Video Compression Techniques: A Survey

Qingrong Guo

Supervisor: Carl McCrosky

Abstract

Numerous bandwidth compression techniques have been developed, such as DPCM, transform coding, and vector quantisation. Based on these techniques, several video compression techniques have been defined. Some of them have been standardised, such as JPEG and MPEG. Some potentially better techniques are under development, such as fractal compression and wavelet transform compression. In this paper, we discuss these techniques in terms of their underlying principles, their implementation mechanisms, and their performance.

1 Introduction

Video transmission is one of the most important high speed network multimedia applications. The traffic characteristics of video sources are significantly different from those of other multimedia sources such as voice and data. Video will have the largest bandwidth requirements and video traffic is highly delay sensitive. For these reasons, video compression is indispensable in video transmission. Also, compressed video information can save storage space.

Video compression codecs generally output variable bit rate (VBR) streams. Different video quality requirements will cause different degrees of bit rate variance. VBR traffic will influence the design, control, and performance analysis of high speed networks. For these reasons, we would like to clearly understand the mechanisms of video coding.

Numerous bandwidth compression techniques have been developed, such as differential pulse code modulation (DPCM), transform coding, and vector quantisation (VQ). Based on these techniques, several video compression techniques are built. Four of the video compression techniques are well known today. They are: Joint Photographic Experts Group (JPEG), a DCT-based transform coding technique; Motion Picture Experts Group (MPEG), a hybrid of DCT-based transform coding and DPCM; fractal image compression, a special kind of transform coding technique; and wavelet transform coding, a technique of combining wavelet transform and vector quantisation. We review these techniques in later sections.

2 Background

2.1 Sources of Redundancy

There are three types of redundancy that can be exploited by image compression systems:

- Spatial Redundancy. In almost all natural images, the values of neighbouring pixels are strongly correlated.
- Spectral Redundancy. Physically, light is made up of an infinite spectrum of frequencies. In image representation, we use a finite set of discrete frequency values to specify color. This finite

set of intensity values is called the spectral band. In images composed of more than one spectral band, the spectral values for the same pixel location are often correlated.

- Temporal Redundancy. Adjacent frames in a video sequence often show very little change.

The compression techniques that handle spatial and spectral redundancy are called intraframe compression techniques; the compression techniques that handle temporal redundancy are called interframe compression techniques.

2.2 Color Systems

The information in each frame consists of pixel values which represent the color of the pixel. Different color is actually caused by the different intensity value to each primary color frequencies (Red, Green, Blue). The format of a color image is dependent on the color system it uses.

The basic system is the (R, G, B) system. It separated a color image into R, G, and B components. Each component consists of different bits (often 8 or 24 bits) for each of these color intensities.

The (R, G, B) system is understandable, but it is not always the most convenient color coordinate system to use. There are two other coordinate systems that are commonly used, which can be defined in terms of R, G, and B. They are called the (Y, I, Q) system and the (Y, U, V) system, respectively. In each of the two systems, the gray-scale component is separated from the color information.

In the (Y, I, Q) system, the Y vector, which represents a weighted sum of the three intensities, is called the luminance. The remaining two components determine color characteristics called hue and saturation. These two components collectively are known as the chrominance. This color system is used by North American television system.

The (Y, U, V) basis is defined similarly. This system is used by European television system. Also, it is used in image compression techniques discussed below.

The difference between these two systems can be seen in their relations with (R, G, B) system:

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.275 & -0.321 \\ 0.212 & -0.528 & 0.311 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & -0.5151 & -0.100 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

One reason for using these two systems comes from the fact that the variance of the chrominance data tends to be less than that of the luminance data. By separating the components in this way, it is easier to handle in video compression.

3 JPEG Compression

JPEG is a standardised image compression mechanism. It is designed for compressing either full-color or gray-scale digital images of "natural" real-world scene. It does not work so well on non-realistic images, such as cartoons or line drawings. It generally does not work well on black-and-white images nor on motion picture compression.

