Three-Dimensional DCT Video Compression Technique Based on Adaptive Quantizers

Raymond Westwater and Borko Furht
Multimedia Laboratory, Florida Atlantic University
Boca Raton, Florida 33431

Abstract

In this paper, we describe a real-time video compression algorithm, based on Three-Dimensional Discrete Cosine Transform. The algorithm, called XYZ video compression, takes advantage of statistical behavior of video data both in spatial and temporal domains. We develop an algorithm for determining optimal quantizers for the three-dimensional DCT. We also add human visual factors into the optimal quantizers. The obtained results show a significant superiority of the proposed algorithm compared to the MPEG standard in term of the quality of the video, compression ratios, and its real-time implementation.

1. Introduction

Considerable research has been devoted to compression of full-motion video. Symmetric multimedia applications, such as videoconferencing and video telephony, require that the video is captured and compressed in real time on the source workstation, transmitted over the network, and then decompressed and played back in real time on the receiving workstation. In asymmetric applications, such as video-on-demand and interactive TV, the video is stored in compressed form on the server, and then transmitted, decompressed, and played back in real time at the receiving stations. In interactive multimedia applications, video is typically stored in compressed form on a CD-ROM, and decompressed and played back in real time.

An ideal video compression technique should satisfy the following requirements:

- Produce a relatively high compression ratio without visible artifacts,
- Provide a real-time compression with inexpensive hardware assistance,
- Provide a real-time decompession for playback using inexpensive hardware or with a software-only solution, and
- Can degrade easily under network overload or on a slow computer platform.

Two video compression standards have recently been developed and approved: H.261/H.263 for video communications [Ste94, Fur95], and MPEG for motion intensive applications [LeG91, Ste94, Fur95]. However, both standards experience a number of drawbacks. For example, low resolution and low quality of decompressed video makes H.261/H.263 standard not applicable in interactive multimedia applications, while complex and computationally expensive encoder makes MPEG not suitable for real-time applications, such as videoconferencing.

The XYZ video compression technique is based on Three-Dimensional DCT (3D DCT). Three-dimensional DCT was originally suggested for compression about twenty years ago [RPR77, NA77], however at that time the computational complexity of the algorithm was to high, it required large buffer of expensive memory, and was not as effective as motion estimation. We have resurrected the 3D DCT based video compression algorithm by developing several enhancements to the original algorithm. These enhancements made the algorithm feasible for real-time compression in applications such as video-on-demand, interactive multimedia, and videoconferencing. The baseline XYZ algorithm was described in [WF95a]. In this paper, we

0-8186-7614-0/96 $5.00 © 1996 IEEE

present an XYZ algorithm based on adaptive quantization, which gives high compression ratios, while maintaining high quality of decompressed video regardless of effects present in a video sequence.

2. Baseline XYZ Video Compression Algorithm

The XYZ motion video compression algorithm relies on a different principle for compression of temporal information than do the MPEG and H.261/H.263 standards. While the MPEG and H.261/H.263 strategies look for motion vectors to represent a frame being compressed, the XYZ strategy more closely resembles the technique adopted by both MPEG and JPEG for intra-frame compression.

A continuous tone image can be represented as a two-dimensional array of pixel values in the spatial domain. The Forward Discrete Cosine Transform (FDCT) converts the two-dimensional image from spatial to frequency domain. In spatial representation the energy distribution of pixels is uniform, while in the frequency domain the energy is concentrated into few low-frequency coefficients.

Pixels in full-motion video are also correlated in the temporal domain, and the FDCT will concentrate the energy of pixels in the temporal domain just as it does in the spatial domain. The XYZ video compression is based on this property.

The XYZ video compression algorithm takes a full-motion digital video stream and divides it into groups of 8 frames. Each group of 8 frames is considered as a three-dimensional image, where X and Y are spatial components, and Z is the temporal component. Each frame in the image is divided into 8x8 blocks (like JPEG), forming 8x8x8 cubes, as illustrated in Figure 1. Each 8x8x8 cube is then independently encoded using the three blocks of the XYZ video encoder: 3D DCT, Quantizer, and Entropy encoder [WF95a]. The block diagram of the XYZ codec is shown in Figure 2.

