RATE CONTROL FOR LAYERED VIDEO COMPRESSION USING MATCHING PURSUITS

Eugene Miloslavsky and Avideh Zakhor

Department of Electrical Eingeering and Computer Sciences, University of California at Berkeley

ABSTRACT

The use of matching pursuits (MP) to compress motion compensated residual video signals and its extension to layered coding have recently been demonstrated in [1, 2]. In this paper we describe a multi-pass rate control scheme for SNR scalable encoding based on MP. The rate control algorithm enforces constant quality on each layer, while keeping the bit budget for each layer at a pre-specified target level. We formulate this as a zero finding problem, and solve it using Newton's method. Experimental results on 14 video sequences are included, showing that layered video can be encoded at constant quality in about 5 encoding iterations per layer, while satisfying bit budget constraints with 1.5% tolerance.

1. INTRODUCTION

Developing scalable video compression algorithms has attracted considerable attention in recent years. Scalable compression is useful in today's heterogeneous networking environment in which different users have different rate, resolution, display, and computational capabilities. Several rate control schemes have recently been proposed for single layer [3, 4, 5, 6] and multi-layer scalable [5, 9] video encoding. The rate control schemes can be roughly classified into two classes: computationally complex, operational rate-distortion methods [3, 4, 5] and analytical, model based approaches [7, 9, 8, 6]. In [3, 4] the average distortion of a sequence of intra coded frames is minimized subject to buffer and delay constraints using Vitterbi and Lagrange multiplier algorithms. In [5], the average distortion of motion-predicted frames is minimized using computationally complex, multi-pass Lagrange multiplier optimization and effective pruning techniques, without considering buffer and delay constraints. Also in [5], a multi-pass rate control algorithm for spatio temporal pyramid video coder is proposed that encodes base layer at a specified distortion, and the enhancement layer to satisfy total bit budget using Lagrange multiplier technique. A computationally simpler, model-based rate control scheme for scalable MPEG-2 encoding is described in [9]. The scheme in [9] dynamically adjusts the bitrates for each frame in each layer to keep the ratio of mean squared errors (MSEs) of each frame in the base layer and enhancement layer constant. The MSE in both layers is free to fluctuate to satisfy total bitrate constraints, but the ratio of MSEs is kept constant. The approach in [9] uses analytical models to predict the relationship between bitrates and MSE in both layers and to adjust the bitrates appropriately by means of a single pass algorithm. In [7] a single layer model-based bit allocation algorithm for minimizing the maximum block distortion over an entire encoded sequence subject to buffer constraints was proposed. Minimizing the maximum block distortion is shown in [7] to lead to constant quality encoding when buffer constraints allow it. The rate-distortion models for all frames in the sequence have to be constructed with a multi-pass algorithm for this approach. In [6] a single layer one-pass rate control scheme is proposed, that uses analytical model to reduce the variation of quality from frame to frame, while satisfying total bit budget with a single pass algorithm.

The problem of constant quality encoding for scalable video has not yet been addressed. In this paper we propose a novel scheme for constant quality scalable video encoding using matching pursuits (MP) codec [1], that satisfies bit budget constraint for each layer. We formulate the constant-quality rate control problem as zero-finding problem, and solve it with a multi-pass algorithm. The algorithm has to be multi-pass to achieve constant quality for each frame in the sequence, while satisfying total bitrate constraints for each layer. The multi-pass aspect of the algorithm precludes its uses for real-time applications. However, the ability to produce constant quality scalable video at desired total bitrates has many useful applications where real-time encoding is not necessary, such as playback. We evaluate the performance of the proposed algorithm using scalable MP (SCAMP) video codec described in [1] with 2 and 3 layers of scalable video. Our rate control scheme is also applicable to any number of layers of scalable MP video.

This paper is organized as follows: section 2 reviews scalable video coding using MP codec introduced in [1]. In section 3 we formulate the constant quality rate control problem as a zero finding problem, and present a general algorithm for adjusting the qualities of each layer in order to generate constant quality video at specified bitrates. The proposed rate control algorithm is evaluated using SCAMP codec in section 4. Finally, section 5 concludes the paper.