3.1 Operation Modes

The JPEG standard encompasses a variety of image formats, both lossless and lossy. There are four main kinds of JPEG compression [1]:

- Sequential encoding: each image component is encoded in a single left-to-right, top-to-bottom scan;

- Progressive encoding: the image is encoded in multiple scans for applications in which transmission time is long (low bandwidth between coder and decoder). It offers a means of producing a quick "rough" decode image;
- Lossless encoding: the image is encoded to guarantee exact recovery of every source image sample value. In ordinary situations, lossy encoding is good enough for humans, because the algorithm achieves much of its compression by exploiting the limitations of the human eye. The lossless mode is for machine-analysis of images;
- Hierarchical encoding: the image is encoded at multiple resolutions, so that lower-resolution versions may be accessed without first having to decompress the image at its full resolution. This mode is used to code an input image as a sequence of increasingly higher-resolution frames.

3.2 Compression Steps

Although the JPEG standard proposes the four operation modes, most of the implementations now on the market focus on the Baseline sequential codec (coder/decoder). The Baseline sequential codec is a rich and sophisticated compression method which will be sufficient for many applications. It represents the main ideas of JPEG. Often, the term JPEG refers to the Baseline sequential codec method.

JPEG is a DCT (Discrete Cosine Transform) based method. We now take the Baseline sequential codec as the example to discuss the processing steps of this method.

Figure 1 [1] shows the key processing steps which are the heart of the DCT-based modes of operation.

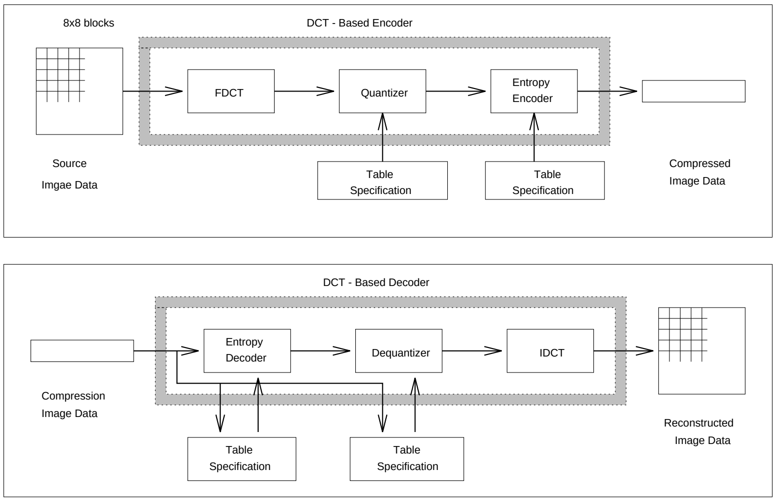

Figure 1: Coding/decoding Steps of DCT-based compression

3.2.1 DCT Encoding

At the first step, the input image is partitioned into non-overlapping 8×8 pixel blocks of component samples. Then, the blocks are transformed to the frequency domain using FDCT (Forward Discrete Cosine Transform) [1]. The equations for the forward and inverse discrete cosine transforms are::

$$F(u,v) = \frac{1}{4}C(u)C(v)[\sum_{x=0}^{7}\sum_{x=0}^{7} f(x,y) * cos\frac{(2x+1)u\pi}{16}cos\frac{(2y+1)v\pi}{16}] \qquad (3-1)$$

$$f(u,v) = \frac{1}{4}[\sum_{x=0}^{7}\sum_{x=0}^{7} C(u)C(v)f(u,v) * cos\frac{(2x+1)u\pi}{16}cos\frac{(2y+1)v\pi}{16}] \qquad (3-2)$$

where C(u), C(v) = $1/\sqrt{2}$ for u, v = 0; C(u), C(v) = 1 otherwise.

f(x,y) is the function of the two spatial dimension x and y. It reflects the signal of color and intensity represented by (Y, U, V). The FDCT takes such a signal as its input and decomposes it into 64 orthogonal basis signals. Each contains one of the 64 unique two-dimension "spatial frequencies" which compose the input "signals spectrum".

The output, F(x,y), is the set of 64 DCT coefficients which can be regarded as the relative amounts of the 2D spatial frequencies contained in the 64-point input signal.

By the nature of DCT, most of the energy of the component samples' blocks have been concentrated into a few coefficients, usually in the top-left corner of the DCT block, which is called the DC coefficient. The other 63 coefficients are called AC coefficients.

For a typical 8×8 sample block from a typical source image, most of the spatial frequencies have zero or near-zero amplitude and need not be encoded.

The FDCT process is shown in Figure 2 [6]. For simplicity, we use a 4x4 block instead of an 8x8 block.

