Image Reconstruction After Transform Coding Using Relative Entropy and Maximum Entropy

John S. Bodenschatz

University of Central Florida

This Masters Thesis (Open Access) is brought to you for free and open access by STARS. It has been accepted for inclusion in Retrospective Theses and Dissertations by an authorized administrator of STARS. For more information, please contact STARS@ucf.edu.

STARS Citation

IMAGE RECONSTRUCTION AFTER TRANSFORM CODING
USING RELATIVE ENTROPY AND MAXIMUM ENTROPY

BY
JOHN S. BODENSCATZ, II
B.S.E.E., University of Virginia, 1986

THESIS
Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Engineering
in the Graduate Studies Program
of the College of Engineering
University of Central Florida
Orlando, Florida

Fall Term
1987
ABSTRACT

Presented are two new methods based on entropy for reconstructing images compressed with the Discrete Cosine transform. One method is based upon a sequential implementation of the Minimum Relative Entropy Principle; the other is based upon the Maximum Entropy Principle. These will be compared with each other and with the conventional method employing the Inverse Discrete Cosine transform.

Chapter 2 describes the traditional use of the Discrete Cosine transform for image compression. Chapter 3 explains the theory and implementation of the entropy-based reconstructions. It introduces a fast algorithm for the Maximum Entropy Principle. Chapter 4 compares the numerical performance of the three reconstruction methods. Chapter 5 shows the theoretical convergence limit of the iterative implementation of the Minimum Relative Entropy Principle to equal the limit of the convergence of the Maximum Relative Entropy method.

Preliminary results of this thesis were presented at Southeastcon '87 in Tampa. Final results will be presented at the Annual Meeting of the American Optical Society in Rochester on October 19, 1987.
ACKNOWLEDGEMENTS

I would like to thank my committee for their help and patience during the past year, especially my committee chairman Dr. N. S. Tzannes, upon whose ideas this work is based; and Dr. Belkerdid, who acted as my advisor in Dr. Tzannes' absence. Dr. Myler gave me much assistance as I was using the Gould computer.
# TABLE OF CONTENTS

LIST OF TABLES ........................................ vi
LIST OF FIGURES ....................................... vii
LIST OF SYMBOLS ....................................... viii

Chapter

1. INTRODUCTION .................................... 1

2. BACKGROUND ....................................... 3
   Introduction ...................................... 3
   Statement of the Problem ......................... 3
   Transforms ...................................... 4
   DCT and Data Reduction ......................... 7
      Introduction .................................. 7
      Definition of DCT ............................. 7
      Advantages ................................... 8
      Choosing the Coefficients to be Retained ........ 9
   Usual Method of Reconstruction (IDCT) .......... 9
   Conclusion ..................................... 13

3. ENTROPIC METHODS FOR RECONSTRUCTION OF
   TRANSFORM CODED IMAGES ......................... 14
   Introduction .................................... 14
   Treatment of Images as Pmfs .................... 14
   Relative Entropy ................................ 15
   MEP ............................................. 16
      Introduction .................................. 16
      Theory and Formulas of MEP ................. 17
      Fast MEP for Transform-Coded Images ........ 19
      Zeroing a Function with Known
      Partial Derivatives ......................... 24
   MREP ........................................... 25
      Introduction .................................. 25
      Iterative MREP ................................ 29
   Conclusion ..................................... 31
LIST OF TABLES

1. MREP mse vs. IDCT mse ........................................ 34
2. MEP mse vs. IDCT mse ........................................ 36
3. MEP Speed with a 4x4 Blocksize .......................... 40
4. MEP Speed with an 8x8 Blocksize ....................... 40
LIST OF FIGURES

1. Discrete Cosine Transform Image Compression and Reconstruction Flowchart ........................................ 4
2. Discrete Cosine Transform Basis Vectors; N = 8; p Denotes Coefficient Order .......................... 10
3. Karhunen-Loeve Transform Basis Vectors; N = 8; p = 0.91; p Denotes Coefficient Order .......... 11
4. Energy "Packing" Efficiency \( n_e \) as a Function of Transform Block Size, \( p = 0.91 \). (a) DCT and KLT (0.91), (b) Slant Transform, (c) KLT (0.36), (d) WHT and Haar Transforms, (e) DFT, and (f) DST .... 12
**LIST OF SYMBOLS**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C(m)$</td>
<td>$m$-th one-dimensional transform coefficient</td>
</tr>
<tr>
<td>$C(m,n)$</td>
<td>the value of a transform coefficient in a two-dimensional matrix</td>
</tr>
<tr>
<td>$C_k$</td>
<td>the $k$-th coefficient, same as $C(m,n)$, except the frequency is denoted by the subscript (see $g_k$). For example: $C_0 = C(0,0)$, $C_1 = C(0,1)$, $C_9 = C(1,1)$ for 8x8 matrix</td>
</tr>
<tr>
<td>DC</td>
<td>Discrete Cosine</td>
</tr>
<tr>
<td>DCT</td>
<td>Discrete Cosine Transform</td>
</tr>
<tr>
<td>$\text{DCT}{X}(m,n)$ or $\text{DCT}{X}$</td>
<td>2-dimensional DCT of vector $X$; identical to $C(m,n)$, except the latter does not specify the data-domain vector or the type of transform</td>
</tr>
<tr>
<td>FT</td>
<td>Fourier Transform</td>
</tr>
<tr>
<td>$f_1(x)$</td>
<td>Any one-dimensional function</td>
</tr>
<tr>
<td>$f_2(x,y)$</td>
<td>Any two-dimensional function</td>
</tr>
<tr>
<td>$g(i,m)$</td>
<td>A single value of a transform vector for a one-dimensional transform; $i$ refers to the corresponding pixel; $m$ refers to the coefficient</td>
</tr>
<tr>
<td>$g(i,j,m,n)$</td>
<td>A single value of a transform vector for a two-dimensional transform; $i$ and $j$ refer to the corresponding pixel; $m$ and $n$ refer to the coefficient</td>
</tr>
</tbody>
</table>
Symbol | Description
--- | ---
$g_k(i,j)$ | The k-th DCT vector, same as $g(i,j,m,n)$, except the frequency is denoted by the subscript (see $C_k$). For example:
- $g_0(i,j) = g(i,j,0,0)$
- $g_1(i,j) = g(i,j,0,1)$
- $g_9(i,j) = g(i,j,1,1)$ for 8x8
IDCT | Inverse Discrete Cosine Transform
IDCT{$f_1(x)$}(i) or IDCT{$f_1$}(i) | Value of a pixel resulting from the use of the one-dimensional IDCT upon the vector $f_1(x)$
IDCT{$f_2(x,y)$}(i,j) or IDCT{$f_2$}(i,j) | Value of a data-domain pixel resulting from the use of the two-dimensional IDCT upon the vector $f_2(x,y)$.
i and j | Designate the pixel in a data-domain matrix
K | Number of coefficients retained less one
KLT | Karhunen-Loeve Transform
k | Integer increment used for looping through the frequencies or coefficients of the DCT
MEP | Maximum Entropy Principle; a reconstruction based on the MEP
MREP | Minimum Relative Entropy Principle; a reconstruction based on the MREP
mse | 1. Mean squared error, MEP mse would be mse between the MEP reconstruction and the original image
2. The function that is minimized in the implementation of the MEP algorithm
m and n | Designate the coefficient in a frequency-domain matrix
N | Number of pixels along one edge of a square matrix
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>p</td>
<td>Pixel value in a uniform prior</td>
</tr>
<tr>
<td>p(</td>
<td>An array used as a prior</td>
</tr>
<tr>
<td>pmf</td>
<td>Probability mass function</td>
</tr>
<tr>
<td>q</td>
<td>Integer increment used to designate Lagrangian multipliers</td>
</tr>
<tr>
<td>r x r</td>
<td>Size of matrix of retained coefficients</td>
</tr>
<tr>
<td>RE</td>
<td>Relative Entropy (same as cross-entropy)</td>
</tr>
<tr>
<td>SQDCF</td>
<td>Square DC function, a new mapping function (see Appendix A)</td>
</tr>
<tr>
<td>X(i) or X(i,j)</td>
<td>One- or two-dimensional vector of Lagrangian multipliers</td>
</tr>
<tr>
<td>x(i) or x(i,j)</td>
<td>Pixel value of a one or two-dimensional signal</td>
</tr>
<tr>
<td>x and y</td>
<td>Integers</td>
</tr>
</tbody>
</table>
CHAPTER 1
INTRODUCTION

This thesis will evaluate three methods of reconstruction for images coded with the Discrete Cosine Transform (DCT): the Inverse Discrete Cosine Transform (IDCT), the sequential implementation of the Minimum Relative Entropy Principle (MREP), and the Maximum Entropy Principle (MEP). Several factors will be compared: accuracy of reconstruction, computational speed, and the "characteristics" of the reconstruction. "Characteristics" refers to probing the nature of the reconstructed image, in hopes of answering such questions as "Does the MREP converge with several passes through the coefficients?" and "Is the MREP reconstruction closest to the MEP reconstruction, the IDCT reconstruction, or to the original?" The analysis will be performed on artificially created binary 16x16 images, as well as on a conventional 512x152, 256 grayness-level picture.

Binary 16x16 images will hopefully provide insights that would be harder to see with more complex images. The initial binary value of the pixels is represented by either a one or a two. Reconstruction produces images with pixel values that are real numbers. Two numerical distance measures, mean squared error and relative entropy, will be employed to find the distance between the reconstruction and the original. To obtain a binary
image, the real numbered images are processed by a thresholding routine that sets the pixels to either one or two (one if below threshold, two if above). The analysis will be primarily numerical, because of the large number of reconstructed versions images.

The Gould IP8000 image processor will be used to manipulate and display the pictures. The hard copies will be obtained on a HP LaserJet printer. The reconstructions will be compared via aforesaid numerical measures of distance.

The second chapter, "Background," defines the problem of image compression, showing the advantages of the DCT method. The usual method of reconstruction using the IDCT is discussed.

The third chapter introduces an alternative method of reconstruction using entropic methods. It includes a discussion of the theoretical basis and the algorithm for the sequential implementation of the MREP and the MEP.

The fourth chapter tabulates the numerical results of this thesis. The work with the binary images is covered first; then, the work with the picture of the Golden Gate bridge is covered. The distances of between the reconstructions and the originals are measured to evaluate the reconstruction methods.

The fifth chapter is entitled "Theoretical Analysis of the MREP Convergence." It shows the iterative MREP to converge to the MEP with repeated passes through coefficients; although, this work does not include a proof that the convergence actually occurs.
CHAPTER 2
BACKGROUND

Introduction

This chapter provides background information. It begins by stating the problem of image reconstruction after transform coding. It provides the definition of a transform and enumerates the advantages of using the Discrete Cosine Transform for image compression. The implementation of the conventional reconstruction method, the Inverse Discrete Cosine Transform, is described. The chapter concludes by highlighting the advantages and disadvantages of this method.

Statement of the Problem

The general area of this thesis is reconstruction of images that have been shortened in some way. More specifically, this thesis focuses on images compressed by a transform. The Discrete Cosine Transform was chosen for this work.

The following diagram shows the process.
Figure 1. Discrete Cosine Transform Image Compression and Reconstruction Flowchart.

It is desired that the reconstruction be close to the original image. To achieve this, the three different methods are used in Step 3. The object of this thesis is to find the optimal method.

**Transforms**

Transform coding, a frequency-domain technique, is the representation of a signal in terms of pre-determined basis functions. The signal is mapped onto a corresponding set of coefficients. This section defines a transform and its inverse in terms of the basis vectors. The properties of a basis vector are discussed, and several popular transforms are mentioned (Shore 1984).

This discussion will use a one-dimensional transform for simplicity. The coefficients are obtained by taking the dot
product, the original signal \( f(i) \) and the basis vector \( g_m(i) \). Since the basis vector is two-dimensional, it will be written as \( g(i,m) \). The output transform coefficients are the inner product of the basis vector and the data. This constitutes the forward transform:

\[
C(m) = \sum_{i=0}^{N-1} g(m,i) \cdot f(i) \tag{2.1}
\]

where the original signal \( f(i) \) is \( N \) components long, and \( C(m) \) is defined for \( m = 0, 1, 2, \ldots, N-1 \) (\( N = \text{blocksize} \)).

There is a one-to-one correspondence between the original signal and the transform. The signal can be reconstructed by taking a weighted sum of the basis functions; this function weighting is determined by the corresponding coefficient (Tzannes 1985). The inverse transform follows:

\[
f(i) = \sum_{i=0}^{N-1} g(m,i) \cdot C(m) \tag{2.2}
\]

The previous two equations make up a transform pair.

The transform is defined by its basis vector, which must be orthonormal; each component must be orthogonal with every other component:
\[
N-1 
\sum_{i=0} g(m,i) \cdot g(n,i) = E
\]
(2.3)
for all \( m,n \) when \( m \neq n \), and they must be normal (Tzannes 1985):

\[
1 = \sum_{i=0}^{N-1} g^2(m,i) \quad \text{for all } m
\]
(2.4)

The two-dimensional forward transform is:

\[
C(m,n) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} g(i,j,m,n) \cdot f(i,j)
\]
(2.5)

where:

- \( C(m,n) \) is defined for \( m,n = 0, 1, 2, \ldots, N-1 \)
- \( f(i,j) \) is defined for \( i,j = 0, 1, 2, \ldots, N-1 \)
- \( j = 0, 1, 2, \ldots, N-1 \)
- \( N \times N \) is the blocksize of \( f(i,j) \)

The two-dimensional inverse transform is:

\[
f(i,j) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} g(i,j,m,n) \cdot C(m,n)
\]
(2.6)
There are many types of transforms used in signal processing: Slant, Walsh-Hadamard, Hadamard-Walsh, Harr, Discrete Fourier, Discrete Cosine, and Karhunen-Loeve. The amount of data compression possible is dependent upon the properties of the transform, the statistics of the data, and the quality required in the reconstructed image.

DCT and Data Reduction

Introduction

This section begins by defining the DCT. Then, its advantages for image compression are discussed. Finally, the method for reducing the data by dropping high order coefficients is explained.

Definition of DCT

All transforms can be defined in terms of the basis vectors. The DCT basis vectors will be written as \( g(i,m) \) and \( g(i,j,m,n) \) for the one-dimensional and two-dimensional cases respectively, and the coefficients as \( C(m) \) and \( C(m,n) \). \( C(m) \) corresponds to Clarke's \( C(p) \) with \( m \) equalling \( p \) (Clarke 1984). The two-dimensional transformations in this work are all square, although the results would apply to other shapes. The one- and two-dimensional transforms are given by equations (2.1), (2.2), (2.5), and (2.6) using the following definitions for the basis vectors.
\[ g(m) = \frac{c(m)}{\sqrt{N}} \cdot \cos \left( \frac{2i + 1}{2n} \right) \]  
(2.7)

\[ g(i,j,m,n) = \frac{c(m)}{N} \cdot \cos \left( \frac{2i + 1}{2n} \right) \cos \left( \frac{2j + 1}{2n} \right) \]  
(2.8)

\[ c(1) = 1 \quad 1 = 0 \]
\[ = 2 \quad 1 \not= 0 \]

This is a common definition used by A. K. Jain, F. A. Kamangar and K. R. Rao, etc. (Jain 1979, Kamangar 1982). It satisfies the criteria for orthogonality as given by equations (2.3) and (2.4). There are some variations in use, some of them do not divide the vector by the square root of the size in the reverse transform and divide by the size (not its root) in the forward direction (Ahmed 1974). The reconstruction methods in this paper could be performed with these different versions, or they could be performed with any other transform whose basis vectors are known.

Advantages

There were several reasons that the DCT was chosen for these studies. Its performance with conventional image data having high inter-element correlation is virtually identical to that of the Karhunen-Loeve (KLT). The DCT basis vectors are quite similar to those of the KLT for a data correlation of 0.91. (see figures 2
Energy packing efficiency was measured as the energy in the largest \( N/2 \) coefficients. The DCT had the highest. This efficiency is desirable as it represents the effectiveness of the possible data reduction (see Figure 4) (Clarke 1984). The symmetry of the DCT transform has permitted the development of many fast algorithms for its computation (Haque 1985; Ahmed, Natarajan, and Rao 1974; Karmangar and Rao 1982). However, the KLT is very slow, as the basis vectors need to be recalculated for each image. Thus, the DCT is widely used in image processing (Shore 1984).

**Choosing the Coefficients to be Retained**

There are two main ways to choose the coefficients to be saved during the compression. They can be chosen by magnitude (the \( n \) largest coefficients) or position (the \( n \) lowest frequency coefficients). The latter method requires that information about the position must be sent. This position information takes up a significant amount of transmission data, and choosing by magnitude does not seem to be significantly better than by position (Mailaender 1985). This work will use the method of choosing coefficients by position.

**Usual Method of Reconstruction (IDCT)**

The usual way to reconstruct a transform-code image is to use the corresponding inverse transform, setting the unknown
Figure 2. Discrete Cosine Transform Basis Vectors; $N = 8$; $p$ Denotes Coefficient Order.
Figure 3. Karhunen-Loeve Transform Basis Vectors; N = 8; p = 0.91; p Denotes Coefficient Order.
Figure 4. Energy "Packing" Efficiency $\eta_E$ as a Function of Transform Block Size, $p = 0.91$. (a) DCT and KLT (0.91), (b) Slant Transform, (c) KLT (0.36), (d) WHT and Haar Transforms, (e) DFT, and (f) DST.
coefficients to zero. The formula for the inverse transform was
given by equation (2.2) for the one-dimensional (1D) version and
equation (2.6) for the two-dimensional (2D) version. The basis
vector was given by equation (2.7) for the 1D transform and
equation (2.8) for the 2D transform. The IDCT can be performed
very quickly due to the symmetry of the transform (Haque 1985).

Conclusion

Transform coding is useful for compressing data before
transmission or storage. The DCT is employed in this thesis, as
it is very popular for data reduction (Clarke 1984). Many
coefficients can be dropped without losing much information. The
IDCT is the normal method of image reconstruction after DCT
coding.
CHAPTER 3

ENTROPIC METHODS FOR RECONSTRUCTION
OF TRANSFORM CODED IMAGES

Introduction

This chapter discusses a new method based on probability theory for reconstructing transform-coded images. The second section shows that the functions representing the images can be viewed as pmfs. The third describes Relative Entropy as a measure of distance for pmfs. The next applies the MEP and develops a fast implementation. The fifth illustrates the use of the MREP to satisfy the coefficients sequentially and a variation that makes additional passes through the sequence of coefficients.

Treatment of Images as Pmfs

The image data will be treated as a probability mass function (pmf). There are two main restrictions on pmfs; they must be positive everywhere, and the sum of the element values must be unity. The images used in this work were positive everywhere. The second restriction is not numerically necessary for the algorithms in this work, if the images have the same grayness level.
Relative Entropy

Relative Entropy has many other names: cross-entropy, discrimination information, directed divergence, and Kullback-Leibler number. It is a means of measuring the distance of two probability mass functions (pmfs). If two signals are identical, the RE between them is zero. Relative Entropy between two pmfs \( f(i) \) and \( p(i) \) is defined as:

\[
RE = \int f(x) \log \frac{f(x)}{p(x)} \, dx
\]  

(3.1)

Applying this to discrete signals yields

\[
RE = \sum_{i=0}^{N-1} f(i) \log \frac{f(i)}{p(i)}
\]  

(3.2)

the two-dimensional version is simply:

\[
RE = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} f(i,j) \log \frac{f(i,j)}{p(i,j)}
\]  

(3.3)

This gives a working definition of RE (Shore 1984).

This measure of distance was chosen to be extremized because it satisfies four criteria: uniqueness, invariance, system independence, and subset independence (Shore 1984).
Entropy itself is defined as:

\[ H = - \sum_{n=0}^{N-1} p(n) \log p(n) \]  

(units are in bits if the logarithm base is two)

This is a measure of the information in the realization of a random variable described by the pmf \( p(n) \). Because of the negative sign, maximizing this entropy corresponds to minimizing RE with respect to a uniform prior.

MEP

Introduction

The following subsection discusses the theoretical basis for the reconstruction after transform coding based upon the Maximum Entropy Principle (MEP). The simultaneous equations to implement this reconstruction are given. The third subsection discusses a new (as far as research shows) algorithm that greatly simplifies the calculations to solve these simultaneous equations. This algorithm results in a function to be driven to zero and the value of each partial derivative. This function is written in terms of the unknowns of the simultaneous equations; when it is zeroed, the solution is found. The last subsection gives a brief description of the numerical method used to zero a function when each partial derivative is known. In summary, the second section introduces a set of simultaneous equations; the third uses them to produce a function, and the fourth zeroes the function.
Theory and Formulas of MEP

The MEP technique produces an image that has maximum entropy (equation 3.4) which is equivalent to minimum relative entropy (equation 3.3) with a uniform prior. All the known constraints are simultaneously satisfied. Nothing is assumed about the unknown coefficients, unlike the inverse transform method.

These constraints are the coefficients $C(m,n)$. Each constraint corresponds to an equation that must be satisfied. The equation is given by equation (2.5). It can be rewritten:

$$0 = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} g(i,j,m,n) f(i,j) - C(m,n)$$

(3.5)

where:

- $N \times N$ is the size of the block that was coded;
- pixel locations are $i,j = 0, 1, 2, ..., N-1$
- before data reduction $m,n = 0, 1, 2, ..., N-1$
- after data reduction $m,n = 0, 1, 2, ..., r$
- an $r \times r$ block of coefficients is retained

Equation (3.5) must be satisfied for each known coefficient.

The mathematics of extremization yield a solution:

$$f(i,j) = p \cdot \exp\left( \sum_{m=0}^{r-1} \sum_{n=0}^{r-1} X(m,n) g(i,j,m,n) \right)$$

(3.6)
where:

- $X$ is the Lagrangian multiplier
- $p$ is the value of a pixel in a uniform prior

The Lagrangian multiplier is normally written as a lowercase lambda.

The $X$s are found by the simultaneous solution of:

$$C(m,n) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} f(i,j) C(i,j,m,n) \quad (3.7)$$

for each known $C(m,n)$.

Consider that a low frequency square of coefficients has been retained. The square contains $r\times r$ coefficients (each side is $r$ long). Substitution of equation (3.6) into equation (3.5) yields:

$$C(m,n) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} g(i,j,m,n) \cdot p \cdot \exp\left[ \sum_{mm}^{r-1} \sum_{nn}^{r-1} x(mm,nn) \cdot g(i,j,mm,nn) \right] \quad (3.8)$$

for each known $C(m,n)$.

For the RE to hold, the gray level of the reconstruction must equal the gray level of the prior. This gray level is given by the lowest-order coefficient. Assuming a uniform prior:

$$p = C(0,0)/N \quad (3.9)$$
For a non-uniform prior, this value would be a function $p(i,j)$ in the above equations.

Thus, equations (3.6) and (3.8) are used for image reconstruction after transform coding using MEP.

Fast MEP for Transform-Coded Images

This section introduces a fast method of implementing MEP for transform-coded images; equations (3.6) and (3.8) are solved using a minimum of calculation. This section was written using the DCT; however, the solution is more general. Any transform whose basis vectors are known could be substituted for the DCT. The IDCT, which is used in the fast MEP reconstruction, would obviously be replaced by the inverse of the transform used for compression. The Square Discrete Cosine mapping Function (SQDCF), which is introduced in this subsection, could be replaced by another function defined in terms of the basis vectors of the compression transform (see Appendix A).

Basically, equations (3.6) and (3.8) will be rewritten in terms of the DCT, the IDCT and the SQDCF. Because of their symmetry, many fast algorithms are available for implementation.

The formula for the image, once the Lagrangian multipliers are known, is given by equation (3.6). Note that the exponent $X(m,n)*g(m,n,i,j)$ is an IDCT. If we can replace the equation with:
\[ f(i,j) = p \cdot \exp[\text{IDCT} \{X\} (i,j)] \quad (3.10) \]

if \(X(m,n) = 0\) for \(m,n \geq r\). IDCT \(\{X\} (i,j)\) is the two-dimensional coefficient vector. It is written this way to indicate it is a transform of \(X(m,n)\), and its members are designated by \((i,j)\). Note that the IDCT only needs to be evaluated once per image block.

