

Interframe Coding of Magnetic Resonance Images

Aria Nosratinia, *Student Member, IEEE*, Nader Mohsenian, *Member, IEEE*, Michael T. Orchard, *Member, IEEE*, and Bede Liu, *Fellow, IEEE*

Abstract— This paper presents a new inter-frame coding method for medical images, in particular magnetic resonance (MR) images. Until now, attempts in using inter-frame redundancies for coding MR images have been unsuccessful. We believe that the main reason for this is twofold: unsuitable inter-frame estimation models and the thermal noise inherent in MRI. The inter-frame model used in this paper is a continuous affine mapping based on (and optimized by) deforming triangles. The inherent noise of MRI is dealt with by using a median filter within the estimation loop. The residue frames are quantized with a zero-tree wavelet coder, which includes arithmetic entropy coding. This particular method of quantization allows for progressive transmission, which aside from avoiding buffer control problems is very attractive in medical imaging applications.

Keywords— Image Compression, Interframe Coding, Warping, Wavelet Transform

I. INTRODUCTION

RECENT years have seen a geometric increase in the volume of medical image data generated in hospitals. Advancements in medical imaging technology and the wide availability and usage of many imaging modalities has created a greater need than ever for efficient coding and compression techniques. In an average sized hospital, many tera-bytes of digital imaging data are generated every year, almost all of which has to be kept and archived. Technologies for storage and retrieval have not kept up with the furious pace of data generation. No less important is the problem of transmission for off-site consultation and diagnosis, and related applications.

This especially holds true with 3-D medical data formed by image sequences. The sheer size of these sequences makes it impractical to use and share this information over existing network bandwidths without efficient compression. This paper proposes a method for inter-frame coding of volumetric medical data, in particular MRI data.

Medical image sequences are data sets representing 3D sampling of some attribute of objects. From a fundamental viewpoint, dependencies along the third dimension can be exploited for coding gain just as well as those in the two dimensional plane of each image. However, many experiments seem to indicate that in practice, there is not much to be gained by exploiting depth dependencies. Roos and Viergever [1], [2], [3], [4], in their thorough studies of

lossless compression of medical data, observed that parameterizing tested methods in the third dimension was not very effective in decorrelation. Chen et al [5] use a 3-D Discrete Cosine Transform (DCT) transformed coding scheme, and report some success in coding of CT data, but find no advantage in using 3-D DCT coding of MRI data compared to intra-frame coding.

An alternative approach is to use “motion” compensated estimation to reduce the redundancies.¹ The most obvious and common of these methods is the so-called block-matching algorithm (BMA) [6], which is borrowed from video coding. Lee et al [7] explored a method where block-matching motion compensation residues are transformed through 2-D DCT, and then the transform coefficients are quantized. While they report coding gains with CT data, they state that the same approach is ineffective with MRI, where they believe the low correlation of the signal is at fault.

Indeed our own early experiments with displacement-based (motion) coding of MRI was less than encouraging. We applied many of the variations of standard block-matching, including exhaustive search strategies for motion vectors, and fractional pixel motion accuracy. We found that, not only did these methods not achieve good compression, their rate-distortion performance was in fact *inferior* to intra-frame coding. Note that this is not necessarily surprising: the paradigm of displacement-based (motion) estimation is built on the assumption that displaced frame differences have lower energy than the original frames. One can imagine signal constructions where the formation of block frame differences actually *increases* the energy. Such cases can arise when block motions do not effectively describe existing inter-frame relationships, or when individual frames have substantial uncorrelated components.

A second look into the structure and origins of MRI data indicates that, although a significant amount of redundancy exists between successive frames of MR data, the structure of this cross dependence is more complicated than temporal (video) sequences. In the case of video sequences, a simple translational model often can characterize the interframe dependencies to a fairly good degree of accuracy, as attested by the large body of work in that area [6]. But such a model is not necessarily appropriate for an image sequence consisting of consecutive spatial slices of an object, where a large amount of deformation is present from frame

Aria Nosratinia is with the Beckman Institute for Advanced Science and Technology, and Coordinated Science Laboratory, at the University of Illinois at Urbana-Champaign, Urbana, IL. He is currently a visiting scholar at the Electrical Engineering Department, Princeton University, Princeton, NJ

Nader Mohsenian is with the IBM Microelectronics Advanced Digital Video Laboratories, Endicott, NY

Michael Orchard and Bede Liu are with the Electrical Engineering Department, Princeton University, Princeton, NJ

¹We recognize that the use of the term “motion” for describing dependencies in volumetric data is, strictly speaking, not entirely satisfactory, and perhaps other terms such as “displacement” are preferable. However, volumetric coding has significant overlap with the field of video coding, where the terminology is well developed and mature. Henceforth, we will use the term “motion” interchangeably with “displacement”

to frame. Furthermore, MRI data contain a large quantity of noise, which is uncorrelated from frame to frame. We believe that the use of unsuitable models, along with the noise amplification inherent in motion compensation, is mainly responsible for the (sometimes surprising) lack of success so far in interframe coding of MR images.

