PROCEEDINGS OF SPIE

SPIEDigitalLibrary.org/conference-proceedings-of-spie

An Annihilation Transform Compression Method For Permuted Images

Robert M. Haralick, Norman C. Griswold, Craig A. Paul

Robert M. Haralick, Norman C. Griswold, Craig A. Paul, "An Annihilation Transform Compression Method For Permuted Images," Proc. SPIE 0087, Advances in Image Transmission Techniques, (13 December 1976); doi: 10.1117/12.954995

SPIE.

Event: 20th Annual Technical Symposium, 1976, San Diego, United States

87, 189-196 (1976)

AN ANNIHILATION TRANSFORM COMPRESSION METHOD FOR PERMUTED IMAGES

Robert M. Haralick
Norman C. Griswold*
Craig A. Paul

Remote Sensing Laboratory, University of Kansas
Lawrence, Kansas

Abstract

In the usual transform compression encoding schemes, an image is divided into regular shaped subimages or blocks, a transform is performed on each block, low energy components thrown away, remaining components encoded and transmitted, and then the received image reconstructed. This paper discusses ways in which the image pixels can be permuted before the image is blocked and a special annihilation transform technique to perform the image coding. Experimental results show the compressed images to have no blocking effects, but a more mottled appearance compared to a discrete cosine transform coding method. The annihilation method on permuted images not only gives reconstructed images better visual quality, but also gives lower RMS error.

Introduction

In the usual transform coding technique, the image is partitioned into regular shaped continuous windows, a fast transform is applied to each window and the high energy components in the transform domain are coded for transmission. In some schemes the mean window is subtracted before transforming and then added back in the reconstruction. Because the high energy components tend to be the low complexity ones (low frequency or sequency), and because the windows are each handled independently, the reconstructed images tend to take on a smooth-blurred appearance with edge lines (blocking) occurring between adjacent windows. In this paper we report the initial results of an experiment designed to scatter the blocking error throughout the image.

The basic idea is to permute the resolution cells of the image before partitioning it into blocks. In this manner non-adjacent resolution cells scattered across the image can come together to form a window which can be transform coded. Of course, this means that the spatial dependency existing between resolution cells in the usual continuous window approach will be gone and we would, therefore, suspect that transform coding done in the usual manner would not work. Certainly, selecting resolution cells at random is not the thing to do. Thus we are led to select a regular permutation in which each window has resolution cell groups spaced at uniform intervals on the original image. Each window, then, is a little replica of the image.

Since the energy is no longer likely to be concentrated in a few basis vectors of the usual fast transforms we have to look for a data dependent set of basis vectors. Eliminating the Karhunen-Loeve transform from consideration because of its excessive complexity, and noticing the cyclic nature of the permuted image, we were led to consider the permuted windows themselves for the basis vectors of the transformation.

The following sections describe the mathematical details of the transformation method which we have named "transformation by annihilation," and the fourth section gives some initial results of the method with some tentative conclusions.

Method of Annihilation

The method of annihilation is based on the idea that successive applications of orthogonal projection operators to a vector will eventually annihilate it. This is illustrated in Figure 1, where successive applications of orthogonal projection operators P_1 and P_2 which project to subspaces V_1 and V_2, respectively, nearly annihilate the vector y_o in the following way: the first orthogonal projection is to the subspaces V_1 and $V_1^\perp$. The vector y_o becomes the partially annihilated vector $(I-P_1)y_o$ which lies in $V_1^\perp$ and the partial reconstruction becomes $r_1 = P_1 y_o$ which lies in V_1. The second orthogonal projection is to the subspaces V_2 and $V_2^\perp$. The vector y_1 becomes the partially annihilated vector $y_2 = (I-P_2)y_1$ and the partial reconstruction becomes $r_2 = r_1 + P_2 y_1$. Notice that

$$\|y_2\| = \|y_o - r_2\| \leq \|y_1\| = \|y_o - r_1\| \leq \|y_o\| \tag{1}$$

indicating that in the annihilation process, what remains of y_o becomes less and less, while the reconstruction gets closer and closer to y_o. The algebraic reason why the reconstruction process works is due to the operator identity

$$I = \prod_{k=1}^{K} (I-P_k) + \sum_{k=1}^{K} P_k \prod_{n=1}^{k-1} (I-P_n) \tag{2}$$

which is proved in Proposition 1.

