Presentation-Oriented Key-Frames Coding Based on Fractals

Luigi Atzori, Daniele D. Giusto, and Maurizio Murroni

This paper focuses on the problem of key-frames coding and proposes a new promising approach based on the use of fractals. The summary, made of a set of key-frames selected from a full-length video sequence, is coded by using a 3D fractal scheme. This allows the video presentation tool to expand the video sequence in a "natural" way by using the property of the fractals to reproduce the signal at several resolutions. This feature represents an important novelty of this work with respect to the alternative approaches, which mainly focus on the compression ratio without taking into account the presentation aspect of the video summary. In devising the coding scheme, we have taken care of the computational complexity inherent in fractal coding. Accordingly, the key-frames are first wavelet transformed, and the fractal coding is then applied to each subband to reduce the search range. Experimental results show the effectiveness of the proposed approach.

Keywords: Fractals, wavelet, video processing, multimedia.

I. Introduction

The surprising diffusion of multimedia applications (from scientific to commercial, and from informative to recreational) over heterogeneous networks has caused a great deal of interest in the scientific community toward signal processing and data transmission fields. In most of these applications, digital video archives are browsed on distributed networks that are subject to buffer congestions and bandwidth constraints. To enable these services, it is important to develop tools to analyze and describe the video content, handle queries from the end-users, and provide results. These operations require the extraction of the essence of the visual content in a compact form so as to permit a fast browsing of huge multimedia archives. Accordingly, a procedure for automatic video data analysis and indexing has become a requirement for efficient database content searching and management. It is mainly made up of the following tasks [1]: feature extraction, structure analysis, abstraction, and indexing. The first task is aimed at providing the major characteristics of the video (such as color, texture, shape, structure, layout, and motion) that can be converted into semantic concepts. Video structure parsing is the next step in overall video-content analysis and is the process of extracting temporal structural information of video sequences or programs. Video abstraction is the process of creating a presentation of visual information about a landscape or the structure of video, which should be much shorter than the original video. Based on the output of the previous tasks, video indices are built so as to enable a fast browsing of the visual content.

Researchers have extensively investigated this topic in the recent past. One commonly adopted approach is a *storyboard* presentation in which thumbnail images are tiled together, usually arranged in a time order to give an overview of the

Manuscript received Sept. 19, 2003; revised Oct. 18, 2005.

Luigi Atzori (phone: + 39 070 675 5902, email: l.atzori@diee.unica.it), Daniele D. Giusto (email: ddgiusto@unica.it), and Maurizio Murroni (email: murroni@diee.unica.it) are with the Department of Electrical and Electronic Engineering, University of Cagliari, Italy.

visual content in the video sequences. In [2], the authors propose a novel approach for video summarization based on graph optimization. Their approach emphasizes both a comprehensive visual-temporal content coverage and visual coherence of the video summary. In [3], a novel pictorial video summary, called a *video snapshot*, which is a bird's eye view of video enabling viewers to grasp the main contents of video at a glance, is presented. Moreover, a comprehensive scoring scheme for content filtering called PRID (pleasurable, representative, informative, and distinctive), and an optimized video visualization algorithm are also proposed. The authors in [4] present a two-stage framework to generate MPEG-7-compliant hierarchical key-frame summaries of video sequences. At the first stage, which is carried out off-line at the time of content production, fuzzy clustering and data pruning methods are applied to given video segments to obtain a non-redundant set of key frames that comprise the finest level of the hierarchical summary. The number of key-frames allocated to each shot or segment is determined dynamically and without user supervision through the use of cluster validation techniques. A coarser summary is generated on-demand in the second stage by reducing the number of key-frames to match the low-level browsing preferences of a user. It is worth noting that the state-of-the-art works about video summarization mainly focus on the extraction of the key-frames.

This paper focuses on the coding of frames generated during the video abstraction task for fast data browsing. We propose a new promising approach that is based on the use of fractals. The summary, made of a set of key-frames, is represented using 3D fractal coding so as to allow the video presentation tool to expand the video sequence in a "natural" way by using the property of the fractals to reproduce the signal at several resolutions. While the primary issue addressed by the proposed technique is data coding in video abstraction, we also take heed of the video presentation issue.

Indeed, in literature several algorithms have been developed for image and video coding and have been adopted by some standardization committees, such as JPEG and MPEG, to give birth to image and video standards (JPEG 2000, MPEG-x). Such standard codecs mainly focus on compression aspects of the dataset. With respect to these, fractal coding of an image and video adds the feature of expansion of the dataset during decoding; this characteristic is not usually provided in standard codecs and has to be considered as an external separate issue if standard codecs are used for key-frames coding. In the past, fractals have been proposed for image data compression by exploiting pseudo-self-similarity inside natural images [5],[6]. The resulting algorithms have the advantage of allowing the expansion of the signal along its dimensions during decoding. However, the fractal representation of a signal has a major weakness, which is the high computational complexity of the encoding process. The computational load, and thus the processing time, increases as the signal dimension increases. This is due to the fact that there are more data to be processed at higher signal dimensions. We address this problem by using a wavelet subband coding scheme [7]. In particular, we perform the fractal coding of each wavelet subband in isolation so as to reduce the search range and the related processing time. To further reduce the high computational load, an active scene detection is used to perform three-dimensional fractal coding only in high-information areas (moving areas), whereas static zones are coded using a two-dimensional coder.

We have also taken care of the coding efficiency by using an adaptive wavelet coefficient quantization procedure. It is based on a histogram analysis of the wavelet coefficients distribution. At the receiving end, the fractal code can be decoded at the desired resolution in the time and spatial dimensions. It is worth noting that, if the computation overhead is an issue during the generation of the code, it is not a matter during the decoding and can be performed in almost real-time.

The paper is organized as follows. In section II, a presentation of fractal coding is given. Section III describes the proposed method in detail. In section IV, we discuss the results relevant to the conducted experiments. Conclusions are drawn in the last section.

