INTERNATIONAL ORGANISATION FOR STANDARDISATION
ORGANISATION INTERNATIONALE DE NORMALISATION
ISO/IEC JTC1/SC29/WG11
CODING OF MOVING PICTURES AND AUDIO

ISO/IEC JTC1/SC29/WG11
MPEG 98/M3833
July 1998

Source: University of California at Berkeley
Purpose: Information and Discussion
Title: SNR scalability using Matching Pursuits
Authors: Eugene Miloslavsky, Sen-ching Samson Cheung, Avideh Zakhor[1]

This document describes our work in texture SNR scalability using Matching Pursuits. Matching Pursuit coding scheme compresses motion compensated residual signal into "atoms" which are selected from a large redundant library of functions. It is currently being considered for the MPEG-4 Visual Standard Version 2. Visual simulation results of our scalable codecs have been presented in the San Jose meeting and will again be shown in Dublin. More details regarding this work can be found in [1].

1. Introduction

Scalability video compression schemes can be broadly classified into two categories: (1) coarse grain scalability with few widely apart available bit rates; (2) fine grain scalability with a continuum of available bit rates. Temporal scalability and spatial scalability tools defined in MPEG-4 Visual Standard are examples of the first class and the video codec developed by Taubman and Zakhor [2] is an example of the second class. However for a Matching Pursuits based codec, a natural way of achieving both fine and coarse scalability is through the number of atoms. In section 2, we will propose a multiple residual image compression system which provides good coding performance and coarse scalability. Visual results will also be presented in the accompanying D1 tape. In section 3, we will examine a one-residual image system offering fine grain scalability at the expense of poorer coding performance.

2. Scalable Matching Pursuit Codec using multiple residual images

The structure of the two-layer scalable codec with multiple residual images are shown in Figure 1. This structure can be easily extended to more than two layers. Nontheless, we will stay with this two-layer structure to simplify the descriptions.

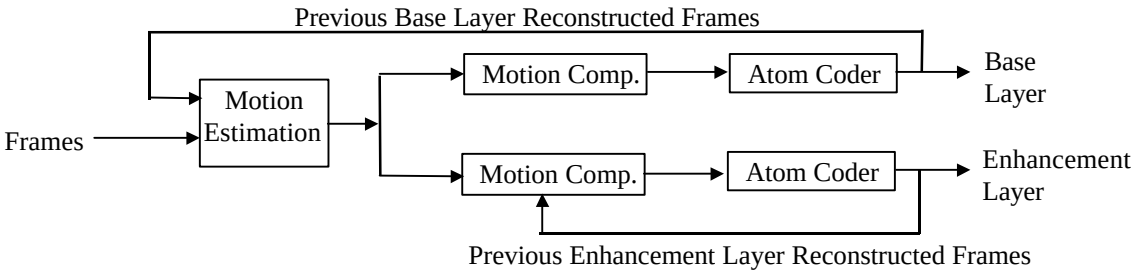


Figure 1: Block diagram for two-residual sclable encoder

[1] For more information on this document, please contact Dr. Avideh Zakhor at avz@eecs.berkeley.edu.

Motion compensation is applied to both layers. Motion estimation is done based on the base layer reconstructed frames and the same set of motion vectors are applied to both layers. Since the enhancement layer will use the atoms of the base layer, the choice of the base layer atoms will affect the qualities of both layers. In the remainder of this section, we will describe a way of adjusting the quality between base and enhancement layers while keeping their bit rates fixed.

Our approach is to (a) consider both the base and enhancement layer residuals when finding the atoms that belong to both layers; and (b) only consider the enhancement residual when finding the remaining atoms that only belong to the enhancement layer. One way to accomplish (a) is to minimize a weighted sum of the error of the base layer and the enhancement layer for finding each atom. That is, instead of focusing just on the base layer residue R_b, atoms are found with respect to $\alpha_1 R_b + \alpha_2 R_e$, where R_e is the enhancement layer residue. Without loss of generality, we assume that α_1 and α_2 are positive and they sum to 1. An important characteristics of this structure is that the enhancement layer decoder only needs to keep track of one motion loop and has no knowledge of what α_1 and α_2 are.