The original unsigned pixel sample values, typically in the range [0,255] are first shifted to signed integers, say in the range [-128,127]. Then each 8x8x8 cube of 512 pixels is transformed into the frequency domain using the Forward 3D DCT:

$$F(u,v,w) = C(u)C(v)C(w) * \sum_{x=0}^{7}\sum_{y=0}^{7}\sum_{z=0}^{7} f(x,y,z) * \frac{\cos((2x+1)u\pi)}{16}\frac{\cos((2y+1)v\pi)}{16}\frac{\cos((2z+1)w\pi)}{16}$$

where:

x,y,z are index pixels in pixel space,
f(x,y,z) is the value of a pixel in pixel space,
u,v,w are index pixels in DCT space,
F(u,v,w) is a transformed pixel value in DCT space, and

$$C(i) = \frac{1}{\sqrt{2}} \quad \text{for } i=0 \qquad C(i) = 1 \quad \text{for } i>0$$

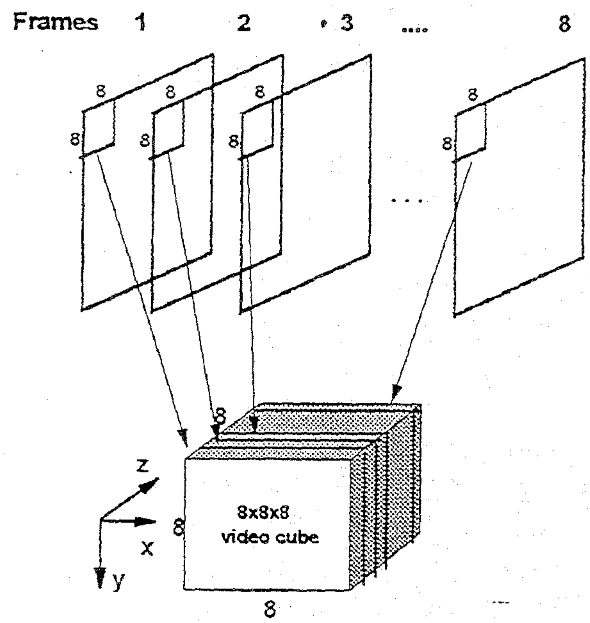

Figure 1. Forming 8x8x8 video cube for XYZ compression.

The transformed 512-point discrete signal is a function in three dimensions, and contains both spatial and temporal information. Most of the energy is contained in few low-frequency coefficients, while the majority of the high-frequency coefficients have zero or near-zero values.

In the next step, all 512 DCT coefficients are quantized using a 512-element quantization table. Quantization introduces minimum error while increasing the number of zero-value coefficients. Quantization may also be used to discard visual information to which the human eye is not sensitive. Quantizer tables may be predefined, or adaptive quantizers may be developed and transmitted with the compressed data.

Quantization is performed according to the following equation:

$$F_q(u,v,w) = \left\lfloor \frac{F(u,v,w)}{Q(u,v,w)} \right\rfloor$$

where:
F(u,v,w) are the elements before the quantization,
F_q(u,v,w) are the quantized elements, and
Q(u,v,w) are the elements from the quantization table.

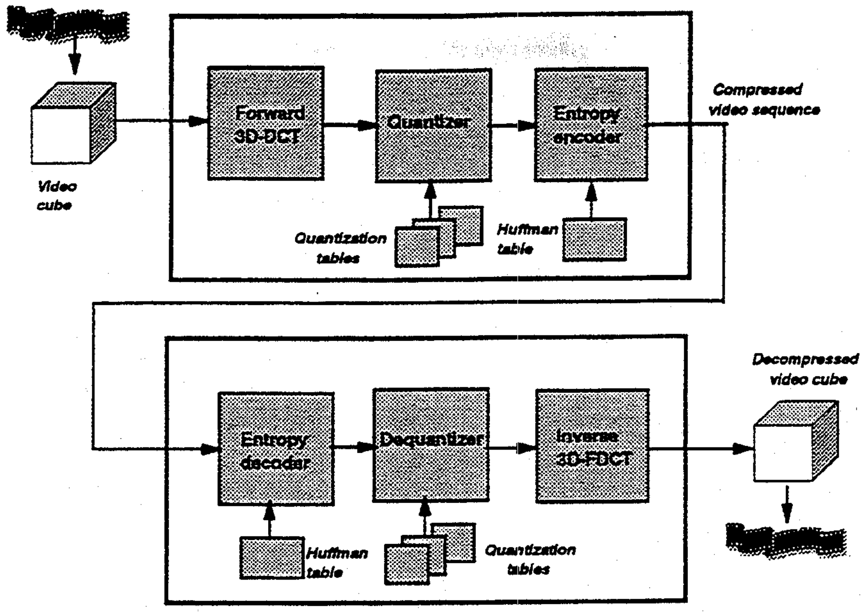

Figure 2. Block diagram of the XYZ codec.

Each quantizer Q(u,v,w) is in the range [1,1024]. The result of the quantization operation is a collection of smaller-valued coefficients, a large number of which are 0. These coefficients are then converted into a compact binary sequence using an entropy coder (in this case, a Huffman coder).

The entropy coding operation starts with reordering the coefficients in descending order of expected value. This sequence has the benefit of collecting sequentially the largest number of zero-valued coefficients. The run-lengths of zero coefficients is computed, and the alphabet of symbols to be encoded becomes the run-length of zeros appended to the length of the non-zero coefficient. This binary sequence represents the compressed 8x8x8 block.

In XYZ decoding, the steps from the encoding process are inverted and implemented in reverse order, as shown in Figure 2.

