate from unit 1 to $i$, then the one with the larger accumulated distortion would be pruned. As we will show in Section 3, the complexity of this type of DP algorithms is *pseudo-polynomial* or exponential time. If predictive coding is assumed, complexity is even higher.

To jointly select predictor frames and QoS levels for encoding and protection of different video frames during network streaming, [19] proposed an integer rounding approach to reduce complexity of a DP algorithm, where DP tables used to store computed local solutions were scaled down to reduce the number of table entries. The authors derived a performance bound for the proposed reduced-complexity DP algorithm; however, this bound gets progressively worse as the scale factor increases.

For multiview, assuming independently coded units, [20] considered a uniform rate allocation among views in a multiview video system, and proposed a DP-based algorithm to select the views for encoding and transmission such that the expected distortion among encoded and synthesized views is minimized given a rate budget. This work is extended in [21] where both the views and the coding rates for each selected view are selected to minimize distortion in a rate-constrained scenario. Due to high complexity, a greedy DP algorithm was proposed. There is no performance guarantee for the greedy DP algorithm, however, and thus the obtained solution can be arbitrarily far from the optimal solution.

### 2.3. Discrete Lagrangian formulation

Instead of the originally posed rate-constrained bit allocation problems, the Lagrangian approach is often used. [5] proposed a trellis-based algorithm to find the optimal Lagrangian solution for predictively coded

[1] [4] was a seminal bit allocation work and proposed the first singular value search strategy but only for independently coded data units.

frames in traditional monoview video. A suitable Lagrange multiplier was found by sweeping from 0 to ∞; it is not clear how this can be done efficiently, and what is the termination condition when a sufficiently good multiplier value is found. [22] also adopted a Lagrangian approach, where the value of the Lagrange multiplier $\lambda$ is empirically modeled as a function of the rate and the distortion when data units are predictively encoded. There is no guarantee that this ad-hoc approach will result in the *optimal* multiplier value upon termination, however. Recently, the authors in [23] proposed to estimate the Lagrange multiplier value $\lambda$ by a stochastic subgradient method, where the $\lambda$ value was fine-tuned until it approached optimality. Then, QP values are estimated from $\lambda$ using a log-linear model. Different from traditional rate-control algorithms, and similarly to our work, the authors assume that $\lambda$ should be constant in a resource allocation problem and not selected after setting the QP values. However, the authors solution is proposed for independent bit allocation problems and not for predictively encoded data units. Moreover, the authors have a stochastic formulation based on probabilities, while we are solving a deterministic discrete problem that does not rely on probabilistic assumptions.

To allocate bits among independent quantizers, [4] defined the notion of singular multiplier values – multipliers with multiple simultaneously optimal Lagrangian solutions – and proposed to iteratively search through neighboring singular multiplier values until a terminating condition (resulting rates of Lagrangian solutions being very close to bit budget) is met. Extending [4,24] addressed a similar rate allocation problem with two bit budget constraints, also using the notion of singular values, in order to achieve an optimal distribution of source and channel bits among wavelet subbands for transmission of scalable video over noisy channels. Our algorithm adopts the singular multiplier value concept [4] and extends it to the case where data units can be predictively coded or left uncoded entirely for subsequent interpolation at the decoder.

### 2.4. Multiview rate allocation

Rate allocation problems have also been investigated recently in multiview video coding applications. [7] extended the trellis optimization approach in [5] to multiview video with predictive coding. The authors considered a system where views can be skipped at the encoder and eventually synthesized using both texture and depth maps at the decoder, and optimized only QPs of the texture maps. In a different framework, [8] tackled the bit allocation problem for both texture and depth data such that the distortion of the camera views and a set of synthetic views, reconstructed at the decoder, is minimized. The authors optimized both the set of coded views and their QPs and adopted a trellis-based solution for an effective search of the optimal coding solution. Both [7] and [8] employed the Lagrangian approach, but no mathematically rigorous strategy was proposed to search for a suitable multiplier value. We will show that our proposal can be applied to multiview coding scenarios also.

## 3. Rate allocation framework

In a classical rate allocation problem, the objective is to minimize the total distortion of a set of data units, each of which may be independently or predictively coded, subject to a rate budget constraint. We first describe the general coding system under consideration. Then, we formulate a rate-constrained bit allocation problem for predictively coded data units as a discrete optimization problem. We present examples that show how our formulation can be applicable in different practical scenarios. Finally, we prove that the formulated problem is NP-hard.

### 3.1. System model

We consider a general coding scenario where we seek to allocate, per unit of time, a total bit budget $B$ to an ordered set of $V$ data units, $\mathcal{V} = \{1, 2, \dots, V\}$. Examples include consecutive frames in monoview video, or neighboring views in a multiview image sequence. We define $\mathbf{v} = \{v_1, v_2, \dots, v_N\}$, where $\mathbf{v} \subseteq \mathcal{V}$, as the subset of $N$ units selected for coding, where $N \leq V$. We assume that a unit $v_n$ can be left uncoded at the encoder, and later interpolated at the decoder using the two surrounding coded units $v_L$ and $v_R$, where $v_L < v < v_R$ and $v_L, v_R \in \mathbf{v}$. As the boundary units cannot be interpolated at the decoder in the same manner, they are always selected for coding, *i.e.*, $v_1 = 1$ and $v_N = V$ are always coded.

Each unit $v_n \in \mathbf{v}$ is coded using a QP $q_{v_n} \in \mathcal{Q}$, where $\mathcal{Q}$ is a discrete set of possible QPs for a given encoder. Denote by $\mathbf{q}$ the set of chosen QPs for the units in $\mathbf{v}$. Assuming that predictive coding is used to code neighboring units, unit $v_n$ coded with QP $q_{v_n}$ using as predictor unit $v_{n-1}$ coded with QP $q_{v_{n-1}}$ has a rate $r_{v_n}(v_{n-1}, q_{v_{n-1}}, q_{v_n})$ and a distortion $\Delta_{v_n}(v_{n-1}, q_{v_{n-1}}, q_{v_n})$. Note that if $v_{n-1}$ and $v_n$ are not consecutive data units in $\mathcal{V}$, then the uncoded intermediate units between coded $v_{n-1}$ and $v_n$ need to be interpolated at the decoder and their quality must be included in the distortion computation. Hence, the distortion term $\Delta_{v_n}(v_{n-1}, q_{v_{n-1}}, q_{v_n})$ accounts for the distortion of all interpolated units $v_i$ in the range $(v_{n-1}, v_n)$ as well as for the distortion of the coded unit $v_n$. Mathematically, $\Delta_{v_n}(v_{n-1}, q_{v_{n-1}}, q_{v_n})$ can be written as:

$$\Delta_{v_n}(v_{n-1}, q_{v_{n-1}}, q_{v_n}) = \sum_{\substack{v_i \in \mathbf{v_i} \\ v_{n-1} < v_i \leq v_n \\ q_{v_{n-1}}, q_{v_n} \in \mathcal{Q}}} d_{v_i}(v_{n-1}, v_n, q_{v_{n-1}}, q_{v_n}) \tag{3}$$

where $d_{v_i}(\cdot)$ is the distortion of the interpolated data unit $v_i$, $v_i \in \mathcal{V}$, using reference units and QPs $(v_{n-1}, q_{v_{n-1}})$ and $(v_n, q_{v_n})$. If $v_{n-1}$ and $v_n$ are consecutive data units, then $v_i = v_n$ and $d_{v_i}(\cdot)$ corresponds to the distortion of coded unit $v_n$ using $v_{n-1}$ for prediction.

The first unit $v_1$ in $\mathcal{V}$ is independently encoded and its distortion depends only on its own QP. For this particular case, Eq. (3) can be re-written as:

$$\Delta_{v_1}(q_{v_1}) = d_{v_1}(q_{v_1}). \tag{4}$$

More general definitions for predictive coding is also possible [5], where the rate and distortion functions depend on the QPs of *all* previous coded data units. However, for complexity reasons, we assume that rate $r_{v_n}$ and distortion $\Delta_{v_n}$ depend only on QP $q_{v_{n-1}}$ of previous unit $v_{n-1}$ used for prediction. This is a good approximation in practical predictive coding, as shown in [8].

### 3.2. Problem formulation

With the above definitions, our objective is to find the optimal subset of data units $\mathbf{v}^* = \{v_1, v_2, \dots, v_N\} \subseteq \mathcal{V}$ along with their corresponding QPs $\mathbf{q}^* = \{q_{v_1}, q_{v_2}, \dots, q_{v_N}\}$ such that the aggregate distortion at the decoder is minimized, subject to a bit budget constraint $B$. Note that $N$ is not a fixed parameter, but rather a variable that is computed when optimizing $\mathbf{v}$. The optimization problem can be defined as follows:

$$\begin{aligned} (\mathbf{v}^*, \mathbf{q}^*) = \arg \min_{\substack{\mathbf{v} \subseteq \mathcal{V} \\ q_{v_n} \in \mathcal{Q}}} \quad & \Delta_{v_1}(q_{v_1}) + \sum_{n=2}^{N} \Delta_{v_n}(v_{n-1}, q_{v_{n-1}}, q_{v_n}) \\ \text{s.t.} \quad & r_{v_1}(q_{v_1}) + \sum_{n=2}^{|\mathbf{v}|} r_{v_n}(v_{n-1}, q_{v_{n-1}}, q_{v_n}) \leq B \end{aligned} \tag{5}$$

where $\Delta_{v_1}(q_{v_1})$ and $r_{v_1}(q_{v_1})$ are the distortion and rate for the first selected unit $v_1$, and $\Delta_{v_n}(\cdot)$ and $r_{v_n}(\cdot)$ are the distortion and rate for a predictively coded unit $v_n$, as described above.

### 3.3. Applications

Our formulation in (5) is sufficiently general for application to different monoview and multiview video rate allocation scenarios. We list a few illustrative examples below.

- *Scenario I: QP selection for independent coded images*. If frames in a monoview video or views in a multiview image sequence are independently coded for maximum random access [4], then our formulation (5) is applicable to optimal selection of QPs, where the rate and distortion of each data unit (image) do not depend on the previous one, *i.e.*, $r_{v_n}(v_{n-1}, q_{v_{n-1}}, q_{v_n}) = r_{v_n}(q_{v_n})$ and $\Delta_{v_n}(v_{n-1}, q_{v_{n-1}}, q_{v_n}) = d_{v_n}(q_{v_n})$.
- *Scenario II: QP selection for differentially coded images*. If each frame in monoview video or view in a multiview image sequence is differentially coded using a previous predictor frame, then (5) can be used, where all units in $\mathcal{V}$ are chosen for coding, for optimal selection of QPs [5,8]. Note that (5) is not applicable to bi-directional prediction like B-frames.
- *Scenario III: Selection of images for coding*. If a consistent quality requirement dictates that all images should be coded at the same pre-defined QP [20], (5) can be used to select a subset of images in monoview video or views in a multiview image sequence to minimize aggregate distortion that includes interpolated images at the decoder.
- *Scenario IV: Mode selection in a group of blocks*. Instead of images, data units can represent code blocks in an image. Given a fixed QP, (5) can be used to select the optimal coding modes for a group of block (GoB) for a given rate constraint, where $\mathcal{Q}$ now represents possible modes a block can take on [25,26].

