Transform Coding of Arbitrarily-Shaped Image Segments

Shih-Fu Chang and David G. Messerschmitt

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

Abstract

Envisioned advanced multimedia video services include both rectangular and arbitrarily-shaped image segments. Image segments of the TV weather reporter produced by the chromo-key technique and image segments produced by video segmentation or image editing are typical examples. In this paper, we investigate efficient transform coding techniques of arbitrarily-shaped image segments. We formulate the optimal representation problem in two different domains — the full rectangular domain and the shape-projected domain. In the former, we still use the traditional rectangular transform coding method (e.g. DCT) but try to find optimal pixel values outside the segment boundary in order to make the transform spectrum as compact as possible. A simple but efficient mirror-image extension technique is proposed. In the shape-projected domain, we project the image segment and all basis functions into the subspace spanned over the image region only. Existing coding algorithms, such as orthogonal transform by Gilge [1] and iterative coding by Kaup and Aach [2], can be intuitively interpreted. To demonstrate the flexibility of the proposed formulation, we also derive a new KLT-like algorithm in the shape-projected domain. We analyze tradeoff 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 proposed 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 formulation, new transform coding techniques, and numerical tradeoff analyses. Currently, we are implementing a software program for AS image object editing and manipulation.

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) [3,4]. In multimedia editing systems, users can create arbitrarily-shaped 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 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 contents, perform anti-aliasing along the boundary to make them smooth, and then we can manipulate the AS video objects as desired.

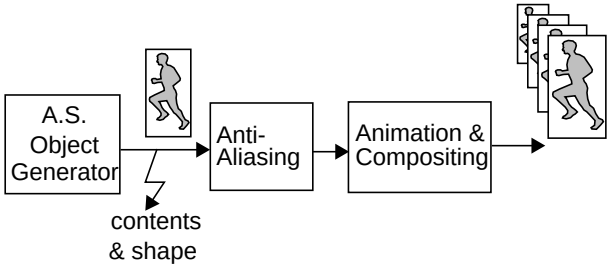

FIGURE 1. An experiment system for manipulating/ compositing arbitrarily-shaped video objects.

A complete representation of AS video objects includes two parts — *shape* and *internal contents*. 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. In this paper, we focus on designing efficient representation of the image contents to achieve good compression and image quality. In particular, we look at block-wise transform coding of the image contents, such as the widely used Discrete Cosine Transform (DCT) [5,6,7,8]. One immediate advantage of using the transform code is that existing codec hardwares can be used to process AS video signals, as well as traditional rectangular video signals.

In block-wise transform coding algorithms, images are separated into small blocks with fixed size, say N pixels by N pixels. Figure 2 shows an example AS video object (miss USA) and illustrates the concept of block structure. In internal blocks, all pixel values are fully defined. The traditional DCT algorithm can be used to encode these blocks efficiently. However, for the boundary blocks, only part of the pixel values are defined. One straightforward approach is to fill zero values outside the boundary and treat the image block as traditional image blocks. But an obvious drawback of this approach is significant increase of the high-

The authors are with the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA 94720. E-mail address: *sfchang, messer@eecs.Berkeley.EDU*

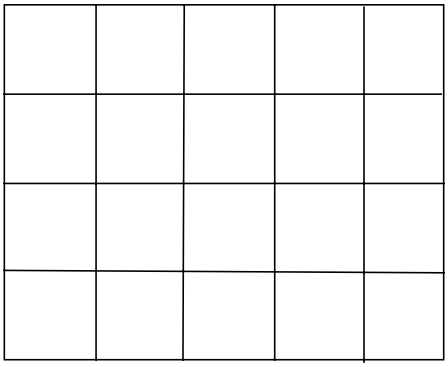

FIGURE 2. An example AS image segment and the grid lines which separate the image into small blocks. Boundary blocks have part of pixel values defined only. The block structure is for demonstrative purpose and is not of accurate scale.