II. Fractal Coding

A fractals theory applied to the image processing field is based on the iterated function system (IFS) and has been used mainly for data compression. The basic idea of the fractal coding based on IFS is to exploit the redundancy given by the self-similarities always contained in natural images. The fractal image can be seen as a collage composed by copies of parts of an original image that have been transformed through opportune geometric and massive transformations (that is, luminance or contrast shift). The mathematical foundation of this technique is the general theory of contractive iterated transformations, based on the works of Barnsley [5] and Jaquin [6]. Basically, fractal coding of an image consists in building a code τ (a particular transformation) such that, if μ_{orig} is the original image, then $\mu_{orig} \approx \tau(\mu_{orig})$. This means that μ_{orig} is approximately self-transforming under τ. If τ is a contractive transformation, μ_{orig} is approximately the attractor of τ, that is $\mu_{orig} \approx \lim_{k \to \infty} \tau^k(\mu_0)$ for any initial image μ_0. The code τ is built on a partition of the original image. Each block R_i of this partition is called a *range block* and is coded independently from the others by a matching (local code τ_i) with another block D_i in the image, called a

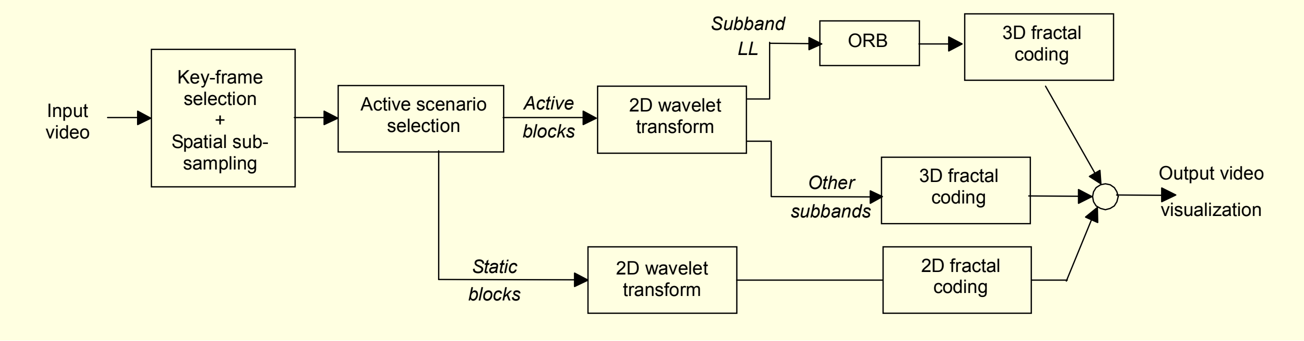

Fig. 1. Scheme of the proposed technique for video summarization.

domain block. If R and D are the range and domain block sizes (in the case of square blocks), respectively, then $D = pR$, where p is the scaling factor used for the local self-similarity search (p>1).

Classical τ_i transforms are both isometries, such as rotations and flips, and massive transforms, such as contrast scaling and grey shifting. If L is the number of range blocks, the fractal code of the initial image is then $\tau(\mu_{orig}) = \bigcup_{i=1}^{L} \tau_i$, where $\tau_i : D_i \to R_i$ and $\tau_i = M_i \circ I_i \circ r_{i,p}$. Equation $M_i(x) = a_i \cdot x_i + b_i$ is an affine operator with scale a_i and shift b_i on the luminance pixel; I_i is a transformation selected from eight discrete isometries; and r_{ip} is a p-factor reduction operator based on averaging. In other words, the task of the fractal encoder is to find for each range block a larger domain block such that, after an opportune transformation, it constitutes a good approximation of the present range block. The fractal code for the original image is a collection of such extracted local codes. This approach proposed by Jacquin in [6], gives a representation of the image as composed by copies of parts of the image itself.

The classical fractal decoding stage consists in an iterated process starting from an arbitrary initial image, μ_0. In fact, if τ is a contractive transformation, the τ's attractor $\tau^{\infty}(\mu_0)$ gives an approximation of the original image μ_{orig} independently from the initial image.

Indeed, the fractal code τ is a collection of linear affine transforms, τ_i, and has no intrinsic size. Hence, we can assume that self-similarities, represented by hatchings between different areas in the original image, are scale independent. Accordingly, the decoding process is resolution independent, that is, at the decoding stage the fractal code enables expansion (zooming in). Practically, this operation consists in increasing, during the decoding stage, the range block's size R, and therefore the domain block's size D (being $D = pR$). For a zoom-in factor z, the new range and domain block sizes are $R'=zR$ and $D'=zD$, but all the local codes τ_i, and consequently the fractal code τ, remain unchanged.

In fractal video coding [8], range and domain blocks become cubes; as a consequence, the number of isometries and massive transforms to be analyzed when generating the fractal code is greater with respect to the image coding framework. This fact raises dramatically the computational load of the matching algorithm. Moreover, at the decoding stage, the entire sequence should be decoded at once, and even the decoding procedure, which is usually the fast part of a fractal process, becomes extremely slow. Therefore, applying fractal coding to video turns out to be possible only when appropriate procedures for data reduction and problem simplification are used. Applying the fractal decoding to a blank sequence with the desired zoom-in factor leads to an expanded version of the sequence in both time and spatial dimensions.

III. Proposed Method

The proposed method relies on the joint use of fractal coding and wavelet subband analysis for video processing. The basic idea is to take advantage of the fractal coding feature to pleasantly reproduce missing motion information from a set of key-frames. It is worth noting that such a feature introduces pleasant results as far as a certain degree of correlation exists between consecutive key-frames. A subband wavelet transform is used to greatly reduce fractal coding time by processing each subband separately. In fact, this allows restricting the fractal-searching domain with respect to fractal spatial coding.

Figure 1 draws the functional blocks of the proposed methodology. The first step consists in the selection of the key-frames from the original video sequence. Several methods have been tested to this purpose, analyzing both efficiency and computational complexity. The obtained key-frames are then spatially sub-sampled and constitute the input to the active scenario selection phase. The aim of identifying the active scenario is to increase the compression ratio by performing a differentiate coding. In particular, the background (static scenario) is coded only once in a group of key-frames (GOK), while for the active scenario all the key-frames are taken into account. In the following sections, these operations are described in detail.

1. Key-Frame Extraction

Temporal video segmentation is the first step towards automatic annotation of digital video for browsing and retrieval. Its goal is to divide the video stream into a set of meaningful and manageable segments that are used as basic elements for indexing. Selected key-frames then represent each shot. There are different techniques to this aim [9], and the majority of proposed algorithms process uncompressed video, as we do. As to the selection of the appropriate algorithm to be used within our framework, we had to consider both speed and efficiency. In particular, we tested three methods at increasing efficiency and decreasing speed, that is, using a *fixed grid*, *pixel comparison*, and *histogram comparison*.