3. The XYZ Compression Using Adaptive Quantizers

In this section, we present the method for determining the optimal quantizers based upon the statistics of actual data. Statistical measures of the frames to be compressed are calculated first, and then based on these statistical measures and the specified maximum NRMS error, optimal quantizers are determined.

Continuous-tone pictures have traditionally been modeled as a stochastic sequence of random variables. An autoregressive model can be developed by using a casual minimum variance representation for the stochastic sequence. The value of pixel x_n can be expressed as the sum of a causal prediction x_{pn} and the error term ε_n:

$$x_n = x_{n-1} + \varepsilon_n$$

A first-order autoregression model is used to minimize the error distribution e_n, where r is the one-step correlation coefficient:

$$x_{pn} = \rho * x_{n-1} + \varepsilon_n$$

The variance s^2 of e_n is calculated from the assumption that its expectation is 0:

$$\sigma^2 = E(\varepsilon_n^2)$$

and s^2 is related to the variance of the pixel distribution s_x^2:

$$\sigma^2 = \sigma_x^2 * (1 - \rho^2)$$

Lloyd and Max have developed the optimal quantizers by minimizing the mean square error of the introduced quantizing noise [Llo82, Max60]. In this case, the optimal quantizers are determined using the variance of the stochastic variables in the transform space to predict the error introduced in pixel space.

The problem of adaptive quantization requires prediction of the error caused in pixel space by the introduction of error in DCT space. This problem is addressed by recalling that the DCT (and all unitary transformations) are distance preserving (the norm of the sum/difference of two vectors is invariant, Parseval's relation):

$$\sum_{x=0,n-1}\left(s_1[x]-s_2[x]\right)^2 = \sum_{u=0,n-1}\left(S_1[u]-S_2[u]\right)^2$$

where:
s_1, s_2 are expressions of pixel values in pixel space,
S_1, S_2 are expressions of pixel values in DCT space, and
the DCT is a unitary transform (i.e., $DCT^{-1} = DCT^T$).

The Mean Square Error (MSE) is defined as:

$$MSE = \sum_{x=0,n-1}\left(s[x]-s_q[x]\right)^2$$

where:
s is the pixel value in pixel space, and
s_q is the pixel value in pixel space after quantization.

Thus Mean Square Error is invariant under the DCT transformation. We define the invariant measure of error Normalized Root Mean Square Error (NRMSE) as:

$$NRMSE = \frac{\sqrt{\frac{1}{n}\sum_{x=0,n-1}\left(s[x]-s_q[x]\right)^2}}{\mu}$$

where μ is the mean pixel value (in pixel space).

The foundation can now been laid for the definition of a criterion for measuring quantization error that is invariant under the DCT. Quantization error is the term used to describe the discrepancy introduced when a component of the DCT is recovered from its quantized value. Quantization is defined as the rounded quotient of the DCT component and a quantizing factor:

$$Q(x) = \left\lceil \frac{x}{q} + .5 \right\rceil$$

where:
x is the DCT component to be quantized,
q is the quantizing factor, and
Q(x) is the quantized value of x.

Quantizing error is defined as:

$$x' = q * Q(x)$$
$$E_q = x - x'$$

where x' is the dequantized value of x.

E_q has a distribution whose mean is 0 if the distribution of x is even (we assume it is Gaussian), and q is odd (if q is even and large, we may assume the mean of E_q is "close" to 0).

The optimal quantizers are a function of the variance of the stochastic variables in the transform space. In one-dimensional space, the transform variances are calculated as:

$$\sigma^2_{uk} = \sigma^2_{ux} * \left[\mathbf{CAC}^T\right](k,k)$$

where:
σ^2_{uk} and σ^2_{xk} are variances of the k^{th} basis in DCT and pixel space, respectively,
C is the DCT transform,
A is the autocorrelation matrix, and thus a function of ρ, and
$\mathbf{CAC}^T(k,k)$ is the k^{th} diagonal of the product $\mathbf{CAC}^T$.

Figure 3 shows the variance of DCT transform variables for different correlation coefficients.

The problem of adaptive quantization requires prediction of the error caused in pixel space by the introduction of error in DCT space. This problem is addressed by recalling that the DCT (and all orthonormal transformations) are energy preserving (the sum of the squares of the basis elements is invariant):

$$\sum_{x=0,n-1} s[x]^2 = \sum_{u=0,n-1} S[u]^2$$

The variance is preserved under the energy preserving property of the DCT. Since the mean of all AC coefficients of the DCT is already 0, only the DC value has a difficult variance to calculate.

The approach taken to solve this problem in XYZ compression is to average all pixels in the sequence of frames to be compressed. This average is subtracted from each of the pixels in the frames, returning a new pixel distribution whose mean is 0.