*Dr. Griswold is currently employed at the Wright Patterson Air Force Base, Dayton, Ohio.

The term $\prod_{k=1}^{K} (I-P_k)$ represents the operator which is trying to annihilate the vector y_o. The term $\sum_{k=1}^{K} P_k \prod_{n=1}^{k-1} (I-P_n)$ represents the operator which gives the reconstruction of y_o from the sum of the successive projections. Propositions 2 and 3 prove the successive partial annihilations get closer to 0 and that the successive partial reconstructions get closer to y_o.

This technique of annihilation and reconstruction can be applied to the transform image coding situation in the following way. Take the image to be coded and block it up into the permuted windows as shown in Figure 2. This permutation operation makes each window a little replica of the entire image. Now consider the set of grey tones in each window as a vector. Because the windows all look alike, the set of vectors can be thought of as a tight ellipsoidal cluster of vectors in Euclidean space. Call these vectors $x_1,\ldots,x_N$. Translate this ball or cluster of vectors to the origin by subtracting its mean. Let $\mu = \frac{1}{N}\sum_{n=1}^{N} X_n$ and define $y_n^\circ = X_n - \mu$, and $r_n^\circ = \mu$. The vectors $y_1^\circ,\ldots,y_N^\circ$ represent what remains of the image to be annihilated, and the vectors $r_1^\circ,\ldots,r_N^\circ$ represent the first iteration reconstructions. Suppose that the k^{th} iteration vectors y_n^k and r_n^k, $n=1,\ldots,N$ have been defined. Define the $(k+1)^{th}$ iteration vectors by:

Let m be any index satisfying $\|y_m^k\| \geq \|y_n^k\|$, $n=1,\ldots,N$. Define $b_k = y_m^k/\|y_m^k\|$. Define the partial annihilations and reconstructions by:

$$y_n^{k+1} = y_n^k - (b_k' y_n^k) b_k$$

$$r_n^{k+1} = r_n^k + (b_k' y_n^k) b_k \tag{3}$$

The vector b_k is the vector of unit norm in a direction of some highest energy vector of the partially annihilated image. Since the vectors of the partially annihilated image tend to form an ellipsoidal ball in the Euclidean space, the vector having the highest energy will tend to be in a direction which is along the major axis of the ellipsoid. A priori, therefore, the vector of the partially annihilated image having highest energy should be in a direction which will subtract the most energy from the partially annihilated image and add the most energy to the partially reconstructed image.

The term $b_k' y_n^k b_k = b_k b_k' y_n^k$ represents the orthogonal projection of y_n^k onto the subspace spanned by the vector b_k. This can be easily verified by noticing that

$$(b_k b_k') = b_k b_k' \quad \text{(symmetry)}$$

$$(b_k b_k')(b_k b_k') = b_k (b_k' b_k) b_k' = b_k b_k' \quad \text{(idempotency)} \tag{4}$$

which are the two conditions required for an operator to be an orthogonal projection operator. The partial reconstruction of r_n^k is, therefore, incremented by the orthogonal projection of y_n^k onto b_k and the same projection vector is subtracted from the partial annihilation, bringing it closer to the origin. Proposition 4 proves that the ratio of the energy in the partial reconstructions to the energy in the partial annihilation increases in successive iterations.

Annihilation Method Compression Ratio and Computational Complexity

Suppose that the image has W windows, each having S resolution cells. There are, then, WS numbers in the image. Suppose that the annihilation method proceeds for K iterations. To transmit the mean and each of the K basis windows (this is the overhead) takes a total of (K+1)S numbers and to transmit the transformed windows takes KW numbers. The total of numbers involved in the compressed data is (K+1)S + KW and the compression ratio is, therefore,

$$\frac{WS}{(K+1)S + KW}. \tag{5}$$

If there were no overhead, the compression ratio would be S/K. The number of iterations that are done in the annihilation method is analogous to the number of high energy components selected in a usual fast transform method. The formula for the compression ratio clearly shows the dependence of the size of the image on the compression ratio. As the number of windows gets larger, the compression ratio tends to S/K. As the number of windows gets smaller, the overhead of transmitting the mean window and selected basis vectors takes a larger and larger percentage of the transmitted bits.

