IEEE Standard Specifications for the Implementations of 8x8 Inverse Discrete Cosine Transform

Sponsor

CAS Standards Committee
of the
IEEE Circuits and Systems Society

Approved December 6, 1990

IEEE Standards Board

Abstract: IEEE Std 1180-1990, *IEEE Standard Specifications for the Implementations of 8x8 Inverse Discrete Cosine Transform,* specifies the numerical characteristics of the 8x8 inverse discrete cosine transform (IDCT) for use in visual telephony and similar applications where the 8x8 IDCT results are used in a reconstruction loop. The specifications ensure the compatibility between different implementations of the IDCT.
Keywords: discrete cosine transform (DCT), inverse discrete cosine transform (IDCT), hybrid coding, mismatch error, IDCT accuracy.

The Institute of Electrical and Electronics Engineers, Inc.
345 East 47th Street, New York, NY 10017-2394, USA

ISBN 1-55937-098-X

IEEE Standards documents are developed within the Technical Committees of the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Board. Members of the committees serve voluntarily and without compensation. They are not necessarily members of the Institute. The standards developed within IEEE represent a consensus of the broad expertise on the subject within the Institute as well as those activities outside of IEEE that have expressed an interest in participating in the development of the standard.

Use of an IEEE Standard is wholly voluntary. The existence of an IEEE Standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE Standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard. Every IEEE Standard is subjected to review at least every five years for revision or reaffirmation. When a document is more than five years old and has not been reaffirmed, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE Standard.

Comments for revision of IEEE Standards are welcome from any interested party, regardless of membership affiliation with IEEE. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments.

Interpretations: Occasionally questions may arise regarding the meaning of portions of standards as they relate to specific applications. When the need for interpretations is brought to the attention of IEEE, the Institute will initiate action to prepare appropriate responses. Since IEEE Standards represent a consensus of all concerned interests, it is important to ensure that any interpretation has also received the concurrence of a balance of interests. For this reason IEEE and the members of its technical committees are not able to provide an instant response to interpretation requests except in those cases where the matter has previously received formal consideration.

Comments on standards and requests for interpretations should be addressed to:

Secretary, IEEE Standards Board
445 Hoes Lane
P.O. Box 1331
Piscataway, NJ 08855-1331
USA

Foreword

(This Foreword is not a part of the IEEE Std 1180-1990, Standard Specifications for the Implementations of 8x8 Inverse Discrete Cosine Transform.)

The Discrete Cosine Transform (DCT) is considered to be the most effective transform coding technique in practice for image and video compression. Using this technique, blocks of video data are converted into the transform domain for more efficient data compression. An Inverse Discrete Cosine Transform (IDCT) is used to convert the transform-domain data back to the spatial domain. An often used block size is 8x8 pixels since it represents a good compromise between coding efficiency and hardware complexity. Because of its effectiveness, the CCITT H.261-1990, Video Codec for Audiovisual Services at px64 kbit/s, developed by the CCITT SGXV (Specialist Group XV), and the still-image compression standard developed by the ISO JPEG (Joint Photographic Experts Group) all include the use of 8x8 DCT in their algorithms. It is anticipated that different manufacturers may implement the IDCT using different architectures, which have different numerical accuracy. Therefore, for a set of given inputs, the outputs of IDCTs from various manufacturers will likely be slightly different. In hybrid DCT/DPCM (Differential Pulse Code Modulation) video coding systems, such as the one described in CCITT H.261-1990, Video Codec for Audiovisual Services at px64 kbit/s, the IDCT results are used in the reconstruction loop in both the encoder and the decoder to reconstruct pictures. When different IDCTs are used in the encoder and the decoder, the difference between the two IDCT outputs, referred to as the IDCT mismatch error, will accumulate. The mismatch error appears as an additional noise in the reconstructed pictures and causes quality degradation. Due to the nature of error accumulation, even a slight IDCT mismatch may cause severe picture quality degradation. To alleviate this mismatching problem, the CCITT SGXV has gone through a long period of studies and discussions and developed a set of specifications for the required accuracy of 8x8 IDCT. The CCITT H.261-1990, Video Codec for Audiovisual Services at px64 kbit/s, has been finalized. Since a timely and widely accepted IDCT standard is needed for this important application, the CCITT SGXV has requested the IEEE CAS (Circuits and Systems) Standards Committee to sponsor it as an IEEE standard. This standard is a direct result of this request.