Let s_{ijk} represent the pixels to be compressed. The average of all pixels, μ is calculated by:

$$\mu = \frac{\sum_{i=0,N_p}\sum_{j=0,N_s}\sum_{k=0,N_f} s[i][j][k]}{N_p * N_s * N_f}$$

where:
N_p is the number of pixels in a scan line (720 for NTSC),
N_s is the number of scan lines (480 for NTSC),
N_f is the number of frames to be compressed (typically 8), and s[i][j][k] is a typical pixel to be compressed.

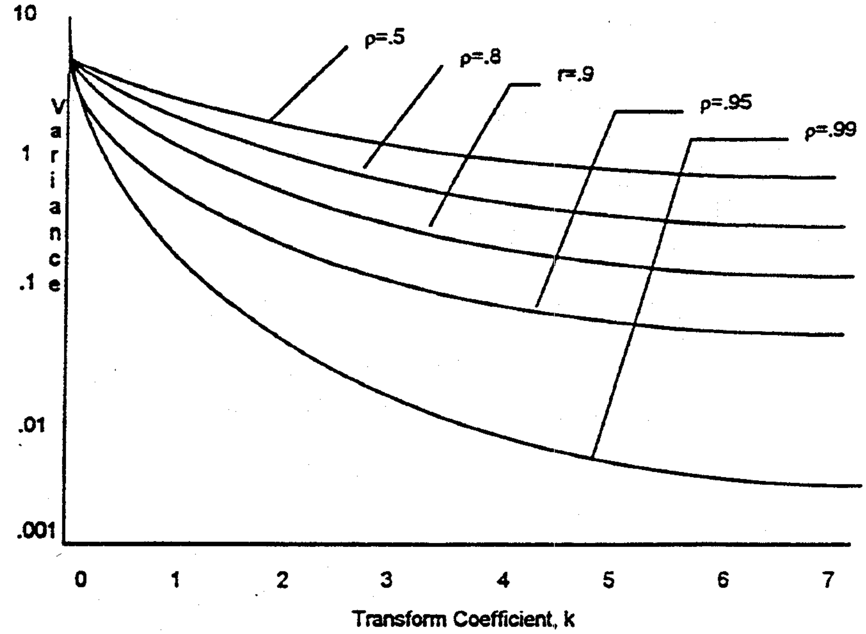

Figure 3 Variance of DCT transform variables foe different correlation coefficients.

Now the transformed pixel distribution s' is calculated from the distribution s as follows:

$$s[i][j][k] = s[i][j][k] - \mu$$

The new distribution s' has a mean of 0. The transformed DC coefficient will have a mean of 0:

$$S[0][0][0] = \frac{\sqrt{2}}{4} \sum_{i=0,7} \sum_{j=0,7} \sum_{k=0,7} s'[i][j][k]$$

Now the variances of all components can be easily calculated in either pixel space or DCT space, as the mean of each variable is now zero in either space. Thus the variance in pixel space for a typical pixel s'[i][j][k] taken from the 8x8x8 cube of pixels to be compressed is calculated across the entire block of frames:

$$\sigma^2_{xijk} = \sum_{x=0,N_p,8} \sum_{y=0,N_s,8} s'^2[i][j][k]$$

where:
σ^2_{xijk} is the variance in pixel space for the pixel s'[i][j][k],
x counts every eighth pixel up the N_p number of pixels in the frame, and
y counts every eighth scan line up the Ns number of scan lines in the frame.

Similarly, variances are calculated in DCT space:

$$\sigma^2_{uxijk} = \sum_{x=0,N_p,8} \sum_{y=0,N_s,8} S'^2[i][j][k]$$

where σ^2_{uijk} is the variance in DCT space for the DCT component S'[i][j][k].

The energy-preserving property of the DCT is used to assert:

$$\sum_{i=0,7} \sum_{j=0,7} \sum_{k=0,7} \sigma^2_{uijk} = \sum_{i=0,7} \sum_{j=0,7} \sum_{k=0,7} \sigma^2_{xijk}$$

4. Generating Quantizer Factors

Quantizer distortion is proportional to the variance of the transformed random variable. However, no closed form expression for optimal bit allocation is known. However, it is reasonable to expect that the number of bits allocated will be proportional to the log of the variance of the variable in transform space. Results similar to this conjecture have been reported in [Sha49, HS63, WJ79]. In this research, we use results given by Shannon [Sha49].

An integer bit allocation algorithm may be used to allocate the expected number of bits needed to represent each DCT component at the desired error level. Once the bit allocation has been completed, the quantizing factors to arrive at the allocated number of bits have be calculated. These quantizers will return the desired error level in pixel space. We begin by initializing all quantizers to their maximum error contribution state:

$$q_{ijk} = \sigma_{ijk}$$

Then we initialize:

$$n_k \leftarrow 0$$
$$d_k \leftarrow \sigma_k^2$$
$$D \leftarrow \sum_k \sigma_k^2$$
$$D_D \leftarrow E^2$$

where:
k ranges through all 512 DCT components,
n_k counts the number of bits allocated to each component,
d_k is the variance of the k^{th} component not yet represented by the bit allocation,
D is the total of d_k,
E is the desired error (input by the agent performing the compression), and D_D is the square of the desired error.

