REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	5/31/97	Annual Performance 5/1/96 - 4/30/97

4. TITLE AND SUBTITLE
Video Compression Algorithms for Transmission and Video

5. FUNDING NUMBERS
N00014-92-J-1732

6. AUTHOR(S)
Avideh Zakhor

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Regents of the University of California
c/o Sponsored Projects Office
336 Sproul Hall
Berkeley, CA 94720-5940

8. PERFORMING ORGANIZATION REPORT NUMBER
442427-23098

9. SPNSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Office of Naval Research
800 North Quincy Street
Arlingotn, VA 22217-5660

10. SPONSORING /MONITORING AGENCY

11. SUPPLEMENTARY NOTES
n/a

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for Public Release
Distribution Unlimited

19970714 056

13. ABSTRACT (Maximum 200 words)

During the past year, we have continued our efforts in image and video compression. We developed a real time software only scalable video compression codec. We have also optimized the scalable coder for transmission over wireless links by jointly optimizing the channel and source coders. We also have developed a video retrieval algorithm using motion, size, and color. We constrained the algorithm to be able to retrieve video in real time and only using information available in compressed bit-streams. In the are of video compression, we continued improving video coding using matching pursuits. We improved the coding efficiency performance by more than 1 dB for many of MPEG-4 test sequences. We have worked on reducing the complexity of the encoder and developed a real time software only low bit rate encoder.
We also addressed the error resilience issue and developed a more error resilient matching pursuit video codec that performs as well as original coder, if not better.

14. SUBJECT TERMS	15. NUMBER OF PAGES
	16
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	

Summary of Activities for 4/31/96 through 4/31/97

Avideh Zakhor
Department of Electrical Engineering and Computer Sciences
University of California
Berkeley, CA 94720

Image and video compression algorithms are an important part of many transmission and storage systems. Over the past year, we addressed issues related to video compression for transmission and storage. Specifically, we dealt with four aspects of video compression:

- real time software implementation of scalable video compression algorithm [10, 11],
- transmission of video over wireless links [1],
- content-based retrieval of video [8], and
- low bit rate video coding [6].

In this report, we will briefly outline our results in these areas.

1 Low Complexity, Software Only, Scalable Video Codec

Developing scalable video compression algorithms has attracted considerable attention in recent years. Generally speaking, scalability refers to the potential to effectively decompress subsets of the compressed bit stream in order to satisfy some practical constraint, e.g., display resolution, decoder computational complexity, and bit rate limitations.

Over the past years, we have developed a three dimensional subband scalable video codec [12, 13]. We have also reduced the complexity of the codec and developed a real-time software only implementation on multi-processors [10, 11]. Here we will go a step further and show a real time software only implementation on Ultra-Sparc workstations.

One of the most intensive computational parts of the codec in [12] is arithmetic coding of multi-rate quantized 3-D subband coefficients. To reduce the complexity of this part, we investigated block coding as an alternative to the arithmetic coding of subband coefficients. Block coding proved to be less complex with very small loss in PSNR performance.

We will now show how hierarchical block coding techniques can be used to code significant maps. A Significance map of a quantization layer is a binary map showing the location of coefficients in the dead-zone, versus those outside the dead-zone. An example of significance maps for quantization layers zero and one are shown in Figure 1. The basic hierarchical block

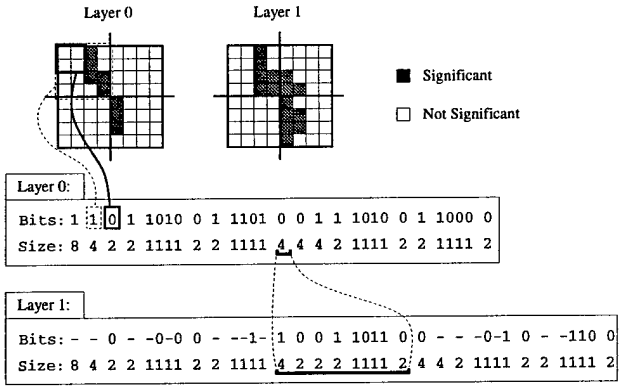


Figure 1: Example of two layers of block coding. "Bits:" is the coded bit-stream. "Size:" is the size of the block corresponding to the above bit.

coding technique we use was presented in an early work by Kunt [3] for two-level images. Kunt's method begins by partitioning an image into 16×16 blocks. If the block contains all zeros, the block is coded as a "0", and the algorithm proceeds to the next block. Otherwise, the block codeword begins with a "1", and the block is subdivided into four 8×8 blocks, each of which are coded the same way. In this manner, the coding proceeds in a recursive manner until 1×1 blocks We show an example of coding the first two layers of an 8×8 block in Figure 1. As seen, the initial layer, layer 0, is coded using Kunt's original method. To code the next layer, we use the information in the previous layer to avoid coding redundant bits. Specifically, any bits that are marked "1" in the previous layer are also assumed to be "1" in the following layer.