### 3.4. NP-hardness proof

We prove that our formulated bit allocation problem (5) is NP-hard via reduction from a well-known NP-hard problem — *Knapsack* (KS) [27]. The binary decision version of KS, which is NP-complete, can be described as follows:

*Binary Decision Problem of KS* — Given a set of $M$ items, each with non-negative weight $w_m$ and profit $c_m$, and a knapsack of capacity $W$, does there exist a subset of items with total weight $\leq W$, such that the total profit is at least $\bar{C}$?

To prove NP-hardness of (5), we consider a more specific problem where each unit $v$, if chosen for coding, can only be independently coded at QP $q$. We reduce the binary decision version of this simplified problem from the KS decision problem as follows. First, we construct two boundary units $v_0$ and $v_{M+1}$ and $M$ intermediate data units corresponding to $M$ items in KS. The distortion of not coding any intermediate unit is $D$ no matter what surrounding units are used as reference for interpolation. Coding the two boundary units at QP $q$ results in rate 2 and distortion 0. Coding an intermediate unit $v$ at QP $q$ results in rate $w_v$ and distortion $D-c_v$. The binary decision problem is: does there exist a subset $\mathbf{v}$ of units selected for coding (each at QP $q$) such that the distortion is no larger than $MD - \bar{C}$, given a rate budget $W + 2$? If the answer is yes, then the chosen subset $\mathbf{v}$ of units in the solution, excluding the two boundary units, has a corresponding subset of items in KS with total weight no larger than $W$ and total profit at least $\bar{C}$. Hence the problem is no easier than the KS decision problem, and thus is also NP-complete. Therefore the optimization version of the problem is NP-hard. Since the specific problem is already NP-hard, the more general problem (5) is no easier, and hence is also NP-hard.

## 4. DP algorithm for Lagrangian problem

We first present an algorithm based on DP that returns an optimal solution to the constrained problem in (5). We then show that the algorithm complexity is exponential. Next, we present an alternative DP algorithm that solves the corresponding Lagrangian relaxed problem in polynomial time for a fixed Lagrangian multiplier.

### 4.1. Constrained DP algorithm

To solve the problem in (5) optimally, we derive a DP algorithm that recursively divides the original problem into smaller sub-problems. When a sub-problem is solved, its solution is stored inside an entry in a DP table, so that subsequent calls to the same sub-problem can simply look up the solved solution in the table [28].

Denote by $\boldsymbol{\Phi}_{v_n}(q_{v_n}, \bar{B})$ the minimum distortion sum for data units from $v_n + 1$ to $V$, given that $v_n$ is coded with QP $q_{v_n}$, and there is an available bit budget of $\bar{B}$, $\bar{B} \leq B$, to code the remaining units $v_{n+1}, \ldots, v_N$. This distortion sum $\boldsymbol{\Phi}_{v_n}(q_{v_n}, \bar{B})$ can be recursively written as:

$$\boldsymbol{\Phi}_{v_n}(q_{v_n}, \bar{B}) = \min_{\substack{v_{n+1}\in\mathcal{V} \mid v_{n+1}>v_n \\ q_{v_{n+1}}\in\mathcal{Q}}} \Delta_{v_{n+1}}(v_n, q_{v_n}, q_{v_{n+1}}) + \mathbf{1}(v_{n+1} < V)\, \boldsymbol{\Phi}_{v_{n+1}}(q_{v_{n+1}}, \bar{B} - r_{v_{n+1}}(v_n, q_{v_n}, q_{v_{n+1}})) \quad (6)$$

where $\boldsymbol{\Phi}_{v_n}(\cdot)$ and $\boldsymbol{\Phi}_{v_{n+1}}(\cdot)$ are the sub-problems of the original problem defined in (5), whose solutions are stored in the entries $(v_n, q_{v_n}, \bar{B})$ and $(v_{n+1}, q_{v_{n+1}}, \bar{B})$ of the DP table, respectively. The indicator function $\mathbf{1}(c)$ returns 1 if the clause $c$ is true, and 0 otherwise.

In words, (6) selects the next unit $v_{n+1}$ to code at QP $q_{v_{n+1}}$, resulting in distortion $\Delta_{v_{n+1}}(v_n, q_{v_n}, q_{v_{n+1}})$ for data units from $v_n$ to $v_{n+1}$ inclusively. This selection reduces the bit budget to $\bar{B} - r_{v_{n+1}}(v_n, q_{v_n}, q_{v_{n+1}})$ for coding the remaining data units. If $v_{n+1} = V$, meaning it is the right boundary unit in $\mathcal{V}$, then the recursive term in (6) is not necessary.

Since the first unit $v_1 = 1$ is always selected for coding, the computation in (6) is solved via the initial call:

$$\min_{q_1\in\mathcal{Q}} \Delta_1(q_1) + \boldsymbol{\Phi}_1(q_1, B - r_1(q_1)). \quad (7)$$

The complexity of this DP algorithm is $O(V^2Q^2B)$, which is bounded by the size of the DP table, $V \times Q \times B$, multiplied by the complexity of computing each table entry, $O(VQ)$. This is polynomial in $B$. However, $B$ is encoded in $\log_2(B)$ bits as input to the algorithm, thus the algorithm is exponential in the size of the input. This complexity is also called *pseudo-polynomial time* in the complexity literature [29]. Therefore, in order to reduce the algorithm complexity, the rate budget dimension $B$ of the DP table should be eliminated.

### 4.2. Lagrangian DP algorithm

Towards the goal of reducing the complexity of the constrained DP algorithm in (6) by eliminating the bit budget or rate dimension $B$ from the DP table, we consider a Lagrangian relaxation of our constrained problem in (5). In the Lagrangian relaxation we move the rate consideration from the constraint to the objective function, resulting in the rate–distortion (RD) formulation:

$$(\mathbf{v}^*, \mathbf{q}^*_{\mathbf{v}}) = \arg \min_{\substack{\mathbf{v}\subseteq\mathcal{V} \\ q_{v_n}\in\mathcal{Q}}} \Delta_{v_1}(q_{v_1}) + \sum_{n=2}^{|\mathbf{v}|} \Delta_{v_n}(v_{n-1}, q_{v_{n-1}}, q_{v_n}) + \lambda \left( r_{v_1}(q_{v_1}) + \sum_{n=2}^{|\mathbf{v}|} r_{v_n}(v_{n-1}, q_{v_{n-1}}, q_{v_n}) \right) \quad (8)$$

where the multiplier $\lambda > 0$ is a parameter that weighs the importance of rate against distortion.

To solve (8) for a given $\lambda$, we follow a similar procedure. We first denote $\boldsymbol{\Phi}_{v_n}(q_{v_n})$ as the minimum RD cost for data units from $v_n + 1$ to $V$ inclusively, given that $v_n$ is coded with QP $q_{v_n}$. Then, $\boldsymbol{\Phi}_{v_n}(q_{v_n})$ can be recursively defined as:

$$\boldsymbol{\Phi}_{v_n}(q_{v_n}) = \min_{\substack{v_{n+1}\in\mathcal{V}^o \mid v_{n+1}>v_n \\ q_{v_{n+1}}\in\mathcal{Q}}} \Delta_{v_{n+1}}(v_n, q_{v_n}, q_{v_{n+1}}) + \lambda\, r_{v_{n+1}}(v_n, q_{v_n}, q_{v_{n+1}}) + \mathbf{1}(v_{n+1} < V)\, \boldsymbol{\Phi}_{v_{n+1}}(q_{v_{n+1}}) \quad (9)$$

where sub-problems solutions of $\boldsymbol{\Phi}_{v_n}(\cdot)$ and $\boldsymbol{\Phi}_{v_{n+1}}(\cdot)$ are now stored in the entries $(v_n, q_{v_n})$ and $(v_{n+1}, q_{v_{n+1}})$ of the DP table, respectively. Then, similarly to the analysis performed for the constrained DP algorithm, for a given $\lambda$, the algorithm (9) has complexity $O(V^2Q^2)$, which does not depends on the rate budget $B$. It has therefore a polynomial time complexity.

We now discuss the relationship between the constrained problem in (5), solvable via (6), and its Lagrangian relaxed version in (8), solvable via (9). Denote by $(\mathbf{v}_\lambda, \mathbf{q}_\lambda)$ an optimal solution of (8) for a given $\lambda$, with resulting distortion and rate $D(\mathbf{v}_\lambda, \mathbf{q}_\lambda)$ and $R(\mathbf{v}_\lambda, \mathbf{q}_\lambda)$, respectively. One can show that, if there exists a multiplier $\lambda^*$ such that $R(\mathbf{v}_{\lambda^*}, \mathbf{q}_{\lambda^*}) = B$, then solution $(\mathbf{v}_{\lambda^*}, \mathbf{q}_{\lambda^*})$ is also an optimal solution of (5). The proof is given in Appendix A for the sake of completeness.

Note that, since $R(\mathbf{v}_\lambda, \mathbf{q}_\lambda)$ is discrete, there may not exist a multiplier $\lambda$ such that $R(\mathbf{v}_\lambda, \mathbf{q}_\lambda) = B$. In this case, we can pick a value $\lambda = \lambda_1$ with a corresponding Lagrangian solution $(\mathbf{v}_{\lambda_1}, \mathbf{q}_{\lambda_1})$, $R(\mathbf{v}_{\lambda_1}, \mathbf{q}_{\lambda_1}) < B$, as an approximate solution to (5) with the following performance bound. Given two solutions of (8) $(\mathbf{v}_{\lambda_1}, \mathbf{q}_{\lambda_1})$ and $(\mathbf{v}_{\lambda_2}, \mathbf{q}_{\lambda_2})$, using respective multipliers $\lambda_1$ and $\lambda_2$, with resulting rates $R(\mathbf{v}_{\lambda_1}, \mathbf{q}_{\lambda_1}) < B < R(\mathbf{v}_{\lambda_2}, \mathbf{q}_{\lambda_2})$, the difference in distortion between Lagrangian solution $(\mathbf{v}_{\lambda_1}, \mathbf{q}_{\lambda_1})$ and the *true* optimal solution $(\mathbf{v}^*, \mathbf{q}^*)$ of (5) is bounded as:

$$|D(\mathbf{v}_{\lambda_1}, \mathbf{q}_{\lambda_1}) - D(\mathbf{v}^*, \mathbf{q}^*)| \le |D(\mathbf{v}_{\lambda_1}, \mathbf{q}_{\lambda_1}) - D(\mathbf{v}_{\lambda_2}, \mathbf{q}_{\lambda_2})|. \tag{10}$$

The proof is given in Appendix B. Clearly, the bound in (10) is tightest when the difference in distortion between the two Lagrangian solutions is the smallest.

