Low Bit-Rate Video Coding with Implicit Multiscale Segmentation

Seung Chul Yoon, Krishna Ratakonda, and Narendra Ahuja, *Fellow, IEEE*

***Abstract*— In this paper, we report on our efforts toward developing a multiscale segmentation based video compression algorithm aimed at very low bit-rate applications such as video teleconferencing and video phones. We introduce novel techniques for multiscale segmentation based motion compensation and residual coding. Our region based forward motion compensation strategy (in terms of direction of motion vector, which is from the previous frame to the current frame) regulates the size and number of regions used, by pruning a multiscale segmentation of video frames. Since regions used for motion compensation are obtained by segmenting the previously decoded frame, the shape of the regions need not be transmitted to the decoder. Furthermore, our hierarchical motion compensation strategy refines an initial region level, coarse motion field to obtain a dense motion field which provides pixel level motion vectors. The refinement procedure does not require any additional information to be transmitted. This motion compensation technique effectively addresses the problem of dealing with "holes" and "overlapping regions" which are inherent to forward motion compensation. Residual coding is performed using a novel method which exploits the fact that the energy of the residual resulting from motion compensation is concentrated in *a priori* predictable positions. We will show that this residual coding technique can also be extrapolated to improve the performance of coders using a block based motion compensation strategy. A fusion of these concepts leads to a gain of 2–3 dB in peak signal-to-noise ratio, apart from significant perceptual improvement, over a generic video coding algorithm using a block based motion compensation strategy (such as H.261 or H.263) for a variety of test sequences.**

***Index Terms*—Hierarchical motion compensation, low bit-rate video compression, multiscale segmentation, residual coding.**

I. Introduction

VIDEO sequences are characterized by spatial and temporal redundancies which have to be exploited by any effective video compression algorithm. Video compression, as implemented in most popular coders, is a two step procedure. The first step, called motion compensation, consists of predicting the current frame using motion information from the previously decoded frame(s). The previously decoded frame from which motion compensation is attempted can either precede or succeed the current frame in display order. Motion compensation is aimed at exploiting the temporal redundancies in the video sequence. In the second step, the residual, which is the difference between the actual video frame and the frame resulting from motion compensation, is partially transmitted according to bit-rate restrictions. Typical algorithms reported in literature focus on utilizing spatial redundancies for effective residual coding. In order to limit the effect of progressive degradation which results from this two step procedure, a refresher frame, which is intra-coded without using motion compensation, is periodically transmitted.

In this paper we will propose novel implicit, multiscale, image segmentation based motion compensation and residual coding strategies which result in both subjective (perceptual) and objective peak signal-to-noise ratio (PSNR) improvements when compared with a generic block based algorithm. Our video compression algorithm is aimed at low bit-rate applications where the performance degradation resulting from block based algorithms is typically unacceptable. Although post processing schemes for dealing with errors resulting from block based motion compensation exist, such algorithms typically result in a blurring of the decoded image. Examples of low bit-rate applications of interest include video teleconferencing and video phones. Furthermore, the techniques that we develop, effectively address some of the problems inherent to video coding with arbitrarily shaped regions and hence can find independent application in other object based video coding strategies.

In the rest of this section, we will review a few relevant motion compensation and residual coding strategies from literature and introduce the proposed approach.

A. Related Previous Work

Many strategies for performing video compression have been explored in the past decade [1]. Standards aimed at different bit-rate requirements have been proposed, which range from the moving picture experts group (MPEG) standard [2], [3] aimed at relatively high bit-rate applications such as data storage in compact discs to the H.261 and H.263 standards [4], [5] which are aimed at low bit-rate applications such as video conferencing and video telephony. Design of low bit-rate video coders is typically more challenging as video quality becomes noticeably poor at low bit-rates thus increasing the problems involved in the design of a compression mechanism which can deliver an acceptable level of perceptual quality. Thus, many of the current papers in literature, which propose novel video compression strategies, tend to focus on low bit-

Manuscript received September 10, 1998; revised April 21, 1999. This work was supported by the U.S. Office of Naval Research under Grant N00014-96-1-0502 and by the National Science Foundation under Grant IRI 93-19038. This paper was recommended by Associate Editor F. Pereira.

S. C. Yoon and N. Ahuja are with the Department of Electrical and Computer Engineering, Beckman Institute for Advanced Science and Technology, University of Illinois at Urbana-Champaign, Urbana, IL 61801 USA.

K. Ratakonda was with the Department of Electrical and Computer Engineering, Beckman Institute for Advanced Science and Technology, University of Illinois at Urbana-Champaign, Urbana, IL 61801 USA. He is now with the IBM T. J. Watson Research Center, Yorktown Heights, NY 10598 USA.

Publisher Item Identifier S 1051-8215(99)08183-5.

rate applications. Although many algorithms claim improved PSNR performance at low bit rates, the applicability of PSNR as a reliable measure for perceptual quality is questionable at such bit rates. We attribute this to the fact that most algorithms use block based motion compensation strategies, with a possibility for variable block size in recent coders [5], [6], which cause significant block artifacts resulting in a loss in perceptual quality, if adequate residual coding is not performed after motion compensation due to bit-rate restrictions. This is indeed the case at the bit rates used in video telephony and teleconferencing.

Motion compensation has been studied extensively both for video compression [6], [7], [8] and general video analysis [9], [10], [11], [12]. The most common form of motion compensation used in a video compression setting is block based, backward (in terms of direction of motion vector, which is from the current frame to the previous frame) motion compensation and its variations. The MPEG-1 and H.261 standards split the image to be coded into 16×16 blocks and find the best backward motion vector[1] for each block, within a specified search range. The H.263 standard allows for a 16×16 block to be split into four 8×8 blocks, the criterion for choice between the larger and smaller blocks being left to the specific implementation of the encoder. The MPEG-2 standard allows for a 16×16 block to be split into two 16×8 blocks, the split being allowed in the vertical direction. The reason behind allowing a split in the vertical direction, but not in the horizontal direction, is that the horizontal motion is much larger than vertical motion in typical video sequences. Recent algorithms have also explored allowing a quad-tree splitting procedure to vary the size of the blocks used for motion compensation. Schuster and Katsaggelos [6] show how to perform a quad-tree partitioning by allocating the optimal number of bits to partitioning, motion compensation, and residual coding in a rate-distortion setting. However, as noted before, block based compensation typically results in a peaky distribution of the residual error, with high residual concentration at block edges and image edges. At low bit rates, when adequate residual coding cannot be performed, this results in perceptually unacceptable performance. The usual fix for this situation is to use a pre- or post-processing filter which smoothes the block artifacts [13], [14]. Such filtering typically leads to a blurring of the true image edges in addition to a reduction of the block artifacts.

Segmentation based motion compensation schemes have also been explored in some detail in literature. Such schemes can be classified into methods which use motion field segmentation [15], [16] and those which use intensity based segmentation [17], [18], [19]. Motion field segmentation partitions the image into regions based on a criterion of similarity in motion. The assumption behind such a scheme is that groups of pixels which have similar motion over the past few frames will continue to move in a similar fashion in successive frames. Such assumptions lead to large errors when the implicit smoothness assumption on motion is violated and lead to little or no advantage over block based methods when averaged over the entire video sequence. Intensity based segmentation partitions the image based on grey scale homogeneity and does not make assumptions on the smoothness of motion. Since the scale, at which segmentation is performed, determines the size (and hence the number of the regions) generated, this has a direct effect on the amount of motion related information generated. At low bit rates it becomes important to optimize the number of bits spent on motion information. However, it is not clear as to what is the "optimal grey level homogeneity scale" at which segmentation is to be performed and how to control the process of segmentation based upon a choice of scale. Recent work [19] proposes to use multiscale segmentation derived from a morphological processing framework. This method still requires the transmission of segmentation information to the decoder which results in an unacceptable amount of overhead at the bit -rates targeted in this paper.

Apart from the difficulties considered above, a major drawback of previous segmentation based schemes is that the segmentation information needs to be explicitly encoded. This typically results in an unacceptable amount of overhead at low bit rates. A possible solution to this problem might be to use forward motion compensation[2] in which segmentation is performed on the previously decoded frame and forward motion vectors are calculated for regions [17]. However, the difficulty with such an approach is that all of the pixels in the frame to be coded cannot be predicted; the leftover pixels result from "holes" (where no prediction is available) or "overlapping regions" (which are covered by more than one translated region). Yokoyama [17] proposes an ad hoc solution in which such regions are predicted by interpolating or averaging the motion vectors from known locations. Such an approach results in large prediction errors in the ambiguous regions thus defeating the purpose of using a segmentation based approach which is to maintain good perceptual quality.

Another motion compensation scheme of interest is the pel recursion algorithm [7], [8]. Biemond and Looijenga [7] formulated a recursive Wiener estimate. The recursive equation is obtained by considering a Taylor series expansion of the intensity image as a function of the motion vector to be estimated. It is well known that the inherent linearization due to Taylor series representation limits the applicability of the algorithm to situations with small motion and where a good initial estimate of motion is available. Pel recursion typically results in fractional pixel accuracy motion vectors. Since images consist of pixel values at discrete locations an interpolation mechanism is needed to extend pel recursion to the discrete setting. Nosratinia and Orchard [8] proposed a scheme for obtaining an optimal linear interpolant by solving a linear least squares formulation in a causal neighborhood. We will use a modified version of these ideas in our motion compensation algorithm.