2. SNR SCALABLE MP VIDEO CODER

All of existing video compression standards have two main stages: 1. motion estimation and compensation, which predicts each frame from its neighboring frames, and produces the prediction error frame; 2. encoding of the prediction

This work was supported by AFOSR contract F49620-96-1-0199, Sun Microsystems, Philips, Hughes Research Laboratories, and California State Program MICRO.

error. All existing video compression standards, such as H.263+ and MPEG-4, use block-based discrete cosine transform (DCT) in the second stage. Although DCT coding is efficient, it introduces undesirable blocking, ringing and mosquito noise artifacts, especially at low bit rates. Neff and Zakhor [2] have recently applied the matching pursuit technique to code the motion prediction error signal with over-complete Gabor bases as dictionary. Video sequences coded using MP suffer from neither blocking nor ringing artifacts. In [1], MP codec described in [2] was extended to provide scalability feature, providing better compression efficiency than H.263+ codec as implemented by the University of Brittish Columbia codec group.

In this section we review one-residual and multi-residual MP-based scalability schemes that have been described in detail in [1]. In one-residual scheme, decoder uses previously reconstructed base layer frame in motion compensation loop for both base and enhancement layers, in order to avoid the drifting problem. Once residual image is found, a certain number of atoms is used to code the base layer and additional atoms are used to code the enhancement layers. This way the decoder can stop at any time after decoding the base layer information without losing track of the encoder. Moreover, if the enhancement layer atoms are coded few at a time as they are found, we can have a scalable coder with resolution of a few atoms, e.g. 100 bits per frame, which would correspond to having very "thin" enhancement layers. The disadvantage of the one-residual scheme is the large gap in PSNR between performances of scalable and non-scalable codecs for higher layers. The multi-residual scheme narrows this gap by forming motion compensated residual in layer i from previously reconstructed frame for layer i. In addition, at the encoder atoms are found using linear combination of residual error images for each layer, determined by weights $\alpha_1, \ldots \alpha_N$, where N is the total number of layers and $\alpha_1, \ldots \alpha_N$ are positive numbers that sum up to 1. Weights α's can be viewed as knobs allowing user to trade off qualities of the layers while keeping the bitrates of each layer fixed.

3. RATE CONTROL ALGORITHM

In this section we describe an algorithm for constant quality encoding of SNR scalable video for a given total bitrate for each layer. Our goal is to code each frame in layer i of the video sequence at some fixed quality q_i. In this work, we measure quality of an encoded frame by Peak Signal to Noise Ratio (PSNR) of its luminance component. The desired bitrates for each layer can be achieved by adjusting the quality of that layer through a zero finding problem.

Suppose we want to encode a sequence so as to produce N SNR scalable layers such that the cumulative bitrates for each layer over T seconds are given by vector $\mathbf{b}^* = [b_1^*, b_2^*, \ldots, b_N^*]$. Let $\mathbf{q} = [q_1, q_2, \ldots, q_N]$ be a vector of qualities for each layer. Let us model SNR scalable encoder that produces n layers of video such that the quality of layer i is q_i with cumulative bitrate b_i as some function $\mathcal{F}$. We can then write $\mathcal{F}(\mathbf{q}) = \mathbf{b}$ to mean that encoding at qualities specified by $\mathbf{q}$ produces cumulative bitrates $\mathbf{b}$. The goal of the quality adjusting algorithm is to find $\mathbf{q}^*$, such that $\mathcal{F}(\mathbf{q}^*) - \mathbf{b}^* = 0$.

The problem of numerically finding zeros of an n-dimensional function is in general a difficult one [10]. However, in our case, $\mathcal{F}$ has some useful properties. For the "one-residual" scalability scheme note that changes in q_i will have no effect on $b_1, b_2, \ldots, b_{i-1}$, because the encoder only keeps track of the base layer residual. In a multi-residual scheme, changes in q_i affect $b_1 \ldots b_N$, since atoms are found based on linear combination of the residual images for different layers. The way q_i's affect b_i's depends heavily on α_i - the weights used in the construction of linear combination of the residuals for different layers.