Figure 2 shows the effect of α_1 and α_2 on five different atom allocations between the base and enhancement layers when we apply the scalable codec to compress 10 seconds of "Container" at 7.5 fps with the total bitrate of 27 kbps. As the value of α_2 increases from 0 to 1, we find that: (a) the PSNR of the base layer decreases by 2 to 3 dB depending on atom allocation, and (b) the PSNR of the enhancement layer increases by 0.1 to 0.5 dB depending on the relative number of atoms between base and enhancement layers. Note that the PSNR change in the enhancement layer is considerably smaller than that of the base layer as α_2 changes from 0 to 1. As seen, for a given rate for the enhancement and base layers, one can modulate the relative PSNR performances of the two layers by choosing the appropriate value of α_1 and α_2.

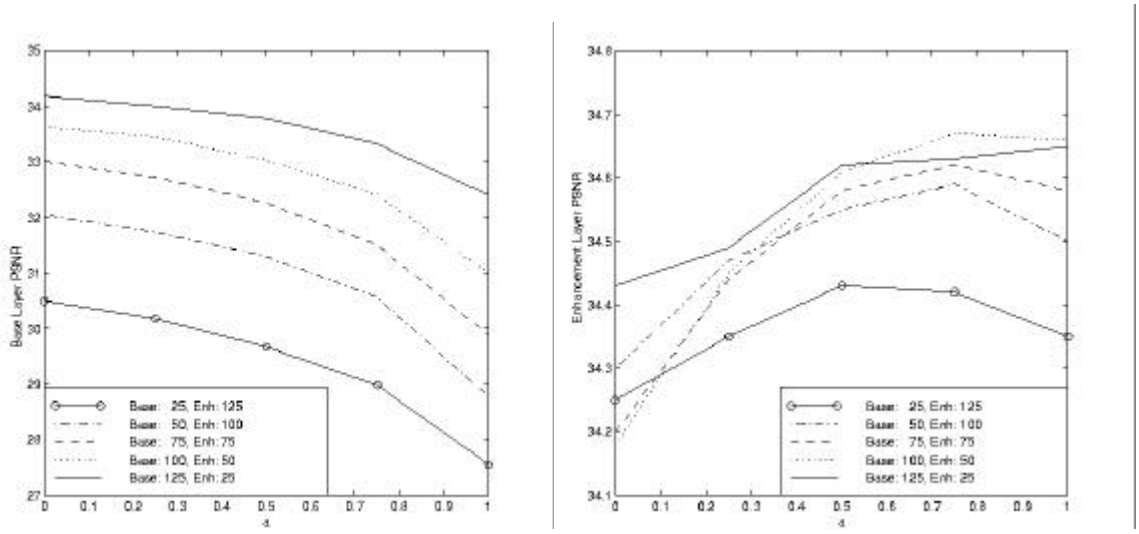


Figure 2: Variations of the average PSNR for the base (left) and enhancement layer (right) based on a = $\alpha_2 = 1 - \alpha_1$. The legend describes the atom distribution between both layers.