The method for calculating the Lagrangian multipliers involves similar substitutions. These multipliers come from the simultaneous solution of equation (3.8) for each known \(C(m,n)\). Call the right-hand side of this equation \(F[X](m,n)\), to indicate that it is a function of all of the known Lagrangian multipliers \(X(mn,nn)\) and that it is two-dimensional. It can be considered a mapping, analogous to the DCT; both are written the same manner in this paper. Thus:

\[
F[X](m,n) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} g(i,j,m,n) \cdot p \cdot \exp[\sum_{mm}^{r-1} \sum_{nn}^{r-1} X(mm,nn) \cdot g(i,j,mm,nn)]
\]

\[C(m,n) = F[X](m,n) \quad \text{for each } m,n = 0, 1, 2, \ldots, r-1 \quad (3.12)\]

Consider the standard distance measure \(\text{mse}\):

\[\text{mse} = \sum_{m=0}^{r-1} \sum_{n=0}^{r-1} [C(m,n) - F[X](m,n)]^2 \quad (3.13)\]
If the mse is zero, the two vectors are identical. Thus, it suffices to drive the mse to zero to solve equation (3.8). This corresponds to (Shore 1984):

\[
M \sum_{r=0}^{\infty} a_r \left( \int s_r(x) q^t(x) \, dx - \overline{s}_r \right)^2 \leq \varepsilon^2 \tag{3.14}
\]

where:

- \( s_r \) is the basis vectors
- \( \overline{s}_r \) is the constraints
- \( q^t \) is the unknown pmf

The method employed for finding the zeros of a function requires the value of that function and every first order partial derivative. Take the derivative of mse (equation 3.12) with respect to \( F(X)(m,n) \) and call this derivative \( dmse \):

\[
dmse(m,n) = 2 \sum_{r=0}^{r-1} \sum_{m=0}^{r-1} \left[ C(m,n) - F(X)(m,n) \right] \cdot dF[X](m,n) \tag{3.15}
\]

where \( dF[X](m,n) \) is the derivative of \( F[X](m,n) \).

Take the derivative of \( F[X](m,n) \) (from equation 3.11) with respect to \( X(m,n) \):
\[
\begin{align*}
    dF[X](m,n) &= \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} g(i,j,m,n)^2 \cdot p \cdot \\
    &\quad \times \exp\left[ \sum_{mm} \sum_{nn} X(mm,nn) \cdot g(i,j,mm,nn) \right] \cdot dX(m,n) \\
    &= p \cdot \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} g(i,j,m,n)^2 \cdot \exp\left[ \text{IDCT}(X)(i,j) \right] \cdot dX(m,n)
\end{align*}
\]

for each \(m, n = 0, 1, 2, \ldots, r-1\).

The exponent in equations (3.11) and (3.16) can be replaced with an IDCT, after setting \(X(m,n) = 0\) for \(n\) or \(m > r-1\):

\[
\begin{align*}
    F[X](m,n) &= \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} g(i,j,m,n) \cdot p \cdot \exp[\text{IDCT}(X)(i,j)] \\
    &= p \cdot \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} g(i,j,m,n)^2 \cdot \exp[\text{IDCT}(X)(i,j)] \cdot dX(m,n)
\end{align*}
\]

Take the constant prior outside of the summation:

\[
\begin{align*}
    F[X](m,n) &= p \cdot \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} g(i,j,m,n) \cdot \exp[\text{IDCT}(X)(i,j)] \\
    &= p \cdot \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} g(i,j,m,n)^2 \cdot \exp[\text{IDCT}(X)(i,j)] \cdot dX(m,n)
\end{align*}
\]

The double summation in the equation for \(F[X](m,n)\) is an IDCT. The equation for \(dF[X](m,n)\) can be modified in a similar manner.

A new mapping function SQDCF is defined. It is similar to the DCT. The only difference is that its basis vector is the square of the basis vector of the DCT. The coefficients are given by:
\[ \text{SQDCF}(f)(m,n) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} g(i,j,m,n)^2 \cdot f(i,j) \quad (3.21) \]

Much of the symmetry employed for fast evaluation of the DCT also applies to this function. This SQDCF mapping function will be investigated in Appendix A. The author has not seen the SQDCF in literature. Thus, equations (3.19) and (3.20) become:

\[ F\{X\}(m,n) = p \cdot \text{DCF}[\exp[\text{IDCT}\{X\}]](m,n) \cdot dX(m,n) \quad (3.22) \]

\[ dF\{X\}(m,n) = p \cdot \text{SQDCF}[\exp[\text{IDCT}\{X\}]](m,n) \cdot dX(m,n) \quad (3.23) \]

Note that the \( i \) and \( j \) of \( \text{DCT}\{X\} \) are no longer present, because of the outer mapping function and that nothing needs to be set to zero before taking the DCT or the SQDCF.

The term \( \exp[\text{IDCT}\{X\}] \) is the same in both equations and only needs to be evaluated once per block (not once for every \( m,n \) as it is not a function of \( m \) or \( n \)). The DCT and the SQDCF only need to be performed once per block to find \( F\{X\}(m,n) \) and \( dF\{X\}(m,n) \). Look back at mse and dmse (equations 3.13 and 3.15), the only other calculations per block required are taking the exponent of the \( \text{IDCT}\{X\} \) (\( N^2 \) exponential calculations), the subtraction of \([ C(m,n) - F\{X\}(m,n) ] \) (\( N^2 \) subtractions), the multiplications in the above formula for dmse and multiplying by prior (2\( N^2 \) multiplications) for finding dmse and mse. These are not much overhead. Thus, each step in the iterative solution for the
Lagrangian multipliers takes little more time than calculating the DCT thrice. As \( N \) becomes larger, the time becomes equal to the time required to take the DCT thrice. This calculation is necessary for each step in the convergence of the nonlinear solution. The procedure that solves the simultaneous equations is covered in Appendix A. This fast MEP algorithm could be applied to any transform with a known transform vector.

Zeroing a Function with Known Partial Derivatives

This section gives a quick description of a method to zero an equation when all of the partial derivatives are known. This method is used to calculate the numerical solution to equation (3.13) with the partial derivatives given by equation (3.15). It is a very sophisticated technique that the author wrote without consulting a reference.

It is easiest to view this problem spacially, where the solution is a point in a space where every axis corresponds to one unknown. In this application, the initial guess was zero. The second guess is related to the first by a step. The direction of the step is the direction in which the value of the function is diminishing the fastest. The size of the first step is an arbitrary value. The second step is 1.25 times the first step, as long as the value of the function at the second point is lower than the value of the function at the first. If the second
function value is higher, the stepsize is cut in half and the algorithm backtracks to the first point.

The algorithm can also be view in terms of a man walking to the bottom of a hill. He steps in the downhill direction. If he is lower than he was before, he takes a larger step downhill; if he is higher than he was before, he goes back to his previous position and takes a smaller step downhill.

There is one additional point to make. The above stepsize is only a factor in the true stepsize and is not used directly. The true stepsize also takes into account the slope and height of the hill at the current location.

The algorithm is resilient. Its method will not diverge, as a standard 1D Newton method can. It can become stuck at a local minimum or at an inflection point; however, neither occur during the MEP calculation.

**MREP**

Introduction

The following subsection justifies the use of Minimum Relative Entropy (MREP) and discusses the implementation to reconstruct images coded with the DCT. The third subsection explains the iterative variation that is computationally more complex.

Shore and Johnson originally proposed the MREP as a means of reconstructing a function. The method combines a previous
estimate of the signal with additional information to obtain a second estimate. This additional information (prior knowledge) comes from the coefficients of the DCT. The prior knowledge is given by equations (2.1) and (2.5). Thus, MREP is applied to reconstruction after transform coding.

Previously, N.S. Tzannes and M.S. Tzannes (1986) introduced a new universal method of decoding transform-coded images using the principle of Minimum Relative Entropy (MREP). In this paper, we examine the MREP's iterative convergence properties by applying MREP to image data compressed by the Discrete Cosine Transform (DCT) and running the iterative algorithm until it stabilizes.

The minimization of a two-dimensional function subject to many constraints, and the usual lack of a prior guess makes the use of the regular MREP quite difficult to implement in practice. However, it can be done if one assumes a uniform prior; then, it is identical to the MEP. To alleviate these problems, the following sequential MREP algorithm was suggested in "Reconstruction of Transform-Coded Data by RE Minimization" (Tzannes and Tzannes 1986).

1. It is desirable to treat the image as a probability mass function (pmf) and there are physical justifications for this treatment (Shore 1984). This could be accomplished by normalizing the image so that its pixel values add up to unity. However, this normalization is not numerically necessary, for this algorithm. If each image in a group has the same sum of pixel values (same dc level), the images can be treated as pmfs, for this work.
2. Assume \( p(j,k) \) is a uniform image and maximize the entropy of \( f(j,k) \) subject to a single constraint that is specified by the first retained coefficient. This results in a first estimate of the reconstructed image. Maximization of the entropy of the image refers to maximizing RE of \( f(i,j) \) when \( p(i,j) \) is uniform. This maximization of the entropy reduces to merely producing a uniform image with the same grayness level as the original. The IDCT given the first coefficient and the rest set to zero would produce an identical image. Thus:

\[
f_0(i,j) = \frac{C(0,0)}{N}
\]

(3.24)

for \( i,j = 0, 1, 2, \ldots, N-1 \). This is analogous to equation (3.9) for the MEP prior.

3. The first estimate obtained by the previous step is now used as the prior \( p(j,k) \) in the MREP with the second retained coefficient as a constraint. The minimization is simple since only one constraint is used and one Lagrangian multiplier needs to be evaluated. The result is the second estimate of the reconstruction.

4. Repeat step 3 using as a prior the previous estimate and, as a constraint, the next coefficient until all retained coefficients are used. The \( n \)'th estimate is given by:

\[
f_k(i,j) = f_{k-1}(i,j)e^{-X_k g_k(i,j)}
\]

(3.25)
where:

\( f_{k-1} \) is prior

\( X \) is the Lagrangian multiplier

\( g_k(\cdot) \) is the transform basis vector

for the one-dimensional case. The \( X \) is found by applying the constraint:

\[
C_k = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} f_{k-1}(i,j) g_k(i,j)
\]  

(3.26)

where the subscript on the "f" refers to the \( k \)'th estimate, and the subscript on the "C" designates that the \( k \)'th coefficient is being used.

Substitution yields (Tzannes and Tzannes 1986):

\[
C_k = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} f_{k-1}(i,j) g_k(i,j)
\]  

(3.27)

where \( g_k(j,k) \) is the transform basis element whose average over the image represents the \( k \)'th coefficient.

This differs slightly from Shore's MREP extremization (1984):

\[
q(X) = p(X) \exp[- \tau - \sum_{r=0}^{M} \beta_r s_r(X)]
\]  

(3.28)
where the $\beta_r$ and $\tau$ are Lagrangian multipliers. $\beta$ is determined by the restriction that the dc level of the prior equal the dc level of the original. $\tau$ is determined by the new information (the constraint) (Shore 1984).

In this implementation, the dc level restriction is relaxed. It is the same as Shore's when $\tau = 0$. This reduces the system to have one equation and one unknown. Subsequently, each pixel is multiplied by a constant, to normalize the dc level. The difference in dc level is less than one percent, for the binary images. Thus, while there is a theoretical difference, the methods are numerically close enough for practical purposes.

This method will be referred to as the one-pass MREP or the sequential MREP, as all of the constraints are passed through one time. Preliminary results presented by N.S. Tzannes and M.A. Tzannes (1986) showed that this algorithm performs well, often better than the IDCT under the same compression ratios. Its use can lead to greater compression of an image for transmission or storage.

**Iterative MREP**

The above algorithm suffers from one fundamental theoretical deficiency which is investigated in this paper. The MEP and MREP, originally proposed for estimating pmfs or spectra demand that the extremization of the functionals be performed under under the simultaneous satisfaction of all prior knowledge (in the form of constraints). The progressive algorithm discussed above does not
do this; it utilizes one constraint at a time and uses the solution obtained in each step as a prior for the next step. Each recursive solution obtained in this manner satisfies only the most recent constraint and not all the previous ones, and thus the last solution satisfies only the last constraint and not all of them. Of course, it is obvious that the information contained in each constraint is not lost -- it gets incorporated in the resultant estimate which is the prior for the next solution. Therefore, the last solution is not only the last constraint, it is also weighted by the prior which embodies in it information about all the previous constraints. Thinking of the Relative Entropy as a distance, the last solution is closest to the prior (which satisfies the previous constraint), which is closest to the previous prior (which satisfies the constraint before), etc. Thus, philosophically, at least, the information in all the constraints is carried on to the last solution. Mathematically, however, the final reconstruction only satisfies the last constraint with no guarantee that all the other constraints are satisfied.

There are three methods to studying this problem.

1. Theoretically

2. Practically, by comparing the results of the progressive algorithm to the regular approach, the IDCT, and seeing which performs better under the same conditions.
3. Practically, by using the last reconstruction as a prior, and running through the constraints a second, a third time, etc., until the solution stabilizes. These will be referred to as additional passes through the constraints. This will be referred to as the iterative MREP, when more than one pass through the coefficients is used.

All three methods are investigated. Methods 2 and 3 are covered in Chapter 4. Method 1 is the subject of Chapter 5.

**Conclusion**

Relative entropy provides alternative methods for reconstructing transform-coded images. If the images are treated as pmfs, the principles of Maximum and Minimum Entropy can be applied. Two algorithms for entropic reconstruction have been presented: one based upon Maximum Entropy and one based upon Minimum Relative Entropy. The former was optimized for speed.
passes is increased. In the sixth, the MEP mse is compared to the IDCT mse. The seventh measures MREP mse relative to the MEP mse to compare the methods and to test for MREP convergence. Finally, in the last subsection, the distance between all of the reconstructions themselves is measured.

Advantages of 16x16 Binary Images

The size of the binary images yielded two advantages. Firstly, it facilitated program development; secondly, it allowed very long calculations to be performed. An example of a long calculation is the MREP reconstruction of a 16 x 16 image, using 1 through 10 pass, for 1x1, 2x2, 3x3, ..., 16x16 coefficients retained.

Test Images

Five images were used: A5, E, EO, NewO, and the O (see Appendix C). The EO is sometimes referred to as the OE. The NewO is a symmetric "o"; the other one is non-symmetric. The first four were used in all of the studies (see Appendix C).
CHAPTER 4
APPLICATION

Introduction

The three methods of reconstruction, the IDCT, the MREP, and the MEP, will be applied to two types of images: binary images and a black-and-white picture of the Golden Gate Bridge. The second section describes the work with the binary image; the third describes the work with the bridge picture. The fourth section discusses the error of the numerical solution to the non-linear equations.

Binary Images
Introduction

This section focuses on the methodology and results from the 16x16 binary images. The next subsection explains why the 16x16 binary images were studied. In the following subsection, the test images are described. The fourth analyzes the mse distance between the reconstructions and the original to judge MREP performance relative to the IDCT. The fifth subsection notes the convergence of the iterative MREP by checking the mse and RE between the reconstruction and the original as the number of
MREP mse vs. IDCT mse

**TABLE 1**
MREP mse vs. IDCT mse

<table>
<thead>
<tr>
<th>RANGE OF COEFFICIENTS WHERE MREP MSE WAS LESS THAN IDCT MSE</th>
<th>NUMBER OF PASSES</th>
<th>IMAGE USED</th>
</tr>
</thead>
<tbody>
<tr>
<td>3x3, 5x5 - 9x9</td>
<td>1</td>
<td>A5</td>
</tr>
<tr>
<td>3x3 - 15x15</td>
<td>&gt;1</td>
<td>A5</td>
</tr>
<tr>
<td>5x5 - 8x8</td>
<td>1,2</td>
<td>E</td>
</tr>
<tr>
<td>5x5 - 8x8, 13x13</td>
<td>&gt;2</td>
<td>E</td>
</tr>
<tr>
<td>4x4, 7x7 - 12x12, 14x14</td>
<td>1</td>
<td>EO</td>
</tr>
<tr>
<td>3x3 - 4x4, 6x6 - 15x15</td>
<td>2</td>
<td>EO</td>
</tr>
<tr>
<td>3x3 - 15x15</td>
<td>&gt;2</td>
<td>EO</td>
</tr>
<tr>
<td>3x3 - 13x13</td>
<td>1</td>
<td>NewO</td>
</tr>
<tr>
<td>3x3 - 15x15</td>
<td>&gt;1</td>
<td>NewO</td>
</tr>
</tbody>
</table>

One of the largest differences in mse occurred with the image NewO. With 13x13 coefficients retained, the MREP 3-pass had an mse that was 20% less than the mse of the IDCT.

**MREP Convergence**

This data shows that the reconstruction improves markedly with the second pass and some with the third. Additional passes are not very significant.
Examination of the tables reveals that the RE between the reconstruction and the original, \( H(\text{recn, org}) \), is monotonically decreasing. The mse between the reconstruction and the original decreases except for infrequent small jumps. These jumps in mse do correspond to reductions in RE. For instance, with the image A5, for 6x6 coefficients retained, between the 4th and the 5th pass, the mse goes up from 5.53993154 to 5.53993470; the RE declines from 10.47380396 to 10.47380261. Another important observation is the behavior when all 16x16 coefficients are retained; MREP converges to zero mse and zero RE, as the number of passes increases.

**MEP mse vs. IDCT mse**

The ratio of the MEP mse over the IDCT mse was plotted for images A5, E, EO, and NewO for a convergence limit of \( 1/10^3 \). The other O was tabulated, but not plotted. The A5 ratio was also plotted for a convergence of \( 1/10 \). The following table (Table 2) will be made by just reading the tables where the graphs do not have sufficient resolution.
Thus, the MEP performance exceed the IDCT in the same cases as the MREP with 3 or more passes.

**MREP mse vs. MEP mse**

A different approach was tried. The ratio of MEP mse to MREP mse was plotted as a function of the number of retained coefficients. This allowed the MREP 1-pass /MEP to be on a different scale than the MREP 2-pass /MEP. The graphs were plotted for 1-, 2-, 3-, 4-, and 10-pass MREP. Thus, the convergence could be viewed from graph to graph.

An improvement on the graphs was achieved by only to plotting up to 13x13 or 14x14 retained coefficients. The graphs tend toward extremes at the ends, and this can make the scale so large
that one cannot see what is going on over the majority of the
coefficient range.

The results illustrated convergence very well. The following
discussion of this ratio range is valid for 2x2 to 13x13
coefficients retained. For one and two passes, the MREP performed
worse than the MEP; the ratio was greater than one. The mse ratio
after one pass was as high as 10 (approximately). The two-pass
ratio was less than 1.1. After three passes, the MREP mse was
within 1% of the MEP mse (and within 0.1% for two of the
pictures). Skipping to the 10-pass MREP, the difference in mse
between the two methods was less than 0.01%.

A similar procedure was used to see if the MREP mse or the
MREP reconverged to that of the IDCT. The ratio of the MREP re to
the IDCT was plotted as a function of coefficients retained. Each
different graph corresponded to a different number of passes. This
procedure made the convergence of the MREP to the MEP obvious; the
lack of convergence of the MREP to the IDCT was also obvious.

Distance Measures Between
the Reconstructions Themselves

At this time, it was decided to directly measure distance
between the reconstructed images themselves. This is the subject
of Appendix E. A quick look shows that MREP converges to MEP.
Picture of Golden Gate Bridge

Introduction

The picture of the Golden Gate Bridge (see Appendix C) was included to test the performance of the algorithms upon realistic data. The picture is composed of 512x512 pixels with 256 greyness levels. The information content in bits per pixel was 6.28 for the pixels taken one at a time (equation 3.4).

The second subsection discusses the block sizes used in the calculations; the third states the number of coefficients retained. In the fourth, the performance of the MEP is compared to that of the IDCT under the mse criteria. The fifth describes the mse of the MREP. The amount of calculation for the MEP relative to the DCT is tabulated in the sixth subsection.

Blocksize

The 512x512 picture is broken up into smaller blocks for processing. The compression and reconstruction are then performed upon each separately. This is much faster than working with the picture as a whole. The block sizes are 4x4, 8x8, and 16x16. Blocking in these sizes is common in video processing (Haque 1985 and Mailaender 1985).

Number of Retained Coefficients

The three methods of reconstruction were used: IDCT, MREP, and MEP. For each blocksize 1x1, 2x2, 3x3, and 4x4 coefficients are retained.
MEP mse vs. IDCT mse

The mse between the reconstructions and the original were measured for the picture of the Golden Gate Bridge. The MEP was superior to the IDCT for five out of nine cases (Appendix H).

Mse of MREP

The second pass of the MREP can decrease the mse by over 1%. The MREP 2-pass method produces a mse that was within 0.1% of the mse of the MEP. Generally, the third pass of the MREP does little to change the mse. The only exception to the above observations occurred in the case of a 4x4 blocksize and 4x4 coefficients retained; the mse decreases from 176.7 with one pass, to 20.4 with two passes, and to 4.6 with 3 passes. This implies a convergence limit of 0.0 mse, which is reasonable.

Number of MEP Iterations

The average number of iterations per block were calculated for the MEP algorithm. The image was the picture of the Golden Gate Bridge; the blocksize was 4x4.

For 3x3 coefficients retained, the MEP required 4.4 (approximately) iterations per block. With each iteration taking about three times as long as an IDCT, the time for reconstruction was about 13.2 times the time required for an IDCT.

For a blocksize of 8x8, Table 4 is obtained.
### TABLE 3
MEP SPEED WITH A 4X4 BLOCKSIZE

<table>
<thead>
<tr>
<th>AVERAGE ITERATIONS PER BLOCK</th>
<th>COEFFICIENTS RETAINED</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1x1</td>
</tr>
<tr>
<td>3.77338</td>
<td>2x2</td>
</tr>
<tr>
<td>4.41498</td>
<td>3x3</td>
</tr>
<tr>
<td>4.75409</td>
<td>4x4</td>
</tr>
</tbody>
</table>

### TABLE 4
MEP SPEED WITH AN 8X8 BLOCKSIZE

<table>
<thead>
<tr>
<th>AVERAGE ITERATIONS PER BLOCK</th>
<th>COEFFICIENTS RETAINED</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1x1</td>
</tr>
<tr>
<td>4.94165</td>
<td>2x2</td>
</tr>
<tr>
<td>5.63916</td>
<td>3x3</td>
</tr>
<tr>
<td>5.95435</td>
<td>4x4</td>
</tr>
<tr>
<td>6.16553</td>
<td>5x5</td>
</tr>
</tbody>
</table>
Error Allowed in Numerical Solution of Non-linear Equations

There are two functions that are iteratively driven to zero to find the Lagrangian multipliers. These two functions are equation (3.22) for the MEP and equation (3.28) for the MREP. The numerical solution necessitates an error value; if the absolute value of the function is less than the error value, the iteration stops.

Error of Non-linear Solution

Two limits of error for the numerical non-linear equation solution algorithm were widely used: $1/10^6$ and $1/10^3$. The former was used for the comparisons presented in this thesis (unless otherwise noted), although there was only a small difference in performance. A limit of $1/100$ increases the mse of the reconstruction. From a standpoint of computational efficiency, $1/10^3$ is optimal.

Conclusion

The work with the binary images illustrated the general superiority of MEP and the MREP 3-pass to the IDCT. The reconstructions of the Golden Gate Bridge picture showed the MEP and MREP 3-pass were only slightly superior to the IDCT. Unfortunately, one grey-level picture is not sufficient to judge the overall performance of these reconstruction methods on
pictorial data. The studies found the MREP to converge to the MEP, as the number of passes increased.

More work needs to be done to perform quantitative and qualitative analysis of the reconstructions after coding real picture data. One could test the affect of the order of the coefficient sequence is important for the MREP technique. Also, one could employ additional prior knowledge in the reconstruction. For example, "extremize the entropy to produce an image that satisfies the knowledge we have about the coefficients (include quantization error) and has integer pixel values between 1 and 256, inclusive."
CHAPTER 5
THEORETICAL ANALYSIS OF MREP CONVERGENCE

Introduction

This chapter shows that the limit of the convergence of the MREP is the same as that of the MEP, as the set of equations in the MREP are the same as the equations in the MEP. No attempt will be made to prove that this convergence does actually occur.

The next section illustrates the convergence limit in a simple one-dimensional example where two coefficients are retained and gives some more information about the implementation of the MREP algorithm. The third section presents a slightly more formal proof.

An Illustration

MREP, looping once through all the known, was said to only satisfy the last coefficient, but contain information from the previous constraints. It was shown numerically to converge for one constraint at a time. When one is solved for one variable, the solution is effectively substituted into the next equation. This constitutes a set of nonlinear equations.

MREP assumes a uniform prior and satisfies the first unknown. The one-dimensional version of the satisfaction of the
known constraint is given by equation (3.28); the resulting solution is given by equation (3.26). Solving these for the first constraint (with k=1):

\[ C_1 = \sum_{i=0}^{N-1} f_0(i) e^{g(i)} g_2(i) \]  
\[ f_1(i) = f_0(i) e^{-X_1g_1(i)} \]

with k = 2 for the 2nd constraint.

\[ C_2 = \sum_{i=0}^{N-1} f_1(i) e^{-X_2g_2(i)} g_2(i) \]

Substituting in solution for \( f_1 \):

\[ C_2 = \sum_{i=0}^{N-1} f_0(i) e^{-X_1g_1(i)} - X_2g_2(i) \]

simplifying:

\[ C_2 = \sum_{i=0}^{N-1} f_0(i) e^{-X_1g_1(i)} - X_2g_2(i) \]
If there were only two coefficients (not including the zero-order coefficient) retained in this one-dimensional matrix, the above equations represent one pass.