It is important to note that constant quality rate control in multi-residual scheme can not work for all possible weights α_i used in constructing linear combination of the residual images for different layers. To see the reason, consider the case $\alpha_1 = 0, \alpha_2 = 1$. In this case, layer 1 atoms are found on the residual image corresponding to layer 2. However, it might not be possible to reduce the energy of layer 1 residual to satisfy the constant quality criterion by finding atoms based on layer 2 residual, since the residuals for layers 1 and 2 are different. Therefore, in order to ensure the error of layer i is bounded by quality criterion specified by q_i, α_i must be large compared to $\alpha_{i+1} \ldots \alpha_N$. This is particularly true as q_i becomes smaller. In the experiments discussed in Section 4 we use $\alpha_1 = 0.75, \alpha_2 = 0.25$ for two layer runs, and $\alpha_1 = 0.75; \alpha_2 = 0.1875; \alpha_3 = 0.0625$ for three layer runs without running into any convergence problems. There are two side benefits to using diminishing α's. First of all, as shown in [1], the diminishing α's provide a reasonable tradeoff between the qualities of various layers. Secondly, diminishing α's imply that atoms for layers $1, 2, \ldots, i-1$ depend primarily on residuals for layers $1, 2, \ldots, i-1$ respectively, which means that changes in q_i will have little effect on $b_1, b_2, \ldots, b_{i-1}$. This fact is utilized in the algorithm described next.

We use a two stage algorithm to find $\mathbf{q}^*$. The first stage is designed to bring us close to the solution, while the second stage refines the estimate from the first stage using Newton's method [10] to find $\mathbf{q}^*$. We need the first stage because Netwon's algorithm by itself converges only when the initial estimate is in the neighborhood of the solution than the first stage by itself. We use Newton's stage because it provides faster convergence in the neighborhood of the solution. The second stage stops when we find $\mathbf{q}^{'}$ such that $(1 - Tol_2)\mathbf{b}^* < \mathcal{F}([\mathbf{q}^{'}]) < (1 + Tol_2)\mathbf{b}^*$, where Tol_2 stands for the tolerance of the second stage, and is a number between 0 and 1 specified by the user. The first stage takes advantage of the properties of $\mathcal{F}$ to find a good estimate for $\mathbf{q}^*$ that is then fed to the second stage.

Although finding zeros of an n-dimensional function is a difficult problem, finding zeros of a one-dimensional function can be done with a simple, robust algorithm. The idea behind the first stage is to solve n one-dimensional zero-finding problems in $q_1, q_2, \ldots, q_N$, to arrive at a close estimate of a zero of an n-dimensional function $\mathcal{F}$. We use bisection algorithm [10] in our first stage to find zeros of one-dimensional functions. We start out by searching for $q_1^{'}$ such that $(1 - Tol_1)b_1^* < \mathcal{F}([q_1^{'}]) < (1 + Tol_1)b_1^*$. Tol_1 stands for tolerance of the first stage, and is a number between 0 and 1 specificed by the user. This search involves multiple encoder runs with a single layer. We then perform a one-dimensional search in q_2, while keeping $q_1 = q_1^{'}$ to find $q_2^{'}$ such that the cumulative bitrate of the second layer is in $[(1 - Tol_1)b_2^*, (1 + Tol_1)b_2^*]$. This search involves multiple encoder runs with two layers where the quality of the first layer is fixed at $q_1^{'}$ and q_2 is adjusted to bring the bitrate of the second layer close to b_2^*. In the process of adjusting q_2 the bitrate of first layer might drift away from b_1^*, but the change in the bitrate of the first layer due to the addition of the second layer should be small, since, as discussed above, α_i's are decreasing with i. We then adjust $q_3, \ldots, q_N$, in the same way.

The second stage of the quality adjusting algorithm uses Newton's method [10] to refine $\mathbf{q}$ from the previous stage. The basic idea behind Newton's method is to use partial derivatives of $\mathcal{F}$ with respect to $\mathbf{q}$ to iteratively move towards zero of $\mathcal{F}(\mathbf{q}) - \mathbf{b}^*$. A Newton step consists of

1. Estimating $\mathbf{J}$, the Jacobian matrix of $\mathcal{F}$ with respect to $\mathbf{q}$.
2. Updating $\mathbf{q}$ using Newton's method and the estimate of $\mathbf{J}$.
3. Evaluating $\mathcal{F}(\mathbf{q})$.

