

# Rate-Distortion Optimal Bit Allocation for Object-based Video Coding

Haohong Wang, Guido M. Schuster<sup>\*</sup>, and Aggelos K. Katsaggelos<sup>+</sup>

Qualcomm CDMA Technology,  
Qualcomm Inc., San Diego, CA 92121  
Email: haohongw@qualcomm.com

<sup>\*</sup> Abteilung Elektrotechnik  
Hochschule für Technik, Rapperswil  
CH-8040 Rapperswil, Switzerland  
Email: guido.schuster@hsr.ch

<sup>+</sup>Department of Electrical and Computer Engineering  
Northwestern University, Evanston, IL 60208, USA  
Email: aggk@ece.northwestern.edu

Manuscript Type: Transaction Paper

Color Illustrations: No

Submit to: IEEE Transaction on Circuits and System for Video Technology

Submission Date: June 16, 2003

Earlier versions of this work appeared in ICME'03 [20] and ICIP'03 [21].

## Abstract

In object-based video encoding, the encoding of the video data is decoupled into the encoding of shape, motion, and texture information, which enables certain functionalities, like content-based interactivity and content-based scalability. The fundamental problem, however, of how to jointly encode this separate information to reach the best coding efficiency has not been studied thoroughly. In this paper, we present an operational rate-distortion optimal scheme for the allocation of bits among shape, motion, and texture in object-based video encoding. Our approach is based on Lagrangian relaxation and dynamic programming. We implement our algorithm on the MPEG-4 video verification model, although it is applicable to any object-based video encoding scheme. The performance is assessed utilizing a proposed metric that jointly captures the distortion due to the encoding of the shape and texture. Experimental results demonstrate that the gains of lossy shape encoding depend on the percentage the shape bits occupy out of the total bit budget. This gain may be small or may be realized at very low bit rates for certain typical scenes.

**Keywords:** Rate-distortion, video coding, object-based video, shape coding, MPEG-4.

## 1. Introduction

In recent years, object-based video coding has become one of the most important topics in the visual communication field, along with the development of new functions for modern interactive multimedia applications. The MPEG-4 standard [1] was developed as the first international multimedia standard that addresses object-based video representation. The central concept in MPEG-4 is that of the video object (VO), which is characterized by intrinsic properties such as shape, texture, and motion. For each arbitrarily shaped video object, each frame of a VO is called a video object plane (VOP). The VOP encoder essentially consists of separate encoding schemes for shape and texture. The purpose of the use of shape is to achieve better subjective picture quality, increased coding efficiency, as well as, object-based video representation and interactivity. The shape information for a VOP, also referred to as an alpha-plane, is specified by a binary array corresponding to the rectangular bounding box of the VOP specifying whether an input pixel belongs to the VOP or not, or a set of transparency values ranging from 0 (completely transparent) to 255 (opaque). In this paper, only a binary alpha plane is considered, although the proposed algorithm can be extended to the grayscale alpha plane cases. The texture information for a VOP is available in the form of a luminance (Y) and two chrominance (U, V) components. It is important to point out that the standard does not describe the method for creating VOs, that is, they may be created in various ways depending on the application.

In object-based video compression, the optimal allocation of bits among shape, texture and motion is a fundamental problem. The difficulty of this optimal bit allocation problem is due to the dependencies between shape and texture of an object, as well as, the dependencies at the macroblock level due to motion compensated predictive coding and differential encoding.

Shape and texture have been jointly considered in object boundary encoding and rate control. In [2,3], vertex based boundary encoding is considered utilizing an adaptive distortion metric which is based on texture information. More specifically, texture gradient information determines the local degree of trust in the accuracy of the shape information and subsequently the allocation of

bits in encoding the boundary (i.e., more bits are allocated to the encoding of the parts of the boundary which are trusted more). In [4-6], adaptive shape-coding control mechanisms are proposed to provide a tradeoff between texture and shape coding accuracy. The shape threshold value,  $\text{Alpha\_TH}$ , which controls the accuracy of the encoded shape, is set adaptively based on previous coding information, thus balancing the bit allocation without introducing excessive distortion. The adjustment of  $\text{Alpha\_TH}$  is mainly based on the number of frames skipped (as indicated by the value of  $\text{FrameSkip}$ ) at a given time instant and a skip threshold ( $\text{Skip\_TH}$ ). For example, if  $\text{FrameSkip} > \text{Skip\_TH}$ ,  $\text{Alpha\_TH}$  is increased, resulting in the allocation of fewer bits and therefore a coarser shape approximation for the next frame. Otherwise,  $\text{Alpha\_TH}$  is reduced, resulting in a better shape approximation for the next frame. In these schemes, however, the distortion from the encoding of shape and texture is neither explicitly nor jointly studied, and the use of  $\text{Alpha\_TH}$ , may not affect the shape approximation as much as expected. So far, none of these approaches provides a rate distortion optimal bit allocation.

In this paper, we propose an operational rate-distortion optimal bit allocation scheme for object-based video coding. The algorithm is based on Lagrangian relaxation and dynamic programming. We implemented our scheme on the MPEG-4 verification model. It is important to point out that we are not proposing an optimal rate control mechanism; instead we assume there is a rate controller available to assign the bit budget for each frame (or VOP for MPEG-4), or that, as a special case, the bit budget is constant for each frame (for example, in some constant-bit-rate applications). Utilizing the bit budget constraint, the proposed algorithm provides the optimal coding parameters to assure the best visual quality of the reconstructed video frame.

The rest of the paper is organized as follows. In section 2, a brief overview of object-based video coding approaches is provided. In section 3, the problem formulation is presented. Section 4 demonstrates the optimal solution. Section 5 provides the implementation details of our method on the MPEG-4 verification model and experimentally demonstrates its performance. We draw conclusions in the last section.

## 2. Overview of object-based video coding

Compared to conventional frame-based video coding, object-based video coding is based on the concept of encoding arbitrarily shaped video objects. The history of object-based video coding can be traced back to 1985, when a region-based image coding technique was first published [7], which was later extended to video encoding [8]. In 1989, an object-based analysis-synthesis coder (OBASC) was developed [9]. The image sequence is divided into arbitrarily shaped moving objects, which are encoded independently. The motivation behind this is that the shape of moving objects is more important than their texture, and that human observers are less sensitive to geometric distortions than coding artifacts of block-based coders. OBASC was mainly successful for simple video sequences.

Shape coding is a relatively new research topic (for a recent review see [10] and references therein). In this work, we only consider the Context-based Arithmetic Encoding (CAE) method [11], which was adopted by the MPEG-4 standard. CAE is a bitmap-based method, which encodes for each pixel whether it belongs to the object or not; it is also a block-based method, where the binary shape information is coded utilizing the macroblock structure, by which binary alpha data are grouped within 16x16 binary alpha blocks (BAB) (see Fig. 1). Each BAB (if neither transparent nor opaque) is coded using CAE. A template of 10 pixels for intra mode (9 pixels for inter mode) is used to define the context for predicting the alpha value of the current pixel. The templates for Intra and Inter BAB encoding are shown in Fig. 2. A probability table is predefined for the context of each pixel. After that, the sequence of pixels within the BAB drives an arithmetic encoder with a pair of alpha value and its associated probability. Due to the support of the models in Fig. 2, the encoding of a BAB depends on its neighbors to the left, above, above-left, and above right.

BABs are classified into transparent, opaque, and border macroblocks using a test against the target BAB containing only zero-valued pixels and BAB containing only 255-valued pixels. In the classification, each BAB is subdivided into 16 elementary sub-blocks, each of size 4x4, and

the BAB is classified as transparent or opaque only if all of these sub-blocks are of the same classification. The sum of absolute differences between the sub-block under test and the target sub-block is compared to  $16 * \text{Alpha\_TH}$ , where the Alpha\_TH can take values from the set  $\{0, 16, 32, 64, \dots, 256\}$ . To reduce the bit-rate, lossy representation of a border BAB might be adopted. According to it, the original BAB is successively downsampled by a conversion ratio factor (CR) of two or four, and then up-sampled back to the full-resolution. In Inter-mode, there are seven BAB coding modes. The motion vector (MV) of a BAB is reconstructed by predictive decoding it as  $MV = MVP + MVD$ , where MVP is the motion vector predictor (see Fig. 3) and MVD (motion vector differences) lies in the range of  $[-16, +16]$ . Upon selection of MVP, the motion compensated (MC) error is computed by comparing the BAB predicted by MVP and the current BAB. If the computed MC error is smaller or equal to  $16 * \text{Alpha\_TH}$ , for all 4x4 sub-block, the MVP is directly utilized. Otherwise, MV is determined by searching around the MVP to find the location that minimizes the sum of absolute differences (SAD) of the 16x16 MC error.