Fixed grid: The key-frames are chosen according to a fixed grid. Let N and M represent the number of frames in the original sequence and the step of the fixed grid, respectively. The number of the selected frames is $R = \left\lceil \frac{N}{M} \right\rceil$. Such an approach is very fast at the expense of an important drawback: frames with crucial temporal semantic content, such as initial frames in a new scene, could be discarded, and frames without semantic content information, such as frames in a static scene, can be selected as a key-frame without any coding benefit.

Pixel comparison: To overcome the drawback arising from the use of a fixed grid, the frames are chosen according to their difference from the previously selected frames [10]. Let $P_k(x, y)$ be the luminance at pixel x-y (x=1,…,X, y=1,…,Y) and frame k (k =1,…). For each couple of consecutive frames k and k+1, the algorithm computes the number $D(k, k+1)$ of pixels that change in value more than threshold T_{diff}:

$$D(k,k+1) = \frac{\sum_{x=1}^{X}\sum_{y=1}^{Y} Diff(k,k+1,x,y)}{X \cdot Y},$$

where

$$Diff(k,\ k+1,\ x,\ y) = \begin{cases} 1 & \text{if} \left|P_k(x,y) - P_{k+1}(x,y)\right| > T_{diff}, \\ 0 & \text{otherwise}. \end{cases}$$

Every frame k with $D(k, k+1)$ greater than threshold T_{PC} is selected as the key-frame.

The main weakness of this method is that it is not able to differentiate cases of significant changes in small areas from cases of non-significant changes in large areas. It follows that scene changes are not detected when a small part of the frame undergoes a major, rapid change. Therefore, it is sensitive to object and camera movements.

Histogram comparison: Comparing the histograms of adjacent frames instead of the grey level can be more robust against camera and object movements [10]. A key-frame is selected when the sum of histogram differences between two successive frames is greater than a given threshold T_{hist}:

$$\text{If } D(k,k+1) = \frac{1}{Z}\sum_{i=1}^{Z}\left|H_k(i) - H_{k+1}(i)\right| > T_{hist}$$

then frame k is a key-frame,

where $H_k(i)$ is the histogram value for grey level i in frame k, and Z is the number of grey levels.

If a key-frame selection procedure is required to be fast, the fixed grid solution is usually used at the expenses of a coding efficiency reduction. The other two solutions provide better results and are used in case of no time constraints during the coding stage. The histogram comparison method is usually selected in this case.

2. Active Scenario Selection

To increase the compression ratio, background in the video sequence key-frames is extracted and encoded once in a group of key-frames. Then, the active part of the sequence can be coded with more precision. To correctly perform the subsequent fractal coding, the remaining active scenario needs to have a shape such that an integer number of cube blocks can be contained. To this purpose, each key-frame k is divided into blocks $B_{i,j}^{k}$ (i=1,…, X/R;j=1,…,Y/R) with a size equal to the starting range block size R. By grouping the sequence of blocks $B_{i,j}^{k}$ in more key frames at the same spatial position (that is, i and j are fixed), we construct a set of parallelepipeds $P_{i,j} = \bigcup B_{i,j}^{k}$. Figure 2 shows an example of blocks $B_{i,j}^{k}$ and relevant parallelepipeds $P_{i,j}$ for a GOK of three key-frames.

A binary mask identifying the background is then built

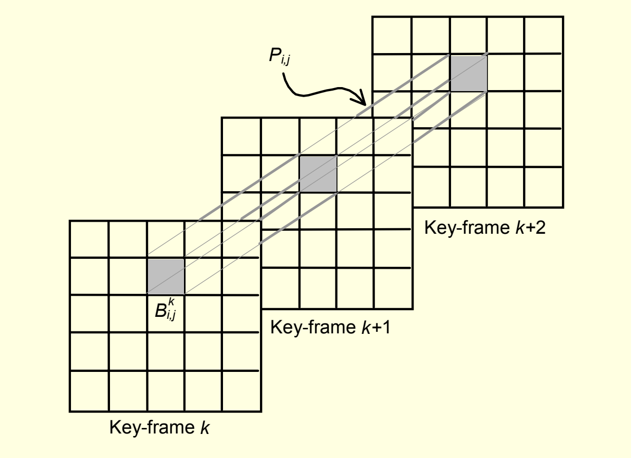

Fig. 2. Blocks $B_{i,j}$ in a sequence of key-frames and relevant parallelepiped $P_{i,j}$ used for the identification of the background scenario.

computing the distance measure between consecutive blocks. The following classification rule is applied: parallelepiped $P_{i,j}$ is classified as belonging to the active scenario if the distance measure for at least two consecutive blocks exceeds a given threshold T_{AB}; the other $P_{i,j}$ belong to the background area.

The separate coding of the background may introduce an annoying artifact: In a long sequence the background region luminance can significantly differ from that in the remaining part of the frame, that is, the active scenario. In order to restrict the artifact visibility, the background is refreshed for every GOK that is usually taken lower than 10 key-frames.

Since the encoding of a background with a small area would not increase the compression ratio, this process is done only if the area of the mask extracted exceeds a given percentage T_{back} of the whole area. Otherwise, the mask is not used and the frame is entirely encoded. Note that the extension of the background area depends on which procedure is used to select the frames to be coded. In the case of the fixed grid algorithm, consecutive key-frames are often characterized by a quite similar background. In case the other two algorithms are used, the presence of a common background area depends on the type of scene changes encountered in the video sequence. Some scene changes are due to the disappearance of a big foreground object; in that case, a similar background may still be found in different key-frames. Most of the other scene changes cause the background to be dissimilar in a GOK. Accordingly, the separation of the active scenario from the background in the proposed algorithm is sometimes inapplicable.

3. Application of the 2D Wavelet Transform

A 2D wavelet transform is used to code both active and moving scenario blocks. As to the static scenario, the wavelet coefficients are computed for only the first frame in each GOK; these are also used for the reconstruction of the background in the remaining frames in the group of key-frames.

Wavelet coefficient statistical distribution at a given resolution and orientation is symmetric with a nearly zero mean and small variance. Generalized Gaussian distribution does approximate quite well such distribution [11]. This generic property of wavelet coefficients suggests the separation, in each subband, of a zone characterized by most of the frame informative content (that is, a *spatially active zone*). In order to guarantee for a higher image quality, this zone has to be more accurately coded with respect to the rest of the frame. In this work, we deployed Daubechies wavelets and we make use of the heuristic algorithm for identifying the spatially active zones proposed in [14]. This is applied to all the subbands but the LL, which is treated as made of only active zones due to its intrinsic importance.