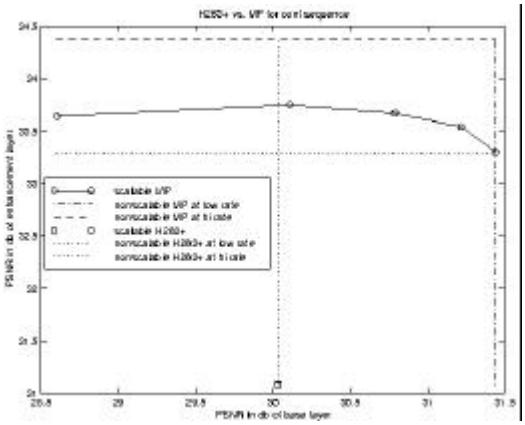
An interesting conclusion to be drawn from Figure 2 is that if one is considerably more concerned with the quality of the enhancement layer than that of the base layer, then $\alpha_2 = 1$ is a better choice than $\alpha_2 = 0$. On the other hand, in applications where the base and enhancement layers are equally important, it is more reasonable to operate at $\alpha_2 = 0$ rather than $\alpha_2 = 1$. It is also interesting to compare the PSNR and bit rate performance of the enhancement layer to the case where no scalability was applied. The non-

scalable codec with 150 atoms achieves PSNR of 34.65 dB and bit rate of 26.1 kbps. Comparing this with the results in Figure 2, the enhancement layer of the scalable coder has at worst 0.5 dB lower PSNR performance at 0.9 kbps higher bit rate. Another observation is that the PSNR of the enhancement layer increases up to some a_2 and then starts decreasing. This is because higher a_2 results in degrading the quality of the base layer, which is used for motion estimation. This results in less efficient motion estimation, reducing the qualities of both the base layer and enhancement layer reconstructed frames. One remedy is to use a weighted combination of the reconstructed frames from both layers to form a reference frame for motion estimation. For the results presented in the section 2.2, the same weights are used in motion estimation as well as atom derivations.

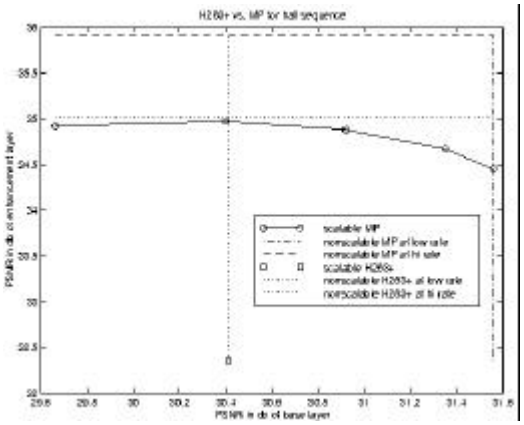
To summarize, the values of a_1 and a_2 can be thought as a knob that controls the quality of the resulting images without affecting the bit rate for each layer.

2.1 Comparison with H.263 version 2 codec

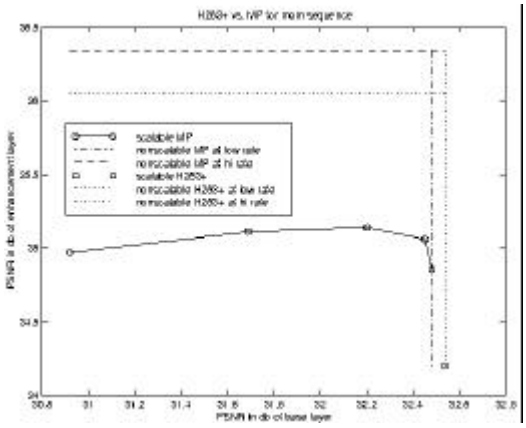
In this section we compare the performances of the scalable MP codec and DCT-based H.263 version 2 scalable codec. There are a number of important differences between the two. First, in our proposed MP codec, the same set of motion vectors is used for the base and enhancement layers, whereas in H.263v2 two sets of motion vectors are used. Second, unlike H.263v2 where enhancement layer frames are predicted bi-directionally from previous enhancement layer frame and current base layer reference frame, the MP enhancement layer frames are only predicted from previous enhancement layer frame. In our comparisons we have used publicly available version 3.1.2 implementation of H.263v2 standard from the University of British Columbia [3].