*We propose an efficient algorithm to find Lagrange multipliers such that the resulting Lagrangian optimal solutions yield the tightest bound possible with respect to the original constrained problem.*

## 5. Search for the optimal Lagrange multiplier

We propose a methodology to identify the "optimal" Lagrange multiplier value via an iterative search. By "optimal", we mean a multiplier value that yields a pair of Lagrangian optimal solutions to (8) with the tightest distortion bound (10) possible with respect to the true optimal solution in (5). We first review the notion of a *singular value* of the Lagrange multiplier, introduced in the context of rate allocation problems in [4]. We then discuss our methodology in the following two subsections.

### 5.1. Singular values of Lagrange multiplier

We first observe that because the rate $R(\mathbf{v}_\lambda, \mathbf{q}_\lambda)$ is discrete, there are different $\lambda$ values at which the optimal solutions of (8) are not unique; these are called *singular values* of Lagrange multiplier [4]. As an example, in Fig. 1 the rate values $R(\mathbf{v}_\lambda, \mathbf{q}_\lambda)$ of the optimal solutions $(\mathbf{v}_\lambda, \mathbf{q}_\lambda)$ to (8) are plotted against the multiplier $\lambda$. It can be seen that at singular values of $\lambda = \{\lambda_0, \lambda_1, \lambda^*, \lambda_2\}$, there are at least two simultaneous optimal solutions to (8), resulting in two different rates, denoted by the superindex $l$ (lower) and $u$ (upper). Singular multiplier values have two important properties:

1. Two neighboring singular values share one common optimal solution to (8).
2. Multipliers $\lambda$ between two neighboring singular values produce the *same* optimal solution as the shared solution of the two singular values.

These two properties are discussed extensively in [4]. As an example, in Fig. 1 neighboring singular values $\lambda^*$ and $\lambda_1$ share an optimal solution $(\mathbf{v}^l_{\lambda^*}, \mathbf{q}^l_{\lambda^*}) = (\mathbf{v}^u_{\lambda_1}, \mathbf{q}^u_{\lambda_1})$ to (8), and multipliers $\lambda$ between these two singular values produce the same optimal solution. These two properties imply the following important corollary: *singular values alone produce all solutions to (8) as $\lambda$ varies from 0 to $\infty$. Thus, it is sufficient to examine only Lagrangian solutions of singular values in order to find the best multiplier value.*

Moreover, it is known [4] that the rate $R(\mathbf{v}_\lambda, \mathbf{q}_\lambda)$ is a non-increasing function with respect to $\lambda$ as shown in Fig. 1.

Suppose now that a singular value $\lambda^*$ has corresponding optimal Lagrangian solutions $(\mathbf{v}^l_{\lambda^*}, \mathbf{q}^l_{\lambda^*})$ and $(\mathbf{v}^u_{\lambda^*}, \mathbf{q}^u_{\lambda^*})$ where $R(\mathbf{v}^l_{\lambda^*}, \mathbf{q}^l_{\lambda^*}) \le B \le R(\mathbf{v}^u_{\lambda^*}, \mathbf{q}^u_{\lambda^*})$. That means that no other singular value will yield a Lagrangian solution with rate either smaller or larger than this pair of solutions, due to the monotonicity of the rate $R(\mathbf{v}_\lambda, \mathbf{q}_\lambda)$. Thus this pair of solutions $(\mathbf{v}^l_{\lambda^*}, \mathbf{q}^l_{\lambda^*})$ and $(\mathbf{v}^u_{\lambda^*}, \mathbf{q}^u_{\lambda^*})$ are the Lagrangian solutions that produce the tightest distortion bound (10) possible, and $\lambda^*$ is the optimal Lagrange multiplier value.

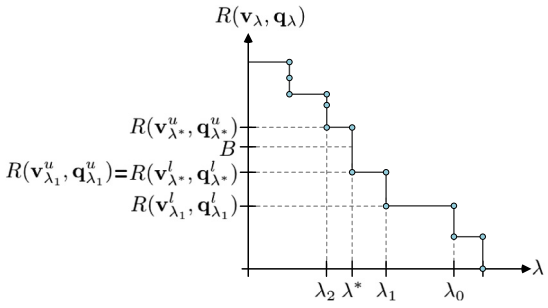



**Fig. 1.** Rate $R(\mathbf{v}_\lambda, \mathbf{q}_\lambda)$ of optimal solution $(\mathbf{v}_\lambda, \mathbf{q}_\lambda)$ to (8) as function of multiplier $\lambda$. Singular values are $\lambda_2, \lambda^*, \lambda_1, \lambda_0$, etc. The singular value $\lambda^*$ produces a pair of Lagrangian solutions $(\mathbf{v}^l_{\lambda^*}, \mathbf{q}^l_{\lambda^*})$ and $(\mathbf{v}^u_{\lambda^*}, \mathbf{q}^u_{\lambda^*})$ with the tightest distortion bound (10) with respect to the optimal solution in (5) with rate constraint $B$.

Importantly, monotonicity also means that in the iterative search for the optimal singular value $\lambda^*$, one only needs to increase/decrease the current $\lambda$ value by examining the rate of the corresponding optimal solution: decrease $\lambda$ if $R(\mathbf{v}_\lambda, \mathbf{q}_\lambda) < B$ and increase it otherwise.

### 5.2. Procedure to compute neighboring singular values

One strategy to search for the optimal singular value is to march through neighboring singular values in the direction of the rate budget $B$ until the pair of optimal solutions to (8) corresponding to the singular value $\lambda^*$, $(\mathbf{v}^l_{\lambda^*}, \mathbf{q}^l_{\lambda^*})$ and $(\mathbf{v}^u_{\lambda^*}, \mathbf{q}^u_{\lambda^*})$, have rates satisfying $R(\mathbf{v}^l_{\lambda^*}, \mathbf{q}^l_{\lambda^*}) \le B \le R(\mathbf{v}^u_{\lambda^*}, \mathbf{q}^u_{\lambda^*})$. For example, in Fig. 1, after checking $\lambda_0$ and $\lambda_1$ successively, one arrives at the optimal singular value $\lambda^*$. Thus the challenge is how to compute a neighboring singular Lagrangian value in the direction of $B$. We accomplish this by storing auxiliary information as the DP algorithm (9) is computed for a fixed multiplier $\lambda$, in order to identify a neighboring optimal solution to (8) if $\lambda$ is increased/decreased appropriately.

Specifically, we compute a neighboring singular multiplier value within the same DP framework (9) developed to solve (8) as follows. Denote by $(v^*_{n+1}, q^*_{v_{n+1}})$ the argument that minimizes the sub-problem $\Phi_{v_n}(q_{v_n})$ in (9) for a given $\lambda$. Further, denote by $\Psi_{v_n}(q_{v_n})$ and $\Upsilon_{v_n}(q_{v_n})$ the distortion and rate of the optimal solution $(v^*_{n+1}, q^*_{v_{n+1}})$ to sub-problem $\Phi_{v_n}(q_{v_n})$ respectively. Then, $\Psi_{v_n}(q_{v_n})$ and $\Upsilon_{v_n}(q_{v_n})$ are computed as:

$$\Psi_{v_n}(q_{v_n}) = \Delta_{v^*_{n+1}}(v_n, q_{v_n}, q_{v^*_{n+1}}) + \Psi_{v^*_{n+1}}(q_{v^*_{n+1}}) \tag{11}$$

$$\Upsilon_{v_n}(q_{v_n}) = r_{v^*_{n+1}}(v_n, q_{v_n}, q_{v^*_{n+1}}) + \Upsilon_{v^*_{n+1}}(q_{v^*_{n+1}}). \tag{12}$$

These variables are stored in separate DP tables as (9) is being solved recursively.

#### 5.2.1. Computing a smaller singular value

Suppose first that $R(\mathbf{v}_\lambda, \mathbf{q}_\lambda) < B$; thus, we need to decrease $\lambda$ in order to increase $R(\mathbf{v}_\lambda, \mathbf{q}_\lambda)$, according to the monotonicity property. To find the neighboring *smaller* singular value $\lambda^-$, where $\lambda^- < \lambda$, we know that $\lambda^-$ and $\lambda$ share an optimal solution $(\mathbf{v}_\lambda, \mathbf{q}_\lambda)$, and that $\lambda^-$ has an additional solution with rate *larger* than $R(\mathbf{v}_\lambda, \mathbf{q}_\lambda)$. This additional globally optimal solution $(\mathbf{v}_{\lambda^-}, \mathbf{q}_{\lambda^-})$ must include a new local solution of a sub-problem $\Phi_{v_n}(q_{v_n})$ as $\lambda$ decreases. Thus, we seek the closest multiplier $\lambda^-$ to $\lambda$, where there exists a sub-problem $\Phi_{v_n}(q_{v_n})$ with a new local solution $(v^-_{n+1}, q^-_{v_{n+1}})$ whose rate $r_{v^-_{n+1}}(v_n, q_{v_n}, q^-_{v_{n+1}})$ is larger than the previous solution optimal solution $r_{v^*_{n+1}}(v_n, q_{v_n}, q_{v^*_{n+1}})$, given $\lambda$.

In particular, for each sub-problem $\Phi_{v_n}(q_{v_n})$ we compute a *singular value candidate* $\lambda^-_{v_n}(q_{v_n})$, where $\lambda^-_{v_n}(q_{v_n}) < \lambda$, as:

$$\lambda^-_{v_n}(q_{v_n}) = \max_{\substack{v \in \mathcal{V} \\ v > v_n \\ q_v \in \mathcal{Q}}} \frac{\Psi_{v_n}(q_{v_n}) - (\Delta_v(v_n, q_{v_n}, q_v) + \Psi_v(q_v))}{(r_v(v_n, q_{v_n}, q_v) + \Upsilon_v(q_v)) - \Upsilon_{v_n}(q_{v_n})} \tag{13}$$

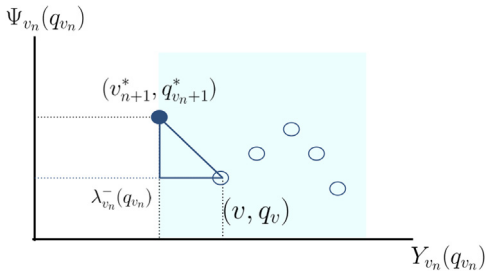



**Fig. 2.** Illustration of the search of singular value candidate $\lambda^-_{v_n}(q_{v_n})$ for the $\Phi_{v_n}(q_{v_n})$ sub-problem. The search is done over the points $(v, q_v)$ with larger rates than current optimal solution $(v^*_{n+1}, q^*_{v_{n+1}})$ (blue region).

where the search for the maximization is performed over the set of units and QPs, $(v, q_v) \in (\mathcal{V}, \mathcal{Q})$, with a *larger* resulting rate:

$$r_v(v_n, q_{v_n}, q_v) + Y_v(q_v) > Y_{v_n}(q_{v_n}). \tag{14}$$

In other words, $\lambda^-_{v_n}(q_{v_n})$ is the closest multiplier value smaller than $\lambda$ where the sub-problem $\Phi_{v_n}(q_{v_n})$ results in a different solution with a larger rate. Geometrically, (13) is computing an RD point on the convex hull to the right of $(v^*_{n+1}, q^*_{v_{n+1}})$ with a larger rate. See Fig. 2.