To reduce the complexity, we use a pyramid structure of sum blocks to compute the coded bit-stream. Moreover, the 3-dimensional subband analysis and synthesis are performed using separable applications of 1-dimensional filters.

1.1 Results

[t] In this section, we compare the performances of the scalable codecs to that of MPEG-1. Table 1 shows the speed performance of MPEG and the 3-D subband codec for four different sequences at six different rates [1]. The speed tests were done on a 170 MHz Ultra-Sparc workstation.

As seen, depending on speed, the subband codec with block encoding is 20 to 60 times

[1] *rd* stands for the sequence "Raiders of the lost ark", *pp* for "Ping Pong", *fb* for "Football" and *md* for "Mother Daughter". $t-1$ and $t-2$ denote one and two layers of temporal decompositions respectively. *ac* stands for scalable coding with arithmetic coding and *bc* stands for scalable coding with block coding. $mpeg.exh_e$ and $mpeg.log_e$ stand for MPEG encoding with exhaustive and logarithmic search respectively.

Table 1: Encoding and decoding speed comparison.

R (kbits/s)	64	256	500	1000	1500	3000
$rd-t2-bc_e$	24.0	24.0	20.0	15.2	12.5	9.9
$rd-t2-bc_d$	24.0	21.4	18.5	14.5	12.8	10.6
$rd-t2-ac_d$	24.0	18.7	11.0	7.0	5.4	3.9
$mpeg_d$	24.0	24.0	24.0	24.0	24.0	24.0
$mpeg.exh_e$	0.4	0.4	0.4	0.4	0.4	0.4
$mpeg.log_e$	1.6	1.6	1.6	1.6	1.6	1.6
$rd-t1-bc_d$	24.0	24.0	19.9	15.6	13.8	10.8
$pp-t1-bc_d$	24.0	20.6	16.7	13.7	11.9	9.2
$fb-t1-bc_d$	24.0	21.3	17.0	13.6	11.8	9.6
$md-t1-bc_d$	24.0	21.3	18.0	14.4	12.4	9.5

faster than exhaustive search MPEG encoding. Even though using logarithmic instead of exhaustive search speeds up MPEG encoding by a factor of 4, it is still considerably slower than scalable codec with block coding. The speed of both scalable codecs, based on arithmetic and block coding are for the most part symmetric with respect to encoding and decoding. As seen, the block coding approach is up to twice as fast as arithmetic coding approach. Decreasing the number of temporal decompositions from 2 to 1, speeds up the encoding/decoding of the block coder.

Table 2 shows the luminance PSNR performance of the 3-D subband codec using block coding and MPEG-1 for four video sequences at rates 0.5, 1, 1.5, and 3 Mbits/s. The scalable codec uses two layers of temporal decomposition unless otherwise stated. It is important to emphasize that for the scalable 3-D subband codec one bit stream at 3 Mbits/s is generated once and its subsets are extracted to obtain other bit streams at other bit rates. On the other hand, for MPEG, a whole different bit stream is generated at the encoder for each bit rate. As seen, except for Ping Pong, the scalable codec performs as good or better than MPEG codec for the other three sequences. Decreasing the number of temporal decompositions from two to one sometimes adversely affects the SNR, and improves encode/decode speed.

2 Scalable Video Transmission over Wireless Channels

The advent of wireless personal communications services in recent years has created a number of challenging research problems in the areas of communications, signal processing and networking. A major challenge in dealing with the wireless channel has to do with its inher-

Table 2: PSNR comparison.

Rates Mbits/s	0.5	1 (t2)	1 (t1)	1.5	3.0
$rd - mpeg$	30.9	34.1		35.9	38.9
$rd - bc$	31.7	35.2	35.2	37.0	39.3
$pp - mpeg$	25.9	28.5		30.3	33.9
$pp - bc$	25.1	28.4	26.8	30.1	33.0
$fb - mpeg$	30.2	33.1		34.9	38.0
$fb - bc$	31.0	34.1	34.3	35.9	38.0
$md - mpeg$	36.0	38.7		40.7	42.9
$md - bc$	37.0	40.4	38.9	42.1	45.0

ent unreliability. This is in contrast with wired networks in which the physical loss is very small, e.g. of the order of 10^{-9}.

The main problem we solve is as follows: given a total number of bits C, and a given binary symmetric channel with bit error probability of error P_e, find the best source coding rate R_s and channel coding rate R_c such that $C = R_s + R_c$, and the expected value of MSE is minimized. This is equivalent to finding the optimal source to channel bit ratio R_s^o/R_c^o, with $R_s^o + R_c^o = C$, such that the distortion is minimized. To find these minima for various CSI's, our approach is to construct distortion curves $D(\frac{R_s}{C-R_s})$ and to locate the minima empirically. We use the scalable coder described in [12] and implemented real time in the previous section.