This paper proposes to use a more advanced motion model for the coding of MRI sequences. We propose to use warping, also known as Control Grid Interpolation (CGI), for inter-frame estimation [8], [9]. Unlike block-matching, which generates piecewise-constant motion fields (with discontinuities), warping generates continuous, piecewise linear motion fields. Such motion fields are much more appropriate for 3-D data, where images represent cross sections of objects. Furthermore, we propose to use a median filter within the motion estimation loop, to remove the noise amplification problem.

The residue frames are then coded through a zero-tree wavelet coder. Wavelet analysis [10], and especially zero-tree wavelet coding [11] has attracted much attention in recent years. The rise of wavelet methods is due mostly to their inherent effectiveness in modeling and coding many useful signals, in particular images. Wavelet coding is particularly suited to our choice of motion estimation and compensation, since our continuous motion fields generate smoother residue fields, which in turn can be coded more efficiently through the coefficients of the lapped wavelet basis functions. We note that, to our knowledge, this is the first instance where warping motion estimation has been employed with zerotree wavelet quantization in any inter-frame coder.

The inter-frame estimation part of the algorithm is explored in Section II. We revisit the concept of warping motion estimation, and establish our method of achieving it through matching of triangular patches in the present frame to the past frame. The in-loop filter and its role in the estimation process is explained. We also address the computational issues in this particular motion compensated estimator, and offer means of trading minimal amounts of performance for considerable computational savings. In Section III, the coding of estimation residues is discussed, where we outline the zero-tree wavelet algorithm we use. Section IV presents experimental results, and Section V offers closing thoughts and conclusions.

II. INTER-FRAME ESTIMATION

The interframe estimation used in our method is based on an affine spatial transform. Spatial transforms are by no means new. Polynomial and rational spatial transforms have been used in the field of computer graphics for many years [12]. In this section, we briefly present the framework for spatial transforms.

A general spatial transformation defines a geometrical relationship (usually one-to-one) between the input and output frames. The mapping can be expressed by specifying either the output coordinate system in terms of that

of the input, or vice versa.

$$[x, y]^T = [X(u, v), Y(u, v)]^T \quad (1)$$

$$[u, v]^T = [U(x, y), V(x, y)]^T \quad (2)$$

where $[u, v]^T$ and $[x, y]^T$ are respectively the coordinates in the input and output images. $X(\cdot, \cdot)$ and $Y(\cdot, \cdot)$ map the input to the output, and hence $\{X, Y\}$ is referred to as the *forward* mapping. In the same manner, $\{U, V\}$ is called the *backward* or *inverse* mapping. We use a backward affine mapping for our interframe estimator.²

Many of the more useful spatial transforms can be characterized by a 3×3 matrix.

$$\mathbf{T}_1 = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \quad (3)$$

with

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \mathbf{T}_1 \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (4)$$

Without loss of generality, the component in the third dimension can be ignored, since we are interested in transforms in 2-D only.

An affine transformation is characterized by

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \quad (5)$$

The affine transform has six degrees of freedom and can be uniquely determined by specifying the coordinates of three non-colinear point in the output frame. A given affine transformation is invertible if the images of three non-colinear points under the transform are also non-colinear. Affine transformation preserves straight lines and parallelism.

Since no more than three points are needed to uniquely specify an affine mapping, this class of transformations is simple, and its computational load is light compared to, for example, perspective transform [12]. It essentially maps an input triangle into an arbitrary triangle at the output.

The affine transform can be parameterized by the differential positions of three non-colinear points. Let (u_k, v_k) and (x_k, y_k) for $k = 0, 1, 2$ be these three points in the said frames. The relationship between the three-point and canonical parameterizations is captured in the following equations

$$\begin{bmatrix} x_0 & x_1 & x_2 \\ y_0 & y_1 & y_2 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_0 & u_1 & u_2 \\ v_0 & v_1 & v_2 \\ 1 & 1 & 1 \end{bmatrix} \quad (6)$$

²Forward mappings necessitate interpolation from known values at arbitrary positions to unknown values on a grid, whereas backward mappings involve interpolations from known values on the grid to unknown values off the grid. The latter is much easier to characterize and solve systematically, and the former offers no particular advantage.

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix} = \frac{1}{D} \begin{bmatrix} x_0 & x_1 & x_2 \\ y_0 & y_1 & y_2 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} v_1 - v_2 & v_2 - v_0 & v_0 - v_1 \\ u_2 - u_1 & u_0 - u_2 & u_1 - u_0 \\ u_1 v_2 - u_2 v_1 & u_2 v_0 - u_0 v_2 & u_2 v_1 - u_1 v_0 \end{bmatrix}^T \quad (7)$$

where

$$D = u_0(v_1 - v_2) - v_0(u_1 - u_2) + (u_1 v_2 - u_2 v_1)$$

In the same way it is possible to write the affine transform parameters in terms of the absolute location of the reference points in the input frame and the *displacement* of these points between the two frames. This is the formulation of choice in the present problem for the following reason.

Because of the nature of the interdependencies of the frames, we are seeking a continuous correspondence (motion) field. Each affine transform generates a continuous correspondence, but a single affine transformation is generally not appropriate for the interframe dependencies throughout a given frame. Therefore, it is advantageous to partition the frame to many regions and optimize a separate transform for each region (we use triangular regions). Given the characteristics of the signal, transforms will do well to preserve continuity across region boundaries. With the formulation in (5), however, it is difficult to enforce this continuity. In contrast, enforcing continuity is very natural when representing the transforms with a three-point formulation as in (6).