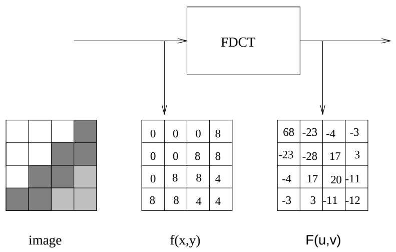

Figure 2: An intuition illustrate of FDCT

3.2.2 Quantisation

The next step, quantisation, is the key to most of the JPEG compression. A 64-element quantisation matrix, where each element corresponds to a coefficient in the DCT block, is used to reduce the amplitude of the coefficients, and to increase the number of zero-value coefficients. Quantisation and dequantisation is performed according to the following equations:

$$F^Q(u,v) = Integer\ Round(\frac{F(u,v)}{Q(u,v)}) \qquad (3-3)$$

$$F^{Q'}(u,v) = F^Q(u,v) * Q(u,v) \qquad (3-4)$$

The last step before the final entropy coding is encoding the DC coefficient by a simple DPCM coding, and ordering the coefficients in a zigzag sequence. DPCM is used because the adjacent blocks will probably have similar average intensities. Also, the DC coefficient contains a significant fraction of the total image energy. Ordering the coefficients will help to facilitate entropy coding by placing low-frequency coefficients (which are more likely to be nonzero) before high-frequency coefficients. The Principle of DPCM can be found in [11].

3.2.3 Entropy Coding

The final DCT-based encoder processing step is entropy coding. This step achieves additional compression losslessly by encoding the quantised DCT coefficients more compactly based on their statistical characteristics.

The JPEG proposal specifies two entropy coding methods: Huffman coding and arithmetic coding. The Baseline sequential codec uses Huffman coding. We do not discuss the detail of these two methods in this paper. They can be found in [3].

Table 1: Quality Levels for Different Compression Rates

Compression Rate (bits/pixel)	Quality
0.25 - 0.5 (96:1 - 48:1)	moderate to good quality, sufficient for some application
0.5 - 0.75 (48:1 - 32:1)	good to very good quality, sufficient for many application
0.75 - 1.5 (32:1 - 16:1)	excellent quality, sufficient for most applications
1.5 - 2.0 (16:1 - 12:1)	indistinguishable from the original, sufficient for the most demands
12 (2:1)	identical to original, lossless compression

The DCT-based decoding process is the inverse process of encoding. It uses the IDCT equation instead of the FDCT in the DCT step, and uses the dequantization equation instead of the quantisation equation.

We have been concentrating on the encoding/decoding of the baseline sequential codec. The process of progressive mode and hierarchical mode is more complex. For example, a DCT progressive mode codec needs an image buffer prior to the entropy coding, so that an image can be stored and then parcelled out in multiple scans with successively improving quality. For the hierarchical mode of operation, a buffer is also needed to build blocks within a larger framework. The lossless mode is a totally different method. It uses DPCM coding, instead of DCT and quantisation (the source of loss). Obversely, it can only get small compression rate. The detailed description of these three compression modes of JPEG can be found in [1].

3.3 Quality of Compression and Quality

The goal of JPEG is to meet the requirement of a variable picture quality and compression rate. The JPEG standard considers the compression rate and image fidelity at the same time. A wide range of image quality ratings is provided; also, the encoder is parameterisable, so that the user can set the desired compression/quality tradeoff. Table 1 characterises the relationship between the compression rate and quality provided by JPEG. Three bytes per pixel is assumed. We can see from the table that JPEG can provide a compression rate of nearly 100:1. At compression rates of 20:1 to 30:1, the quality of the picture is still "excellent".

4 MPEG Compression

MPEG is an international standard for the compression of digital audio and video transmission. What we consider here is the MPEG first-phase (MPEG-1) video compression standard.

4.1 The Challenge of Motion Picture Compression

The MPEG compression approach uses a combination of JPEG (for still images) and CCITT H.261 [5] (for video conferencing) standards. Because video is a sequence of still images, it is possible to compress the image using techniques similar to JPEG. However, the quality and speed requirement demand a very high compression not achievable with intraframe coding alone. It is necessary to introduce interframe coding techniques to reduce temporal redundancy.

Furthermore, video applications have special requirements that have to be met by the compression techniques. These special features are [4]: random access, fast forward/reverse searches, reverse playback, audio-visual synchronisation, robustness in the presence of error, coding/decoding delay, edit-ability, and format flexibility.