To subtract the mean window requires about 2WS operations, WS adds to compute the mean and WS subtracts to subtract it out. Each iteration entails computing the energy for each window (2WS operations), selecting the window having maximum energy, normalizing it, and taking the dot product of it with each window (2WS operations). Thus for K iteration, there are 2(K+1)WS operations.

A discrete cosine fast transform technique would involve at best $2S \log_2 S$ operations per window to take the transform and these are W windows. This makes a total of $2WS \log_2 S$. When S = 256 and K = 20, the annihilation method is at worst no more than three times more complex in the amount of computations required.

Results

To test this idea of permuting an image and compressing it by a method of annihilation, we selected a video quality image of a truck on a bridge. The scene is one in the class of images a remotely piloted vehicle might see. The image was the central 160 X 160 selected from a 512 X 512 scene. The truck image is shown in Figure 3a. Taking every 10^{th} resolution cell in a line and every 10^{th} line, we created the permuted image blocked into 16 X 16 windows, shown in Figure 3b. Compression began by subtracting the mean permuted window from each permuted block on the permuted image. Then the annihilation transform method was applied for 20 iterations for an approximate 12:1 component compression ratio (neglecting overhead). Taking into account the overhead on this 160 X 160, the compression ratio would fall to about 4:1. Almost three times as many numbers would be required to transmit the mean window and basis vectors than is required to transmit the transformed windows. For the RPV situation, the overhead should be computed for a 512 X 512 image. For a 20 iteration annihilation where 20 transformed components out of 256 are selected, the compression ratio for the 512 X 512 image is about 10:1. Note that in all these calculations, no attempt has been made to optimally bit encode the transformed values. The 10:1 reconstructed annihilated image is shows in Figure 3c.

For comparison purposes, a two-dimensional discrete cosine transform coding method was tried. Here, the original image was blocked into 16 X 16 windows and the mean window subtracted out. The two dimensional discrete cosine transform of each block was computed by doing a 16 X 1 discrete cosine by rows and then by columns. Selecting the 25 highest energy components for transmission, the reconstructed compressed image is shown in Figure 3d; this is also about a 10:1 compression.

Examining Figures 3c and 3d to make a visual comparison, we find that the discrete cosine method has a smoother appearance and very definite blocking showing on the edges of some windows. It tends to smear high contrast objects and its lines tend to be more continuous. The annihilation method gives an image having a mottled texture, whose contrast is better, and whose geometry of high contrast objects is better. The overall visual quality of the annihilation transformed image is better.

Figure 3e shows the error image from the annihilation method. Black indicates a negative error, white indicates a positive error, and grey indicates little or no error. Most of the error is concentrated around the boundary of the truck.

Figure 4 shows a graph of the energy retained in the reconstructed image as a function of the number of transform components selected. The annihilation method has the advantage in the beginning because the mean window of the permuted image has more energy (565,438) than the mean window of the unpermuted image (484,247). However, as the graph shows, the annihilation method retains the energy advantage for any value of selected components.

We wondered whether the discrete cosine could go as well as the annihilation method on the permuted image. Results showed the discrete cosine having a much more uniform distribution of energy for its components and consequently a much higher error. For example, the percentage energy retained by the 25 highest energy components of the discrete cosine transform operating on the permuted image was 97.56% compared to the 99.17% energy the discrete cosine transform retained for its 25 highest energy components when operating on an unpermuted image.

Likewise, the annihilation does not work as well on the unpermuted image. Results showed that the percentage energy retained by the first 13 components generated was 98.54% compared to the 98.92% energy retained by the annihilation method when operating on the permuted image. This corresponds to a RMS error of 5.79 for the annihilation on the unpermuted image compared to a RMS error of 4.97 for the annihilation on the permuted image.

