Comparison of Transform Coding Techniques for Two-Dimensional Arbitrarily-Shaped Images[1]

Shih-Fu Chang

Dept. of Electrical Engineering
Columbia University
New York, NY 10027

sfchang@ctr.columbia.edu

David G. Messerschmitt

Dept. of EECS
University of California at Berkeley
Berkeley, CA 94720

messer@eecs.berkeley.edu

Nov. 1993

(for publication on *ACM/Springer-Verlag Multimedia Systems* journal)

1. Part of this work was presented in ACM Multimedia Conference, Anaheim, Aug. 1993.

Abstract

Envisioned advanced multimedia video services include arbitrarily-shaped (AS) image segments as well as regular rectangular images. Images segments of the TV weather reporter produced by the chromo-key technique [1] and image segments produced by video analysis and image segmentation[2,3,4] are typical examples of AS image segments. This paper explores efficient intraframe transform coding techniques for general two-dimensional (2D) AS image segments, treating the traditional rectangular images as a special case. In particular, we focus on transform coding of the partially-defined image blocks along the boundary of the AS image segments.

We recognize two different approaches — the *brute-force* transform coding approach and the *shape-adaptive* transform coding approach. The former fills up the uncovered area with the optimal redundant data such that the resulting transform spectrum is compact. A simple but efficient mirror-image extension technique is proposed. Once augmented into full image blocks, these boundary blocks can be processed by traditional block-based transform techniques like the popular Discrete Cosine Transform (DCT). In the second approach, we change either the transform basis or the coefficient calculation process adaptively based on the shape of the AS image segment. We propose an efficient *shape-projected* problem formulation to reduce the dimension of the problem. Existing coding algorithms, such as the orthogonal transform by Gilge [5] and the iterative coding by Kaup and Aach [6], can be intuitively interpreted. We also propose a new adaptive transform basis by applying the same principle as that used in deriving the DCT from the optimal Karhunen-Loeve Transform (KLT).

We analyze the tradeoff relationship between compression performance, computational complexity, and codec complexity for different coding schemes. Simulation results show that complicated algorithms (e.g. iterative, adaptive) can improve the quality by about 5-10 dB at some computational or hardware cost. On the other hand, the simple mirror-image extension technique improves the quality by about 3-4 dB without any overheads. The contributions of this paper lie in efficient problem formulations, new transform coding techniques, and numerical tradeoff analyses.

Keywords: transform coding, arbitrarily-shaped image segments, object-oriented video coding, structured video.

1. Introduction

In envisioned advanced multimedia video services, displayed video objects can in general be rectangular (e.g. window graphic interface) or arbitrarily-shaped (AS) (e.g. chroma-keyed TV weather reporter) [1,7]. In multimedia editing systems, users can create AS video objects manually or by segmentation algorithms. Users can then manipulate each individual video object or composite multiple video objects together. In the so-called object-oriented video coding algorithms, AS video objects are segmented, compressed, and transmitted separately [5,6]. Separate video objects are composited together at the receiver to reconstruct the original video signal. Figure 1 illustrates a block diagram for the AS video object editing system we are currently prototyping. After AS video objects are extracted, we need to encode their shape and internal pixel values, and perform anti-aliasing along the boundary to remove the "jagged" artifacts. Then, we can manipulate the AS video objects in the desired fashion, such as animation and compositing.

A complete representation of AS video objects includes two parts — *shape* and *image pixel values*. The former represents the boundary information of the object; the latter represents the internal color intensity variation. Both these two components are required for general manipulation of AS video objects, such as overlap, translation, and scaling [8,9]. In this paper, we focus on designing efficient representations of the image pixel values to achieve satisfactory compression and image quality. In particular, we look at intraframe block-wise transform coding, such as the widely used Discrete Cosine Transform (DCT) [10,11,12,13]. Note that we assume the shape information is encoded and transmitted separately from the pixel values. At the receiver, the shape information is needed for reconstructing or manipulating the AS image segments. Figure 2 shows a diagram which describes the relationship between the coding of the shape and the image pixels. We need to mask the reconstructed image pixels with the received shape information in order to reconstruct the original AS image segment. Efficient algorithms for encoding the shape information and anti-aliasing are described in [14].

In block-wise transform coding algorithms, images are separated into small blocks with fixed size, say N pixels by N pixels. Figure 3 shows an example of an AS video object (Miss USA) and illustrates the concept of block structure. In internal blocks, all pixel values are defined. The traditional DCT algorithm can be used to encode these blocks efficiently. However, for the boundary blocks, the pixel values are not completely defined. One straightforward approach is to fill zero values outside the boundary and treat the image block as traditional image blocks. A drawback of this approach is the potential significant increase of the high-order transform coefficients and thus potential serious degradation of the compression performance.

In this paper, we investigate two classes of transform coding techniques — *brute-force full-block transform* and *shape-adaptive transform*. The former explores innovative ways of filling the redundant data in the uncovered area of the boundary blocks and then take the traditional full-block DCT. The latter changes the transform basis or the coefficient computation process adaptively based on the shape of the input block. The iterative approximation method proposed by Kaup and Aach [6] and the adaptive orthogonal transform proposed by Gilge *et al.* [5] belong to this class of techniques. We propose an efficient problem formulation — the shape-projected domain, base on which we can easily interpret existing adaptive transform techniques and derive new ones. We derive a new KLT-like transform basis to demonstrate the flexibility of the proposed formulation.

Afterwards, we compare the performance of different transform coding techniques and illustrate the tradeoff relationship between the compression performance, computational

complexity, and codec complexity. The contributions of this paper lie in an efficient problem formulations, new transform coding techniques, and numerical tradeoff analyses.

2. The Brute-Force Full-Block Transform

As mentioned earlier, image segments are separated into small blocks, e.g. N pixels by N pixels each. For AS image segments, boundary blocks usually have pixel values partially defined. Let P(x,y) represent the pixel values within this N pixel × N pixel block area, called *R*. Let *B* represent the occupied region within the block, as shown in Figure 4. A partially-filled image block has P(x,y) defined within region *B*. The brute-force full-block transform coding technique fills up the redundant area outside the boundary and then utilize the traditional block-wise transform coding.