Let S^m be the m-th subband in one-level wavelet decomposition; we denote by $w^m_{x,y}$ (x=1,…,X/2, y=1,…,Y/2) the wavelet coefficients of S^m, and by $p^m(\upsilon)$ the histogram of $w^m_{x,y}$, where υ ranges between the minimum and the maximum wavelet coefficients values. In $p^m(\upsilon)$, we identify two thresholds υ^m_1 and υ^m_2 starting from the wavelet coefficient value with the highest frequency ($\upsilon^m_{\max}$) and moving to the tails of the distribution, as shown in Fig. 3. These thresholds are identified on the basis of the following condition: $\int_{\upsilon_1^m}^{\upsilon_2^m} p^m(\upsilon)d\upsilon = K$, where K is a parameter selected in the range $(0,1]$. These thresholds identify the wavelet coefficients constituting the *active zone* in S^m, that is, $S^m_{az} = \{(x,y): w^m_{x,y} \notin [\upsilon^m_1, \upsilon^m_2]\}$. Accordingly, an active zone is identified by those coefficients with values located on the distribution's tails.

On the basis of this classification process, a binary-value mask is generated, indicating the position of active zone coefficients within the subband. The coefficients not belonging to an active zone are discarded, while the S^m_{az} coefficients are fractal encoded. The K parameter is the same for all the subbands and controls the speed up and the accuracy of the fractal coding process. Indeed, the higher the values of K, the higher the speed-up factors and the lower the final visual quality achieved.

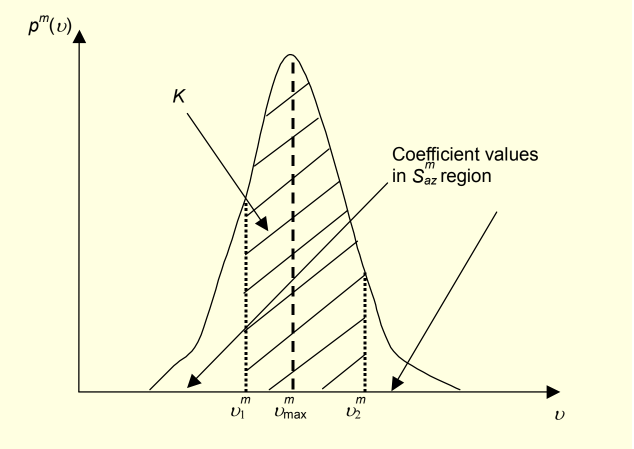

Fig. 3. Classification of the wavelet subband coefficient based on histogram $pm(\upsilon)$: the coefficients not belonging to the range between $\upsilon_1{}^m$ and $\upsilon_2{}^m$ belong to the active zone.

4. Application of 3D and 2D Fractal Coding

3D fractal coding is applied to the active scenario within a group of pictures, and a different code is separately computed for each subband. To generate the range cubes, an adaptive partition, named octree, has been used, and is shown in Fig. 4. This allows us encoding a GOK (more precisely, the active scenario in a GOK) by using larger range cubes in homogeneous regions and smaller ones in regions characterized by spatial/temporal details. Accordingly, the active scenario is initially divided in $R \times R \times R$ cubes (R represents the highest and starting dimension for a range cube), and to these the fractal coding is applied. Then, a coding distortion is separately computed for each of the eight $R/2 \times R/2 \times R/2$ sub-cubes within each $R \times R \times R$ one. If the distortion measure for at least one sub-cube is higher than a prefixed threshold (T_{RC}), the basic cube is divided into these eight sub-cubes. This process is then applied to each of these sub-cubes; it is recursively carried out until the distortion measure is lower than the defined threshold for each sub-cube or a maximum number of octree partitioning levels has been reached.

The use of 3D fractals in video sequence coding allows us to obtain good results in terms of compression ratio and video quality. This justifies its use, notwithstanding the computational complexity of the encoding phase (in non-real-time applications). But the most important advantage of using this technique is the possibility to obtain an approximation of the original image at a resolution different from the one used for code generation, thus obtaining an expansion of the original data. This is performed by increasing the range and domain cube sizes during decoding while leaving the fractal code τ unchanged.

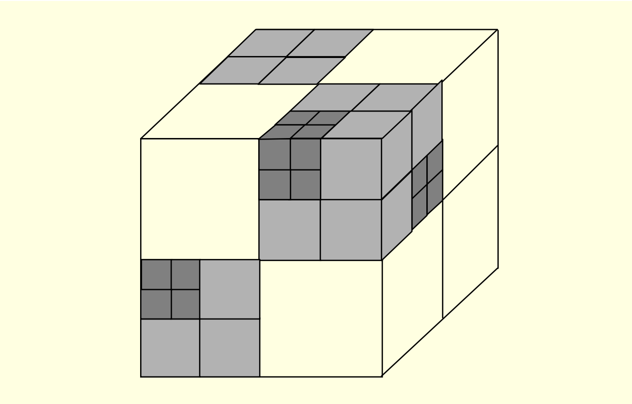

Fig. 4. A two-level octree GOK partition.

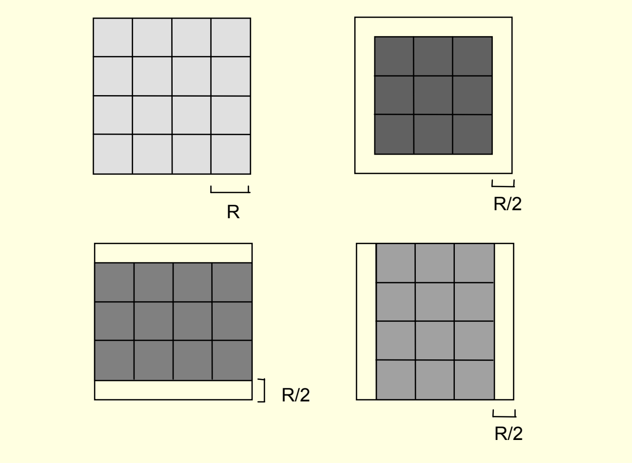

Fig. 5. Overlapped range blocks: four different partitions of each frame in range blocks are used when coding the LL subband.