The emergence of a new globally optimal solution $(\mathbf{v}_{\lambda^-}, \mathbf{q}_{\lambda^-})$ can stem from any sub-problem $\Phi_{v_n}(q_{v_n})$ as $\lambda$ decreases. To identify the first sub-problem $\Phi_{v_n}(q_{v_n})$ that results in a new local solution, we compute $\lambda^-$ as the *largest* (closest to $\lambda$) of all singular value candidates $\lambda^-_{v_n}(q_{v_n})$:

$$\lambda^- = \max_{v_n \in \mathcal{V}, q_{v_n} \in \mathcal{Q}} \lambda^-_{v_n}(q_{v_n}). \tag{15}$$

Denote by $(v^-_n, q^-_{v_n})$ the argument that maximizes (15). Further, denote by $(v^-_{n+1}, q^-_{v_{n+1}})$ the argument that maximizes (13) for sub-problem $\Phi_{v^-_n}(q^-_{v_n})$. We show that given $\lambda^-$, the sub-problem $\Phi_{v^-_n}(q^-_{v_n})$ has two simultaneously optimal solutions.

**Lemma 1.** *Using singular value $\lambda^-$, sub-problem $\Phi_{v^-_n}(q^-_{v_n})$ has two solutions, $(v^*_{n+1}, q^*_{v_{n+1}})$ and $(v^-_{n+1}, q^-_{v_{n+1}})$, that are simultaneously optimal.*

**Proof.** First, because $\lambda^-$ is the largest among all singular value candidates, when the multiplier is $\lambda^-$, the sub-problems other than $\Phi_{v^-_n}(q^-_{v_n})$ still have the same optimal solutions as before when the multiplier was $\lambda$. This means that except for $(v^-_n, q^-_{v_n})$, distortions $\Psi_{v_n}(q_{v_n})$ and rates $Y_{v_n}(q_{v_n})$ for all the sub-problems remain the same when the multiplier is $\lambda^-$. Consider now the sub-problem $\Phi_{v^-_n}(q^-_{v_n})$. For candidates $(v, q_v)$ with rates smaller than $(v^*_{n+1}, q^*_{v_{n+1}})$, optimal sub-problem solution when the multiplier was $\lambda$, means that the RD cost of $(v^*_{n+1}, q^*_{v_{n+1}})$ remains smaller than these candidates $(v, q_v)$ when multiplier is now smaller. One can then show that the definition $\lambda^-_{v_n}(q_{v_n})$ in (13) implies that RD costs of $(v^*_{n+1}, q^*_{v_{n+1}})$ and $(v^-_{n+1}, q^-_{v_{n+1}})$ are the same. Finally, because $(v^*_{n+1}, q^*_{v_{n+1}})$ and $(v^-_{n+1}, q^-_{v_{n+1}})$ are neighboring convex-hull points, candidates $(v, q_v)$ different from $(v^-_{n+1}, q^-_{v_{n+1}})$ with rates larger than $(v^*_{n+1}, q^*_{v_{n+1}})$, must result in a larger RD cost than $(v^*_{n+1}, q^*_{v_{n+1}})$ and $(v^-_{n+1}, q^-_{v_{n+1}})$ when multiplier is $\lambda^-$. □

Importantly, we note that the computed $\lambda^-$ in (15) *only guarantees* that a new local solution has emerged from a sub-problem $\Phi_{v_n}(q_{v_n})$. However, the globally optimal solution will not change *if* the changed sub-problem $\Phi_{v_n}(q_{v_n})$ is not part of the global solution. Nonetheless, successive moves to $\lambda^-$ will eventually trigger a change in the global solution, resulting in a new rate $R(\mathbf{v}_\lambda, \mathbf{q}_\lambda)$.

#### 5.2.2. Computing a larger singular value

Similarly, we can compute the neighboring singular value larger than $\lambda$. The singular value candidate $\lambda^+_{v_n}(q_{v_n})$ larger than $\lambda$ for each sub-problem $\Phi_{v_n}(q_{v_n})$ is computed as:

$$\lambda^+_{v_n}(q_{v_n}) = \min_{\substack{v \in \mathcal{V} \\ v > v_n \\ q_v \in \mathcal{Q}}} \frac{\Big(\Delta_v(v_n, q_{v_n}, q_v) + \Psi_v(q_v)\Big) - \Psi_{v_n}(q_{v_n})}{Y_{v_n}(q_{v_n}) - (r_v(v_n, q_{v_n}, q_v) + Y_v(q_v))} \tag{16}$$

where the search for the minimization is performed over the set of units and QPs, $(v, q_v) \in (\mathcal{V}, \mathcal{Q})$, with a *smaller* resulting rate:

$$Y_{v_n}(q_{v_n}) > r_v(v_n, q_{v_n}, q_v) + Y_v(q_v). \tag{17}$$

Then, the singular value $\lambda^+$ is the *smallest* (closest to $\lambda$) of all singular value candidates $\lambda^+_{v_n}(q_{v_n})$:

$$\lambda^+ = \min_{v_n \in \mathcal{V}, q_{v_n} \in \mathcal{Q}} \lambda^+_{v_n}(q_{v_n}). \tag{18}$$

#### 5.2.3. DP table update

From (15) and (18), we know that, given an initial $\lambda$ value, the neighboring smaller ($\lambda^-$) or larger singular value ($\lambda^+$) can be found. The decision of which direction we should march to find the next singular value is determined by the rate of the computed solution $R(\mathbf{v}_\lambda, \mathbf{q}_\lambda)$. Then, we use (9) to update entries in the DP table given new multiplier $\lambda^-$ or $\lambda^+$. Note that only a subset of the DP table entries need to be updated. Specifically, given a new singular value $\lambda^-$ (15) or $\lambda^+$ (18) associated with sub-problem $\Phi_{v_n}(q_{v_n})$, only the DP entries corresponding to $(v, q)$ (computed with (13) or (16)), require updates using (9).

Since in (9) only the candidates $(v_{n+1}, q_{v_{n+1}})$ are considered given the sub-problem $\Phi_{v_n}(q_{v_n})$, when there is a new singular value $\lambda^-$ or $\lambda^+$ associated with sub-problem $\Phi_{v^-_n}(q^-_{v_n})$ or $\Phi_{v^+_n}(q^+_{v_n})$, only the DP entries $(v, q)$ found through , require updates using (9).

#### 5.2.4. Complexity analysis

Computing (13) or (16) for each sub-problem has complexity $O(VQ)$. There are $V \times Q$ sub-problems, and hence computing $\lambda^-$ or $\lambda^+$ has complexity $\mathcal{O}(V^2Q^2)$. Denote by $m$ the number of iterations until the optimal singular multiplier value is found. Thus the multiplier search complexity is $\mathcal{O}(mV^2Q^2)$.

The number of iterations depends on how far from the optimal multiplier is the initial $\lambda$ value. To reduce the number of iterations, we propose a hybrid coarse-/fine-grained multiplier search strategy. First, we perform a classical binary search on the positive real line, as done in [4], to produce big changes in $\lambda$ to approach the optimal multiplier. When binary search fails to yield new solutions, we apply our fine-grained singular-value search until the optimal multiplier value is found. The search strategy for the best multiplier $\lambda^*$ is summarized in Algorithm 1.

## 6. Experimental results

### 6.1. Experimental setup

We now demonstrate the performance of our proposed rate allocation algorithm (9) with optimal Lagrange multiplier selection in monoview and multiview video coding problems. Each data unit represents a full frame in a monoview video or a view in a multiview video.

We consider the monoview video datasets *HallMonitor* (352 × 288, 30 fps) [30,31], *Kimono* (1920 × 1080, 24 fps), provided by Nakajima Laboratory of the Tokyo Institute of Technology, *ChinaSpeed* (1024 × 768, 30 fps) and *RaceHorses* (832 × 480, 30 fps). These sequences have a GOP size of 1 s, namely 30 frames or 24 frames. Our algorithm is used to select frames and corresponding QPs for coding in each GOP. To encode these sequences, we used the reference software HM 15.0 [32] of the High Efficiency Video Coding (HEVC) standard [33], after disabling the rate control option and replacing it with our solution.

**Algorithm 1** Search of the optimal Lagrange multiplier

Initialize $\lambda$
*Coarse-grained search:* Perform a binary search of $\lambda$ until no new solutions can be reached.
*Fine-grained search:*
Solve (8) via (9) with unique solution $R(\mathbf{v}_\lambda, \mathbf{q}_\lambda)$.
**if** $R(\mathbf{v}_\lambda, \mathbf{q}_\lambda) < B$ **then**
    Find singular value $\lambda^-$ via (15), with $\lambda^- < \lambda$.
    $\lambda \leftarrow \lambda^-$
**else**
    Find singular value $\lambda^+$ via (18), with $\lambda^+ > \lambda$.
    $\lambda \leftarrow \lambda^+$
**end if**
**repeat**
    Update the DP entries $(v_j, q_{v_j})$ that needs to be modified, $v_j < v$, when $v$ is associated to $\lambda$.
    Find simultaneous solutions $(\mathbf{v}^l_\lambda, \mathbf{q}^l_\lambda)$ and $(\mathbf{v}^u_\lambda, \mathbf{q}^u_\lambda)$, where $R(\mathbf{v}^l_\lambda, \mathbf{q}^l_\lambda) < R(\mathbf{v}^u_\lambda, \mathbf{q}^u_\lambda)$.
    **if** $R(\mathbf{v}^u_\lambda, \mathbf{q}^u_\lambda) < \bar{B}$ **then**
        Find singular value $\lambda^-$ via (15), with $\lambda^- < \lambda$.
        $\lambda \leftarrow \lambda^-$
    **else if** $\bar{B} < R(\mathbf{v}^l_\lambda, \mathbf{q}^l_\lambda)$ **then**
        Find singular value $\lambda^+$ via (18), with $\lambda^+ > \lambda$.
        $\lambda \leftarrow \lambda^+$
    **end if**
**until** $R(\mathbf{v}^l_\lambda, \mathbf{q}^l_\lambda) \le \bar{B} \le R(\mathbf{v}^u_\lambda, \mathbf{q}^u_\lambda)$
$(\mathbf{v}^l_\lambda, \mathbf{q}^l_\lambda)$ is the best approximate Lagrangian solution.

