Video Compression for Very Low Bit-Rate Communications Using Fractal and Wavelet Techniques

Yarish Brijmohan and Stanley H. Mneney, Senior Member, SAIEE

Abstract—In this paper, we propose a very low bit-rate video compression algorithm that is based on fractal coding in the wavelet domain. Fractal compression techniques utilise local self-similarities present in images, to form images of properly transformed parts of itself. The recent discovery of the link between fractal compression and wavelets, have improved the performance of fractal coders. The self-similarity property of fractal techniques lends itself well to video coding since there are high spatial and temporal correlation between video frames.

In the proposed scheme, each frame is decomposed using the pyramidal multiresolution wavelet transform. Thereafter a motion detection operation is used in which the subtrees are partitioned into motion and non-motion subtrees. The non-motion subtrees are easily coded by a binary decision, whereas the moving ones are coded using the fractal variable tree size coding scheme and the wavelet SPIHT (Set Partitioning in Hierarchical Trees) Algorithm. All intra-frame compression is performed using the wavelet SPIHT coder.

Simulation results show that the proposed scheme has improved performance over the H.263 compression standard, hence making it suitable for low bit rate video.

I. INTRODUCTION

LOW bit-rate video compression is intended for video communication systems that operate on channels with low bandwidth. With the introduction of mobile communication devices, research in this field has been fast expanding. Uncompressed video requires large volumes of storage capacity, which far exceeds the available bandwidth. Table I illustrates the storage and time requirements to transmit a one minute uncompressed video sequence over a low bandwidth channel. It can clearly be seen that some mode of compression must be introduced to reduce the time required to transmit the video.

The main idea of video compression is to exploit redundancies that are present in the video. There are two compression standards present that have been developed for low bit-rate applications. These are the H.263 [10] and MPEG-4 [9] video compression standards. Both standards utilise wavelet transform based techniques. Wavelet transform techniques combine both transform and subband coding. The wavelet transform provides a multi-resolution and multi-frequency decomposition of images. Wavelet analysis provides images with certain properties that lend itself to image and video compression [1]. One of the most important properties is that the wavelet transform decorrelates mutually dependant parts of the image and performs an energy compaction of the samples representing the image.

The other compression technique that has the potential for high compression is the fractal coding technique. Fractal compression differs from the standard transform coder methods. They were created as a result of the study of iterated function systems (IFS). Fractal methods store images as contraction maps of which the image is the fixed point. The decoding procedure in which the image is recovered by iterating the maps to its fixed point is simple. However, the recovered image suffers from the tiling effect, especially at low bit-rates.

This paper describes a new algorithm that is used for compression of video sequences. This algorithm uses both wavelet and fractal coding schemes. A brief overview of wavelet and fractal techniques is presented in Section II. These techniques are then extended and combined to form the new video coder, presented in section III. Finally, in section IV, simulation results are provided using the codec on different test sequences.

II. WAVELET AND FRAC TAL COMPRESSION TECHNIQUES

A. Wavelet Transform Coders

Wavelet analysis is similar to subband analysis; therefore subband techniques will be used to compute the wavelet transform. For a full introduction to wavelet analysis, the reader is referred to [1].

The 2-D wavelet representation of an image creates a so called hierarchical tree structure. This tree is a set of wavelet coefficients corresponding to the same spatial location and orientation. A square block in the spatial (image) domain

<table>
<thead>
<tr>
<th>Media Type</th>
<th>Television Video</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size</td>
<td>352 x 288</td>
</tr>
<tr>
<td>Duration</td>
<td>1 min (25 frames/sec)</td>
</tr>
<tr>
<td>Bits per Pixel</td>
<td>24 bpp</td>
</tr>
<tr>
<td>Uncompressed Size</td>
<td>456 MB</td>
</tr>
<tr>
<td>Transmission Bandwidth</td>
<td>60.8 Mb/s</td>
</tr>
<tr>
<td>Transmission Time</td>
<td>21 hours 7 mins</td>
</tr>
<tr>
<td>(56K Modem)</td>
<td></td>
</tr>
</tbody>
</table>
produces an assembly of three trees and coefficients in the highest level approximation subband (see Fig. 1). Coefficients at the highest scale (lowest frequency) are normally called the coarse scale coefficients, while those at level 1, the finest scale coefficients.

The coefficients in the lower frequency subbands can be thought of as having four children (or descendants) in the next higher subband. Each of these children will in turn have four children in the next higher subband. The definition of a quad-tree is given as a tree of locations in a wavelet transform, with a root, its children and each of their children, and so on, all the way back to the first level of the wavelet transform.