Our solution to the optimization is based on a variation of Lagrange Multipliers, similar to the one developed in [9], with the exception that we are considering optimization of two sets of variables instead of one [1]. A complete mathematical derivation can be found in [1].

2.1 Results

To test the above algorithm numerically, we use Rate-Compatible Punctured Convolutional Codes [2] for channel coding, in order to achieve unequal error protection without changing the structure of the channel codec. We use 600 frames of the digitized video "raiders of the lost ark" to compute the distortion functions, and apply our proposed bit allocation strategy to search for the optimal source to channel coding ratio R_s^o/R_c^o for various CSI ranging from .001 to .05. The total bit budget is assumed to be 250 kbits/s. We see in Figure 2 that there exists a unique distortion minimum for various P_e.

To show that our optimization strategy is essential in high error rate environment, we compare the performance of our joint source/channel codec to one that employs equal error protection only. This codec also operates at the optimal source to channel coding rate,

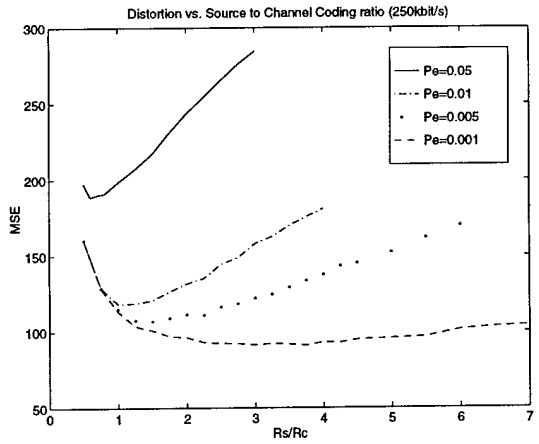


Figure 2: MSE vs. R_s/R_c for various CSI

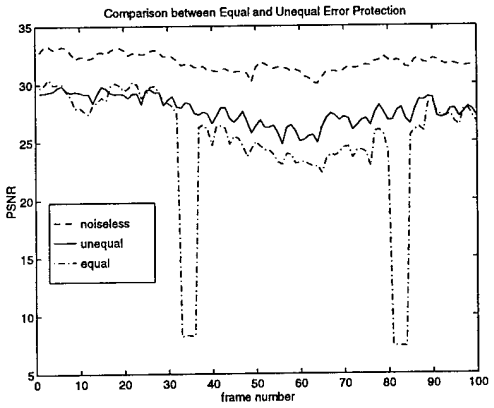


Figure 3: Comparison between equal and non-equal error protection.

R_s^o/R_c^o. However, it distributes R_s^o source bits using traditional bit allocation theory that assumes a noiseless channel, then codes these source bits with R_c^o channel bits equally. In Figure 2.1 we see the performance of both codecs compare to the case when the channel is noiseless. Because the important source bits are not protected with higher priority, we see that occasionally the video suffers disastrous effects from channel noise.

3 Video Compression using Matching Pursuits

All existing video compression standards are hybrid systems. That is, the compression is achieved in two main stages. The first stage, motion compensation and estimation, predicts each frame from its neighboring frames, compresses the prediction parameters, and produces

the prediction error frame. The second stage codes the prediction error. All existing video compression standards use block-based discrete cosine transform (DCT) to code the residual error. Although, DCT video coding is efficient, it introduces undesirable effects onto the video sequence. Video sequence compressed using block-DCT approaches suffer from "blocking" artifacts, especially at low bit rates. Moreover, due to bit rate restrictions, some blocks are only represented by one or a small number of coarsely quantized transform coefficients, hence the decompressed block will only consist of these basis vector. This will cause artifacts commonly known as ringing and mosquito noise. To solve this problem, we developed a coder that tries to match the residual error after motion compensation wherever it occurs using an over-complete basis set.

3.1 Matching Pursuits

Representing a signal using an over-complete basis set implies that there is more than one representation for the signal. For coding purposes, we are interested in representing the signal with the fewest basis vectors. This is an NP-complete problem [4]. Different approaches have been investigated to find or approximate the solution. Matching pursuits is a multistage algorithm, which in each stage finds the basis vector that minimizes the mean-squared-error [4].