For a second pass, the solution for $x_2$ would be put into:

$$C_1 = \sum_{i=0}^{N-1} f_0(i) e^{-x_2 g_2(i)} g_1(i)$$

(5.6)

and $x_3$ would be the unknown. This can be written as solving for $x_1$, where $x_1 = \text{sum of all prior solutions} = x_3 + x_1$.

$$C_1 = \sum_{i=0}^{N-1} f_0(i) e^{-x_2 g_2 V(i)} g_1(i)$$

(5.7)

Then $x_2$ would be solved for as the unknown, with $x_1$ constant.

$$C_2 = \sum_{i=0}^{N-1} f_0(i) e^{-x_1 g_1(i)} - x_2 g_2(i)$$

(5.8)

Additional passes would iterate through the previous two equations.

These two equations, that are being driven to zero, are the same as the equation (3.6) for the MEP. Technically, the MEP would also be solving for $x_0$, by the MREP notation. This is also solved by the MREP, via the normalization after finding each
Lagrangian multiplier. Thus, MREP solves the same simultaneous
equation as the MEP.

The zero-order coefficient could be looped through once per
pass, instead of being satisfied after every non-linear solution;
this method would probably converge. There are two main
advantages in performing the normalization each time.

Firstly, it is very quick, as it simply entails summing the
pixels and multiplying them by a factor. Actually, it would
probably suffice to add the same constant to each pixel. Since
addition is faster than multiplication, it would probably be a bit
faster. This was not investigated numerically.

Secondly, the definition of relative entropy requires that
the two images have the same dc level; that is the basis of the
MREP equation. Although, the success of the "simultaneous" method
implies that the normalization might only be necessary once per
pass.

Non-rigorous Proof

This proof assumes that a system of simultaneous equations
(specifically the MEP equation, 3.7) can be solved with the
following algorithm. Assume all unknowns, save one, are set to an
initial value and solve one equation for that unknown. Substitute
this solution into the next equation and solve for another
unknown, setting the other unknowns equal to the initial value.
Repeat until all the equations are used. (The one-pass MREP is an
implementation of this method upon the MEP equation.) This loop through the equations is repeated until the values obtained for the unknowns cease to change (converge). If this happens, the solution to the set of simultaneous equations has been found; there is no proof that this will occur for an arbitrary set of equations.

The main task is to show that the iterative MREP equations reduce to the MEP set of simultaneous equations. This is straightforward. Two "compressions" are involved. First, an expression for the solution after the one-pass MREP is obtained from the separate equations. Secondly, this expression is "compressed" to show the result after several passes of the MREP. The generalized form will correspond to the MEP equation.

The main problem lies in the notation. Many different subscripts and increments will be used and it may become confusing. A quick overview will be helpful. The data-domain increments are "i" and "j." The frequency-domain increments are "m" and "n;" however, they are not used because each known frequency coefficient corresponds to a separate non-linear equation. The equations are more easily visualized sequentially; thus, the subscript "k" will designate the frequency. There are K known coefficients. The range of "k" will be 1 to K. If k = 0, the prior for a given pass is indicated. The Lagrangian multipliers, "\(X\)," will be superscripted by a "q,r." The
frequency is designated by "q" (q = 0 means zero frequency) and "r" is used to designate the pass (r = 1 means the first pass).

The expression for the solution after one pass is:

\[ f_{K,1}(i,j) = f_{0,1}(i,j) e^{\sum_{q=0}^{k} X_{q,1} g_q(i,j)} \]  

The "1" means this is the first pass. Each \( X_{q,1} \) came from:

\[ C_k = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} f_{0,1}(i,j) e^{\sum_{q=0}^{k} X_{q,1} g_q(i,j)} \]  

for \( k = 1, 2, 3, \ldots, K \).

The above form represents K separate equations. It is solved for \( k = 1, 2, \ldots, K \) in sequence. The unknown Lagrangian multipliers are assumed to be zero. (The fact that \( X_{1,1} \) represents the greyness level is not important. It will be treated as any other Lagrangian multiplier.) After the first multiplier, \( X_{1,1} \), is found, \( k \) is set to 2 to find the second. The estimate of \( X_{1,1} \) is used to find \( X_{2,1} \). The estimates of \( X_{1,1} \) and \( X_{2,1} \) are used to find \( X_{3,1} \). This process continues until the first pass is complete, using all the known coefficients.
Next, the generalized form of the MREP solution after several passes will be developed. The solution after the second pass of the MREP is:

\[ f_{K,2}(i,j) = f_{0,2} e^{\sum_{q=0}^{k} X_{q,2} g_q(i,j)} \]  \hspace{1cm} (5.11)

Substituting in the value of \( f_{0,2}(i,j) \) yields:

\[ f_{K,2}(i,j) = f_{0,2} e^{\sum_{q=0}^{k} X_{q,1} g_q(i,j) \sum_{q=0}^{k} X_{q,2} g_q(i,j)} \]  \hspace{1cm} (5.12)

where \( f_{0,1} \) is the uniform prior of the first pass.

Note that:

\[ f_{0,2}(i,j) = f_{K,1}(i,j) \]  \hspace{1cm} (5.13)

as the prior of the second pass equals the value of the solution after the first pass. More generally, with "r" for the number of the pass:

\[ f_{0,r}(i,j) = f_{K,r-1}(i,j) \]  \hspace{1cm} (5.14)

The Lagrangian multipliers for equation (5.12) are obtained from:
for $k = 1, 2, 3, \ldots, K$.

This is identical to equation (5.10) except for the change of the second subscript of the $f(\ )$ and the $X(\ )$ from a "1" to a "2."

The value of prior $f_0,2(i,j)$ can be substituted in to yield:

$$C_K = \sum_{q=0}^{N-1} \sum_{i=0}^{N-1} f_{0,2}(i,j) e \cdot g_{K}(i,j) \ (5.15)$$

for $k = 1, 2, 3, \ldots, K$.

This was the same substitution that was performed between equations (5.11) and (5.12). Equation (5.12) gives the image after the second pass; equation (5.16) gives the Lagrangian multipliers after the second pass. A more general solution is desired. Next, the equations for a third pass will be written to clarify the pattern.

$$f_{K,3}(i,j)=f_{0,1} e \cdot e \cdot e \ (5.17)$$
where $f_{0,1}$ is the uniform prior of the first pass.

\[
C_k = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} f_{0,1}(i,j) \cdot e^{k \sum_{q=1}^{k} \chi_{q,1} g_{q}(i,j)}
\]

(5.18)

for $k = 1, 2, 3, \ldots, K$.

The previous two equations have three arrays of Lagrangian multipliers. The solution of each non-linear equation comes up with a new value for each Lagrangian multiplier. This new solution is in effect merely added to the old solution. This is the way the algorithm is implemented.

There is another way of viewing this process that is numerically equivalent. The old solution of a Lagrangian multiplier can be considered discarded when it is solved for again. It does not affect the new solution (except for the addition). The three arrays of unknowns can be algebraically combined into one. This new array will be called $Y()$. Rewriting the two MREP equations yields:

\[
f(i,j) = f_{0,1} \cdot e^{\sum_{q=0}^{k} Y_q g_q(i,j)}
\]

(5.19)

where $f_{0,1}$ is the uniform prior of the first pass.
\[ C_k = \sum_{q=1}^{N-1} \sum_{i=0}^{N-1} Y_q g_q(i,j) e^{f_0,1(i,j)} e^{g_k(i,j)} \]  

(5.20)

for \( k = 1, 2, 3, \ldots, K \).

At every pass, the \( Y(\cdot) \)s are solved for again. The old value corresponding to the one being solved for is only used as a prior, it is otherwise discarded. The other old values are used in the equation. There is now only one array of unknowns, no matter how many passes are made. Equation (5.20) represents one set of simultaneous equations derived from the sequence of equations from the iterative MREP. With the use of "m,n" in lieu of "q" to designate the frequency, equations (5.19) and (5.20) are identical to equations (3.6) and (3.8) of the MEP solution.

**Conclusion**

Thus, if the iterative MREP algorithm does converge, it is mathematically equivalent to MEP. However, no attempt has been made to prove that this convergence actually does occur.
APPENDICES
APPENDIX A
SQDCF, A MAPPING FUNCTION

This appendix contains a short description of a mapping function called Squared Discrete Cosine Function and the calculation of the SQDCF via the DCT. This unusual function is of interest as it is used in the fast MEP method.

The difference between this function and the DCT is that the "basis vectors" of the SQDF are the DCT vectors squared. As these vectors are not orthogonal, it is not a transform, only a mapping function. The mapping of a one-dimensional vector, X, of length N is:

\begin{equation}
\text{SQDCF}\{X\}(m) = \sum_{i=0}^{N-1} X(i) \cos^2 \left( \frac{(2i + 1)m\pi}{2N} \right) \tag{A.1}
\end{equation}

The two-dimensional version is:

\begin{equation}
\text{SQDCF}\{X\}(m,n) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} X(i,j) \cos^2 \left( \frac{(2i+1)m\pi}{2N} \right) \cos^2 \left( \frac{(2j+1)n\pi}{2N} \right) \tag{A.2}
\end{equation}
The next part shows the calculation of the SQDCF via the DCT. By using the simple substitution, equation (A.1) can be written:

\[
\text{SQDCF}\{X\}(m) = \sum_{i=0}^{N-1} X(i) * \frac{1}{2} * \left\{ 1 + \cos \left( \frac{(2i + 1)m\pi}{N} \right) \right\} \quad (A.3)
\]

\[
\text{SQDCF}\{X\}(m) = \frac{1}{2} \sum_{i=0}^{N-1} X(i) + \frac{1}{2} \sum_{i=0}^{N-1} X(i) * \cos \left( \frac{(2i + 1)m\pi}{N} \right) \quad (A.4)
\]

The calculation of equation (A.4) is only slightly slower than the calculation of the standard DCT.
APPENDIX B
COMPUTER PROGRAMS
The following program performs reconstruction of a 512x512 picture using MEP and IDCT.

```c
program main
  tues 7/14
  su 7/12 removed calc of g( ,,,,)
  removed extra writes
  made in terms of 16 for substitution(routine def_g.)
  changed loop thru blocks to stop at a
  function of block size (not 31 but 512/siz-1)
  in re changed output filenames
  It will take the 2d-dct w/o using 1d-dct's.
  The idct is done in a similar manner.
  the non-integer variables are : double precision
    the simultaneous will converge to
    1/10**3 in lieu of 1/10*3
    the iterative will converge to
    1/10**3 in lieu of 1/10*3

  su11m0s6  mrep not used sim to 10**-6
  su11m0s20 mrep not used sim to 10**-20
  wk
    1110  remove -x used x
    1100  speed drtni
    1050  wk
  su1050  sim to 1/10**12
  su850   removed some extra writes
  845   wk
  su845   the simultaneous will converge to
          1/10**7 in lieu of 1/10*3
          the iterative will converge to
          1/10**7 in lieu of 1/10*3
  650   wk
  su650   1-10 passes thru coef
  640   wk
  su640   correctly labels the output files (no of iter)
  su630   wk is fastest , the mrep uses previous output as
           prior
  s500   wk
  shifts simultaneous satification output
  uses coef 1x1-16x16
```
double precision g(16,16,16,16), g1(16,16), gsq(16,16)
g(16,16,16,16) only used in ref but left to keep proper
spacing for common block
common g, g1, gsq
call defg1sq
call menu
stop
end

subroutine defg1sq
integer i, m
double precision g(16,16,16,16), g1(16,16), gsq(16,16), pie, c, y
common g, g1, gsq
y = -1.0
pie = dacos(y)
do 10 m = 1, 16
  if (m.eq.1) then
    c = 1.0
  else
    c = 2.0**(0.5)
  end if
  do 20 i = 1, 16
    gl(i, m) = c/((16.0)**(0.5))*dcos((2.0*i-1.0)*(m-1.0)*pie/2.0/16.0)
    gsq(i, m) = gl(i, m)**2.0
  end do
10 continue
20 continue
end

subroutine menu
integer c, fifteen, iter, i, j, vblockno, hblockno
character tp11*80, tp12*80, tp13*80, tp14*80, tp15*80, tpout20*20
double precision tp(0:15, 0:15)
double precision x
character rl1*80, rl2*80, rl3*80, rl4*80, rl5*80
tl1*80, tl2*80, tl3*80, tl4*80, tl5*80
ol1*80, ol2*80, ol3*80, ol4*80, ol5*80

character r1*80, r2*80, r3*80, r4*80, r5*80
t1*80, t2*80, t3*80, t4*80, t5*80
character o1*80, o2*80, o3*80, o4*80, o5*80

subroutine menu
integer c, fifteen, iter, i, j, vblockno, hblockno
c = number of non-zero dct coefficients
integer*2 bigorg(0:511, 0:511), bigidct(0:511, 0:511)
integer*2 bigsim(0:511, 0:511)
integer retain
double precision dcta(0:15, 0:15), orga(0:15, 0:15), idcta(0:15, 0:15)
double precision rela(0:15,0:15),temp(0:15,0:15),sima(0:15,0:15)
character name*20
read (*,*) retain
  c     retain =3
  call redar(bigorg)
c  loop thru ea 16x16 block
do 10 vblockno=0,(512/16)-1
do 11 hblockno=0,(512/16)-1
do 20 j=0,15
do 30 i=0,15
  orga(i,j) = float(bigorg(i+vblockno*16,j+hblockno*16))
30 continue
20 continue
c  call cmtdct(orga,dcta,1)
call afilter(dcta,retain)
call d2mm(retain,dcta,sima)
do 40 j=0,15
do 50 i=0,15
  bigsim(i+vblockno*16,j+hblockno*16)= int(sima(i,j)+0.5)
50 continue
40 continue
c  write (*,*) 'did a block !!! ',hblockno,'vbk= ',vblockno
11 continue
c  write (*,*) ' Fortran: vblockno= ', vblockno, '************
10 continue
name = 'me'
call printa(bigsim,name)
end

subroutine d2mm(p,coef,outa)
double precision outa(1:16,1:16)
c
double precision coef(16,16)
integer p

c  integer i,siz
  integer ier,iend
  double precision prior,xst,eps,rsiz
  double precision dexpo(16,16),x(16,16)
double precision g2(16,16,16,16)
common g2
logical zeros
siz = 16
rsiz = 16.0
c  if your going to subroutine ascale
  prior = coef(1,1)/rsiz
write (*,*) 'd2m: cof(11) ',coef(1,1),' prior ',prior,' siz',siz,
eps = 1.0/10.0**3.0
iend = 500
xst = 0.0

CALL M2DRTNI(X,XST,EPS,IEND,IER,COEF,ZEROS,P)

IF (ZEROS) THEN
    WRITE (*,*) '************ zero slope **'
END IF

DO 10 II = 1, SIZ
   DO 20 JJ = 1, SIZ
      DEXPO(II, JJ) = 0
      DO 30 MM = 1, P
         DO 40 NN = 1, P
            IF ((ITTTL.EQ.1).AND.(NN.EQ.1)).NE.FALSE. THEN
               DEXPO(II, JJ) = DEXPO(II, JJ) + X(MM, NN)*G2(II, JJ, MM, NN)
            END IF
         END DO
      END DO
   END DO
10 CONTINUE
20 CONTINUE

DEXPO(II, JJ) = DEXP(DEXPO(II, JJ))

X(1, 1) = 0.0
CALL AFILTER(X, P)
CALL CMTDCT(X, DEXPO, 2)

WRITE (*,*) '284 prior = ', PRIOR, DEXP = ', DEXPO(I, J)
OUTA(I, J) = PRIOR*DEXP(DEXPO(I, J))

DO 50 I = 1, SIZ
   DO 60 J = 1, SIZ
      WRITE (*,*)
   60 CONTINUE
50 CONTINUE

CALL SHIFT(COEFF, OUTA, DEXPO)
CALL COPY (DEXPO, OUTA)

END

SUBROUTINE M2DRTNI(X, XST, EPS, IEND, IER, COEF, ZEROS, P)
DOUBLE PRECISION X(16, 16), XST, EPS, COEF(16, 16)
INTEGER IEND, IER, P
LOGICAL ZEROS

INTEGER M, N, IREPT
DOUBLE PRECISION STEPFAC, OLDX, DERFSUM, STEPSIZE
DOUBLE PRECISION OLDFX(16, 16), DERFX(16, 16)
ISIZ = 16
SIZ = 16.0
IER = 0
OLDF = 10000

STEPFACT = 2.0
CALL SET(X, XST)
X(1, 1) = 0

CALL M2FCT(X, F, DERF, COEF, P)
IREPT = 0

30 CONTINUE
zeros = .true.
derfsum = 0.0
do 40 m = l, p
do 50 n = 1, p
c    if (((m.eq.1).and.(n.eq.1)).eq..false.) then
derfsum = defsum + (deff(m,n))**2
c    end if
50 continue
40 continue
derfsum = defsum**0.5
stepsize = f/defsum*stepfactor
do 60 m = l, p
do 70 n = 1, p
c    if (((m.eq.1).and.(n.eq.1)).eq..false.) then
oldx(m,n) = x(m,n)
if (((deff(m,n).eq.0).eq..false.) then
zeros = .false.
x(m,n) = x(m,n) - stepsize*deff(m,n)/defsum
end if
70 continue
60 continue
ol df = f
call m2fct(x,f,deff,coef,p)
if ((ol df.lt.f).and.(stepfactor.gt.0.0001)) then
do 75 i = l, p
do 80 j = l, p
c    x(i,j) = oldx(i,j)
c80 continue
c75 continue
call copy(oldx, x)
f = ol df
stepfactor = stepfactor/2
write (*,*), 'm2drtni: -- stepfctr=', stepfactor
else
if (stepfactor.gt.0.0001) then
stepfactor = stepfactor*1.25
write (*,*), 'm2drtni: ++ stepfactor=', stepfactor
else
stepfactor = 10
end if
end if
if (((irept.lt.iend).and((zeros).eq..false.)
* .and.((dabs(f).gt.eps)) goto 30
write (*,*),'irept=', irept
write (*,*),'zeros=', zeros
write (*,*), 'f = ', f
end
subroutine m2fct (x,f,deff,coef,p)
integer ii, jj, m, n, isiz, p
double precision sum1(16,16), sum2(16,16), dexpo(16,16)
double precision g2(16,16,16,16), derv(16,16)
double precision x(16,16), coef(16,16), fac
common g2
double precision, siz, f
siz = 16.0
isiz = 16

\begin{verbatim}
  do 10 ii=1,isiz
    do 20 jj=1,isiz
      dexpo(ii,jj) = 0.0
      do 30 mm=1,p
        do 40 nn=1,p
          dexpo(ii,jj) = dexpo(ii,jj) + x(mm,nn) * g2(ii,jj,mm,nn)
        \enddo 30
      \enddo 20
      dexpo(ii,jj) = dexp(dexpo(ii,jj))
    \enddo 10
  \enddo

  call afilter(x,p)
  call cmtdct(x,dexpo,2)
  fac = coef(1,1)/siz
  do 11 ii=1,isiz
    do 12 jj=1,isiz
      dexpo(ii,jj) = fac * dexp(dexpo(ii,jj))
    \enddo 12
  \enddo 11
  \endverbatim

write (*,*) 'finished dexp loop'
f=0.0

\begin{verbatim}
  do 100 m=1,p
    do 110 n=1,p
      sum1(m,n) = 0.0
      sum2(m,n) = 0.0
      do 120 i=1,isiz
        do 130 j=1,isiz
          sum1(m,n) = sum1(m,n) + g2(i,j,m,n) * dexpo(i,j)
          sum2(m,n) = sum2(m,n) + g2(i,j,m,n) ** 2 * dexpo(i,j)
        \enddo 130
      \enddo 120
      \enddo 110
    \enddo 100
  \enddo

  take dct of dexpo put into sum1
  unneeded sum1's will go unused
  call cmtdct(dexpo,sum1,1)
  call cmtdct(dexpo,sum2,3)
  do 200 m=1,p
    do 210 n=1,p
      if (((n.eq.1) .and. (m.eq.1)) .eq. .false.) then
        f = f + (sum1(m,n) - coef(m,n)) ** 2
        derv(m,n) = sum2(m,n) * 2 * (sum1(m,n) - coef(m,n))
      \endif
  \enddo 210
\end{verbatim}
subroutine redar (ireconi)
   integer*2 ireconi(512,512)
   integer i,j
   character chr*20
   chr = 'inp'
   open(15, file=chr ,status='old')
   rewind 15
   do 100 i=1,512
      do 200 j=1,512
         read (15,*) ireconi(i,j)
      200 continue
   100 continue
   close (15)
   return
end

subroutine printa (ireconi,chr)
   integer*2 ireconi(512,512)
   integer i,j
   character chr*20
   open(15, file=chr ,status='new')
   do 100 i=1,512
      do 200 j=1,512
         write (15,*) ireconi(i,j)
      200 continue
   100 continue
   close (15)
   return
end

subroutine ascale(dcta,rela,thres,13)
   double precision dcta(16,16),rela(16,16)
   double precision thres(16,16),f,sum
   character t3*80,t4*11
   call sumbl(rela,sum)
   f= dcta(1,1)*16/sum
   write (*,*),'ascale: factor = ',f
   call multbl(rela,f,thres)
   t3= 'rel ent w/ 1-pass. normalized'
   t4= 'factor = '
c    write (l3,47) t3,t4,f
        format (a41,a22,f8.4)
        return
end

c subroutine shift(dcta,rela,thres)
    integer i,j
    double precision dcta(16,16),rela(16,16)
    double precision thres(16,16),f,sum
    call sumbl (rela,sum)
    f= (dcta(1,1)*16.0-sum ) /16.0/16.0
    do 10 i=1,16
        do 20 j=1,16
            thres(i,j) =rela(i,j)+f
        20 continue
    10 continue
    return
end

subroutine sumbl(rela,sum)
    double precision rela(0:15,0:15),sum
    integer i,j
    sum= 0.0
    do 12 i=0,15
        do 22 j=0,15
            sum =rela(i,j)+ sum
        22 continue
    12 continue
    return
end

subroutine mutbl(rela,f,thres)
    double precision rela(0:15,0:15),thres(0:15,0:15),f
    integer i,j
    do 10 i=0,15
        do 20 j=0,15
            thres(i,j) =rela(i,j)*f
        20 continue
    10 continue
    return
end

subroutine afilter(orga,j)
    double precision orga(0:15,0:15)
    integer j,l,n
    do 120 n=0,15
        do 140 l=j,15
            orga(n,l)= 0
        140 continue
        do 120 continue
    120 continue

continue
do 35 n = 0, 15
do 45 l = j, 15
org a ( l , n) = 0
continue
35 continueend
c subroutine cmtdct (in, out, q)
double precision in(0:15,0:15), out(0:15,0:15), temp(0:15,0:15)
double precision x(0:15), y(0:15)
integer i, q, j
c write (*,*) 'cmtdct'
do 10 i = 0, 15
do 20 j = 0, 15
x(j) = in(i, j)
20 continue
if (q.eq.1) then
call mtdct (x, y)
end if
if (q.eq.2) then
call mtidct (x, y)
end if
if (q.eq.3) then
call mtsqdct (x, y)
end if
do 25 j = 0, 15
temp(i, j) = y(j)
25 continue
do 40 i = 0, 15
do 50 j = 0, 15
x(j) = temp(j, i)
50 continue
c call mtdct (x, y)
if (q.eq.1) then
call mtdct (x, y)
end if
if (q.eq.2) then
call mtidct (x, y)
end if
if (q.eq.3) then
call mtsqdct (x, y)
end if
do 60 j = 0, 15
out(j, i) = y(j)
60 continue
do 40 continue
c write (*,*) 'cmtdct out(3,7) = ',out(3,7)
c return
cend

c subroutine mtdct(x,y)
integer i,k
double precision x(0:15),y(0:15)
double precision g2(16,16,16,16), gl(0:15,0:15)
common g2,gl
do 20 k=0,15
  y(k)=0.0
do 10 i=0,15
  y(k)= y(k) + x(i)*gl(i,k)
10 continue
20 continue
c write (*,*) 'mtdct y(3) = ',y(3)
c return
cend

subroutine mtidct(x,y)
integer i,k
double precision x(16) ,y(16)
double precision g2(16,16,16,16), g1(16,16)
common g2,gl
内涵 1 do 20 i=1,16
  y(i) = 0.0
do 10 k=1,16
  y(i) = y(i) +x(k)*gl(i,k)
10 continue
20 continue
return
end

subroutine mtsqdct(x,y)
integer i,k
double precision x(16) ,y(16)
double precision g2(16,16,16,16), g1(16,16), gsq(16,16)
common g2,gl,gsq
内涵 2 do 20 i=1,16
  y(i) = 0.0
do 10 k=1,16
  y(i) = y(i) +x(k)*gsq(k,i)
10 continue
20 continue
return
end

subroutine set(orga,r)
c sets orga = r
double precision orga(16,16),r
integer i, j

do 10 i = 1, 16
    do 20 j = 1, 16
        orga(i, j) = r
    20 continue
10 continue
return
end

subroutine copy(ina, outa)
double precision ina(16,16), outa(16,16)
integer i, j

do 10 i = 1, 16
    do 20 j = 1, 16
        outa(i, j) = ina(i, j)
    20 continue
10 continue
return
end

subroutine xpobl(rela, thres)
double precision rela(0:15, 0:15), thres(0:15, 0:15)
integer i, j

do 10 i = 0, 15
    do 20 j = 0, 15
        thres(i, j) = dexp(rela(i, j))
    20 continue
10 continue
return
end