Once the image data, P(x,y), is extended to the full block, we can use traditional block-wise transform coding to represent the block as follows,

$$P(x,y) = \sum_i a_i \cdot f_i(x,y) \quad ; \text{x,y} \in R \qquad \text{(EQ 1)}$$

where f_i's are *basis functions* defined on the full-block area, *R*. Namely, P(x,y) is transformed to a set of coefficients a_i, which can be used to completely or partially reconstruct the original image. For the purpose of compression, we would like to use as small number of coefficients as possible to obtain an accurate reconstruction, $\hat{P}(x,y)$. The resulting error term is defined as

$$error = \sum (P(x,y) - \hat{P}(x,y))^2 \ , \text{x, y} \in B \qquad \text{(EQ 2)}$$

Note the summation is executed over the occupied region only because error terms outside the boundary are discarded when we apply the shape information at the receiver.

If we fix the choice of basis functions, e.g. use the N×N DCT basis functions, the objective can be interpreted as finding the optimal P(x,y) values outside region *B* so that the transform coefficients, a_i, present the highest energy compaction. The concept is illustrated in Figure 4. However, it is difficult to quantitatively formulate this abstract property of *energy compactness*. An example discussed in [6] uses the entropy definition

$$-\sum_i \left(\left(\frac{|f_i|}{\sum |f_i|} \right) \cdot \log \left(\frac{|f_i|}{\sum |f_i|} \right) \right) \qquad \text{(EQ 3)}$$

to emulate the energy compactness of the transform spectrum. The problem with this definition is that the final choice usually ends up with few large spectrum components, which may cause over-flow problems, though the spectrum "entropy" is low. Furthermore, it's difficult to find the optimal solution which minimizes the above entropy term.

2.1 Mirror Image Extension

Despite the difficulty in quantifying the compactness of spectrum coefficients, the approach of filling the region outside the boundary with optimal redundant data does provide a

freedom for us to optimize the transform spectrum. The simplest method to augment a partially defined image segment into a full block image is stuffing zero's outside the image boundary. However, it is well known that this method may introduce sharp edges on the boundary and thus high-frequency components in the transform spectrum.

The traditional band-limited extrapolation approach assumes the input signal has a limited bandwidth the same size as the defined signal samples [15, 4]. However, the goal of this approach is to find remaining undefined signal samples rather than to achieve good data compression. The resulting spectrum may not be compressible at all. That is, all spectrum coefficients may turn out to be non-trivial. In addition, it may have the singularity problem in solving the associated linear system if the spectrum passband is fixed.

One promising alternative is to extend each image segment with its "*mirror image*" outside the image boundary. This approach has been used in literatures to solve the finite-extent issue in subband coding of images [17]. Figure 5 shows a partially defined block and its extension in one dimension. In general, the defined pixels may not occupy exactly one half of the block. We may need to duplicate the given pixel values several times and truncate it at the block boundary. For a 2D image segment, we can apply this 1D mirror image extension in one direction first, and then in another direction. This mirror-image extension technique is simple but efficient. Its compression performance will be described later.

3. Shape-Adaptive Approach

As described in Equation 2, we are only concerned with reconstruction errors within the image boundary, i.e. errors within the covered region *B*. An equivalent but perhaps more efficient approach to finding the optimal representation of AS image segments is to perform optimization only in the subspace defined over region *B*, denoted by S_B. Basically, we project the AS image segment and all basis functions into the subspace S_B and find the optimal representation there. The redundant pixel values and their associated errors outside the boundary can thus be automatically ignored. However, since the subspace varies with the image shape, the optimal transform bases for different image shapes may also be different. This is the reason why this is called the *shape-adaptive* approach.

3.1 The Shape-Projected Subspace

Instead of filling data outside the image boundary and applying a full-block rectangular transform, we can focus on the defined image pixels only, i.e. P(x,y) values within region *B*. Mathematically, let's define S_R as the linear space spanned over the entire square block *R*, and S_B as the subspace spanned over the covered region *B*. For example, in Figure 6, space S_R has a dimension equal to 16, while the dimension of subspace S_B is equal to 4. One possible basis for subspace S_B is shown in Figure 6(b). Each basis matrix has a single non-zero element.

Every arbitrarily-shaped image segment, P(x,y), can be considered as a vector in S_B. To completely represent this vector, we need to find a complete set of independent vectors, say $\{b_i\}$, in S_B and describe P(x,y) as a linear combination of b_i's. The distinction between this approach and that in the previous section is that the entire problem domain now is confined only in the subspace S_B. We don't have to worry about the redundant data outside the image boundary, i.e.

vector components outside the subspace S_B. If we want to use traditional block-based transform bases, say f_i (e.g. the DCT basis), we can project these basis functions into subspace S_B,

$$\hat{f}_i = Project(f_i, S_B) \quad \text{(EQ 4)}$$

and describe vector P(x,y) as a linear combination of $\hat{f}_i$'s. In actuality, the above projection simply removes the components of f_i outside subspace S_B.[1] Figure 7 illustrates an example when the dimension of S_B equals two.

The important issue that remains is the problem of finding optimal basis functions in subspace S_B such that we can use the least number of coefficients to reconstruct the image segment vector with satisfactory errors. The above formulation does provide a flexible platform to derive new transform bases and evaluate their performance. We will describe some existing approaches and propose a new approach in the following subsections.

3.2 Successive Approximation Algorithms Revisited

Using the existing full-block 2D DCT basis to represent AS image segments is attractive since existing decoders for rectangular images can be used without modifications. However, as described earlier, the shape-projected DCT basis functions, $\{\hat{d}_i\}$, are generally mutually dependent and not orthogonal. There are multiple solutions for Equation 1 if $\{\hat{d}_i\}$ are used as the basis functions. Instead of finding a fully accurate representation, Kaup and Aach [6] proposed a successive approximation method to calculate only the most significant coefficients. In this section, we first briefly review Kaup and Aach's approach based on our shape-projected subdomain formulation. Then, we apply their technique to constant-rate and constant-quality compression. Some subtle issues imposed by quantization of transform coefficients are also addressed.