Most of the previous video compression algorithms adapted techniques native to still image compression to do residual coding [2], [4], [20]. Such an approach is far from optimal since it does not exploit the redundancies introduced by

[1] The criterion used for finding the best matched motion vector is that of minimizing the mean squared error.

[2] In the terminology of this paper, the forward in forward motion compensation refers to the direction of motion vectors, which map a region/block in the previous frame to the current frame.

the fact that the residual is synthetically generated through the process of motion compensation. Block discrete cosine transform (DCT) coding, with 8×8 blocks, is typically the most common residual coding formulation and is used in all standard implementations. Although other schemes involving vector quantization [20] have been explored, they are not as popular. Vector quantization in particular requires a good code book for obtaining high performance. However, the design of a generic code book is difficult due to (a) local minima problems of the popular design algorithms (such as K means) and (b) variability of the video content. Many DCT-based implementations [6], [17] code the residual in only selected 8×8 blocks to meet bit-rate restrictions.

B. Video Compression with Implicit Segmentation

In the proposed approach we start with a multiscale segmentation of the previous frame provided by the algorithm in [21], [22]. A scale parameter in the segmentation algorithm controls the grey level homogeneity of the regions into which the image is partitioned. At a coarse scale the image is partitioned into a few large, relatively inhomogeneous regions while at a fine scale we obtain more homogeneous regions at the expense of a decrease in the size of the individual regions. The algorithm provides segmentation at as many different scales as are naturally present within the image, ranging from coarse to fine. A significant feature of the algorithm is that a parent-child relationship is preserved across the different scales of segmentation (see Fig. 1). In other words, a region at a coarse scale of segmentation can only split into smaller regions at finer scales. No new regions which partake of more than one region at a coarse scale can form at finer scales. This results in a tree structured representation for multiscale segmentation. To draw a parallel, we obtain exactly the same kind of representation as the popular quad-tree segmentation, which is used extensively in lossy compression literature, except that we obtain arbitrarily shaped regions in the place of square blocks.

The hierarchical, multiscale segmentation based motion compensation scheme effectively addresses many issues which have made previously proposed segmentation based coders [17], [23], [24], [25] unattractive for video compression. We use a forward motion compensation strategy instead of the conventional backward motion compensation and therefore avoid sending the segmentation related information to the decoder. In this sense, the segmentation information that we use is implicit. Forward motion compensation, however, poses its own challenges as "holes" and "overlapping regions" exist in the motion compensated image. Thus the motion field after forward motion compensation is typically sparse. We call this a "coarse motion field." We propose a novel strategy which uses partial backward motion compensation followed by a modified pel recursion algorithm to fill in the motion information in such unpredictable regions. After this procedure, the motion field covers every pixel in the frame to be coded and hence it is dense. We call this a "dense motion field". The hierarchical process of first computing a coarse motion field followed by a dense motion field results in improved robustness in the motion estimates.

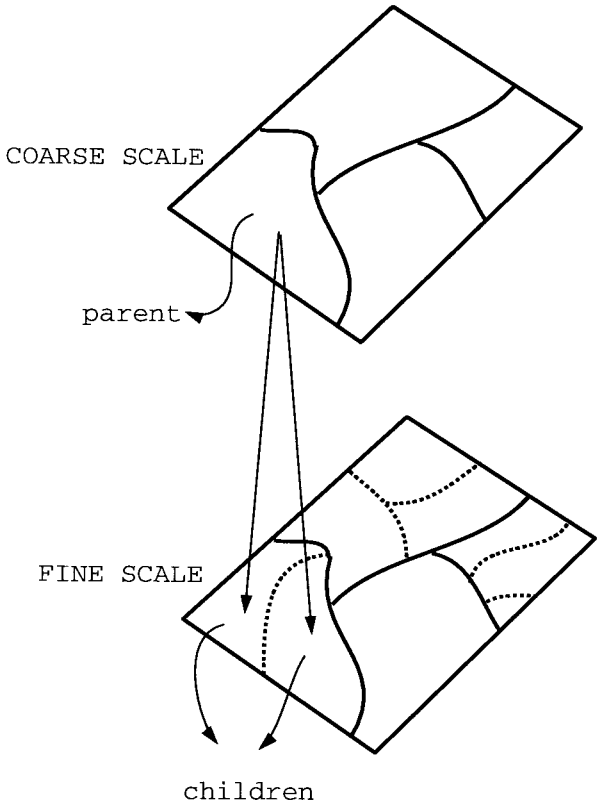


Fig. 1. Illustrating the parent child relationship between regions at the coarse scale and a fine scale segmentation of the image.

Another novel feature in our motion compensation algorithm is its ability to control the number of regions in the implicit segmentation used for motion compensation, thus allowing adaptive control of the amount of motion information transmitted to the decoder. The key idea behind this rate control mechanism derives from the multiscale nature of the segmentation algorithm [21], [22] which provides us with multiple scales of segmentation, thus allowing us to pick and choose the size and number of regions which form the implicit segmentation of the image used for motion compensation.

Our residual coding strategy also differs radically from the typical scheme used in most video compression algorithms [2], [3], [4], [5], [6]. Previously proposed residual coders typically consider the residual to be an image and therefore employ strategies native to still image compression algorithms. However, such strategies are in essence not exploiting the fact that the residual is not a natural image but synthetically generated by the process of motion compensation. Practical experiments on video sequences revealed that most of the residual ($\sim$90%) is concentrated in a tiny portion of the image ($\sim$10%). Furthermore, the parts of the image where most of the residual energy is concentrated can be reliably predicted using the locations of the edges in the previously decoded image and the region based motion vectors. It may be noted that in this observation, the location of high residual energy concentration can be pre-predicted, and is not tied to a particular motion compensation scheme. Therefore, our residual coding strategy naturally extends to block based

motion compensation schemes used in all of the standards [2], [3], [4], [5].

C. Overview

The next section describes our segmentation based motion compensation scheme in detail. The residual coding strategy is outlined in Section III. Specific details regarding the choice of various parameters and other implementation details are presented in Section IV. Results and a discussion of their implications are presented in Section V.

II. Motion Compensation

In this section we will outline the motion compensation strategy used within our video compression algorithm which has four major components:

1) a multiscale segmentation algorithm which is used to segment the previously transmitted frame;
2) coarse motion field generation: a forward motion compensation strategy which forms an initial prediction for the frame to be encoded using forward motion vectors which represent the translation of regions from the previous frame to the current frame;
3) dense motion field generation: a strategy for filling in the "holes" and "overlapping regions" where motion information from the forward motion compensation strategy was either not available or is ambiguous. This involves using motion vectors in the backward direction for relatively large regions where a prediction is not available and a modified pel recursion algorithm [7], [8] to fill in the motion compensated prediction of the current frame for the rest of the pixels;
4) a computationally inexpensive rate control strategy which uses segmentation at multiple scales, which differ in the number and size of regions into which the image is partitioned, to regulate the amount of motion information transmitted.

As noted in the introduction, the forward motion compensation strategy typically generates a coarse motion field which does not provide an intensity estimate for each pixel in the frame to be coded. Thus we may think of the third step in the above list as refining this coarse motion field to obtain a dense motion field which provides reliable initial intensity estimates for all the pixels in the frame to be coded. An advantage of this hierarchical motion compensation strategy is that the coarse motion field results in the generation of reliable motion estimates for initializing the pel recursion algorithm used in generating the dense motion field. As noted in [7], pel recursion works best when the initial estimate of the motion is good, so that the linearization implied by the Taylor series expansion is validated. We also modify the pel recursion algorithm along the lines of [8] to obtain reliable intensity estimates by finding the causally optimal interpolant. Further details are provided in Section II-C.

In the rest of the section details of the segmentation algorithm (Section II-A), coarse motion field generation (Section II-B), dense motion field generation (Section II-C), and the rate control mechanism (Section II-D) are provided.