On the other hand, due to the implicit independent coding of adjacent blocks/cubes in the image/video, fractal coding introduces a disturbing blockness effect in the decoded image/video sequence. This is particularly visible in edged areas and increases as the applied decoding zoom-in factor increases.

In [13], a variant at the classical fractal coding is proposed to reduce blockness distortion in 2D fractal coding: Overlapped range blocks (ORB) are used instead of a normal image partition. This consists in taking four partitions of the image according to the sketches in Fig. 5: The first is obtained with the normal partitioning of the frame in the range blocks; the others are constructed to have blocks overlapping those resulting from the first partition. The fractal code $\tau^{(j)}$ for each partition j (j=1,…,4) is separately generated and coded.

As a result, in the image there are three different regions, as shown in Fig. 6. Each region C_i (i=I, II, III) is characterized by having a different number of fractal codes: one, two, or four. During decoding, the image grey level is obtained by combining the results of the different fractal decoding. In particular, the average of the values obtained with the different partitions of the original image is applied. In C_I, the averaging is computed from all the four fractal codes, and in C_{II} the averaging is computed only on two values, while in C_{III} the averaging is not applied since only the results of the first partition are available.

The main problem with this technique is that the averaging produces a heavily smoothing effect on the image. To face this problem, a fractal decoding with adaptive averaging is proposed in [6]. It is based on the following consideration: In the case of perfect fractal coding, the four independent codes (in C_I) provide quite similar results. It is reasonable to suppose that the fractal coding has a rather stable behavior and then to assume that better results can be obtained by obtaining the average from only the two closest results among the four. These should reduce the smoothing effect in the central part of

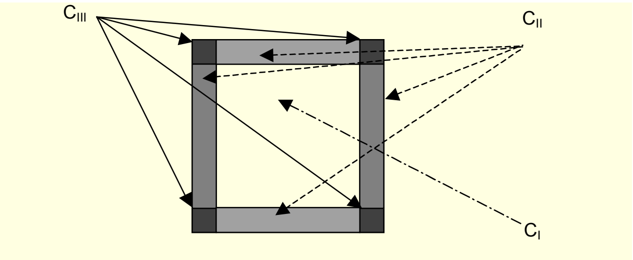

Fig. 6. By using the four-partitioning in Fig. 5, we generate three different regions in each frame: four different fractal codes are available for C_I, only two for C_{II}, and only one for C_{III}.

the frame. Differently, for the other regions, the normal averaging is performed.

We extended this methodology to the 3D fractal coding by dividing the active scenario in the GOK into four different partitions. This constitutes the starting partition from which we derive the *octree* according to the previously defined procedure. To reduce the computation overhead and to limit the increase of the number of bits necessary to code the video, the ORB is applied only to the LL subband.

To increase the post-processing performances, we applied this procedure on a 3×3 mask centered in the pixel to be decoded. Then, in C_I we estimate the filtered value based on 36 values (9 for each partition). In particular, we get the median value of these, ordered square overlapping (OSO). This turns out to be a better estimation than that proposed in [6] due to the exploitation of the correlation between adjacent pixels. A new problem may arise when using ORB/OSO and the separate coding of the static and active scenarios. A situation could happen where the two scenarios do not match, introducing an undesired visual effect around the active scenario. To overcome this problem, the background mask is enlarged to get an overlapping between the two scenarios. This is obtained by applying the dilation operator to the background mask.

Fractal coding is also applied to the background scenario but only along two directions since we code the background information for only the first GOK frame. To identify the active blocks, a special short code is used to represent these. This expedient is also used when introducing the background information into the decoded active scenario.

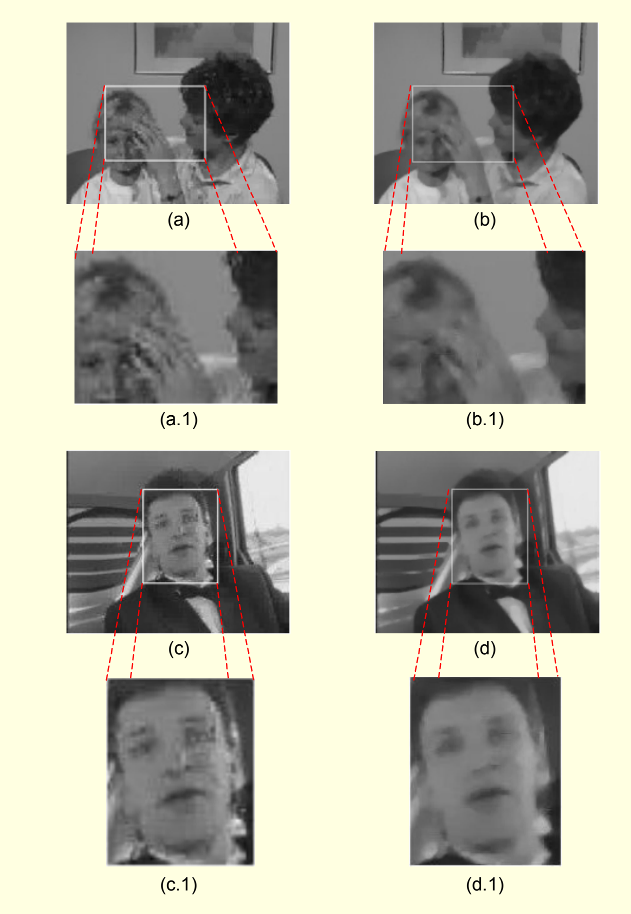

Fig. 7. Enlarged areas captured from the video test sequence 3D decoded: a) and c) without ORB/OSO; b) and d) with ORB/OSO.

IV. Experiments

Extensive experiments have been carried out on several sequences characterized by different spatial and temporal activity and different formats. In this section, we present the results relevant to the application of the proposed method to the combination of test sequences, ‘Mother and daughter + Carphone’ and ‘Claire + Miss America,’ each made of 256 frames in CIF format. The test parameters used are shown in Table 1. These have been chosen to achieve a target quality level (PSNR) of about 25 dB. A twofold test procedure has been set up. A first set of experiments has been carried out to show how an opportune policy of problem simplification based on video processing techniques allows one to address the issue of video summarization and presentation of a multimedia content by means of fractals. A further testing phase has been performed with the aim of comparing the efficiency of the proposed scheme to alternative techniques in terms of the overall visual quality.

Table 1. Experiment settings.