The continuity argument motivates the following implementation. The two frames involved in the estimation process (past frame $k - 1$ and present frame k) are partitioned into triangular patches. The luminance values of the pixels in each triangular patch in the present frame is estimated by the luminance of the corresponding position in the corresponding triangle in the past frame (Figure 1). Since the vertices are shared among neighboring triangles, this setup creates a continuous motion field.

Let \vec{d}_a , \vec{d}_b , \vec{d}_c represent the displacement vectors of the vertices of an arbitrary triangle and \vec{d}_r be the displacement vector of pixel r inside the triangle as shown in Figure 2. \vec{d}_r can be computed using linear interpolation (note that this is merely the same affine transform in disguise):

$$\vec{d}_r = (1 - p - q)\vec{d}_c + p\vec{d}_b + q\vec{d}_a \quad (8)$$

p and q are independent of the parameters of the individual affine maps, and depend only on the relative position of a pixel in the triangle. They can therefore be computed off-line and stored in a look-up table. This, however, is only possible if the triangular grid is regular.

Once the above mechanism is in place, the interframe estimation is reduced to finding optimal locations for the triangle vertices. The displacement of the vertices is computed in the coder through an iterative optimization strategy that at each step calculates the best local deformation for a given triangle. These displacement vectors are then passed to the decoder along with the estimation error (also known as the residue frame). This method can therefore

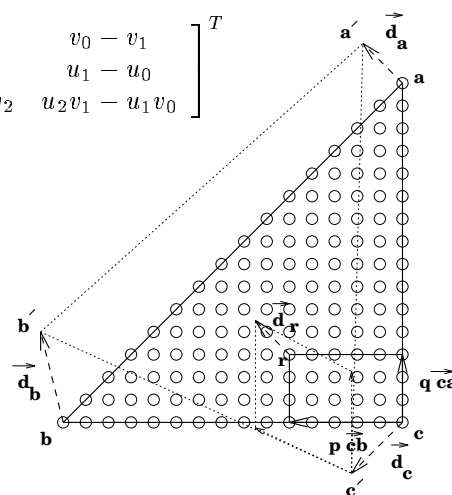


Fig. 2. Calculation of the displacement of a pixel in terms of that of the vertices

be considered in the class of forward motion compensated coders.

At each step of the optimization, all nodes in the triangular mesh in the past frame are kept stationary except one. The optimal position of that one node, according to a minimum mean absolute error criterion, is computed and the node updated. Then, the same procedure is repeated for the next node in the raster scan order. In order to avoid cross-overs and degenerate cases, the allowable new positions for a moving vertex are within the inverse map of a hexagon in the present frame, which is formed by the six triangles that share the given vertex (shown as gray areas in Figure 3). The change in the position of any vertex affects the error frame at most in an area that is bound by the hexagon mentioned above. Therefore, error calculations can be limited to this area.

The iterative procedure finds a local optimum, hence a good choice for the starting point can help reduce the eventual estimation error. We initialize the algorithm with a displacement that is obtained by a conventional block-based motion compensation method. 16×16 blocks are centered around each vertex and the optimal minimum-absolute-difference position for the block indicates the starting point for the vertex. The vertices at the boundary of the image are initialized differently: their starting position is taken to be the projection of the starting position of the nearest interior vertex.

This iterative raster scan optimization is repeated until a stopping criterion is achieved, or a preset maximum number of iterations is exceeded. After the optimization is complete, the displaced position of the vertices is losslessly coded with a Huffman code.

A. In-Loop Filter

The magnetic resonance images contain a fairly large amount of thermal noise, which is mainly caused by low signal levels. For practical and safety reasons, this noise cannot easily be reduced through cooling the reception coils (antennae). Also, there is little if any correlation between