For all these reasons, two interframe coding techniques are introduced to MPEG: predictive and interpolative.

Hence, the MPEG video compression algorithm relies on two basic techniques:

- Block-based motion compensation for reduction of temporal redundancy.

- Transform domain (DCT) based compression for the reduction of spatial redundancy.

The DCT-based compression here is same as the technique in JPEG. We therefore concentrate on temporal redundancy reduction in this section.

4.2 Temporal Redundancy

MPEG exploits temporal redundancy by computing a frame-to-frame difference signal called a prediction error. The technique that does the prediction is called motion compensation. Motion compensation is a block-based approach. The basic unit the motion compensation works on is called a "macroblock", which is of size 16x16 pixels. When encoding a frame, one macroblock in the frame, called "target block", is matched with a set of blocks of the same size in the previous (or next) frame, called reference frames. The block in the reference frame that "best matches" the target block is used as the prediction for the latter. The difference between the two blocks is computed. The result is the prediction error. A vector, called motion vector, is used to describe the displacement (position in frame) between the best-matching block and the target block. This relationship is illustrated in Figure 3.

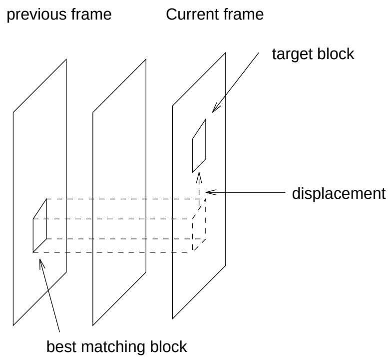

Figure 3: Target block, best-matching block

The motion vector information is also encoded and transmitted along with the prediction error. The prediction error itself is transmitted using the DCT-based intraframe encoding techniques we discussed before.

4.3 Motion Compensation

In simple words, motion compensation produces the prediction and motion vector. There are two kinds of motion compensation techniques in MPEG: prediction and interpolation.

Prediction assumes that "locally" the current picture can be modelled as a translation of the picture at some previous time. Locally means that the amplitude and the direction of the displacement need not be the same everywhere in the picture.

Interpolation, also called bidirection temporal prediction, is a key feature of MPEG. In interpolation, some of the video frames are encoded using two reference frames. One in the past and one in the future. A block in those frames can be predicted by another block from the past reference frame (forward prediction), or from the future reference frame (backward prediction), or by the average of two blocks one from each reference frame (interpolation). In every case, the block from the reference frame is associated with a motion vector. So interpolation uses two motion vectors. The interpolation technique improves random access, reduces the effect of errors, and contributes significantly to image quality.

Accordingly, there are four types of macroblock: intra, forward-predicted, backward-predicted, and average. They represent the four different prediction modes.

4.4 Types of Frame

To meet the requirement of random access and implementation of motion compensation, MPEG considers three types of frames. They are:

- Intraframe (I): they are simply a frame coded as a still image, not using any past history. "I" frames provide entry points for random access.

- Predicted frames (P): they are coded with reference to a past picture ("I" or "P") and will in general be used as a reference for future predicted pictures.

- Bidirectional prediction (B): they are predicted from the closest two "I" or "P" frames. One in the past and one in the future. Because it can get the reference from two direction, so it provide the highest amount of compression. Bidirectional pictures are never used as reference.

The sequence of decoded frames usually goes like: IBBPBBPBBPBBIBBPBBPB ... ,where there are 12 frames from I to I. This may be changed according to the time space between each random access point you need. The ratio of P's to B's is based on experience.

4.5 Performance of the Technique

MPEG provides high levels of compression (50:1 to 200:1)[6]. It compresses video signals to about 1.5 Mbps with acceptable quality.

Also, MPEG is very flexible in supporting a wide range of spatial and temporal resolutions. It is capable of using a large range of bit rates.

A new phase of activities of the MPEG committee is studying video compression algorithms for higher resolution signals at compressed bit rates up to 10 Mbps.

5 Fractal Image Compression

Fractal compression is a totally new method in image compression. It is based on the fractal transform, which comes from the theory of fractal geometry. This theory supposes the real world image is naturally "self-similar".

5.1 Principle of the Technique

This technique achieves redundancy reduction by describing the original image through smaller copies or parts of the same image [12].