3.2.1 Perfect Reconstruction vs. Non-Perfect Reconstruction

If a linear representation can reconstruct the original function without any error, it is a *perfect-reconstruction* (PR) representation. Otherwise, it's a *non-perfect-reconstruction* (non-PR) representation. If the subspace spanned by the representation basis functions can not contain the image segment vector, then the PR property cannot be achieved. As mentioned earlier, the shape-projected DCT basis functions $\{\hat{d}_i\}$ form a mutually *dependent* but *complete* set of vectors in the shape-projected subspace. We should be able to choose *m* independent vectors from the projected DCT vectors to form a basis in the subspace and achieve the PR property, where *m* is the dimension of the shape-projected subspace. The issue now is to find the basis that can produce the best energy compaction.

Kaup and Aach used a successive approximation algorithm to iteratively project the image vector to each basis function and choose the basis function with the largest projection during each iteration, i.e.,

$$Project(r(n), \hat{d}_{opt}) = \max_i (Project(r(n), \hat{d}_i)) \quad \text{(EQ 5)}$$

1. Another interpretation of projection is to force all those component values of f_i outside S_B to be zero.

$$r(n+1) = r(n) - Project(r(n), \hat{d}_{opt}) \quad \text{(EQ 6)}$$

where $r(n)$ is the residual error in the n*th* iteration, $\{\hat{d}_i\}$ are the shape-projected transform basis functions (e.g., the shape-projected DCT basis functions), and $\hat{d}_{opt}$ is the optimal basis function with the largest projection in each iteration. Note that the same basis function could be chosen repetitively since $\{\hat{d}_i\}$ are not orthogonal. However, one problem is that the number of transform coefficients may exceed m (the image segment size) without achieving the PR property.

One way to achieve the PR property in the above iterative algorithm is to accumulate the chosen basis functions in each iteration and project the image segment to the entire set of chosen basis function, rather than to a single basis function only. Namely, the following operations are performed during each iteration.

$$D_i(n) = \text{Union}(D_{opt}(n-1), \hat{d}_i) \quad \text{(EQ 7)}$$

$$Project(r_0, D_{opt}(n)) = \max_i (Project(r_0, D_i(n))) \quad \text{(EQ 8)}$$

$$r(n+1) = r_0 - Project(r_0, D_{opt}(n)) \quad \text{(EQ 9)}$$

where $D_{opt}(n)$ is the entire set of optimal basis functions accumulated from iteration 1 to n, $r(n)$ is the residual error after n iterations, and r_0 is the initial residual error, i.e. the original image segment. In each iteration, we keep the set of basis functions chosen from last iteration and add an additional basis function to minimize the residual error (i.e. maximizing the projection). The dimension of subspace spanned by the basis functions is incremented by one in each iteration[1]. Note that in order to find the optimal basis function during each iteration, we need to project the image segment vector to every possible set of basis functions, each of which requires solving a complete system of linear equations. This computation overhead is significant.

Another interpretation of the above PR iterative approximation algorithm is that during each iteration we not only add a new basis function, but also make the remaining unchosen basis functions and the residual error orthogonal to the chosen basis functions by projection, i.e.,

$$\hat{d}_{opt}(n) = \max_i (Project(r(n), \hat{d}_i(n))) \quad , \forall \quad \hat{d}_i \notin D_{opt} \quad \text{(EQ 10)}$$

$$D_{opt}(n) = \text{Union}(D_{opt}(n-1), \hat{d}_{opt}(n)) \quad \text{(EQ 11)}$$

$$\hat{d}_i(n+1) = \hat{d}_i(n) - Project(\hat{d}_i(n), \hat{d}_{opt}(n)) \quad , \forall \quad \hat{d}_i \notin D_{opt} \quad \text{(EQ 12)}$$

$$r(n+1) = r(n) - Project(r(n), \hat{d}_{opt}(n)) \quad \text{(EQ 13)}$$

where $\hat{d}_{opt}(n)$ is the new basis function added to the chosen set D_{opt} in iteration n. Essentially, we reduce the dimension of the residual error and remaining basis functions successively. During

---

1. This is true until the residual error becomes zero.

each iteration, since all remaining unchosen vectors are orthogonal to the chosen set, D_{opt}, the optimal basis function is simply the one with the largest projection of the residual error. The complex process of iteratively solving a complete linear equation system in Equation 8 is avoided.

This approximation algorithm can achieve the PR property after m steps, since only independent basis functions are chosen. Also, the residual error decreases faster than the non-PR approximation method described above with some extra computational overhead.

3.2.2 Constant Rate vs. Constant Quality

The above successive approximation algorithm successively increases the number of coefficients and reduces the residual error. As discussed, the residual error always decreases to zero after m steps for the PR approximation but not for the non-PR approximation. In practice, the number of coefficients used is determined by the available output transmission capacity of the encoder, the acceptable reconstructed image quality, and the affordable processing power of the hardwares. Rate control can be easily achieved by limiting the number of generated coefficients. Quality control can be performed by measuring the final residual energy. Lastly, the computational complexity depends on the number of iterations performed. These controls are further complicated by quantization of the transform coefficients, which will be discussed in the following.

3.2.3 Quantization

Transform coefficients are usually further quantized to increase the compression ratio. Small coefficients may be truncated to zero after quantization. Thus, after quantization, the proportionality between the recovered image quality and the number of iterations may become invalid. The reason is twofold. First, small coefficients obtained in later iterations are truncated to zero. They will not increase the recovered image quality level. Second, existing coefficients may be changed when new coefficients are added (particularly in the PR approximation technique). These changes may cause the quantized approximation more distant from the perfect representation and thus increase the prediction error. Figure 8 shows the peak signal-to-noise ratio (PSNR) of a simple image segment during each iteration of successive approximation. The PSNR after quantization begins to drop after four iterations. One way to avoid this problem is to integrate the quantization into the optimization process. Namely, change Equation 8 to the following

$$Project(r_0, D_{opt}(\mathrm{n})) = \max_i \left(Quantz \left(Project \left(r_0, D_i(n) \right) \right) \right) \qquad \text{(EQ 14)}$$

In other words, we choose the basis function with the largest projection *after* quantization. This definitely increases the computational complexity, but the recovered image quality, as shown in Figure 8, becomes non-decreasing and generally higher than that obtained from the original approach. Another way to avoid the quality decline due to over-iteration is to end iteration when quality after quantization starts to drop or reach a preset quality goal.