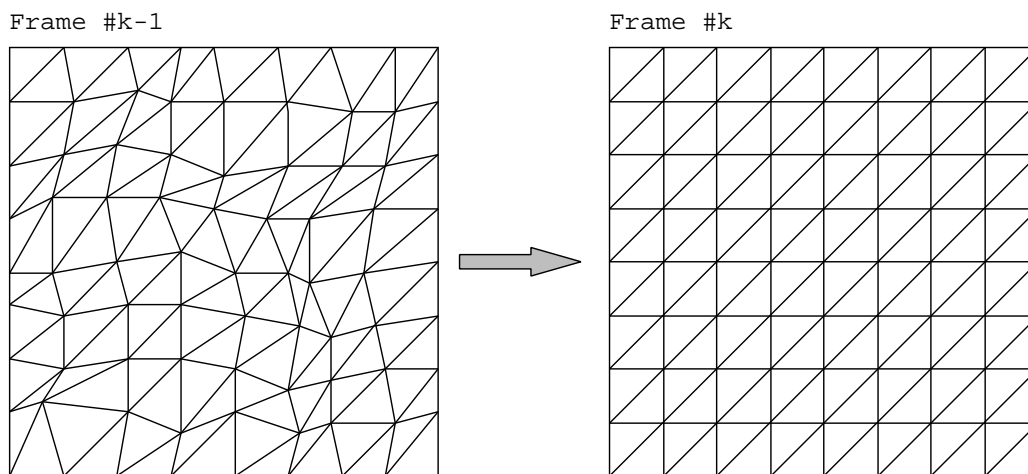


Fig. 1. Correspondence (motion) mapping through deformable triangles

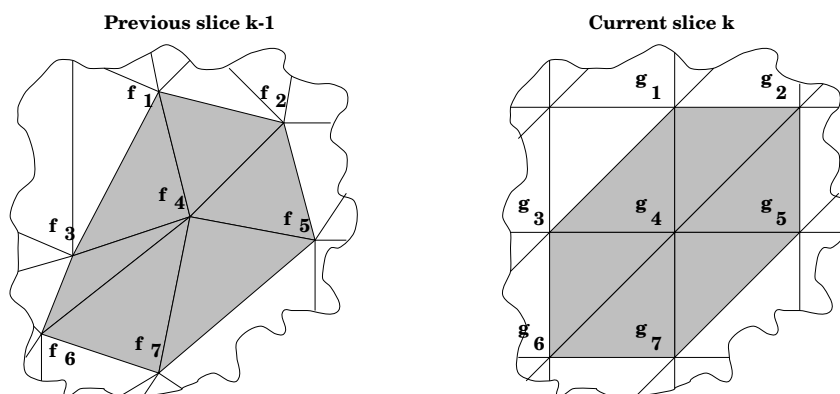


Fig. 3. Hexagonal area of allowable update locations for optimization of a node (past frame), and hexagonal area over which the error is accumulated (present frame)

this noise and the actual images, and between the noise in two consecutive frames. To illustrate the effects of this noise on an interframe estimator, consider the following simplified example: denote the underlying signals in the two frames – without the noise – as $f_{k-1}(x)$ and $f_k(x)$, and assume that the signal in frame k is perfectly estimated from frame $k-1$, through a space-varying motion field:

$$f_k(x) = f_{k-1}(x - v_k(x))$$

Now consider the observed signals $h_{k-1}(x) = f_{k-1}(x) + n_{k-1}(x)$ and $h_k(x) = f_k(x) + n_k(x)$, where $n_{k-1}(x)$ and $n_k(x)$ are σ^2 i.i.d. noise processes, with n_{k-1} and n_k uncorrelated with each other, and with f_{k-1} , f_k , and v_k . Given the motion field $v_k(x)$, the best available estimate of f_k from h_{k-1} would be :

$$\hat{f}_k(x) = h_{k-1}(x - v_k(x))$$

$$E \left[|f_k(x) - \hat{f}_k(x)|^2 \right] = E \left[|n_k(x) - n_{k-1}(x - v_k(x))|^2 \right] = 2\sigma^2 \quad (9)$$

hence the estimation has an error variance of $2\sigma^2$.

But this is not all: the motion field itself is not given *a-priori*, and has to be estimated from the intensity signals. In other words, the best realistic estimate of the new frame

is $\hat{f}_k(x) = h_{k-1}(x - \hat{v}_k(x))$. The noise interferes with the process of motion estimation also, and thus has another detrimental effect which is much more difficult to quantify due to the nonlinear relationship of motion and intensity fields.

A straight forward approach to this issue would be to pre-process the frames, possibly with a low-pass or stochastic filter. The drawback of this approach is that the signal and noise characteristics are not known. Attempting to extract noise characteristics from the noisy frames themselves is also not very promising, since it is unclear how much of the new content of a frame is useful information, and what portion of it is noise.

Therefore, one is bound to consider all the information of each frame as important and code it accordingly. However, while we do code the present frame in its entirety, it is perfectly acceptable to use – for the purposes of estimation – any function of the past frame. Thus, we propose to perform motion estimation between the filtered past frame and the (noisy) present frame. The advantage of this filtering is two-fold, corresponding to two detrimental effects of the noise: the motion estimates become more reliable, and the additive effect of noise, in the process of forming the residue frame, is reduced.

In our experiments we used a 5×5 median filter. Our motivation in using a median filter (vs. linear filtering) has been its better performance close to the boundaries, where it creates less blurring, as well as its computational complexity, which is much lower than a comparable linear filter.

The complete block diagram of the coder, with the filter in the estimation loop, is shown in Figure 4.

B. Computational Complexity

While detailed discussions of implementation issues are outside the scope of this paper, we briefly comment on the computational complexity of various segments of our inter-frame estimator, and offer options that allow tradeoffs of significant amounts of computation for a minimal reduction in rate-distortion performance.

The warping estimator can be computationally expensive if implemented directly. The sources of the computational expenditure are interpolations in motion and intensity domains, plus the iterative procedure of optimizing node motions. We consider each of these issues separately.

The iterative process of finding optimal node motions can be performed effectively through a process similar to iterated conditional modes of Besag [13], where the lattice of nodes is partitioned into sets such that the energy functions of the nodes in each set are independent. Optimization within each set and iterating between sets makes for very fast convergence. In our case, four sublattices are formed and energy functions typically reach equilibrium after two iterations. Note that such a structure is not strictly necessary, but does improve convergence speed.

Interpolations use a considerable portion of the computational budget in warping estimation. Recall that interpolation of motions is done with affine coefficients – denoted p and q – to compute motion parameters at individual pixels. Interpolation in the intensity domain is necessary because motion vectors resulting from affine (or bilinear) interpolation are not integer valued any more, and do not necessarily point to pixel locations.