Suppose we want to represent a signal $f[i]$ using basis vectors from an over-complete dictionary (basis set) $\mathcal{G}$. Individual dictionary vectors can be denoted as:

$$w_\gamma[i] \in \mathcal{G}. \tag{1}$$

Here γ is an indexing parameter associated with a particular dictionary element. The decomposition begins by choosing γ to maximize the absolute value of the following inner-product:

$$t = < f[i], w_\gamma[i] >, \tag{2}$$

where t is the transform (expansion) coefficient. A residual signal is computed as:

$$R[i] = f[i] - t\, w_\gamma[i]. \tag{3}$$

This residual signal is then expanded in the same way as the original signal. The procedure continues iteratively until either a set number of expansion coefficients are generated or some energy threshold for the residual is reached. Each stage k yields a dictionary structure specified by γ_k, an expansion coefficient $t[k]$, and a residual R_k, which is passed on to the next stage. After a total of M stages, the signal can be approximated by a linear function of the dictionary elements:

$$\hat{f}[i] = \sum_{k=1}^{M} t[k]\, w_{\gamma_k}[i]. \tag{4}$$

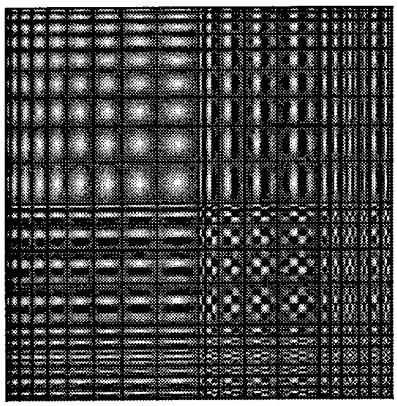


Figure 4: Separable two-dimensional 20×20 Gabor dictionary.

3.2 Coder Description

We have used a modified version of matching pursuit algorithm to represent the motion compensation residual error. First, the coder divides each motion residual into blocks and measures the energy of each block. The center of the block with the largest energy value is adopted as an initial estimate for the inner-product search. A dictionary of Gabor basis vectors, shown in Figure 4, is then exhaustively matched to an $S \times S$ window around the initial estimate. The exhaustive search can be thought of as follows. Each $N \times N$ dictionary structure is centered at each location in the search window, and the inner-product between the structure and the corresponding $N \times N$ region of image data is computed. The largest inner-product is then quantized. The location, basis vector index, and quantized inner-product are then coded together.

The decoder needs to know the basis function used to represent the residual error and its locations, and the value of the quantized inner-product. For a more efficient coder, the bases index and the inner-product are coded using variable length codes (VLC). To code atom positions, the atoms are sorted in position order from left to right and top to bottom within the residual image. A differential coding strategy employs three basic codeword tables. The first table P1 is used at the beginning of a screen line to indicate the horizontal distance from the left side of the image to the location of the first atom on the line. For additional atoms on the same line, the second table P2 is used to transmit the inter-atom distances. The P2 table also contains an escape code indicating that no additional atoms exist on the current line. The escape code, when used, is always followed by a P3 code, indicating how many lines in the image may be skipped before the next line containing coded atoms. The P3 code is then followed by a P1 code, since the next atom will be the first on a particular line. No special codeword is needed to indicate the end of the atom field, since the number of coded atoms is transmitted as header information.

Table 3: The average luminance PSNR of different sequences at different bit rates when coding using a DCT coder (MPEG-4 VM), zero-tree subband coder (ZTS), and matching pursuit coder (MP).

Sequence	Format	Rate		PSNR (dB)		
		Bit	Frame	DCT	ZTS	MP
CONTAINER-SHIP	QCIF	10 K	7.5	29.43	28.01	30.99
HALL-MONITOR	QCIF	10 K	7.5	30.04	28.44	31.17
MOTHER-DAUGHTER	QCIF	10 K	7.5	32.50	31.07	32.74
CONTAINER-SHIP	QCIF	24 K	10.0	32.77	30.44	34.21
SILENT-VOICE	QCIF	24 K	10.0	30.89	29.41	31.71
MOTHER-DAUGHTER	QCIF	24 K	10.0	35.17	33.77	35.56
COAST-GUARD	QCIF	48 K	10.0	29.00	27.65	29.84
NEWS	CIF	48 K	7.5	30.95	29.97	32.02