3.3 Adaptive Transform Bases

Intuitively, the spatial statistics of an AS image segment varies with its irregular shape. Thus, it requires different optimal transform bases. For example, image pixels of a single line may prefer a 1D DCT basis while a square image block may prefer a 2D DCT basis. In this section, we describe the approach which uses adaptive transform bases based on the shape information of the

input image segment. As shown in Figure 9, the shape information is also available at the receiver, where the correct transform basis can be used to reconstruct the original image signal.

3.3.1 Existing Orthogonal Transform Bases

Finding the transform coefficients can be greatly simplified if the basis functions form an orthogonal set, in which case the coefficients can be obtained by simple projection. (If the basis functions are not orthogonal, then we need to solve a complete linear system.) Also, orthogonal basis functions usually imply good energy decoupling in the transform spectrum and thus improve the compression performance.

One way to construct an orthogonal transform basis is to reshape the arbitrarily-shaped image segment into a 1D array and apply the 1D DCT basis. DCT is known to be close to the optimal Karhunen-Loeve Transform (KLT) if the image contents have high spatial correlation. However, except the single-line images, arbitrarily-shaped image segments usually do not have exact 1D spatial correlations. Furthermore, the dimension of the 1D DCT basis changes with the image segment size. This will also make the codec design complex.

Another way to construct orthogonal basis functions in the subspace S_B is to use the Gram-Schmidt algorithm, as proposed in [5]. The Gram-Schmidt algorithm can extract an orthogonal subset of functions out of a larger set of arbitrary functions. One possible initial set of functions for the Gram-Schmidt algorithm is the traditional 2D DCT basis. Suppose the dimension of the full block is n and the dimension of subspace S_B is m ($m \leq n$). Let d_i's represent the original DCT basis function and $\hat{d}_i$'s represent their projected version in the subspace S_B. It can be shown that $\dim(\text{span}\{d_i\})=n$[1], $\dim(\text{span}\{\hat{d}_i\})=m$, and $\{\hat{d}_i\}$ are mutually dependent if $m<n$.

Actually, in the Gram-Schmidt algorithm, we still have a great deal of flexibility in choosing different orthogonal subset from a larger set of functions. In later simulations, we start from the DCT basis functions with the smallest zonal order. The final choice of orthogonal basis depends on the input image shape.

3.3.2 A New Orthogonal Transform — KLT-Like Transform

The KLT can be shown to be the best transform algorithm for the rectangular image segments if the spatial statistics of the input images are known. The DCT can be derived from the KLT if the image assumes a first-order Markovian model with high spatial correlation [4]. We propose a new transform basis using the above implication. We propose a new transform basis using this implication. Using the same assumption of a first-order Markovian model, we can find the variance-covariance matrix for an arbitrarily-shaped image segment. For example, if the image segment has m pixels, then we can rearrange the image segment, P(x,y), to a 1D array of m elements, and define a $m \times m$ variance-covariance matrix, C, with

$$C_{ij} = (\lambda_1)^{|k-l|} \cdot (\lambda_2)^{|p-q|} \qquad \text{(EQ 15)}$$

where λ_1 and λ_2 are the correlation coefficients in x and y direction respectively, P(k,p) is the i-th element in the 1D array, and P(l,q) is the j-th element in the 1D array. Figure 10 shows an example

1. $\dim(\text{span}\{d_i\})$ stands for the dimension of the vector space spanned by the vector set $\{d_i\}$.

of a 4-pixel segment in a 4×4 image block. For simplicity, we assume that λ_1 equals λ_2 in later simulations.

Using a technique similar to that for deriving DCT from KLT, we can set the correlation coefficients λ_1 (λ_2) to a value close to unity (e.g. 0.9) and find the eigenvectors of the above variance-covariance matrix, C. We can prove that these eigenvectors form an orthogonal basis in the subspace, S_B, so long as λ_1 and λ_2 are less than 1 [18]. Hopefully, these KLT-like transform bases can encode AS image segments as well as the DCT basis for the traditional rectangular image blocks. We will show the compression performance of this technique in the next section.

4. Performance Comparison

In this section, we use the irregular shaped image segment shown in Figure 3 (Miss USA) as the test case to simulate the performance of various transform coding schemes described in this paper. Only the partially defined boundary blocks (8 pixels × 8 pixels each) are used. As discussed earlier, it's difficult to have a quantitative measure of energy compactness of a transform spectrum. Instead, here we try to evaluate the rate-distortion performance of each transform scheme. The distortion is measured by the peak signal-to-noise ratio (PSNR) of the recovered image. The rate is represented by the compression ratio, i.e. the number of pixels inside the image segment divided by the number of non-zero transform coefficients after quantization. The results are shown in Figure 11. We use the uniform quantizers in the simulations.

There are basically three different groups of coding schemes in Figure 11. Algorithms in the first group use adaptive transform bases (section 3.3). They include 1D DCT, the proposed KLT-like transform basis, and DCT-based orthogonal transform bases proposed by Gilge *et al.* These algorithms change the transform basis when the image segment shape is changed. The transform bases are orthogonal and complete in the shape-projected subspace S_B. Therefore, the perfect reconstruction property is assured if transform coefficients are not quantized. At the decoders, the adaptive transform basis can be recalculated in the real time or pre-calculated and stored in the memory in advance. However, for the latter case, the required memory storage could be quite large due to the wide variety of possible shapes.

The second group of algorithms are modified versions of the successive approximation proposed by Kaup and Aach [6]. As discussed in section 3.2, the iteration process can be based on the quality or rate constraints. For example, a quality-based scheme may iterate until the PSNR reaches 50 dB; a rate-based scheme may iterate until the number of transform coefficients exceeds 25% of the number of the original pixels. On average, for the same performance level, the quality-based schemes need fewer iterations than the rate-based schemes. The reason is that the quality-based schemes can adapt to the local activity of individual image blocks and spend more computations on busy image blocks than on flat ones. The overhead of this successive approximation algorithm is only in the encoders. Existing decoders can be used to reconstruct the image segment without any modifications. Note that this group of algorithms can achieve perfect reconstruction (PR) at some cost of extra computations, as discussed in Section 3.2. The PR iterative scheme usually has a slightly higher quality than the non-PR iterative schemes at the same compression rate.