While motion interpolation requires a significant number of multiplications, the multipliers are by fixed, pre-determined numbers p and q . Interpolation coefficients p and q can be kept in look-up registers, which reduces memory access cycles. Furthermore, they can be quantized to sums of powers of two, which allows multiplications to be performed by a small number of add and shift operations, largely reducing computation time.

Intensity domain interpolations are also required, because interpolated motions are not integer-valued and do not necessarily point to pixel locations. Based on known results from fractional-pixel motion estimators, motion resolutions beyond half-pixel contribute very little to coder performance. This observation suggests a favorable computation-performance tradeoff by limiting resolutions to half-pixel. One can interpolate the past frame to half pixel accuracy – which involves only three add and one shift operations per interpolation – and then motion vectors will be quantized to half-pixel resolution, so that they use

the half-pixel interpolated intensity field without any further computation. This will reduce the computations considerably, while having a minimal effect on performance. Moreover, half-pixel interpolation of the past frame can be performed in parallel to other tasks in a hardware implementation of the coder.

Finally, our choice of in-loop filter was partially motivated by computational considerations. Median filters require no multiplication, but only a small number of compare operations. The CPU time used for median filtering is negligible compared to the search and interpolation phases of the algorithm.

Altogether, our inter-frame motion estimator requires up to twice the CPU times spent for the standard block-matching algorithm. The overall picture, however, is somewhat different, since any inter-frame coder needs not only to find motions, but also to code the residues. In an overall comparison of CPU times (taking the residue coder into account), the new coder required between 30 to 60 percent more CPU time than a comparable block-matching coder.

III. CODING OF THE ERROR FRAME

We code the residue signal with a zero-tree/wavelet coder similar to [11], [14]. At the present, zero-tree coding is one of the most efficient techniques for coding individual frames. In addition to efficiency, the version of zero-tree coding used here has the progressive transmission property, which is useful in many medical applications. Moreover, it has the advantage over VQ techniques that it requires no training. Therefore, there is no question of over-training or whether a coder is geared towards a particular data set.

The heart of the residue coder is a wavelet transform and inverse transform pair. The wavelet transform is a unitary decomposition of the signal into its components at different local resolutions. Such a decomposition allows us to employ our knowledge of the frequency spectrum of medical images in the bit assignment problem. At the same time, because the transform retains information about the locality of energy, important features such as edges can be preserved at the same time as giving low frequency information a high priority.

The wavelet transform has a subband tree structure; a result of down-sampling at each subband stage. Each node at the top level of the tree is associated with three child nodes, and every other node (except leaves) is associated with four children (Figure 5).

The zero-tree wavelet coder utilizes not only the decorrelating properties of the transform (which allow us to use scalar quantizers efficiently), but also the dependence of the wavelet coefficients corresponding to the same spatial location but in different subbands. The main idea behind this method was generated by observing that in most natural as well as medical images, whose main features are associated with edges and textures, there are large areas in the higher frequency bands of the wavelet transform where coefficients are insignificant and hence can be coarsely quantized (mostly to zero). If there is a way to efficiently predict these patches of insignificant coefficients from other

same names. The dominant list includes the nodes that have just become significant, and the subordinate list the nodes that became significant in the previous iterations. The algorithm is as follows:

1. *Set initial threshold:* Set threshold T to the highest power of 2 less than the maximum absolute value of all wavelet coefficients c_n (n enumerates the coefficients along the scan path),

$$k = \lfloor \log_2(\max_n |c_n|) \rfloor$$

$$T = 2^k$$

2. *Initialize index lists:* All the nodes at the top level of the wavelet tree are put into the dominant list. The subordinate list is initialized to an empty list.
3. *Initialize states:* Set $S_{old} = IT$ for all nodes.
4. *Dominant pass:* For each entry n in the dominant list do:
 - (a) Save the old state S_{old} and find the new state S_{new} at the current threshold, using the definition of the states above.
 - (b) Transmit the code for state transition $S_{old} \rightarrow S_{new}$
 - (c) If no change in state ($S_{old} = S_{new}$) go to step 5.
 - (d) If $S_{old} \neq SR$ and $S_{new} \neq IR$ (the node has just become significant) then add index n to the subordinate list (for the next run) and output the sign of the node corresponding to n .
 - (e) If $S_{old} \neq IR$ and $S_{new} \neq SR$ (the children of the node have just become significant), add the indices corresponding to the immediate children of the node to the dominant list.
 - (f) If $S_{new} = ST$ then remove the index n from the dominant list.
5. *Subordinate pass:* For all entries in the subordinate list, transmit the next (i.e. $(k - 1)$ -th) most significant bit.
6. *Threshold update:* Set $T \leftarrow T/2$, decrement k and go to step 4.