The Normalized Root Mean Square Error at each iteration is calculated as:

$$NRMSE = \frac{\sqrt{\frac{D}{n}}}{\mu}$$

The algorithm is iterated while the Normalized Root Mean Square Error exceeds the desired error, i.e., while:

$$D_D < \frac{\sqrt{\frac{D}{n}}}{\mu}$$

At each step of the iteration, find the index i for the component which will most minimize the unrepresented variance by being allocated one bit:

$$d_i = \max(d_k)$$

Allocate one bit to the i^{th} component, and reduce the variance, and calculate the new quantizer:

$$n_i \leftarrow n_i + 1$$
$$D \leftarrow D - {3 * d_i}/{4}$$
$$d_i \leftarrow {d_i}/{4}$$
$$q_i = {q_i}/{2}$$

When the algorithm converges, the optimal quantizers have been calculated.

5. Adding Human Visual Factors

It may be desirable to modify the optimum quantizers by weighing them with factors determined by evaluating human visual acuity. By analogy, JPEG continuous tone picture quantizers are skewed by factors intending to take advantage of the reduced human visual sensitivity to rapid small changes in the picture. It is well-known that the human eye is insensitive to motion changes of 1/60 of a second, and this fact was used to establish the frame rate of television transmission in the United States.

The process of quantizing coefficients in DCT space relies heavily on the decorrelation property of the DCT. Since the DCT decorrelates the random variables in DCT space, each DCT coefficient may be individually processed. Now a different view of DCT coefficients can be developed.

A model of human visual acuity can be developed based on DCT coefficients. Since DCT coefficients can be individually processed, we can also study their individual visibility. Two-dimensional human visual acuity has been modeled by a modulation transfer function of radial frequency in cycles/degree of visual angle subtended. Figure 4 illustrates typical visual acuity curves. The results of this work have been published in [CR90].

The relative visibility of DCT basis vectors in the spatial dimensions is documented in Table 1.

In order to determine the correct curve for perceived amplitude versus frequency of change over time, test samples were played to test audiences. Each of the three-dimensional DCT components was used to build test cases individually for playback at their natural frequency.

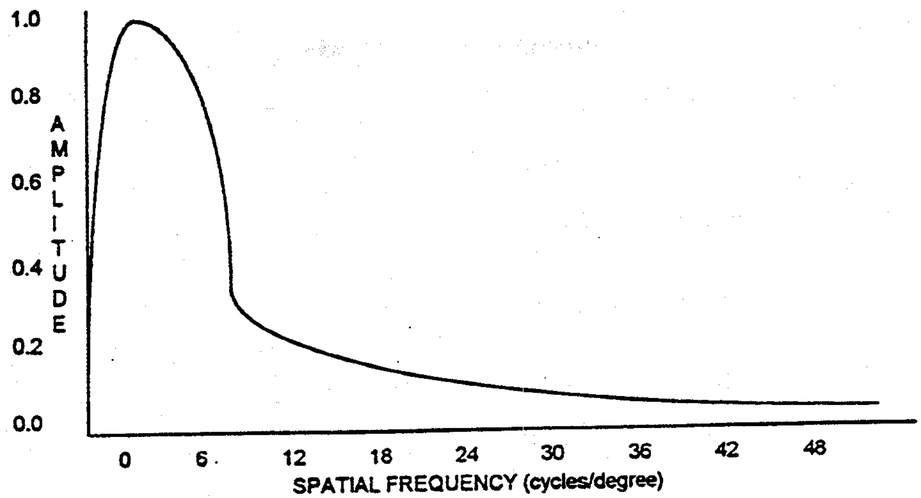

Figure 4. Empirically determined human visual actuity for continuous tone pictures.

.4942	1.000	.7023	.3814	.1856	.0849	.0374	.0160
1.000	.4549	.3085	.1706	.0845	.0392	.0174	.0075
.7023	.3085	.2139	.1244	.0645	.0311	.0142	.0063
.3814	.1706	.1244	.0771	.0425	.0215	.0103	.0047
.1856	.0845	.0645	.0425	.0246	.0133	.0067	.0032
.0849	.0392	.0311	.0215	.0133	.0075	.0040	.0020
.0374	.0174	.0142	.0103	.0067	.0040	.0022	.0011
.0160	.0075	.0063	.0047	.0032	.0020	.0011	.0006

Table 1. Relative visibility of DCT basis vectors in the spatial dimensions.

6. Experimental Results

In this section, we present experimental results obtained by applying (a) baseline (non-adaptive) XYZ algorithm, (b) adaptive algorithm, and (c) adding human visual factors to the algorithm. These algorithms were applied to a number of video clips.