If $\mathcal{F}(\mathbf{q}) < (1-Tol_2)\mathbf{b}^*$ or $\mathcal{F}(\mathbf{q}) > (1+Tol_2)\mathbf{b}^*$, then another Newton step is performed until we find $\mathbf{q}$ such that $(1 - Tol_2)\mathbf{b}^* < \mathcal{F}(\mathbf{q}) < (1 + Tol_2)\mathbf{b}^*$.

Since $\mathcal{F}$ does not have an analytical expression, we estimate $\mathbf{J}$ numerically for each Newton step. Let $\mathcal{F}(\hat{\mathbf{q}}) = \hat{\mathbf{b}}$ and suppose we want to estimate $\mathbf{J}$ at $(\hat{\mathbf{q}}, \hat{\mathbf{b}})$. Estimating $\mathbf{J}$ in each Newton step takes N iterations for N layers, where by iteration we mean one encoding of entire video sequence, or, in other words, a single evaluation of $\mathcal{F}$. In the first iteration we perturb the quality of layer 1 and find the resulting bitrates for each layer. This allows us to calculate partial derivatives of the bitrates of every layer with respect to the quality parameter of the first layer. In the same fashion we then collect partial derivatives with respect to the quality parameters of layers $2, \ldots, N$, thus arriving at an estimate of $\mathbf{J}$. More formally, let $\mathbf{q}^k$, defined as $\mathbf{q}^k = [\hat{q}_1, \hat{q_2}, \ldots, \hat{q}_{k-1}, (1+\beta)\hat{q}_k, \hat{q}_{k+1}, \ldots \hat{q}_N]$, be a vector of quality parameters for k^{th} iteration of Jacobian estimation procedure, and let $\mathbf{b}^k = \mathcal{F}(\mathbf{q}^k)$ be the resulting bitrates. Then, $\mathbf{J}_{mk} = \frac{b_m^k - \hat{b}_m}{(1+\beta)\hat{q}_k - \hat{q}_k} = \frac{b_m^k - \hat{b}_m}{\beta \hat{q}_k}$. Note that the magnitude of perturbation is proportional to the magnitude of the quality parameter. Experimentally, $\beta = 0.05$ was found to give robust estimates of $\mathbf{J}$, and is used in the experiments described in section 4. Smaller values of β lead to noisier estimates, while larger values of β give coarser approximations of partial derivatives.

To summarize, for a sequence with N layers, estimating Jacobian takes N iterations. Therefore, one Newton step takes $N + 1$ iterations, since after estimating Jacobian, we need to update $\mathbf{q}$ and evaluate $\mathcal{F}$ at the updated $\mathbf{q}$.

4. RESULTS

Tol_1 and Tol_2 are set to 0.1 and 0.015 respectively in all the experiments described in this section. Intra frames are coded using TMN.H263+ advanced mode with AC/DC prediction. Macroblock level rate control is used to produce desired PSNR for intra frames. The decision to code a frame as intra frame is based on three factors: 1) *cvar* - the variance of pixels on the current frame, 2) *dvar* - the variance of pixels on the difference of current and previous frames, 3) *intra_mb* - the percentage of intra macroblocks on the current frame. A frame is coded as intra if $0.8 * dvar > cvar$ or $intra\_mb > 0.2$.

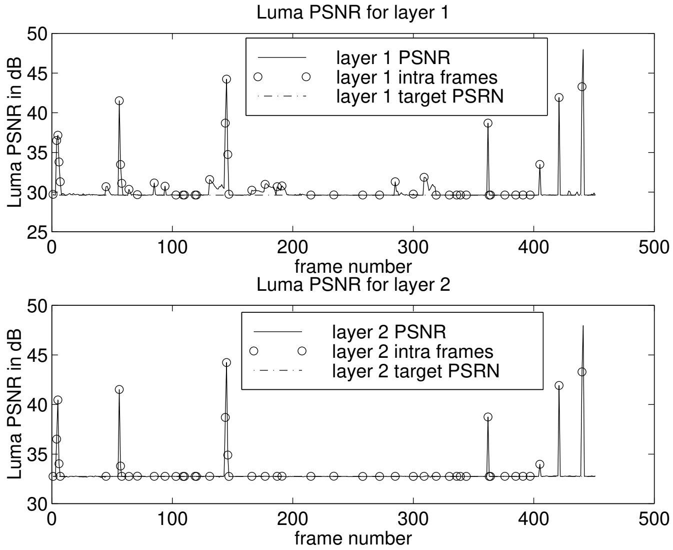


Figure 1: Final Luma PSNR as a function of frame number for Preacher Trailer 2 sequence coded with multi-residual scheme at 24 and 48 Kbits/s.