3.3 Results

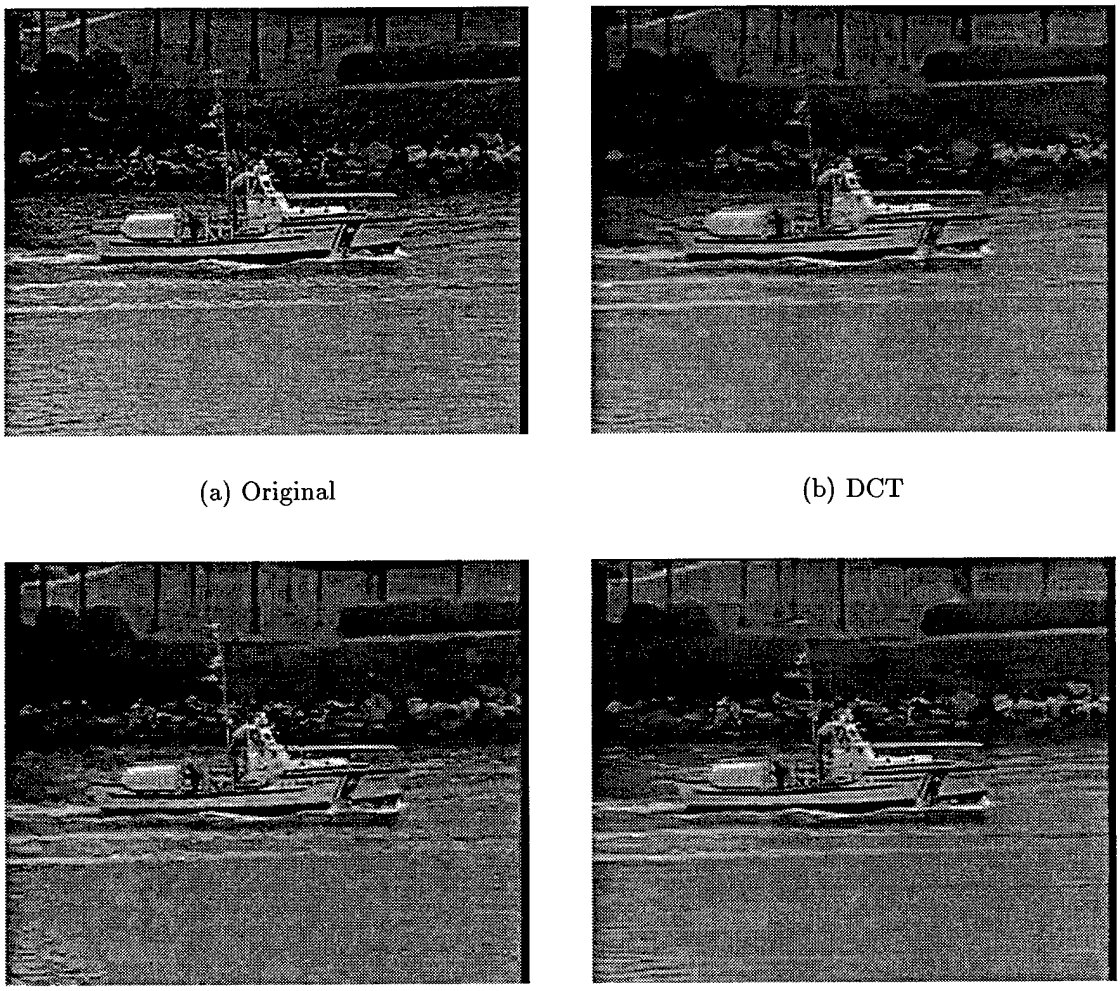
Figure 5 shows frame 250 of the 15 frame/s CIF COAST-GUARD sequence coded at 112 Kbits/s using DCT, subband, and matching pursuit coders. The matching pursuit coded frame does not suffer from the blocky artifacts, which affect the DCT coders as shown in Figure 5(b). Moreover, it does not suffer from the ringing noise, which affects the subband coders as shown in Figure 5(c).

Video sequences coded using matching pursuit do not suffer from either blocking or ringing artifacts, since the basis vectors are only coded when they are well-matched to the residual signal. As bit rate decreases, the distortion introduced by matching pursuit coding takes the form of a gradually increasing blurriness (or loss of detail).

Figure 6 compares the PSNR performance of the matching pursuit coder [6] to a DCT (MPEG-4 verification model, VM) coder and a zerotree subband coder [5] when coding the COAST-GUARD sequence at 112 Kbits/s. The matching pursuit coder [6], in this example, has consistently higher PSNR than the MPEG4 and the zerotree subband [5] coders. Table 3 shows the average luminance PSNRs for different sequences at different bit rates. In all examples mentioned in Table 3 the matching pursuit coder has higher average PSNR than the DCT coder.

3.4 Real Time Encoding using Matching Pursuits

The main objection to video coding using matching pursuit is the complexity of the encoder. The decoder is very simple and its complexity linearly increases with bit rate (number of atoms). The encoder, however, involves an exhaustive search over a small region using all possible basis functions. This limits the applicability of matching pursuits for software only



(a) Original

(b) DCT

(c) Subband

(d) Matching pursuit

Figure 5: Frame 250 of COAST-GUARD sequence, original shown in (a), coded at 112 Kbits/s using: (b) DCT based coder (MPEG-4 VM), (c) zerotree subband coder, and (d) matching pursuit coder. Blocking artifacts can be noticed on the DCT coded frame. Ringing artifacts can be noticed on the subband coded frame.