The third group of coding algorithms directly extend the image segments into full image blocks and apply the traditional 2D DCT algorithm. Two results are shown in Figure 10 — zero-stuffing and mirror-image extension proposed in Section 2.1. After augmentation, the image

segments are treated as the regular rectangular image blocks. No overheads are introduced while perfect reconstruction is assured.

From the R/D curves shown in Figure 10, we can see that the adaptive-basis schemes (the 1st group) and the iterative schemes (the 2nd group) outperform the most straightforward scheme (i.e. zero-stuffing) by a quality difference of 5-10 dB. The only exception is the 1D DCT, which suffers a lower performance (about 3-4 dB difference) at high compression ratios compared to other complicated schemes. This is a reasonable result since an arbitrarily-shaped 2D image segment usually does not have the spatial correlations similar to those found in 1D image sequence.

In order to avoid severe computational overhead, we use the non-PR iterative scheme in our simulations. However, a large number of iteration (20 iterations in average) is still required for the iterative method to achieve the performance shown in Figure 11. During each iteration, the residual vector needs to be projected to 64 possible basis vectors. The computation overhead is still quite significant.

A satisfactory performance is observed for the proposed mirror-image extension method. It can achieve a 3-4 dB compression gain over the zero-stuffing method without any significant overheads.

Figure 12 shows the enlarged image (by pixel replication) of the right shoulder area of the Miss USA test image shown in Figure 3. Its size is 72 pixels $\times$ 48 pixels and the block size is 8 pixels $\times$ 8 pixels. The original image and the reconstructed images by the representative schemes of the above three different groups of transform coding methods are displayed. Note the internal fully-defined image blocks are all the same for these three reconstructed images. They all use the traditional DCT transform. The difference occurs in the partially-defined boundary image blocks only.

From the displayed images, we can see that the advanced coding scheme, i.e. the DCT-based adaptive orthogonal transform, has the highest subjective quality, as expected from its superior objective quality (PSNR). Both the mirror-image extension method and the straightforward zero-stuffing technique suffers noticeable texture distortion near the boundary. However, the mirror-image extension seems to keep more accurate values for abrupt transitions on the boundary. This more or less corresponds to its moderate objective quality gain (3-4 dB) over the zero-stuffing technique described earlier.

Table 1 lists some major characteristics and compression performance for these coding algorithms. This comparison should be useful for system-level designs. If the processing resources are abundant, fancy algorithms like adaptive or iterative methods can be used to improve the reconstructed image quality. Otherwise, we can use simple mirror-image extension technique to achieve a fairly good reconstructed image quality. In addition, both adaptive and iterative algorithms require revision of the current codec hardwares, but the mirror-image extension technique is compatible with existing hardwares.

5. Conclusions and Future Work

Arbitrarily-shaped (AS) image segments will become more and more popular in the future advanced video applications. In this paper, we investigate efficient transform coding schemes for

AS image segments. We formulate the problem in two different domains — the straightforward full block rectangular domain and the shape-projected subdomain. In the former, we still use the traditional rectangular transform coding method but try to find optimal pixel values outside the segment boundary in order to make the transform spectrum as compact as possible. We propose a simple but efficient mirror-image extension technique to augment the partially-defined boundary image blocks into fully-defined rectangular blocks. In the latter, we project the image segment and all basis functions into the subspace spanned over the covered region only. The problem dimension is reduced and optimization is pursued in the shape-projected subspace. Existing coding algorithms such as the adaptive orthogonal technique proposed by Gilge and iterative method proposed by Kaup and Aach can be easily interpreted by using this flexible formulation. We also propose a new adaptive transform basis based on the implication of the optimal KLT transform. The biggest distinction between the above two approaches is that the latter changes the transform basis or the coefficient calculation process adaptively based on the shape information while the former uses a fixed transform algorithm.

Another focus of this paper is to analyze and compare the compression performance of different coding methods. In particular, we investigate the tradeoff between computational complexity, codec complexity, and recovered image quality for different coding methods. Using the image segment shown in figure 2 as a test case, we found that fancy algorithms like iterative algorithms or adaptive algorithms have a quality gain of about 5-10 dB (compared to the zero-stuffing technique) at some cost of extra computations or memory. Our proposed mirror-image extension method achieves a 3-4 dB gain compared to the zero-stuffing technique without any significant overheads. These analyses of performance and tradeoffs are useful for system-level designs to choose appropriate coding schemes.

As mentioned earlier, the shape information is necessary for AS image object manipulation. It needs to be encoded and transmitted separately from the image contents. We design efficient algorithms for shape coding and anti-aliasing in [14]. Besides, the compression schemes for the shape and image pixels will have strong impacts on the efficiency and flexibility of later manipulations of the AS image segments [8]. Designing compression algorithms for easy manipulation and designing efficient manipulation algorithms for given compression formats are two highly interrelated issues. Future extensions of this work include compression/manipulation algorithm co-design, and three-dimensional video objects.

6. References

1. Sandbank, C.P., *Digital Television*, John Wiley & Sons, 1990.

2. Musmann, H.G., M. Hotter, and J. Ostermann, “Object-Oriented Analysis-Synthesis Coding of Images,” Signal Processing: Image Communication, 1, 1989, pp.117-138.

3. Hotter, M., “Object-Oriented Analysis-Synthesis Coding Based on Moving Two-Dimensional Objects,” Signal Processing: Image Communication 2, 1990, pp.409-428.

4. Jain, A., *Fundamentals of Digital Image Processing*, Prentice-Hall Inc., 1989.

5. Gilge, M., T. Engelhardt, and R. Mehlan, “Coding of Arbitrarily Shaped Image Segments Based on A Generalized Orthogonal Transform,” Signal Processing: Image Communication 1, 1989, pp. 153-180.

6. Kaup, A. and T. Aach, “A New Approach Towards Description of Arbitrarily Shaped Image Segments,” IEEE International Workshop on Intelligent Signal Processing and Communication Systems, Taipei, Taiwan, March, 1992.