For the multiview video datasets, we consider the sequences: *Shark* (1920 × 1088, 30 fps, 9 views), provided by NICT for MPEG FTV standardization [34], *UndoDancer* (1920 × 1088, 25 fps, 5 views) [35], *SoccerLinear2* (1600 × 1200, 60 fps, 7 views) [36] and *PoznanHall2* (1920 × 1088, 25 fps, 3 views). To encode these sequences we used the reference software HTM 13.0 [37] of the 3D extension of HEVC (3D-HEVC) [38] for multiview video, replacing the available rate control with our solution. We used a GOP size of 8 frames and an intra-period of 24 frames as defined under the common test conditions by JCT-3V [39]. Our algorithm is used to select the views to be encoded and their corresponding QPs; hierarchical B-frames [33] are used in the temporal dimension to exploit temporal redundancy as done in the standard. A cascading quantization parameter (CQP) strategy [40] is used to assign the QPs to the frames in the GOPs, where a $\Delta$QP is added to the QP value of the anchor frame to generate the QP values of the various frames in the GOP. We consider a $\Delta$QP vector equal to $\{0, 1, 2, 3, 4, 4, 3, 4\}$ for the successive frames in the GOP, as suggested in the reference software of 3D-HEVC [37].

Since each view consists of a color and a depth image pair, we use our algorithm to select image pairs for coding and QPs for the color images only, while QPs for the depth images are fixed at 30, so that 3D geometry information is coded accurately for high-quality virtual view synthesis. While QPs for color and depth images can be jointly selected for optimal RD performance as done in [8] for static multiview image sequences, in practice depth images represent only a small fraction of the total bitrate (about 10%), and we optimize only color image QPs. We note that the extension of our optimization to include selection of multiple QPs for a given data unit is possible and is left for future work.

In our experiments, we compute PSNR of the luminance component (Y-PSNR) for all the decoder-side data units that are decoded or interpolated if they are left uncoded at the encoder. In particular, to reconstruct uncoded frames in monoview video sequences we use a popular temporal up-sampling method based on motion estimation [41,42]. To construct missing views in multiview video sequences, we use a simple depth-image based rendering (DIBR) [43,44] method at the decoder where the color pixels from the closest right and left coded views are projected to the missing intermediate view given per-pixel disparity information provided by the corresponding depth images.

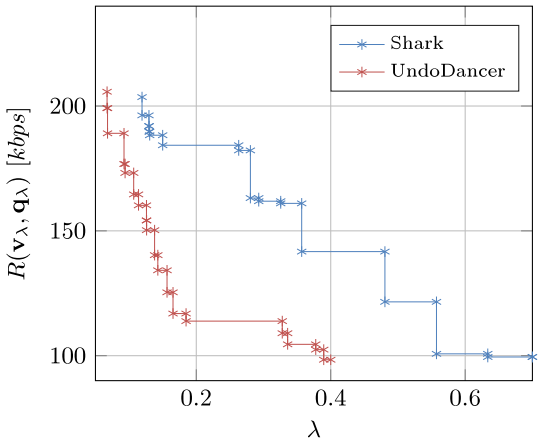



**Fig. 3.** Relationship between the rate $R(\mathbf{v}_\lambda, \mathbf{q}_\lambda)$ and the Lagrange multiplier $\lambda$ for the *Shark* and *UndoDancer* multiview video sequences when views are independently encoded.

In the following, the performance of our algorithm is illustrated for monoview and multiview video sequences in two scenarios: (i) when data units are independently coded and (ii) when data units are predictively coded. Then, we use current rate control (RC) solutions implemented in state-of-the-art encoders in order to illustrate the importance of a proper choice of Lagrangian multipliers.

### 6.2. Independent coding

We first evaluate the performance of our algorithm for the case of independently encoded units for monoview and multiview video sequences. The available set of QPs for the coding units are $Q = \{25, 26, \ldots, 51\}$ for both cases.

To examine the behavior of our algorithm, we show in Fig. 3 the relationship between the rate $R(\mathbf{v}_\lambda, \mathbf{q}_\lambda)$ of optimal Lagrangian solutions and Lagrange multiplier $\lambda$ for the multiview sequences *Shark* and *UndoDancer*. In particular, we illustrate the iterative multiplier search process to identify, among all Lagrangian solutions (8) for any $\lambda$, one that minimizes the aggregate distortion subject to a rate budget, which in this case is $B = 200$ kbps. Multiplier $\lambda$ is initialized to be $\lambda = 0.7$ and $\lambda = 0.4$ for the two sequences *Shark* and *UndoDancer* respectively. The optimal solution $(\mathbf{v}_\lambda, \mathbf{q}_\lambda)$ is reached in these two cases at singular value $\lambda = 0.1193$ for *Shark* $\left(R(\mathbf{v}_\lambda, \mathbf{q}_\lambda) = 196.26\right)$ and at singular value $\lambda = 0.0673$ for *UndoDancer* $\left(R(\mathbf{v}_\lambda, \mathbf{q}_\lambda) = 199.15\right)$. Note that the number of iterations $m$ determines the complexity of our algorithm, as explained in Section 5.2.4. Using (10), the distortion bound for each sequence can be computed using the two simultaneously optimal solutions at the optimal singular value, which is 0.01 dB and 0.03 dB for *Shark* and *UndoDancer*, respectively.

*This shows in practice that the plots for rate $R(\mathbf{v}_\lambda, \mathbf{q}_\lambda)$ versus multiplier $\lambda$ are typically dense with samples (i.e., RD plots are in general convex), and the resulting distortion bounds are tight.*

Fig. 4 shows the performance of our rate allocation algorithm in terms of average Y-PSNR given different rate budget constraint values $B$ for the monoview video sequences *HallMonitor* and *Kimono*. It is important to note that our algorithm always outputs a solution with a rate that is under the rate budget $B$.

Similarly, Fig. 5 shows the performance of our algorithm in terms of achieved rate and average Y-PSNR, by setting different rate budgets for the multiview video sequences *Shark*, *UndoDancer* and *SoccerLinear2*. It can be seen how our algorithm manages to closely reach (always from below) the imposed rate constraints. In this case, our algorithm solutions do not skip any view at the encoder.

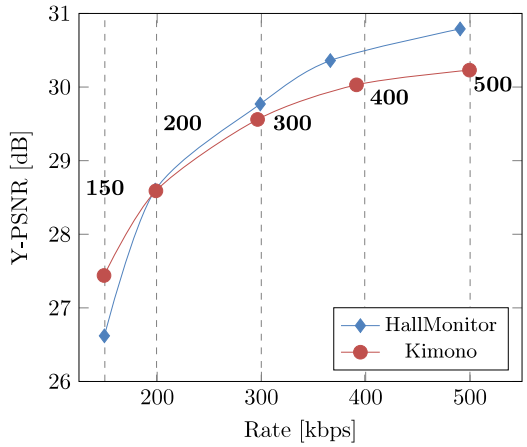



**Fig. 4.** Actual rate $R$ and average Y-PSNR value for the proposed algorithm when a rate budget $B$ is considered, $B = [150, 200, 300, 400, 500]$ kbps. Monoview video sequences: *HallMonitor* and *Kimono* with independently encoded frames.

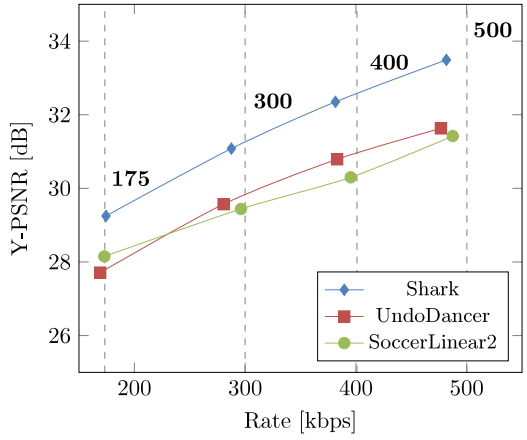



**Fig. 5.** Actual rate $R$ and average Y-PSNR value for the proposed algorithm when a rate budget $B$ is considered, $B = [175, 300, 400, 500]$ kbps. Multiview video sequences: *Shark*, *UndoDancer* and *SoccerLinear2* with independently coded views.

### 6.3. Predictive coding

We consider now the predictive coding case, and in particular, an $IPP\ldots$ predictive coding structure with one intra (I) coded data unit and subsequent predictively (P) coded units in each GOP. As the computational complexity in the rate allocation increases for predictive coding compared to the independent coding case – rate and distortion terms now depend on previous coded units – we decrease the granularity of the available QPs in the search space of our algorithm to $\mathcal{Q} = \{25, 28, 31 \cdots, 51\}$.

We observe that our assumption that the rate and distortion functions depend only on one reference frame (see Section 3.1) sometimes leads to small under-estimation of the coding rate. Thus, we employ a simple post-processing step, where the Lagrangian solution to (9) of a singular multiplier value $\lambda$ closest to the budget $B$ with actual aggregate coding rate below the budget is used. For different rate constraints $B$, Fig. 6 shows the RD performance of our algorithm for the *HallMonitor*, *Kimono*, *ChinaSpeed* and *RaceHorses* monoview sequences.

By including the proposed post-processing step, we see from Fig. 6 that our algorithm provides an encoding solution that fulfills a given a rate budget constraint. Moreover, our algorithm uses a sparse set of QPs, meaning that our results can be improved if the QP sampling increases. Note that, different rate budget constraint values were considered for

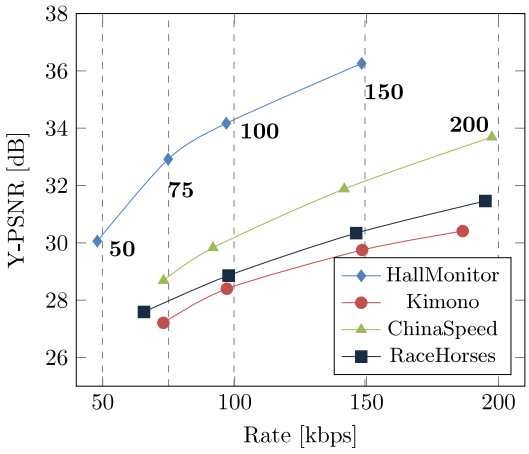



**Fig. 6.** Actual rate $R$ and average Y-PSNR value for the proposed algorithm when a rate budget $B$ is considered. Monoview video sequences: *Kimono*, *ChinaSpeed* and *RaceHorses* with $B = [75, 100, 150, 200]$ kbps and *Hall Monitor* with $B = [50, 75, 100, 150]$ kbps. Frames are predictively coded.

different sequences. This is due to the fact that sequence characteristics are different, where less complex and/or lower resolution sequences were assigned lower rate budget constraint values $B$.