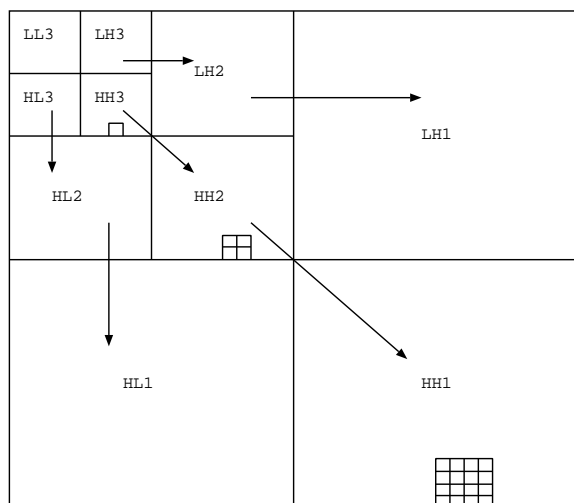


Fig. 5. Subband structure

This tree description has one more symbol than the one used by Shapiro (namely the IR symbol). Although simplicity of the tree description alphabet is crucial to the efficiency of the subsequent arithmetic coding operation, experiments indicate that in this case, the gain achieved by a better description of the tree overrides the loss of efficiency in the entropy coder.

Entropy coding is performed with an arithmetic coder [16], [17]. We used a conditional probability model for the coder. The probability of the tree symbols are conditioned on S_{old} , and the bit information has a separate model.

IV. EXPERIMENTAL RESULTS

We tested our coding scheme on two data sets. The first test sequence consists of 11 256×256 MR slices of a human heart, with a resolution of 12 bits. We used the bi-orthogonal wavelets of [18] for the zero-tree quantizer. Figure 10 compares coding efficiencies at different bit-rates between the inter-frame and intra-frame techniques. Note that in order to measure only the effect of inter-frame estimation, we have used the same zero-tree coder for the intra-frame results.

SNR values are computed as

$$SNR = 10 \log_{10} \frac{E(\mathbf{x}^2)}{\frac{1}{N} \sum_{i=1}^N (x_i - \hat{x}_i)^2} \quad (10)$$

where x_i and \hat{x}_i are original and reconstructed pixels and $N = 256 \times 256$ is the size of the image. $E(\mathbf{x}^2)$ is the expected value of random field \mathbf{x} representing the original image. One of the original slices and its coded counterpart are shown in Figures 8 and 9.

The second data set consists of 40 MR slices of a human brain. This data set is coded in a similar manner and

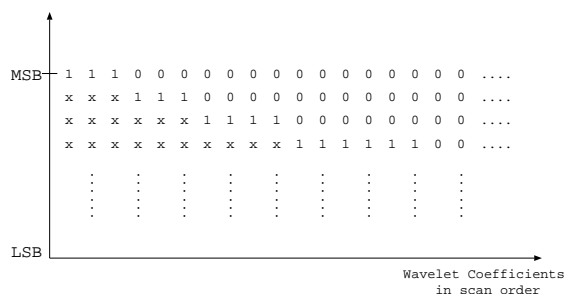


Fig. 6. Bit plane profile for raster scan ordered wavelet coefficients

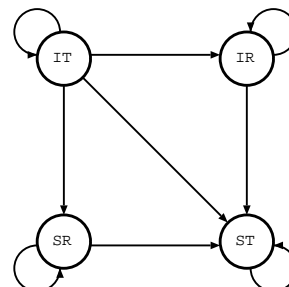


Fig. 7. State transition diagram for the zero tree coder

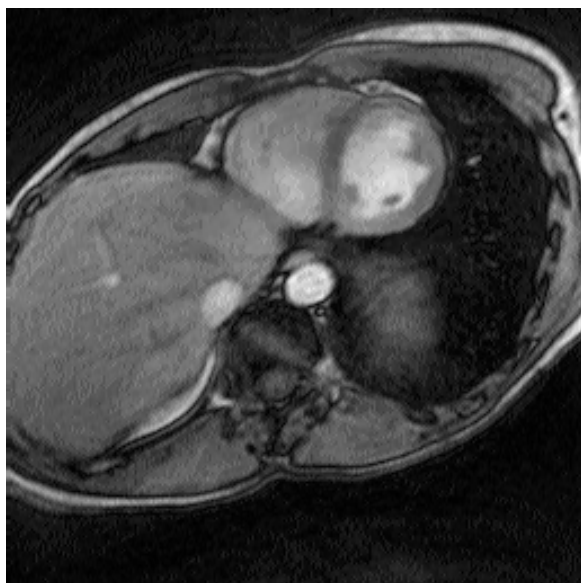


Fig. 8. Original heart slice



Fig. 9. Interframe encoded heart slice at 1.4 bpp

the corresponding rate-distortions are shown in Figure 11. Examples of the original and coded slices are given in Figures 12 and 13.

Overall, for a given SNR, bitrate savings of between 5 to 15 per cent have been achieved with the proposed interframe method, compared to intraframe coding. Compared to the block-matching algorithm, our method offers typical bitrate savings of about 30%. Subjective comparisons at three different bitrates on both image sequences also evaluated images coded with the affine coder higher than those coded either intra-frame or with block-matching.