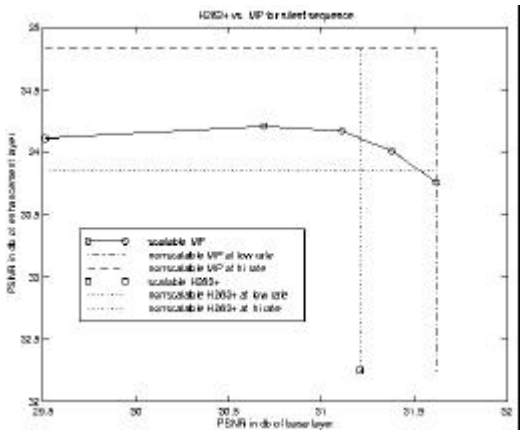
(a) Container, 7.5fps



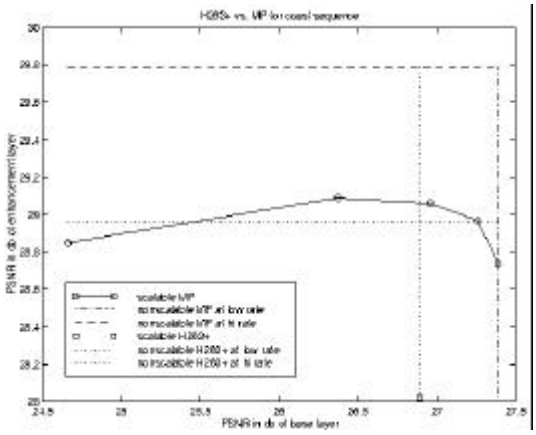
(b) Hall Monitor, 7.5 fps



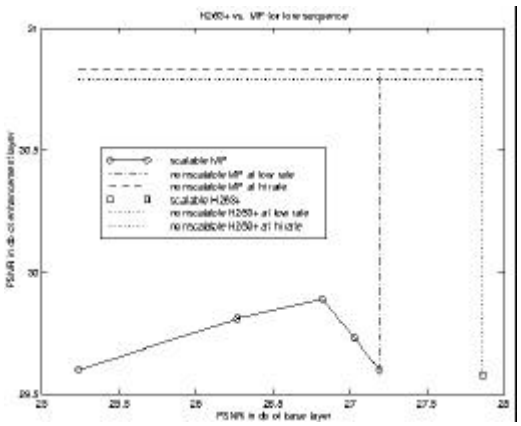
(c) Mother and Daughter, 10 fps



(d) Silent Voice, 10 fps



(e) Coastguard, 10 fps



(f) Foreman, 10 fps

Figure 3: Comparison of H.263v2 (H.263+) and MP codecs in scalable and non-scalable modes

Figure 3 shows the plots of base and enhancement layers' PSNRs for MP and H.263v2 codecs for six different sequences. The circles correspond to PSNRs of scalable MP with values of a_1 varying from 0 to 1 in increments of 0.25. In all comparisons, rate control is determined by running H.263v2 codec with a fixed quantization step size for all frames in the sequence. MP codec uses the same intra frames as H.263v2 codec, and the same number of bits for both base and enhancement layer of each frame up to precision of about 50 bits. Since the quantization step size in H.263v2 codec only takes integer values from 1 to 31, the total bit rates of scalable and non scalable H.263v2 runs are slightly different. Non scalable MP runs are based on the first frames and bit rates generated by non scalable H.263v2 runs, with the bits spent on all but the first frame prorated in such a way that they add up to the total bit rate of the enhancement layer for the scalable run of H.263v2. This way scalable MP, scalable H.263v2 and non scalable MP runs use the same number of bits, with non scalable H.263v2 runs producing slightly different bit rates.