Table 1 presents the number of iterations for stages 1 and 2 and average PSNR's for various bitrates and sequences. The duration of PS1, Head, Preacher 1 and Preacher 2 sequences is 1 minute; all other sequences are 10 seconds long. The frame rate is 10 and 15 frames per second (fps) for sequences coded at (48,112) and (112,256) Kbits/s respectively. Frame rate is 7.5 fps for sequences coded at all other rates. Container, Hall, Mother, Coast and Foreman sequences have only one intra frame, while PS1, Head, Preacher1 and Preacher2 sequences have 7, 18, 61, and 49 intra frames respectively, detected with the scene change algorithm described earlier. The proposed algorithm performs robustly and converges to the desired values of quality parameters under a variety of conditions. The difference between maximum and minimum luminance PSNR over all frames in each sequence in each layer is shown in columns 8 and 14 of Table 1. As seen, for most sequences this difference is about 0.1dB, and thus PSNR of all frames in a sequence is close to the average PSNR. The reason for large PSNR variations for Coast, PS1, Foreman, Head, Preacher 1 and 2 sequences is the fact that quantization parameter for intra frames can not exceed 31 - so that PSNR for some near-black, low-information intra frames is much higher than the average PSNR.

Figure 1 shows Luma PSNR for each frame of Preacher 2 sequence coded with multi-residual scheme at 24 and 48Kbps. The spikes in PSNR are produced by low-information, easy to code intra frames encoded with quantization parameter equal to 31, and the low-information inter frames that

Table 1: The average PSNR and number of coder iterations for various sequences at different bit rates with different number of layers for SCAMP codec using one-residual and multi-residual schemes. Bits used to code every layer of every sequence are within 1.5% of the desired bit budget.

Sequence	Bitrate in Kbits/s	One-residual scheme						Multi-residual scheme					
		Iterations per Stage		PSNR (dB)				Iterations per Stage		PSNR (dB)			
				Average			Max-Min			Average			Max-Min
		1	2	Y	Cr	Cb		1	2	Y	Cr	Cb	
Container	10K	7	3	31.30	37.01	36.84	0.07	5	3	31.04	36.75	36.64	0.07
	24K			32.02	37.73	37.52	0.07			34.07	39.53	39.57	0.07
Hall	10K	4	9	31.86	36.28	38.81	0.09	2	12	31.41	35.98	38.38	0.09
	24K			32.78	37.28	39.04	0.08			35.60	39.32	40.74	0.16
Mother	10K	9	6	32.79	37.95	39.07	0.06	7	6	32.60	37.76	38.96	0.09
	24K			33.74	38.65	39.54	0.05			35.36	40.64	41.33	0.11
Container	10K	10	4	31.30	37.01	36.84	0.07	10	4	30.96	36.73	36.70	0.06
	24K			32.02	37.73	37.52	0.07			33.80	39.06	39.12	0.07
	48K			32.97	39.06	38.73	0.05			35.81	41.61	41.70	0.07
Hall	10K	8	16	31.90	36.41	38.82	0.10	4	8	31.36	35.77	38.36	0.09
	24K			32.84	37.34	39.08	0.09			35.33	38.97	40.55	0.15
	48K			33.94	38.68	39.98	0.10			37.12	39.48	41.55	0.16
Mother	10K	13	8	32.79	37.91	38.95	0.08	10	4	32.55	37.92	38.83	0.07
	24K			33.74	38.67	39.46	0.04			35.10	40.03	40.81	0.09
	48K			34.92	40.55	40.92	0.04			37.49	42.70	43.28	0.10
Coast	48K	6	3	29.87	39.32	41.06	0.03	7	3	29.71	39.15	41.00	0.04
	112K			31.55	41.04	42.59	0.03			32.23	41.94	43.45	0.03
Coast	112K	7	3	27.14	37.21	39.11	0.70	8	3	26.99	36.82	38.83	0.85
	256K			28.58	38.40	39.44	0.02			29.30	38.86	40.46	0.02
Foreman	48K	6	3	31.00	37.68	37.93	0.06	7	3	30.82	37.29	37.72	0.05
	112K			32.98	39.43	39.87	0.03			34.33	40.68	41.40	0.04
Foreman	112K	4	3	28.68	35.01	35.29	3.58	3	3	28.58	34.67	34.84	3.68
	256K			30.13	36.42	36.71	2.10			31.73	37.86	38.54	0.47
PS1	10K	9	0	32.81	37.25	38.00	0.26	6	3	32.68	37.21	37.90	0.34
	20K			33.41	37.69	38.46	0.08			34.64	39.06	39.63	0.20
Headline News	24K	7	3	28.91	35.75	37.99	24.92	6	3	28.75	35.52	37.81	25.10
	48K			29.97	36.55	38.53	23.78			31.28	37.72	39.36	22.42
Preacher Trailer 1	24K	7	3	30.53	38.84	38.78	15.62	5	0	30.52	38.80	38.69	15.62
	48K			32.25	39.84	40.00	13.70			33.19	41.21	41.17	13.95
Preacher Trailer 2	24K	5	3	30.11	39.70	39.38	18.30	6	3	30.06	39.63	39.29	18.38
	48K			31.82	40.39	40.43	16.46			32.94	41.77	41.50	15.31