A. Multiscale Segmentation Algorithm

The objective of segmentation is to partition the image into regions which are intrinsically similar and extrinsically dissimilar (with respect to all adjacent regions) in terms of grey level homogeneity. Multiscale segmentation aims at facilitating image segmentation at all geometric and photometric scales at which structure is present in the image. Previous compression algorithms used clustering or region growing [16], [17], [25], in order to achieve segmentation. These approaches produce errors in the resulting segmentation e.g., in the edge locations and in the delineation of regions e.g., in their homogeneity. A recent transform [21] which possesses desirable properties of multiscale segmentation, forms the basis of the segmentation algorithm used in this paper. In its continuous form, the transform maps a continuous two-dimensional (2-D) grey scale image $I(x,y)$ into a family of attraction force fields $\mathbf{F}[x,y,\sigma_g(x,y),\sigma_s(x,y)]$, as follows:

$$\mathbf{F}[x,y,\sigma_g(x,y),\sigma_s(x,y)] = \int\int_R d_g[\Delta I,\sigma_g(x,y)]\cdot d_s(\boldsymbol{r},\sigma_s(x,y))\frac{\boldsymbol{r}}{||\boldsymbol{r}||}\,dw\,dv$$

where $R = \mathit{domain}\{I(x,y)\}$, $\boldsymbol{r} = (v-x)\boldsymbol{i} + (w-y)\boldsymbol{j}$, and $\Delta I = |I(x,y) - I(v,w)|$. The transform has a similar form for a discrete image. It analyzes the intensities present in a neighborhood of the pixel, and produces a force vector for each pixel in the image. This force field makes the regions explicit in such a way as to make their extraction easy. Associated with each pixel is a homogeneity scale σ_g, which reflects the homogeneity of the region into the pixel groups and a spatial scale σ_s which controls the neighborhood over which the transform is applied. The spatial scale parameter σ_s controls the spatial distance function $d_s(\cdot)$, and the homogeneity scale parameter σ_g controls the homogeneity distance function $d_g(\cdot)$. Considering the various desirable properties for the distance functions it has been found that the optimum form is that of a box-car window (other forms, such as a 2-D Gaussian, are computationally expensive and yield little or no advantage over the simpler box-car window)

$$d_g(\Delta I,\sigma_g) \sim B_{\Delta I}(\sigma_g)$$
$$d_s(\boldsymbol{r},\sigma_s) \sim B_{||\boldsymbol{r}||}(\sigma_s)$$

where

$$B_x(y) = \begin{cases} 1, & |x| \le y \\ 0, & \text{else.} \end{cases}$$

By using a spatially invariant σ_g and computing the optimal σ_s, the transform can be applied to image segmentation. The selection of an optimal σ_s is equivalent to using the appropriate amount of spatial information to identify regions based on homogeneity. Further details may be obtained in [22], [26]. Results of a typical multiscale segmentation using the transform of the image Lena are given in Fig. 2.

B. Coarse Motion Field

In Section II-A we saw how to construct the multiscale segmentation of a given image. Selecting the right scale of segmentation for motion compensation will be explained in

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

Fig. 2. (a) Actual image and (b) different scales of segmentation given by the multiscale segmentation algorithm. (c) and (d) Different scales of segmentation given by the multiscale segmentation algorithm.

Section II-D. Given a segmentation at some specified scale of the previous image, we will describe in this section how to obtain a coarse forward motion field which generates a partial prediction of the intensities in the frame to be coded. We would like to ascribe a forward motion vector to each region in the previous image, within a prescribed search range, so as to minimize the mean squared error.

The first question to settle is whether to use a simple translational motion model to depict the region motion or to use a more complicated model such as the affine model. Since real motion of the region can be better captured using an affine model, we expect to obtain a better fit with such a model. On the other hand, we would have to pay the extra cost of sending additional rotational motion parameters for each region which might lead to a unacceptable overhead in very low bit-rate situations. In practice, we found that using an affine motion model is not justified for relatively simple sequences like Miss America. Since the primary application of our video compression strategy is in video teleconferencing which is characterized by very low bit rates and relatively simple motion, we chose to use translational motion vectors. In order to apply our algorithm to more complicated sequences, it would be optimal to use some strategy to switch between the affine motion model and the translational motion model on either a per frame or per region basis. Further details of such an optimal switching strategy are beyond the scope of this paper.

In order to find the translational motion vector for each region, we use a full search over all motion vectors within a specific range. To be more specific, a region in the segmentation of the previous frame is moved around within a search window in the current frame and that motion vector is chosen which yields the minimum mean squared error. In the cases where parts of the region overflow the edges of the frame to

be coded we do not use those parts of the region which fall outside the image to predict the intensities in the frame to be coded. Thus the translational motion vector that we find minimizes the mean squared error only within the frame to be coded.

If a region has a motion vector which maps it to such a location in the frame to be coded which is already covered due to the motion of other regions, transmitting the motion vector of that region to the decoder is not necessary. In our implementation, we have a boolean flag for each region which is either set to one or zero to depict whether the motion vector of the region is going to be transmitted to the decoder or not.

It is to be noted that the coarse motion field that we generate using this algorithm will have holes and overlapping regions. Prediction of the intensity values at such locations will be dealt with during the generation of the dense motion field in the next section.

C. Dense Motion Field

Pixels in the frame to be coded which cannot be accurately predicted using the coarse motion field (holes and overlapping regions) are themselves connected groups of pixels; each such connected group of ambiguous pixels is called a secondary region in our terminology. Obtaining a prediction of the intensities of the pixels in the secondary regions would lead to a pixel-wise dense motion field.

There are two coding strategies that we envisage using to predict the intensities of pixels in secondary regions. The first strategy involves using a backward motion vector for a given secondary region which minimizes the mean squared prediction error. Another strategy is to utilize a modified pel recursion algorithm to find the motion vectors for the pixels within the secondary regions. It is to be noted that the requirements for good performance of the pel recursion algorithm viz., applicability of a Taylor series approximation and good prior estimates for the motion vector are satisfied in our case, since we apply the pel recursion algorithm at only a few, well spread out locations within the image; the image intensities at other locations having been predicted using the coarse motion field. This is contrary to the application of the pel recursion algorithm in literature [7], [8] whence it has been used as a stand alone algorithm which performs all the motion prediction. The pel recursion algorithm will be discussed in more detail in Section IV.

Experimental evidence suggested that using a backward motion vector typically leads to a better motion prediction of the intensities when compared with the prediction obtained with the pel recursion algorithm to predict the intensities of pixels within secondary regions. However, the disadvantage of using a backward motion vector is that it has to be transmitted to the decoder. On the other hand the pel recursion algorithm does not need any information to be transmitted to the decoder. Thus, it would be better to use backward motion compensation for large secondary regions while resorting to the pel recursion algorithm for relatively small secondary regions. Another advantage of using the pel recursion only for small portions of the image is that we can avoid the excessive iterative computation and matrix inversion which pel recursion requires.

We classify secondary regions into two classes for compression purposes: secondary regions that are size-wise large are called class-one secondary regions and those which are size-wise small are called class-two secondary regions. The choice of an exact size threshold for affecting this classification will be discussed in Section IV.

Fig. 3 illustrates the process of obtaining the dense motion field from the coarse motion field. The dark regions correspond to the locations of pixels where the intensity prediction using the coarse motion field is either not available or is ambiguous. It is seen that backward motion compensation eliminates the large secondary regions while pel recursion takes care of the small (and typically widely scattered) secondary regions.

D. Rate Control for Motion Compensation

Since the application of choice for our video compression algorithm is teleconferencing, we assume that each frame is assigned the same number of bits (approximately) i.e., the bit rate is equally divided among various frames. Note that a strategy which distributes the bit rate unequally a mong different frames would lead to temporal fluctuations in the bit rate, which in turn leads to difficulties in real time display of the video at the decoder.

In order to convey the motion compensation information, we have to transmit the motion vectors generated during the process of construction of the coarse and dense motion fields to the decoder. As noted in Section II-C we do not need to transmit any information regarding the pel recursion motion estimation to the decoder. Also, segmentation is always carried out on the previously decoded frame; thus no segmentation information needs to be transmitted to the decoder. Hence our rate control strategy should make sure that the number of bits spent on transmitting motion related information to the decoder should be optimally balanced with the number of bits spent on residual coding (to be covered in Section III).

As noted earlier, segmentation of the previous frame with a multiscale segmentation algorithm gives us segmentation of the image at all photometric and geometric scales at which is structure present within the frame. In other words, for each value of the grey level homogeneity scale (σ_g), we can obtain a segmentation of the image. Using a fine scale of segmentation would mean that the region size would be small and hence the number of regions in the segmentation of the image would be large. This would mean that we need to transmit a large number of motion vectors if we segment the previous image at a fine scale. On the other hand, each region would be highly homogeneous thus leading to a small residual. The opposite would be the case if we segment the image at a coarse scale i.e., the amount of motion information would be small but the residual would be larger. This leads us to the conclusion that there is an optimal σ_g at which the segmentation of the previous image would be optimal. The rate control strategy thus reduces to choosing an optimal scale for segmenting the image.