V. DISCUSSION AND CONCLUSION

This paper presented a new coding algorithm for the compression of medical image sequences, in particular MRI. The results show a 5 to 15 per cent improvement

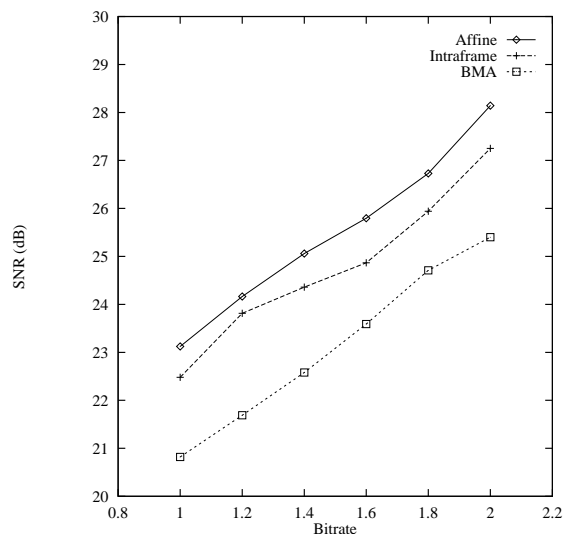


Fig. 10. Average SNR at different bitrates for the heart sequence. The three curves represent the affine coder, intra-frame coder, and block-matching coder.

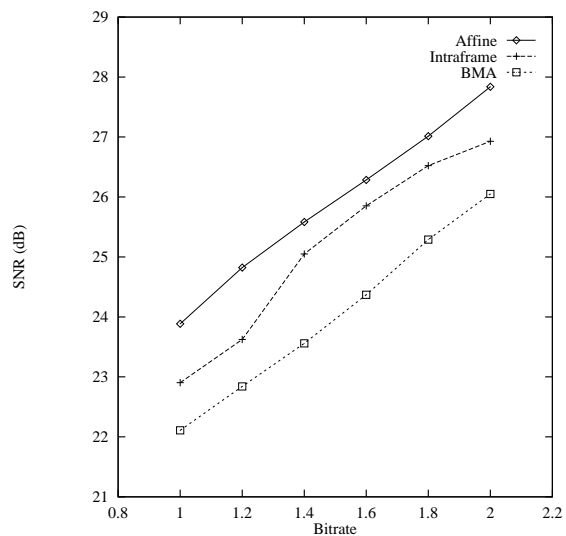


Fig. 11. Average SNR at different bitrates for the brain sequence. The three curves represent the affine coder, intra-frame coder, and block-matching coder.

in compression ratio compared to intra-frame coding, and more than 30% compared to standard block-matching. The proposed method is based on a piecewise affine, continuous one-to-one mapping between frames. Such a model for inter-frame dependencies is particularly attractive for coding medical image sequences that consist of slices of 3-D objects. The only time such a model is completely unable to make a good prediction is when the new slice goes through a protruding tissue for the first time, and the cross section does not have a correspondence with the past frame. But one can argue that in such a case, no interframe estimation technique can possibly work and the new information has to be coded.

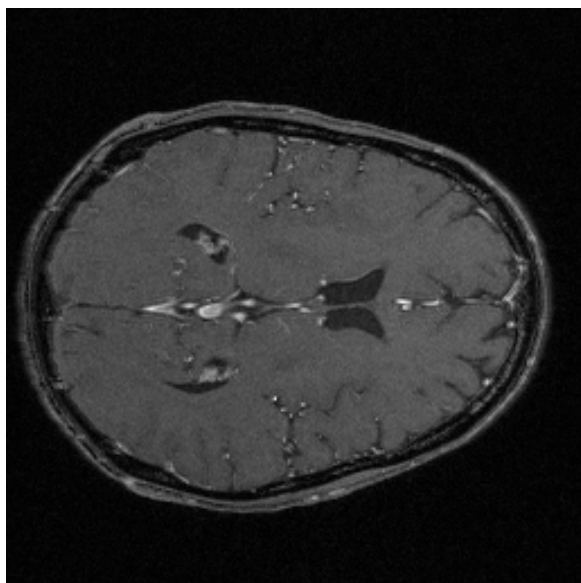


Fig. 12. Original brain slice

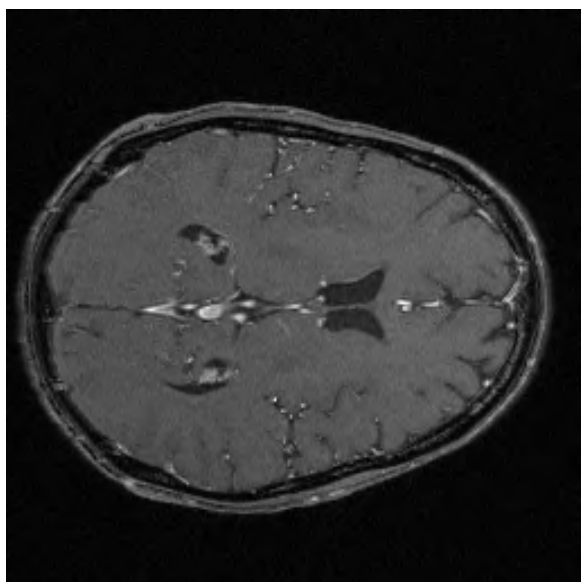


Fig. 13. Interframe encoded brain slice at 1 bpp

Another factor that contributes to the difficulty of the displacement estimation problem is the large amount of noise present in MRI. In our method, this noise was reduced through median filtering. Although a filtered image is much easier to code, the resulting quality is unacceptable. Therefore the filter is placed within the estimation loop, so that the filtered version is used for motion estimation, but the original version is quantized.