Fractal image compression is based on iterated function system (IFS) theory [7]. In this theory, Barnsley supposes that all real-world images are rich in affine redundancy; that is, under suitable affine transformation, large bits of real-world image look like smaller bits of the same real-world image. Based on this theory, a block oriented coding scheme is proposed in [8]. The image to be encoded is partitioned into non-overlapping, square blocks, called range blocks. The task of fractal encoding is to find a larger block of the same image called a domain block for every range block in such a way that a contractive transmission τ of the domain block is a good approximation of the range block. τ is a combination of a geometrical transformation, γ, and a luminance transformation, λ.

The data for the transformations are recorded. They are called fractal codes. The compressed image consists of all the fractal codes.

Because it need only a few transformation to generate a close approximation of a scene, and only a few numbers are required to describe each transformation, the image can be represented very compactly.

5.2 Affine Transformation

The central part of the fractal compression is the affine transformation. An affine transformation is a mathematical function made up from some combinations of a rotations, scalings, and translations of the coordinate axes in n-dimensional space. A simple example in two dimensions is [9]:

$$W\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix} = \begin{pmatrix} ax+by+e \\ cx+dy+f \end{pmatrix}$$

The matrix $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ determines the rotation and scaling; $\begin{pmatrix} e \\ f \end{pmatrix}$ determines the translation. The values a, b, c, d, e, and f are the affine coefficients for this transformation.

In order to apply W to big parts and result in the smaller parts, the affine transformation should be "contractive". Here is an example of contractive affine transformation:

$$W\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 0.5 & 0.0 \\ 0.0 & 0.5 \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

An IFS is a collection of contractive affine transformations.

5.3 Coding and Decoding Process

Coding is the process of partitioning the domain block and specifying a set of affine transformations [8]. In the first step, the original digital image is partitioned into non-overlapping range blocks. The set of all range block must cover the entire image. Next, the program defines a collection of possible domain blocks, which must be larger than the range blocks. They can overlap and need not cover the entire image.

For each range block, the program must choose the domain block that, after an appropriate affine transformation is applied, most closely matches the range block. The contractive affine transformation consists of two parts of transformations: the geometrical part γ_i maps image blocks from a domain block to a range block. It shrinks and deforms the image within the domain block. It consists of a spatial contraction and a position shift; the the luminance part λ_i changes the contrast and the brightness of a block. The range blocks have two different sizes. A larger block can be split into up to four non-overlapping smaller blocks. Decisions about the splitting of a large block are made during the encoding of the image block over this block cell. Thus a partition constructed this way is image dependent; it allows the coder to use small blocks to take advantage of smooth varying image areas, and to use small blocks to capture detail in complex areas. That allows control of the compression rate.

The natural decoding scheme simply consists in iterating a fractal code τ on any initial image u_0. To start the decoding, get the first range partition information from the fractal code. For each range block, read the affine coefficients for this range block, locate the domain block specified by the transformation, and map the contents of this domain block to the appropriate range block. When this is done for each range block, a new image is created from the former image. Do the same thing in the second iteration. The iteration can be repeated until a stable decoded image is observed [8].

5.4 Features of the technique

Fractal image compression has some special features compared with JPEG image compression:

- Resolution independent: When an image is captured by a camera or a scanner, it acquires a scale determined by the sampling resolution of that device. If software is used to zoom in on the image, there is no additional detail, just bigger pixels. A fractal image is different. Because the affine transformations are spatially contractive, detail is created at finer and finer resolutions with each iteration. So the fractal image compression is faithful to the analogy image, not only to its digital representation. This allows images to be output at higher resolution than they are input.

- High compression rate: Its compression rate is typically from 4:1 to 100:1. Color images can be compressed to a greater extent than grey-scale image. Some papers indicate that the compression rate can reach 10000:1, but only when zooming into the resolution independent images is considered.

- Quality: Compared with JPEG, fractal encoding is better for high compression rates (> 40:1). JPEG is better for low compression rates.

The main disadvantage of fractal compression is the slow speed of encoding. Encoding may takes very much longer time than decoding. This is because of the pairing process of each range block to a domain block involves a lot of searching. Some papers have proposed methods to speed up this process [14] [13] [12].

6 Image Coding Using Wavelet Transform

Wavelet transform is an image coding method that has recently received increased attention. It is based on approximation of a function (for example, an image) as a combination of other functions called "wavelets".

In image coding, the wavelet transforms are typically followed by vector quantisations, which is another method for data compression [18].