T_{hist}	GOK	T_{AB}	T_{back}	T_{RC}	R	Octree level
13	8 frames	7	30%	30	$4 \times 4 \times 4$	3

As to the first experiment set, the aim was to analyze the impact of each processing block deployed in the devised system with reference to the sketch shown in Fig. 1. In the following, we report the results obtained in terms of compression ratio, final visual quality, and total processing time at both coding and decoding sides. In particular, Fig. 7 highlights the visual benefits achieved using ORB/OSO procedures by showing two representative enlarged areas; blockness artifacts are heavily reduced in both test sequences. On the other hand, such a procedure introduces an increase in the coding/decoding times, which, however, can be limited with the help of a subband wavelet coding as shown in Tables 2 and 3. In Table 2, the last three columns refer to the encoding times when applying the proposed method without the ORB module, with it,

Table 2. Key-frames coding time comparison: without ORB, with ORB, and with ORB and wavelet.

Coding					
Sequence	Number of key-frames	Format	Time (s)		
			No ORB	ORB	ORB and wavelet
Mother&Daughter + Carphone	58	QCIF	44	173	54
Claire + Miss America	56	QCIF	18	164	64

Table 3. Decoding time comparison: without OSO, with OSO, and with OSO and wavelet.

Decoding					
Sequence	Number of frames	Format	Time (s)		
			No OSO	OSO	OSO and wavelet
Mother&Daughter + Carphone	256	CIF	20	342	118
Claire + Miss America	256	CIF	19	337	120

Table 4. Benefits of introducing the background separation in terms of compression ratio and decoding time.

Sequence	Format	No background separation		Background separation		H.264/AVC	
		CR	Time (s)	CR	Time (s)	CR	Time (s)
Mother& Daughter + Carphone	CIF	110	118	125	87	670	4.8
Claire + Miss America	CIF	132	120	171	44	710	4.3

and with it while using the wavelet transform. The last three columns in Table III refer to the opposite procedure at the decoder. Note that the coder has found 58 and 56 key frames in the two test sequences, and these frames have been coded in QCIF format. The reduction in the processing time when using the wavelet is quite evident. This is due to the fact that, in this case, the median filtering is applied to a smaller number of coefficients, which are those relevant to the LL subband only.

The results in Table 4 show the compression performance improvement obtained by the introduction of the background separation procedure. As to this, the improvement for the sequence 'Claire + Miss America' is higher than that for the sequence 'Mother and Daughter + Carphone.' As a matter of fact, the former sequence is composed by frames with a strong inter-frame temporal correlation (low velocity in the movements of the objects) and a large background area in the entire sequence (small area occupied by the moving objects). This characteristic also influences the outcome of the next experimentation that was set up to study the impact of the background separate coding in terms of decoding times. A decrease in the decoding time was expected with separate encoding of the background, since the corresponding area is decoded once for the entire GOK. This was confirmed by the experiments: Decreases in the decoding time equal to 26% and 63% have been computed for the two sequences when the selection of the background is applied. This reflects the fact that the advantage in the selection of the background regarding the decrease of the decoding times is strictly related to the characteristics of the sequence to be encoded. In Table 4, we are also comparing the performance of the proposed algorithm with respect to those obtained with a video compression standard. In particular, we have compressed the key-frames at the same spatial resolution (QCIF), and final visual quality with the last coding standard H.264/AVC, which is known to provide quite high compression ratios. This comparison allows us to evaluate the cost in terms of compression ratio and decoding time that we have to pay in order to have the zooming functionalities provided by the proposed scheme.

In the second experiment phase, we analyzed the performance and effectiveness of the proposed method by comparing the achieved results to those obtained, under the same constraints (that is, spatial and temporal expansion), by applying the frame replication and classical interpolation techniques. In fact, the originality of our work within the framework of video summarization is in the use of the properties of implicit interpolation of fractals during the decoding stage of the process to obtain an enhanced version of the sequence. This motivates the use of the state-of-the-art techniques, such as frame replication and interpolation, as benchmark systems.

A measure of the overall visual quality achieved was obtained by comparing the expanded sequence to the original video. To measure the quality achieved in the reproduction of the considered test sequences, we refer to the video quality assessment formalized in [14] and [15]. As to this, the jerkiness is defined as the perception, by human vision faculties, of originally continuous motion as a sequence of distinct 'snapshots.' Usually, jerkiness is present when the position of a moving object within the video scene is not updated rapidly enough. More in general, the total error generated by an incorrect coding of a moving object on a video sequence is representative of spatial distortion and incorrect positioning of the object. Three features can be extracted to measure the above impairments. One of these features is extremely related to jerkiness, while the other two represent a measure of the average and total distortion of the expanded video due to both jerkiness and spatial artifacts such as blurring/smearing, blocking, edge busyness, and others. The features are computed by comparing the original and expanded sequences. For that reason, the extracted features belong to the class of reference metrics [14].

The features extraction is accomplished according to the process as follows. A single-frame temporal alignment of the expanded and original video is performed. For each aligned video image pair, a difference image is created by subtracting the image of the expanded sequence from the image of the original. The standard deviation of such difference images (SDDI) is calculated. From the temporal history of the SDDI, the following three features are then computed:

- The temporal mean of SDDI (TM-SDDI) mainly related to the average distortion caused by spatial distortion and jerkiness
- The temporal standard deviation of SDDI (TSD-SDDI) primarily associated to jerkiness
- The temporal root mean square of SDDI (TRMS-SDDI) representative of the total spatial distortion and jerkiness

Details on the exact computation of the above metrics are given in appendix A.

Tables 5 and 6 report the normalized values of the three features previously described for the two test sequences. In both cases, the results show that the proposed method overcomes the two benchmark techniques in terms of both spatial distortion and jerkiness. As to jerkiness, this is due to the capacity of the fractal interpolation to approximate with high accuracy three-dimensional block motion. The use of a 3D fractal code supported by ORB/OSO processing allows smoothing the total motion of the sequence, thus enhancing the quality of the final presentation of the sequences without the introduction of time discontinuity and avoiding artificial jerky motion. The jerkiness benefits of the proposed method with respect to the benchmark systems are more perceptible in the 'Mother and Daughter + Carphone' sequence. In fact, this sequence presents a higher temporal variation compared to the other. Indeed, for both sequences the proposed technique provided the same jerkiness level (that is, TSD-SDDI equal to 0.04), whereas the classic frame replication and interpolation techniques did better with the 'Claire + Miss America.' The reason for this outcome has still to be found on the low temporal activity within the sequence. The different nature of the two test sequences also influences the performance of the compared systems in relation to the total and average spatial distortion represented by the TM-SDDI and TRMS-SDDI features, respectively. For 'Mother and Daughter + Carphone' the experimentation provided higher values than for 'Claire + Miss America.' In both cases, the performance of the compared methods is similarly affected by the nature of the sequences.