In many other applications, such as the ISO JPEG still-image compression, since the IDCT results are not used in a reconstruction loop, the requirements of the IDCT are different from those specified for the CCITT H.261-1990, Video Codec for Audiovisual Services at px64 kbit/s. Furthermore, the desired quality of the reconstructed picture in the still-picture application is much higher than that in the low bit-rate visual telephony application. Therefore, the accuracy of the Forward DCT is also important to insure high quality reconstruction. To fulfill the needs for this type of high-quality still-image compression, the IDCT may have to be specified jointly with the DCT. In recognition of this, in the future we intend to revise this standard to include both CCITT and ISO JPEG specifications if JPEG develops another set of DCT/IDCT specifications. This would make this standard a two-level standard, one level for JPEG-like algorithms and another for CCITT H.261-1990, Video Codec for Audiovisual Services at px64 kbit/s, -like algorithms.

The following members of the IEEE CAS Standards Committee, which served as the working group for this standard, balloted and approved this standard for submission to the IEEE Standards Board:

Ming-Ting Sun, *Chair*

D. Y. Fong
A. M. Gottlieb
D. Graupe
D. Hein
R. T. Hsing
S.M. Kang
A. Ligtenberg
K. Mondal
P. Pirsch
S. A. Rajala
K.R. Rao
M. Snelgrove
H. Sun
T. Thong
K. H. Tzou
M. Vetterli

Special thanks to the following individuals for also contributing to the development of this document:

M. Anderson
M. A. Haque
B. G. Haskell
T. Koga
W. P. Li
M. L. Liou
K. Ohzeki
S. Okubo
G. K. Wallace
K. M. Yang

When the IEEE Standards Board approved this standard on December 6, 1990, it had the following membership:

Marco W. Migliaro, *Chair*
James M. Daly, *Vice Chair*
Andrew G. Salem, *Secretary*

Dennis Bodson
Paul L. Borrill
Fletcher J. Buckley
Allen L. Clapp
Stephen R. Dillon
Donald C. Fleckenstein
Jay Forster*
Thomas L. Hannan
Kenneth D. Hendrix
John W. Horch
Joseph L. Koepfinger*
Irving Kolodny
Michael A. Lawler
Donald J. Loughry
John E. May, Jr.
Lawrence V. McCall
L. Bruce McClung
Donald T. Michael*
Stig Nilsson
Roy T. Oishi
Gary S. Robinson
Terrance R. Whittemore
Donald W. Zipse

*Member Emeritus

Contents

IEEE Standard Specifications for the Implementation of 8x8 Inverse Discrete Cosine Transform

1. Introduction

This standard specifies the numerical characteristics of the 8x8 inverse discrete cosine transform (IDCT) for use in visual telephony and similar applications where the 8x8 IDCT results are used in a reconstruction loop. The specifications ensure the compatibility between different implementations of the IDCT.

1.1 Purpose. In hybrid DCT/DPCM video coding systems, such as the one described in CCITT H.261, Video Codec for Audiovisual Services at px64 kbit/s, the IDCT results are used in the reconstruction loops in both the encoder and the decoder to reconstruct pictures. A simplified encoder-decoder block diagram of a hybrid DCT/DPCM system is shown in Fig 1. In a circumstance where IDCTs of different implementations are used in the encoder and the decoder, the IDCT outputs may be slightly different due to different numerical accuracies. These differences will accumulate in the loop and appear as additional noises in the reconstructed pictures. Without proper care, the quality degradation of the reconstructed pictures will become more and more severe as the coding process goes on.

The purpose of this standard is to solve the quality degradation problem due to the IDCT mismatch in the encoder and the decoder. It is assumed that the coding system undergoes forced intra-frame coding, i.e., refreshment occasionally. This standard specifies the requirements of the 8x8 IDCT so that if an 8x8 IDCT meets the specified standard, the IDCT mismatch will not result in noticeable quality degradation before the system is refreshed.

1.2 Applications. The 8x8 IDCT standard specified here was developed by CCITT SGXV for the CCITT px64 kb/s visual telephony. In order to allow different implementations of IDCTs, this standard adopts a set of very stringent requirements to alleviate the problem due to the possible mismatch between the IDCT in the encoder and the IDCT in the decoder. Therefore, this standard is likely to meet the requirements in many other video coding applications as well.