The architecture of this type of system is presented in Figure 4. Where DWT and IDWT stand for discrete wavelet transform, inverse discrete wavelet transform, respectively [16].

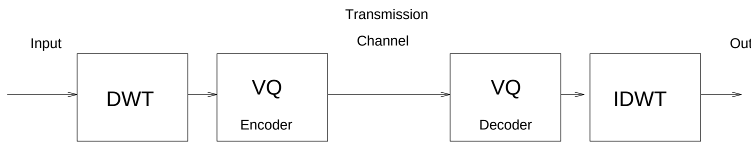

Figure 4: Image Compression System based on Wavelet Transform

The original image is processed using the discrete wavelet transform. Then the resulting coefficients are encoded by means of vector quantisation.

6.1 Wavelet Transform

Wavelet transform represents any arbitrary function f as a superposition of wavelets. Any such superposition decomposes f into different scale levels, where each level is then further decomposed with a resolution adapted to the level [17].

The forward and inverse wavelet transforms can each be efficiently implemented in O(n) time by a pair of appropriately designed quadrature mirror filters (QMF) [15].

Each QMF pair consists of a lowpass filter (H) and a highpass filter (G) which split a signal's bandwidth in half. The impulse response of H and G are mirror images, and are related by:

$$g_n = (-1)^{1-n} h_{1-n} \qquad (6-1)$$

The filter pair of forward and inverse transform QMFs are $(\tilde{H}, \tilde{G})$ and (H,G) respectively.

An example of image decomposition is shown in Figure 5 [16].

In Figure 5 [16], $f_{LL}(x, y)$ is the sub-image of average signal of the original image f(x,y). The other three, f_{LH}, f_{HL}, and f_{HH} are detail signals which are directionally sensitive. They emphasis the horizontal image features, the vertical features, and the diagonal features, respectively.

This transform can be recursively performed on the average signal. The number of transformations performed depends on the desired compression rate and the size of the original image.

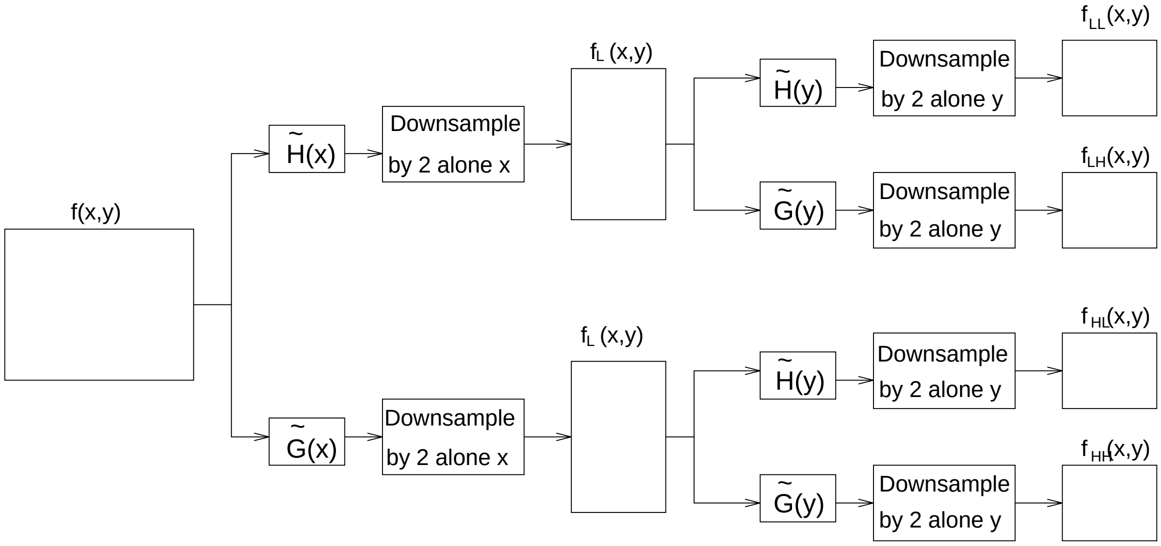

Figure 5: Block diagram of the 2-D forward wavelet transform

After the forward wavelet transform is completed, we get a matrix of coefficients that comprise the average signal and the detail signal of each scale.

The image reconstruction process is an inverse wavelet transform which use (H,G) filter on each scale.