1) SPIHT Algorithm

The SPIHT (Set Partitioning in Hierarchical Trees) compression algorithm, introduced by Said and Pearlman [2], makes effective use of the wavelet tree structure in compressing images. SPIHT is based on the ideas of the zerotree hypothesis which states that if a wavelet coefficient $c$ at a coarse scale is insignificant with respect to a given threshold $T$, i.e., $|c| < T$ then all wavelet coefficients of the same orientation at finer scales are also likely to be insignificant with respect to $T$.

The term hierarchical trees refer to the quadtrees while set partitioning refers to the way these quadtrees partition the wavelet transform values at a given threshold. The SPIHT algorithm uses a partitioning of the quadtree in a manner that tends to keep insignificant coefficients together in larger subsets. The partitioning decisions are binary decisions that are transmitted to the decoder. SPIHT is a fully embedded wavelet coding algorithm that progressively refines the most significant coefficients. Hence the coding can be stopped at any point to meet the required compression ratio.

SPIHT uses a set of rules that partition the quadtrees according to their significance. Thus, three lists are used to keep track of the order in which elements are tested for significance. These lists are:

- LSP – The list of significant pixels which contains the coordinates of coefficients that are found to be significant.
- LIP – The list of insignificant pixels which contains the coordinates of coefficients that are insignificant.
- LIS – The list of insignificant sets, which contain the coordinates of the roots of potential zerotrees.

The SPIHT algorithm is the most commonly used wavelet compression technique, due to its simplicity and efficiency. It has been shown that this method can compress images in real time [3].

B. Fractal Block coders

Fractal compression techniques are based on IFS, in which the image is described by sets of equations that provide contractive mappings. The first effective fractal coder was introduced by Jacquin [4], who noticed that a part of an image is similar to another part of the image. This coder makes use of this local self similarity for compression, by forming images from properly transformed copies of parts of itself. This system requires the image to be divided into smaller, non-overlapping blocks called range blocks. Thereafter, larger domain blocks are constructed from the same image. The idea of this coder is to find an affine transformation of a domain block that closely matches each range block.

Two issues need to be considered in the design of fractal based compression systems. The first is the size and shape of range and domain blocks. Since range blocks are the attractors, the types of affine transformations become limited, since it depends on the size and shape of the range block. The simplest partitioning scheme is to divide the image into non-overlapping square range blocks of a fixed size. The domain blocks are normally twice the size of the range blocks and can overlap. Results reveal that small block sizes yields small compression ratios and a large block size, a large compression ratio. However, the clarity of the recovered image suffers with large block sizes. The downfall of the fixed size partition is that the image is partitioned without considering the contents of the image. There are regions of the image that will be covered well using small range block sizes and similarly, there are regions that could be covered well with larger range blocks. This will increase the compression ratio and maintain the clarity. This observation leads to the use of the variable block size partitioning technique.

The second matter to consider is the type of affine transformations to perform on domain blocks. It is required that these transforms be contractive in order for a fixed point to be reached in the decoding stage. The transforms are normally flip operations such as: horizontal, vertical, and diagonal flip; or rotation operations such as: 90°, 180°, and 270° rotation. Also the identity block forms one of the transformations.

Decompression is simpler and much faster than the coding process. Using the stored transform data, an iterative process commences, until the final image is reached.

C. Combination Fractal and Wavelet coders

Since self-similarities do exist in the multi-resolution wavelet representation, fractal compression can be performed on wavelet coefficients. Fractal block coding was extended to the wavelet domain by the works of Davis [5] and Krupnik [6] independently. Davis introduced a new term, the wavelet subtree, which consists of wavelet coefficients from the same spatial location but with different resolution and orientation.
He then demonstrated that using haar wavelets for the wavelet transform is equivalent to the original fractal block coder.

Consider a 4 level DWT of an image (Fig. 2). The term subtree refers to the horizontal, vertical and diagonal quadtree structures at the highest level, together with its node at the coarsest scale. A domain block of size 16x16 forms a wavelet tree structure with its root nodes at the 4th level, whereas a range block of size 8x8 has its nodes at the 3rd level. Using the spatial domain analogy, here, a subtree at level 4 must be used to predict the subtree at level 3. Thus the main idea of wavelet-based fractal coding is to approximate a range subtree by a domain subtree under the proper affine transform. The conversion of commonly used affine transformations to the wavelet domain is provided in [11].

1) Variable Tree Size Fractal coding

As in fractal block coding, to achieve a higher compression, variable block sizes must be used. The extension of this to the wavelet domain is to use subtrees of different sizes. This method is presented by Zhang Y [10] and is summarized as follows: For a given range subtree, a best matched transformed domain subtree must be sought. If the distortion is less than a predefined threshold, then the range subtree is fractal quantized. On the other hand, if the distortion is greater than the threshold, the range subtree is split into four children subtrees. The three root nodes are then removed from the tree and scalar quantized. The length of these subtrees have changed, therefore we need to construct a new domain search pool consisting of nodes at the next finest level. This process is shown in Fig. 2, where the shaded regions represent the pruned range and domain subtrees. The distortion check is performed again. This adaptive partitioning scheme continues until all the children range subtrees are coded or a minimum dimension range subtree is reached, in which the range subtree is coded with the least distortion.