The following program performs 512x512 picture reconstruction with MREP and IDCT.

program main
Sun 7./12 removed extra writes
made equ in terms of 16 for substitution
changed big loop (thru blocks) to be a function of block size
changed output filenames to r1p r2p r3p
It will take the 2d-dct w/o using 1d-dct's.
The idct is done in a similar manner.
the non-integer variables are: double precision
the simultaneous will converge to 1/10**3 in lieu of 1/10*3
the iterative will converge to 1/10**3 in lieu of 1/10*3
double precision g(16,16,16,16), g1(16,16), gsq(16,16)
common g, g1, gsq
call defineg
call defg1sq
call menu
stop
end

subroutine defineg
integer i, j, m, n
double precision pie, y, s, c, d
double precision g(16,16,16,16)
common g
y = -1.0
s = 16.0
pie = dacos(y)
do 10 i=1,16
   do 20 j=1,16
      do 30 m=1,16
         do 40 n=1,16
            if (m.eq.1) then
               c=1
            else
               c= 2.0**0.5
            end if
            if (n.eq.1) then
               d=1.0
            else
               d= 2.0**0.5
            end if
            g(i,j,m,n)=c*d/s*dcos(((2*i-1)*(m-1)*pie)/(2*s))
            * dcos(((2*j-1)*(n-1)*pie)/(2*s))
            continue
         30 continue
      20 continue
   10 continue
end

subroutine defglsq
integer i,m
double precision g(16,16,16,16), g1(16,16),gsq(16,16), pie,c,y
double precision s,sq
common g , g1 , g sq
s = 16.0
sq = s**0.5
y = -1.0
pie = dacos(y)
do 10 m = 1 ,16
   if (m.eq.1) then
      c = 1.0
   else
      c= 2.0**0.5
   end if
   do 20 i = 1,16
      g1(i,m) = c/sq*dcos ((2.0*i-1.0)*(m-1.0)*pie/2.0/s)
      gsq(i,m) = g1(i,m)**2.0
      write (*,'(10f5.1)') 'dfgl: ', g1(i,m)
   20 continue
10 continue
end
double precision x
character rl1*80,rl2*80,rl3*80,rl4*80,rl5*80
character tl1*80,tl2*80,tl3*80,tl4*80,tl5*80
character ol1*80,ol2*80,ol3*80,ol4*80,ol5*80

c subroutine menu
integer c,fifteen,iter,i,j,vblockno,hblockno

  c = number of non-zero dct coefficients
integer*2 bigorg(0:511,0:511)
integer*2 bigrel1(0:511,0:511)
integer*2 bigrel2(0:511,0:511),bigrel3(0:511,0:511)
integer retain
double precision dcta(0:15,0:15),orga(0:15,0:15),idcta(0:15,0:15)
double precision rela(0:15,0:15),temp(0:15,0:15),simax(0:15,0:15)
character name*20
read (*,*) retain

  retain =3
  call redar(bigorg)

  loop thru ea 16x16 block
  do 10 vblockno=0,(512/16)-1
  do 11 hblockno=0,(512/16)-1
  do 20 j=0,15
  do 30 i=0,15
    orga(i,j) = float(bigorg(i+vblockno*16,j+hblockno*16))
  30 continue
  20 continue
  call cmtdct(orga,dcta,1)

  iter =1
  call mmereco(retain,dcta,rela,iter)
  do 80 j=0,15
  do 90 i=0,15
    bigrel1(i+vblockno*16,j+hblockno*16)= int(rela(i,j)+0.5)
  90 continue
  80 continue

  iter =2
  call mmereco(retain,dcta,rela,iter)
  do 81 j=0,15
  do 91 i=0,15
    bigrel2(i+vblockno*16,j+hblockno*16)= int(rela(i,j)+0.5)
  91 continue
  81 continue

  iter =3
  call mmereco(retain,dcta,rela,iter)
  do 82 j=0,15
  do 92 i=0,15
    bigrel3(i+vblockno*16,j+hblockno*16)= int(rela(i,j)+0.5)
  92 continue
  82 continue
c  write(*,*), 'did a block,!!!,' , hblockno, ' vblockno=', vblockno 
11 continue 

c  write(*,*) 'Fortran: vblockno= ', vblockno, ' ************' 
10 continue 

name = 'r1' 
call printa(bigre11,name) 
name = 'r2' 
call printa(bigre12,name) 
name = 'r3' 
call printa(bigre13,name) 
end 

c subroutine redar (ireconi) 
integer*2 ireconi(512,512) 
integer i,j 
character chr*20 
chr = 'inp' 
199 format(a20) 
open(14, file=chr, status='old') 
rewind 14 
do 100 i=1,512 
do 200 j=1,512 
read(14,*) ireconi(i,j) 
200 continue 
100 continue 
close (14) 
return 
end 

c subroutine printa (ireconi,chr) 
integer*2 ireconi(512,512) 
integer i,j 
character chr*20 
199 format(1x,i3) 
open(14, file=chr, status='new') 
do 100 i=1,512 
do 200 j=1,512 
write(14,*) ireconi(i,j) 
200 continue 
100 continue 
close (14) 
return 
end 

c subroutine ascale(dcta,rela,thres,13) 
double precision dcta(16,16),rela(16,16) 
double precision thres(16,16),f,sum
character l3*80,t3*45,t4*11

character l3*80,t3*45,t4*11

call sumbl(rela,sum)
f = dcta(l,1)*16/sum
call multbl(rela,f,thres)
return
end

c subroutine sumbl(rela,sum)
double precision rela(0:15,0:15),sum
integer i,j
sum = 0.0
do 12 i=0,15
do 22 j=0,15
   sum = rela(i,j)+ sum
22 continue
12 continue
return
end

c subroutine multbl(rela,f,thres)
double precision rela(0:15,0:15),thres(0:15,0:15),f
integer i,j
do 10 i=0,15
do 20 j=0,15
   thres(i,j) = rela(i,j)*f
20 continue
10 continue
return
end

c subroutine cmtdct(in,out,q)
double precision in(0:15,0:15),out(0:15,0:15),temp(0:15,0:15)
double precision x(0:15),y(0:15)
integer i,q,j

write (*,*),'cmtdct'
do 10 i=0,15
do 20 j=0,15
   x(j) = in(i,j)
20 continue
if (q.eq.1) then
call mtdct(x,y)
end if
if (q.eq.2) then
call mtidct(x,y)
end if
if (q.eq.3) then
call mtsq dct(x,y)
end if
25 continue
 temp(i,j)=y(j)
73

25     continue
10     continue

40     do 50 i = 0, 15
50     do 50 j = 0, 15
50     continue

10     if (q .eq. 1) then
11     call mtdct(x, y)
12     end if
13     if (q .eq. 2) then
14     call mtidct(x, y)
15     end if
16     if (q .eq. 3) then
17     call mtsq dct(x, y)
18     end if
19     do 60 j = 0, 15
20     out(j, i) = y(j)
20     continue

40     continue

C     write ('*, *) 'cmtdct out(3,7) = ', out(3,7)

C     return

end

subroutine mtdct(x, y)
integer i, k
double precision x(0:15), y(0:15)
double precision g2(16, 16, 16, 16), g1(0:15, 0:15)
common g2, g1
20     do 10 k = 0, 15
10     y(k) = 0.0
20     continue

C     write ('*, *) 'mtdct y(3) = ', y(3)
return
end

subroutine mtidct(x, y)
integer i, k
double precision x(16), y(16)
double precision g2(16, 16, 16, 16), g1(16, 16)
common g2, g1
20     do 10 i = 1, 16
10     y(i) = 0.0
20     continue

C     write ('*, *) 'mtidct y(3) = ', y(3)
return
end
subroutine mtsq dct(x,y)
   integer i,k
   double precision x(l6),y(l6)
   double precision g2(16,16,16,16), g1(16,16)
   g2 common g2,gl,gsq
   do 20 i=1,16
       y(i) = y(i) +x(k)*gsq(k,i)
   continue
   return
end

subroutine mrereco(p,y,old,iter)
   p= no of coef to reconstruct with
   y= input array of coef.
   reconsu= output recon array
   n= ibl= image block length
   procedure for image reconstruction using iterative minimum
   cross entropy me
   double precision f,derf,x,xst,eps,s,y(l6,16)
   integer p,k,n,i,ii,iend,iter
   double precision ascin(16,16),old(16,16),g(16,16,16,16)
   common g
   character ch80*80
   n=16
   xst=0.0
   iend=500
   eps=0.001
   write (*,*) 'reconstructing with ',p,' coef'
   k=1
   s=float(n)
   if (iter.eq.1) then
      fac = y(l,1)/s
      call set(old,fac)
   end if
   do 12 ii=l,iter
      do 14 mm=l,p
         write (*,*) ' next row mm= ',mm
         do 89 nn=l,p
            write (*,*) f,derf,x,xst,eps,s,y(l6,16)
   continue
   return
end
if ( ( not( mm.eq.1) ) .or. ( not (nn.eq.1)) ) then
    call drtni(x,f,derf,xst,eps,iend,ier,mm,nn,old,y,n)
    if (ier.ne.0) print*,'ier=',ier
    do 35 i=1,n
        do 45 j=1,n
            asc(i,j)= old(i,j)*dexp(x*g(i,j,mm,nn))
        continue
    call ascale(y,ascin,old,ch80)
end if
continue
continue
continue
continue
write (*,*),'ne more fini: old(1,1) = ',old(1,1)
return
end

subroutine drtni

purpose
to solve general nonlinear equations of the form f(x)=0
by means of newton-s iteration method.

usage
call drtni (x,f,derf,fct,xst,eps,iend,ier,mm,nn)
parameter fct requires an external statement.

description of parameters
x  - double precision resultant root of equ f(x)=0
f  - double precision resultant function value at root x
derf - double precision resultant value of derivative at root x
fct  - name of the external subroutine used, it computes
to given argument x function value f and derivative
   its parameter list must include x,f,derf, where
   all parameters are double precision.
xst - double precision input value which specifies the
      intial guess of the root x.
eps - single precision input value which specifies the
      upper bound of the error of result x.
iend - max. nu. of iteration steps specified.
ier - resultant error parameter coded as follows
   ier=0 - no error
   ier=1 - no convergence after iend iteration steps
   ier=2 - at any iteration step derivative derf was
   equal to zero.

remarks
the procedure is bypassed and gives the error message ier =2
if at any iteration step derivative of f(x) is equal to 0.
possibly the procedure would be successful if it is started
once more with another initial guess xst.

subroutines and function subprograms required
the external subroutine fct(x,f,derf,?,?) must be furnished
by the user.

method
solution of equation f(x)=0 is done by means of newton-s
iteration method, which starts at the initial guess xst of
a root x. convergence is quadratic if the derivative of
f(x) at root x is not equal to zero. one iteration step
requires one evaluation of f(x) and one evaluation of the
derivative of f(x). for test on satisfactory accuracy see
formula (2) of mathematical description.
for reference, see r. zurmuehl, praktische mathematik fuer
ingenieure und physiker, springer, berlin/goettingen/

subroutine drtni(x,f,derf, xst,eps,iend,ier,mm,nn,sb,y,n)

double precision x,f,derf,xst,eps,dx
double precision b(430,16,16), y(16,16),sb(16,16)
double precision y(16,16),sb(16,16)
integer i,n,iend,ier,mm,nn

prepare iteration
ier =0
x=xst
call fct(x, f,derf,mm,nn,sb,y,n)

start iteration loop
i = 0
if (f) 1,7,1
equation is not satisfied by x
1 if (derf) 2,8,2

iteration is possible
2 x=x+f/derf
i= i+1
x=x-dx
call fct(x,f,derf,mm,nn,sb,y,n)
test on satisfactory accuracy
if (i.gt.iend) goto 20
if (dabs(f).gt.eps) goto 1
end of iteration loop
return
no convergence after iend iteration steps. error return.
20 continue
write (*,*) 'drtni: failed to converge in ',i,' steps'
  ier=1
7 return

c error return in case of zero divisor
8 ier=2
return
end

subroutine fct(x,f,derf,mm,nn,sb,y,n)
double precision x,f,derf,sum1,sum2
double precision sb(16,16), y(16,16),g(16,16,16,16)
double precision newl(16,16),sb(16,16)
common g
integer n
sum1=0.0
sum2=0.0
do 10 i=1,n
  do 20 j=1,n
    newl(i,j)= sb(i,j)*dexp(x*g(i,j,mm,nn))*g(i,j,mm,nn)
    sum1=sum1+newl(i,j)
    sum2=sum2+newl(i,j)*g(i,j,mm,nn)
20 continue
10 continue
f=sum1-y(mm,nn)
derf=-sum2
return
end

subroutine set(orga,r)
c sets orga = r
double precision orga(16,16),r
integer i,j
do 10 i=1,16
  do 20 j=1,16
    orga(i,j) = r
 20 continue
10 continue
return
end

The following program works with binary images performing the MREP, MEP, and IDCT reconstructions. It was used to generate Appendix F.
the non-integer variables are : double precision
the simultaneous will converge to
1/10**6
the iterative will converge to
1/10**6

double precision g(16,16,16,16) ,g1(16,16),gsq(16,16)
common g ,g1,gsq
call defineg
call defglsq
call menu
stop
end

subroutine defineg
integer i,j,m,n
double precision pie,y ,s,c,d
double precision g(16,16,16,16)
common g
y= -1.0
s = 16.0
pie = dacos(y)
do 10 i=l,16
   do 20 j=l,16
      do 30 m=l,16
         if (m.eq.1) then
            c=1
         else
            c= 2.0**0.5
         end if
         do 40 n=l, 16
            if (n.eq.1) then
               d=1.0
            else
               d= 2.0**0.5
            end if
            g(i,j,m,n)=c*d/s*dcos(((2*i-l)*(m-l)*pie)/(2*s))*
dcos(((2*j-l)*(n-l)*pie)/(2*s))
40 continue
30 continue
20 continue
10 continue
end

subroutine defglsq
integer i,m
double precision g(16,16,16,16), g1(16,16),gsq(16,16), pie,c,y
common g, gl, gsq
y = -1.0
pie = dacos(y)
do 10 m = 1, 16
  if (m.eq.1) then
    c = 1.0
  else
    c = 2.0**0.5
  end if
do 20 i = 1, 16
  gl(i,m) = c/4.0*dcos ((2.0*i-1.0)*(m-1.0)*pie/32.0)
  gsq(i,m) = gl(i,m)**2.0
  write (*,*) 'valgl: ', gl(i,m)
  20 continue
  10 continue
end
subroutine menu
integer c,k,fifteen,iter
c = number of non-zero dct coefficients
double precision dcta(0:15,0:15), orga(0:15,0:15), idcta(0:15,0:15)
double precision rela(0:15,0:15), temp(0:15,0:15), thres(0:15,0:15)
double precision mepa(0:15,0:15)
character tpl1*80, tpl2*80, tpl3*80, tpl4*80, tpl5*80, tpout20*20
double precision tp(0:15,0:15)
double precision x
c character tempil3*80
c character str0*3, str1*3, str2*3, out20*20
c do 946 inc = (incb+3), (incb+3)
c
fifteen = 15
c = 16
c write (*,*) 'what do you want with the arrays'
call redar(orga, ol2, ol3)
call cmtdct(orga, dcta, 1)
c controls the number of pixels used in reconstruction
c
c i15= 'METHOD      MSE       RE1'
r15= '   R2'
write (*,288) i15, r15
288 format (a30,a30)
do 129 inc = 0, 14
c = 16 -inc

call afilter (dcta,c)
call cmtdct(dcta,idcta,2)
call mse(idcta,orga,il5)
write (*,245) ' idct/org ',il5
245 format (a12,a62)
c
build 16(c,dcta,mepa)
call mse(mepa,orga,t15)
write (*,245) ' mep/org ',t15
call mse(mepa,idcta,t15)
write (*,245) ' mep/idct ',t15
m=16

do 283, iter = 1,10
call mrereco(c,dcta,rela,m,iter)
call ascale(dcta,rela,thres,rl3)
call copy(thres,rela)
call mse (rela,orga,rl5)
write (*,853) ' rel/iter',/org',rl5
call mse (rela,idcta,rl5)
write (*,853) ' rel/iter',/idct ',rl5
call mse (rela,mepa,rl5)
write (*,853) ' rel/iter',/mep ',rl5
write (rl3,397) ' iterative min relative entropy: ',iter,' passes'
397 format (a34,i2,a8)
282 format (a1,i2)
853 format(a5,i2,a5,a60)
283 continue

c end iterative loop begin simultaneous

c call label (str0,str1,n1,str2,n12,out20)

c continue

c end

c subroutine d2rm(p,coef,outa)
double precision outa(1:16,1:16)
double precision coef(16,16)
integer p

integer i,siz
integer ier,iend
double precision prior,xst,eps,rsiz
double precision dexpo(16,16),x(16,16)
double precision g2(16,16,16,16)
common g2
logical zeros
siz = 16
rsiz = 16.0
if your going to subroutine ascale
prior = coef(l,1)/rsiz
write (*) 'd2m: cof(l,1) ',prior ',prior,' siz',siz
eps = 1.0/10.0**9.0
iend =500
xst =0.0
call m2drtni(x,xst,eps,iend,ier,coef,zeros,p)
if (zeros) then
write (*) ' ********* zero slope **'
end if
do 10 ii=l,siz
do 20 jj=l,siz
dexpo(ij,jj)= 0
do 30 mm=l, p
do 40 nn=l, p
if (((mm.eq.l).and.(nn.eq.1)).eq..false.)then
dexpo(ii,jj)=dexpo(ii,jj)+x(mm,nn)*g2(ii,jj,mm,nn)
end if
continue
continue
x(l,1) = 0.0
call afilter(x,p)
call cmtdct(x,dexpo,2)
do 50 i=l,siz
do 60 j=l,siz
write (*) '284 prior = ',prior,' dexp= ', dexpo(i,j)
outa(i,j) = prior*dexp(dexpo(i,j))
continue
continue
write (*) 'outa(i,j) = prior*dexp(dexpo(i,j))
50 continue
50 continue
call shift(coef,outa,dexpo)
call copy (dexpo,outa)
end
subroutine m2drtni(x,xst,eps,iend,ier,coef,zeros,p)
double precision x(16,16),xst,eps,coef(16,16)
integer iend,ier,p
logical zeros
integer m,n,irept
double precision stepfactor,oldf,f,derfsum ,stepsize
double precision oldx(16,16), derf(16,16)
isiz =16
siz = 16.0
ier = 0
oldf = 10000
stepfactor = 2.0

call set (x, xst)

c x(1,1) = 0
call m2fct (x, f, derf, coef, p)
irept = 0

30 continue
zeros = .true.
derfsum = 0.0
do 40 m = 1, p
  do 50 n = 1, p
    c if (((m.eq.1).and.(n.eq.1)).eq..false.) then
      derfsum = derfsum + (derf(m,n))**2
    c end
    if
50 continue
40 continue
derfsum = derfsum**0.5
stepsize = f/derfsum*stepfactor

do 60 m = 1, p
  do 70 n = 1, p
    c if (((m.eq.1).and.(n.eq.1)).eq..false.) then
      oldx(m,n) = x(m,n)
    if ((derf(m,n).eq.0).eq..false.) then
      zeros = .false.
      x(m,n) = x(m,n) - stepsize*derf(m,n)/derfsum
    c end
    if
70 continue
60 continue
oldf = f
call m2fct (x, f, derf, coef, p)
if ((oldf.lt.f).and.(stepfactor.gt. 0.0001)) then
  c do 75 i = 1, p
  c do 80 j = 1, p
  c x(i,j) = oldx(i,j)
80 continue
75 continue
call copy (oldx, x)
f = oldf
stepfactor = stepfactor/2
c write (*,*) 'm2drtni: -- stepfctr=', stepfactor
else
  if (stepfactor.gt.0.0001) then
    stepfactor = stepfactor*1.25
  write (*,*) 'm2drtni: ++ stepfactor=', stepfactor
else
  stepfactor = 10
end if
end if
if ((irept.lt.iend).and.((zeros).eq..false.)
  .and.(dabs(f).gt.eps)) goto 30
c  write (*,*) 'irept=',irept
c  write (*,*) 'zeros=',zeros
c  write (*,*) 'f=',f
end
subroutine m2fct (x,f,derf,coef,p)
  integer ii,jj,m,n,isiz,p
  double precision sum1(16,16),sum2(16,16),dexpo(l6,16)
  double precision g2(16,16,16,16),derf(16,16)
  double precision x(16,16),coef(16,16),fac
  common g2
  double precision, siz,f
  siz = 16.0
  isiz =16
  do 10 ii=l,isiz
    do 20 jj=l,isiz
      dexpo(ii,jj) =0.0
      do 30 mm=l,p
        do 40 nn=l,p
          dexpo(ii,jj) = dexpo(ii,jj) + x(mm,nn)*g2(ii,jj,mm,nn)
        continue
        continue
      dexpo(ii,jj)= dexp(dexpo(ii,jj))
    continue
    continue
    call afilter(x,p)
    call cmtdct(x,dexpo,2)
    fac = coef(1,1)/siz
    do 11 ii=l,isiz
      do 12 jj=l,isiz
        dexpo(ii,jj)= fac*dexp(dexpo(ii,jj))
      continue
      continue
    write (*,*) 'finished dexp loop'
    f=0.0
    do 100 m=l,p
      do 110 n=l,p
        sum1(m,n) = 0.0
        sum2(m,n) = 0.0
        do 120 i=l,isiz
          do 130 j=l,isiz
            sum1(m,n)=sum1(m,n) * g2(i,j,m,n) *dexpo(i,j)
            sum2(m,n)=sum2(m,n)+g2(i,j,m,n)**2*dexpo(i,j)
          continue
          continue
        take dct of dexpo put into sum1
unneeded sum1's will go unused

```fortran
     call cmtdct(dexpo,sum1,1)
     call cmtdct(dexpo,sum2,3)
     do 200 m=1,p
     do 210 n=1,p
     if (((n.eq.1).and.(m.eq.1)).eq..false.) then
         f=f+(sum1(m,n)-coef(m,n))**2
         derf(m,n)=sum2(m,n)*2*(sum1(m,n) - coef(m,n))
     end if
     210 continue
     200 continue
end
```

```fortran
subroutine redar (ireconi,12,13)
    double precision ireconi(16, 16)
    integer i ,j
    character chr*20,l2*80,l3*80
    chr = 'inp'
    199 format(a20)
    open(l5, file=chr ,status='old')
    rewind 15
    do 100 i=1,16
    do 200 j=1,16
        read (15,*) ireconi(i,j)
    200 continue
    100 continue
    l1 = ' '
    close (15)
    return
end
```

```fortran
subroutine mse(orga,temp,15)
    character 15*80
    double precision orga(16,16 ),temp(16,16)
    integer i,j
    double precision x,y,z
    x=0.0
    do 10 i=1,16
    do 20 j=1,16
        x=x + (orga(i,j)-temp(i,j))**2
    20 continue
    10 continue
    y=x**0.5
    call relent (orga,temp,x)
    call relent (temp,orga,z)
    write (l5,47) ' ',y,' ','x', ' ',z
    47 format (1x,a2,f12.8,a5,f12.8,a5,f12.8)
```
subroutine ascale(dcta,rela,thres,l3)
  double precision dcta(16,16),rela(16,16)
  double precision thres(16,16),f,sum
  character l3*80,t3*45,t4*11
  call sumbl(rela,sum)
  f= dcta(1,1)*16/sum
  call multbl(rela,f,thres)
  t3= 'rel ent w/ 1 -pass. normalized'
  t4= 'factor = '
  write (13,47) t3,t4,f
47  format (a41,a22,f8.4)
return
end