Finally, Fig. 7 shows the performance in terms of rate and average quality of our algorithm for the predictive coding of the *Shark*, *UndoDancer*, *SoccerLinear2* and *PoznanHall2* multiview video sequences. The solution obtained with our algorithm satisfies the rate budget most of the time with our original algorithm, and there is usually no need to use the post-processing step to modify the obtained solutions. This is due to the length of the prediction paths. In the case of multiview video, the maximum length of the (inter-view) prediction path is 9 views (*e.g.*, *Shark*), compared to 30 and 24 frames (GOP size) in the *HallMonitor*, *Kimono*, *ChinaSpeed* and *RaceHorses* monoview video sequences. This means that, for the multiview video case, the effect of previously coded units in a current predicted unit is much more limited than in monoview video cases, thus making our assumption of Section 3.1 more reasonable.[2]

### 6.4. Rate control (RC) schemes for HEVC and 3D-HEVC

To better appreciate the benefits of the optimal choice of Lagrangian multipliers in our basic rate control solution, we now illustrate the performance of more complex rate control (RC) schemes [11,12] adopted by the reference software HM 15.0 of the HEVC standard for monoview video sequences, and by the reference software HTM 13.0 of the 3D-HEVC standard for multiview video. These solutions are more complex in the sense that the rate control is done at multiple levels: frame/view level, coding tree unit (CTU) level and block level; while our scheme only optimizes at the frame/view level. Here, we only aim to illustrate that the correct selection of multipliers at the frame/view level *already* leads to noticeable coding gain. Combining our proposed algorithm with finer-grained level RC algorithms is left for future work. In order to obtain more meaningful comparisons, we fix the QP value of the depth images when RC schemes of the reference software for 3D-HEVC is evaluated, so that only QPs of the color images of the multiview video sequences are optimized for our algorithm.

[2] The solution of *PoznanHall2* sequence encoded with $B = 75$ kbps, represents the only case where a data unit (a view, in this case) was skipped during the encoding process in the predictive coding case, for both monoview and multiview video sequences. This explains the drop in quality for the presented curve. In particular, for $B = 75$ kbps, the solution of our algorithm can be either (67.86 kbps, 28.13 dB) or (82.54 kbps, 33.64 dB) when no view is skipped.

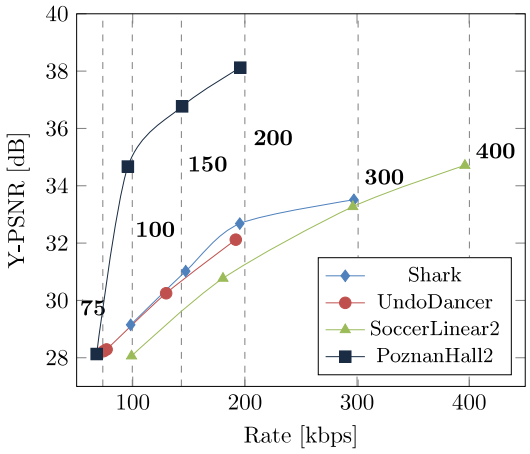



**Fig. 7.** Actual rate $R$ and average Y-PSNR value for the proposed algorithm when a rate budget $B$ is considered. Multiview video sequences and $B$ values: *UndoDancer* and *PoznanHall2* with $B = [75, 100, 150, 200]$ kbps, *Shark* with $B = [100, 150, 200, 300]$ kbps and *SoccerLinear2* with $B = [100, 200, 300, 400]$ kbps. Views are predictively coded.

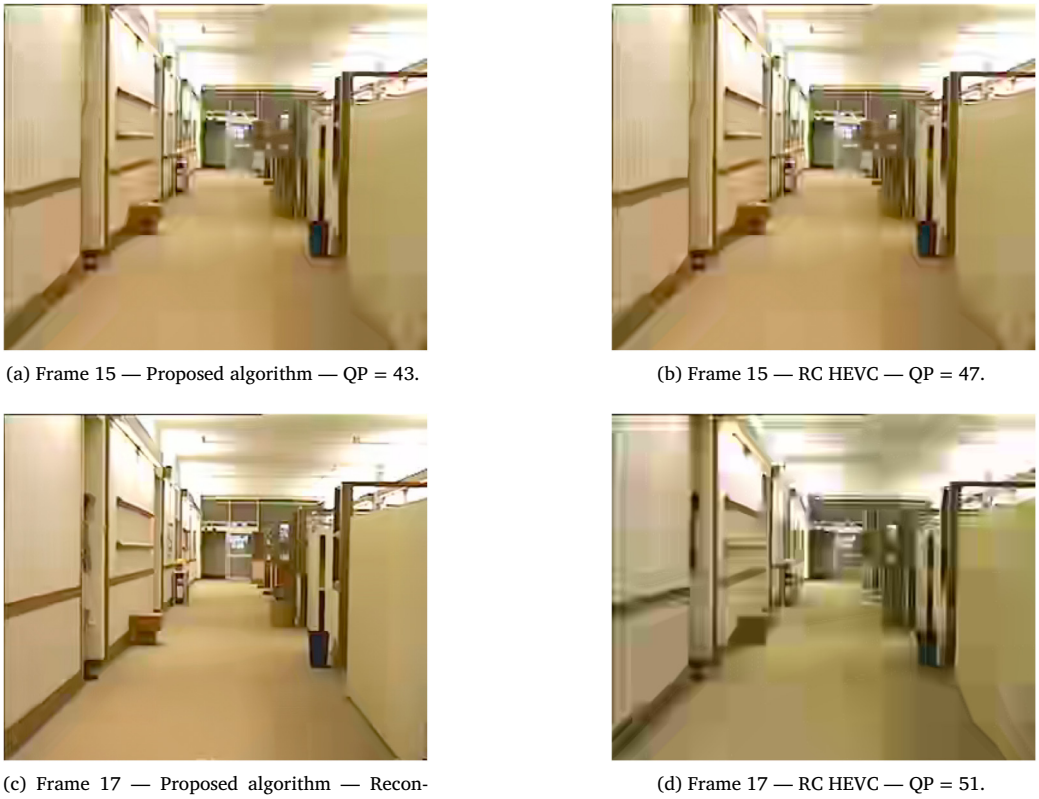
#### 6.4.1. Independent coding

Fig. 9 shows the performance of both our rate allocation algorithm and the RC of HEVC, in terms of average Y-PSNR given a rate budget constraint $B$ for the monoview video sequences *HallMonitor* and *Kimono*. We see that our algorithm always outputs a solution with a rate that is under the rate budget $B$ and achieves a higher quality, with a Y-PSNR gain of up to 2.34 dB. The corresponding visual quality is illustrated in Fig. 8 for *HallMonitor* for frames 15 and 17, when the rate budget is $B = 150$ kbps. Our algorithm tends to skip frames with low motion, as frame 17, which are then efficiently interpolated at the decoder. This permits to achieve a higher visual quality than the one of the RC of HEVC that has to use a higher QP value to satisfy the rate budget, as frames are not skipped.

Similarly, Fig. 10 shows the performance of our algorithm and the RC of 3D-HEVC in terms of average Y-PSNR, with different rate budgets for the multiview sequences *Shark* and *SoccerLinear2*. Although our algorithm has a good performance compare with the RC of 3D-HEVC, this gain is smaller than for the monoview sequences. The main reason is that the frame-to-frame differences along the temporal dimension in monoview videos are relatively smaller, making skipping frames a more attractive option. Our algorithm solution in this case does not skip any view at the encoder, and still it achieves a higher average Y-PSNR compared to the RC of 3D-HEVC. Proving that, the good performance of our algorithm does not reside on its capability of skipping data units at the encoder. Moreover, this result shows the impact of a proper selection of a Lagrangian multiplier value.

#### 6.4.2. Predictive coding

As in Section 6.3 the QP set granularity of our proposed solution is decreased while the RC schemes adopted by HM 15.0 and HTM 13.0 optimize QPs for the different frames with a finer granularity. In Figs. 11 and 12, the Rate–Distortion performance of our algorithm is presented next to the RC solutions of HEVC and 3D-HEVC encoders, respectively. The results are shown for *HallMonitor*, *RaceHorses* and *Kimono* monoview video sequences (Fig. 11) and for *Shark*, *PoznanHall2* and *SoccerLinear2* multiview video sequences (Fig. 12). In general, our



(a) Frame 15 — Proposed algorithm — QP = 43.

(b) Frame 15 — RC HEVC — QP = 47.

(c) Frame 17 — Proposed algorithm — Reconstructed frame.

(d) Frame 17 — RC HEVC — QP = 51.

**Fig. 8.** Visual quality illustration for the *HallMonitor* monoview video sequence with independently encoded frames when the proposed algorithm and the RC of HEVC are used ($B = 150$ kbps). (a) and (b) Show frame 15 encoded according to our proposed algorithm and the RC of HEVC, respectively. (c) Shows frame 17, that has been skipped at the encoder and reconstructed at the decoder according to the proposed algorithm, achieving a higher visual quality compared to the RC of HEVC output in (d).

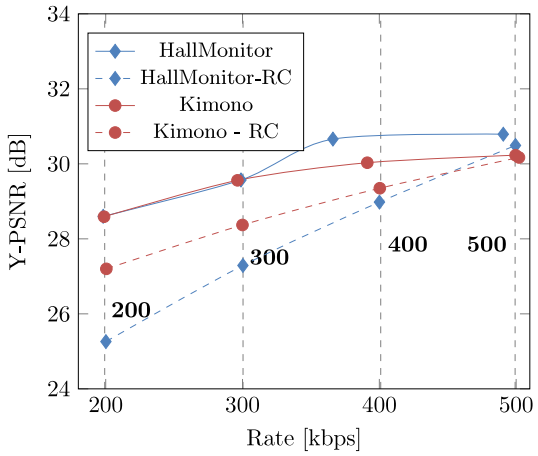



**Fig. 9.** Actual rate $R$ and average Y-PSNR value for the proposed algorithm and for the RC of HEVC, for the *Hall Monitor* and *Kimono* monoview video sequences with independently coded frames. Rate budget: $B = [200, 300, 400, 500]$.