Experimental results have illustrated that wavelet based compression methods on images outperform conventional fractal techniques in terms of compression ratio and recovered image quality. Also, wavelet-based fractal coders have improved performance on the spatial domain fractal coders, but they still do not compare with the wavelet ones. However, coders that combine both fractal and wavelet techniques in the wavelet domain have a slight improved performance on the SPIHT method, with a significant complexity increase. It has also been noted that the use of bi-orthogonal filters in the wavelet decomposition improves the image quality from the haar transformed coder, since it removes the tiling effect.

III. FRACTAL AND WAVELET CODING ON WAVELET SUBTREES FOR VIDEO COMPRESSION

A. System Overview

The proposed system is based on the Fractal Compression of Wavelet Subtrees (FCWS) video compression scheme developed by Brijmohan et al. [11]. The block diagram of this video system is shown in Fig. 3. All video frames are captured from a video source and transferred to the system in a raw formatted matrix.

Each captured frame is transformed using the 4 level, multi-resolution pyramidal wavelet decomposition. Thereafter, the wavelet transform is split into subtrees. Each subtree then represents a motion structure that is highly correlated at different layers of the transform.

This scheme exploits the correlation and redundancy between consecutive frames in its compression algorithm. Hence both intra-frame and inter-frame coding need to be considered. Intra-frame coding involves the coding of the frame by itself, using image compression techniques such as the SPIHT algorithm. Inter-frame coding, on the other hand, is the coding of a frame using information from previous frames. Since a large portion of the previous frame will be repeated in the current frame, it is not required to code the entire frame. If we can separate the motion and non motion portions, a large reduction in the bit-rate will occur. Thus it is a requirement that a motion detection (MD) criterion to be used.

Initially, a decision needs to be taken of which frame coding method to use (Fig. 3). The first frame is always intra-frame coded. Thereafter, inter-frame coding is executed for a known number of frames.

B. Motion Detection

As in the FCWS algorithm, the motion detection criterion is based on the mean square error (MSE). A subtree in the current frame is compared to the same subtree in the previous frame. If the MSE between them is less than a predefined threshold, then the subtree is a non-motion subtree and can be coded with a binary decision. On the contrary, if the MSE is greater than the threshold, then this subtree needs to be coded by the proposed technique (Section C).

C. Coding of motion Subtrees

The coding of the motion subtrees is divided into two parts. The first part involves fractal coding of the motion subtree. In particular, the variable tree size algorithm is used ([8] and
The domain search pool is formed from the previous frame. The motion coder must find the best match transformed domain subtree to match the range subtree. The distortion is then compared to a threshold. If the distortion is less than the threshold, then the subtree is fractal quantised. If the distortion is greater than the threshold, the three nodes of the subtree are removed and scalar quantized. The subtree is split into four children subtrees with nodes now at level 3. The new domain subtrees are constructed from the previous frame, with nodes also at level 3. For each of the children subtree, a best match transformed domain subtree is sought. If the distortion is less than a predefined threshold, that child subtree is fractal quantized and encoded as its parent. However, if the distortion is greater than the threshold, the co-ordinates of the three nodes of the subtree are added to the LIP and LIS for SPIHT encoding. However, if three or more of the children subtrees do not find a matching domain subtree within a given threshold, the entire subtree from level 4 is SPIHT encoded, with the root node co-ordinate being added to the LIP and LIS.

The second portion of this scheme involves the implementation of the SPIHT algorithm. The standard SPIHT algorithm is performed on the LIP and LIS, after all the subtrees have been fractal coded. First the difference of the current frame and the previous frame is taken to reduce the magnitudes of the wavelet coefficients. Thereafter the SPIHT algorithm is executed.

After the frame coding is completed, the frame is then decoded and stored in memory as a reference to code the next frame. The frame memory refreshes periodically with the newly decoded frame. Since the domain subtrees of the fractal coded part is formed from the previous frame, only one step fractal decoding is enough to obtain the subtree of the original frame.

The efficiency of any coding method can be improved by entropy coding its output. Entropy coders remove the redundancy in the encoding of data. Here, the output of each frame is entropy coded using the adaptive arithmetic coding algorithm (described in [12]) before it can be transmitted.