subroutine shift(dcta,rela,thres)
  integer i,j
  double precision dcta(16,16),rela(16,16)
  double precision thres(16,16),f,sum
  call sumbl (rela,sum)
  f= (dcta(1,1)*16.0-sum ) /256.0
  do 10 i=1,16
    do 20 j=1,16
      thres(i,j) =rela(i,j)+f
20 continue
10 continue
return
end

subroutine sumbl(rela,sum)
  double precision rela(0:15,0:15) ,sum
  integer i,j
  sum= 0.0
  do 12 i=0,15
    do 22 j=0,15
      sum =rela(i,j)+ sum
22 continue
12 continue
return
end

subroutine multbl(rela,f,thres)
  double precision rela(0:15,0:15),thres(0:15,0:15),f
  integer i,j
  do 10 i=0,15
    do 20 j=0,15
      thres(i,j) =rela(i,j)*f
20 continue
10 continue
return
end

subroutine athreshold(orga,temp,13,x)
character 13*80
double precision orga(0:15, 0:15),temp(0:15, 0:15)
integer i,j
double precision x
write (13,47) 'after thresholding with ',x
47 format (a24,f8.4)
do 10 i=0,15
do 20 j=0,15
if (orga(i,j) .ge.x) then
  temp(i,j)=2.0
else
  temp (i , j ) = 1.0
end if
20 continue
10 continue
end

subroutine afileout(orga,ll,12,13,14,15,c,k,ch8, * tp,tpl1,tpl2,tpl3,tpl4,tpl5,a,b,tpout20)
double precision orga(0:15,0:15)
integer i,j,k,c,a,b
character ch8*20,11*80,12*80,13*80,14*80,15*80,ch(0:15)*1
character tpl1*80,tpl2*80,tpl3*80,tpl4*80,tpl5*80,tpout20*20
double precision tp(0:15,0:15)
c write (*,*) 'afilout: opening ch8= ',ch8,' k= ',k
c write (*,*) '13 = ',13
open ( 12,file=ch8, status='new')
c write (*,*) '1= fast, 2=one digit,3=reg 4=2dig,5=x-'
do 110 i=0,15
if (k.eq.1) then
  write (12,20) (orga(i,j),j=0,15)
20 format (1x,16f3.0)
end if
if (k.eq.2) then
  write (12,120) (orga(i,j),j=0,15)
120 format (1x,16f5.0)
end if
if (k.eq.3) then
  write (12,25) (orga(i,j),j=0,15)
25 format (1x,16f5.2)
end if
if (k.eq.4) then
  write (12,121) (orga(i,j),j=0,15)
121 format (1x,16f4.1)
end if
110 continue
if (k.eq.5) then
  do 112 i=0,15
        do 132 j=0,15
          if (orga(i,j).eq.0) then
            ch(j)=''
          end if
          if (orga(i,j).eq.1) then
            ch(j)='-'
          end if
          if (orga(i,j).eq.2) then
            ch(j)='x'
          end if
        end if 132 continue
    write (12,122) (ch(j),j=0,15)
  122 format (lx,16a1)
  112 continue
end if
write (12,*) 11
write (12,*) 12
write (12,*) 13
write (12,*) 14
if (k.eq.5) write (12,911) 15
         911 format (lx,1a17)
if ((k.eq.5).eq..false.) write (12,*) 15
write (12,*) 'an n*n coefficient matrix was used with n= ',c

if k=5 print out the next one

if (k.eq.5) then
  do 512 i=0,15
        do 532 j=0,15
          if (orga(i,j).lt.0.5) then
            ch(j)=''
          end if
          if (tp(i,j).eq.1) then
            ch(j)='-'
          end if
          if (tp(i,j).eq.2) then
            ch(j)='x'
          end if
        end if 532 continue
    write (12,522) (ch(j),j=0,15)
  522 format (lx,16a1)
  512 continue
write (12,*) 11
write (12,*) tp11
write (12,*) tp12
write (12,*) tpl3
write (12,*) tpl4
if (k.eq.5) write (12,511) tpl5
511 format (1x,1a17)
if ((k.eq.5).eq..false.) write (12,*) tpl5
write (12,*) 'an n*n coefficient matrix was used with n= ',c
close (12)
end if
end

subroutine afilter(orga,j)
double precision orga(0:15,0:15)
integer j,l,n
  do 120 n=0,15
    do 140 l=j,15
      orga(n,l)= 0
    140 continue
  120 continue
  do 35 n=0,15
    do 45 l=j,15
      orga(l,n)= 0
    45 continue
  35 continue
end

subroutine cmtdct(in,out,q)
double precision in(0:15,0:15),out(0:15,0:15),temp(0:15,0:15)
double precision x(0:15),y(0:15)
integer i,q,j
  do 10 i=0,15
    do 20 j=0,15
      x(j)=in(i,j)
    20 continue
    if (q.eq.1) then
      call mtdct(x,y)
    end if
    if (q.eq.2) then
      call mtidct(x,y)
    end if
    if (q.eq.3) then
      call mtsqdc(x,y)
    end if
    do 25 j=0,15
      temp(i,j)=y(j)
    25 continue
  10 continue
c     write (*,*) 'cmtdct temp(3,7) = ',temp(3,7)
do 40 i= 0,15
do 50 j=0,15
     x(j)= temp(j,i)
50      continue
c     call mtdct(x,y)
if (q.eq.1) then
     call mtdct(x,y)
end if
if (q.eq.2) then
     call mtidct(x,y)
end if
if (q.eq.3) then
     call mtsqdct(x,y)
end if
     do 60 j=0,15
          out(j,i)=y(j)
60      continue
40      continue
     c     write (*,*) 'cmtdct out(3,7) = ',out(3,7)
c
     return
end

subroutine mtdct(x,y)
    integer i,k
    double precision x(0:15),y(0:15)
    double precision g2(16,16,16,16), gl(0:15,0:15)
    common g2,gl
    do 20 k=0,15
         y(k)=0.0
    do 10 i=0,15
         y(k)= y(k) + x(i)*gl(i,k)
10      continue
20      continue
     c     write (*,*) 'mtdct y(3) = ',y(3)
     return
end

subroutine mtidct(x,y)
    integer i,k
    double precision x(16),y(16)
    double precision g2(16,16,16,16), g1(16,16)
    common g2,gl
    do 20 i=l,16
         y(i)=0.0
    do 10 k=l,16
         y(i) = y(i) +x(k)*g1(i,k)
10      continue
20      continue
     c     write (*,*) 'mtidct y(3) = ',y(3)
     return
end
subroutine mtsqdct(x,y)
  integer i,k
  double precision x(16),y(16)
  double precision g2(16,16,16,16), gl(16,16), gsq(16,16)
  common g2,gl,gsq
  do 20 i=l,16
    y(i) = 0.0
  do 10 k=l,16
    y(i) = y(i) +x(k)*gsq(k,i)
  10 continue
  20 continue
  return
end

subroutine relent (in1,in2,out)
  c finds relative entropy between 2 blocks
  double precision in1(16,16), in2(16,16),out
  integer i,j
  out=0.0
  do 10 i=l,16
    do 20 j=l,16
      if(in1(i,j) .lt.0.00001) then
        print *, in1(i,j)
      else
        out= out+ in1(i,j) *log(in1(i,j)/in2(i,j))
      end if
    20 continue
  10 continue
  return
end

from file ddrtni.ftn

subroutine mrereco(p,y,old,n,iter)
  c p= no of coef to reconstruct with
  c sub= input array of coef.
  c reconsu= output recon array
  c n= ibl= image block length (either 4 or 8)
  c procedure for image reconstruction using iterative minimum
  c cross entropy me
double precision f, derf, x, xst, eps, s, y(16, 16)
integer p, k, n, i, ii, iend, iter
double precision ascin(16, 16), old(16, 16), g(16, 16, 16, 16)
common g
character ch80*80

c write (*, *) 'p= ', p
xst=0.0
iend=500
eps = 1.0/10.0**6.0
pie=3.141592654

c write (*, *) 'reconstructing with ', p, ' coef'
k=1
s=float(n)
c
if (iter.eq.1) then
  fac = y(l, l)/s
  call set(old, fac)
end if
do 12 ii=l, iter
do 15 mm=l, p
  write (*, *) next row mm= ', mm
do 16 nn=1, p
  if ( ( not ( mm.eq.1 ) ).or. ( not ( nn.eq.1) ) ) then
    call drtni(x, f, derf, xst, eps, iend, ier, mm, nn, old, y, n)
  write (*, *) 'nn= ', nn, ' x= ', x
  if (ier.ne.0) print*, 'ier=', ier, ier
  do 35 i=l, n
    do 45 j=l, n
      ascin(i, j)= old(i, j)*dexp(x*g(i, j, mm, nn))
  continue
  continue
  call ascale(y, ascin, old, ch80)
  end if
45 continue
15 continue
12 continue

c write (*, *) 'ne mre fini: old(1, 1) = ', old(1, 1)
return
end

subroutine drtni

purpose
to solve general nonlinear equations of the form f(x)=0
by means of newton-s iteration method.

usage
call drtni(x, f, derf, fct, xst, eps, iend, ier, mm, nn)

parameter fct requires an external statement.
description of parameters
x - double precision resultant root of equ f(x)=0
f - double precision resultant function value at root x
derf - double precision resultant value of derivative :aa root x
fct - name of the external subroutine used. it computes
to given argument x function value f and derivative
derf. its parameter list must include x,f,derf, where
all parameters are double precision.
xst - double precision input value which specifies the
intital guess of the root x.
eps - single precision input value which specifies the
upper bound of the error of result x.
iend - max. nu. of iteration steps specified.
ier - resultant error parameter coded as follows
  ier=0 - no error
  ier=1 - no convergence after iend iteration steps
  ier=2 - at any iteration step derivative derf was
       equal to zero.

remarks
the procedure is bypassed and gives the error message ier =2
if at any iteration step derivative of f(x) is equal to 0.
possibly the procedure would be successful if it is started
once more with another initial guess xst.

subroutines and function subprograms required
the external subroutine  fct(x,f,derf,?,,?) must be furnished
by the user.

method
solution of equation f(x)=0 is done by means of newton-s
iteration method, which starts at the initial guess xst of
a root x. convergence is quadratic if the derivative of
f(x) at root x is not equal to zero. one iteration step
requires one evaluation of f(x) and one evaluation of the
derivative of f(x). for test on satisfactory accuracy see
formula (2) of mathematical description.
for reference, see r. zurmuehl, praktische mathematik fuer
ingenieure und physiker, springer, berlin/goettingen/

subroutine drtni(x,f,derf, xst,eps,iend,ierr:mm,nn,sb,y,n)

double precision x,f,derf,xst,eps,dx

double precision b(430,16,16), y(16,16),sb(16,16)
double precision  y(16,16),sb(16,16)
integer i,n,iend,ier:mm,nn
c prepare iteration
  ier = 0
  x=xst
  call fct(x, f, derf, mm, nn, sb, y, n)

  c start iteration loop
  i = 0
  if (f) 1, 7, 1
    equation is not satisfied by x
  1 if (derf) 2, 8, 2

  c iteration is possible
  2 x=x+f/derf
  i= i+1
  x=x-dx
  call fct(x, f, derf, mm, nn, sb, y, n)

  c test on satisfactory accuracy
  if (i.gt.iend) goto 20
  if (dabs(f).gt.eps) goto 1

  c end of iteration loop
  return

  c no convergence after iend iteration steps. error return.
  20 continue
  write(*,*) 'drtni: failed to converge in ',i,' steps'
  ier=1
  7 return

  c error return in case of zero divisor
  8 ier=2
  return
end

subroutine fct(x, f, derf, mm, nn, sb, y, n)
double precision x, f, derf, sum1, sum2
double precision sb(16,16), y(16,16), g(16,16,16,16)
double precision new1(16,16), sb(16,16)
common g
integer n
sum1=0.0
sum2=0.0
  do 10 i=1,n
    do 20 j=1,n
      new1(i,j)= sb(i,j)*dexp(x*g(i,j,mm,nn))*g(i,j,mm,nn)
      sum1=sum1+new1(i,j)
      sum2=sum2+new1(i,j)*g(i,j,mm,nn)
    20 continue
  10 continue
  f=sum1-y(mm,nn)
derf=-sum2
return
end

c subroutine label(str3n0,str3n1,n1,str3n2,n2,out20)
character str3n0*3,str3n1*3,str3n2*3,out20*20
integer n1
write (out20,47) 'd',str3n0,' ',str3n1,' ',n1,' ',str3n2,' ',n2
47 format (a1,a3,a1,a3,a1,i2,a1,a3,a1,i4)
return
end

c subroutine combine(st1,st2,out)
character st1*20,st2*20,out*40
write (out,47) st1,st2
47 format (a20,a20)
return
end

c subroutine drawrect(orga,xl,yl,x2,y2,r)
c makes a xl,yl to x2,y2 rect of orga = to r
c
double precision orga(0:15,0:15),r
integer i,j,xl,yl,x2,y2
do 10 i=xl,x2
   do 20 j=yl ,y2
      orga(i,j)=r
20 continue
10 continue
return
end

c subroutine set(orga,r)
c sets orga = r
c
double precision orga(16,16),r
integer i,j
do 10 i=1,16
   do 20 j=1,16
      orga(i,j) = r
20 continue
10 continue
return
end

c subroutine copy(ina,outa)
do double precision ina(16,16),outa(16,16)
integer i,j
do 10 i=1,16
   do 20 j=1,16
      outa(i,j) = ina(i,j)
20 continue
10 continue
return
end

subroutine xpobl(rela, thres)
double precision rela(0:15,0:15), thres(0:15,0:15)
integer i,j
do 10 i=0,15
do 20 j=0,15
thres(i,j) = dexp(rela(i,j))
continue
10 continue
return
end
APPENDIX C
BINARY TEST IMAGES AND THE GOLDEN GATE BRIDGE

Binary Test Images

Image A5:

-------------
--x------------
---xxx---------
---xx--xx------
-xx---xx--xxxxx-
-xx---xx--xxxxx-
-xxxxxxx--xx----
-xxxxxxx--xx----
-xx---xx--xxxx--
-xx---xx--xxxxx-
-------------xx-
-------------xx-
------------xx--
-----------xxx--
-----------xx---

Image E:

-------------
--xxxxxxxxxx--
---xxxxxxxxx--
---xxxxxxxxx--
---xxx---------
---xxx---------
---xxxxxxxxx--
---xxxxxxxxx--
---xxxxxxxxx--
---xxxxxxxxx--
-------------
-------------
-------------
-------------
-------------
Picture of the Golden Gate Bridge
APPENDIX D

GRAPHS AND TABLES OF ERROR
BETWEEN RECONSTRUCTION AND
ORIGINAL FOR MEP AND IDCT FOR
ALL 4 BINARY IMAGES

Note: The "New0" is simply the "0". The convergence limit is the limit of the error as the Newton non-linear solution method converges to zero.
MSE
Ratio of MEP mse / IDCT mse for image: A5
X-value is number of coefficients retained

Convergence Limit: 1/10**3
Variables: double precision
MSE

Ratio of MEP mse / IDCT mse for image: E
X value is the number of coefficients retained

Convergence Limit: 1/10**3
Variables: double precision
MSE

Ratio of MEP mse / IDCT mse for image: EO
X value is the number of coefficients retained

Convergence Limit: 1/10**3
Variables: double precision
Ratio or MEP mse / IDCT mse for image: New 0
X value is the number of coefficients retained

Convergence Limit: 1/10**3
Variables: double precision
the non-integer variables are: double precision
the MEP will converge to
$1/10^{**6}$
this includes a shift to normalize greyness level after reconstruction.

Image used:

```
-------------
-----x-------
----xxx------
-----xx------
-----xx------
-------xx----
-------xx----
-------xx----
-------xx----
-------xx----
-------xx----
```

<table>
<thead>
<tr>
<th>N</th>
<th>Method</th>
<th>MSE</th>
<th>RE: recn x org</th>
<th>org x recn</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>MEP</td>
<td>7.03829980</td>
<td>17.55254011</td>
<td>16.69795912</td>
</tr>
<tr>
<td>2</td>
<td>idct</td>
<td>7.02090915</td>
<td>17.46160240</td>
<td>16.61680268</td>
</tr>
<tr>
<td>3</td>
<td>MEP</td>
<td>6.32082376</td>
<td>13.97969985</td>
<td>13.87475769</td>
</tr>
<tr>
<td>3</td>
<td>idct</td>
<td>6.36202443</td>
<td>14.20157969</td>
<td>14.10041438</td>
</tr>
<tr>
<td>4</td>
<td>MEP</td>
<td>6.10283319</td>
<td>12.84424271</td>
<td>12.96620230</td>
</tr>
<tr>
<td>4</td>
<td>idct</td>
<td>6.10544076</td>
<td>12.88445234</td>
<td>13.01326134</td>
</tr>
<tr>
<td>5</td>
<td>MEP</td>
<td>5.74751512</td>
<td>11.31845586</td>
<td>11.62258414</td>
</tr>
<tr>
<td>5</td>
<td>idct</td>
<td>5.76460163</td>
<td>11.42050742</td>
<td>11.74303887</td>
</tr>
<tr>
<td>6</td>
<td>MEP</td>
<td>5.53994606</td>
<td>10.4730389</td>
<td>10.88036800</td>
</tr>
<tr>
<td>6</td>
<td>idct</td>
<td>5.56642803</td>
<td>10.61233813</td>
<td>11.02811602</td>
</tr>
<tr>
<td>7</td>
<td>MEP</td>
<td>5.14235868</td>
<td>9.06846993</td>
<td>9.45839005</td>
</tr>
<tr>
<td>7</td>
<td>idct</td>
<td>5.21114184</td>
<td>9.38685482</td>
<td>9.81343355</td>
</tr>
<tr>
<td>8</td>
<td>MEP</td>
<td>3.56399912</td>
<td>4.45828612</td>
<td>4.49391813</td>
</tr>
<tr>
<td>8</td>
<td>idct</td>
<td>3.68085725</td>
<td>4.97073488</td>
<td>5.08986580</td>
</tr>
<tr>
<td>9</td>
<td>MEP</td>
<td>3.04125694</td>
<td>3.25320072</td>
<td>3.29729913</td>
</tr>
<tr>
<td>N</td>
<td>Method</td>
<td>MSE</td>
<td>RE: recn x org</td>
<td>org x recn</td>
</tr>
<tr>
<td>----</td>
<td>--------</td>
<td>--------</td>
<td>------------------</td>
<td>------------</td>
</tr>
<tr>
<td>9</td>
<td>idct</td>
<td>3.17117118</td>
<td>3.79406342</td>
<td>3.87543085</td>
</tr>
<tr>
<td>10</td>
<td>MEP</td>
<td>2.58887596</td>
<td>2.31295558</td>
<td>2.32669551</td>
</tr>
<tr>
<td>10</td>
<td>idct</td>
<td>2.66266656</td>
<td>2.78152339</td>
<td>2.77142393</td>
</tr>
<tr>
<td>11</td>
<td>MEP</td>
<td>2.08976767</td>
<td>1.51229971</td>
<td>1.53419167</td>
</tr>
<tr>
<td>11</td>
<td>idct</td>
<td>2.17505877</td>
<td>1.87922527</td>
<td>1.89026210</td>
</tr>
<tr>
<td>12</td>
<td>MEP</td>
<td>1.68077699</td>
<td>0.98254266</td>
<td>0.98713372</td>
</tr>
<tr>
<td>12</td>
<td>idct</td>
<td>1.74112547</td>
<td>1.18964096</td>
<td>1.18847816</td>
</tr>
<tr>
<td>13</td>
<td>MEP</td>
<td>1.47311918</td>
<td>0.76261123</td>
<td>0.76289828</td>
</tr>
<tr>
<td>13</td>
<td>idct</td>
<td>1.53459202</td>
<td>0.92121287</td>
<td>0.91844357</td>
</tr>
<tr>
<td>14</td>
<td>MEP</td>
<td>1.19701624</td>
<td>0.54595815</td>
<td>0.54613127</td>
</tr>
<tr>
<td>14</td>
<td>idct</td>
<td>1.33268600</td>
<td>0.71199815</td>
<td>0.71072539</td>
</tr>
<tr>
<td>15</td>
<td>MEP</td>
<td>0.81654311</td>
<td>0.24653073</td>
<td>0.24660839</td>
</tr>
<tr>
<td>15</td>
<td>idct</td>
<td>0.87922835</td>
<td>0.29691391</td>
<td>0.29694794</td>
</tr>
<tr>
<td>16</td>
<td>MEP</td>
<td>0.00060147</td>
<td>0.00000011</td>
<td>0.00000011</td>
</tr>
<tr>
<td>16</td>
<td>idct</td>
<td>0.00000651</td>
<td>0.00000000</td>
<td>0.00000000</td>
</tr>
</tbody>
</table>
the non-integer variables are: double precision
the MEP will converge to
1/10**6

this includes a shift to normalize greyness
level after reconstruction.

Image used:

---xxxxxxxxx----
---xxxxxxxxx----
---xxxx--------
---xxx---------
---xxxx--------
---xxxxxxxxx----
---xxxxxxxxx----
---xxxxxxxxx----
---xxxx--------
---xxx---------
---xxxxxxxxx----
---xxxxxxxxx----
---xxxxxxxxx----

(@egin table header
Number of Coef. image: E
Retained
NxN

<table>
<thead>
<tr>
<th>N</th>
<th>Method</th>
<th>MSE</th>
<th>RE: recn x org</th>
<th>org x recn</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>MEP</td>
<td>7.42647217</td>
<td>19.41535038</td>
<td>18.75045136</td>
</tr>
<tr>
<td>2</td>
<td>idct</td>
<td>7.41426495</td>
<td>19.34781578</td>
<td>18.68809214</td>
</tr>
<tr>
<td>3</td>
<td>MEP</td>
<td>6.24295589</td>
<td>13.17777392</td>
<td>13.58772884</td>
</tr>
<tr>
<td>3</td>
<td>idct</td>
<td>6.19361329</td>
<td>13.01751550</td>
<td>13.45375296</td>
</tr>
<tr>
<td>4</td>
<td>MEP</td>
<td>6.12206731</td>
<td>12.62029543</td>
<td>12.91067104</td>
</tr>
<tr>
<td>4</td>
<td>idct</td>
<td>6.02158157</td>
<td>12.22882584</td>
<td>12.53840961</td>
</tr>
<tr>
<td>5</td>
<td>MEP</td>
<td>5.08777726</td>
<td>8.91046625</td>
<td>9.12265575</td>
</tr>
<tr>
<td>5</td>
<td>idct</td>
<td>5.13099455</td>
<td>9.13802366</td>
<td>9.40274754</td>
</tr>
<tr>
<td>6</td>
<td>MEP</td>
<td>4.95422327</td>
<td>8.49376116</td>
<td>8.71753859</td>
</tr>
<tr>
<td>6</td>
<td>idct</td>
<td>5.03315263</td>
<td>8.85517599</td>
<td>9.14595402</td>
</tr>
<tr>
<td>7</td>
<td>MEP</td>
<td>3.52056939</td>
<td>4.40051993</td>
<td>4.44646187</td>
</tr>
<tr>
<td>7</td>
<td>idct</td>
<td>3.68530018</td>
<td>5.07184480</td>
<td>5.18721664</td>
</tr>
<tr>
<td>8</td>
<td>MEP</td>
<td>3.46991543</td>
<td>4.25643046</td>
<td>4.26109111</td>
</tr>
<tr>
<td>8</td>
<td>idct</td>
<td>3.59124497</td>
<td>4.77561921</td>
<td>4.82475678</td>
</tr>
<tr>
<td>9</td>
<td>MEP</td>
<td>2.75246215</td>
<td>2.37331856</td>
<td>2.38960168</td>
</tr>
<tr>
<td>9</td>
<td>idct</td>
<td>2.54684581</td>
<td>2.18686439</td>
<td>2.21766550</td>
</tr>
<tr>
<td>10</td>
<td>MEP</td>
<td>2.65265636</td>
<td>2.10727950</td>
<td>2.12197931</td>
</tr>
<tr>
<td>10</td>
<td>idct</td>
<td>2.35564268</td>
<td>1.84137701</td>
<td>1.85665896</td>
</tr>
<tr>
<td>11</td>
<td>MEP</td>
<td>2.38662019</td>
<td>1.78299641</td>
<td>1.79715849</td>
</tr>
<tr>
<td>11</td>
<td>idct</td>
<td>2.18544627</td>
<td>1.56552610</td>
<td>1.57714640</td>
</tr>
<tr>
<td>12</td>
<td>MEP</td>
<td>2.23384062</td>
<td>1.65980487</td>
<td>1.67251826</td>
</tr>
<tr>
<td>12</td>
<td>idct</td>
<td>2.16548059</td>
<td>1.60193204</td>
<td>1.61282084</td>
</tr>
<tr>
<td>13</td>
<td>MEP</td>
<td>1.79555222</td>
<td>1.10852040</td>
<td>1.11308786</td>
</tr>
<tr>
<td>13</td>
<td>idct</td>
<td>1.80048554</td>
<td>1.15481731</td>
<td>1.15841623</td>
</tr>
<tr>
<td>14</td>
<td>MEP</td>
<td>1.68070209</td>
<td>0.90047981</td>
<td>0.90600541</td>
</tr>
<tr>
<td>14</td>
<td>idct</td>
<td>1.55860038</td>
<td>0.78998025</td>
<td>0.79487563</td>
</tr>
<tr>
<td>15</td>
<td>MEP</td>
<td>0.88111794</td>
<td>0.24609731</td>
<td>0.24621556</td>
</tr>
<tr>
<td>15</td>
<td>idct</td>
<td>0.82383574</td>
<td>0.23621649</td>
<td>0.23623482</td>
</tr>
<tr>
<td>16</td>
<td>MEP</td>
<td>0.00068268</td>
<td>0.00000015</td>
<td>0.00000015</td>
</tr>
<tr>
<td>16</td>
<td>idct</td>
<td>0.00000566</td>
<td>0.00000000</td>
<td>0.00000000</td>
</tr>
</tbody>
</table>
the non-integer variables are: double precision
the MEP will converge to
$1/10^{6}$
this includes a shift to normalize greyness
level after reconstruction.