Texture coding approaches in MPEG-4 are similar to most of the existing video coding standards. The VOPs are divided into 8x8 blocks followed by 2D 8x8 discrete cosine transforms (DCT). The resulting DCT coefficients are quantized and the DC coefficients are also quantized (QDC) using a given step size. After quantization, the DCT coefficients in a block are zigzag scanned and an 1D string of coefficients is formed for entropy coding. In MPEG-4, the texture content of the macroblock's bitstream depends to a great extent on the reconstructed shape information. Before encoding, low-pass extrapolation (LPE) padding [12] (non-normative tool) is applied to each 8x8 boundary (non-transparent and non-opaque) blocks. This involves taking the average of all luminance/chrominance values over all opaque pixels of the block, and all transparent pixels are given this average value. If the adaptive DC prediction is applied, the predicted DC selects either the QDC value of the immediately previous block or that of the block immediately above it. With the same prediction direction, either coefficients from the first row or the first column of a

previously coded block are used to predict the coefficients of the current blocks. For predictive VOPs (P-VOPs), the texture data can be predictively coded, and a special padding technique called macroblock-based repetitive padding is applied on the reference VOP before motion estimation. After motion estimation, a motion vector and the corresponding motion-compensated residual are generated for each block. The motion vectors are coded differentially (see Fig. 4), while the residual data are coded as intra-coded texture data, as mentioned above.

At the decoder side, the shape and texture data are composed to reconstruct the original video data. Since in this paper we assume that the shape information is represented by a binary alpha plane, after composition, only the corresponding texture within the shape boundary is kept (an example is shown in Fig. 5).

### 3. Problem Formulation

The problem at hand is to control both the shape and texture coding parameters to minimize the total (shape, texture, and motion) bit rate required to transmit a video sequences at some acceptable level of quality. These coding parameters for MPEG-4 will be explained in detail in section 5.1. We therefore formulate the following optimization problem

$$\text{Minimize } R, \text{ subject to } D \leq D_{max}, \quad (1)$$

where  $R$  is the total bit rate per frame,  $D$  is the resulting frame distortion, and  $D_{max}$  is the maximum allowable distortion. The same techniques, to be presented later, can be applied to solve the dual problem, that is,

$$\text{Minimize } D, \text{ subject to } R \leq R_{budget}, \quad (2)$$

where  $R_{budget}$  is the available bit budget per frame.

In object-based video, both overlapping and non-overlapping VOPs are allowed. However, in both cases, video objects are encoded and transmitted separately. The user at the decoder side has the flexibility to determine the composition order of the objects; therefore, it is not practical to consider the joint encoding of various video objects unless there is a feedback-channel from the

decoder providing to the encoder the information about the composition order of the video objects. In this paper, we do not assume the existence of such feedback-channel. Therefore, each video object forms a separate video sequence for processing, and hence the terms of frame and VOP are used interchangeably hereafter. That is, in this work, the optimal bit allocation among shape, motion, and texture is considered at the VOP level.

### 3.1 Rate

Let us denote by  $\{m_1, m_2, \dots, m_N\}$  the  $N$  raster-scan ordered macroblocks in the VOP, and by  $s_i$  the shape classification of each  $8 \times 8$  block of a BAB associated with  $m_i$ . The set of coding parameters for a macroblock is referred to as the shape (or texture) decision vector. Let us denote by  $V = \{V_1, V_2, \dots, V_N\}$  the set of admissible shape decision vectors for the macroblocks, by  $v_i$  a shape decision vector for the  $i$ th macroblock ( $v_i \in V_i$ ), and  $v = \{v_1, v_2, \dots, v_N\}$  the set of shape decision vectors for the VOP. Then,  $s_i$  is a function of the shape decision vectors  $v_{i-a}, \dots, v_i$ , where  $a$  is the number of past macroblocks, macroblock  $m_i$  depends on. Typically,  $a$  is greater than 1, recalling that the encoding of a BAB depends on its neighbors to the left, above, above left and above right. Similarly, let us denote by  $W = \{W_1, W_2, \dots, W_N\}$  the set of admissible texture decision vectors,  $w_i$  a texture decision vector for  $m_i$  ( $w_i \in W_i$ ), and  $w = \{w_1, w_2, \dots, w_N\}$  the set of texture decision vectors for the VOP. Let us denote by  $R_{S_i}(v_{i-a}, \dots, v_i)$  the shape bit rate including the shape motion vector if present, and  $R_{T_i}(s_i, w_{i-b}, \dots, w_i)$  the texture bit rate including the texture motion vector if present, for macroblock  $m_i$ , where  $b$  is the number of previous macroblocks the texture of  $m_i$  depends on. The reason that  $R_{T_i}(s_i, w_{i-b}, \dots, w_i)$  depends on  $s_i$  is that the texture for transparent macroblocks inside the VOP is not coded. Clearly,

$$R = R_{\text{symax}} + \sum_{i=1}^N R_{S_i}(v_{i-a}, \dots, v_i) + \sum_{i=1}^N R_{T_i}(s_i, w_{i-b}, \dots, w_i), \quad (3)$$



where  $R_{syntax}$  represents the bits allocated to the data structure syntax of the VOP, and it is a fixed value if we assume each VOP frame is packed into a separate packet.

### 3.2 Distortion

In MPEG-4 shape and texture are measured separately [13]. The following expression is used to measure shape distortion

$$D_n = \frac{S_M}{S_o}, \quad (4)$$

where  $S_M$  is the number of mismatched pixels in the reconstructed VOP and  $S_o$  the number of opaque pixels in the original VOP. This is also termed “relative area error”.

The combined-channel peak signal-to-noise ratio (PSNR) is used to measure texture distortion.

For an  $l \times h$  pixel image in the 4:2:0 format, it is given in decibels (dB) by

$$PSNR = 10 \log \left\{ \frac{1.5 \times l \times h \times 255^2}{\sum_{x=0}^{l-1} \sum_{y=0}^{h-1} (Y(x,y) - Y'(x,y))^2 + \sum_{x=0}^{l/2-1} \sum_{y=0}^{h/2-1} [(U(x,y) - U'(x,y))^2 + (V(x,y) - V'(x,y))^2]} \right\}, \quad (5)$$

where  $Y(x,y)$ ,  $U(x,y)$ ,  $V(x,y)$  are the original YUV intensity values of pixel  $(x,y)$ ,  $Y'(x,y)$ ,  $U'(x,y)$ ,  $V'(x,y)$  are the corresponding reconstructed pixel intensity values, and the factor 1.5 is due to down sampling of the chrominance components by a factor of 2.

The metrics above are used separately and therefore they misrepresent the quality of the object-based encoded video. For example, a high PSNR when combined with a large  $D_n$  might provide a low quality video. A joint shape and texture distortion metric is therefore required. However, to the best of our knowledge, no such metric has been adopted and therefore it still remains an open problem.

The establishment of such a metric is not the focus of this paper, but instead the development of a framework for the optimal bit allocation in object-based video coding, given a joint shape and

texture distortion metric. In order to proceed with the development of the algorithm and demonstrate its performance, we propose the following distortion metric

$$D = \alpha D_n + (1 - \alpha) \beta D_T, \quad \alpha \in [0, 1], \quad (6)$$

where  $D_n$  is defined in (4),  $\alpha$  is a control parameter determined by the application,  $\beta$  is a scale parameter to ensure that the value of  $D_n$  and  $D_T$  are in the same range, and  $D_T$  is the texture distortion of the object expressed as

$$D_T = \frac{\sum_{(x,y) \in S_o^Y} (Y(x,y) - Y'(x,y))^2 + \sum_{(x,y) \in S_o^{UV}} [(U(x,y) - U'(x,y))^2 + (V(x,y) - V'(x,y))^2]}{1.5 S_o}, \quad (7)$$

where  $S_o^Y$  and  $S_o^{UV}$  denotes the set of opaque pixels in the Y and U, V planes, respectively,  $S_o$  the number of pixels in  $S_o^Y$  and  $S_o^{UV}$ . The distortion metric in (6) accounts for errors due to both shape and texture encoding, and it is based on the conventional distortion criteria for image/video quality evaluation. By adjusting the value of  $\alpha$ , the user can determine the relative importance between shape and texture distortion. It is mentioned here that distortion metrics other than (6) can be used with the framework presented here. Clearly, the resulting optimal solution as well as the complexity of the optimization algorithm depends on the specific distortion method used.