6.1 Sensitivity of the XYZ Algorithm to Various Video Effects

In this section, we present the results obtained when applying the XYZ algorithm to various video clips, which show different video effects, such as camera break, camera panning and zooming, and fast camera movement. Pan and zoom sequences are chosen from the movie 'Total Recall', another pan and camera break sequences are from the movie 'Interview with a Vampire', and a fast camera movement sequence is from the movie 'Interceptor'. The results are summarized in Table 2.

Note from Table 2 that the XYZ algorithm shows very little sensitivity to camera break achieving almost the same compression ratio and NRMSE as in the case with no camera break. However, in the other three cases: camera panning, camera zooming, and fast camera movement, the XYZ algorithm shows a decrease in the compression ratio compared to a normal sequence without these effects, which is an expected result. In the next section, we present the XYZ algorithm using quantization based on human visual factors, which is capable of achieving higher compression ratios in all these cases.

Movie Clip Video Effect	Compression Ratio	Normalized RMSE Error
Dick Tracy Typical Motion	34.5	0.079
Interview with the Vampire Camera Break	32.5	0.087
Interview with the Vampire Camera Panning	26.1	0.085
Total Recall Camera Panning	17.5	0.049
Total Recall Camera Zoom	23.9	0.042
Interceptor Fast Motion	26.0	0.025

Table 2 XYZ algorithm applied to various video effects.

6.2 Adaptive Quantizers

We applied optimal quantizers, described in Section 3, to two critical cases: video clips with camera panning and camera zooming. The obtained results, shown in Table 3, indicate that significant improvements in the compression ratios can be obtained while maintaining the NRMS error close to 8%.

Movie clip/ Video effect	Non-adaptive XYZ Compression ratio/ NRMSE	Adaptive XYZ Compression ratio/ NRMSE
Total Recall Camera panning	17.6/0.049	35.9/0.089
Total Recall Camera zooming	23.9/0.042	53.7/0.065

Table 3. Results obtained by applying adaptive quantizers.

6.3 Quantizers Based on Human Visual Factors

The main drawback of adaptive quantizers lies in the complexity of the corresponding algorithm, which is about four times more complex than a non-adaptive XYZ algorithm [WF96b]. Therefore, in the following experiments we applied the XYZ algorithm based on human visual factors, described in Section 5. We first determined threshold visibility, defined as the magnitude of the DCT component at which artifacts are first visible. The visibility of DCT coefficients was tested at a distance of 6 times the screen size. The results showed a surprising trend - while the threshold visibility was strongly related to the frequency of the DCT component in all three dimensions, relative visibility was virtually independent of time, and only slightly related to spatial frequency. The primary relationship proved to be between intensity of the reference frame and relative visibility coefficient. Thus, a non-linear quantizer was used to model human visual acuity. The first quantizer step was taken to be a function of the threshold visibility (plus 1). Subsequent steps were taken, for these experiments, to be a uniform step size. This size was chosen differently in the experiments.

The following series of experiments were performed:

- **Experiment 1.** First step is threshold visibility+1. Second and following steps are taken to be the minimum of the first step and 20.
- **Experiment 2.** First step is maximum of threshold visibility+1 and 8. Second and following steps are taken to be the minimum of the first step and 20. This experiment is expected to factor in relative insensitivity of the human eye to large areas of constant color.
- **Experiment 3.** First step is maximum of (threshold visibility+1)/2 and 4. Second and following steps are taken to be the minimum of the first step and 10.

The results are summarized in Table 4.

VIDEO CLIP	Experiment #1 Compression Ratio / NRMSE	Experiment #2 Compression Ratio / NRMSE	Experiment #3 Compression Ratio / NRMSE
Susie	64.7 / 0.053	109.5 / 0.054	71.0 / 0.049
Cheerleaders	33.1 / 0.102	40.7 / 0.102	27.1 / 0.086
Carousel	38.1 / 0.209	47.6 / 0.209	29.3 / 0.175
Dick Tracy	59.4 / 0.131	88.5 / 0.131	56.5 / 0.112
Total Recall	50.1 / 0.078	69.8 / 0.079	43.9 / 0.066
Vampire /break	57.5 / 0.150	83.8 / 0.150	53.0 / 0.125
Vampire / Pan	49.8 / 0.151	69.7 / 0.152	44.8 / 0.128
Interceptor	54.9 / 0.047	74.9 / 0.047	47.2 / 0.039

Table 4. Results of the XYZ video compression using Human Visual Factors-based based quantization.

The accompanying Figures 5 and 6 show frame 3 (the frame most susceptible to artifacts) of sequences "Susie" and "Cheerleaders", respectively.

The 5-second clip of 'Susie' was compressed in about one minute on the MasPar computer, suggesting it may be compressible close to real-time on the TMS32080 [WF96b]. The compression ratios of 64.7, 109.5, and 71.0 include YUV sub-sampling of RGB data (a 2:1 factor). A videotape of the returned data showed virtually no artifacts at about 6X the viewing distance.