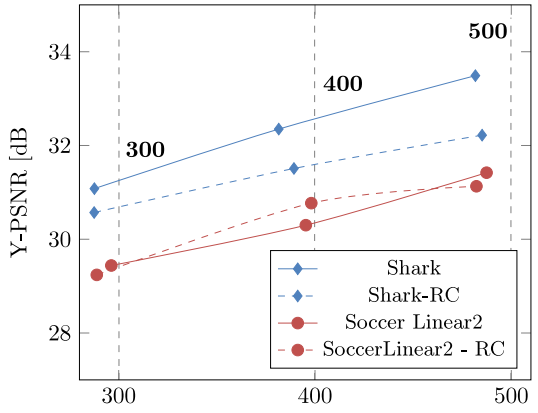



**Fig. 10.** Actual rate $R$ and average Y-PSNR value for the proposed algorithm and for the RC of HEVC, for the *Shark* and *SoccerLinear2* multiview video sequences with independently coded views. Rate budget: $B = [300, 400, 500]$.

algorithm manages to provide solutions with rates below the rate constraint, while the RC solution do not always lead to a good performance, particularly at low rate budgets. This is evident for the *Kimono* dataset where the rates of some solutions of the RC of HEVC are far above the rate budget constraint. In particular, for $B = 75$ kbps and $B = 100$ kbps, the RC solutions are 96.44 kbps and 120.00 kbps, while our algorithm solutions are 73.01 kbps and 97.06 kbps, respectively. A similar behavior can be seen for the *Shark* and the *PoznanHall2* multiview datasets with $B = 100$ kbps and $B = 75$ kbps, respectively, where the RC solution of 3D-HEVC is slightly over the rate constraint.

Although our algorithm shows good performance on average, the gain of our algorithm in terms of Y-PSNR is smaller for predictive coding than for independent coding. This is due to the fact that in predictive coding skipped data units have higher impact in the overall quality, which is one of the reasons of quality gains for independently encoded units. Indeed, when units are skipped at the encoder during predictive coding, the distance between a coded unit and its reference increases, which reduces the coding performance. Thus, these results show that the good performance of our algorithm does not uniquely depend on its capability for skipping data units at the encoder.

In general, from these results we can conclude that when our algorithm is close to the rate budget (*i.e.*, the granularity of the available QPs is not affecting the solution) it achieves a quality that is generally higher than the one in the RC of 3D-HEVC. These results show the

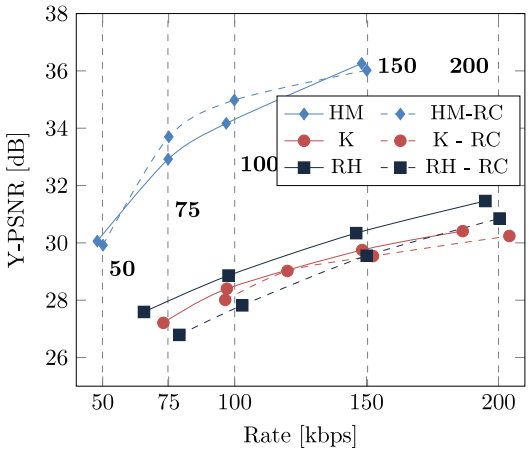



**Fig. 11.** Actual rate $R$ and average Y-PSNR value for the proposed algorithm and for the RC of HEVC, for the *HallMonitor* (HM), *Kimono* (K) and *RaceHorses* (RH) monoview video sequences with predictively coded frames. Rate budget: $B = [50, 75, 100, 150]$ for *HallMonitor* and $B = [75, 100, 150, 200]$ for *Kimono* and *RaceHorses*.

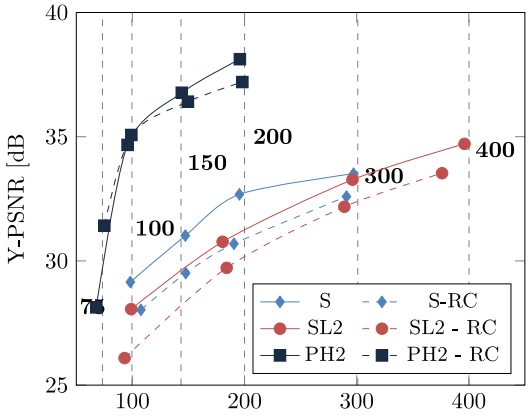



**Fig. 12.** Actual rate $R$ and average Y-PSNR value for the proposed algorithm and for the RC of HEVC, for the *Shark* (S), *SoccerLinear2* (SL2) and *PoznanHall2* (PH2) multiview video sequences with predictively coded views. Rate budget: $B = [100, 150, 200, 300]$ for *Shark*, $B = [100, 200, 300, 400]$ for *SoccerLinear2* and $B = [75, 100, 150, 200]$ kbps for *PoznanHall2*.

importance of optimizing the Lagrange multiplier value in a rate control problem. It is important to note that we are not proposing a competitor to current RC solutions. In fact, current RC solutions and our algorithm are complementary solutions, and one could imagine merging both frameworks for better performance.

## 7. Conclusion

Traditional rate allocation problems in video coding are first converted to unconstrained formulations via Lagrangian relaxation, which are typically easier to solve using optimization techniques like dynamic programming. However, the search for the "optimal" Lagrange multiplier – one that results in a distortion-minimizing Lagrangian solution among those that satisfy the original rate constraint – remains elusive. In this paper, we are the first in the literature to construct an iterative search strategy to identify this optimal multiplier for the general dependent coding scenario. The strategy has been integrated into a general Lagrangian-based dynamic programming algorithm to efficiently solve rate allocation problems in video communication applications. The potential of our new algorithm has been illustrated in representative compression problems with monoview and multiview

video sequences, and rate control solutions adopted by the reference softwares, HEVC and 3D-HEVC, have been used to appreciate the rate–distortion performance of our proposed algorithm. It is important to note that we are not proposing a competitor to current RC solutions, but rather an optimization algorithm that finds the optimal Lagrange multiplier value.

## Appendix A. Proof of optimality

We prove here that, if an optimal solution $(\mathbf{v}_{\lambda^*}, \mathbf{q}_{\lambda^*})$ to the unconstrained Lagrangian problem corresponding to multiplier value $\lambda^*$ satisfies the rate constraint exactly, *i.e.*,

$$R(\mathbf{v}_{\lambda^*}, \mathbf{q}_{\lambda^*}) = B, \tag{A.1}$$

then $(\mathbf{v}_{\lambda^*}, \mathbf{q}_{\lambda^*})$ is also the optimal solution to the original constrained problem. The optimality of $(\mathbf{v}_{\lambda^*}, \mathbf{q}_{\lambda^*})$ implies that:

$$D(\mathbf{v}_{\lambda^*}, \mathbf{q}_{\lambda^*}) + \lambda^* R(\mathbf{v}_{\lambda^*}, \mathbf{q}_{\lambda^*}) \leq D(\mathbf{v}, \mathbf{q}) + \lambda^* R(\mathbf{v}, \mathbf{q}), \quad \forall \mathbf{v}, \mathbf{q}.$$

Rearranging the terms, we get:

$$\begin{aligned} \lambda^* \left[R(\mathbf{v}_{\lambda^*}, \mathbf{q}_{\lambda^*}) - R(\mathbf{v}, \mathbf{q})\right] &\leq D(\mathbf{v}, \mathbf{q}) - D(\mathbf{v}_{\lambda^*}, \mathbf{q}_{\lambda^*}) \\ \lambda^* \left[B - R(\mathbf{v}, \mathbf{q})\right] &\leq D(\mathbf{v}, \mathbf{q}) - D(\mathbf{v}_{\lambda^*}, \mathbf{q}_{\lambda^*}). \end{aligned}$$

Now we restrict our solution space to a subspace $S$ where $R(\mathbf{v}, \mathbf{q}) \leq B$. Then,

$$\begin{aligned} 0 \leq \lambda^* \left[B - R(\mathbf{v}, \mathbf{q})\right] \leq D(\mathbf{v}, \mathbf{q}) - D(\mathbf{v}_{\lambda^*}, \mathbf{q}_{\lambda^*}), \quad \forall (\mathbf{v}, \mathbf{q}) \in S \\ D(\mathbf{v}_{\lambda^*}, \mathbf{q}_{\lambda^*}) \leq D(\mathbf{v}, \mathbf{q}), \quad \forall (\mathbf{v}, \mathbf{q}) \in S. \end{aligned}$$

We can thus conclude $(\mathbf{v}_{\lambda^*}, \mathbf{q}_{\lambda^*})$ is an optimal solution to the original constrained problem as well. □

## Appendix B. Performance bound

We prove here the performance bound given in Eq. (10). Let $(\mathbf{v}_{\lambda_1}, \mathbf{q}_{\lambda_1})$ and $(\mathbf{v}_{\lambda_2}, \mathbf{q}_{\lambda_2})$ be two solutions of the problem in (8) using $\lambda_1$ and $\lambda_2$ with resulting rates:

$$R(\mathbf{v}_{\lambda_1}, \mathbf{q}_{\lambda_1}) < B < R(\mathbf{v}_{\lambda_2}, \mathbf{q}_{\lambda_2}). \tag{B.2}$$

We can derive a performance bound for the feasible solution $(\mathbf{v}_{\lambda_1}, \mathbf{q}_{\lambda_1})$ as follows. Denote by $(\mathbf{v}^*, \mathbf{q}^*)$ the optimal solution to the original constrained problem. By the optimality of the solution $(\mathbf{v}_{\lambda_2}, \mathbf{q}_{\lambda_2})$, we can write:

$$\begin{aligned} 0 \leq \lambda^* \left[R(\mathbf{v}_{\lambda_2}, \mathbf{q}_{\lambda_2}) - R(\mathbf{v}^*, \mathbf{q}^*)\right] \leq D(\mathbf{v}^*, \mathbf{q}^*) - D(\mathbf{v}_{\lambda_2}, \mathbf{q}_{\lambda_2}) \\ D(\mathbf{v}_{\lambda_2}, \mathbf{q}_{\lambda_2}) \leq D(\mathbf{v}^*, \mathbf{q}^*) \end{aligned} \tag{B.3}$$

where the second line is true because $B < R(\mathbf{v}_{\lambda_2}, \mathbf{q}_{\lambda_2})$ and $R(\mathbf{v}^*, \mathbf{q}^*) \leq B$. By the optimality of the solution $(\mathbf{v}^*, \mathbf{q}^*)$, we also know that:

$$D(\mathbf{v}^*, \mathbf{q}^*) \leq D(\mathbf{v}, \mathbf{q}), \quad \forall (\mathbf{v}, \mathbf{q}) \in S \tag{B.4}$$

where, $S$ denotes the set of solutions that have a total rate lower than $B$. Note that $S$ includes $(\mathbf{v}_{\lambda_1}, \mathbf{q}_{\lambda_1})$, since $R(\mathbf{v}_{\lambda_1}, \mathbf{q}_{\lambda_1}) < B$. Combining the inequalities in (B.3) and (B.4), we can write:

$$\begin{aligned} D(\mathbf{v}_{\lambda_2}, \mathbf{q}_{\lambda_2}) \leq D(\mathbf{v}^*, \mathbf{q}^*) \leq D(\mathbf{v}_{\lambda_1}, \mathbf{q}_{\lambda_1}) \\ \left|D(\mathbf{v}_{\lambda_1}, \mathbf{q}_{\lambda_1}) - D(\mathbf{v}^*, \mathbf{q}^*)\right| \leq \left|D(\mathbf{v}_{\lambda_1}, \mathbf{q}_{\lambda_1}) - D(\mathbf{v}_{\lambda_2}, \mathbf{q}_{\lambda_2})\right| \end{aligned} \tag{B.5}$$

which concludes the proof. □

## References


[1] A.M. Tekalp, Digital Video Processing, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1995.