Let us now denote by  $D_i(s_i, w_{i-1}, \dots, w_i)$  the distortion for macroblock  $m_i$  defined as

$$D_i(s_i, w_{i-1}, \dots, w_i) = \alpha d_{i,A} + (1 - \alpha) \beta \left\{ \sum_{(x,y) \in S_o^Y} d_{i,Y}(x,y)^2 + \sum_{(x,y) \in S_o^{UV}} [d_{i,U}(x,y)^2 + d_{i,V}(x,y)^2] \right\} / 1.5, \quad (8)$$

where  $d_{i,A}$  is the number of mismatched (alpha plane) pixels inside the macroblock, and  $d_{i,Y}(x,y)$ ,  $d_{i,U}(x,y)$ , and  $d_{i,V}(x,y)$  are the differences in intensity values for the Y, U and V components at pixel  $(x,y)$  of macroblock  $m_i$ . It is noted here that in obtaining the reconstructed intensity value for pixel  $(x,y)$  of  $m_i$ ,  $w_{i-1}$ ,  $w_{i-2}$  ..., and  $w_{i-b}$  were also used, due to the differential encoding of DCT coefficients and motion vectors. Due to (8) it is now clear that the distortion in Eq. (6) is equal to

$$D = \sum_{i=1}^N D_i(s_i, w_{i-b}, \dots, w_i) / S_o. \quad (9)$$

#### 4. Optimal solution

In this section, we provide an optimal solution for problem (1), which can be rewritten as

$$\begin{aligned} & \text{Minimize } \sum_{i=1}^N R_{S_i}(v_{i-a}, \dots, v_i) + \sum_{i=1}^N R_{T_i}(s_i, w_{i-b}, \dots, w_i), \\ & \text{such that } \sum_{i=1}^N D_i(s_i, w_{i-b}, \dots, w_i) \leq S_o D_{\max}, \end{aligned} \quad (10)$$

where the minimization is performed with respect to the variables  $s_i, v_{i-a}, \dots, v_i$ , and  $w_{i-b}, \dots, w_i$ .

We now define the set of admissible decision vectors  $U = V \times W$ . For each macroblock  $m_i$  there is a decision vector  $u_i = (v_i, w_i)$  and (10) can be simplified as

$$\text{Minimize } \sum_{i=1}^N [R_i(u_{i-a}, \dots, u_i)], \text{ such that } \sum_{i=1}^N D_i(u_{i-a}, \dots, u_i) \leq S_o D_{\max}, \quad (11)$$

where  $R_i(u_{i-a}, \dots, u_i) = R_{S_i}(v_{i-a}, \dots, v_i) + R_{T_i}(s_i, w_{i-b}, \dots, w_i)$ ,  $D_i(u_{i-a}, \dots, u_i) = D_i(s_i, w_{i-b}, \dots, w_i)$ , and  $u = [u_1, u_2, \dots, u_N]^T$  (without loss of generality, we assume that  $a \geq b$ ).

We derive a solution to problem (11) using the Lagrange multiplier method to relax the constraint, so that the relaxed problem can be solved using a shortest path algorithm. We first define the Lagrangian cost function

$$J_\lambda(u) = \sum_{i=1}^N [R_i(u_{i-a}, \dots, u_i) + \lambda D_i(u_{i-a}, \dots, u_i)], \quad (12)$$

where  $\lambda$  is the Lagrange multiplier. It has been shown in [14] and [15] that if there is a  $\lambda^*$  such that  $u^* = \arg[\min_u J_{\lambda^*}(u)]$ , and which leads to  $\sum_{i=1}^N D_i(u_{i-a}, \dots, u_i) = S_o D_{\max}$ , then  $u^*$  is also an

optimal solution to (11). It is well known that when  $\lambda$  sweeps from zero to infinity, the solution to (12) traces the convex hull of the operational rate distortion function, which is a non-increasing function. Hence, bisection or the fast convex search presented in [16] can be used to find  $\lambda^*$ .

Therefore, if we can find the optimal solution to the unconstrained problem

$$\text{Minimize}_u \sum_{i=1}^N [R_i(u_{i-a}, \dots, u_i) + \lambda D_i(u_{i-a}, \dots, u_i)], \quad (13)$$

we can find the optimal  $\lambda^*$ , and the convex hull approximation to the constrained problem (11).

To implement the algorithm for solving the optimization problem (13), we create a cost function  $C_k(u_{k-a}, \dots, u_k)$ , which represents the minimum total rate and distortion up to and including macroblock  $m_k$ , given that  $u_{k-a}, \dots, u_k$  are the decision vectors for macroblocks  $m_{k-a}, \dots, m_k$ . Clearly,

$$J_\lambda(u) = \min_{u_{N-a}, \dots, u_N} C_N(u_{N-a}, \dots, u_N). \quad (14)$$

The key observation for deriving an efficient algorithm is the fact that given  $a+1$  decision vectors  $u_{k-a-1}, \dots, u_{k-1}$  for macroblocks  $m_{k-a-1}, \dots, m_{k-1}$ , and the cost function  $C_{k-1}(u_{k-a-1}, \dots, u_{k-1})$ , the selection of the next decision vector  $u_k$  is independent of the selection of the previous decision vectors  $u_1, u_2, \dots, u_{k-a-2}$ . This is true since the cost function can be expressed recursively as

$$C_k(u_{k-a}, \dots, u_k) = \min_{u_{k-a-1}, \dots, u_{k-1}} [C_{k-1}(u_{k-a-1}, \dots, u_{k-1}) + R_k(u_{k-a}, \dots, u_k) + \lambda D_k(u_{k-a}, \dots, u_k)]. \quad (15)$$

The recursive representation of the cost function above makes the future step of the optimization process independent from its past step, which is the foundation of dynamic programming. It is mentioned here that the presented algorithm can also be used to solve the dual problem (2).

The problem can be converted into a graph theoretic problem of finding the shortest path in a directed acyclic graph (DAG) [17]. The computational complexity of the algorithm is  $O(N \times |U|^{\max(a,b)+1})$ , with  $|U|$  denoting the cardinality of  $U$ , which depends directly on the value of  $a$  and  $b$ , but is still much more efficient than the exponential computational complexity of an exhaustive search algorithm. In [18], a similar DP algorithm is applied to solve the bit allocation problem between displacement vector field and displaced frame difference in motion-compensated video coding.

The proposed optimal scheme is also applicable to grayscale alpha plane cases, where the gray-level alpha plane is encoded as its support function and the alpha values on the support, where the

support function is encoded by binary shape coding and the alpha values are encoded as texture data. Clearly, Eq. (6) has to be modified to consider the new factor brought in.

## 5. Experimental results

The problem formulation and solution approach presented in this paper are general and applicable to any texture encoding method and block-based shape coding methods. In our experiment, we implemented the proposed optimal bit allocation scheme utilizing the MPEG-4 verification model [1,13,19], and thus selecting the conventional 2D DCT method for texture encoding. We report the simulation details in section 5.1 and the experimental results in section 5.2.

### 5.1 System simulations

In the MPEG-4 verification model for all macroblocks of a VOP, the same value of Alpha\_TH is used to control the shape approximation level and the shape motion vector. In addition, the values of shape parameters, such as BAB type and CR, are transmitted to the decoder, while the value of Alpha\_TH is not. This means that the parameter Alpha\_TH only indirectly affects the rate-distortion characteristic of the video codec, and it is therefore not directly accounted for in the optimization.

The encoding schemes of shape and texture make the macroblocks highly dependent on their neighbors. In shape coding, the border macroblocks at the VO level need to be further processed by CAE, which makes the encoding of a BAB depend on its neighbors to the left, above, above-left, and above right. The processes of adaptive DC/AC prediction and the motion vector prediction and encoding involve the current macroblock and its neighbors. As shown in Fig. 6, the encoding of the current macroblock will only depend on the encoding of macroblocks 1, 2, 3 and 4. In other words, after the decision vectors for macroblocks 1, 2, 3, 4, and the current macroblock are decided, the rate for the current macroblock is fixed for both inter and intra mode encoding.