For video and image applications where the 8x8 IDCT results are not used in a reconstruction loop, such as ISO JPEG still-image compression, the requirements of the IDCT may be relaxed. In such cases, a set of looser requirements may be preferred since devices meeting these looser requirements are likely less expensive. Furthermore, the forward discrete cosine transform (FDCT) may also be specified to guarantee picture quality. Currently, ISO is planning to develop a different set of DCT/IDCT specifications suitable for still image applications. In recognition of this, in the future, we intend to revise this standard to include both the CCITT and JPEG specifications, if JPEG develops its own set of DCT/IDCT specifications. This would make this standard a two-level standard; one level for JPEG-like algorithms, and another for CCITT H.261-1990, Video Codec for Audiovisual Services at px64 kbit/s, -like algorithms.

2. Mismatch Issues in Hybrid DCT/DPCM Coding

In a hybrid two-dimensional DCT/DPCM coding system, the input pictures are divided into blocks of two-dimensional pixels and the differences between blocks in a current frame and blocks in the previous frame are formed. The difference block is then coded and processed by the two-dimensional Forward DCT and quantization. The quantized transformed coefficients are transmitted. The CCITT H.261-1990, Video Codec for Audiovisual Services at px64 kbit/s, is an example of hybrid DCT/DPCM coding with several enhanced features.

In the inter-frame coding mode, the Switch SW is in Position 1, as indicated in Fig 1. In this case, both the encoder and the decoder contain a reconstruction loop. The proper

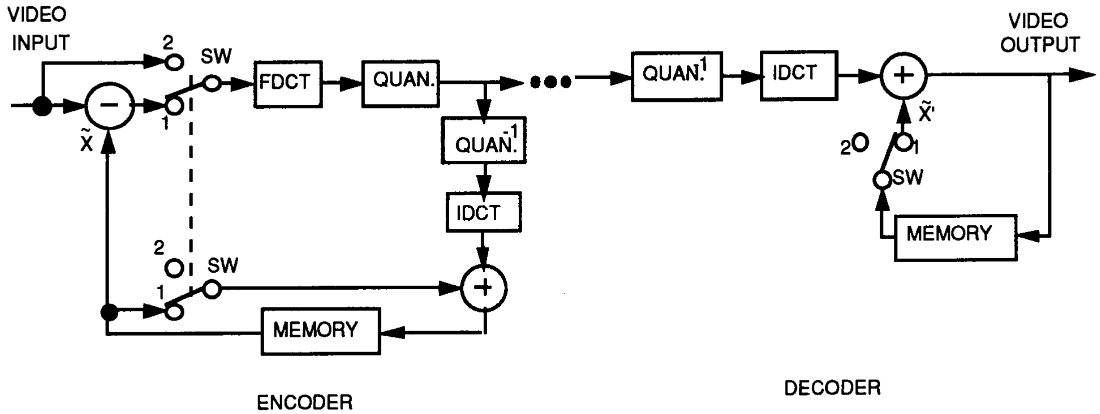

Fig 1
Simplified Encoder-Decoder Block Diagram for a Hybrid DCT/DPCM System

operation of inter-frame coding is based on the assumption that both the encoder and the decoder have an identical copy of the reconstructed previous frame. Therefore, the transmission of the differences between the current frame and the reconstructed previous frame allows the decoder to properly reconstruct the current frame. If the inverse discrete cosine transform (IDCT) in the decoder is identical to that in the encoder, the reconstructed previous frame in the encoder, $\tilde{x}$ is the same as that in the decoder, $\tilde{x}'$. However, it may occur that the IDCTs used in the encoder and decoder are implemented differently, resulting in slightly different outputs. If the mismatch problem is not properly taken care of, it may result in substantial quality degradation in the reconstructed pictures since the mismatch errors will accumulate in the reconstruction loop.

Since the FDCT is outside the reconstruction loop and is not needed in the decoder, it does not cause mismatch problems. Also, manufacturers usually use similar architectures and hardware to implement both the FDCT and IDCT. The resultant accuracy of the FDCT/IDCT pair in an encoder usually is more than sufficient for most applications. Thus, the FDCT is not considered in this standard.

2.1 The Effects of IDCT Mismatches. For the encoder-decoder pair using an identical IDCT, the quality degradation of the reconstructed picture (assuming there is no transmission error) is a result of quantizing the transformed coefficients. Improved picture quality can be expected when the system is operated at higher bit rates.

When the IDCT mismatch exists in the system, the reconstructed picture suffers quantization errors, as well as mismatch errors. The picture quality degradation due to the IDCT mismatch is most distinctive by its cumulative nature. Due to the DPCM loop in the decoder, a mismatch error will be added to the reconstructed previous frame, and the new reconstructed frame is saved in the frame memory. Therefore, the mismatch errors accumulate. As the coding process goes on, the visual effect of mismatch error will become more and more noticeable. For example, if the mismatch always produces a small positive error, +1, at a particular pixel, the accumulated error will become +15 after 15 frames being inter-frame coded.