Table 5. Normalized values of the quality metrics for 'Mother & Daughter' and 'Carphone.'

Metric	Fractal	Frame replication	Interpolation
TM-SDDI	0.43	0.52	0.46
TSD-SDDI	0.04	0.07	0.06
TRMS-SDDI	0.44	0.54	0.46

Table 6. Normalized values of the quality metrics for 'Claire' and 'Miss America.'

Metric	Fractal	Frame replication	Interpolation
TM-SDDI	0.17	0.21	0.18
TSD-SDDI	0.04	0.06	0.05
TRMS-SDDI	0.14	0.22	0.16

The results presented in this section refer to a situation of dense key-frames selected from the original video sequence. This is an advantage for the video presentation operation, which can exploit the high degree of similarity between consecutive frames. Differently, in the case of sparse frames, the performance would be worse, and in some circumstances the reconstruction of missing data may fail. This problem has been investigated with the last experimentation test. By referring to the *histogram comparison* procedure for key-frames selection, we set the threshold T_{hist} to higher values to achieve a more sparse key-frames distribution. Figure 8 shows the behaviour of the quality metrics for 'Mother & Daughter' and 'Carphone.' As expected, the results show that when increasing the distance between consecutive key-frames the performance of the three compared methods decreases. The frame replica suffers more than the others from the missing correlation between distant frames. The simple replication of the spatial and temporal information produces a fast degradation of the performance for all the quality parameters for T_{hist} higher than 20, which approximately corresponds to selecting one frame for every 25. The proposed method and the interpolation show a similar behaviour with a predominance of the proposed system. This is a confirmation that the fractal interpolator outperforms classical interpolators on zooming applications [6]. Similar results have been obtained with the sequences 'Claire' and 'Miss America.'

Finally, for the sake of fairness it is worth noting that all tests have been performed on an MS-Windows environment running on an INTEL Pentium IV - 1.4 GHz machine with a RAM memory of 256 MB. Due to the inherent structure of the proposed codec scheme, which allows a rather naturally parallelization of the whole process, we can say that a drastic

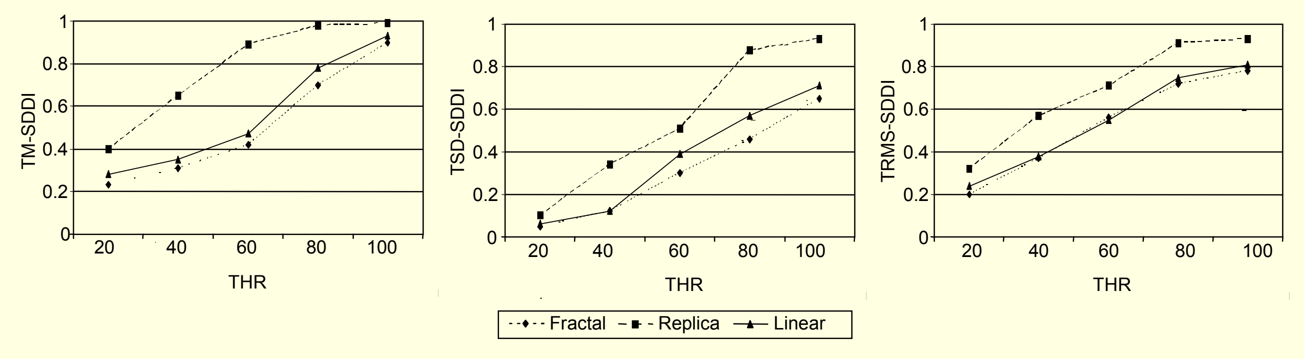

Fig. 8. Behaviour of the quality features vs. the histogram threshold for ‘Mother & Daughter’ and ‘Carphone.’

reduction of the processing delay can reasonably be obtained in the case of parallel processor systems. This is a realistic scenario in the context of a database centralized system in charge of the video abstraction. Indeed, within the proposed scheme the coding stage is the time consuming part of the whole process. On the other hand, the decoding, and thus the content extraction procedure performed at the user side, does not require any particular powerful system.

V. Conclusions

A novel and promising approach to the coding of video key-frames based on the joint use of fractals and wavelets has been proposed. This aims at exploiting the advantages of fractals to expand a video sequence in both temporal and spatial dimensions in a ‘natural’ way. This feature represents an important novelty of this work with respect to the alternative approaches, which mainly focus on the compression ratio without taking into account the presentation aspect of the video summary. From these experiments, we have observed that the use of the wavelets and the separate coding of the background and foreground components allow one to both increase the compression ratio and reduce the decoding time.

In this work, we have used some key-frame extraction procedures that identify the key-frames at scene changes. However, for summarization purposes, it is sometimes better to select representative frames from the middle of the scenes, being more informative than those at the end of the scenes. Indeed, this observation requires the introduction of some changes in some steps of the proposed technique, such as background coding and key-frame grouping. This issue will be addressed in future work.

Appendix

Let A, B be two sequences composed by n frames of N×M size, single-frame temporal aligned; let each pair of video frames be represented by the index p={1, 2, 3,..., n} and call $d_p(i, j)$ the difference image created from each pair p by subtracting the image of sequence A from the image of sequence B. Then,
the standard deviation of $d_p(i,j)$, $SDDI_p$ is

$$SDDI_p=\sqrt{\left[\frac{1}{N\times M}\sum_i\sum_j d_p^2(i,j)\right]-\left[\frac{1}{N\times M}\sum_i\sum_j d_p(i,j)\right]^2}.$$

The temporal mean of SDDI (TM-SDDI) is

$$TM-SDDI=\frac{1}{n}\sum_{p=1}^{n}(SDDI_p).$$

The temporal standard deviation of SDDI (TSD-SDDI) is

$$TSD-SDDI=\sqrt{\frac{1}{n}\sum_{p=1}^{n}SDDI_p^2-(TM-SDDI)^2}.$$

The temporal root mean square of SDDI (TRMS-SDDI) is

$$TRMS-SDDI=\sqrt{\frac{1}{n}\sum_{p=1}^{n}(SDDI_p)^2}.$$

References

[1] N. Dimitrova, H. Zhang, B. Shahraray, M. Sezan, T. Huang, and A. Zakhor, “Applications of Video-Content Analysis and Retrieval,” *IEEE Multimedia*, vol. 9, no. 3, July-Sept. 2002, pp. 44-55.