Since motion compensation is applied to those frames of the video sequence where many of the objects within one frame

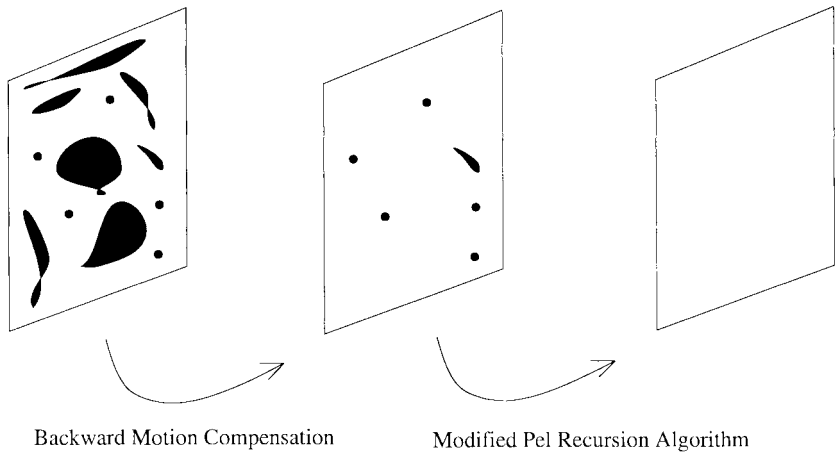


Fig. 3. Dense motion field generation: starting with the secondary regions in the frame to be coded (which are black), the figure illustrates that class-one secondary regions are predicted by backward motion compensation and then the remaining class-two secondary regions are predicted using the modified pel recursion algorithm.

are common to the next frame, one would expect that the value of σ_g at which we obtain optimal rate control would be similar for nearby frames. Thus, if we know the correct scale of segmentation for the previous motion compensated frame, we would expect that the correct scale for the next image would be either the same or slightly different. This observation leads to a drastic reduction in the amount of computation needed to find the optimal scale for transmitting the motion compensation information.

Although σ_g is in general a real number, only certain discrete values of σ_g are of interest as the segmentation of the image changes only at these values. In other words, if we start with a large σ_g (corresponding to a coarse scale of segmentation) and decrease it gradually, regions at the coarser scale would subdivide into finer regions only for certain discrete values of σ_g. The segmentation algorithm reported in [26] can automatically select these discrete values of σ_g at which there is a significant change in the segmentation of the image. Let $\sigma_{g,i}$, represent the discrete values of σ_g arranged in ascending order of magnitude with increasing values for the subscript i (larger σ_g implies coarser scale).

Given the grey level homogeneity scale $(\sigma_{g,k})$ at which the previous motion compensated image was segmented, we additionally check only the two adjacent values i.e., $\sigma_{g,k-1}$ and $\sigma_{g,k+1}$ at which there was a significant change in the segmentation of the image. For these three scales viz., $\sigma_{g,k-1}$, $\sigma_{g,k}$, $\sigma_{g,k+1}$ we can perform motion compensation and residual coding (as described in the next section) constrained by the number of bits allocated to the current frame and select that scale of segmentation which leads to the least mean squared error (and hence best PSNR). This approach cannot be used for the first motion compensated frame in the video sequence or for the first frame after a refresher frame. In these cases we do a full search through all possible σ_g. As stated before, the number of such scales is finite (typically 5–10 in practice) because the segmentation algorithm of [26] chooses the most appropriate scales for generating the hierarchical representation automatically. In practice, we found that an upper limit on the total number of motion vectors to be transmitted is also required to constrain the algorithm from transmitting too much motion information and little or no residual information. Although PSNR optimality would sometimes lead to the conclusion that transmitting little or no residual information leads to better quality video, we found that transmitting some residual is important in removing perceptually annoying artifacts.

It is to be noted that the above rate control strategy is not optimal in the sense that we do not fully explore all possible segmentation scales. However, we found that the tremendous increase in computational complexity which results from such processing is not validated by a corresponding improvement in perceptual quality.

III. Residual Coding

The residual coding scheme that we propose exploits the fact that the residual has been generated through a process of motion compensation in order to better code the residual. It will be seen that the proposed scheme can be extended to block motion compensation and any transform based residual coding scheme along similar lines, although we apply it only in the context of our hierarchical motion compensation strategy and a DCT based residual coding scheme in our paper. Since block motion compensation forms a basis for many popular video compression schemes in literature, we will explain how our scheme may be extended to that situation wherever necessary.

The key point which aids the proposed scheme is that we are compressing a residual generated by a *motion compensated video stream* and not still images as generally assumed by most residual coding schemes. Thus we expect that the residual image error will be concentrated in certain areas of the image. These areas can be predicted, given that we know the motion compensation scheme.

We conducted experiments on practical data which suggested that 80–90% of the residual energy is concentrated in *predictable locations* which form about 10% of the frame to be coded. These experiments were conducted using the region based motion compensation scheme and some of the standard sequences used for testing video teleconferencing applications

(such as Miss America and Claire). The predicted regions are found by using the rules outlined in the Section III-A.

The residual coding scheme proposed in this paper makes use of the above key observation to improve the performance of any transform based residual coder. We first generate an image mask predicting the location of residual energy concentration. Now we assume that the residual exists only in the predicted locations and is zero at other locations. So we are free to vary the values of the pixels at such locations, where residual is predicted to be zero, to obtain transform coefficients which lead to a better quality of the coded image. Since the decoder can also generate the same mask, decoding is possible. It may be noted that we *do not need* any further information to be transmitted in order to implement the proposed scheme.

The question to be answered is: how should the "free pixels" be chosen? For DCT or wavelets, the free pixels need to be chosen so that most of the energy is concentrated in the low frequencies. In other words we would like to use the freedom given in the choice of the free pixels to maximally pack the energy in the low frequencies of the transform domain. This problem has been addressed recently by us in a general context [27]. We had proposed an iterative algorithm which leads to an optimal choice of the coefficients, which we will use in the context of generating the residual.

A. Generating the Residual Energy Prediction Mask

The first step in implementing the proposed scheme would be to find the residual energy prediction mask locating the positions in the error image where the energy is concentrated. This depends on the motion compensation scheme employed, as different motion compensation schemes lead to a different distribution of residual error. Furthermore, the procedure for generating the prediction mask should be repeatable at the decoder without the transmission of any information.

As already observed, for a block based motion compensation scheme the pixels that cause most of the error lie at:

- the edges of the blocks with nonzero motion vectors. It is well known [28] that residual energy is large at the edges of a block whose motion has been predicted using full search block motion estimation;
- the image edge locations in the previous decoded frame. Block motion compensation does not compensate for a region boundary which passes within a block, thus leading to large errors at places where there are edges in the previous image. These edges can be detected using a simple edge detection algorithm like the Laplacian or Canny edge detectors;
- the predicted location of image edges in the frame to be coded (i.e., translated versions of image edge locations in the previous frame to the current frame).

For region based motion compensation, we can similarly conclude that most of the error would be concentrated at:

- the edges of the translated regions in the frame to be coded. Since the actual affine motion of the region was approximated with a translational motion vector, large error occurs at the region edges;
- the holes and other ambiguous regions which are filled in with the modified pel recursion algorithm during the formation of the dense motion field;
- the image edges in the previous frame also contribute to some error although to a lesser degree when compared with the block motion compensation strategy. Unlike the case of block based segmentation, where we needed to use a separate edge detection algorithm to locate these edges, we can use the segmentation of the previous image (which is used in motion compensation) to determine the edge locations directly.

$$\begin{matrix} 1&1&1&1&1&0&0&0\\ 1&1&0&0&0&1&1&1\\ 1&1&1&1&1&0&0&0\\ 1&1&1&1&1&1&0&0\\ 1&1&1&1&1&0&0&0\\ 1&1&1&1&1&1&1&0\\ 0&0&1&1&1&1&0&0\\ 0&0&0&0&0&1&1&1 \end{matrix} \qquad \begin{matrix} 1&1&1&1&0&0&0&0\\ 1&1&1&0&0&0&0&0\\ 1&1&0&0&0&0&0&0\\ 1&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0 \end{matrix}$$

(a) (b)

Fig. 4. Constraint sets for residual coding: (a) in the spatial domain the values where residual is predicted to be large (indicated by one) should not be changed and (b) in the DCT domain of the block, all coefficients lying outside a sub-block containing the DC coefficient should be zero (such coefficients which are constrained to be zero are indicated by a zero in the figure).

B. Iterative Algorithm

Once the mask predicting the error locations is found as described in the previous section, we can obtain the optimal values of the "free pixels" by using an iterative algorithm. For the sake of convenience, we assume that we are coding the residual of an 8×8 block using the popular DCT based residual coding scheme. Let the prediction mask for this block be $P(i,j)$, the residual be $R(i,j)$, and the operator $\mathcal{D}(\cdot)$ represent the 8×8 DCT operator, where i and j represent the coordinates of an arbitrary pixel within the block. $P(\cdot,\cdot)$ is a binary mask with one representing positions of high residual energy and zero representing positions of low residual energy. Two natural constraints which restrict the values of the free pixels are as follows (see Fig. 4).

1) The constraint set C_1 constrains the residual at pixels where the residual energy is high (as given by $P(\cdot,\cdot)$) to their actual value. Mathematically, $C_1 = \{I(\cdot,\cdot) : I(i,j) = R(i,j) \text{ if } P(i,j) = 1\}$.
2) The constraint set C_2 is aimed at packing most of the residual energy in the low frequency coefficients in the DCT domain. So we consider C_2 to be the set of all blocks with high frequency DCT coefficients equal to zero. To be precise, all the coefficients outside a sub-block [see Fig. 4(b)] containing the DC coefficient are constrained to be zero in the DCT of the block. This is reminiscent of the constraint used in [18]. Mathematically, $C_2 = \{I(\cdot,\cdot) : \mathcal{D}(I)(i,j) = 0 \text{ if } (i+j > M)\}$. The choice of a particular value for M was not found to be crucial and will be discussed in Section IV.