2.2 Mitigation of Mismatch Error. Due to the reconstruction loops in the system, the mismatch error will keep accumulating when the system is operated in the inter-frame mode. As long as the mismatch exists, the mismatch error will become noticeable when the system stays in the inter-frame mode long enough. To solve the mismatch problem, the two following methods have been suggested:

(1) To use the intra-frame mode periodically to reset the accumulated IDCT mismatch errors.

(2) To eliminate the mismatch by standardizing the architecture and internal accuracy of the IDCT.

The first method results in some degrees of loss in coding effectiveness depending on how often the picture blocks are refreshed. The second method imposes extremely stringent restrictions on the IDCT implementation and may hinder future development of novel implementations. After some extensive debates among experts from various chip manufacturers and telecommunication industries, the second approach was dropped.

2.3 Considerations of Specifying IDCT Mismatch Errors. In the circumstance of no intra-frame coding allowed, even an insignificant IDCT mismatch will result in very noticeable degradation when the system stays in the inter-frame mode long enough. The requirement of no mismatch between the encoder IDCT and decoder IDCT would impose very stringent restrictions on the hardware implementation of the IDCT. Consequently, a standard that allows a small amount of mismatch is more favorable. If a forced intra-frame coding, i.e., refreshment, is used periodically, the mismatch error will be considered acceptable as long as the corresponding picture degradation is not noticeable before the picture is refreshed. The refresh period considered for the CCITT visual telephony is 132 frames.

3. Description of the Recommended Standard for 8x8 IDCT

3.1 Definitions. The 8x8 FDCT is defined as follows:

$$X(u,v)=(1/4)C(u)C(v)\sum_{i=0}^{7}\sum_{j=0}^{7}x(i,j)\cos\left(\frac{(2i+1)u\pi}{16}\right)\cos\left(\frac{(2j+1)v\pi}{16}\right) \quad \text{(Eq 1)}$$

where

$x(i,j), i,j=0,\ldots,7,$

is the pixel value,

$X(u,v), u,v=0,\ldots7,$

is the transformed coefficient,

$C(0)=1/\sqrt{2}$, and $C(u)=C(v)=1$, $u,v=1,\ldots,7$.

The IDCT is defined as :

$$x(i,j)=(1/4)\sum_{u=0}^{7}\sum_{v=0}^{7}C(u)C(v)X(u,v)\cos\left(\frac{(2i+1)u\pi}{16}\right)\cos\left(\frac{(2j+1)v\pi}{16}\right) \quad \text{(Eq 2)}$$

3.2 Procedure for Accuracy Measure-ment. The setup for measuring the accuracy of a proposed IDCT is shown in Fig 2. The procedure is described as follows:

(1) Generate random integer pixel data values in the range -L to +H according to the attached random number generator (in C Language) in the Appendix. Arrange into 8x8 blocks by allocating each set of consecutive 8 numbers in a row. Data sets of 10,000 blocks each should be generated for (L=256, H=255), (L=H=5) and (L=H=300).

(2) For each 8x8 block, perform a separable, orthogonal, matrix multiply FDCT, defined in Eq 1, using at least 64-bit floating point accuracy.

(3) For each 8x8 block, round the 64 resulting transformed coefficients to the nearest integer values. Then clip them to the range –2048 to +2047. This is the 12-bit input data to the inverse transform.

(4) For each 8x8 block of 12-bit data produced by step 3, perform a separable, orthogonal, matrix multiply IDCT, defined in Eq 2, using at least 64-bit floating point accuracy. Round the resulting pixels to the nearest integer, and clip to the range –256 to +255. These blocks of 8x8 pixels are the "reference" IDCT output data.

(5) For each 8x8 block of 12-bit data produced by step 3, use the proposed IDCT chip or an exact-bit simulation thereof to perform an IDCT. Clip the output to the range –256 to +255. These blocks of 8x8 pixels are the "test" IDCT output data.