Various parameters used to generate the results in Figure 3 are summarized below:

Sequences	**Container**	**Hall**	**Mom**	**Silent**	**Coastguard**	**Foreman**
Base layer QP	16	17	14	13	22	24
Enh. Layer QP	12	10	9	9	17	16
Non scalable coder QP	9	8	7	8	14	13
Framerate (fps)	7.5	7.5	7.5	10	10	10
Base layer total bit rates and MP non scalable bit rates (kbps)	10.63	10.14	9.29	22.96	24.46	24.81
Enh. Layer total bit rates and MP non scalable bit rates (kbps)	22.76	23.14	23.50	44.82	47.77	46.68
Bit rates (kbps) for H.263v2 non scalable coder	22.25	24.32	25.21	41.42	46.62	48.60

From Figure 3 we see that for most sequences, if we keep the base layer PSNR for MP and H.263v2 identical by exploiting alpha factor, MP outperforms H.263+ by 0.5dB to 2.5dB at the enhancement layer.

This is true for all sequences except for Foreman and Mother-Daughter sequences where MP does worse at the base layer. We also see that the difference between performances of MP scalable and non scalable coders for the enhancement layers between 0.63 and 1.46 dB, depending on the sequence and values of a_1 and a_2, while for H.263v2 codec the gap ranges from 0.84 to 2.65 dB. Also, with the exception of Foreman and Mother Daughter sequences, MP non scalable codec outperforms H.263v2 non scalable codec at both bit rates. MP codec also allows to flexibly trade-off PSNR of the base and enhancement layers, as discussed before, providing a continuum of operating points. As a_1 changes from 0 to 1, the change in PSNR is much larger in the base layer than in the enhancement layer. So, from a practical point of view, a_1 = 0.75 represents a good compromise between the qualities of base and enhancement layers.

2.2 Visual Results

A series of experiments are performed by using our scalable codec to compress a number of MPEG test sequences. Two-layered codec is used for all runs and three-layered codec is also used for some sequences. All the simulations are listed as follows:

Sequences	Frame rate, fps	Bitrates, kbps (base/enh1/enh2)
Container, QCIF	7.5	10/24, 10/24/48
Hall, QCIF	7.5	10/24, 10/24/48
Mother & Daughter, QCIF	7.5	10/24, 10/24/48
Container, QCIF	10	24/48
Silent Voice, QCIF	10	24/48
Mother & Daughter, QCIF	10	24/48
Coastguard, QCIF	10	48/112
Foreman, QCIF	10	48/112
News, CIF	7.5	48/112
Coastguard, CIF	15	112/256
Foreman, CIF	15	112/256
News, CIF	15	112/256

All the layers for the first frame are coded using H.263 version 2 with Annex I (Advanced Intra Coding), Annex J (Deblocking Filter) and Annex O (SNR scalability) options. For the rest of the sequence, the base layer is coded using the Matching Pursuit syntax described in MPEG4 VM V9.0 (N1869). As for the enhancement layers, since the motion vectors and mode information are already included in the base layer, the combined motion-texture mode in VM V9.0 cannot be used. Thus we have reverted back to an earlier version of the syntax described in VM V8.0 (N1796) where the atom information are coded independently of the macroblock information.

For the two-layered codec, the weights assigned for the base layer and the enhancement layer are 0.75 and 0.25 respectively. For the three-layered one, the weights are 0.75, 0.1875, 0.0625, starting from that of the base layer. As for the rate control, we have designed our codec to mimic the constant quantization method used in DCT. The method to achieve that is to stop the atom search when the maximum residual energy of all the macroblocks falls below a specific constant Q. Q is chosen for each layer in each sequence to achieve the target bit rates within two percents. The results are compiled in the following tables:

Table 1 : Simulation Results for Two-Layered Codec

Seq.	Base Layer				Enhancement Layer			
	Bits	Y-PSNR	U-PSNR	V-PSNR	Bits[2]	Y-PSNR	U-PSNR	V-PSNR
Cont10_24	102434	31.152	37.596	37.505	141036	33.886	39.562	39.498
Hall10_24	102280	31.995	36.523	39.092	143017	35.361	39.136	40.715
Mom10_24	103264	33.136	39.459	40.261	143297	35.602	41.059	41.903
Cont24_48	243046	33.790	39.703	39.570	244653	35.565	41.283	41.131
Silent24_48	246046	32.013	36.585	37.929	247869	34.597	39.030	39.685
Mom24_48	243201	35.912	41.553	42.248	246915	37.763	42.666	43.468
Coast48_112	493888	29.749	39.660	41.266	659562	32.220	41.707	42.668
Fore48_112	494639	30.892	37.642	38.067	658284	34.398	40.594	41.430
News48_112	488735	32.658	37.468	38.403	657721	36.203	40.576	41.237
Coast112_256	1147403	27.059	37.664	39.491	1470729	29.504	39.465	40.293
Fore112_256	1140382	28.546	35.128	35.512	1477702	32.182	38.222	39.260
News112_256	1144221	35.376	39.935	40.550	1473948	38.238	42.089	42.885

Table 2: Simulation Results for Three-Layered Codec

Sequences	Cont10_24_48	Hall10_24_48	Mom10_24_48
Base Layer			
Bits	103264	101718	102956
Y-PSNR	31.253	31.950	33.084
U-PSNR	37.599	36.637	39.524
V-PSNR	37.472	39.217	40.328
Bits[2]	143913	142513	141734
Enhacement Layer 1			
Y-PSNR	33.773	35.100	35.471
U-PSNR	39.190	38.937	40.735
V-PSNR	39.290	40.425	41.501
Enhancement Layer 2			
Bits[2]	243581	246746	247321
Y-PSNR	35.692	36.922	37.528
U-PSNR	41.690	39.433	42.657
V-PSNR	41.419	41.245	43.362

3. Scalable Matching Pursuit codec with single residual image

Figure 4 illustrates our basic approach to SNR scalability in the "one residual image" scheme. As seen, the motion compensation residual image is formed from the previously reconstructed base layer frame in order to avoid the drifting problem. Furthermore, the encoder keeps track of only one residual image namely the one corresponding to the base layer. Once the 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 layer. This way the decoder can stop at any time after decoding the base layer information without losing track of the

[2] Differential Layer only.

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. In the next section we will describe a practical atom position coding method that greatly improves coding of small groups of atoms and present performance results of a fine grain SNR scalable MP based codec with one residual image.

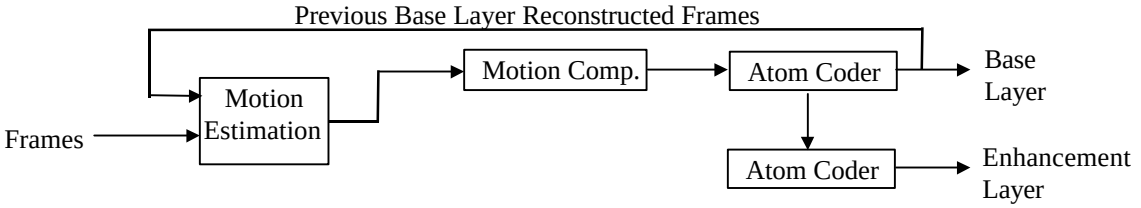


Figure 4: Block Diagram for one-residual scalable encoder

3.1 NumberSplit: A method for coding the position of atoms

The NumberSplit algorithm, used for coding atom positions for a small number of atoms at a time, is based on divide and conquer idea. First, total number of atoms, *T*, coded on a given residual image is transmitted in the header. Then the image is divided in two halves along a larger dimension and the number of atoms in the left or top half (depending on how the image was split) is coded. Note that if we assume that each atom falls uniformly and independently of other atoms onto either half then the number of atoms in the first half is binomially distributed on $\{0,\ldots,T\}$ with p = 0.5. Since we know the total number of atoms on an image and the distribution of the number of atoms in the first half, we can construct a Huffman table to encode the number of atoms in the first half. The total number of atoms and the number of atoms on the first half allows decoder to calculate the number of atoms in the second half. This algorithm is then applied recursively to the halves of the image until there are no more atoms in a given half image or until the size of the half image is one pixel. The Huffman tables used in encoding are built dynamically, depending on the number of atoms to be coded, which allows NumberSplit method to avoid inefficiencies of fixed tables at a cost of more computation.