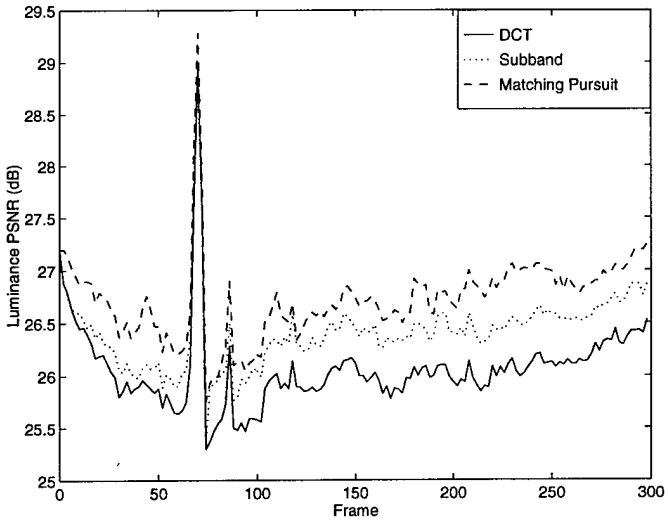


Figure 6: Frame-by-frame distortion of the luminance component of the COAST-GUARD sequence, reconstructed from 112 Kbits/s MPEG-4 VM bit stream (solid line), a zerotree subband bit-stream (dotted line), and from a matching pursuit bit stream (dashed line). Consistently, the matching pursuit coder had the highest PSNR while the DCT coder had the lowest PSNR.

real time encoding. We addressed this problem and developed a real time software only encoder that runs on Pentium 200 MHz PCs.

The complexity of matching pursuits was reduced mainly by approximating the computation of the inner-product and reducing the search. The following summarizes these approximations:

- The number of basis functions has been reduced from 20×20 basis function to 17×17 basis functions. This alone does not reduce the complexity significantly. The choice of the new basis functions, however, does. Eleven out of the seventeen basis functions have the same scale. These eleven functions consist of the same Gaussian multiplied by sines and cosines. Thus, they can be computed using fast Fourier Transform (FFT).

- While searching, the Gaussian part of the Gabor function is approximated using quantized Gaussian functions. Thus, the multiplications are reduced to the number of quantized coefficients.

- It's known that

$$< f, g > \leq \sqrt{< f, f >< g, g >}. \tag{5}$$

We can find the norm, $< f, f >$, of the search area and compare it the best value of the inner-product so far (In our case $< g, g >= 1$). If it was smaller than the largest

Table 4: The average PSNR and speed of the fast matching pursuit coder (MP) compared to MPEG-4 verification model (VM). All are 10 Kbits/s QCIF sequences.

Sequence	PSNR (dB)			Time per frame (ms)		
	VM	MP	ΔPSNR	VM	MP	MP/VM
CONTAINER-SHIP	29.55	30.21	0.66	0.131	0.150	1.15
HALL-MONITOR	29.96	30.63	0.67	0.119	0.138	1.16
MOTHER-DAUGHTER	32.45	32.56	0.11	0.132	0.132	0.99

inner-product found so far, no search is needed. We can relax this condition in expense of less accuracy by multiplying the norm of the search area by a factor less than one. This will speed the search. The loss in quality depends on the chosen factor.

- The search area is reduced from 16×16 to 8×8 search area. This increased the speed by a factor of 4.
- The search is subsampled in the horizontal direction for all basis functions except for the bases function with scale 1. The search is subsampled in the vertical direction according to the support of the basis functions. The search is vertically subsampled only for functions with large support.
- After the best function is found, a local search using the best function is done on a 3×3 window. The search is done by computing the inner-products accurately.

As expected, reducing complexity reduces the quality of the compressed video. However, visually, the differences are very small and in PSNR they are small. Moreover, the fast version of matching pursuits still outperforms the DCT based approach. Table 4 shows the average PSNR values for the fast version and compare them to the MPEG-4 verification model (VM). It should be noted here that the motion search for the examples in the table is not exhaustive. A combination of subsampled and step search is used for motion estimation.

3.5 Error-Resilient Matching Pursuit Coding

When transmitting video over noisy channels, it is important for bit-streams to be robust to transmission errors. It is also important, in case of errors, for the error to be limited to a small region and not to propagate to other areas. The position coding mechanism described in Section 3.2 does not limit the error in a frame. That is, if an error occurs in the middle of a frame, the whole frame will be lost.

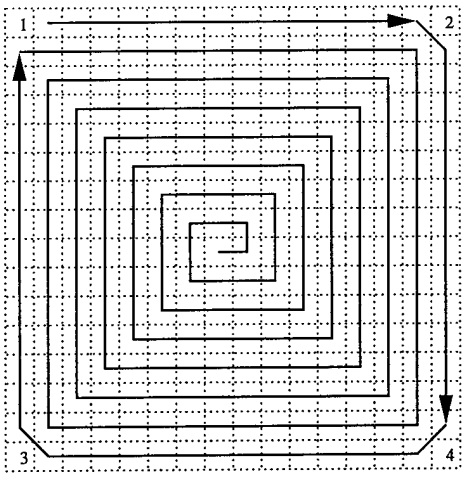


Figure 7: Scan used to code th atoms in a macroblock. First pixels 1, 2, 3, and 4 are coded.