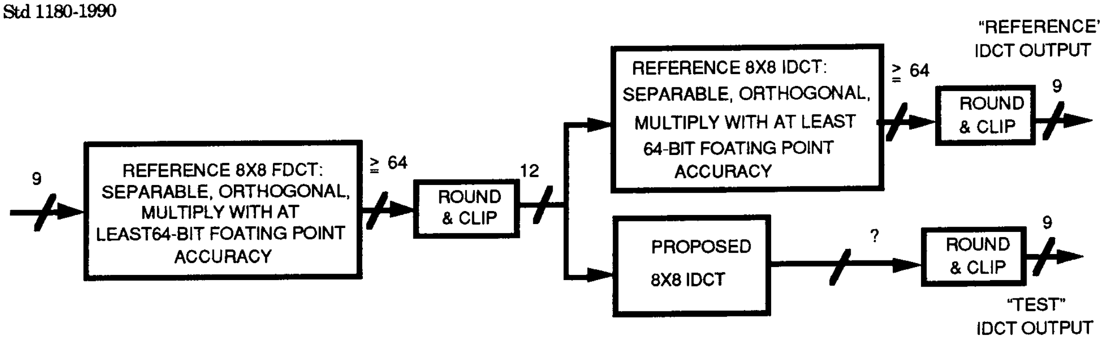

Fig 2
Setup for Measuring the Accuracy of a Proposed 8x8 IDCT

(6) For each of the 64 IDCT output pixels and for each of the 10,000 block data sets generated above, measure the peak, mean, and mean square errors between the "reference" data and the "test" data.

(7) Rerun the measurements using exactly the same data values of step 1, but change the sign on each pixel. (NOTE: The resulted test data sets are in the ranges (−255,256), (−5,5) and (−300,300), respectively.)

The above measured errors shall meet the specifications stated in 3.3.

3.3 Requirements on 8x8 IDCT Accuracy. The above measured errors shall meet the following specification:

(1) For any pixel location, the peak error (*ppe*) shall not exceed 1 in magnitude.

(2) For any pixel location, the mean square error (*pmse*) shall not exceed 0.06.

(3) Overall, the mean square error (*omse*) shall not exceed 0.02.

(4) For any pixel location, the mean error (*pme*) shall not exceed 0.015 in magnitude.

(5) Overall, the mean error (*ome*) shall not exceed 0.0015 in magnitude.

(6) For all-zero input, the proposed IDCT shall generate all-zero output.

The definition of the above error terms is described as follows. Let $x_k(i,j)$ be the "reference" IDCT output, as described in Item (4), 3.2, at pixel location (i,j) in the kth block and $\hat{x}_k(i,j)$ be the "test" IDCT output, as described in Item (5), 3.2, where $i,j = 0,\ldots,7$ and $k=1,\ldots,10{,}000$.

The error, $e_k(i,j)$, between the "test" IDCT output and the "reference" IDCT output is defined as:

$$e_k(i,j) = \hat{x}_k(i,j) - x_k(i,j). \qquad \text{(Eq 3)}$$

The peak error, $ppe(i,j)$, at pixel location (i,j) is defined as the peak value of $e_k(i,j)$, $k=1,\ldots,10{,}000$. The mean square error, $pmse(i,j)$, at pixel location (i,j) is defined as:

$$pmse(i,j) = \frac{\sum_{k=1}^{10000} e_k^2(i,j)}{10000}. \qquad \text{(Eq 4)}$$

The overall mean square error, *omse*, is defined as:

$$omse = \frac{\sum_{i=0}^{7}\sum_{j=0}^{7}\sum_{k=1}^{10000} e_k^2(i,j)}{64\times 10000}. \qquad \text{(Eq 5)}$$

The mean error, $pme(i,j)$, at pixel location (i,j) is defined as:

$$pme(i,j) = \frac{\sum_{k=1}^{10000} e_k(i,j)}{10000}. \qquad \text{(Eq 6)}$$

The overall mean error, *ome*, is defined as:

$$ome = \frac{\sum_{i=0}^{7}\sum_{j=0}^{7}\sum_{k=1}^{10000} e_k(i,j)}{64\times 10000}. \qquad \text{(Eq 7)}$$

4. Mismatch Due to Exact Values (integer+1/2) of the IDCT Output.

4.1 Description of the Problem. With the specification listed in 3.3 and the specified refresh time, another IDCT mismatch may still occur. This mismatch is caused by converting the IDCT output of exact values at (m+1/2) into integer values, where m is an integer. The problem is explained as follows. Assuming that only one IDCT input component $X(u,v)$ is non-zero, the IDCT output $x(i,j)$ is given according to the definition in Eq (2):

$$x(i,j) = 1/4\ B(u,i)\ B(v,j)\ X(u,v)$$

where

$$B(w,z) = C(w) \cos\ [w(2z+1)\ \pi/16].$$

The product of $B(u,i)$ and $B(v,j)$ becomes rational numbers $\pm 1/2$ when u,v = 0 or 4. In this case, the IDCT output is given as follows:

$$x(i,j) = \pm 1/8\ X(u,v).$$

If $X(u{=}0\ ,\ v{=}0\ )$ is equal to (8m+4), then the IDCT output is (m+1/2). This exact value of (m+1/2) is the source of problem. Depending on the internal intermediate computation, the output may be slightly less or greater than the exact value of (m+1/2), resulting in rounded number m or (m+1).