The work presented here has generated some new questions. The use of a median filter, for instance, was somewhat arbitrary. It would be interesting to know that, given a statistical model for the signal and noise, what is the optimal (linear?) filter to be placed in the estimation loop. Because of the highly non-linear nature of motion estimation, this is a difficult problem.

One possible way to improve the compression ratio is

to generalize the zero-tree quantization technique to include a rate-distortion optimization [19]. However, this removes the progressive transmission property and it is arguable whether the gain in compression is worth the inconvenience. Such a decision will ultimately depend on the details of the system requirements. Also, a wavelet-tree technique implicitly depends on a decaying spectrum, and a spectral analysis of the residue frames indicates that their energy content is more or less constant across the frequency bands. It seems reasonable then that a more general tree structure such as the wavelet packet tree [20] or time-frequency tree (a tree structure that allows splitting both in space and in frequency) may increase the coding efficiency even further. These questions will be the subject of future research.

ACKNOWLEDGEMENT

The authors would like to thank the Siemens Corporation for providing the MRI image sequences.

REFERENCES

- [1] P. Roos, M. A. Viergever, M. C. A. V. Dijke, and J. H. Peters, "Reversible intraframe compression of medical images," *IEEE Trans. Medical Imaging*, vol. MI-7, pp. 328-336, December 1988.
- [2] P. Roos and M. A. Viergever, "Reversible intraframe compression of medical images: A comparison of decorrelation methods," *IEEE Trans. Medical Imaging*, vol. MI-10, pp. 538-547, December 1991.
- [3] P. Roos and M. A. Viergever, "Reversible 3-d decorrelation of medical images," *IEEE Trans. Medical Imaging*, vol. MI-12, pp. 413-420, September 1993.
- [4] M. A. Viergever and P. Roos, "Hierarchical interpolation: an efficient method for reversible compression of images," *IEEE Engineering in Medicine and Biology*, vol. 12, pp. 48-55, March 1993.
- [5] K. K. Chan, C. C. Lau, K. S. Chuang, and C. A. Marioka, "Visualization and volumetric compression," in *Proc. SPIE Image Capture, Formatting and Display*, vol. 1444, pp. 250-255, 1991.
- [6] H. G. Mussman, P. Pirsch, and H. J. Grallert, "Advances in picture coding," *Proc. IEEE*, vol. 73, pp. 523-548, April 1985.
- [7] H. Lee, Y. Kim, A. H. Rowberg, and E. A. Riskin, "Statistical distributions of DCT coefficients and their application to an interframe compression algorithm for 3-d medical images," *IEEE Trans. Medical Imaging*, vol. MI-12, pp. 478-485, September 1993.
- [8] G. J. Sullivan and R. L. Baker, "Motion compensation for video compression using control grid interpolation," in *Proc. IEEE ICASSP*, vol. 4, pp. 2713-2716, May 1991.
- [9] Y. Nakaya and H. Harashima, "An iterative motion estimation method using triangular patches for motion compensation," in *Proc. SPIE Visual Comm. Image Processing*, vol. 1605, pp. 546-557, November 1991.
- [10] M. Vetterli and C. Herley, "Wavelets and filter banks: Theory and design," *IEEE Trans. Signal Process.*, vol. 40, pp. 2207-2232, September 1992.
- [11] J. Shapiro, "Embedded image coding using zero-trees of wavelet coefficients," *IEEE Trans. Signal Process.*, vol. 41, pp. 3445-3462, December 1993.
- [12] G. Wolberg, *Digital Image Warping*. Los Alamitos, CA: IEEE Computer Society Press, 1990.
- [13] J. Besag, "On the statistical analysis of dirty pictures," *Journal of the Royal Statistical Society, Part B*, vol. 48, no. 3, pp. 259-302, 1986.
- [14] A. Said and W. A. Pearlman, "Image compression using the spatial orientation tree," in *Proc. IEEE ISCAS*, vol. 5, pp. 279-282, 1993.
- [15] A. S. Lewis and G. Knowles, "Image compression using the 2-d wavelet transform," *IEEE Trans. Image Processing*, vol. 1, pp. 244-250, April 1992.
- [16] I. H. Witten, R. M. Niel, and J. G. Cleary, "Arithmetic coding

- for data compression," *Communications of the ACM*, vol. 30, pp. 520–540, June 1987.
- [17] T. C. Bell, J. G. Cleary, and I. H. Witten, *Text Compression*. Englewood Cliffs, New Jersey: Prentice Hall, 1990.
 - [18] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using wavelet transform," *IEEE Trans. Image Processing*, vol. 1, pp. 205–221, April 1992.
 - [19] Z. Xiong, N. P. Galatsanos, and M. T. Orchard, "Marginal analysis prioritization for image compression based on a hierarchical wavelet decomposition," in *Proc. IEEE ICASSP*, vol. 5, (Minneapolis, Minnesota), pp. 546–549, April 1993.
 - [20] Z. Xiong, K. Ramchandran, M. T. Orchard, and K. Asai, "Wavelet packets based image coding using joint space frequency optimization," in *Proc. IEEE ICIP*, vol. 3, (Austin, Texas), pp. 324–328, November 1994.