In real residual images, atoms are placed at the locations where motion estimation is ineffective. For this reason atoms are not distributed uniformly and independently on image - they tend to "cluster" around the regions of high residual error. One heuristic way to tune the NumberSplit algorithm to the real life images is to modify the binomial distribution to account for clustering. It is easy to see that if atoms tend to cluster, the probability of many atoms being in the same half of image is higher than if atoms are independent and identically distributed. To account for this we emphasize the tails of binomial distribution in the following way: all probabilities of splits that are smaller than some fraction *f* of the maximum probability in the distribution are set to $f \times$ maximum probability, followed by re-normalization of the distribution The value of $f = 0.2$ was used in the simulations to explore trade-off between rate granularity and compression efficiency for scalable video. The following table demonstrates that this method is more efficient than the fixed Huffman table method used in the VM when the size of the atom group becomes smaller.

Table 3 : Bit rates usingNumberSplit and fixed Huffman tables for sending 50 atoms in base layer and 100 atoms in enhancement layer in groups of various sizes for container sequence at 7.5 fps

Size of the groups (atoms)	Bit rates (kbps) using NumberSplit	Bit rates (kbps) using fixed Huffman tables	NumberSplit gain (in %)
5	30.16	33.39	10.70
10	29.56	31.83	7.67
20	28.90	30.34	4.99
50	27.96	28.80	3.02
100	27.19	27.78	2.15

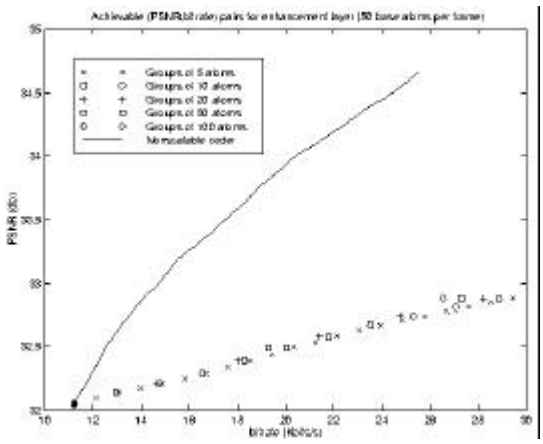
3.2 Fine Scalability: granularity vs. coding efficiency

We have developed a finely scalable codec based on the approach shown in Figure 4 in which the enhancement layer atoms are coded in groups of *N* atoms at a time, where *N* can range from 5 to 100. We use NumberSplit algorithm for efficient position coding of each group of *N* atoms in the enhancement layer. Figure 5 shows the average PSNR versus bit rate characteristics of the above scalable codec based on the NumberSplit position coding, for 10 seconds of the Container sequence coded at 7.5 frames per second. Three different allocations of atoms between layers have been used: (50,100), (75,75) and (100,50) atoms coded in base and enhancement layers respectively on each frame. In all three cases the total number of atoms is 150.

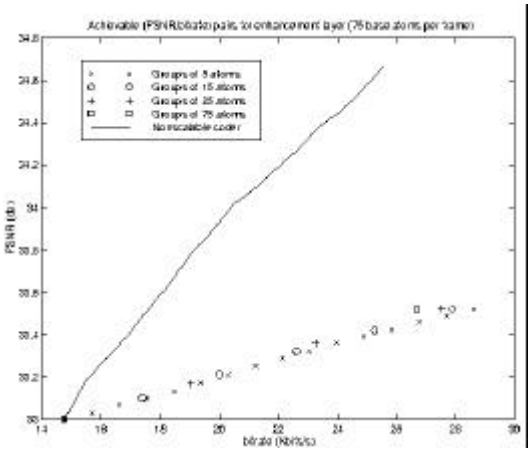
Several observations can also be made from the results in Figure 5: First, as the group size is increased, the bit rate required to achieve the same PSNR value drops. In fact, for the case where there are 50 atoms in the base layer, using group size of 5 atoms instead of 100 atoms produces a 10% increase in bit rate for the full enhancement layer, but allows for 900 bits/s levels of granularity.