7. Chen, W.-L., S.-F. Chang, P. Haskell, and D.G. Messerschmitt, “Structured Video Model for Interactive Multimedia Video Services,” prepared for submission.

8. Chang, S.-F. and D.G. Messerschmitt, “A New Approach to Decoding and Compositing Motion-Compensated DCT-Based Images,” IEEE International Conf. on Acoustics, Speech, and Signal Processing, Apr. 1993, pp.421-424.

9. T. Porter and T. Duff, “Compositing Digital Images,” Computer Graphics, Vol. 18, pp. 253-259, 1984.

10. Clarke, R.J., “Transform Coding of Images,” Academic Press, 1985.

11. CCITT Recommendation H.261, “Video Codec for Audiovisual Services at px64 kbits/s”

12. Standard Draft, JPEG-9-R7, Feb. 1991

13. Standard Draft, MPEG Video Committee Draft, MPEG 90/ 176 Rev. 2, Dec. 1990.

14. Takahashi, M., S.-F. Chang, and D.G. Messerschmitt, “Joint Shape Representation and Anti-Aliasing for Arbitrarily-Shaped Image Objects,” IEEE Intern. Workshop on Intelligent Signal Processing and Communication Systems, Oct. 1993, Sendai, Japan.

15. Soltanian-Zadeh, H. and A.E. Yagle, “Fast Algorithms for Extrapolation of Discrete Band-Limited Signals,” IEEE International Conf. on Acoustics, Speech, and Signal Processing, Apr. 1993, pp. 591-594.

16. Foley, J.D., A. Dam, S. Feiner, and J. Hughes, “Computer Graphics: Principles and Practice,” 2nd ed., Addison-Wesley, 1990.

17. Karlsson, G. and M. Vetterli, “Extension of Finite Length Signals for Sub-band Coding,” Signal Processing, Vol. 17, No.2, pp.161-168, June, 1989.

18. Golub, G.H., and Charles F. Van Loan, *Matrix Computations*, 2nd edition, the John Hopkins University Press, 1989.

Table 1. Characteristics of several transform coding algorithms for arbitrarily-shaped image segments.

	Transform Bases	Iterative Computations	Perfect Reconstruction (PR)	Compression Gain[a]
Orthogonal_DCT	adaptive, orthogonal[b]		Yes	6-12 dB
KLT-like	adaptive, orthogonal		Yes	5-10 dB
1D DCT	adaptive, orthogonal		Yes	2.7-7 dB
Kaup & Aach's iterative method	static	Yes[c]	Possible	5-10 dB
Mirror-image extension	static, orthogonal		Yes	2.7-4 dB
Zero-stuffing	static, orthogonal		Yes	—

a. This is the gain of the average PSNR (in comparison to the zero-stuffing method) at the fixed compression rate. Note that the average PSNR is computed over the boundary image blocks only.

b. orthogonal with respect to the shape-projected subspace

c. 20 iterations for the R/D curve shown in Figure 11

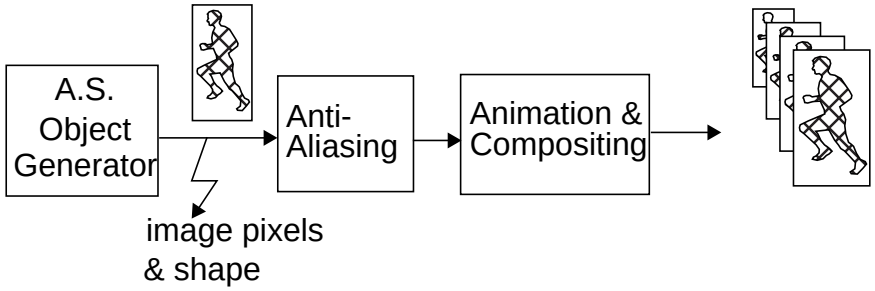


FIGURE 1. An experiment system for manipulating/compositing arbitrarily-shaped (AS) video objects.

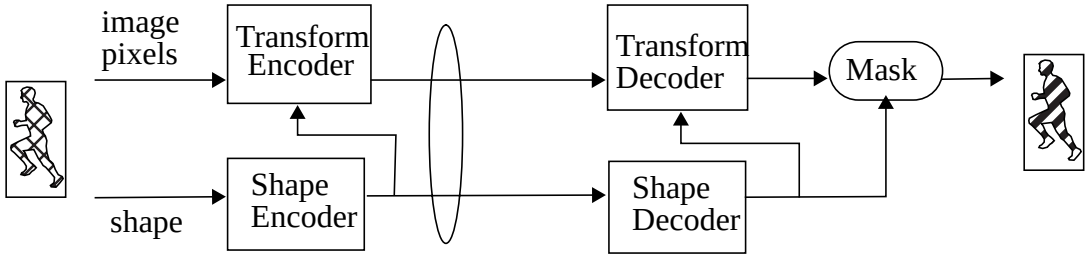


FIGURE 2. A complete representation of the AS image segments include both the image pixels and the shape information, which are encoded and transmitted separately. The shape information can be used to assist in improving the transform coding of the image pixels, which will be described later.

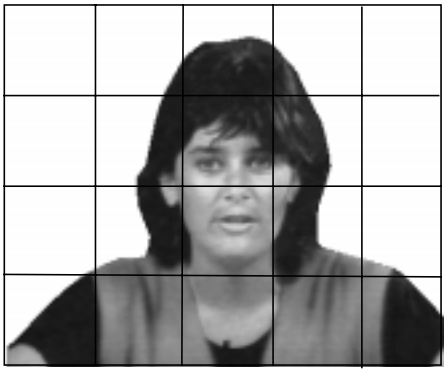


FIGURE 3. An example AS image segment and the grid lines which separate the image into small blocks. Boundary blocks are not completely defined. The block structure is for demonstrative purpose and is not of accurate scale.