Table 1 compares the energy retained, RMS error, and compression ratio of the annihilation method against the discrete cosine transform. Note that the compression ratio computation takes into account all the overhead associated with each method and is calculated assuming a 512 X 512 image. The table shows the clear superiority of the annihilation method over the discrete cosine transform method. For example, at about a 10:1 compression ratio the annihilation method has a 4.1865 RMS error while the discrete cosine method has a 4.3772 RMS error. The energy preserved by the annihilation method is 99.24% while the energy preserved by the discrete cosine is 99.17%. The annihilation method gives about a 5% improvement in RMS error.

Initial results with the annihilation transform method show it outperforms the discrete cosine transform in terms of RMS error and visual quality. Further experiments should be tried with different variations on permutations. Perhaps a hybrid discrete cosine/annihilation method transform could be designed which would do better than either of them singly. We hope to report on just such experiments in the near future.

Proposition 1: Let $P_1,\ldots,P_K$ be operators and I be the identity operator. Then,

$$I = \prod_{k=1}^{K} (I-P_k) + \sum_{k=1}^{K} P_k \prod_{n=1}^{k-1} (I-P_n).$$

Proof: The proof is by induction on K. Certainly the equation is true for $K = 1$ since

$$I = I - P_1 + P_1.$$

Now suppose it is true for $K = N$. We will show it is true for $K = N + 1$.

$$\prod_{k=1}^{N+1} (I-P_k) + \sum_{k=1}^{N+1} P_k \prod_{n=1}^{k-1} (I-P_n) = (I-P_{N+1}) \prod_{k=1}^{N} (I-P_k) + P_{N+1} \prod_{n=1}^{N} (I-P_n) + \sum_{k=1}^{N} P_k \prod_{n=1}^{k-1} (I-P_n)$$

$$= (I-P_{N+1}+P_{N+1}) \prod_{n=1}^{N} (I-P_n) + \sum_{k=1}^{N} P_k \prod_{n=1}^{k-1} (I-P_n)$$

$$= \prod_{n=1}^{N} (I-P_n) + \sum_{k=1}^{N} P_k \prod_{n=1}^{k-1} (I-P_n).$$

By the supposition that the equation is true for $K = N$, we have that the right-hand side is the identity operator I.

Proposition 2: Let P be an orthogonal projection operator. Then $\|PX\| \leq \|X\|$.

Proof: $\|X\|^2 = \|((I-P)+P)X\|^2$

$$= X'((I-P)+P)'((I-P)+P)X$$

$$= X'(I-P)'(I-P)X + X'(I-P)'PX + X'P'(I-P)X + X'P'PX$$

But $(I-P)'P = (I-P)P = P - PP = P - P = 0$. So,

$$\|X\|^2 = \|(I-P)X\|^2 = \|(I-P)X\|^2 + \|PX\|^2.$$

Since $\|(I-P)X\|^2 \geq 0$, $\|X\|^2 \geq \|PX\|^2$ and $\|PX\| \leq \|X\|$.

Proposition 3: Let $P_1,\ldots,P_{K+1}$ be orthogonal projection operators.

Then $\|(I - \prod_{k=1}^{K} (I-P_k))X\| \leq \|(I - \prod_{k=1}^{K+1} (I-P_k))X\|$.

Proof: $I - \prod_{k=1}^{K+1} (I-P_k) = I - P_{K+1} + P_{K+1} - \prod_{k=1}^{K+1} (I-P_k)$

$$= (I-P_{K+1}) - (I-P_{K+1}) \prod_{k=1}^{K} (I-P_k) + P_{K+1}$$

$$= (I-P_{K+1}) [I - \prod_{k=1}^{K} (I-P_k)] + P_{K+1}$$

$$\|(I - \prod_{k=1}^{K+1} (I-P_k))X\|^2 = \|((I-P_{K+1}) [I - \prod_{k=1}^{K} (I-P_k)] + P_{K+1})X\|^2$$

$$= \|(I-P_{K+1}) [I - \prod_{k=1}^{K} (I-P_k)]X\|^2 + \|P_{K+1}X\|^2$$

$$= \|(I - \prod_{k=1}^{K} (I-P_k))X\|^2 - \|P_{K+1}(I - \prod_{k=1}^{K} (I-P_k))X\|^2 + \|P_{K+1}X\|^2$$