IV. EXPERIMENTAL RESULTS

Computer simulations using the proposed algorithm were carried out on standard QCIF video sequences of 150 frames, with only the luminance part considered for the implementation. Two test sequences that were used are the “Akiyo” and “Salesman”. A four level, pyramidal wavelet decomposition was performed using the bi-orthogonal 9-7 filters. To reduce the bit-rate, a nearest neighbour search was used for the fractal coding part, with the search region restricted to -4 to + 4.

To assess the performance of the coder, the PSNR measurement is used which compares the reconstructed frame to the original frame. This codec obtains average bit-rates of 0.059bpp and 0.064bpp with the average PSNR as 32.41dB and 29.61dB for the test sequences “Akiyo” and “Salesman” respectively. The PSNR of the “Salesman” sequence is less since it has more motion structures. Illustrated in Fig. 4 and Fig. 5 are some recovered frames from both the sequences.

Table II details the time requirements to transmit the “Akiyo” video sequence over a low bandwidth channel. The bandwidth required to transmit this sequence is about a third of the available bandwidth, thus making the algorithm suitable for real time video steaming.

To assess the performance of the proposed coder against the H.263+ and FCWS algorithms, a comparison table is provided (Table III) on the “Akiyo” sequence. The results reveal that the visual quality of the proposed scheme is superior to that of both the H.263+ and the FCWS algorithms, at high and low bit-rates. Furthermore, the MPEG-4 standard obtains a recovered image quality of 34.43 dB at 40kbps, which is still less than the proposed video coder.

V. CONCLUSION

This paper proposes an extension to the very low bit-rate FCWS video compression algorithm. This algorithm performs both fractal and wavelet compression in the wavelet domain. The results obtained by the proposed coder illustrate the effectiveness of the algorithm, in compressing video sequences for low bit-rate communication systems. Also, this system has a higher performance than both the H.263+ and MPEG-4 video compression standards.
REFERENCES

Yarish Brijmohan received the B.Sc. degree (summa cum laude) in Electronic Engineering from the University of Natal, Durban, South Africa in 2002. He is currently pursuing a M.Sc. degree in Electronic Engineering at the School of Electrical, Electronic and Computer Engineering at University of KwaZulu-Natal. His research interests image and video coding using wavelet transforms and fractals. (e-mail: brijmohan1@nu.ac.za).

Stanley H. Mneney obtained his B.Sc. (Hons) Eng. degree from the University of Science and Technology, Kumasi, Ghana in 1976. In 1979 he completed his M.ASc. from the University of Toronto in Canada. In a Nuffic funded project by the Netherlands government he embarked on a sandwich Ph.D programme between the Eindhoven University of Technology and the University of Dar es Salaam, the latter awarding the degree in 1988. He currently holds the position of Associate Professor at the University of KwaZulu-Natal. His research interests include theory and performance of telecommunication systems, low cost rural telecommunications services and networks, digital signal processing applications, EMC, and RF design.

TABLE II
TRANSMISSION CAPABILITIES OF THE COMPRESSED AKIYO SEQUENCE OVER A LOW BANDWIDTH CHANNEL

<table>
<thead>
<tr>
<th>Media Type</th>
<th>QCIF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size</td>
<td>176 x 144</td>
</tr>
<tr>
<td>Duration</td>
<td>15 sec (10 frames/sec)</td>
</tr>
<tr>
<td>Bits per Pixel</td>
<td>8 bpp</td>
</tr>
<tr>
<td>Uncompressed Size</td>
<td>3.8 MB</td>
</tr>
<tr>
<td>Compressed Size</td>
<td>28 kB</td>
</tr>
<tr>
<td>Transmission Bandwidth</td>
<td>14.95 kb/s</td>
</tr>
<tr>
<td>Transmission Time</td>
<td>4.66 s</td>
</tr>
</tbody>
</table>

TABLE III
PERFORMANCE OF THE PROPOSED ALGORITHM COMPARED WITH THE H.263 STANDARD AND FCWS VIDEO COMPRESSION ALGORITHM ON THE “AKIYO” TEST SEQUENCE

<table>
<thead>
<tr>
<th>Rate (kbps)</th>
<th>H.263</th>
<th>FCWS</th>
<th>Proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PSNR (dB)</td>
<td>PSNR (dB)</td>
<td>PSNR (dB)</td>
</tr>
<tr>
<td>20</td>
<td>31.38</td>
<td>32.05</td>
<td>32.87</td>
</tr>
<tr>
<td>25</td>
<td>32.16</td>
<td>32.39</td>
<td>34.06</td>
</tr>
<tr>
<td>40</td>
<td>34.41</td>
<td>33.04</td>
<td>35.01</td>
</tr>
<tr>
<td>50</td>
<td>35.18</td>
<td>33.32</td>
<td>36.09</td>
</tr>
</tbody>
</table>

Fig. 3. Block diagram of the proposed scheme