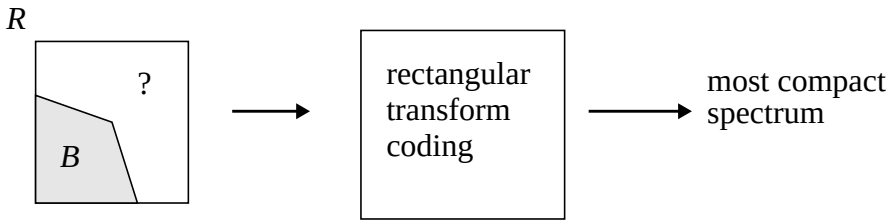


FIGURE 4. Find the optimal pixel values outside the boundary of image segment P, so that the transform spectrum has the most compact energy spectrum.

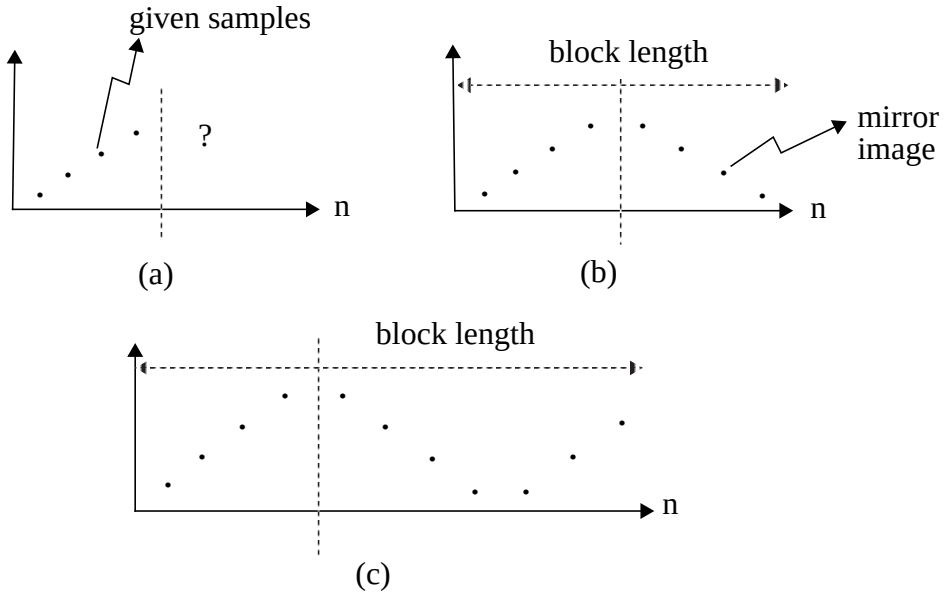


FIGURE 5. Fill the outside redundant region with the mirror image of the internal pixels. (a) original segment. (b)the segment size equals one half of the block size. (c)apply the mirror image recursively when the segment size is not one half of the block size.

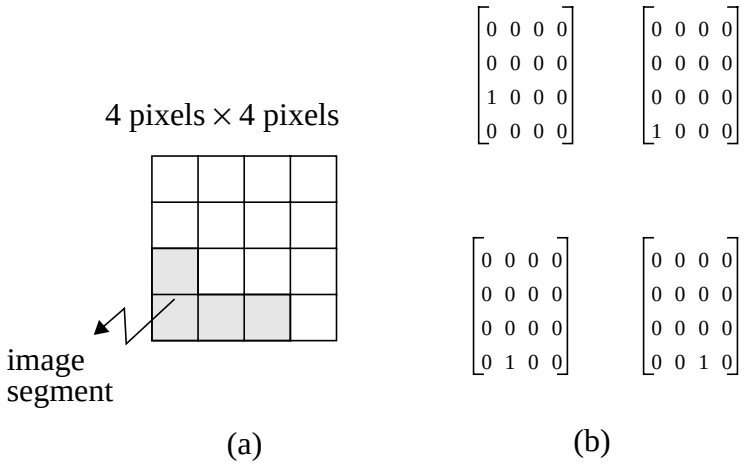


FIGURE 6. (a)A partially-filled image block in a 4×4 area. A canonical basis of the subspace is shown in (b).

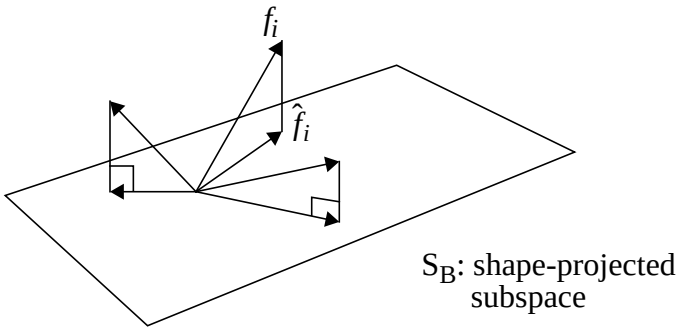


FIGURE 7. Project the image segment and all representation basis functions into the subspace spanned over the covered area only.

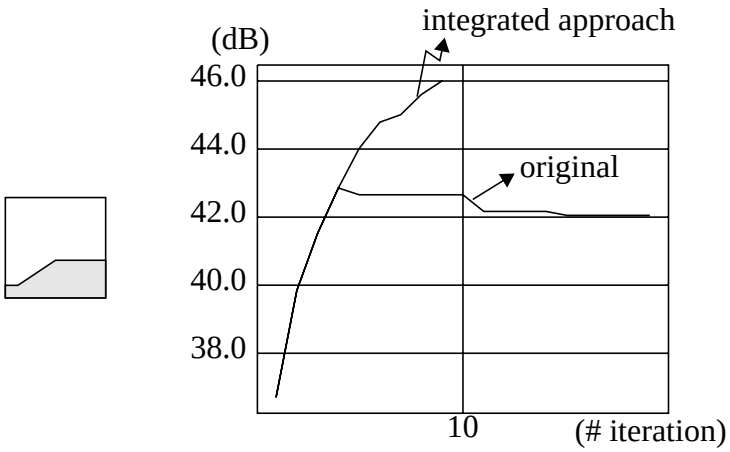


FIGURE 8. A simple image segment and its PSNR in each iteration of the PR successive approximation coding algorithm. The *original* method finds the minimal residual error *before* quantization, while the *integrated* method finds the minimal residual errors *after* quantization. We use uniform quantization here.