$$= \|(I - \prod_{k=1}^{K} (I-P_k))X\|^2 - \{\|P_{K+1}X\|^2 - 2X'P_{K+1} \prod_{k=1}^{K} (I-P_k) + \|P_{K+1} \prod_{k=1}^{K} (I-P_k)X\|^2\} + \|P_{K+1}X\|^2$$

$$\geq \|(I - \prod_{k=1}^{K} (I-P_k))X\|^2 - 2\ \|X\|\ \|P_{K+1} \prod_{k=1}^{K} (I-P_k)X\| + \|P_{K+1} \prod_{k=1}^{K} (I-P_k)X\|^2$$

$$\geq \|(I - \prod_{k=1}^{K} (I-P_k))X\|^2 + \|P_{K+1} \prod_{k=1}^{K} (I-P_k)X\|\ [2\|X\| - \|P_{K+1} \prod_{k=1}^{K} (I-P_k)X\|]$$

Since when P is an orthogonal projection operator, $\|PX\| \leq \|X\|$; hence,

$\|P_{K+1} \prod_{k=1}^{K} (I-P_k)X\| \leq \| \prod_{k=1}^{K} (I-P_k)X \| \leq \ldots . \leq \|X\|$. So, $2\|X\| - \|P_{K+1} \prod_{k=1}^{K} (I-P_k)X\| \geq 2\|X\| - \|X\| = \|X\|$.

Therefore, $\|(I - \prod_{k=1}^{K+1} (I-P_k))X\|^2 \geq \|(I - \prod_{k=1}^{K} (I-P_k))X\|^2 + \|P_{K+1} \prod_{k=1}^{K} (I-P_k)X\|\ \|X\|$

$$\geq \|(I - \prod_{k=1}^{K} (I-P_k))X\|^2$$

and we must have $\|(I - \prod_{k=1}^{K} (I-P_k))X\| \leq \|(I - \prod_{k=1}^{K} (I-P_k))X\|$.

Proposition 4: Let $P_1,\ldots,P_{K+1}$ be orthogonal projection operators and I be the identity operator. Then,

$$\frac{\|\sum_{k=1}^{K} P_k \prod_{n=1}^{k-1} (I-P_n)X\|}{\| \prod_{n=1}^{K} (I-P_n)X\|} \leq \frac{\|\sum_{k=1}^{K+1} P_k \prod_{n=1}^{k-1} (1-P_n)X\|}{\| \prod_{n=1}^{K+1} (I-P_n)X\|}$$

Proof: By proposition 1, $I = \prod_{k=1}^{K} (I-P_k) + \sum_{k=1}^{K} P_k \prod_{n=1}^{k-1} (I-P_n)$

so we must have $I - \prod_{k=1}^{K} (I-P_k) = \sum_{k=1}^{K} P_k \prod_{n=1}^{k-1} (I-P_n)$.

By proposition 2 since $(I-P_{K+1})$ is an orthogonal projection operator,

$$\|(I-P_{K+1})\ [\prod_{n=1}^{K} (I-P_n)X]\| \leq \| \prod_{n=1}^{K} (I-P_n)X\|.$$

By proposition 3, since $P_1,\ldots,P_{K+1}$ are orthogonal projection operators

$$\|(I - \prod_{k=1}^{K} (I-P_k))X\| \leq \|(I - \prod_{k=1}^{K+1} (I-P_n))X\|.$$

Now by dividing the inequality we obtain

$$\frac{\|(I - \prod_{k=1}^{K} (I-P_k))X\|}{\|\prod_{n=1}^{K} (I-P_n)X\|} \leq \frac{\|(I - \prod_{k=1}^{K+1} (I-P_n))X\|}{\|\prod_{n=1}^{K+1} (I-P_n)X\|}$$