As is the case with all video compression standards, only the structure of the decoder is specified by the standard. Our optimization work addresses the optimal selection of the coding parameters so that the video is coded to meet certain constraints in quality or bit rate. From the MPEG-4 VOP data structure, the adjustable parameters for intra-mode include BAB type, CR, ST, and quantization step size. For inter-mode coding, the additional parameters MVDs and MVD need to be taken into account (see Table 1 for definition of these coding parameters). In MPEG-4, it is possible to modify the quantizer value at the macroblock level, although only a small adjustment (-1 or -2 or +1 or +2) in the value of the most recent quantizer is permitted.

We assume that the  $N$  macroblocks  $m_1, m_2, \dots, m_N$  of the current VOP are arranged into  $K$  columns, and they are numbered following the horizontal scan order. If the block in Fig. 6 represents a VOP, then  $N=49$  and  $K=7$ . Therefore,  $a$  in Eq. (11) is equal to  $K+1$ . Every macroblock has a BAB type  $b_i \in B_i$ , an MVDs ( $ms_i \in MS_i$ ), a CR ( $c_i \in C_i$ ), an ST ( $s_i \in S_i$ ), a quantizer step size  $q_i \in Q_i$ , and an MVD ( $mv_i \in MV_i$ ) associated with it, where  $B_i$  is the set of all admissible BAB type for  $m_i$ ,  $MS_i$  is the set of all admissible shape motion vectors for  $m_i$ ,  $C_i$  is the set of all admissible CR for  $m_i$ ,  $S_i$  is the set of all admissible ST for  $m_i$ ,  $Q_i$  is the set of all admissible quantizer step sizes for  $m_i$ , and  $MV_i$  is the set of all admissible texture motion vector for  $m_i$ . Let us define a decision vector  $u_i = [b_i, ms_i, c_i, s_i, q_i, mv_i] \in U_i$  for every macroblock  $m_i$ , with  $U_i = B_i \times MS_i \times C_i \times S_i \times Q_i \times MV_i$  the admissible decision vector set for  $m_i$ .

## 5.2 Experimental results

A number of experiments have been conducted with the QCIF sequences “Akiyo”, “Bream”, “Children” and “Cyclamen”, and for various choices for the value of  $\alpha$ , some of which are reported here. We implemented our codec using the proposed bit allocation scheme and the basic coding algorithms specified in MPEG-4 video verification model. In our experiments, we set  $\beta = 1/255$  in Eq. (6), so that the two quantities  $D_n$  and  $D_T$  are in comparable range. We demonstrate the optimal bit allocation results and discuss their dependency on the value of  $\alpha$  in

the first 2 sets of experiments. Then, we use MoMuSys, a software implementation of MPEG-4 Video Verification Model, as a reference to compare the MPEG-4 results with the ones obtained with the proposed algorithm. It is important to point out that we are using the same algorithm with MoMuSys as demonstrated by the third set of experiments.

In the first set of experiments, we encode the first “Children” frame with various  $\alpha$  values ( $\alpha=0.2$ ,  $\alpha=0.5$ , and  $\alpha=0.8$ ) and various bit budgets (4000, 8000, and 12000). Figure 7 shows the results and Table 2 shows the detailed rate and distortion for each case. In Table 2, the values of  $D$  in Eq. (6),  $D_n$  in Eq. (4), along with  $D_{T,PSNR}$  defined by  $D_{T,PSNR} = -10 \log_{10} \frac{D_T}{255^2}$ . The last measure is shown since it represents the standard way for measuring texture distortion. From this figure and table, it is easy to verify that for  $\alpha=0.8$ , since shape is given higher weight than texture it has been well reserved for all bit rates, while this is not the case for lower values of  $\alpha$  and low bit rates.

In the second set of experiments, we explore the scheme of optimal bit allocation between shape and texture by encoding the first 30 frames of the “Bream” sequence in both intra and inter mode. In Fig. 8 (a), the rate-distortion curves (with  $\alpha=0.5$ ) obtained by the proposed optimal approach are shown. The vertical axis represents the average distortion  $D_{dB}$  over 30 frames, where  $D_{dB} = -10 \log_{10} D$ . The corresponding average shape and texture bit rates are shown in Fig. 8 (b) (Please notice that there is an one-to-one correspondence between the operating points in Figs 8(a) and (b)). It is clear from Fig. 8 (a) that Inter-mode encoding saves a considerable number of bits compared to Intra-mode encoding, and from Fig. 8 (b) that shape information only represents a small portion (in most cases, less than 20%) of the overall bit budget. It is interesting to observe that the texture vs. shape bit rate tradeoff is similar for both intra and inter encoding, and there is a salient turning point in each curve. Let us take a closer look at Fig. 8 (b); for Intra-mode encoding, when the PSNR is between 4 dB and 10.4 dB, the shape bits increase from 360 to around 770 and texture bits increase from 3,000 to around 10,000; after that, the texture bits go up

steeply from 10,000 to 50,000 and shape bits only change slightly. Similarly, for Inter-mode encoding, when the PSNR is between 5 dB and 16 dB, the shape bits increase from 300 to around 650, and the texture bits increase from 3,000 to 10,000. However, when the PSNR keeps increasing to 28.5 dB, the texture bits go up steeply from 10,000 to 40,000 and shape bits only show a slight change. It is very important to notice that the shape bit rate for the “turning point” of the curve is equal to or almost equal to the rate for lossless shape, which means that the proposed optimal approach ends up coding the shape losslessly when the VOP bit budget is high enough. The loss of shape information in this case will cause higher distortion to the reconstructed video than the same loss (in bits) of texture data.

The same experiments was conducted with  $\alpha = 0.2$  and  $\alpha = 0.8$ . For  $\alpha = 0.8$ , the curve in Fig. 9 accepts the following interpretation: for low bit rates, the bits are first allocated to maintain the lowest quality of texture (e.g., QP=31), and the rest of the bits are assigned to shape; as the bit rate increases, the shape is losslessly coded, and the rest of the bits are assigned to code the texture. For  $\alpha = 0.2$  (Fig. 10), the shape clearly is assigned a smaller weight than texture, and therefore the shape is not losslessly encoded until the overall bit rate is considerably increased.

What the experimental results above demonstrate is that in all cases there is a “turning point” in the “average texture bit rate vs. average shape bit rate” plots, below which the shape is encoded in a lossy fashion and above which the shape is encoded in a lossless fashion. That is, after the turning point the proposed algorithm converts to a lossless shape algorithm followed by an operational RD optimal texture algorithm (henceforth referred to as *lossless shape algorithm*).

An important question that arises then is what the location of this turning point depends on? It depends on the complexity of the shape or on the ratio of texture over shape bits, or the percentage of the texture and shape bits with respect to the total bit rate. If this ratio is close to one (i.e., shape and texture bits are about 50%-50%) then there are more bits that if taken away from the shape (resulting in lossy shape) might make a difference in improving the quality of the texture. If, however, this ratio is very large (for example, 20, i.e., the lossless shape is only about



5% of the total bit rate) then this turning point occurs at low bit rates. Now this ratio is also a function of  $\alpha$ ; that is when  $\alpha$  increases the ratio decreases, which comes from the fact that texture is decided (through  $\alpha$ ) to be “less important” than shape and therefore fewer bits are allocated to it (while clearly the bits for lossless shape are constant).

Clearly one of the benefits of the proposed algorithm is that the determination of this turning point is done automatically, that is, one does not have to predict or guess or estimate through any other means (i.e., analysis of the shape complexity) its position. The benefit of the proposed algorithm over the lossless shape algorithm increases the higher (in terms of shape bits) this turning point is (that is, there are more operating points for which lossy shape is the optimal solution). In demonstrating through an example the potential benefit of the proposed algorithm over the lossless shape algorithm for rates below the turning point we present the following experiment.

In Fig. 11, we encode the “Cyclamen” sequence and show the rate-distortion curve by both approaches, where we use distortion  $D$  in Fig. 11 (a) and texture only distortion  $D_T$  in dB in Fig. 11 (b). The gain is up to 2 dB, although the benefits decrease when the bit rate increases as expected. In Fig. 12 (c), we show the resulted first frame when the bit budget is 4850 bits, where the lossless shape approach (as shown in Fig. 12 (a)) spends 2516 bits for shape while the proposed approach (as shown in Fig. 12 (b)) only spends 1045 bits for shape. The savings of 1471 shape bits by the proposed approach improved the texture quality by about 1.7dB. In addition, we find that the compressed bit rate range of the proposed approach can be extended to a much lower bit rate compared to the lossless shape approach, which might be of importance in very low bit rate applications.