It is to be noted that the constraint sets C_1 and C_2 do not intersect in general. We are interested in searching for a solution in C_1 which is closest to C_2 in the mean squared sense. In other words, we would like to find that distribution of free pixels which maximally packs the residual energy in the low frequency coefficients in the DCT domain. Such a solution can be obtained by solving the problem in the projection on convex sets formalism [29]. Searching for a solution involves alternately projecting on to C_1 and C_2 and stopping the iterative process after a final projection on to the set C_1 upon convergence. We chose to stop the algorithm after a fixed number of iterations.

C. Switched Residual Coding Strategy

In the previous two sections we described a residual coding strategy which is applicable to each 8×8 block and a DCT based transform coder. Note that the primary assumption in designing this strategy is that the residual energy in places where $P(i,j) = 0$ is negligible i.e., most of the energy is concentrated at such positions as predicted by the mask. However this may not be satisfied in practice, as the reliability of the segmentation, which in turn determines reliability of the prediction mask, is itself dependent on the quality of the previously coded image which degrades with time if a refresher frame is not transmitted. Thus we obtain better performance if we use the usual DCT based decoding scheme for some blocks. In order to account for this problem we use a boolean flag to determine which scheme we are using to code each DCT block. Results comparing the advantage of the proposed method over the usual DCT based coding scheme for different quantization step sizes (assuming a uniform quantizer) are presented in Section V.

IV. Implementation Details

In this section we will specify the choice of various control parameters used in Sections II, III, as well as the scheme used for initializing and formulating the modified pel recursion algorithm of Section II-C. The reason for repeating the derivation of the pel recursion algorithm is that in our case, we exploit additional features which arise due to differences in problem setting. In Section IV-C, we will explain the decision statistic for sending refresher frames (I-frames) periodically.

A. Modified Pel Recursion Algorithm

We start by expressing the intensity value of the frame to be predicted as the value of the previous image at a location shifted by the motion vector estimate. Let $I_k(\cdot,\cdot)$ be the image to be predicted and $I_{k-1}(\cdot,\cdot)$ be the previous image, where the subscript k denotes the temporal position of the frame within the video sequence. If $d = (d_i, d_j)'$ is the motion vector at the location (i,j) within the image to be predicted, we can write

$$I_k(i,j) = I_{k-1}(i - d_i, j - d_j). \tag{1}$$

If (d_i^{l-1}, d_j^{l-1}) is the motion vector estimate at iteration $l-1$, we may write the displaced frame difference $E(i,j)$ as

$$E(i,j) = I_k(i,j) - I_{k-1}\big(i - d_i^{l-1}, j - d_j^{l-1}\big). \tag{2}$$

Now, using (1) we can write the displaced frame difference as

$$E(i,j) = I_{k-1}(i - d_i, j - d_j) - I_{k-1}\big(i - d_i^{l-1}, j - d_j^{l-1}\big). \tag{3}$$

By using the Taylor series expansion of the image $I_{k-1}(i,j)$ around the location $(i - d_i^{l-1}, j - d_j^{l-1})$, we obtain

$$\begin{aligned} &I_{k-1}(i - d_i, j - d_j) \\ &\quad = I_{k-1}\big(i - d_i^{l-1}, j - d_j^{l-1}\big) - \big(d_i - d_i^{l-1}, d_j - d_j^{l-1}\big) \\ &\qquad \times \nabla I_{k-1}\big(i - d_i^{l-1}, j - d_j^{l-1}\big) + v_{k-1}(i,j) \end{aligned} \tag{4}$$

where $v_{k-1}(i,j)$ is the approximation error term due to linearization. Thus, using (3) and (4) leads to the following recursive update in terms of the displaced frame difference $E(i,j)$

$$\begin{aligned} E^l(i,j) = &-\big(d_i - d_i^{l-1}, d_j - d_j^{l-1}\big) \\ &\times \nabla I_{k-1}\big(i - d_i^{l-1}, j - d_j^{l-1}\big). \end{aligned} \tag{5}$$

Assuming that the same motion vector works in some neighborhood of n pixels around the current pixel, we can formulate a set of equations that the motion vector needs to satisfy

$$Z^l = G^{l-1}\big(d^l - d^{l-1}\big) + V^{l-1} \tag{6}$$

where $Z^l = [E^l(i_1,j_1)E^l(i_2,j_2)\ldots E(i_n,j_n)]'$, $G^{l-1} = [\nabla I_{k-1}(i_1 - d_i^{l-1}, j_1 - d_j^{l-1})\nabla I_{k-1}(i_2 - d_i^{l-1}, j_2 - d_j^{l-1})\ldots \nabla I_{k-1}(i_n - d_i^{l-1}, j_n - d_j^{l-1})]'$ and V^{l-1} is the error vector at the $(l-1)$th iteration.

It is a straight forward exercise to solve the above equation using stochastic linear estimation [30], to obtain the biased minimum variance estimate as

$$d^l - d^{l-1} = \Big(G^{l-1\prime} Q_v^{l-1^{-1}} G^{l-1} + Q_u^{l-1}\Big)^{-1} G^{l-1\prime} Q_v^{l-1^{-1}} Z^l. \tag{7}$$

The usual practice is to assume that Q_v^{l-1} and Q_u^{l-1} (the covariance matrices) to be of the form (variance) $\times$ (identity matrix). At this juncture it may be pointed out that in our case we can specify a different form for Q_v^{l-1}, since the pixels in the neighborhood can come through different processes viz., forward motion compensation, backward motion compensation, or pel recursion. Since the reliability of the estimates in these three processes is quite different, we expect them to have different reliability in estimation. Thus Q_v^{l-1} can be thought of as a general diagonal matrix, rather than a scaled identity matrix. The per-frame estimates of the three variances which determine Q_v^{l-1} can be transmitted to the decoder with a negligible increase in bit rate. In our current implementation, we restrict ourselves to using the scaled identity matrix approximation for Q_v^{l-1} to avoid additional computational complexity.

Typical algorithms for pel recursive estimation use only pixels from a *causal neighborhood* of the current pixel. This is necessitated by the fact that all the pixels in the current frame were being processed using the pel recursion algorithm. In preliminary versions of the algorithm we used a causal window to obtain the neighboring motion values to be used within our algorithm. However, we note that no such restriction is

necessary in our case. In fact, we can use the nine nearest neighbors whose motion vectors have already been determined by previous processing, looking in a 5×5 window containing the current pixel at its center.

Pel recursion leads to fractional pixel accuracy motion vectors. Since the previous image from which motion compensation is discrete, we need an interpolating mechanism. Nosratinia and Orchard [8] show how to find the optimal interpolant by making the assumption that the optimal interpolant will have the same form in a causal neighborhood. It is again to be noted that causality is not *necessary* in our setting, although one may use a causal neighborhood for processing simplicity. We note that each motion vector, if it is fractional, refers to four nearest intensities in the previous frame. If we are processing N pixels in a causal neighborhood, let A represent the $4 \times N$ matrix of intensity values from the previous frame, each row of the matrix corresponding to the four nearest intensities pointed to by the motion vector corresponding to each pixel. Let ϕ be the 1×4 vector corresponding to the optimal interpolant which is to be determined. Finally, let h be the vector of intensities of the N pixels in the current frame. Thus, finding the optimal interpolant reduces to solving the equation $A\phi = h$. Thus, the least squares solution is found by using the pseudo-inverse of A and is given by $\hat{\phi} = (A'A)^{-1}A'h$.

B. Control Parameters

In Sections II and III we differed the specification of a few control parameters which are important for implementation.

1) Maximum number of motion vectors (Section II-C): We regulate the maximum number of motion vectors so that we ensure transmission of some amount of residual information. Although we optimize the amount of motion information with the amount of residual transmitted within a PSNR setting, we found that transmitting some residual information is important in reducing perceptual artifacts although it might not be optimal from a PSNR perspective. To this end, we limit the maximum amount of motion information transmitted to 75% of the bit rate.
2) Size threshold for classification of secondary regions (Section II-C): We found that most of the secondary regions are typically small or large; the number of secondary regions with medium size is small. Thus a choice for the classification threshold is not crucial. We avoid computationally expensive optimization strategies and choose a threshold of 16 pixels i.e., all the regions which are smaller than 16 pixels are classified into class-two secondary regions and those larger than 50 pixels are classified into class-one secondary regions. It is to be noted that this threshold is dependent on the resolution of the frame. The specification of 16 corresponds to QCIF resolution. The threshold needs to be scaled appropriately for higher resolution video formats.
3) Quantization step size for residual coding (Section III): We found that larger quantization step sizes work better at lower bit rates. Experimental results for residual coding that will be shown in Section V correspond to two different quantization step sizes of 16 and 32. Later on, we will show overall algorithm performance for only a step size of 16.
4) Low frequency threshold (Section III-B): The parameter M $(1 \leq M \leq 8)$ was used to determine which subset of coefficients of a block of 8×8 DCT were classified as low frequency coefficients. We found that $M = 2$ or $M = 3$ typically resulted in good performance. For the experimental results presented in this paper we used $M = 3$.