Image used:

```
----------xxxxx-
----------xxxxx-
----------xx----
----------xx----
----------xxxxx-
----x-----xxxxx-
---xxx----xx----
--xx-xx---xx----
--x---x---xxxxx-
-xx---xx--xxxxx-
--x---x---------
--xx-xx---------
---xxx----------
----x-----------
```

<table>
<thead>
<tr>
<th>Number</th>
<th>Coef.</th>
<th>Retained</th>
</tr>
</thead>
<tbody>
<tr>
<td>N x N</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>N</th>
<th>Method</th>
<th>MSE</th>
<th>RE: recn x org</th>
<th>org x recn</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>MEP</td>
<td>6.53362235</td>
<td>15.18025775</td>
<td>14.33138122</td>
</tr>
<tr>
<td>2</td>
<td>idct</td>
<td>6.51773649</td>
<td>15.11009328</td>
<td>14.27189510</td>
</tr>
<tr>
<td>3</td>
<td>MEP</td>
<td>6.26661886</td>
<td>13.82616926</td>
<td>13.39508712</td>
</tr>
<tr>
<td>3</td>
<td>idct</td>
<td>6.26824483</td>
<td>13.84590611</td>
<td>13.41555343</td>
</tr>
<tr>
<td>4</td>
<td>MEP</td>
<td>5.99221136</td>
<td>12.62895147</td>
<td>12.41981804</td>
</tr>
<tr>
<td>4</td>
<td>idct</td>
<td>6.03331887</td>
<td>12.83409744</td>
<td>12.61611859</td>
</tr>
<tr>
<td>5</td>
<td>MEP</td>
<td>5.46855199</td>
<td>10.29525531</td>
<td>10.40675203</td>
</tr>
<tr>
<td>5</td>
<td>idct</td>
<td>5.46868138</td>
<td>10.34249012</td>
<td>10.46485075</td>
</tr>
<tr>
<td>6</td>
<td>MEP</td>
<td>5.26218285</td>
<td>9.50412345</td>
<td>9.69595757</td>
</tr>
<tr>
<td>6</td>
<td>idct</td>
<td>5.27142147</td>
<td>9.57383575</td>
<td>9.78700400</td>
</tr>
<tr>
<td>7</td>
<td>MEP</td>
<td>4.83970506</td>
<td>8.08462452</td>
<td>8.23188319</td>
</tr>
<tr>
<td>7</td>
<td>idct</td>
<td>4.88013297</td>
<td>8.28491321</td>
<td>8.48429076</td>
</tr>
<tr>
<td>8</td>
<td>MEP</td>
<td>4.32590819</td>
<td>6.56474977</td>
<td>6.70376881</td>
</tr>
<tr>
<td>8</td>
<td>idct</td>
<td>4.44467868</td>
<td>7.00792825</td>
<td>7.20123445</td>
</tr>
<tr>
<td>9</td>
<td>MEP</td>
<td>3.05199053</td>
<td>3.39905102</td>
<td>3.50567841</td>
</tr>
<tr>
<td>N</td>
<td>Method</td>
<td>MSE</td>
<td>RE: recn x org</td>
<td>org x recn</td>
</tr>
<tr>
<td>----</td>
<td>--------</td>
<td>-----------</td>
<td>----------------</td>
<td>-----------</td>
</tr>
<tr>
<td>9</td>
<td>idct</td>
<td>3.29917863</td>
<td>4.15522499</td>
<td>4.27066004</td>
</tr>
<tr>
<td>10</td>
<td>MEP</td>
<td>2.59946370</td>
<td>2.50060912</td>
<td>2.58259891</td>
</tr>
<tr>
<td>10</td>
<td>idct</td>
<td>2.86602679</td>
<td>3.21878528</td>
<td>3.31853367</td>
</tr>
<tr>
<td>11</td>
<td>MEP</td>
<td>2.19884963</td>
<td>1.78938969</td>
<td>1.83514347</td>
</tr>
<tr>
<td>11</td>
<td>idct</td>
<td>2.42849988</td>
<td>2.35734738</td>
<td>2.41158263</td>
</tr>
<tr>
<td>12</td>
<td>MEP</td>
<td>1.78998806</td>
<td>1.19386106</td>
<td>1.21747120</td>
</tr>
<tr>
<td>12</td>
<td>idct</td>
<td>1.99396027</td>
<td>1.62459983</td>
<td>1.65523208</td>
</tr>
<tr>
<td>13</td>
<td>MEP</td>
<td>1.64603270</td>
<td>1.02621158</td>
<td>1.03881964</td>
</tr>
<tr>
<td>13</td>
<td>idct</td>
<td>1.85723033</td>
<td>1.44881061</td>
<td>1.45326012</td>
</tr>
<tr>
<td>14</td>
<td>MEP</td>
<td>1.42568738</td>
<td>0.82996977</td>
<td>0.84293201</td>
</tr>
<tr>
<td>14</td>
<td>idct</td>
<td>1.71120174</td>
<td>1.25510071</td>
<td>1.26439450</td>
</tr>
<tr>
<td>15</td>
<td>MEP</td>
<td>1.07012494</td>
<td>0.48566309</td>
<td>0.48892441</td>
</tr>
<tr>
<td>15</td>
<td>idct</td>
<td>1.32209837</td>
<td>0.75993542</td>
<td>0.76331346</td>
</tr>
<tr>
<td>16</td>
<td>MEP</td>
<td>0.00071602</td>
<td>0.00000016</td>
<td>0.00000016</td>
</tr>
<tr>
<td>16</td>
<td>idct</td>
<td>0.00000634</td>
<td>0.00000000</td>
<td>0.00000000</td>
</tr>
</tbody>
</table>
the non-integer variables are: double precision
the MEP will converge to
$1/10^{6}$
this includes a shift to normalize greyness level after reconstruction.

Image used:

```
----------xx----
----------
----------
----------
-------xx----xx-
-------x------x-
-------xx----xx-
--------xxxxxx--
----------xx----
```

<table>
<thead>
<tr>
<th>Number of Coef. Retained</th>
<th>image: NewO</th>
</tr>
</thead>
<tbody>
<tr>
<td>NxN</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>N</th>
<th>Method</th>
<th>MSE</th>
<th>RE: recn x org</th>
<th>org x recn</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>MEP</td>
<td>4.61570972</td>
<td>7.68271919</td>
<td>7.02134043</td>
</tr>
<tr>
<td>2</td>
<td>idct</td>
<td>4.59666398</td>
<td>7.61426789</td>
<td>6.96593476</td>
</tr>
<tr>
<td>3</td>
<td>MEP</td>
<td>4.45878877</td>
<td>7.15044939</td>
<td>6.68926407</td>
</tr>
<tr>
<td>3</td>
<td>idct</td>
<td>4.47877368</td>
<td>7.23118480</td>
<td>6.75682492</td>
</tr>
<tr>
<td>4</td>
<td>MEP</td>
<td>4.31022707</td>
<td>6.56972896</td>
<td>6.31289914</td>
</tr>
<tr>
<td>4</td>
<td>idct</td>
<td>4.31262126</td>
<td>6.58345621</td>
<td>6.32421455</td>
</tr>
<tr>
<td>5</td>
<td>MEP</td>
<td>4.11673428</td>
<td>5.93538239</td>
<td>5.83127486</td>
</tr>
<tr>
<td>5</td>
<td>idct</td>
<td>4.12675073</td>
<td>5.99244636</td>
<td>5.88027436</td>
</tr>
<tr>
<td>6</td>
<td>MEP</td>
<td>4.04312391</td>
<td>5.78573834</td>
<td>5.68628590</td>
</tr>
<tr>
<td>6</td>
<td>idct</td>
<td>4.09590764</td>
<td>5.96288290</td>
<td>5.85228764</td>
</tr>
<tr>
<td>7</td>
<td>MEP</td>
<td>3.21032272</td>
<td>3.71663728</td>
<td>3.82409522</td>
</tr>
<tr>
<td>7</td>
<td>idct</td>
<td>3.39773091</td>
<td>4.22096877</td>
<td>4.32565759</td>
</tr>
<tr>
<td>8</td>
<td>MEP</td>
<td>2.75799320</td>
<td>2.83530838</td>
<td>2.92334164</td>
</tr>
<tr>
<td>8</td>
<td>idct</td>
<td>3.02608006</td>
<td>3.49304366</td>
<td>3.57760520</td>
</tr>
<tr>
<td>9</td>
<td>MEP</td>
<td>2.75670635</td>
<td>2.83436082</td>
<td>2.92274056</td>
</tr>
<tr>
<td>N</td>
<td>Method</td>
<td>MSE</td>
<td>RE: recn x org</td>
<td>org x recn</td>
</tr>
<tr>
<td>----</td>
<td>--------</td>
<td>--------</td>
<td>----------------</td>
<td>------------</td>
</tr>
<tr>
<td>9</td>
<td>idct</td>
<td>3.02608006</td>
<td>3.49304366</td>
<td>3.57760520</td>
</tr>
<tr>
<td>10</td>
<td>MEP</td>
<td>2.28969895</td>
<td>2.03126447</td>
<td>2.12097045</td>
</tr>
<tr>
<td>10</td>
<td>idct</td>
<td>2.62445554</td>
<td>2.68400454</td>
<td>2.80450370</td>
</tr>
<tr>
<td>11</td>
<td>MEP</td>
<td>1.65508925</td>
<td>1.13067732</td>
<td>1.15712482</td>
</tr>
<tr>
<td>11</td>
<td>idct</td>
<td>2.00816666</td>
<td>1.70314425</td>
<td>1.73972000</td>
</tr>
<tr>
<td>12</td>
<td>MEP</td>
<td>1.64865002</td>
<td>1.11332245</td>
<td>1.13856183</td>
</tr>
<tr>
<td>12</td>
<td>idct</td>
<td>1.98458733</td>
<td>1.65447229</td>
<td>1.68694463</td>
</tr>
<tr>
<td>13</td>
<td>MEP</td>
<td>1.35157099</td>
<td>0.79083811</td>
<td>0.79716533</td>
</tr>
<tr>
<td>13</td>
<td>idct</td>
<td>1.70321000</td>
<td>1.27702900</td>
<td>1.28587638</td>
</tr>
<tr>
<td>14</td>
<td>MEP</td>
<td>0.54470436</td>
<td>0.14302950</td>
<td>0.14287747</td>
</tr>
<tr>
<td>14</td>
<td>idct</td>
<td>0.75788227</td>
<td>0.27840163</td>
<td>0.27760304</td>
</tr>
<tr>
<td>15</td>
<td>MEP</td>
<td>0.39325096</td>
<td>0.06899315</td>
<td>0.06900538</td>
</tr>
<tr>
<td>15</td>
<td>idct</td>
<td>0.50762974</td>
<td>0.11626281</td>
<td>0.11617604</td>
</tr>
<tr>
<td>16</td>
<td>MEP</td>
<td>0.00043888</td>
<td>0.00000006</td>
<td>0.00000006</td>
</tr>
<tr>
<td>16</td>
<td>idct</td>
<td>0.00000471</td>
<td>0.00000000</td>
<td>0.00000000</td>
</tr>
</tbody>
</table>
APPENDIX E
MEASUREMENT OF THE DISTANCES BETWEEN RECONSTRUCTIONS FOR THE IMAGE A5

rel 3 refers to the 3-pass MREP

RE1 of IDCT/ORG means \( \ln \frac{\text{IDCT}(i)}{\text{ORG}(i)} \)