In the last set of experiments, we first encode the first “Bream” frame as an I-VOP, and compare the result from our proposed optimal approach with that of MoMuSys, by exhaustively trying all combinations of parameters (Alpha\_TH and QP) which result in all possible operating points, as shown in Fig. 13(a). As expected, our result in addition to providing solutions on the convex hull

of all operating points, also demonstrates a small gain in RD quality, due to the selection of adjustable parameters different than Alpha\_TH or QP. As mentioned before, Alpha\_TH only indirectly affects the rate-distortion characteristic of the video encoding, so that the operating points generated by exhaustively trying all possible Alpha\_TH values may not cover all possible combinations of the coding parameters (like CR, ST), which explains the small RD gains. Our experimental results on other sequences also show the same tendency. In addition, we encode the first “Bream” frame losslessly and use it as a reference to encode the second frame as a P-VOP. We again compare our result with MoMuSys by exhaustively trying all combinations of coding parameters, as shown in Fig. 13(b). The same conclusions are drawn for this experiment. Clearly the benefit of the proposed algorithm is that it is guaranteed to achieve the operational rate distortion optimal performance in an efficient way, that is, without requiring an exhaustive search or relying on the available rate controller.

## **6. Conclusions**

In this paper, we presented an operational rate-distortion optimal bit allocation scheme among shape, motion, and texture for the encoding of object-based video. The solution of the optimization problem is obtained by applying the Lagrange multiplier method and dynamic programming. We addressed the question of under how much is the gain of lossy shape encoding (the outcome of the optimal algorithm) over a lossless shape algorithm. We also described the factors that affect this gain and considered experimental results. There may indeed be cases when the shape bits are only a small fraction of the total bits and hence lossy shape is only an optimal solution for very low bit rates for which the quality may not be acceptable. On the other hand there are scenes for which lossy shape encoding provides a benefit over lossless shape.

Finally, the proposed optimal joint shape and texture encoding scheme is specific to codecs with block-based shape coding approaches. Different mechanisms will be needed for codes employing other shape coding methods.

## Reference

- [1] MPEG-4 video VM 18.0, ISO/IEC JTC1/SC29/WG11 N3908, Pisa, Jan. 2001.
- [2] L. P. Kondi, F. W. Meier, G. M. Schuster, and A. K. Katsaggelos, "Joint optimal object shape estimation and encoding", in *Proc. SPIE Conf. on Visual Communications and Image Processing*, pp. 14-25, SPIE vol. 3309, San Jose, CA, Jan. 28-30, 1998.
- [3] L. P. Kondi, G. Melnikov, and A. K. Katsaggelos, "Joint optimal coding of texture and shape", in *Proc. IEEE International Conference on Image Processing*, Volume III, pp. 94-97, Thessaloniki, Greece, Oct. 2001.
- [4] A. Vetro, H. Sun, and Y. Wang, "MPEG-4 rate control for multiple video objects", *IEEE Trans. Circuits and System for Video Technology*, Vol. 9, NO. 1, pp. 186-199, Feb. 1999.
- [5] A. Vetro, H. Sun, and Y. Wang, "Joint shape and texture rate control for MPEG-4 encoders", in *Proc. IEEE International Conference on Circuits and Systems*, pp. 285-288, Monterey, USA, June 1998.
- [6] H. Le, T. Chiang, and Y. Zhang, "Scalable rate control for MPEG-4 video", *IEEE Trans. Circuits and System for Video Technology*, Vol. 10, NO. 6, pp. 878-894, Sept. 2000.
- [7] M. Kunt, "Second-generation image coding techniques", *Proceedings of the IEEE*, Vol. 73, pp. 549-574, Apr. 1985.
- [8] W. Li and M. Kunt, "Morphological segmentation applied to displaced difference coding", *Signal Processing*, Vol. 38, pp. 45-56, July 1994.
- [9] H. Musmann, M. Hotter, and J. Ostermann, "Object-oriented analysis-synthesis coding of moving images", *Signal Processing: Image Communication*, Vol. 1, pp. 117-138, Oct. 1989.
- [10] A. K. Katsaggelos, L. Kondi, F. W. Meier, J. Ostermann, and G. M. Schuster, "MPEG-4 and rate distortion based shape coding techniques", *Proc. IEEE*, pp.1126-1154, June 1998.
- [11] N. Brady, F. Bossen, and N. Murphy, "Context-based arithmetic encoding of 2D shape sequences", in *Special session on shape coding, ICIP'97*, Santa Barbara, USA. Pp. 29-32, 1997.

- [12] Andre Kaup, "Object-based texture coding of moving video in MPEG-4", *IEEE Trans. Circuits and System for Video Technology*, Vol. 9, NO. 1, pp. 5-15, Feb. 1999.
- [13] N. Brady, "MPEG-4 standardized methods for the compression of arbitrarily shaped video objects", *IEEE Trans. Circuits and System for Video Technology*, Vol. 9, NO. 8, pp. 1170-1189, Dec. 1999.
- [14] H. Everett, "Generalized Lagrange multiplier method for solving problems of optimum allocation of resources", *Oper. Res.*, vol. 11, pp. 399-417, 1963.
- [15] Y. Shoham and A. Gersho, "Efficient bit allocation for an arbitrary set of quantizers", *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, pp. 1445-1453, Sept. 1988.
- [16] G. M. Schuster and A. K. Katsaggelos, "Fast and efficient mode and quantizer selection in the rate distortion sense for H.263", in *SPIE Proc. Conf. Visual Communications and Image Processing*, pp. 784-795, March 1996.
- [17] G. M. Schuster and A. K. Katsaggelos, *Rate-Distortion based video compression: optimal video frame compression and object boundary encoding*, Kluwer Academic Publishers, 1997.
- [18] G. M. Schuster and A. K. Katsaggelos, "A theory for the optimal bit allocation between displacement vector field and displaced frame difference", *IEEE Journal on Selected Areas in Communications*, Vol. 15, NO. 9, pp. 1739-1751, Dec. 1997.
- [19] T. Sikora, "The MPEG-4 video standard verification model", *IEEE Trans. Circuits and Systems for Video Technology*, Vol. 7, NO. 1, pp. 19-31, Feb. 1997.
- [20] H. Wang, G. M. Schuster, A. K. Katsaggelos, "Operational rate-distortion optimal bit allocation between shape and texture for MPEG-4 video coding", in *Proc. Int. Conf. Multimedia & Expo*, pp. 257-260, Baltimore, USA, July 2003.
- [21] H. Wang, G. M. Schuster, A. K. Katsaggelos, "Object-based video compression Scheme with Optimal Bit Allocation Among Shape, Motion and Texture", in *Proc. Int. Conf. Image Processing*, Barcelona, Spain, Sept. 2003.

BAB type	BAB coding modes (for example, transparent or opaque)
CR	Conversion Ratio (subsampling ratio for CAE coding)
ST	Scan Type (predefined scan order of BAB pixels)
MVDs	Motion vector data for shape
MVD	Motion vector data for texture
QP	Texture quantization step size

Table 1. Definition of MPEG-4 coding parameters

<b>Rate Budget</b>	<b>Shape bits</b>	<b>D</b>	<b>D<sub>n</sub></b>	<b>D<sub>T,PSNR</sub>(dB)</b>
<i>4000</i>	<i>362</i>	<i>0.968</i>	<i>0.323</i>	<i>23.54</i>
<i>8000</i>	<i>503</i>	<i>0.352</i>	<i>0.093</i>	<i>27.87</i>
<i>12000</i>	<i>790</i>	<i>0.176</i>	<i>0.024</i>	<i>30.76</i>

(a)  $\alpha=0.2$

<b>Rate Budget</b>	<b>Shape bits</b>	<b>D</b>	<b>D<sub>n</sub></b>	<b>D<sub>T,PSNR</sub>(dB)</b>
<i>4000</i>	<i>493</i>	<i>0.652</i>	<i>0.102</i>	<i>23.27</i>
<i>8000</i>	<i>858</i>	<i>0.236</i>	<i>0.016</i>	<i>27.47</i>
<i>12000</i>	<i>1002</i>	<i>0.115</i>	<i>0.002</i>	<i>30.50</i>