C. Refresher Frames

For video teleconferencing applications, a refresher frame needs to be sent at shot changes or if the number of motion compensated frames exceeds a threshold. The threshold on the number of consecutive motion compensated frames without a refresher is set to 128. We use a histogram based thresholding criterion to decide when a shot change has occurred. In this context we define the action measure between the kth and $(k+1)$th video frames to be

$$A(k) = \sum_i |h_k(i) - h_{k+1}(i)|^2$$

where h_k is the histogram of the kth frame. A shot change is said to occur whenever the action measure crosses a particular threshold and a refresher frame is sent. However, the value of threshold needs to adapt to the shot content. Let μ and σ be the mean and variance of the action measure, when the statistics are computed over the last N frames ($N = 10$ in practice). The threshold is then set to $\mu + \alpha * \sigma$. Note that a large α leads to missed shot detections while a small value for α leads to many false alarms. Following [31], we use a value of five for α, which was found to work well in practice. Note that this problem of shot detection can also be formulated in a Neyman–Pearson hypothesis testing frame work, assuming Rayleigh distributions with differing variances for the action measure under different hypothesis. However, we found that the improvements were marginal, if any, using this approach. We attribute this to the fact that the simple adaptive thresholding scheme results in accurate shot detection in most cases.

V. Results

A. Comparison with a Generic Block Based Coder

In this section we will compare the performance of the proposed coder with a generic block based coder as used in the H.261 or the H.263 standards [4], [5]. All performance comparison is performed on the luminance (Y) component of the video frames. In order to make an objective comparison, we used the same quantization strategies to quantize DCT coefficients for both the coders. We used a uniform quantizer with a quantization step size of 16 for the AC coefficients and a step size of 1 for the DC coefficient. The Huffman codes for motion vectors and DCT coefficients were the same for both the coders. These coding tables were derived directly from

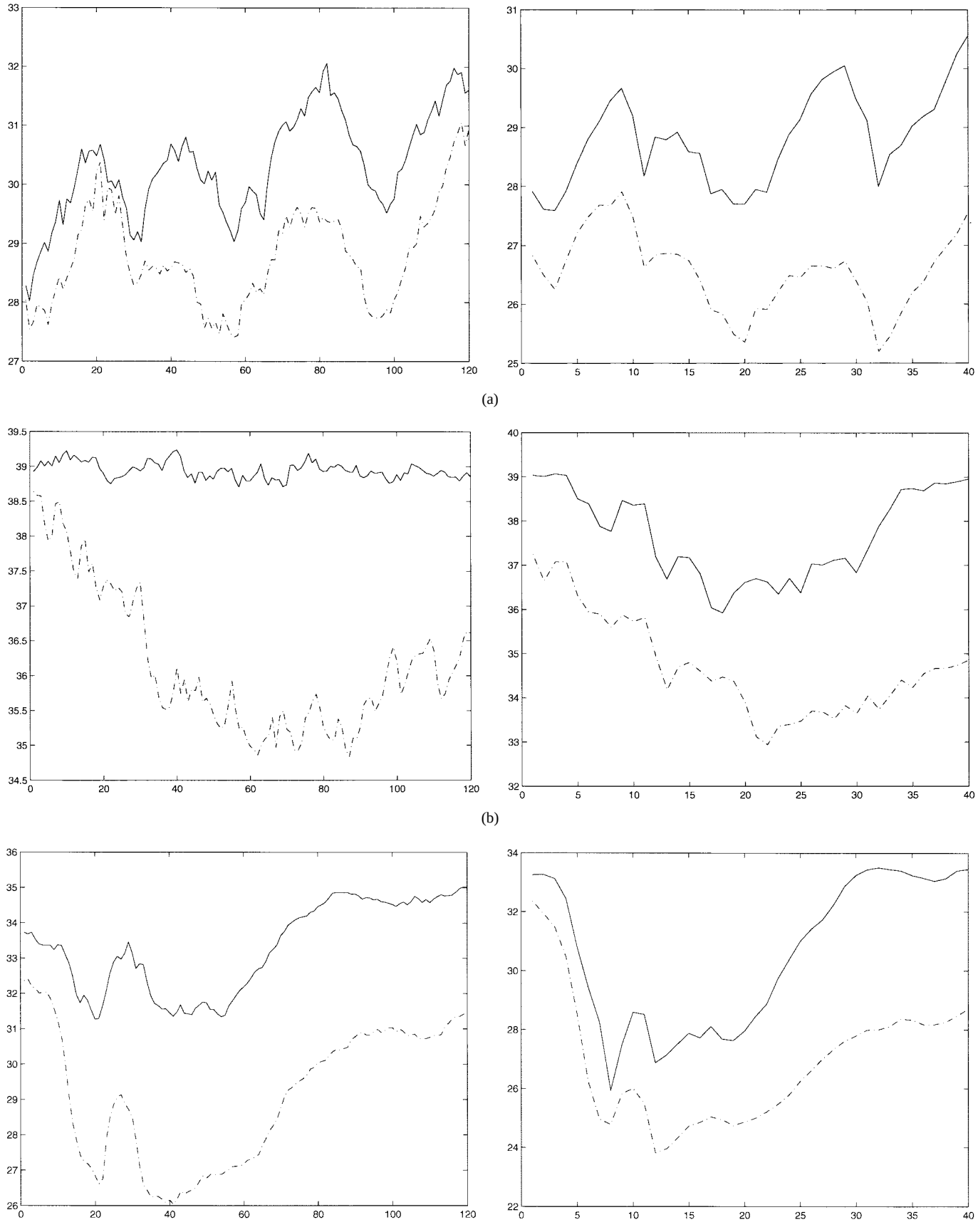
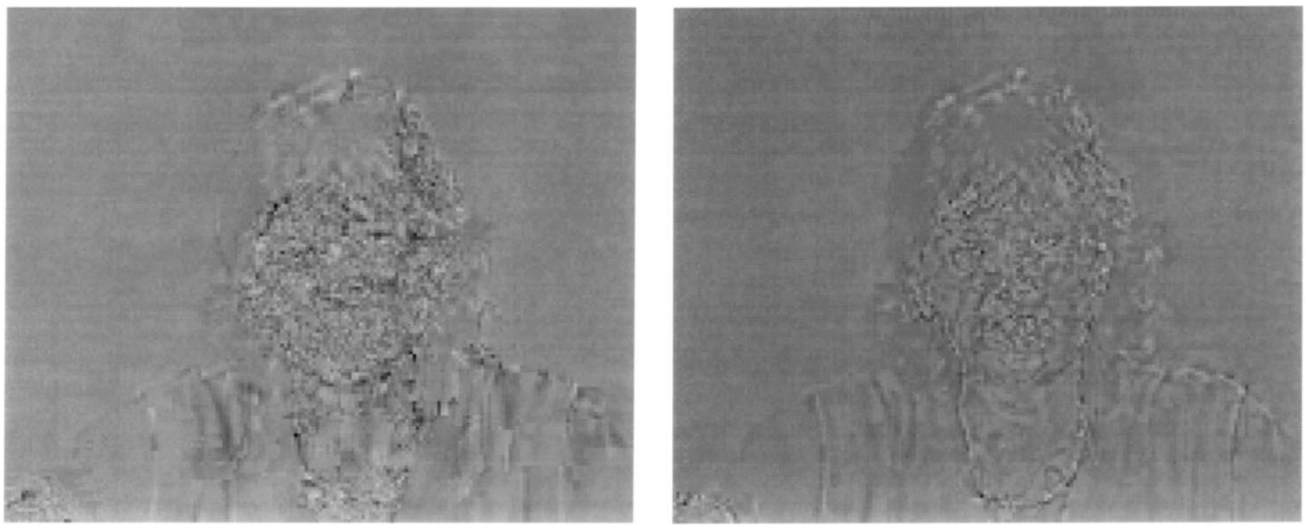


Fig. 5. Comparative results at 1280 bits per luminance frame. X axis: frame number and Y axis: PSNR. The graphs on the right corresponding to a frame rate of 7.5 Hz (every fourth frame is coded) and those on the left correspond to a frame rate of 30 Hz. Dashed line: block based scheme. Solid line: proposed region based scheme. (a) Car phone sequence, (b) Miss America sequence, and (c) Susie sequence.



(a) (b)

Fig. 6. Residual images from Miss America sequence ($\mathcal{M} = 2$): (a) block based approach and (b) region based approach.

those proposed in the H.261 standard and therefore are more optimal for the block based coder. Both coders used integer precision motion vectors for the following experiments. We found that using a half-pixel precision motion vector yielded similar improvements in performance in both coders. Boolean flags for the proposed compression strategy were directly coded without any compression. No refresher frames were sent during the encoding of the video sequences.

The frame bit rate was held (approximately) fixed for both the coders at 1280 bits. This bit rate corresponds to a bit rate of 9.6 kbps if every fourth frame is coded and a bit rate of 38.4 kbps if all the frames are coded, which are reasonable for video teleconferencing applications. The reason for using fixed bit rate for each frame, as explained before, is that variable bit rate (at constant PSNR quality) tends to produce fluctuations in bit rate which are unacceptable within a video teleconferencing setup.