This 5-second clip of 'Cheerleaders' was also compressed in about one minute on the MasPar. The compression ratios attained for the three experiments were 33.1, 40.7, and 27.1. A videotape of the returned data has been mistaken for the uncompressed clip at about 6X the viewing distance.

7. Conclusion

XYZ video compression compares favorably with other compression algorithms. Compression ratios exceed those of other algorithms, and compression times are comparable to other algorithms capable of high compression ratios. The XYZ encoder complexity is significantly lower than the complexity of the H.261/H/263 and MPEG algorithms (2.5 to 5 times), due to the fact that no motion estimation is necessary. On the other hand, the XYZ decoder complexity is about 2 times higher compared to these two algorithms. Table 5 summarizes the comparison of XYZ compression with other popular video compression algorithms [WF96b].

The proposed XYZ algorithm is a real-time algorithm. Notably, compression takes place in real-time and does not require off-line processing. The expected time to perform this compression compares with the decompression time of other algorithms featuring lower compression performance. Thus the algorithm enables high-quality video teleconferencing.

ALGORITHM	Average Compression Ratio	Relative Encoder Complexity	Relative Decoder Complexity
XYZ	75:1	2	2
H.261/H.263	50:1	5	1
MPEG	30:1	10	1
Wavelet	20:1	1	1
MJPEG	10:1	1	1

Table 5. Rough comparison of popular video compression algorithms.

The algorithm takes advantage of Human Visual System features to generate highest-quality playback. Compression ratios, superior to those of MPEG, are reached with equivalent visual distortion.

The compression ratios for the XYZ algorithm is extremely high. Compression ratios are about 2-3 times those of MPEG at the same bit rate. Compression ratios of 30:1 show no apparent artifacts, and ratios of up to 100:1 show no artifacts at normal viewing distances. Transmission times are correspondingly reduced, and the algorithm may enable wide-area video transmission, interactive television, videoconferencing, and other video-on-demand applications.

Hardware and software implementations issues of the XYZ compression technique have been explored in [WF96b], and substantial progress toward an inexpensive, real-time 3D compression engine has been made. The computational complexity of the 3D DCT has been reduced by developing a fast 3D DCT algorithm.

References

[CR90] B. Chitpraset and K. Rao, "Human Visual Weighed Progressive Image Transmission", *IEEE Transactions on Communications*, Vol. 38, 1990.

[Fur95] B. Furht, "A Survey of Multimedia Compression Techniques and Standards. Part II: Video Compression", *Journal of Real-Time Imaging*, Vol. 1, No. 5, November 1995, pp. 319-338.

[HS63] J. Huang and P. Schultheiss, "Block Quantization of Correlated Gaussian Random Variables", *IEEE Transactions on Communications Systems*, Vol. 11, 1963, pp. 289-296.

[LeG91] D. LeGall, "MPEG: A Video Compression Standard for Multimedia Applications", *Communications of the ACM*, Vol. 34, No. 4, April 1991, pp. 45-68.

[Llo82] S.P. Lloyd, "Least Squares Quantization in PCM", *IEEE Transactions on Information Theory*, Vol. 28, No. 2, March 1982, pp. 129-137.

[Max60] J. Max, "Quantizing for Minimum Distortion", *Transactions of IRE*, Vol. 6, March 1960, pp. 7-12.

[NA77] T. Natarajan and N. Ahmed, "On Interframe Transform Coding", *IEEE Transactions on Communications,* Vol. 25, No. 11, November 1977, pp. 1323-1329.

[PM93] W.B. Pennebaker and J.L. Mitchell, "JPEG Still Image Data Compression Standard", Van Nostrand Reinhold, 1993.

[RPR77] J. Roes, W. Pratt, and G. Robinson, "Interframe Cosine Transform Image Coding", *IEEE Transactions on Communications*, Vol. 25, No. 11, November 1977, pp. 1329-1338.

[RY90] K.R. Rao and R. Yip, "Discrete Cosine Transform - Algorithms, Advantages, Applications", *Academic Press*, 1990.

[Sha49] C. Shannon, "The Mathematical Theory of Communication", *University of Illinois Press*, 1949.

[Ste94] R. Steinmetz, "Data Compression in Multimedia Computing - Standards and Systems", Part I and II, *Journal of Multimedia Systems*, Vol. 1, 1994, pp. 166-172, and 187-204.

[WF96a] R. Westwater and B. Furht, "The XYZ Algorithm fro Real-Time Compression of Full-Motion Video", *Journal of Real-Time Imaging*, Vol. 2, No. 1, February 1996, pp. 19-34.

[WF96b] R. Westwater and B. Furht, "Real-Time Video Compression: Techniques and Algorithms", *Kluwer Academic Publishers*, Norwell, MA, 1996.