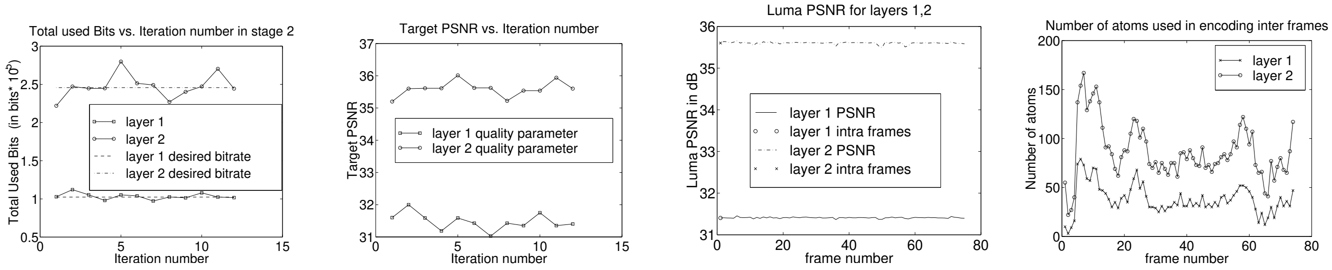
follow. Disregarding the degenerate case of low-information frames, the proposed algorithm consistently produces constant PSNR, scalable bit stream at desired bitrates.

The number of iterations per layer to achieve desired bitrates in Table 1 is between 2.5 and 8, with the mean at 5. Note that the sum of iterations in the first and second stages corresponds to the total number of coder invocations for the entire video sequence needed to achieve desired bitrates at constant quality. [1] The total number of iterations per layer, averaged over all 14 sequences and both scalability schemes, as a function of Tol_2 is shown in Figure 2. As expected, bit budget constraints with larger tolerance are satisfied with fewer iterations.

Figures 3 (a,b) shows total used bits and target qualities for each iteration of the second stage for layers 1 and 2 of Hall sequence coded at 10 and 24 Kbps using multi-residual scheme. It can be seen that the qualities in each layer are adjusted in such a way that the total used bits approach the desired bit budgets. From figure 3 (a) it may seem that in the third iteration the total bitrates were close enough to the desired bit budget. However, closer examination reveals that the total bits used in the first layer were 2.9% greater than the desired bit budget for that iteration. It took 12 iterations, or, equivalently, 3 Newton steps to bring total used bits within $Tol_2 = 1.5\%$ of the desired bit budgets. Figures 3(c,d) show Luma PSNR and number of atoms respectively for each frame of the same Hall sequence for layers 1 and 2. It is clear that by varying the number of atoms spent on each frame, the PSNR of the Luma component is brought very closely to the target qualities for layers 1 and 2.

[1] The number of Newton steps is the number of iterations for the second stage reported in Table 1, divided by $N+1$, where N is the number of layers.