A particular example causing the IDCT input values to be (8m+4) is illustrated as follows. If the quantization levels of an uniform quantizer are chosen as ..., –2.5g, –1.5g, 0, 1.5g, 2.5g, ..., where g is the quantization step size, then the quantizer output levels become ..., –20, –12, 0, 12, 20, ... for g=8. If the component X(0,0) is the only non–zero component, the quantized version always results in IDCT outputs with an exact value of (m+1/2).

Second order combinations such as X(0,0) and X(4,0) components may cause the same problem for (8m+2) or (8m+6) quantization values, and this happens for quantization step sizes 4 or 12, for example, if the same type of uniform quantizer mentioned earlier is used. The mismatch effect, however, was found to be almost unrecognizable.

4.2 Solution to the Mismatch Due to the Exact Values (m+1/2). According to the above discussion, two factors jointly cause this mismatch. The first factor is that, based on the IDCT equation, some combinations of IDCT input values may lead to exact values (m+1/2) as IDCT outputs. The second factor is that, depending on internal intermediate computation, the above IDCT outputs may be slightly greater or less than the exact value of (m+1/2), resulting in rounded number m or (m+1). In order to maintain the strategy of allowing various implementations by specifying tolerance limit against the "reference" IDCT, the second factor causing the mismatch is hard to eliminate.

Consequently, the solution adopted by CCITT is to properly choose the reconstruction levels for the quantizer so that the first factor causing mismatch is circumvented. Since the quantizer is not a part of the IDCT, the following solution to the mismatch problem due to the exact values (m+1/2) shall not be considered as a requirement on the IDCT accuracy. However, it should be understood that two IDCT devices conforming to the tolerance specification described in 3.3 may still be subject to mismatch unless the reconstruction levels of the quantizer are properly chosen.

There are many combinations of the IDCT input components that can result in an exact value (m+1/2). However, the study done by the CCITT indicated that the mismatch effect is almost unrecognizable for those cases caused by higher order combinations. Therefore, the reconstruction levels of the quantizer should be designed to alleviate the mismatch problem for the case of reconstruction from a single IDCT input component. A simple cure to this problem is to avoid even values for the reconstruction levels.

As an example, for the above mentioned uniform quantizer adopted in CCITT H.261–1990, Video Codec for Audiovisual Services at px64 kbit/s, the reconstruction levels (REC) are defined as follows:

$$REC = QUANT * (2*LEVEL+1) \; ; \quad LEVEL>0 \qquad QUANT = \text{'odd''}$$
$$REC = QUANT * (2*LEVEL-1) \; ; \quad LEVEL<0$$

$$REC = QUANT * (2*LEVEL+1)-1 \; ; \quad LEVEL>0 \qquad QUANT = \text{'even''}$$
$$REC = QUANT * (2*LEVEL-1)+1 \; ; \quad LEVEL<0$$

$$REC = 0; \qquad LEVEL = 0,$$

where *QUANT* ranges from 1 to 31 whose value corresponds to half of the step size.

Appendix

(This Appendix is not a part of IEEE Std 1180-1990, IEEE Standard Specification for the Implementations of 8x8 Inverse Discrete Cosine Transform, but is included for information only.)

C-Program Listing for the Random Number Generator

```
                        /*L and H must be long, ie, 32 bits*/
long rand(L,H)
long L,H;
{
static long     randx = 1; /*long is 32 bits*/
static double   z = (double) 0x7fffffff;

long            i,j;
double          x;      /*double is 64 bits*/
                randx = (randx * 1103515245) + 12345;
                i = randx & 0x7ffffffe;         /*keep 30 bits*/
                x = ( (double)i ) / z;          /*range 0 to 0.99999... */
                x *= (L+H+1);                   /*range 0 to < L+H+1 */
                j = x;                          /*truncate to integer */
                return(j-L);                    /*range -L to H*/
}
```