(b)  $\alpha=0.5$

<b>Rate Budget</b>	<b>Shape bits</b>	<b>D</b>	<b>D<sub>n</sub></b>	<b>D<sub>T,PSNR</sub>(dB)</b>
<i>4000</i>	<i>768</i>	<i>0.280</i>	<i>0.030</i>	<i>22.99</i>
<i>8000</i>	<i>1014</i>	<i>0.096</i>	<i>0.001</i>	<i>27.30</i>
<i>12000</i>	<i>1028</i>	<i>0.046</i>	<i>0</i>	<i>30.44</i>

(c)  $\alpha=0.8$

Table 2. Experimental data for first “Children” frame

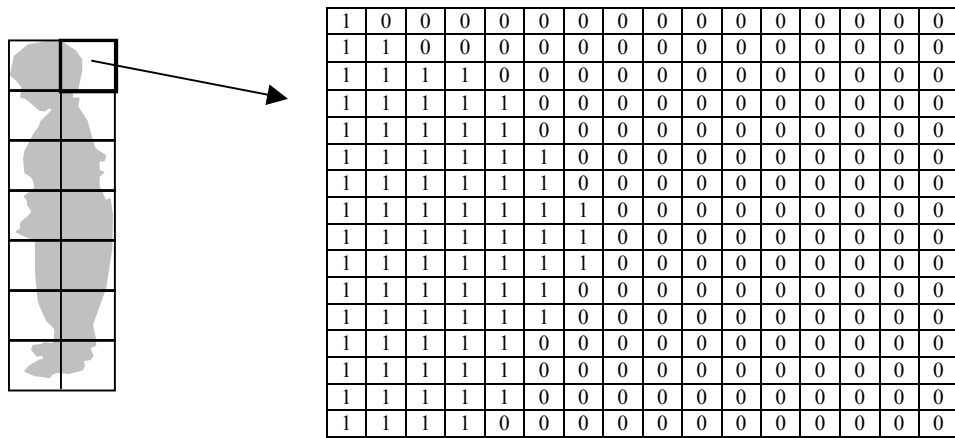


Figure 1. Block-based shape representations (The pixels marked 0 are transparent pixel and those marked 1 are opaque pixels)

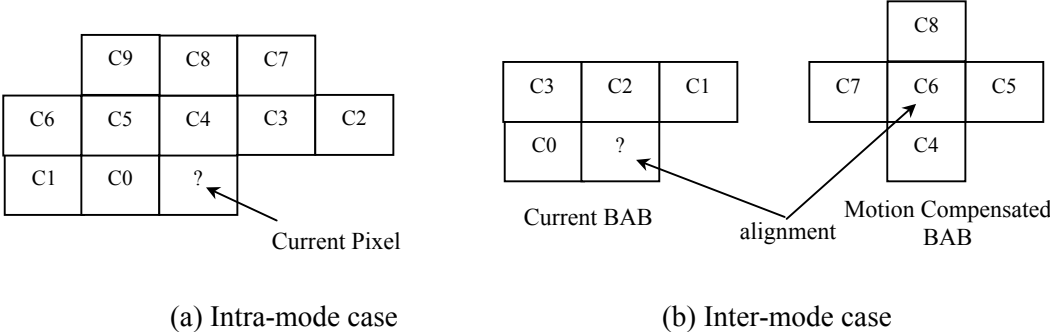


Figure 2. Context for CAE encoding

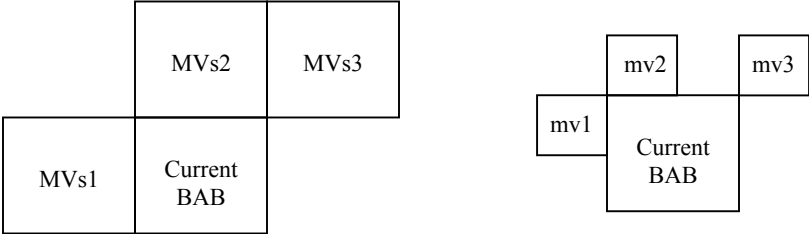


Figure 3. Candidates for motion vector predictors (MVPs): MVs1, MVs2, MVs3 represent shape motion vectors for 16x16 macroblocks, while mv1, mv2, and mv3, are texture motion vectors for 8x8 blocks. The order for selecting MVPs is MVs1, MVs2, MVs3, mv1, mv2, and mv3.