We addressed this problem and developed a new position coding mechanism that limits the effect to a macroblock (16×16 pixels). The new position scheme codes atoms that are in the same macroblock together. The atoms of each macroblock are reordered according to the scan shown in Figure 7. Afterwards, the atoms are coded differentially. Four VLC tables are used to code the atoms. The VLC table used to code the atoms of a macroblock is chosen according to the number of atoms in that macroblock.

This new coding approach is more error-resilient. Moreover, the sequence coded using this approach has slightly higher PSNR than the older method. Table 5 shows the PSNR of the new (macroblock-based) and the old (frame-based) methods.

4 Motion Indexing of Video

A valuable tool in the management of visual records is the ability to automatically "describe" and index the content of video sequences in a meaningful manner. Such a facility would allow recovery of desired video segments or objects from a very large database of image sequences. The efficient use of stock film archives and identification of specific activities in surveillance videos are usually cited as potential applications.

A parallel goal to creating such a database is the use of compressed video in the indexing and searching functions. More specifically, the elements of the compressed sequence themselves should serve as search keys. The concept of compression is thus extended from only producing an efficient representation, to also providing a meaningful one. This idea is embodied in the term "content based video"[7].

The development of a representation technique driven by database considerations such as hierarchical and "meaningful" representations, has inspired the use of motion of objects

Table 5: The average luminance PSNR of different sequences at different bit rates when coding using the frame-based position MP coder (old) and the macroblock-based position MP coder (new).

Sequence	Format	Rate		PSNR (dB)	
		Bit	Frame	Frame-based	Macroblock-based
CONTAINER-SHIP	QCIF	10 K	7.5	31.99	31.08
HALL-MONITOR	QCIF	10 K	7.5	31.17	31.35
MOTHER-DAUGHTER	QCIF	10 K	7.5	32.74	32.75
CONTAINER-SHIP	QCIF	24 K	10.0	34.21	34.27
SILENT-VOICE	QCIF	24 K	10.0	31.71	31.92
MOTHER-DAUGHTER	QCIF	24 K	10.0	35.56	35.69
COAST-GUARD	QCIF	48 K	10.0	29.84	29.90
FOREMAN	QCIF	48 K	10.0	30.78	30.81
NEWS	CIF	48 K	7.5	32.02	32.16
COAST-GUARD	CIF	112 K	15.0	26.73	26.59
FOREMAN	CIF	112 K	15.0	28.62	28.61
NEWS	CIF	112 K	15.0	35.26	35.44
STEFAN	SIF	1 M	30.0	29.52	29.63
MOBILE-CALENDER	SIF	1 M	30.0	26.86	26.93

to index a video database. We applied this concept to a street surveillance application [8]. A segmentation and tracking program analyzes compressed video of a scene and extracts the trajectories of moving objects, represented as two dimensional curves parameterized by time. The coarse-scale components of these trajectories are stored as keys in an index. A user who wishes to find an object moving in a particular way draws a trajectory, which is then matched against those in the index.

The present video database system is shown schematically in Figure 8. It comprises components to achieve trajectory extraction, index building, and easy user interaction. Motion vectors from MPEG 1-compressed video form the sole input to the system. From these, the trajectories of objects in a fixed scene are extracted and represented by their wavelet transforms. The multi-resolution nature of the wavelet representation is the key to using inexact or incomplete queries, allowing imprecise matching and retrieval of desired clips of video.

A graphical user interface that is particularly designed for the surveillance application accepts hand-drawn queries and returns matches pictorially and in order of increasing distance from the query. Figure 9 shows the results of a typical query.

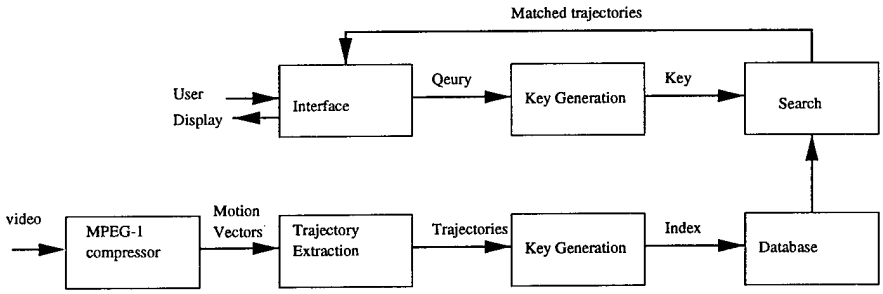


Figure 8: *The database system, showing the interaction of the three components.*