[WJ79] S. Wang and A. Jain, "Applications of Stochastic Models for Image Data Compression", *Report, Signal and Image Processing Lab*, Department of Electrical and Computer Engineering, UC at Davis, September 1979.

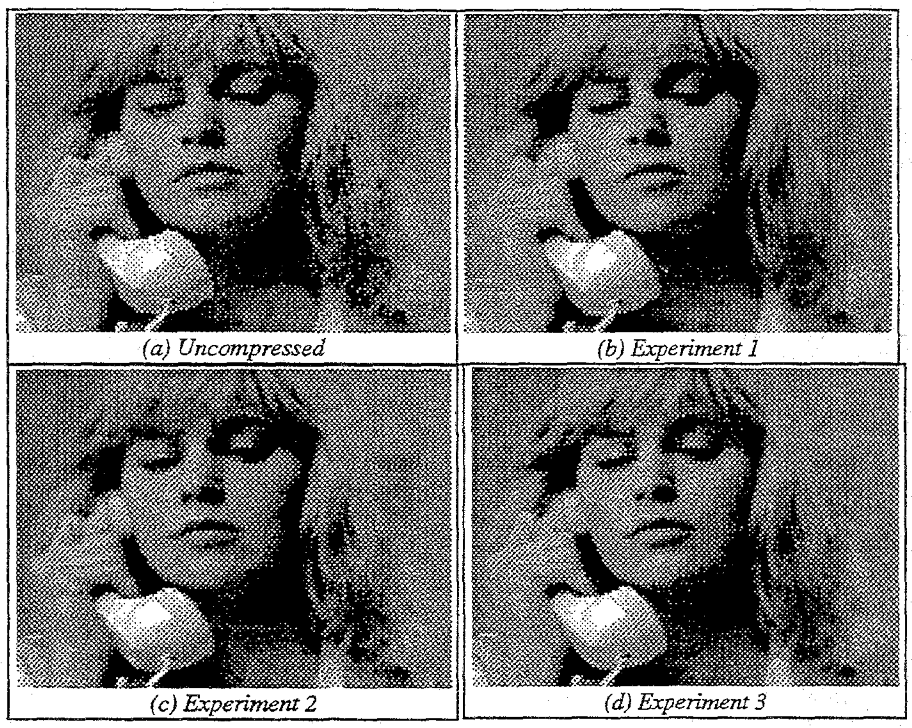

Figure 5. Frame 3 of 'Susie' video clip.
Compression ratios obtained in three experiments are: 64.7, 109.5, and 71.0, respectively.

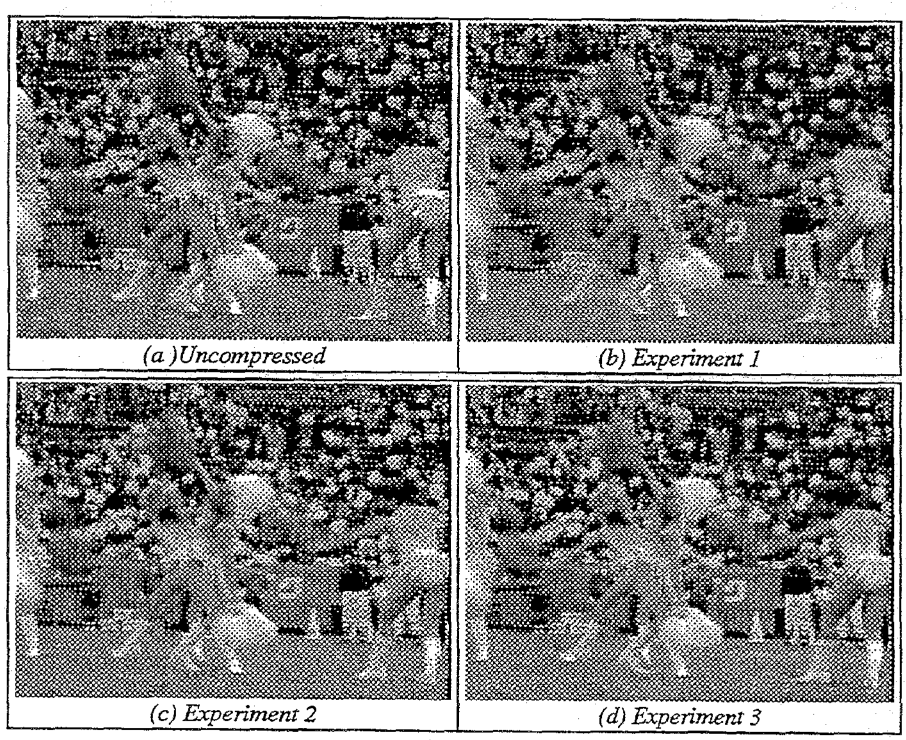

Figure 6. Frame 3 of 'Cheerleaders' video clip.
Compression ratios obtained in three experiments are: 33.1, 40.7, and 27.1, respectively.