The second observation to be made from Figure 5 is that the PSNR of the enhancement layer improves as the relative number of the atoms in the enhancement layer to base layer decreases. This happens because more bits are being allocated to the base layer, so that the images in the prediction loop are of better quality, and enhancement layer frames encode much less important structural information. These enhancement layer frames can be coded better with fewer bits than the enhancement layer frames corresponding to a smaller base layer and coded with more bits. The relationship between quality of a base layer and of the corresponding enhancement layer is non linear.

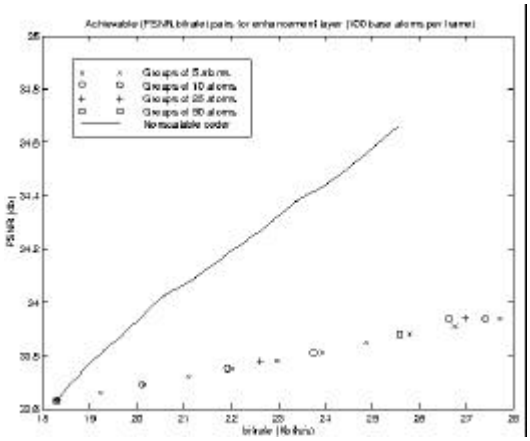
Finally, comparing the PSNR after the enhancement layer with that of a non scalable coder, the loss is between 0.72 and 1.78 depending on the atom allocation. This is mainly due to the fact that refinements produced by atoms in the enhancement layer are not propagated to the next frame via motion compensation. This is in contrast with the results obtained in section 1 where there is a much smaller gap between the performances of the scalable and non-scalable codecs.



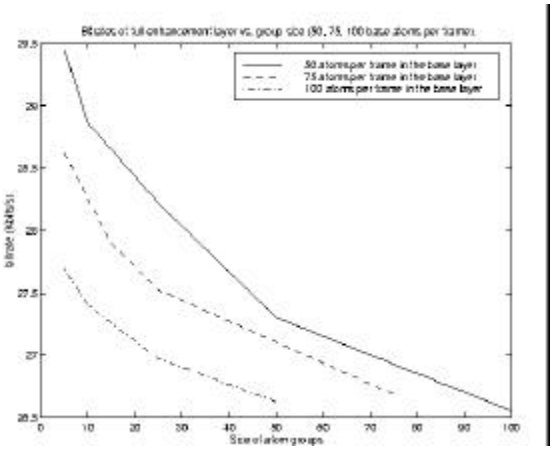
(a) 50 Base atoms per frame



(b) 75 Base atoms per frame



(c) 100 Base atoms per frame



(d) Bit rates vs Group size

Figure 5 : Achievable (PSNR, bitrate) points for enhancement layer as a function of size of groups in which atoms are coded for (a) 50 atoms coded at the base layer, (b) 75 atoms coded at the base layer, (c) 100 atoms coded at the base layer, (d) bit rates for the full enhancement layer as a function of the size of the coding groups

4. References

1. O. Al-Shaykh, E. Miloslavsky, T. Nomura, R. Neff, and A.Zakhor, "Video Compression Using Matching Pursuits," *IEEE Trans. Circuits and Systems for Video Technology*, submitted, October, 1997.
2. D. Taubman and A. Zakhor, "Mult-Rat 3-D Subband Coding of Video," *IEEE Transaction on Image Processing*, Vil. 3, No. 5, pp. 572-588, Sept. 1994.
3. University of British Columbia, H.263v2 TMN software, *ftp://monet.ee.ubc.ca/pub/tmn/ver-3.1.2*