Figure 3: Hall sequence coded with multi-residual scheme at 10 and 24 Kbits/s. (a) Total used bits in each layer as a function of iteration number in stage 2 for Hall sequence. (b) Target quality parameters for each layer as a function of iteration number for stage 2 for Hall sequence. (c) Final Luma PSNR as a function of frame number for Hall sequence. (d) Number of atoms used to code every inter frame.



(a) (b) (c) (d)

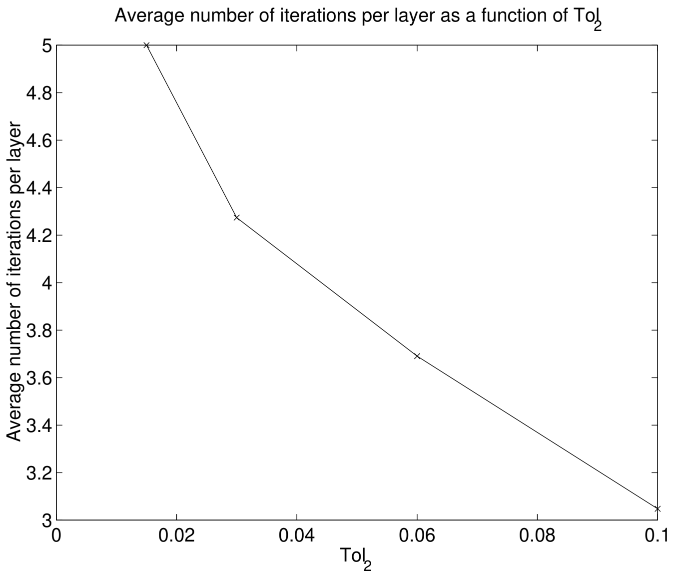


Figure 2: Average number of iterations per layer as a function of Tol_2.

5. CONCLUSIONS

In this work, we demonstrated a non-causal, constant quality rate control algorithm for SNR scalable MP codec, that adjusts the qualities of the layers to match desired bitrates. The proposed algorithm performs robustly for both single residual and multi-residual scalability schemes, and converges to the desired bitrates in as few as 2.5 iterations per layer.

6. REFERENCES

[1] O. Al-Shaykh, E. Miloslavsky, T. Nomura, R. Neff, A. Zakhor, "Video Compression Using Matching Pursuits", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol 9, Number 1, pp. 123-143, February 1999.

[2] R. Neff and A. Zakhor, "Very low bit rate video coding based on matching pursuits", *IEEE CSVT*, Vol.7, No.1, pp. 158-171, February, 1997.

[3] C.Y. Hsu, A. Ortega, A. Reibman, "Joint selection of source and channel rate for VBR video transmission under ATM policing constraints", *IEEE Journal on Selected Areas in Communications*, Vol.15, pp. 1016-28, Aug. 1997.

[4] D. Lin, "Jointly optimal video coding and rate control for VBR transmission", *Proceedings of IEEE International Conference on Image Processing*, Vol.3, pp. 939-943, 1998.

[5] K. Ramchandran, A. Ortega, M. Vetterli, "Bit allocation for dependent quantization with applications to multiresolution and MPEG video coders", *IEEE Transactions on Image Processing*, Vol.3, pp. 533-45, September 1994.

[6] L. Lin, A. Ortega, "Perceptually based video rate control using pre-filtering and predicted rate-distortion characteristics", *Proceedings of IEEE International Conference on Image Processing*, Vol.2, pp. 57-60, 1997.

[7] D. Hsu, J. Vitter, E. Linzer, "Lexicographic bit allocation for MPEG Video Coding", *Proceedings of IEEE International Conference on Image Processing*, Vol.1, pp. 322-326, 1997.

[8] U. Horn, T. Wiegand, B. Girod, "Bit allocation methods for closed-loop coding of oversampled pyramid decomposition", *Proceedings of IEEE International Conference on Image Processing*, Vol.2, pp. 17-20, 1997.

[9] S.K. Kwon, J.K. Kim, "Adaptive Bitrate Allocation in Spatial Scalable Video Coding of Fixed Total Bitrate", *IEICE Trans. on Fundamentals of Electr., Communicat. and CS*, Vol. E81-A.5, pp. 950-6, May 1998.

[10] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, "Numerical Recipes in C", *Cambridge University press*, Second Edition, Chapter 9, 1994.