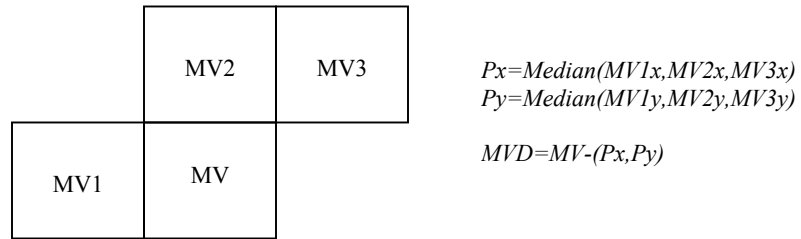


Figure 4. Motion vector prediction

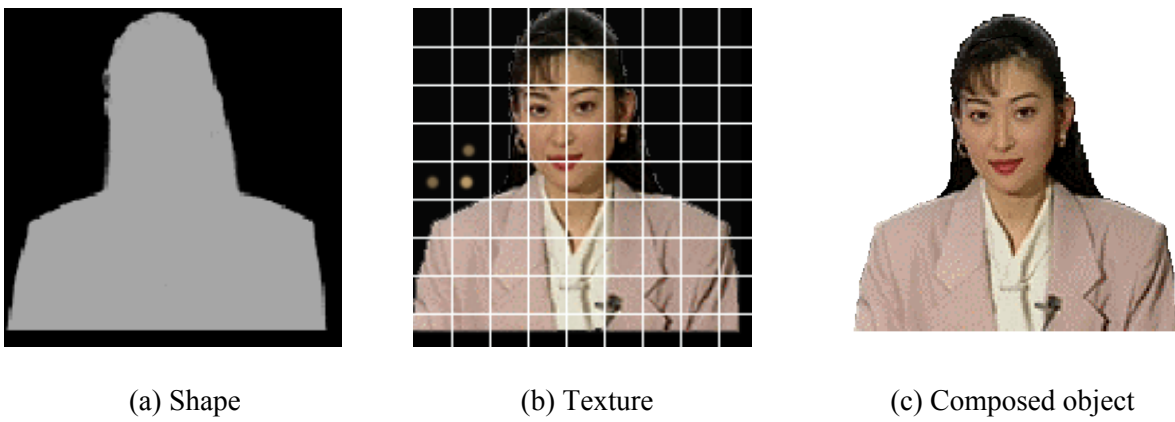


Figure 5. Example of composition of shape and texture

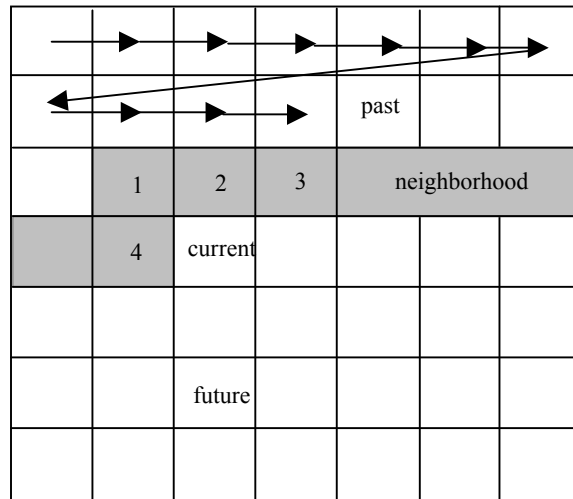


Figure 6. The neighborhood of the current macroblock



$R_{\text{budget}}=4000$

$R_{\text{budget}}=8000$

$R_{\text{budget}}=12000$

(a)  $\alpha = 0.2$



$R_{\text{budget}}=4000$

$R_{\text{budget}}=8000$

$R_{\text{budget}}=12000$

(b)  $\alpha = 0.5$



$R_{\text{budget}}=4000$

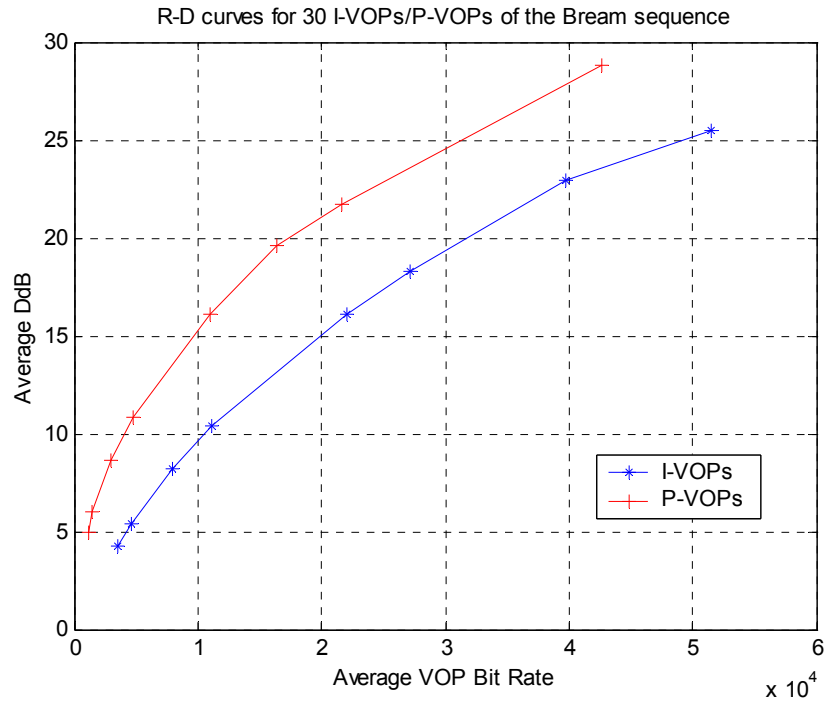
$R_{\text{budget}}=8000$

$R_{\text{budget}}=12000$

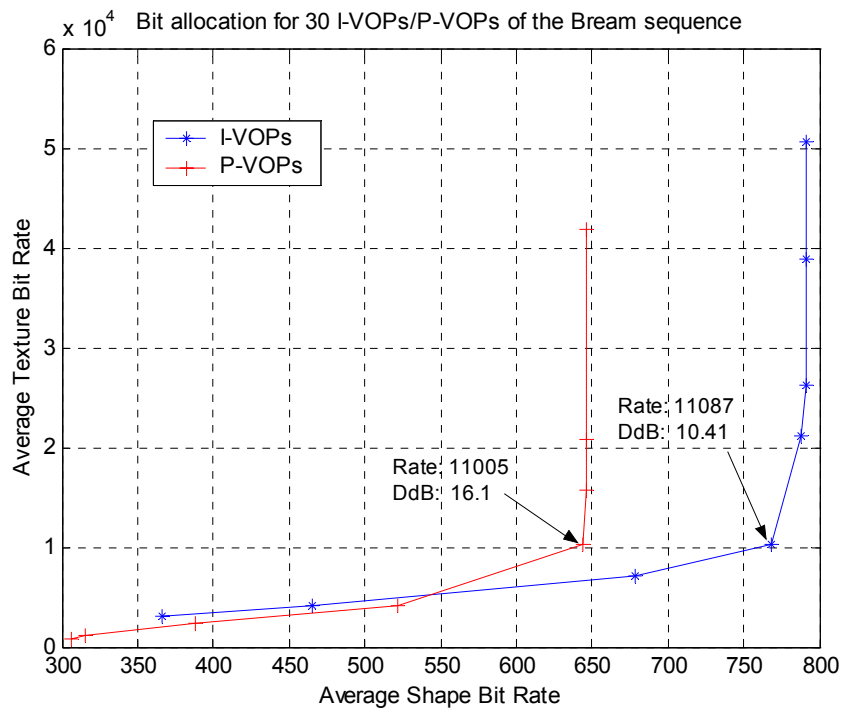
(c)  $\alpha = 0.8$

Figure 7. Reconstructed frame 0 of “Children” sequence for various values of  $\alpha$  and bit budget





(a) VOP PSNR versus VOP bit rate



(b) Shape bit rate versus texture bit rate

Figure 8. Experimental results for “Bream” sequence ( $\alpha = 0.5$ )

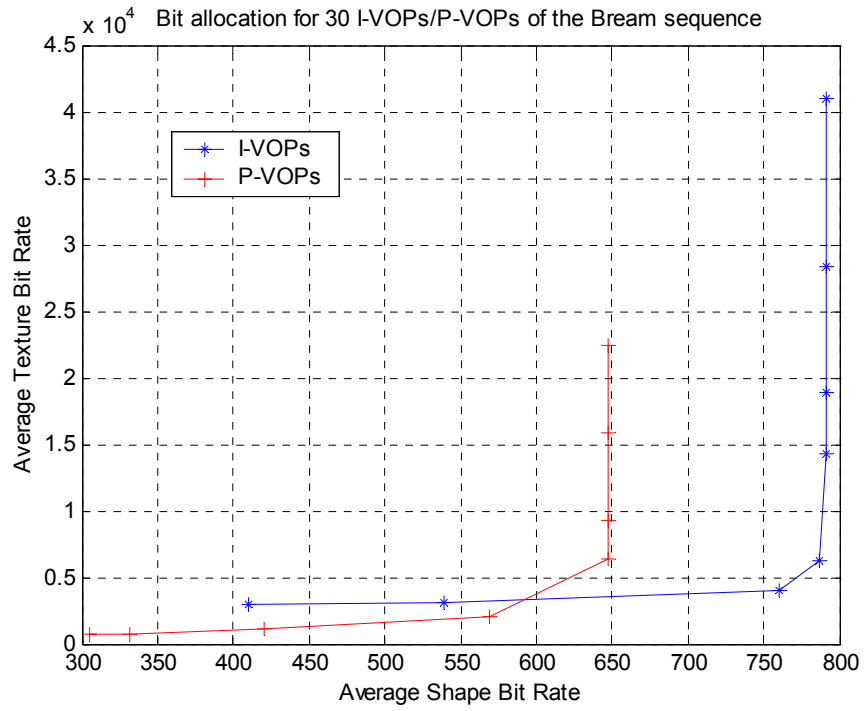


Figure 9. Experimental results for “Bream” sequence ( $\alpha=0.8$ )