RE2 of IDCT/ORG = RE1 of ORG/IDCT

Image: A5
Convergence limit = \(1/10^9\) for MEP and \(1/10^6\) for MREP

<table>
<thead>
<tr>
<th>METHOD</th>
<th>MSE</th>
<th>RE1</th>
<th>RE2</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDCT/ORG</td>
<td>0.00000651</td>
<td>0.00000000</td>
<td>0.00000000</td>
</tr>
<tr>
<td>MEP/ORG</td>
<td>0.00001473</td>
<td>0.00000000</td>
<td>0.00000000</td>
</tr>
<tr>
<td>MEP/IDCT</td>
<td>0.00001890</td>
<td>0.00000000</td>
<td>0.00000000</td>
</tr>
<tr>
<td>rel 1/ORG</td>
<td>1.03763134</td>
<td>0.40061775</td>
<td>0.39760002</td>
</tr>
<tr>
<td>rel 1/IDC</td>
<td>1.03763340</td>
<td>0.40061888</td>
<td>0.39760118</td>
</tr>
<tr>
<td>rel 1/MEP</td>
<td>1.03762859</td>
<td>0.40061571</td>
<td>0.39759796</td>
</tr>
<tr>
<td>rel 2/ORG</td>
<td>0.21942138</td>
<td>0.01650705</td>
<td>0.01652786</td>
</tr>
<tr>
<td>rel 2/IDC</td>
<td>0.21942515</td>
<td>0.01650760</td>
<td>0.01652840</td>
</tr>
<tr>
<td>rel 2/MEP</td>
<td>0.21941921</td>
<td>0.01650676</td>
<td>0.01652757</td>
</tr>
<tr>
<td>rel 3/ORG</td>
<td>0.05070269</td>
<td>0.00110973</td>
<td>0.00111136</td>
</tr>
<tr>
<td>rel 3/IDC</td>
<td>0.05070276</td>
<td>0.00110974</td>
<td>0.00111137</td>
</tr>
<tr>
<td>rel 3/MEP</td>
<td>0.05070291</td>
<td>0.00110975</td>
<td>0.00111138</td>
</tr>
<tr>
<td>rel 4/ORG</td>
<td>0.01269510</td>
<td>0.00006108</td>
<td>0.00006109</td>
</tr>
<tr>
<td>rel 4/IDC</td>
<td>0.01269267</td>
<td>0.00006106</td>
<td>0.00006107</td>
</tr>
<tr>
<td>rel 4/MEP</td>
<td>0.01269635</td>
<td>0.00006109</td>
<td>0.00006110</td>
</tr>
<tr>
<td>rel 5/ORG</td>
<td>0.00275554</td>
<td>0.00000322</td>
<td>0.00000322</td>
</tr>
<tr>
<td>rel 5/IDC</td>
<td>0.00275322</td>
<td>0.00000322</td>
<td>0.00000322</td>
</tr>
<tr>
<td>rel 5/MEP</td>
<td>0.00275694</td>
<td>0.00000322</td>
<td>0.00000322</td>
</tr>
<tr>
<td>rel 6/ORG</td>
<td>0.00077066</td>
<td>0.00000026</td>
<td>0.00000026</td>
</tr>
<tr>
<td>rel 6/IDC</td>
<td>0.00077059</td>
<td>0.00000026</td>
<td>0.00000026</td>
</tr>
<tr>
<td>rel 6/MEP</td>
<td>0.00077166</td>
<td>0.00000026</td>
<td>0.00000026</td>
</tr>
<tr>
<td>rel 7/ORG</td>
<td>0.00021295</td>
<td>0.00000002</td>
<td>0.00000002</td>
</tr>
<tr>
<td>rel 7/IDC</td>
<td>0.00021484</td>
<td>0.00000002</td>
<td>0.00000002</td>
</tr>
<tr>
<td>rel 7/MEP</td>
<td>0.00021306</td>
<td>0.00000002</td>
<td>0.00000002</td>
</tr>
<tr>
<td>Relation</td>
<td>IDCT/ORG</td>
<td>MEP/ORG</td>
<td>MEP/IDCT</td>
</tr>
<tr>
<td>----------</td>
<td>----------</td>
<td>---------</td>
<td>----------</td>
</tr>
<tr>
<td>8/ORG</td>
<td>0.81651531</td>
<td>0.00000000</td>
<td>0.00000000</td>
</tr>
<tr>
<td>8/IDC</td>
<td>0.81651531</td>
<td>0.00000000</td>
<td>0.00000000</td>
</tr>
<tr>
<td>8/MEP</td>
<td>0.81651531</td>
<td>0.00000000</td>
<td>0.00000000</td>
</tr>
<tr>
<td>9/ORG</td>
<td>0.81651531</td>
<td>0.00000000</td>
<td>0.00000000</td>
</tr>
<tr>
<td>9/IDC</td>
<td>0.81651531</td>
<td>0.00000000</td>
<td>0.00000000</td>
</tr>
<tr>
<td>9/MEP</td>
<td>0.81651531</td>
<td>0.00000000</td>
<td>0.00000000</td>
</tr>
<tr>
<td>10/ORG</td>
<td>0.81651531</td>
<td>0.00000000</td>
<td>0.00000000</td>
</tr>
<tr>
<td>10/IDC</td>
<td>0.81651531</td>
<td>0.00000000</td>
<td>0.00000000</td>
</tr>
<tr>
<td>10/MEP</td>
<td>0.81651531</td>
<td>0.00000000</td>
<td>0.00000000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Relation</th>
<th>IDCT/ORG</th>
<th>MEP/ORG</th>
<th>MEP/IDCT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/ORG</td>
<td>1.29840862</td>
<td>0.62333789</td>
<td>0.61757791</td>
</tr>
<tr>
<td>1/IDC</td>
<td>1.01651895</td>
<td>0.38111109</td>
<td>0.37869482</td>
</tr>
<tr>
<td>1/MEP</td>
<td>1.00803508</td>
<td>0.37376299</td>
<td>0.37104740</td>
</tr>
<tr>
<td>2/ORG</td>
<td>0.83825212</td>
<td>0.26097953</td>
<td>0.26116281</td>
</tr>
<tr>
<td>2/IDC</td>
<td>0.24181764</td>
<td>0.02222242</td>
<td>0.02227911</td>
</tr>
<tr>
<td>2/MEP</td>
<td>0.20328875</td>
<td>0.01461039</td>
<td>0.01463393</td>
</tr>
<tr>
<td>3/ORG</td>
<td>0.81701481</td>
<td>0.81701481</td>
<td>0.00865944</td>
</tr>
<tr>
<td>3/IDC</td>
<td>0.13952738</td>
<td>0.00866355</td>
<td>0.00865944</td>
</tr>
<tr>
<td>3/MEP</td>
<td>0.04867307</td>
<td>0.00101334</td>
<td>0.00101474</td>
</tr>
<tr>
<td>4/ORG</td>
<td>0.81672295</td>
<td>0.24665722</td>
<td>0.24657988</td>
</tr>
<tr>
<td>4/IDC</td>
<td>0.13033496</td>
<td>0.00770181</td>
<td>0.00769566</td>
</tr>
<tr>
<td>4/MEP</td>
<td>0.01162809</td>
<td>0.00005095</td>
<td>0.00005096</td>
</tr>
<tr>
<td>5/ORG</td>
<td>0.81659099</td>
<td>0.24661375</td>
<td>0.24653157</td>
</tr>
<tr>
<td>5/IDC</td>
<td>0.12978652</td>
<td>0.00765132</td>
<td>0.00764737</td>
</tr>
<tr>
<td>5/MEP</td>
<td>0.00247682</td>
<td>0.00000265</td>
<td>0.00000265</td>
</tr>
<tr>
<td>6/ORG</td>
<td>0.81652090</td>
<td>0.24660935</td>
<td>0.24652913</td>
</tr>
<tr>
<td>6/IDC</td>
<td>0.12979264</td>
<td>0.00764836</td>
<td>0.00764493</td>
</tr>
<tr>
<td>6/MEP</td>
<td>0.00069369</td>
<td>0.00000021</td>
<td>0.00000021</td>
</tr>
<tr>
<td>7/ORG</td>
<td>0.81651299</td>
<td>0.24660868</td>
<td>0.24652894</td>
</tr>
<tr>
<td>7/IDC</td>
<td>0.12979684</td>
<td>0.00764818</td>
<td>0.00764474</td>
</tr>
<tr>
<td>7/MEP</td>
<td>0.00018128</td>
<td>0.0000001</td>
<td>0.00000001</td>
</tr>
<tr>
<td>8/ORG</td>
<td>0.81651531</td>
<td>0.24660867</td>
<td>0.24652892</td>
</tr>
<tr>
<td>8/IDC</td>
<td>0.12979650</td>
<td>0.00764820</td>
<td>0.00764473</td>
</tr>
<tr>
<td>8/MEP</td>
<td>0.00047000</td>
<td>0.00000000</td>
<td>0.00000000</td>
</tr>
<tr>
<td>9/ORG</td>
<td>0.81651629</td>
<td>0.24660867</td>
<td>0.24652892</td>
</tr>
<tr>
<td>9/IDC</td>
<td>0.12979614</td>
<td>0.00764821</td>
<td>0.00764473</td>
</tr>
<tr>
<td>9/MEP</td>
<td>0.0002052</td>
<td>0.00000000</td>
<td>0.00000000</td>
</tr>
<tr>
<td>10/ORG</td>
<td>0.81651636</td>
<td>0.24660867</td>
<td>0.24652892</td>
</tr>
<tr>
<td>10/IDC</td>
<td>0.12979609</td>
<td>0.00764821</td>
<td>0.00764473</td>
</tr>
<tr>
<td>10/MEP</td>
<td>0.0001668</td>
<td>0.00000000</td>
<td>0.00000000</td>
</tr>
<tr>
<td>IDCT/ORG</td>
<td>1.33268600</td>
<td>0.71072539</td>
<td>0.71199815</td>
</tr>
<tr>
<td>MEP/ORG</td>
<td>1.19699510</td>
<td>0.54613265</td>
<td>0.54595441</td>
</tr>
<tr>
<td>MEP/IDCT</td>
<td>0.24030941</td>
<td>0.02568248</td>
<td>0.02561445</td>
</tr>
<tr>
<td>rel 1/ORG</td>
<td>1.50905440</td>
<td>0.88073178</td>
<td>0.86845742</td>
</tr>
<tr>
<td>rel 1/IDC</td>
<td>0.96548334</td>
<td>0.34856069</td>
<td>0.34811867</td>
</tr>
<tr>
<td>rel 1/MEP</td>
<td>0.93350907</td>
<td>0.32362576</td>
<td>0.32250158</td>
</tr>
<tr>
<td>rel 2/ORG</td>
<td>1.20093295</td>
<td>0.55540888</td>
<td>0.55695840</td>
</tr>
<tr>
<td>rel 2/IDC</td>
<td>0.30603676</td>
<td>0.03654588</td>
<td>0.03661904</td>
</tr>
<tr>
<td>rel 2/MEP</td>
<td>0.17563390</td>
<td>0.01099327</td>
<td>0.01100470</td>
</tr>
<tr>
<td>rel 3/ORG</td>
<td>1.19685483</td>
<td>0.54637599</td>
<td>0.54664116</td>
</tr>
<tr>
<td>rel 3/IDC</td>
<td>0.24470610</td>
<td>0.02635486</td>
<td>0.02630138</td>
</tr>
<tr>
<td>rel 3/MEP</td>
<td>0.04087351</td>
<td>0.00068639</td>
<td>0.00068697</td>
</tr>
<tr>
<td>rel 4/ORG</td>
<td>1.19733956</td>
<td>0.54613499</td>
<td>0.54598154</td>
</tr>
<tr>
<td>rel 4/IDC</td>
<td>0.24013321</td>
<td>0.02571239</td>
<td>0.02564173</td>
</tr>
<tr>
<td>rel 4/MEP</td>
<td>0.00847850</td>
<td>0.00002726</td>
<td>0.00002726</td>
</tr>
<tr>
<td>rel 5/ORG</td>
<td>1.19708184</td>
<td>0.54614332</td>
<td>0.54595353</td>
</tr>
<tr>
<td>rel 5/IDC</td>
<td>0.24021440</td>
<td>0.02568437</td>
<td>0.02561555</td>
</tr>
<tr>
<td>rel 5/MEP</td>
<td>0.00156517</td>
<td>0.0000107</td>
<td>0.0000107</td>
</tr>
<tr>
<td>rel 6/ORG</td>
<td>1.19699543</td>
<td>0.54613527</td>
<td>0.54595428</td>
</tr>
<tr>
<td>rel 6/IDC</td>
<td>0.24030696</td>
<td>0.02568261</td>
<td>0.02561455</td>
</tr>
<tr>
<td>rel 6/MEP</td>
<td>0.00039963</td>
<td>0.00000007</td>
<td>0.00000007</td>
</tr>
<tr>
<td>rel 7/ORG</td>
<td>1.19698913</td>
<td>0.54613221</td>
<td>0.54595428</td>
</tr>
<tr>
<td>rel 7/IDC</td>
<td>0.24031355</td>
<td>0.02568246</td>
<td>0.02561449</td>
</tr>
<tr>
<td>rel 7/MEP</td>
<td>0.0009486</td>
<td>0.00000000</td>
<td>0.00000000</td>
</tr>
<tr>
<td>rel 8/ORG</td>
<td>1.19697919</td>
<td>0.54613193</td>
<td>0.54595428</td>
</tr>
<tr>
<td>rel 8/IDC</td>
<td>0.24031095</td>
<td>0.02568248</td>
<td>0.02561448</td>
</tr>
<tr>
<td>rel 8/MEP</td>
<td>0.00030401</td>
<td>0.00000000</td>
<td>0.00000000</td>
</tr>
<tr>
<td>rel 9/ORG</td>
<td>1.19699286</td>
<td>0.54613200</td>
<td>0.54595428</td>
</tr>
<tr>
<td>rel 9/IDC</td>
<td>0.24031000</td>
<td>0.02568249</td>
<td>0.02561448</td>
</tr>
<tr>
<td>rel 9/MEP</td>
<td>0.00022462</td>
<td>0.00000000</td>
<td>0.00000000</td>
</tr>
<tr>
<td>rel 10/ORG</td>
<td>1.19699292</td>
<td>0.54613203</td>
<td>0.54595428</td>
</tr>
<tr>
<td>rel 10/IDC</td>
<td>0.24030993</td>
<td>0.02568249</td>
<td>0.02561448</td>
</tr>
<tr>
<td>rel 10/MEP</td>
<td>0.00002432</td>
<td>0.00000000</td>
<td>0.00000000</td>
</tr>
<tr>
<td>IDCT/ORG</td>
<td>1.53459202</td>
<td>0.91844357</td>
<td>0.92121287</td>
</tr>
<tr>
<td>MEP/ORG</td>
<td>1.47308789</td>
<td>0.76298957</td>
<td>0.76260831</td>
</tr>
<tr>
<td>MEP/IDCT</td>
<td>0.35111643</td>
<td>0.04900938</td>
<td>0.04870938</td>
</tr>
<tr>
<td>rel 1/ORG</td>
<td>1.72610107</td>
<td>1.08547402</td>
<td>1.06771285</td>
</tr>
<tr>
<td>rel 1/IDC</td>
<td>0.98131218</td>
<td>0.35423241</td>
<td>0.35381503</td>
</tr>
<tr>
<td>rel 1/MEP</td>
<td>0.90953643</td>
<td>0.30585226</td>
<td>0.30510443</td>
</tr>
<tr>
<td>rel 2/ORG</td>
<td>1.47641425</td>
<td>0.77114591</td>
<td>0.77265722</td>
</tr>
<tr>
<td>rel 2/IDC</td>
<td>0.39866770</td>
<td>0.05887913</td>
<td>0.05875872</td>
</tr>
<tr>
<td>rel 2/MEP</td>
<td>0.16918666</td>
<td>0.01004538</td>
<td>0.01004942</td>
</tr>
<tr>
<td>rel 3/ORG</td>
<td>1.47243826</td>
<td>0.76284795</td>
<td>0.76322586</td>
</tr>
<tr>
<td>rel 3/IDC</td>
<td>0.35391766</td>
<td>0.04959869</td>
<td>0.04932697</td>
</tr>
<tr>
<td>rel 3/MEP</td>
<td>0.03936991</td>
<td>0.03936991</td>
<td>0.00061719</td>
</tr>
<tr>
<td>rel 4/ORG</td>
<td>1.47327085</td>
<td>0.76284723</td>
<td>0.76263223</td>
</tr>
<tr>
<td>rel 4/IDC</td>
<td>0.35096703</td>
<td>0.04903331</td>
<td>0.04873332</td>
</tr>
<tr>
<td>rel 4/MEP</td>
<td>0.00797966</td>
<td>0.00002394</td>
<td>0.00002394</td>
</tr>
<tr>
<td>rel 5/ORG</td>
<td>1.47314812</td>
<td>0.76290439</td>
<td>0.76260915</td>
</tr>
<tr>
<td>rel 5/IDC</td>
<td>0.35105470</td>
<td>0.04901089</td>
<td>0.04871026</td>
</tr>
<tr>
<td>rel 5/MEP</td>
<td>0.00140859</td>
<td>0.0009952</td>
<td>0.0009952</td>
</tr>
<tr>
<td>rel 6/ORG</td>
<td>1.47309075</td>
<td>0.76290036</td>
<td>0.76260834</td>
</tr>
<tr>
<td>rel 6/IDC</td>
<td>0.35111287</td>
<td>0.04900952</td>
<td>0.04879453</td>
</tr>
<tr>
<td>rel 6/MEP</td>
<td>0.00035122</td>
<td>0.00000005</td>
<td>0.00000005</td>
</tr>
<tr>
<td>rel 7/ORG</td>
<td>1.47308471</td>
<td>0.76289604</td>
<td>0.76260829</td>
</tr>
<tr>
<td>rel 7/IDC</td>
<td>0.35111946</td>
<td>0.04900934</td>
<td>0.04870940</td>
</tr>
<tr>
<td>rel 7/MEP</td>
<td>0.00007689</td>
<td>0.00000000</td>
<td>0.00000000</td>
</tr>
<tr>
<td>rel 8/ORG</td>
<td>1.47308599</td>
<td>0.76289530</td>
<td>0.76260829</td>
</tr>
<tr>
<td>rel 8/IDC</td>
<td>0.35111825</td>
<td>0.04900934</td>
<td>0.04870940</td>
</tr>
<tr>
<td>rel 8/MEP</td>
<td>0.00002210</td>
<td>0.00000000</td>
<td>0.00000000</td>
</tr>
<tr>
<td>rel 9/ORG</td>
<td>1.47308655</td>
<td>0.76289533</td>
<td>0.76260829</td>
</tr>
<tr>
<td>rel 9/IDC</td>
<td>0.35111765</td>
<td>0.04900934</td>
<td>0.04870940</td>
</tr>
<tr>
<td>rel 9/MEP</td>
<td>0.00001461</td>
<td>0.00000000</td>
<td>0.00000000</td>
</tr>
<tr>
<td>rel 10/ORG</td>
<td>1.47308660</td>
<td>0.76289537</td>
<td>0.76260829</td>
</tr>
<tr>
<td>rel 10/IDC</td>
<td>0.35111758</td>
<td>0.04900934</td>
<td>0.04870940</td>
</tr>
<tr>
<td>rel 10/MEP</td>
<td>0.00000136</td>
<td>0.00000000</td>
<td>0.00000000</td>
</tr>
<tr>
<td>IDCT/ORG</td>
<td>1.74112547</td>
<td>1.18847816</td>
<td>1.18964096</td>
</tr>
<tr>
<td>MEP/ORG</td>
<td>1.68074670</td>
<td>0.98712890</td>
<td>0.98253943</td>
</tr>
<tr>
<td>MEP/IDCT</td>
<td>0.41612845</td>
<td>0.07038479</td>
<td>0.06972954</td>
</tr>
<tr>
<td>rel 1/ORG</td>
<td>1.89243231</td>
<td>1.29438568</td>
<td>1.26428790</td>
</tr>
<tr>
<td>rel 1/IDC</td>
<td>0.97438525</td>
<td>0.35216393</td>
<td>0.35147928</td>
</tr>
<tr>
<td>rel 1/MEP</td>
<td>0.87159773</td>
<td>0.28189370</td>
<td>0.28174653</td>
</tr>
<tr>
<td>rel 2/ORG</td>
<td>1.68236810</td>
<td>0.99407223</td>
<td>0.99071977</td>
</tr>
<tr>
<td>rel 2/IDC</td>
<td>0.45012960</td>
<td>0.07827378</td>
<td>0.07791045</td>
</tr>
<tr>
<td>rel 2/MEP</td>
<td>0.15180280</td>
<td>0.00817620</td>
<td>0.00818103</td>
</tr>
<tr>
<td>rel 3/ORG</td>
<td>1.68008965</td>
<td>0.98649000</td>
<td>0.98297481</td>
</tr>
<tr>
<td>rel 3/IDC</td>
<td>0.41751437</td>
<td>0.07080120</td>
<td>0.07016514</td>
</tr>
<tr>
<td>rel 3/MEP</td>
<td>0.03331223</td>
<td>0.00043546</td>
<td>0.00043564</td>
</tr>
<tr>
<td>rel 4/ORG</td>
<td>1.68090569</td>
<td>0.98704267</td>
<td>0.98255458</td>
</tr>
<tr>
<td>rel 4/IDC</td>
<td>0.41601567</td>
<td>0.07040393</td>
<td>0.06974488</td>
</tr>
<tr>
<td>rel 4/MEP</td>
<td>0.00626106</td>
<td>0.00001527</td>
<td>0.00001527</td>
</tr>
<tr>
<td>rel 5/ORG</td>
<td>1.68078905</td>
<td>0.98712709</td>
<td>0.98253979</td>
</tr>
<tr>
<td>rel 5/IDC</td>
<td>0.41610882</td>
<td>0.07038575</td>
<td>0.06973011</td>
</tr>
<tr>
<td>rel 5/MEP</td>
<td>0.00107057</td>
<td>0.00000048</td>
<td>0.00000048</td>
</tr>
<tr>
<td>rel 6/ORG</td>
<td>1.68074902</td>
<td>0.98713231</td>
<td>0.98253934</td>
</tr>
<tr>
<td>rel 6/IDC</td>
<td>0.41612847</td>
<td>0.07038473</td>
<td>0.06972966</td>
</tr>
<tr>
<td>rel 6/MEP</td>
<td>0.00002398</td>
<td>0.00000002</td>
<td>0.00000002</td>
</tr>
<tr>
<td>rel 7/ORG</td>
<td>1.68074399</td>
<td>0.98712915</td>
<td>0.98253932</td>
</tr>
<tr>
<td>rel 7/IDC</td>
<td>0.41612933</td>
<td>0.07038475</td>
<td>0.06972963</td>
</tr>
<tr>
<td>rel 7/MEP</td>
<td>0.00005672</td>
<td>0.00000000</td>
<td>0.00000000</td>
</tr>
<tr>
<td>rel 8/ORG</td>
<td>1.68074476</td>
<td>0.98712825</td>
<td>0.98253931</td>
</tr>
<tr>
<td>rel 8/IDC</td>
<td>0.41612874</td>
<td>0.07038479</td>
<td>0.06972963</td>
</tr>
<tr>
<td>rel 8/MEP</td>
<td>0.00002471</td>
<td>0.00000000</td>
<td>0.00000000</td>
</tr>
<tr>
<td>rel 9/ORG</td>
<td>1.68074513</td>
<td>0.98712819</td>
<td>0.98253931</td>
</tr>
<tr>
<td>rel 9/IDC</td>
<td>0.41612860</td>
<td>0.07038479</td>
<td>0.06972963</td>
</tr>
<tr>
<td>rel 9/MEP</td>
<td>0.00002184</td>
<td>0.00000000</td>
<td>0.00000000</td>
</tr>
<tr>
<td>rel 10/ORG</td>
<td>1.68074517</td>
<td>0.98712821</td>
<td>0.98253931</td>
</tr>
<tr>
<td>rel 10/IDC</td>
<td>0.41612860</td>
<td>0.07038479</td>
<td>0.06972963</td>
</tr>
<tr>
<td>rel 10/MEP</td>
<td>0.00002167</td>
<td>0.00000000</td>
<td>0.00000000</td>
</tr>
<tr>
<td>IDCT/ORG</td>
<td>2.17505877</td>
<td>1.89026210</td>
<td>1.87922527</td>
</tr>
<tr>
<td>MEP/ORG</td>
<td>2.08974301</td>
<td>1.53418394</td>
<td>1.51229428</td>
</tr>
<tr>
<td>MEP/IDCT</td>
<td>0.55463822</td>
<td>0.12546744</td>
<td>0.12336016</td>
</tr>
<tr>
<td>rel 1/ORG</td>
<td>2.23185020</td>
<td>1.81134653</td>
<td>1.75381757</td>
</tr>
<tr>
<td>rel 1/IDC</td>
<td>0.98027819</td>
<td>0.36803651</td>
<td>0.36488448</td>
</tr>
<tr>
<td></td>
<td>rel 7/ORG</td>
<td>2.58883987</td>
<td>2.32668406</td>
</tr>
<tr>
<td>---</td>
<td>----------</td>
<td>------------</td>
<td>------------</td>
</tr>
<tr>
<td></td>
<td>rel 7/IDC</td>
<td>0.63618408</td>
<td>0.17424921</td>
</tr>
<tr>
<td></td>
<td>rel 7/MEP</td>
<td>0.00002717</td>
<td>0.00000000</td>
</tr>
<tr>
<td></td>
<td>rel 8/ORG</td>
<td>2.58883980</td>
<td>2.32668358</td>
</tr>
<tr>
<td></td>
<td>rel 8/IDC</td>
<td>0.63618376</td>
<td>0.17424916</td>
</tr>
<tr>
<td></td>
<td>rel 8/MEP</td>
<td>0.00002300</td>
<td>0.00000000</td>
</tr>
<tr>
<td></td>
<td>rel 9/ORG</td>
<td>2.58883974</td>
<td>2.32668350</td>
</tr>
<tr>
<td></td>
<td>rel 9/IDC</td>
<td>0.63618378</td>
<td>0.17424914</td>
</tr>
<tr>
<td></td>
<td>rel 9/MEP</td>
<td>0.00002308</td>
<td>0.00000000</td>
</tr>
<tr>
<td></td>
<td>rel 10/ORG</td>
<td>2.58883973</td>
<td>2.32668350</td>
</tr>
<tr>
<td></td>
<td>rel 10/IDC</td>
<td>0.63618379</td>
<td>0.17424914</td>
</tr>
<tr>
<td></td>
<td>rel 10/MEP</td>
<td>0.00002312</td>
<td>0.00000000</td>
</tr>
<tr>
<td></td>
<td>IDCT/ORG</td>
<td>3.17117118</td>
<td>3.87543085</td>
</tr>
<tr>
<td></td>
<td>MEP/ORG</td>
<td>3.04123094</td>
<td>3.29728492</td>
</tr>
<tr>
<td></td>
<td>MEP/IDCT</td>
<td>0.63617124</td>
<td>0.16048044</td>
</tr>
<tr>
<td></td>
<td>rel 1/ORG</td>
<td>3.11974020</td>
<td>3.51409872</td>
</tr>
<tr>
<td></td>
<td>rel 1/IDC</td>
<td>0.94300255</td>
<td>0.33348927</td>
</tr>
<tr>
<td></td>
<td>rel 1/MEP</td>
<td>0.66698487</td>
<td>0.17237829</td>
</tr>
<tr>
<td></td>
<td>rel 2/ORG</td>
<td>3.04164789</td>
<td>3.30271603</td>
</tr>
<tr>
<td></td>
<td>rel 2/IDC</td>
<td>0.65220548</td>
<td>0.16387511</td>
</tr>
<tr>
<td></td>
<td>rel 2/MEP</td>
<td>0.10111241</td>
<td>0.00344737</td>
</tr>
<tr>
<td></td>
<td>rel 3/ORG</td>
<td>3.04123518</td>
<td>3.29725921</td>
</tr>
<tr>
<td></td>
<td>rel 3/IDC</td>
<td>0.63584194</td>
<td>0.16062756</td>
</tr>
<tr>
<td></td>
<td>rel 3/MEP</td>
<td>0.01627981</td>
<td>0.00010702</td>
</tr>
<tr>
<td></td>
<td>rel 4/ORG</td>
<td>3.04121057</td>
<td>3.29717925</td>
</tr>
<tr>
<td></td>
<td>rel 4/IDC</td>
<td>0.63609370</td>
<td>0.16047722</td>
</tr>
<tr>
<td></td>
<td>rel 4/MEP</td>
<td>0.00220610</td>
<td>0.0000192</td>
</tr>
<tr>
<td></td>
<td>rel 5/ORG</td>
<td>3.04122373</td>
<td>3.29726926</td>
</tr>
<tr>
<td></td>
<td>rel 5/IDC</td>
<td>0.63617193</td>
<td>0.16047869</td>
</tr>
<tr>
<td></td>
<td>rel 5/MEP</td>
<td>0.00032292</td>
<td>0.00000005</td>
</tr>
<tr>
<td></td>
<td>rel 6/ORG</td>
<td>3.04122867</td>
<td>3.29728417</td>
</tr>
<tr>
<td></td>
<td>rel 6/IDC</td>
<td>0.63617525</td>
<td>0.16048028</td>
</tr>
<tr>
<td></td>
<td>rel 6/MEP</td>
<td>0.00006489</td>
<td>0.00000000</td>
</tr>
<tr>
<td></td>
<td>rel 7/ORG</td>
<td>3.04122869</td>
<td>3.29728451</td>
</tr>
<tr>
<td></td>
<td>rel 7/IDC</td>
<td>0.63617397</td>
<td>0.16048041</td>
</tr>
<tr>
<td></td>
<td>rel 7/MEP</td>
<td>0.00002973</td>
<td>0.00000000</td>
</tr>
<tr>
<td></td>
<td>rel 8/ORG</td>
<td>3.04122862</td>
<td>3.29728409</td>
</tr>
<tr>
<td></td>
<td>rel 8/IDC</td>
<td>0.63617366</td>
<td>0.16048039</td>
</tr>
<tr>
<td></td>
<td>rel 8/MEP</td>
<td>0.00002926</td>
<td>0.00000000</td>
</tr>
<tr>
<td></td>
<td>rel 9/ORG</td>
<td>3.04122861</td>
<td>3.29728409</td>
</tr>
<tr>
<td></td>
<td>rel 9/IDC</td>