[2] P.S.P. Wang, Pattern Recognition, Machine Intelligence and Biometrics, Springer Publishing Company, Incorporated, 2011.

[3] P. Kauff, N. Atzpadin, C. Fehn, M. Müller, O. Schreer, A. Smolic, R. Tanger, Depth map creation and image-based rendering for advanced 3DTV services providing interoperability and scalability, Image Commun. 22 (2) (2007) 217–234. http://dx.doi.org/10.1016/j.image.2006.11.013.

[4] Y. Shoham, A. Gersho, Efficient bit allocation for an arbitrary set of quantizers, IEEE Trans. Acoust. Speech Signal Process. 36 (9) (1988) 1445–1453.

[5] K. Ramchandran, A. Ortega, M. Vetterli, Bit allocation for dependent quantization with applications to multiresolution and MPEG video coders, IEEE Trans. Image Process. 3 (5) (1994).

[6] S. Liu, C.-C.J. Kuo, Joint temporal-spatial bit allocation for video coding with dependency, IEEE Trans. Circuits Syst. Video Technol. 15 (1) (2005) 15–26.

[7] J.-H. Kim, J. Garcia, A. Ortega, Dependent bit allocation in multiview video coding, in: IEEE International Conference on Image Processing, Genoa, Italy, 2005.

[8] G. Cheung, V. Velisavljevic, A. Ortega, On dependent bit allocation for multiview image coding with depth–image-based rendering, IEEE Trans. Image Process. 20 (11) (2011) 3179–3194.

[9] M. Wang, M. van der Schaar, Operational rate-distortion modeling for wavelet video coders, IEEE Trans. Signal Process. 54 (9) (2006) 3505–3517.

[10] M. Kaaniche, A. Fraysse, B. Pesquet-Popescu, J.-C. Pesquet, Accurate rate-distortion approximation for sparse Bernoulli-generalized Gaussian models, in: IEEE International Conference on Acoustics, Speech and Signal Processing, 2014, pp. 2020–2024.

[11] B. Li, H. Li, L. Li, J. Zhang, Rate control by R-lambda model for HEVC, in: JCTVC-K0103, Joint Collaborative Team on Video Coding, JCT-VC, Changai, China, Oct. 2012.

[12] W. Lim, H. Jo, D. Sim, JCT3V ??? Inter-view MV-based rate prediction for rate control of 3D multi-view video coding, JCT3V-F0166, Joint Collaborative Team on Video Coding (JCT-VC), Geneva, Switzerland, Oct. 2013.

[13] S. Ma, J. Si, S. Wang, A study on the rate distortion modeling for high efficiency video coding, in: IEEE International Conference on Image Processing, 2012, pp. 181–184.

[14] K. Stuhlmuller, N. Farber, M. Link, B. Girod, Analysis of video transmission over lossy channels, IEEE J. Sel. Areas Commun. 18 (6) (2000) 1012–1032.

[15] M. Wang, K. Ngan, H Li, An efficient frame-content based intra frame rate control for high efficiency video coding, IEEE Signal Process. Lett. 22 (7) (2015) 896–900.

[16] H. Choi, J. Yoo, J. Nam, D. Sim, I. Bajic, Pixel-wise unified rate-quantization model for multi-level rate control, IEEE J. Sel. Top. Signal Process. 7 (6) (2013) 1112–1123.

[17] B. Lee, M. Kim, T. Nguyen, A frame-level rate control scheme based on texture and nontexture rate models for high efficiency video coding, IEEE Trans. Circuits Syst. Video Technol. 24 (3) (2014) 465–479.

[18] A. Ortega, K. Ramchandran, Rate-distortion methods for image and video compression, IEEE Signal Process. Mag. 15 (6) (1998) 23–50.

[19] G. Cheung, W.-T. Tan, C. Chan, Reference frame optimization for multiple-path video streaming with complexity scaling, IEEE Trans. Circuits Syst. Video Technol. 17 (6) (2007) 649–662.

[20] A. De Abreu, L. Toni, T. Maugey, N. Thomos, P. Frossard, F. Pereira, Multiview video representations for quality-scalable navigation, in: IEEE Int. Conf. on Visual Communications and Image Processing, Valletta, Malta, 2014. doi:10.1109/PCS.2013.6737710.

[21] A. De Abreu, L. Toni, N. Thomos, T. Maugey, F. Pereira, P. Frossard, Optimal layered representation for adaptive interactive multiview video streaming, J. Vis. Commun. Image Represent. 33 (2015) 255–264.

[22] S. Li, C. Zhu, Y. Gao, Y. Zhou, F. Dufaux, M. Sun, Lagrangian multiplier adaptation for rate-distortion optimization with inter-frame dependency, IEEE Trans. Circuits Syst. Video Technol. 26 (1) (2016) 117–129. http://dx.doi.org/10.1109/TCSVT.2015.2450131.

[23] C.H. Kuo, Y.L. Shih, S.C. Yang, Rate control via adjustment of Lagrange multiplier for video coding, IEEE Trans. Circuits Syst. Video Technol. 26 (11) (2016) 2069–2078. http://dx.doi.org/10.1109/TCSVT.2015.2501921.

[24] G. Cheung, A. Zakhor, Bit allocation for joint source/channel coding of scalable video, IEEE Trans. Image Process. 9 (3) (2000) 340–356. http://dx.doi.org/10.1109/83.826773.

[25] G. Sullivan, T. Wiegand, Rate-distortion optimization for video compression, IEEE Signal Process. Mag. (1998).

[26] G. Cheung, Directed acyclic graph based mode optimization for H.263 video encoding, in: IEEE International Conference on Image Processing, Thessaloniki, Greece, 2001.

[27] M.R. Garey, D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, Freeman, 1999.

[28] T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, Introduction to Algorithms, third ed., The MIT Press, 2009.

[29] C.H. Papadimitriou, K. Steiglitz, Combinatorial Optimization: Algorithms and Complexity, Dover, 1998.

[30] Y.-M. Chen, I. Bajic, P. Saeedi, Coarse-to-fine moving region segmentation in compressed video, in: 10th Workshop on Image Analysis for Multimedia Interactive Services, 2009, WIAMIS '09, 2009, pp. 45–48. http:/dx.doi.org/10.1109/WIAMIS.2009.5031428.

[31] Y.-M. Chen, I. Bajic, Compressed-domain moving region segmentation with pixel precision using motion integration, in: IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, 2009, PacRim 2009, 2009, pp. 442–447. http:/dx.doi.org/10.1109/PACRIM.2009.5291331.

[32] HM 15.0 software. URL https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/branches/HM-15.0-dev/.
[33] G.J. Sullivan, J. Ohm, W.-J. Han, T. Wiegand, Overview of the high efficiency video coding (HEVC) standard, IEEE Trans. Circuits Syst. Video Technol. 22 (12) (2012).
[34] Shark multiview sequence. URL http://www.fujii.nuee.nagoya-u.ac.jp/NICT/NICT.htm.
[35] D. Rusanovskyy, P. Aflaki, M.M. Hannuksela, Undo dancer 3DV sequence for purposes of 3DV standardization, in: ISO/IEC JTC1/SC29/WG11 MPEG2010/M20028, Geneva, Switzerland, 2011.
[36] P. Goorts, S. Maesen, M. Dumont, S. Rogmans, P. Bekaert, Free viewpoint video for soccer using histogram-based validity maps in plane sweeping, in: Proceedings of the Ninth International Conference on Computer Vision Theory and Applications, VISAPP 2014, INSTICC, 2014.
[37] HTM 13.0 software. URL https://hevc.hhi.fraunhofer.de/svn/svn_3DVCSoftware/tags/HTM-13.0/.
[38] K. Müller, H. Schwarz, D. Marpe, C. Bartnik, S. Bosse, H. Brust, T. Hinz, H. Lakshman, P. Merkle, F. Rhee, G. Tech, M. Winken, T. Wiegand, 3D high-efficiency video coding for multi-view video and depth data, IEEE Trans. Image Process. 22 (9) (2013) 3366–3378. http://dx.doi.org/10.1109/TIP.2013.2264820.
[39] K. Müller, A. Vetro, Common test conditions of 3DV core experiments, JCT3V-G1100, Joint Collaborative Team on Video Coding, JCT-VC, San Jose, USA, January 2014.
[40] Y. Liu, Q. Dai, Z. You, W. Xu, Rate-prediction structure complexity analysis for multi-view video coding using hybrid genetic algorithms, Proc. SPIE 6508 (2007). http://dx.doi.org/10.1117/12.703849. p. 650804–650804-8.
[41] L. Rakêt, L. Roholm, A. Bruhn, J. Weickert, Motion compensated frame interpolation with a symmetric optical flow constraint, in: Advances in Visual Computing, in: Lecture Notes in Computer Science, vol. 7431, Springer, Berlin, Heidelberg, 2012, pp. 447–457.
[42] J. Zhai, K. Yu, J. Li, S. Li, A low complexity motion compensated frame interpolation method, in: IEEE International Symposium on Circuits and Systems, Kobe, Japan, 2005.
[43] M. Schmeing, X. Jiang, Depth image based rendering, in: P.S. Wang (Ed.), Pattern Recognition, Machine Intelligence and Biometrics, Springer, Berlin, Heidelberg, 2011, pp. 279–310. http://dx.doi.org/10.1007/978-3-642-22407-2_12.
[44] S. Zinger, L. Do, P.H.N. de With, Free-viewpoint depth image based rendering, J. Vis. Commun. Image Represent. 21 (5–6) (2010) 533–541. http://dx.doi.org/10.1016/j.jvcir.2010.01.004.