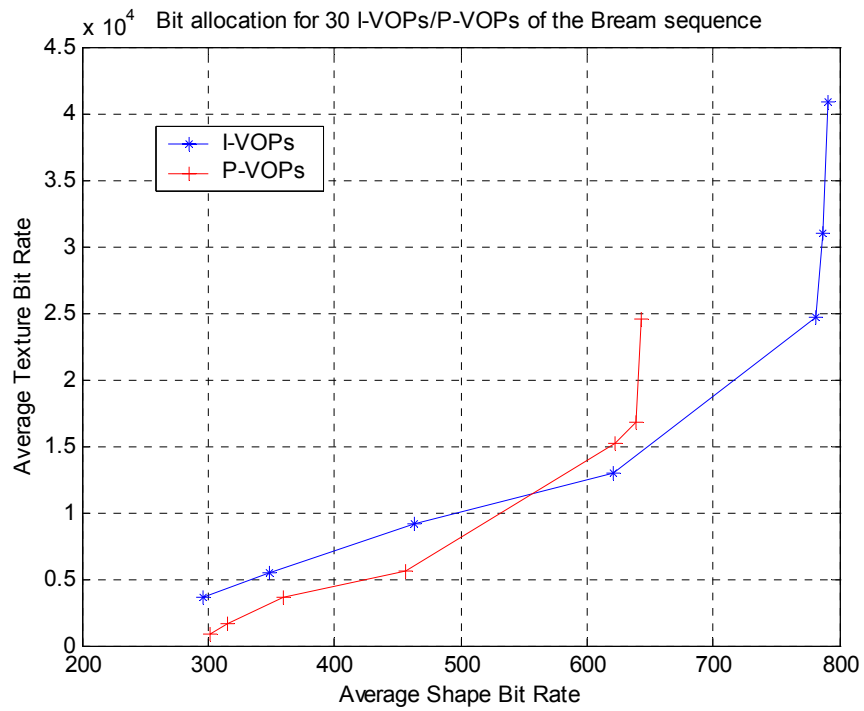
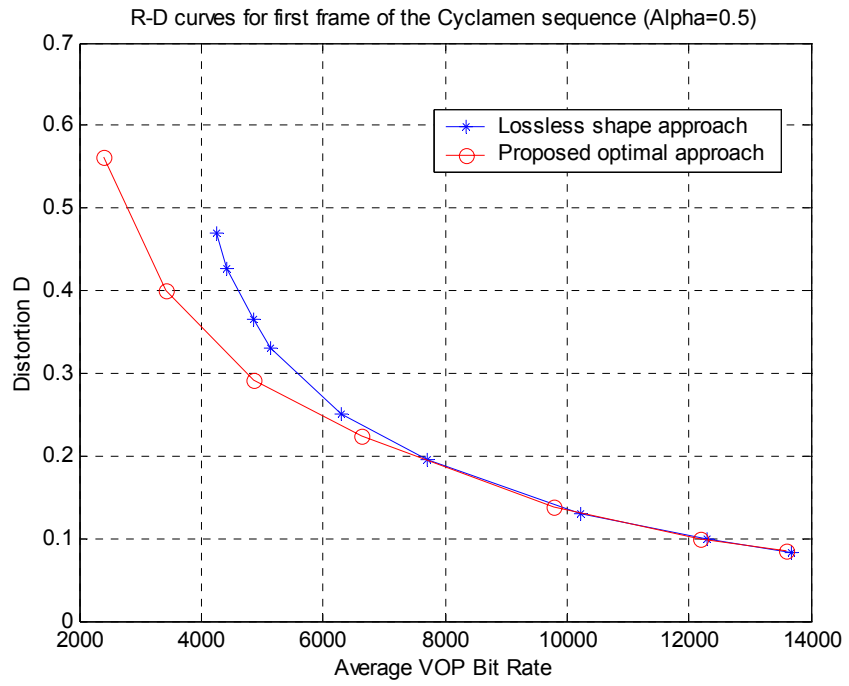
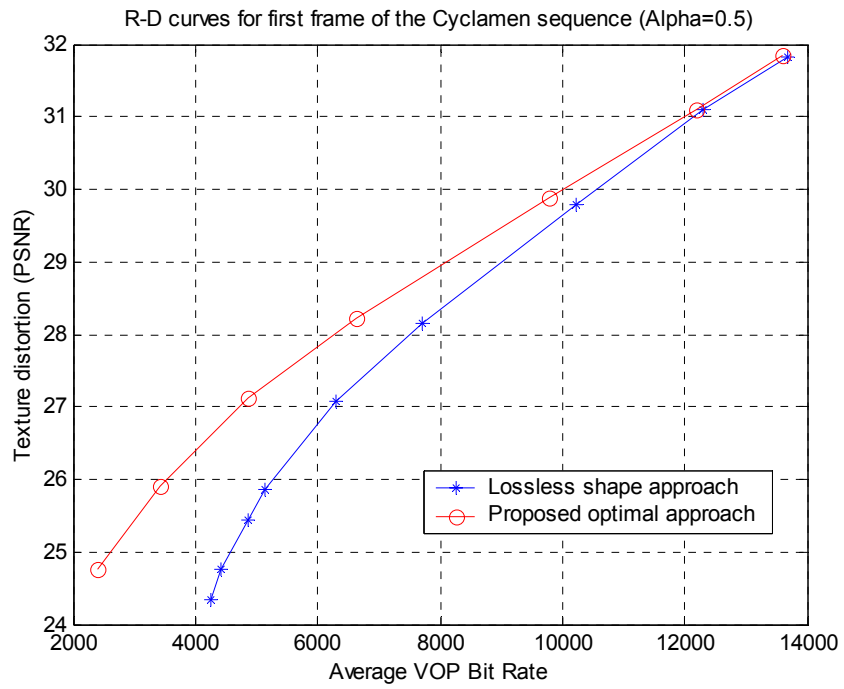


Figure 10. Experimental results for “Bream” sequence ( $\alpha=0.2$ )



(a)



(b)

Figure 11. Rate-distortion curve for encoding “Cyclamen” sequence ( $\alpha=0.5$ )

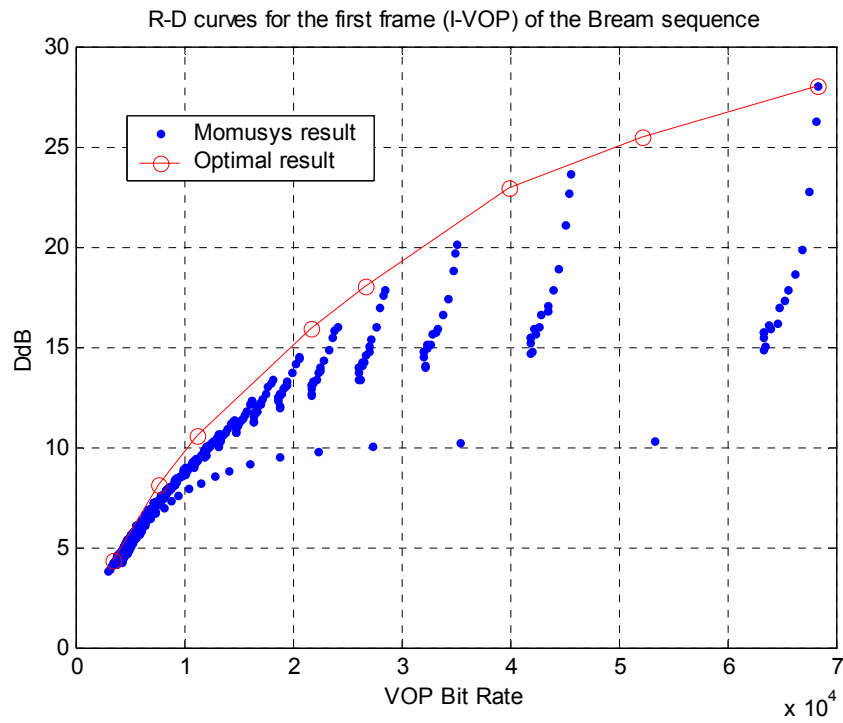


(a) Lossless shape approach  
 $D=0.365, D_{T,PSNR}=25.43\text{dB}$

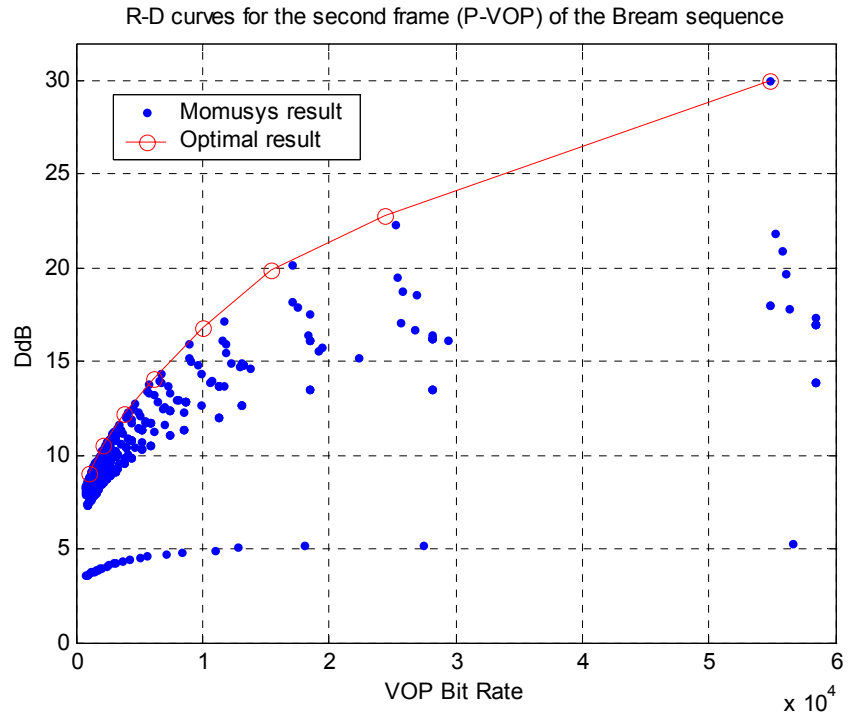
(b) Proposed approach  
 $D=0.292, D_{T,PSNR}=27.13\text{dB}$

(c) Original frame

Figure 12. Rate-distortion curve for encoding “Cyclamen” sequence ( $\alpha=0.5$ )



(a) Intra-mode testing results



(b) Inter-mode testing results

Figure 13. Compare our proposed optimal approach with MoMuSys