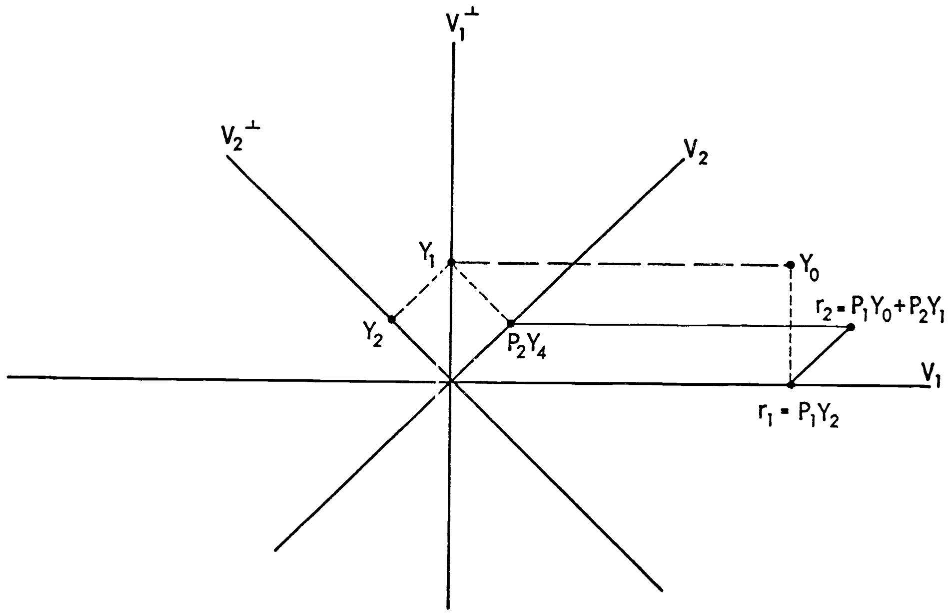
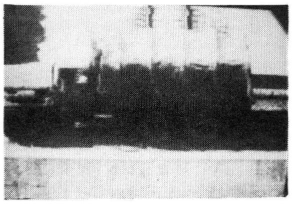


Figure 1 illustrates the geometry of the successive applications of orthogonal projection operators to annihilate and reconstruct the vector Y_0. After two projections Y_0 has become Y_2 and the partial reconstruction has become $r_1 = P_1Y_0 + P_2Y_1$, a vector close to Y_0. The distance between Y_0 and r_2 is precisely the norm $\|Y_2\|$.

1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24
25	26	27	28	29	30
31	32	33	34	35	36

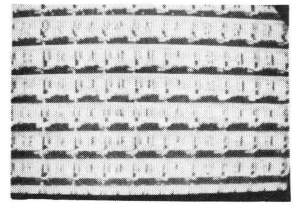
1	4	2	5	3	6
19	22	20	23	21	24
7	10	8	11	9	12
25	28	26	29	27	30
13	16	14	17	15	18
31	34	32	35	33	36

Figure 2 illustrates how an image is permuted and blocked into small windows which each will look like a replica of the image. The left part of the figure labels each resolution cell in the image. The right part of the figure shows the blocking and permuting.



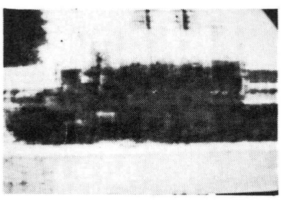
a

ORIGINAL TRUCK



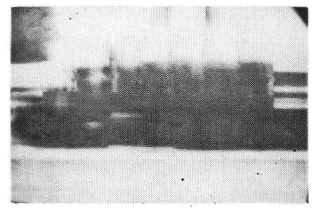
b

PERMUTED TRUCK



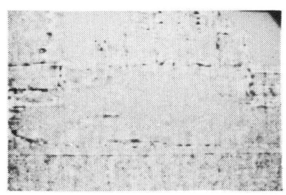
c

10:1 COMPRESSION
ANNIHILATION METHOD



d

10:1 COMPRESSION
DISCRETE COSINE



e

ERROR IMAGE
ANNIHILATION METHOD

Figure 3 shows the original truck scene (a) the permuted truck image (b), a 10:1 compression by the annihilation method (c), a 10:1 compression by the discrete cosine transform (d) and the error image from the annihilation method (e).