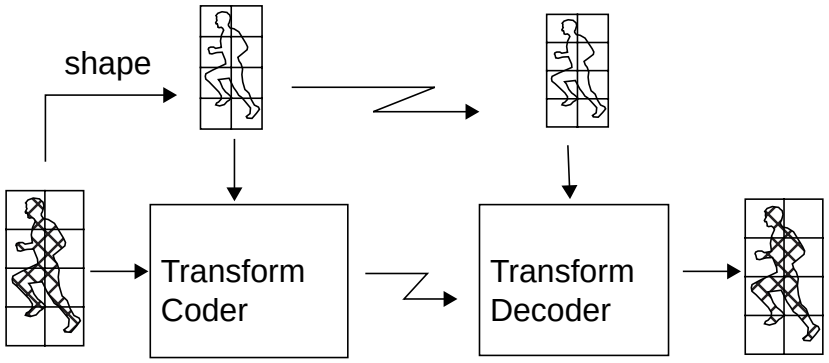
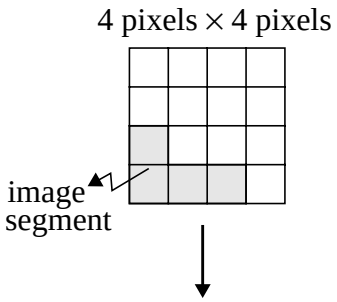


FIGURE 9. Use the shape information to assist in choosing the optimal transform basis. The shape information is also available at the receiver and thus the correct transform basis can be used to reconstruct the original image.



$$\begin{bmatrix} 1 & \lambda_2 & \lambda_1\lambda_2 & \lambda_1^2\lambda_2 \\ \lambda_2 & 1 & \lambda_1 & \lambda_1^2 \\ \lambda_1\lambda_2 & \lambda_1 & 1 & \lambda_1 \\ \lambda_1^2\lambda_2 & \lambda_1^2 & \lambda_1 & 1 \end{bmatrix}$$

1D array: {P(3,1), P(4,1), P(4,2), P(4,3)}

(a) (b)

FIGURE 10. (a)Reshape the image segment into a 1D array and (b)construct its variance-covariance matrix based on the 1st-order Markovian model.

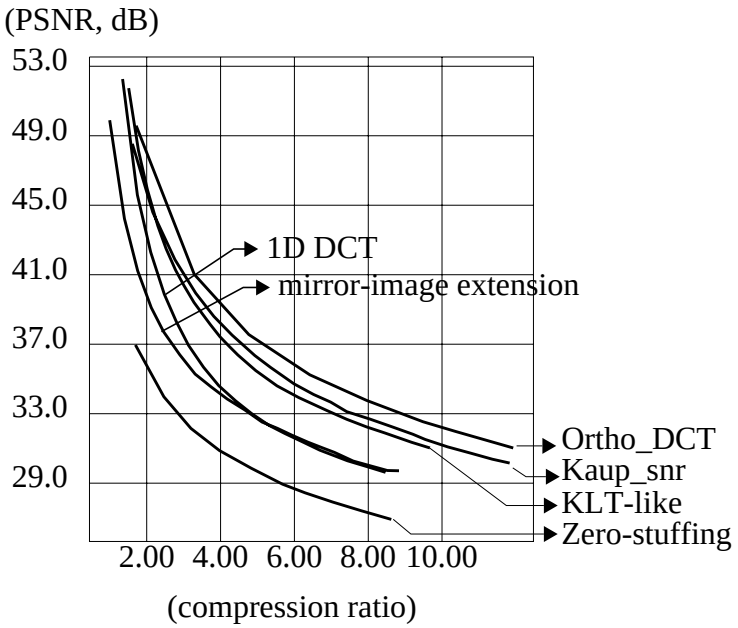


FIGURE 11. Rate/Distortion curves for various transform coding schemes for the image segment shown in Figure 3 by using uniform quantizers. (Kaup_snr represents the Kaup & Aach's successive algorithm which iterates until the PSNR before quantization exceeds 50 dB.)

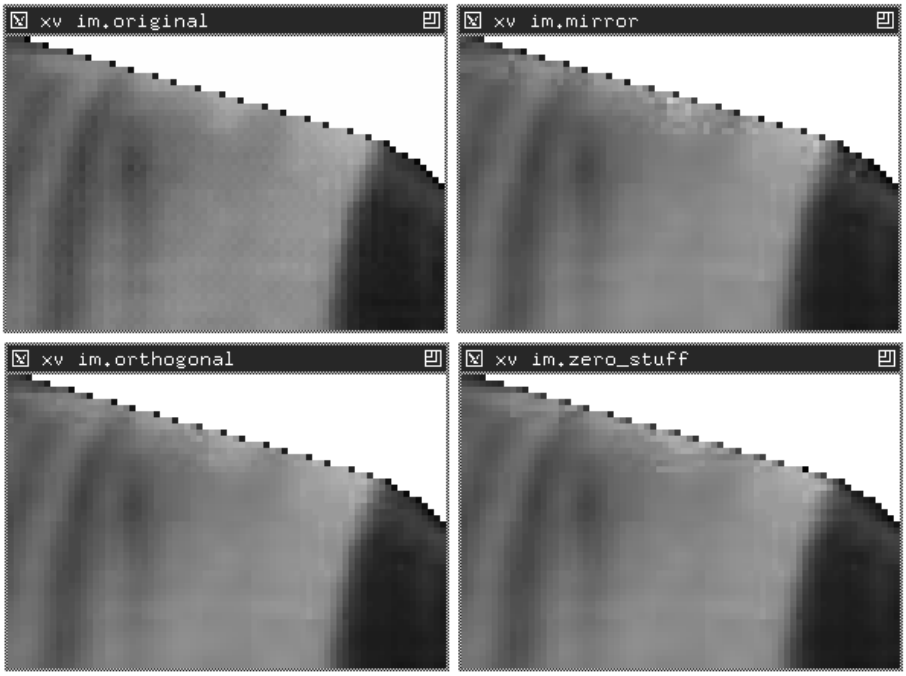


FIGURE 12. An enlarged portion (by pixel replication) of the test image (Miss USA). (a)original (b)DCT-based orthogonal transform (c)mirror-image extension (d)zero-stuffing.

(a)	(c)
(b)	(d)

List of Figure and Table Captions