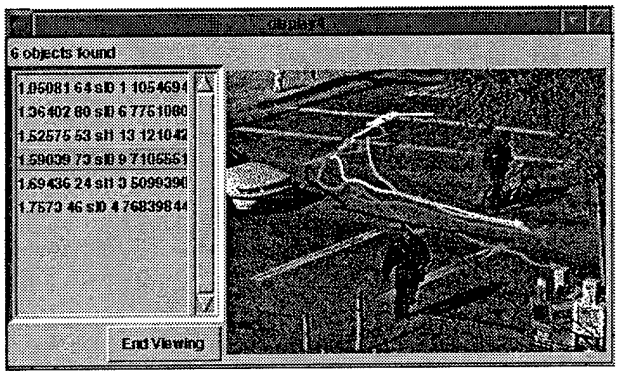


Figure 9: *Example query results. The query was for left-hand turns.*

5 Evaluation

Both the feature extraction and database parts of the system were tested. Approximately 20 minutes of video were analyzed both manually and by the proposed segmentation and tracking algorithm for trajectories. The number of correctly extracted trajectories was determined by comparing a graph of extracted trajectories to those observed by a human viewing the video. 409 objects were detected manually, of which the algorithm found 338. The tracking program also produced 141 false detections, representing noise and short fragments of actual trajectories. The results in Table 6 show the rates of missed detection for three categories of objects. As seen, the algorithm is more successful in tracking vehicles than people. This is due to problems in segmentation resulting from the more complex motions of people in the scene, and to their smaller size. The imprecision and noisiness of the MPEG motion estimates is simply inadequate to describe these motions correctly, or to recognize them at all. Tracking fails as well when motions are too similar, or occlusions too great. The segmentation and tracking algorithms work easily in real time.

The recall precision of the database was tested next. The precision for a given recall rate

Table 6: *Segmentation results.*

Object Type	Manual Detections	Algorithm Detections	Rate
People	209	163	78%
Bicycles	6	6	100
Motor Vehicles	194	169	87

Table 7: *Precision for 90% recall rate.*

Query	Tracking Detections	Precision at 90% Recall
People, cross near $\rightarrow$ far	35	86%
People, cross far $\rightarrow$ near	30	75%
People, near side, E $\rightarrow$ W	32	69%
People, near side, W $\rightarrow$ E	29	78%
People, far side, W $\rightarrow$ E	11	65%
Car Left Turns	6	100%
Cars E $\rightarrow$ W	68	93%
Cars W $\rightarrow$ E	72	80%

r is defined as

$$precision = \frac{r * N_{objects}}{r * N_{objects} + N_{false\_detections}}$$

where $N_{objects}$ is the number of objects in the database determined by a human to match the query, and $N_{false\_detections}$,the number of false detections, is the number of objects not matching the query activity that are found before a total of $r * N_{objects}$ matching objects are found. Table 7 shows the precision rates for the 90% recall rate for various hand-drawn queries. The numbers show that the indexing and retrieval strategy is significantly more successful for vehicles than for pedestrians.

References

[1] G. Cheung and A. Zakhor, "Joint source-channel coding of scalable video over noisy channels ," *Proc. of ICIP,* vol. 3, pp. 767-770, 1996.

[2] J.Hagenauer, "Rate-Compatible Punctured Convolutional Codes (RCPC Codes) and their Applications," *IEEE Trans. Comm.*, vol.36, pp. 389-399, Apr. 1988.

[3] M. Kunt, "Block Coding of Graphics: A Tutorial Review," *Proc. IEEE*, Vol. 68, No. 7, Jul. 1980.

[4] S. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Trans. Signal Processing,* Vol. 41, No. 12, pp. 3397-3415, December 1993.

[5] S. A. Martucci, I. Sodagar, T. Chiang, and Y. -Q. Zhang, "A zerotree wavelet coder," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 7, no. 1, pp. 109-118, February, 1997.

[6] R. Neff and A. Zakhor, "Very low bit rate video coding based on matching pursuits," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 7, no. 1, pp. 158-171, Feb. 1997.

[7] A. Pentland, R. Picard, S. Scarloff, "Photobook: Tools for Content-Based Manipulation of Image Databases", *SPIE Conf. Storage and Retrieval of Image and Video Databases II*, 2185, Feb 1994.

[8] E. Sahouria and A. Zakhor, "Video indexing based on object motion," Submitted to *IEEE Trans. Image Processing,* 1997.

[9] Y.Shoham and A.Gersho, "Efficient Bit Allocation for an Arbitrary Set of Quantizers," *IEEE Trans. ASSP*, vol.36, pp. 1445-1453, Sep. 1988.

[10] D. Tan and A. Zakhor, "A real time software decoder for scalable video on multi-processors",*presented at 1996 Packet Video Workshop,* Australia, Mar. 1996.

[11] D. Tan and A. Zakhor, "Real time software implementation of scalable video codec," *Proc. ICIP,* vol. 1, pp. 17-20, 1996.

[12] D. Taubman and A. Zakhor, "Multirate 3-D subband coding of video," *IEEE Trans. Image Processing*, vol. 3, no. 5, pp. 572-88, Sept. 1994.

[13] D. Taubman and A. Zakhor, "A common framework for rate and distortion based scaling of highly scalable compressed video," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 6, no. 4, pp. 329-354, Aug. 1996.