[2] A.M. Ferman and A.M Tekalp, “Two-Stage Hierarchical Video Summary Extraction to Match Low-Level User Browsing Preferences,” *IEEE Trans. Multimedia*, vol. 5, Issue 2, June 2003, pp. 244-256.

[3] Ma Yu-Fei and Hong-Jiang Zhang “Video Snapshot: A Bird View of Video Sequence,” *Proc. 11th Int’l Multimedia*

Modelling Conf. MMM 2005, 12-14 Jan. 2005, pp. 94-101.

[4] Shi Lu Lyu and M.R. King, “Video Summarization by Spatial-Temporal Graph Optimization,” *Proc. Int’l Symp. on Circuits and Systems*, vol. 2, 23-26 May 2004, pp. 197-200.

[5] S. Barnsley and M.F. Demko, “Iterated Function Systems and the Global Construction of Fractal,” *Proc. Royal Soc. London*, Ser. A399, 1985, pp. 243-275.

[6] A.E. Jaquin, “Image Coding Based on a Fractal Theory of Iterated Contractive Image Transformation,” *IEEE Trans. Image Processing*, vol. 1, no. 1, Jan. 1992, pp. 18-30.

[7] M. Polvere and M. Nappi, “Speed-Up in Fractal Image Coding: Comparison of Methods,” *IEEE Trans. Image Processing*, vol. 9, no. 6, Sept. 2000, pp. 1002-1009.

[8] K.U. Barthel and T. Voye, “Three-Dimensional Fractal Video Coding,” *Proc. IEEE Int. Conf. Image Processing*, pp. III 260-263, Washington, D.C., 1995.

[9] I. Koprinska and S. Carrato, “Video Segmentation: A Survey,” *Signal Processing: Image Communication*, vol. 16, no. 5, Jan. 2001, pp. 477-500.

[10] A. Nagasaka and Y. Tanaka, “Automatic Video Indexing and Full-Video Search for Object Appearances,” *E. Knuth, L.M. Wegner,* eds., *Visual Database Systems II*, Elsevier, Amsterdam, 1995, pp. 113-127.

[11] M. Vetterli and J. Kovacevic, *Wavelets and Subband Coding*, Prentice Hall, U.S.A. 1995.

[12] M. Ancis and D.D. Giusto, “Image Data Compression by Adaptive Vector Quantization of Classified Wavelet Coefficients,” *Proc. IEEE PACRIM Conf.*, Victoria, Canada, 1997, pp. 330-333.

[13] E. Reusens, “Overlapped Adaptive Partitioning for Image Coding Based on Theory of Iterated Function Systems,” *Proc. IEEE ICASSP*, vol. 5, Adelaide, Australia, 1994, pp. V/569-V/572.

[14] S. Wolf, NTIA Report 90-264, “Features for Automatic Quality Assessment of Digitally Transmitted Video,” 1990.

[15] ANSI T1.801.03-1996, “American National Standard for Telecommunications–Digital Transport of One-Way Video Signals–Parameters for Objective Performance Assessment,” *Alliance for Telecommunications Industry Solutions*, 1200 G Street, N. W., Suite 500, Washington DC, 2005.

Luigi Atzori was born in Cagliari, Italy, in 1971. He received the MS degree in electronic engineering in 1996 and the PhD degree in electronic and computer engineering in 1999, both from the University of Cagliari. In 1994, he spent four months as a Visiting Student at the Technical University in Braunschweig, Germany. At present, he is an Assistant Professor in Telecommunications at the Dept. of Electrical and Electronic Engineering, University of Cagliari. His main research topics of interest are in multimedia signal processing and transmission: error recovery and concealment, video post-processing, IP Telephony playout buffering, and video streaming. He is also interested in data network performance analysis. He has published more than 40 journal articles and refereed conference papers. He has been awarded a Fulbright Scholarship (11/2003-05/2004) to work on video streaming at the Department of Electrical and Computer Engineering, University of Arizona, and is a member of AEI.

Daniele D. Giusto received the MS degree in electronic engineering and the PhD degree in telecommunications from the University of Genoa, Genoa, Italy, in 1986 and 1990. Since 1994, he has been a permanent faculty member in the Department of Electrical and Electronic Engineering, University of Cagliari, Cagliari, Italy, where he became a Full Professor of Telecommunications in 2002. In 1995 and 1998, he was a Visiting Professor at the Institute for Telecommunications, Technical University of Braunschweig, Germany. His research interests are in the areas of image and video processing and coding, multimedia systems, digital television, pictorial databases, and personal communications. He is a Member of the Executive Board of CNIT, the Italian University Consortium for Telecommunications. He was the recipient of the 1993 AEI Ottavio Bonazzi Best Paper Award and co-recipient of the 1998 IEEE Chester Sall Best Paper Award. Since 1999, he has been acting as the Head of the Italian delegation within the ISO-JPEG international standardization committee. He has been a Guest Editor for several journals and acted as General Chair for the PACKET VIDEO 2000 International Workshop and the 1st International IEEE-SPIE Workshop on JPEG2000.

Maurizio Murroni graduated with honors (Summa cum laude) in electronic engineering in 1998 at the University of Cagliari, and in the same year he received an award for his thesis from Telecom Italia, Inc. In 1998, he became an Erasmus visiting student at CVSSP Group under Prof. Maria Petrou, in the School of Electronic Engineering, Information Technology and Mathematics, University of Surrey, Guildford, U.K. and a visiting PhD student at the Image Processing Group under Prof. Yao Wang, Polytechnic University, Brooklyn, NY, USA, in 2000. In 2001, he received the PhD degree in communications, from the University of Cagliari, and in December 2002, he became an Assistant Professor of Communications at the Department of Electrical and Electronic Engineering (DIEE) of the University of Cagliari. Since 1998, he has contributed to the research and teaching activities of the Multimedia Communication Lab (MCLab) at DIEE. His research focuses on multimedia data transmission and processing, traffic modeling, and QoS for new technology multimedia networks. He is a member of IEEE.