Fig. 5(a)–(c) shows the comparative performance of the coders for typical video teleconferencing sequences (carphone, Susie, and Miss America). In each case the graph on the right corresponds to a frame rate of 30 Hz and the graph on the left to a frame rate of 7.5 Hz (every fourth frame). It is seen that the proposed coder out performs the block based coder by about 2–3 dB consistently. Table I summarizes the average quality for these cases.

In order to display images, we magnified them (since the image resolution is small) by replacing each pixel with a block of $M \times M$ for magnification by a factor of $\mathcal{M}$ (pixel replication). Wherever necessary, we also display the original image to allow for objective comparison in spite of the filtering introduced by the Laser printer. Fig. 6 shows the difference images ($\mathcal{M}$(magnification) $= 2$) for a frame from the miss America sequence. Fig. 7 shows a part of the Miss America frame for both the approaches as well as the original ($\mathcal{M} = 4$). These images were obtained by coding at 7.5 Hz (every fourth frame). Fig. 8 shows a frame of the Susie sequence for both the approaches as well as the original at 7.5 Hz. Fig. 9 shows a frame of Susie sequence for both the approaches at 30 Hz.

TABLE I
AVERAGE PSNR FOR CODING THE Y-COMPONENT OF THE SEQUENCES WITH THE PROPOSED AND BLOCK BASED APPROACHES AT 1280 BITS/FRAME. (a) CODE EVERY FOURTH FRAME (7.5 Hz) AND (b) CODE ALL FRAMES (30 Hz)

	Proposed	Block based
Miss America	38.1 dB	35.4 dB
Susie	31.33 dB	28.17 dB
Car Phone	29.03 dB	27.06 dB
Claire	36.26 dB	34.55 dB

(a)

	Proposed	Block based
Miss America	39.06 dB	36.07 dB
Susie	33.67 dB	30.10 dB
Car Phone	30.14 dB	28.17 dB

(b)

The overall perceptual improvement due to the proposed approach is quite evident. The improvement in performance for the 7.5 Hz case is much more pronounced than for the 30 Hz case. This is to be expected, since a gain of 2 dB in PSNR at higher quality does not lead to as much perceptual improvement as the same PSNR differential at a lower quality.

B. Comparison of the Residual Coding Schemes

We also present results comparing our adaptive residual coding scheme with the usual block DCT based coding scheme. As mentioned in Section III, both the schemes transmit quantized DCT coefficients. Our method gains over the generic DCT based coding scheme by cleverly utilizing the fact that the positions of high residual concentration can be pre-predicted. Since the prediction mechanism can break down for some blocks, we propose to switch coding with our strategy with the DCT based scheme (one bit needs to be transmitted). Such a coder always performs better than the baseline block

(a) (b) (c)

Fig. 7. Part of a frame from Miss America sequence ($\mathcal{M} = 4$): (a) original, (b) block based approach, and (c) region based approach.

(a) (b) (c)

Fig. 8. Frame from Susie sequence ($\mathcal{M} = 2$): (a) original, (b) block based approach (7.5 Hz), and (c) region based approach (7.5 Hz).

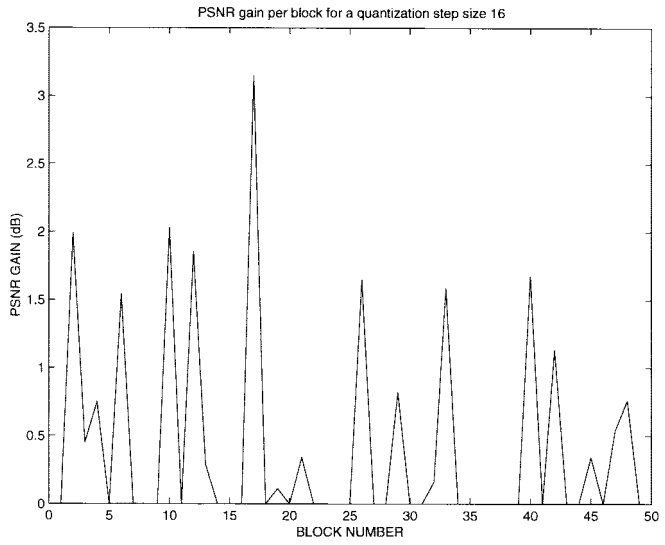
(a) (b)

Fig. 9. Frame from Susie sequence ($\mathcal{M} = 2$): (a) block based approach (30 Hz) and (b) region based approach (30 Hz).

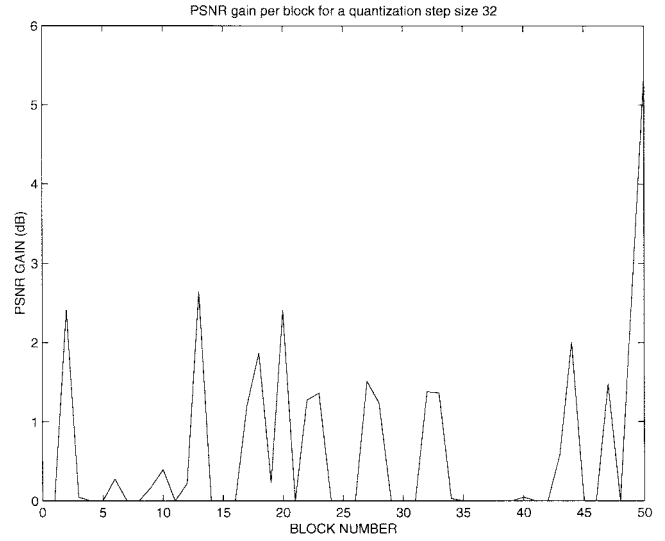
DCT scheme. Fig. 10 shows the improvement (in dB PSNR) over the generic coder, when the quantization step size of AC coefficients is 16 and 32. Note that this improvement was obtained *only due to improvement in residual coding*. In other words, both coders are coding exactly the *same residual blocks*. The block numbers are not correlated between the two graphs. No refresher frame was sent in the simulations and the blocks are taken from the first seven frames (approximately) for step size 16 and the first five frames (approximately) for step size 32. It may be observed that the average advantage due to the proposed method decreases as block number increases. This occurs due to the fact that prediction degrades as the frame number (and block number) increase due to lack of refresher frames.

VI. Conclusion

In this paper, multiscale image segmentation is used to develop a video compression algorithm for low bit-rate ap-



(a)



(b)

Fig. 10. Comparative results for residual coding: the PSNR gain of the proposed method over normal DCT based coding scheme per block number. Quantization step sizes are (a) 16 and (b) 32.

plications. The key ideas of the scheme presented include the following: a) the algorithm uses multiscale segmentation and selects the segmentation at a scale which is optimal for compression; b) a novel method is introduced to deal with occluded regions which normally degrade the performance of region based techniques; c) pel recursion and linear prediction methods are employed to fine tune motion estimation; d) region segmentation is performed on the *previously decoded frame* (so we do not need to encode any segmentation information); and e) residual coding exploits the fact that locations of high residual energy concentration occupy small portions of the image and are *a priori* predictable. A fusion of these important ideas leads to a gain of about 2–3 dB in PSNR over the block matching algorithm for a variety of head-and-shoulder sequences using a fully functional video coder (when the bit rate is constrained to be the same for both schemes).

REFERENCES

[1] R. Forchheimer and T. Kronander, "Image coding: From waveforms to animation," *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. 37, pp. 2008–2023, Dec. 1989.

[2] MPEG Video Group, "Coding of moving pictures and associated audio for digital storage media at up to about 1.5 mbit/s," *Int. Standard ISO/IEC 11172 (MPEG-1)*, Nov. 1992.

[3] MPEG Video Group, "Generic coding of moving pictures and associated audio information," *Int. Standard ISO/IEC 13818 (MPEG-2)*, Nov. 1994.

[4] International Telecommunication Union, "Draft recommendation H.261—Video codec for audiovisual services at $p \times 64$ kbit/s," 1990.

[5] International Telecommunication Union, "Draft recommendation H.263—Video coding for low bit rate communication," Dec. 1995.

[6] G. M. Schuster and A. K. Katsaggelos, "A video compression scheme with optimal bit allocation between segmentation, motion, and residual error," *IEEE Trans. Image Processing*, vol. 6, pp. 1487–1502, Nov. 1997.

[7] J. Biemond, L. Looijenga, and D. E. Boekee, "A pel-recursive Wiener-based displacement estimation algorithm for interframe image coding applications," in *Proc. SPIE—Visual Commun. Image Processing II*, Cambridge, MA, Oct. 1987, vol. 845, pp. 424–431.

[8] A. Nosratinia and M. T. Orchard, "Discrete formulation of pel-recursive motion compensation with recursive least squares updates," in *Proc. IEEE Int. Conf. Acoustics, Speech Signal Processing*, Minneapolis, MN, Apr. 1993, vol. 5, pp. 229–232.

[9] M. T. Orchard, "New pel-recursive motion estimation algorithms based on novel interpolative kernels," in *Proc. SPIE—Visual Commun. Image Processing'92*, Boston, MA, 1992, vol. 1818, pp. 85–96.