<td>0.63617366</td>
<td>0.16048039</td>
</tr>
<tr>
<td></td>
<td>rel 9/MEP</td>
<td>0.00002925</td>
<td>0.00000000</td>
</tr>
<tr>
<td></td>
<td>IDCT/ORG</td>
<td>3.68085725</td>
<td>5.08986580</td>
</tr>
<tr>
<td></td>
<td>MEP/ORG</td>
<td>3.56396930</td>
<td>4.49390578</td>
</tr>
<tr>
<td></td>
<td>MEP/IDCT</td>
<td>0.64810045</td>
<td>0.15764811</td>
</tr>
<tr>
<td></td>
<td>rel 1/ORG</td>
<td>3.66091645</td>
<td>4.72323491</td>
</tr>
<tr>
<td>rel</td>
<td>IDC</td>
<td>MEP</td>
<td>ORG</td>
</tr>
<tr>
<td>------</td>
<td>-------</td>
<td>-------</td>
<td>-------</td>
</tr>
<tr>
<td>1</td>
<td>0.96333998</td>
<td>0.33204187</td>
<td>0.32802938</td>
</tr>
<tr>
<td>2</td>
<td>3.56237217</td>
<td>4.49755327</td>
<td>4.46133204</td>
</tr>
<tr>
<td>3</td>
<td>0.66327465</td>
<td>0.16057026</td>
<td>0.15994876</td>
</tr>
<tr>
<td>4</td>
<td>0.09332134</td>
<td>0.00305181</td>
<td>0.00305402</td>
</tr>
<tr>
<td>5</td>
<td>0.64837869</td>
<td>0.15766943</td>
<td>0.15698078</td>
</tr>
<tr>
<td>6</td>
<td>0.01453231</td>
<td>0.00008586</td>
<td>0.00008590</td>
</tr>
<tr>
<td>7</td>
<td>3.56393328</td>
<td>4.49378621</td>
<td>4.45827975</td>
</tr>
<tr>
<td>8</td>
<td>0.64799337</td>
<td>0.15764311</td>
<td>0.15689627</td>
</tr>
<tr>
<td>9</td>
<td>0.00189874</td>
<td>0.00000132</td>
<td>0.00000132</td>
</tr>
<tr>
<td>10</td>
<td>3.56397455</td>
<td>4.49390234</td>
<td>0.15689498</td>
</tr>
<tr>
<td>11</td>
<td>0.64808781</td>
<td>0.15764869</td>
<td>0.00000002</td>
</tr>
<tr>
<td>12</td>
<td>0.00022035</td>
<td>0.0000002</td>
<td>4.45827843</td>
</tr>
<tr>
<td>13</td>
<td>3.56396852</td>
<td>4.49390594</td>
<td>0.15689495</td>
</tr>
<tr>
<td>14</td>
<td>0.64810356</td>
<td>0.15764809</td>
<td>0.00000000</td>
</tr>
<tr>
<td>15</td>
<td>0.00000185</td>
<td>0.00000000</td>
<td>4.45827843</td>
</tr>
<tr>
<td>16</td>
<td>3.56396842</td>
<td>4.49390567</td>
<td>0.15689495</td>
</tr>
<tr>
<td>17</td>
<td>0.64810323</td>
<td>0.15764808</td>
<td>0.00000000</td>
</tr>
<tr>
<td>18</td>
<td>0.00000182</td>
<td>0.00000000</td>
<td>4.45827843</td>
</tr>
<tr>
<td>19</td>
<td>3.56396843</td>
<td>4.49390566</td>
<td>0.15689495</td>
</tr>
<tr>
<td>20</td>
<td>0.64810324</td>
<td>0.15764808</td>
<td>0.00000000</td>
</tr>
<tr>
<td>21</td>
<td>0.00000189</td>
<td>0.00000000</td>
<td>4.45827843</td>
</tr>
<tr>
<td>22</td>
<td>5.21114184</td>
<td>9.81343355</td>
<td>9.38685482</td>
</tr>
<tr>
<td>23</td>
<td>5.14234834</td>
<td>9.45834966</td>
<td>9.06846657</td>
</tr>
<tr>
<td>24</td>
<td>0.37767784</td>
<td>0.05917869</td>
<td>0.05879711</td>
</tr>
<tr>
<td>25</td>
<td>5.16699009</td>
<td>9.53435462</td>
<td>9.12250212</td>
</tr>
<tr>
<td>26</td>
<td>0.52063656</td>
<td>0.11416658</td>
<td>0.11283262</td>
</tr>
<tr>
<td>27</td>
<td>0.37020363</td>
<td>0.05436192</td>
<td>0.05403539</td>
</tr>
<tr>
<td>28</td>
<td>5.14230749</td>
<td>9.46509084</td>
<td>9.06929391</td>
</tr>
<tr>
<td>29</td>
<td>0.38384706</td>
<td>0.05999590</td>
<td>0.05962461</td>
</tr>
<tr>
<td>30</td>
<td>0.04668706</td>
<td>0.00082714</td>
<td>0.00082753</td>
</tr>
<tr>
<td>31</td>
<td>5.14219136</td>
<td>9.45845803</td>
<td>9.06848449</td>
</tr>
<tr>
<td>32</td>
<td>0.37797761</td>
<td>0.05918233</td>
<td>0.05881513</td>
</tr>
<tr>
<td>33</td>
<td>0.00637613</td>
<td>0.00018040</td>
<td>0.0001804</td>
</tr>
<tr>
<td>34</td>
<td>5.14233071</td>
<td>9.45830509</td>
<td>9.06846687</td>
</tr>
<tr>
<td>35</td>
<td>0.37765879</td>
<td>0.05917835</td>
<td>0.05879750</td>
</tr>
<tr>
<td>36</td>
<td>0.00093526</td>
<td>0.00000037</td>
<td>0.00000037</td>
</tr>
<tr>
<td>37</td>
<td>5.14234725</td>
<td>9.45834607</td>
<td>9.06846650</td>
</tr>
<tr>
<td>38</td>
<td>0.37767294</td>
<td>0.05917881</td>
<td>0.05879714</td>
</tr>
<tr>
<td>39</td>
<td>0.00010490</td>
<td>0.00000000</td>
<td>0.00000000</td>
</tr>
<tr>
<td>40</td>
<td>5.14234787</td>
<td>9.45835149</td>
<td>9.06846650</td>
</tr>
<tr>
<td>41</td>
<td>0.37767856</td>
<td>0.05917872</td>
<td>0.05879713</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IDCT/ORG</th>
<th>MEP/ORG</th>
<th>MEP/IDCT</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.21114184</td>
<td>9.81343355</td>
<td>9.38685482</td>
</tr>
<tr>
<td>5.14234834</td>
<td>9.45834966</td>
<td>9.06846657</td>
</tr>
<tr>
<td>0.37767784</td>
<td>0.05917869</td>
<td>0.05879711</td>
</tr>
<tr>
<td>5.16699009</td>
<td>9.53435462</td>
<td>9.12250212</td>
</tr>
<tr>
<td>0.52063656</td>
<td>0.11416658</td>
<td>0.11283262</td>
</tr>
<tr>
<td>0.37020363</td>
<td>0.05436192</td>
<td>0.05403539</td>
</tr>
<tr>
<td>5.14230749</td>
<td>9.46509084</td>
<td>9.06929391</td>
</tr>
<tr>
<td>0.38384706</td>
<td>0.05999590</td>
<td>0.05962461</td>
</tr>
<tr>
<td>0.04668706</td>
<td>0.00082714</td>
<td>0.00082753</td>
</tr>
<tr>
<td>5.14219136</td>
<td>9.45845803</td>
<td>9.06848449</td>
</tr>
<tr>
<td>0.37797761</td>
<td>0.05918233</td>
<td>0.05881513</td>
</tr>
<tr>
<td>0.00637613</td>
<td>0.00018040</td>
<td>0.0001804</td>
</tr>
<tr>
<td>5.14233071</td>
<td>9.45830509</td>
<td>9.06846687</td>
</tr>
<tr>
<td>0.37765879</td>
<td>0.05917835</td>
<td>0.05879750</td>
</tr>
<tr>
<td>0.00093526</td>
<td>0.00000037</td>
<td>0.00000037</td>
</tr>
<tr>
<td>5.14234725</td>
<td>9.45834607</td>
<td>9.06846650</td>
</tr>
<tr>
<td>0.37767294</td>
<td>0.05917881</td>
<td>0.05879714</td>
</tr>
<tr>
<td>0.00010490</td>
<td>0.00000000</td>
<td>0.00000000</td>
</tr>
<tr>
<td>5.14234787</td>
<td>9.45835149</td>
<td>9.06846650</td>
</tr>
<tr>
<td>0.37767856</td>
<td>0.05917872</td>
<td>0.05879713</td>
</tr>
<tr>
<td>rel 6/MEP</td>
<td>0.00001830</td>
<td>0.00000000</td>
</tr>
<tr>
<td>rel 7/ORG</td>
<td>5.14234766</td>
<td>9.45835155</td>
</tr>
<tr>
<td>rel 7/IDC</td>
<td>0.37767914</td>
<td>0.05917869</td>
</tr>
<tr>
<td>rel 7/MEP</td>
<td>0.00001706</td>
<td>0.00000000</td>
</tr>
<tr>
<td>rel 8/ORG</td>
<td>5.14234763</td>
<td>9.45835148</td>
</tr>
<tr>
<td>rel 8/IDC</td>
<td>0.37767912</td>
<td>0.05917869</td>
</tr>
<tr>
<td>rel 8/MEP</td>
<td>0.00001766</td>
<td>0.00000000</td>
</tr>
<tr>
<td>rel 9/ORG</td>
<td>5.14234762</td>
<td>9.45835146</td>
</tr>
<tr>
<td>rel 9/IDC</td>
<td>0.37767911</td>
<td>0.05917869</td>
</tr>
<tr>
<td>rel 9/MEP</td>
<td>0.00001773</td>
<td>0.00000000</td>
</tr>
<tr>
<td>rel 10/ORG</td>
<td>5.14234762</td>
<td>9.45835146</td>
</tr>
<tr>
<td>rel 10/IDC</td>
<td>0.37767911</td>
<td>0.05917869</td>
</tr>
<tr>
<td>rel 10/MEP</td>
<td>0.00001773</td>
<td>0.00000000</td>
</tr>
<tr>
<td>IDCT/ORG</td>
<td>5.56642803</td>
<td>11.02811602</td>
</tr>
<tr>
<td>MEP/ORG</td>
<td>5.53993731</td>
<td>10.88033247</td>
</tr>
<tr>
<td>MEP/IDCT</td>
<td>0.24699515</td>
<td>0.02471669</td>
</tr>
<tr>
<td>rel 1/ORG</td>
<td>5.54656090</td>
<td>10.89048553</td>
</tr>
<tr>
<td>rel 1/IDC</td>
<td>0.33959755</td>
<td>0.04860747</td>
</tr>
<tr>
<td>rel 1/MEP</td>
<td>0.23866196</td>
<td>0.02387087</td>
</tr>
<tr>
<td>rel 2/ORG</td>
<td>5.54009548</td>
<td>10.88592757</td>
</tr>
<tr>
<td>rel 2/IDC</td>
<td>0.24997580</td>
<td>0.02503159</td>
</tr>
<tr>
<td>rel 2/MEP</td>
<td>0.02770889</td>
<td>0.00031249</td>
</tr>
<tr>
<td>rel 3/ORG</td>
<td>5.53993154</td>
<td>10.88065839</td>
</tr>
<tr>
<td>rel 3/IDC</td>
<td>0.24703560</td>
<td>0.02471748</td>
</tr>
<tr>
<td>rel 3/MEP</td>
<td>0.00176712</td>
<td>0.00000137</td>
</tr>
<tr>
<td>rel 4/ORG</td>
<td>5.53993470</td>
<td>10.88034016</td>
</tr>
<tr>
<td>rel 4/IDC</td>
<td>0.24699779</td>
<td>0.02471674</td>
</tr>
<tr>
<td>rel 4/MEP</td>
<td>0.00020401</td>
<td>0.00000000</td>
</tr>
<tr>
<td>rel 5/ORG</td>
<td>5.53993676</td>
<td>10.88033310</td>
</tr>
<tr>
<td>rel 5/IDC</td>
<td>0.24699586</td>
<td>0.02471669</td>
</tr>
<tr>
<td>rel 5/MEP</td>
<td>0.00002905</td>
<td>0.00000000</td>
</tr>
<tr>
<td>rel 6/ORG</td>
<td>5.53993694</td>
<td>10.88033425</td>
</tr>
<tr>
<td>rel 6/IDC</td>
<td>0.24699606</td>
<td>0.02471669</td>
</tr>
<tr>
<td>rel 6/MEP</td>
<td>0.00001432</td>
<td>0.00000000</td>
</tr>
<tr>
<td>rel 7/ORG</td>
<td>5.53993694</td>
<td>10.88033443</td>
</tr>
<tr>
<td>rel 7/IDC</td>
<td>0.24699612</td>
<td>0.02471669</td>
</tr>
<tr>
<td>rel 7/MEP</td>
<td>0.00001402</td>
<td>0.00000000</td>
</tr>
<tr>
<td>rel 8/ORG</td>
<td>5.53993694</td>
<td>10.88033445</td>
</tr>
<tr>
<td>rel 8/IDC</td>
<td>0.24699613</td>
<td>0.02471669</td>
</tr>
<tr>
<td>rel 8/MEP</td>
<td>0.00001402</td>
<td>0.00000000</td>
</tr>
<tr>
<td>rel 9/ORG</td>
<td>5.53993694</td>
<td>10.88033445</td>
</tr>
<tr>
<td>rel 9/IDC</td>
<td>0.24699613</td>
<td>0.02471669</td>
</tr>
<tr>
<td>rel 9/MEP</td>
<td>0.00001402</td>
<td>0.00000000</td>
</tr>
<tr>
<td>rel 10/ORG</td>
<td>5.53993694</td>
<td>10.88033445</td>
</tr>
<tr>
<td>rel 10/IDC</td>
<td>0.24699613</td>
<td>0.02471669</td>
</tr>
<tr>
<td>rel 10/MEP</td>
<td>0.00001402</td>
<td>0.00000000</td>
</tr>
<tr>
<td>IDCT/ORG</td>
<td>5.76460163</td>
<td>11.74303887</td>
</tr>
<tr>
<td>MEP/ORG</td>
<td>5.74750597</td>
<td>11.62276946</td>
</tr>
<tr>
<td>MEP/IDCT</td>
<td>0.26472093</td>
<td>0.02780918</td>
</tr>
<tr>
<td>rel 1/ORG 5.75566913</td>
<td>11.62887343</td>
<td>11.33818607</td>
</tr>
<tr>
<td>------------------</td>
<td>-------------</td>
<td>-------------</td>
</tr>
<tr>
<td>rel 1/IDC 0.34137343</td>
<td>0.04764255</td>
<td>0.04750711</td>
</tr>
<tr>
<td>rel 1/MEP 0.22178003</td>
<td>0.01974736</td>
<td>0.01973153</td>
</tr>
<tr>
<td>rel 2/ORG 5.74746231</td>
<td>11.62532604</td>
<td>11.31855510</td>
</tr>
<tr>
<td>rel 2/IDC 0.26564762</td>
<td>0.02790963</td>
<td>0.02787629</td>
</tr>
<tr>
<td>rel 2/MEP 0.01547255</td>
<td>0.00010049</td>
<td>0.00010050</td>
</tr>
<tr>
<td>rel 3/ORG 5.74750095</td>
<td>11.62288178</td>
<td>11.31845536</td>
</tr>
<tr>
<td>rel 3/IDC 0.26475436</td>
<td>0.02780985</td>
<td>0.02777653</td>
</tr>
<tr>
<td>rel 3/MEP 0.00134868</td>
<td>0.00000076</td>
<td>0.00000076</td>
</tr>
<tr>
<td>rel 4/ORG 5.74750443</td>
<td>11.62276160</td>
<td>11.31845640</td>
</tr>
<tr>
<td>rel 4/IDC 0.26472051</td>
<td>0.02780918</td>
<td>0.02777578</td>
</tr>
<tr>
<td>rel 4/MEP 0.000011926</td>
<td>0.00000001</td>
<td>0.00000001</td>
</tr>
<tr>
<td>rel 5/ORG 5.74750595</td>
<td>11.62276780</td>
<td>11.31845459</td>
</tr>
<tr>
<td>rel 5/IDC 0.26472074</td>
<td>0.02780918</td>
<td>0.02777577</td>
</tr>
<tr>
<td>rel 5/MEP 0.00001072</td>
<td>0.00000000</td>
<td>0.00000000</td>
</tr>
<tr>
<td>rel 6/ORG 5.74750600</td>
<td>11.62276923</td>
<td>11.31845459</td>
</tr>
<tr>
<td>rel 6/IDC 0.26472093</td>
<td>0.02780918</td>
<td>0.02777577</td>
</tr>
<tr>
<td>rel 6/MEP 0.0000253</td>
<td>0.00000000</td>
<td>0.00000000</td>
</tr>
<tr>
<td>rel 7/ORG 5.74750599</td>
<td>11.62276930</td>
<td>11.31845459</td>
</tr>
<tr>
<td>rel 7/IDC 0.26472094</td>
<td>0.02780918</td>
<td>0.02777577</td>
</tr>
<tr>
<td>rel 7/MEP 0.00000213</td>
<td>0.00000000</td>
<td>0.00000000</td>
</tr>
<tr>
<td>rel 8/ORG 5.74750599</td>
<td>11.62276930</td>
<td>11.31845459</td>
</tr>
<tr>
<td>rel 8/IDC 0.26472094</td>
<td>0.02780918</td>
<td>0.02777577</td>
</tr>
<tr>
<td>rel 8/MEP 0.00000211</td>
<td>0.00000000</td>
<td>0.00000000</td>
</tr>
<tr>
<td>rel 9/ORG 5.74750599</td>
<td>11.62276930</td>
<td>11.31845459</td>
</tr>
<tr>
<td>rel 9/IDC 0.26472094</td>
<td>0.02780918</td>
<td>0.02777577</td>
</tr>
<tr>
<td>rel 9/MEP 0.00000211</td>
<td>0.00000000</td>
<td>0.00000000</td>
</tr>
<tr>
<td>rel 10/ORG 5.74750599</td>
<td>11.62276930</td>
<td>11.31845459</td>
</tr>
<tr>
<td>rel 10/IDC 0.26472094</td>
<td>0.02780918</td>
<td>0.02777577</td>
</tr>
<tr>
<td>rel 10/MEP 0.00000211</td>
<td>0.00000000</td>
<td>0.00000000</td>
</tr>
<tr>
<td>IDCT/ORG 6.10544076</td>
<td>13.01326134</td>
<td>12.88445234</td>
</tr>
<tr>
<td>MEP/ORG 6.10283924</td>
<td>12.96622734</td>
<td>12.84242307</td>
</tr>
<tr>
<td>MEP/IDCT 0.2185412</td>
<td>12.96949673</td>
<td>0.01949611</td>
</tr>
<tr>
<td>rel 1/ORG 6.10181607</td>
<td>12.94902322</td>
<td>12.86104487</td>
</tr>
<tr>
<td>rel 1/IDC 0.29278188</td>
<td>0.03649500</td>
<td>0.03627065</td>
</tr>
<tr>
<td>rel 1/MEP 0.20251676</td>
<td>0.01680982</td>
<td>0.01680190</td>
</tr>
<tr>
<td>rel 2/ORG 6.10288836</td>
<td>12.96629615</td>
<td>12.8426084</td>
</tr>
<tr>
<td>rel 2/IDC 0.21857600</td>
<td>0.01951869</td>
<td>0.01948682</td>
</tr>
<tr>
<td>rel 2/MEP 0.00648485</td>
<td>0.00001776</td>
<td>0.00001776</td>
</tr>
<tr>
<td>rel 3/ORG 6.10283940</td>
<td>12.96627092</td>
<td>12.8442315</td>
</tr>
<tr>
<td>rel 3/IDC 0.21854537</td>
<td>0.01949678</td>
<td>0.01946913</td>
</tr>
<tr>
<td>rel 3/MEP 0.00038444</td>
<td>0.00000006</td>
<td>0.00000006</td>
</tr>
<tr>
<td>rel 4/ORG 6.10283943</td>
<td>12.96623518</td>
<td>12.8442309</td>
</tr>
<tr>
<td>rel 4/IDC 0.21854373</td>
<td>0.01949669</td>
<td>0.01946907</td>
</tr>
<tr>
<td>rel 4/MEP 0.00002673</td>
<td>0.00000000</td>
<td>0.00000000</td>
</tr>
<tr>
<td>rel 5/ORG 6.10283942</td>
<td>12.96623397</td>
<td>12.8442309</td>
</tr>
<tr>
<td>rel 5/IDC 0.21854367</td>
<td>0.01949669</td>
<td>0.01946907</td>
</tr>
<tr>
<td>rel 5/MEP 0.00002369</td>
<td>0.00000000</td>
<td>0.00000000</td>
</tr>
<tr>
<td>rel 6/ORG 6.10283942</td>
<td>12.96623397</td>
<td>12.8442309</td>
</tr>
<tr>
<td>rel</td>
<td>0RG</td>
<td>IDC</td>
</tr>
<tr>
<td>-----</td>
<td>-----</td>
<td>-----</td>
</tr>
<tr>
<td>6/IDC</td>
<td>0.21854367</td>
<td>0.01949669</td>
</tr>
<tr>
<td>6/MEP</td>
<td>0.00002376</td>
<td>0.00000000</td>
</tr>
<tr>
<td>7/ORG</td>
<td>6.10283942</td>
<td>12.96623397</td>
</tr>
<tr>
<td>7/IDC</td>
<td>0.21854367</td>
<td>0.01949669</td>
</tr>
<tr>
<td>7/MEP</td>
<td>0.00002376</td>
<td>0.00000000</td>
</tr>
<tr>
<td>8/ORG</td>
<td>6.10283942</td>
<td>12.96623397</td>
</tr>
<tr>
<td>8/IDC</td>
<td>0.21854367</td>
<td>0.01949669</td>
</tr>
<tr>
<td>8/MEP</td>
<td>0.00002376</td>
<td>0.00000000</td>
</tr>
<tr>
<td>9/ORG</td>
<td>6.10283942</td>
<td>12.96623397</td>
</tr>
<tr>
<td>9/IDC</td>
<td>0.21854367</td>
<td>0.01949669</td>
</tr>
<tr>
<td>9/MEP</td>
<td>0.00002376</td>
<td>0.00000000</td>
</tr>
<tr>
<td>10/ORG</td>
<td>6.10283942</td>
<td>12.96623397</td>
</tr>
<tr>
<td>10/IDC</td>
<td>0.21854367</td>
<td>0.01949669</td>
</tr>
<tr>
<td>10/MEP</td>
<td>0.00002376</td>
<td>0.00000000</td>
</tr>
<tr>
<td>IDCT/ORG</td>
<td>6.36202443</td>
<td>14.10041438</td>
</tr>
<tr>
<td>MEP/ORG</td>
<td>6.32082559</td>
<td>13.87477387</td>
</tr>
<tr>
<td>MEP/IDCT</td>
<td>0.23906526</td>
<td>0.02478753</td>
</tr>
<tr>
<td>rel 1/0RG</td>
<td>6.32795006</td>
<td>13.84475586</td>
</tr>
<tr>
<td>rel 1/IDC</td>
<td>0.29027932</td>
<td>0.04179758</td>
</tr>
<tr>
<td>rel 1/MEP</td>
<td>0.19181175</td>
<td>0.01663347</td>
</tr>
<tr>
<td>rel 2/0RG</td>
<td>6.32101931</td>
<td>13.87306095</td>
</tr>
<tr>
<td>rel 2/IDC</td>
<td>0.23870557</td>
<td>0.02486837</td>
</tr>
<tr>
<td>rel 2/MEP</td>
<td>0.01216930</td>
<td>0.00006554</td>
</tr>
<tr>
<td>rel 3/IDC</td>
<td>0.23903829</td>
<td>0.02478866</td>
</tr>
<tr>
<td>rel 3/MEP</td>
<td>0.00084153</td>
<td>0.00000300</td>
</tr>
<tr>
<td>rel 4/0RG</td>
<td>6.32082582</td>
<td>13.87477098</td>
</tr>
<tr>
<td>rel 4/IDC</td>
<td>0.23906488</td>
<td>0.02478754</td>
</tr>
<tr>
<td>rel 4/MEP</td>
<td>0.00004645</td>
<td>0.00000000</td>
</tr>
<tr>
<td>rel 5/0RG</td>
<td>6.32082543</td>
<td>13.87477660</td>
</tr>
<tr>
<td>rel 5/IDC</td>
<td>0.23906649</td>
<td>0.02478749</td>
</tr>
<tr>
<td>rel 5/MEP</td>
<td>0.00001287</td>
<td>0.00000000</td>
</tr>
<tr>
<td>rel 6/0RG</td>
<td>6.32082541</td>
<td>13.87477693</td>
</tr>
<tr>
<td>rel 6/IDC</td>
<td>0.23906659</td>
<td>0.02478749</td>
</tr>
<tr>
<td>rel 6/MEP</td>
<td>0.00001292</td>
<td>0.00000000</td>
</tr>
<tr>
<td>rel 7/0RG</td>
<td>6.32082541</td>
<td>13.87477695</td>
</tr>
<tr>
<td>rel 7/IDC</td>
<td>0.23906659</td>
<td>0.02478749</td>
</tr>
<tr>
<td>rel 7/MEP</td>
<td>7.00001294</td>
<td>0.00000000</td>
</tr>
<tr>
<td>rel 8/0RG</td>
<td>6.32082541</td>
<td>13.87477695</td>
</tr>
<tr>
<td>rel 8/IDC</td>
<td>0.23906659</td>
<td>0.02478749</td>
</tr>
<tr>
<td>rel 8/MEP</td>
<td>0.00001294</td>
<td>0.00000000</td>
</tr>
<tr>
<td>rel 9/0RG</td>
<td>6.32082541</td>
<td>13.87477695</td>
</tr>
<tr>
<td>rel 9/IDC</td>
<td>0.23906659</td>
<td>0.02478749</td>
</tr>
<tr>
<td>rel 9/MEP</td>
<td>0.00001294</td>
<td>0.00000000</td>
</tr>
<tr>
<td>rel 10/0RG</td>
<td>6.32082541</td>
<td>13.87477695</td>
</tr>
<tr>
<td>rel 10/IDC</td>
<td>0.23906659</td>
<td>0.02478749</td>
</tr>
<tr>
<td>rel 10/MEP</td>
<td>0.00001294</td>
<td>0.00000000</td>
</tr>
<tr>
<td>IDCT/ORG</td>
<td>7.02090915</td>
<td>16.61680268</td>
</tr>
<tr>
<td>MEP/ORG</td>
<td>7.03829741</td>
<td>16.69795548</td>
</tr>
<tr>
<td>Relation</td>
<td>ORG</td>
<td>IDC</td>
</tr>
<tr>
<td>----------</td>
<td>-----</td>
<td>-----</td>
</tr>
<tr>
<td>1/ORG</td>
<td>7.03832156</td>
<td>16.69339138</td>
</tr>
<tr>
<td>1/IDC</td>
<td>0.08516566</td>
<td>0.00305704</td>
</tr>
<tr>
<td>1/MEP</td>
<td>0.05232799</td>
<td>0.00112153</td>
</tr>
<tr>
<td>2/ORG</td>
<td>7.03829957</td>
<td>16.69785573</td>
</tr>
<tr>
<td>2/IDC</td>
<td>0.06891978</td>
<td>0.00192993</td>
</tr>
<tr>
<td>2/MEP</td>
<td>0.00158706</td>
<td>0.00000100</td>
</tr>
<tr>
<td>3/ORG</td>
<td>7.03829729</td>
<td>16.69795292</td>
</tr>
<tr>
<td>3/IDC</td>
<td>0.06893255</td>
<td>0.00192880</td>
</tr>
<tr>
<td>3/MEP</td>
<td>0.00001704</td>
<td>0.00000000</td>
</tr>
<tr>
<td>4/ORG</td>
<td>7.03829727</td>
<td>16.69795405</td>
</tr>
<tr>
<td>4/IDC</td>
<td>0.06893291</td>
<td>0.00192879</td>
</tr>
<tr>
<td>4/MEP</td>
<td>0.00000589</td>
<td>0.00000000</td>
</tr>
<tr>
<td>5/ORG</td>
<td>7.03829727</td>
<td>16.69795406</td>
</tr>
<tr>
<td>5/IDC</td>
<td>0.06893291</td>
<td>0.00192879</td>
</tr>
<tr>
<td>5/MEP</td>
<td>0.00000592</td>
<td>0.00000000</td>
</tr>
<tr>
<td>6/ORG</td>
<td>7.03829727</td>
<td>16.69795406</td>
</tr>
<tr>
<td>6/IDC</td>
<td>0.06893291</td>
<td>0.00192879</td>
</tr>
<tr>
<td>6/MEP</td>
<td>0.00000592</td>
<td>0.00000000</td>
</tr>
<tr>
<td>7/ORG</td>
<td>7.03829727</td>
<td>16.69795406</td>
</tr>
<tr>
<td>7/IDC</td>
<td>0.06893291</td>
<td>0.00192879</td>
</tr>
<tr>
<td>7/MEP</td>
<td>0.00000592</td>
<td>0.00000000</td>
</tr>
<tr>
<td>8/ORG</td>
<td>7.03829727</td>
<td>16.69795406</td>
</tr>
<tr>
<td>8/IDC</td>
<td>0.06893291</td>
<td>0.00192879</td>
</tr>
<tr>
<td>8/MEP</td>
<td>0.00000592</td>
<td>0.00000000</td>
</tr>
<tr>
<td>9/ORG</td>
<td>7.03829727</td>
<td>16.69795406</td>
</tr>
<tr>
<td>9/IDC</td>
<td>0.06893291</td>
<td>0.00192879</td>
</tr>
<tr>
<td>9/MEP</td>
<td>0.00000592</td>
<td>0.00000000</td>
</tr>
<tr>
<td>10/ORG</td>
<td>7.03829727</td>
<td>16.69795406</td>
</tr>
<tr>
<td>10/IDC</td>
<td>0.06893291</td>
<td>0.00192879</td>
</tr>
<tr>
<td>10/MEP</td>
<td>0.00000592</td>
<td>0.00000000</td>
</tr>
</tbody>
</table>
REFERENCES


Mailaender, Laurence E. Interview by author, January 1986, at Stanford Telecommunications, McLean, VA.