6.2 Vector Quantisation

The wavelet transform itself does not compress the bandwidth requirement of the image. It only produces a set of wavelet coefficients. Compression is achieved by quantising and encoding these coefficients.

The principle of vector quantisation is encoding a sequence of samples (vector) rather than encoding each sample individually [17].

Encoding is performed by approximating the sequence to be coded by a vector belonging to a catalogue of shapes called a codebook.

Each of the vectors in the codebook is indexed. At the encoding stage, the index of the vector in the codebook most closely describing the sample set to be encoded is selected to represent this set. To reconstruct the sample set, the decoder must have the same codebook as the coder.

6.3 Performance

Comparing wavelet compression with Baseline JPEG, at compression rate less than 25:1, JPEG is better than the simple wavelet coder. At compression rates above 30:1, JPEG performance rapidly deteriorates, while the wavelet coder degrades gracefully well beyond a rate of 100:1 [15].

7 Conclusion

We have presented a survey of video compression techniques in this paper. As standardised techniques, JPEG and MPEG present well defined user interface. Fractal and wavelet compression show their potential prospect in future video network applications.

References

[1] Wallace, K. The JPEG Still Picture Compression Standard. In *Communication of the ACM*, Vol. 34, No. 4 (Apr. 1991), pp. 31-44.

[2] Cockroft, G. and Hourvitz, L. Nextstep: putting JPEG to Multiple Uses. In *Communication of the ACM*, Vol. 34, No. 4 (Apr. 1991), pp. 45.

[3] Held and Gilbert. *Data compression: echniques and applications: hardware and software considerations.* Chichester, West Sussex, England; New York: Wiley, 1991.

[4] Gall, Didier L. MPEG: A Video Compression Standard for Multimedia Application. In *Communication of the ACM*, Vol. 34, No. 4 (Apr. 1991), pp. 47-58.

[5] Liou, M. Overview of the px64 kbit/s Video Coding Standard. *Communication of the ACM*, Vol. 34, No. 4 (Apr. 1991), pp. 67-81.

[6] Ang, P., Ruetz, P. and Auld, D. Video Compression makes big gains. In *IEEE Spectrum*, Vol. 28, No. 10 (Oct. 1991), pp. 16-19.

[7] Hurd, P. , Gustavus, A. , Barnsly, F. Fractal Video Compression. In *IEEE Computer Society International Conference*, Feb. 1992, pp. 41-42.

[8] Jacquin, E. Image Coding Based on a Fractal Theory of Iterated Contractive Image Transformations. In *IEEE Transaction on Image Processing*, Vol. 1. No. 1 (Jan. 1992), pp. 18-30.

[9] Anson, F. Fractal Image Compression. In *Byte*, Vol. 18, No. 10 (Oct. 1993), pp. 195-202.

[10] Barnsley, F. and Sloan, D. A Better Way to Compress Image. In *Byte*, Vol. 13, No. 1 (Jan. 1988), pp. 215-223.

[11] Cappellini, V. *Data Compression and Error Control Techniques with Applications.* Academic Press, 1988.

[12] Barthel, Kai Uwe, Voye, Thomas and Noll, Peter. Improved Fractal Image Coding. Technical report 94-01, Technical University of Berlin, Germany, Jan. 1994.

[13] Bani-Eqbal, B. Speeding up Fractal Image Compression. Technical report 94-09, University of Manchester, UK, Sept. 1994.

[14] Saupe, D. Accelerating Fractal Image Compression by Multi-Dimensional Nearest Neighbor Search. In *Proceedings DCC'95 Data Compression Conference*, Match 1995.

[15] Hilton, L., Jawerth, D. and Sengupta, A. Compressing Still and Moving Images With Wavelets. In *Multimedia Systems*, Vol. 2, No.3, (Aug. 1993), pp. 18-30.

[16] Shnaider, M. , Paplinski, P. Wavelet Transform in Image Coding. Technical Report 94-11, Monash University, Australia, Nov. 1994.

[17] Antonini, M. , Barlaud, M. Image Coding Using Wavelet Transform. In *IEEE Transaction on Image Processing*, Vol. 1. No. 2 (Apr. 1992), pp. 205-220.

[18] Nasrabadi, M. Image Coding Using Vector Quantisation: A Review. In *IEEE Transaction on Communication*, Vol. 36. No.8 (Aug. 1988), pp. 957-971.