[10] N. Nandhakumar and J. K. Aggarwal, "On the computation of motion from a sequence of images—A review," in *Proc. IEEE*, vol. 76, pp. 917–935, Aug. 1988.

[11] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, pp. 185–203, 1981.

[12] R. Rajagopalan, M. T. Orchard, and M. T. Brandt, "Motion field modeling for video sequences," *IEEE Trans. Image Processing*, vol. 6, pp. 1503–1516, Nov. 1997.

[13] A. Kaup, "Adaptive constrained least squares restoration for removal of blocking artifacts in low bit rate video coding," in *Proc. IEEE Int. Conf. Acoustics, Speech Signal Processing*, Munich, Germany, Apr. 1997, vol. 4, pp. 2913–2916.

[14] R. Kutka, A. Kaup, and M. Hager, "Quality improvement of low data-rate compressed video signals by pre- and postprocessing," in *Proc. SPIE—Digital Compression Technologies Syst. Video Commun.*, Berlin, Germany, Oct. 1996, vol. 2952, pp. 42–49.

[15] M. T. Orchard, "Predictive motion-field segmentation for image sequence coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 3, pp. 54–70, Feb. 1993.

[16] K. W. Stuhlmuller and B. Girod, "Motion segmentation for region-based coding," in *Proc. IEEE Int. Conf. Image Processing Applicat.*, Dublin, Ireland, July 1997, vol. 2, pp. 650–654.

[17] Y. Yokoyama, Y. Miyamoto, and M. Ohta, "Very low bit rate video coding using arbitrarily shaped region-based motion compensation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 5, pp. 500–507, Dec. 1995.

[18] H. H. Chen, M. R. Civanlar, and B. G. Haskell, "A block transform coder for arbitrarily shaped image segments," in *Proc. IEEE Int. Conf. Image Processing*, Austin, TX, Nov. 1994, vol. 1, pp. 85–89.

[19] P. Salembier, F. Marques, M. Pardas, J. R. Morros, I. Corset, S. Jeannin, L. Bouchard, F. Meyer, and B. Marcotegui, "Segmentation-based video coding system allowing the manipulation of objects," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, pp. 60–74, Feb. 1997.

[20] D. P. de Garrido, L. Ligang, and W. A. Pearlman, "Conditional entropy-constrained vector quantization of displaced frame difference subband signals," in *Proc. IEEE Int. Conf. Image Processing*, Austin, TX, Nov. 1994, vol. 1, pp. 745–749.

[21] N. Ahuja, "A transform for the detection of multiscale image structure," in *Proc. IEEE Comput. Soc. Conf. Comput. Vision Pattern Recognition*, New York, June 1993, pp. 780–781.

[22] N. Ahuja, "A transform for multiscale image segmentation by integrated edge and region detection," *IEEE Trans. Pattern Analy. Machine Intelligence*, vol. 18, pp. 1211–1235, Dec. 1996.

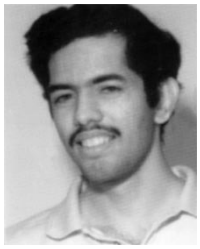
[23] P. Salembier, L. Torres, F. Meyer, and C. Gu, "Region-based video coding using mathematical morphology," in *Proc. IEEE*, vol. 83, pp. 843–57, June 1995.

[24] T. Ebrahimi, "A new technique for motion field segmentation and coding for very low bit rate video coding applications," in *Proc. IEEE Int. Conf. Image Processing*, Austin, TX, Nov. 1994, vol. 2, pp. 433–437.

[25] F. Bartolini, V. Cappellini, A. Mecocci, and R. Vagheggi, "A segmentation based motion compensated scheme for low rate video coding," in *Proc. IEEE Int. Conf. Image Processing*, Austin, TX, Nov. 1994, vol. 2, pp. 457–461.
[26] M. Tabb and N. Ahuja, "Multiscale image segmentation by integrated edge and region detection," *IEEE Trans. Image Processing*, vol. 6, pp. 642–655, May 1997.
[27] K. Ratakonda and N. Ahuja, "Discrete multidimensional linear transforms over arbitrarily shaped supports," *IEEE Int. Conf. Acoustics, Speech, Signal Processing*, Munich, Germany, Apr. 1997, vol. 4, pp. 3041–3044.
[28] M. T. Orchard and G. J. Sullivan, "Overlapped block motion compensation: An estimation-theoretic approach," *IEEE Trans. Image Processing*, vol. 3, pp. 693–709, Sept. 1994.
[29] B. Javidi and J. L. Horner, Eds., *Real Time Optical Information Processing*. New York: Academic, 1994.
[30] Luenberger, *Optimization by Vector Space Methods*. New York: Wiley, 1994.
[31] A. Hanjalic, M. Ceccarelli, R. Lagendijk, and J. Biemond, "Automation of systems enabling search on stored video data," in *Proc. SPIE—Storage Retrieval Image Video Databases*, San Jose, CA, Feb. 1997, vol. 3022, pp. 427–438.

Seung Chul Yoon received the B.S. and the M.S. degrees in electronics engineering from Yonsei University, Seoul, Korea, in 1990 and 1992, respectively. He is currently working toward the Ph.D. degree at the Beckman Institute for Advanced Science and Technology of the University of Illinois at Urbana-Champaign.

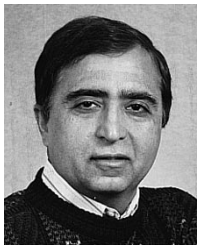
He is presently a Research Assistant at the Beckman Institute for Advanced Science and Technology of the University of Illinois at Urbana-Champaign. In the summer of 1999, he was a summer intern at the HRL Laboratories, Malibu, CA. His research interests include digital video processing, image processing, and computer vision.



Krishna Ratakonda received the B.Tech. degree with honors in electronics engineering from the Institute of Technology, Banaras Hindu University, India, in 1994. He received the M.S. degree and the Ph.D. degrees in electrical engineering from the University of Illinois at Urbana-Champaign in 1996 and 1999, respectively.

Since August 1998, he has been with the IBM T. J. Watson Research Center, Yorktown Heights, NY. In the summer of 1997, he was a summer intern at the Sharp Labs of America Limited, Camas, WA. His research interests are in digital video processing, digital image acquisition and processing, and computer vision.

Dr. Ratakonda received the Robert Chien Award for Outstanding Graduate Research in the Department of Electrical and Computer Engineering at UIUC (1998), a Sundaram Seshu Fellowship (1996), the CS-AI Award for Academic Excellence (1995), the Best Paper Award of IEEE Regional Student Paper Ccontest (1994), National Merit Scholar (1990–94), and State Merit Scholar (1988–90).



Narendra Ahuja (M'79–SM'85–F'92) received the B.E. degree with honors in electronics engineering from the Birla Institute of Technology and Science, Pilani, India, in 1972, the M.E. degree with distinction in electrical communication engineering from the Indian Institute of Science, Bangalore, India, in 1974, and the Ph.D. degree in computer science from the University of Maryland, College Park, in 1979.

From 1974 to 1975 he was Scientific Officer in the Department of Electronics, Government of India, New Delhi. From 1975 to 1979 he was at the Computer Vision Laboratory, University of Maryland, College Park. Since 1979 he has been with the University of Illinois at Urbana-Champaign where he is currently a Professor in the Department of Electrical and Computer Engineering, the Coordinated Science Laboratory, and the Beckman Institute. His interests are in computer vision, robotics, image processing, image synthesis, sensors, and parallel algorithms. His current research emphasizes integrated use of multiple image sources of scene information to construct three-dimensional descriptions of scenes; the use of integrated image analysis for realistic image synthesis; parallel architectures and algorithms and special sensors for computer vision; and use of the results of image analysis for a variety of applications including visual communication, image manipulation, video retrieval, robotics, and scene navigation. He has co-authored the books *Pattern Models* (New York: Wiley, 1983) and *Motion and Structure from Image Sequences* (New York: Springer-Verlag, 1992), and co-edited the book *Advances in Image Understanding*, (Piscataway, NJ: IEEE Press, 1996).

Dr. Ahuja received the 1999 Emanuel R. Piore award of the IEEE and the 1998 Technology Achievement Award of the International Society for Optical Engineering. He was selected as Associate (1998–99) and Beckman Associate (1990–91) in the University of Illinois Center for Advanced Study. He received University Scholar Award (1985), Presidential Young Investigator Award (1984), National Scholarship (1967–72), and President's Merit Award (1966). He is a Fellow of the American Association for Artificial Intelligence, International Association for Pattern Recognition, Association for Computing Machinery, American Association for the Advancement of Science, and International Society for Optical Engineering. He is a member of the Optical Society of America. He is on the editorial boards of the journals IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE; *Computer Vision, Graphics, and Image Processing*; *Journal of Mathematical Imaging and Vision*; *Journal of Pattern Analysis and Applications*; *International Journal of Imaging Systems and Technology*; and *Journal of Information Science and Technology*; and a guest co-editor of the *Artificial Intelligence Journal*'s special issue on vision.