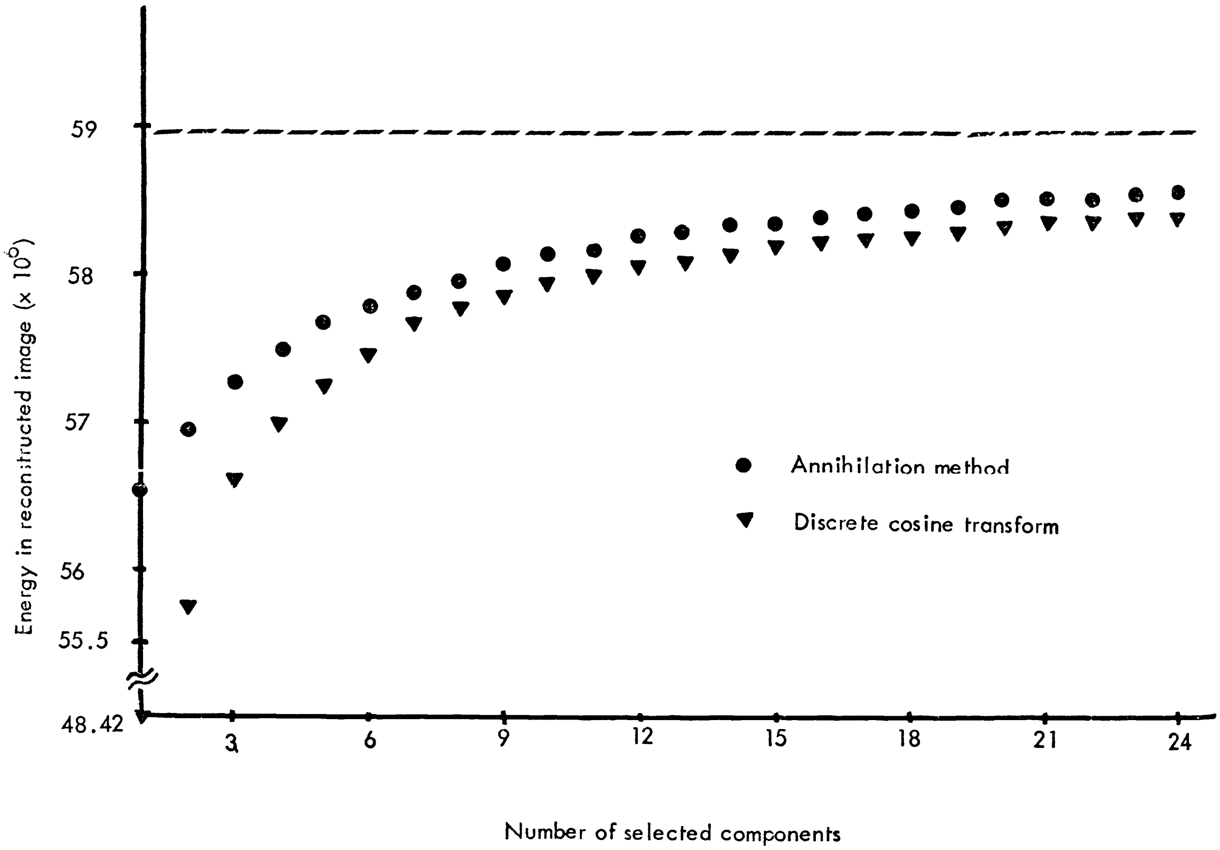


Figure 4 compares the energy in the reconstructed truck image as a function of the number of transform components selected. The annihilation method has the advantage in the beginning and retains it for any value of selected components. The asymptote at about $58.95 * 10^6$ is the total energy in the truck image.

# Components Selected	Compression Ratio		RMS Error		% Energy Preserved	
	Annihilation	Discrete Cosine	Annihilation	Discrete Cosine	Annihilation	Discrete Cosine
16	12.64	15.74	4.6199	5.2820	99.07	98.79
20	10.13	12.63	4.1865	4.9288	99.24	98.95
21	9.66	12.04	4.0901	4.6217	99.27	99.07
25	8.13	10.12	3.7330	4.3772	99.39	99.17
33	6.17	7.69	3.0809	4.0377	99.59	99.29

Table 1 Compares the Annihilation Method with the Discrete Cosine Transform Method. The compression ratio computation takes into account all overhead associated with each method and is calculated assuming a 512 X 512 image. No attempt was made to do optimal bit assignment which would, of course, increase the compression ratios.

This work was sponsored in part by the National Science Foundation Grant ENG 74-18981.