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 outside the boundary in the boundary blocks and then take the traditional full-block DCT. The latter changes the transform basis functions adaptively, based on the shape of the input block. The iterative approximation method proposed by Kaup and Aach [2] and the adaptive orthogonal transform proposed by Gilge *et al*. [1] both belong to this class of techniques. We will propose the shape-projected domain as an efficient problem formulation, based on which we can easily interpret adaptive transform bases. We will also derive a new KLT-like transform bases to demonstrate the flexibility of our proposed formulation.

Afterwards, we will compare the performance of different transform coding techniques and illustrate the tradeoff relation between the compression performance, computational complexity, and codec complexity. The contributions of this paper lie in an efficient problem formulation, new transform coding techniques, and numerical tradeoff analyses.

2. Approach I: 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 part of the pixel values defined only. Let P(x,y) represent the pixel values within a N pixel × N pixel block area, called *R*. Let *B* represent the occupied region within the block, as shown in figure 3. An irregular shaped image segment has P(x,y) defined within region *B* only. 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.

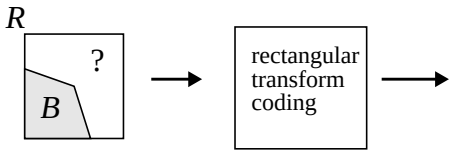

FIGURE 3. Find the optimal pixel values outside the boundary of image segment P, so that the transform spectrum has the most compact energy spectrum.

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,

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

where f_i's are basis functions defined on the full-block area, *R*. Namely, P(x,y) is transformed to coefficients a_i, which can be used to reconstruct the original signal completely or partially. The objective is to use as least coefficients as possible to obtain an accurate reconstruction, $\hat{P}(x, y)$. The resulting error term can be defined as

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

Note the summation is executed over the occupied region, *B*, only, because error terms outside the boundary will be discarded when we apply the shape information at the receiver.

If we fix the choice of basis functions, e.g. use 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 highest energy compaction. The concept is illustrated in figure 2. However, it is difficult to quantitatively formulate the abstract property — "*energy compaction*". An example discussed in [2] is to use 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. But the problem with this definition is that the final choice usually will end up with few large spectrum components which may cause overflow problems, though the spectrum "entropy" is low. Furthermore, optimization with respect to the above objective function is difficult.

But, the approach to 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 by stuffing zero's outside the image boundary. But it is well known this method may introduce sharp edges on the boundary and high-frequency components in the transform spectrum. A

more promising method is to extend the image segment with its "*mirror image*" outside the boundary. Figure 4 shows a simple example in one dimension. In general, the support of the original image sequence is not exactly one half of the block size, we may need to duplicate the image sequence several times and truncate it at the block boundary. For a 2D image segment, we can apply this 1D mirror image extension technique in one direction first, and then once again in another direction.

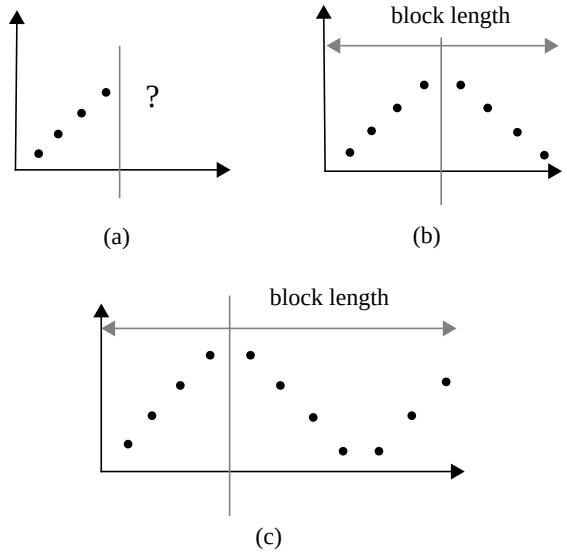

FIGURE 4. Fill the outside redundant region with the mirror image of the internal contents. (a) original segment. (b)the segment size equals one half of the block size. (c)apply the mirror image recursively when the segment size is not one half of the block size.

This mirror-image extension technique is simple but efficient. Its compression performance will be described in section 4. Actually, there are other techniques which can be applied to extrapolate the image segment and fill up the whole image block [11,12]. We are currently investigating their effectiveness.

3. Approach II: Shape Adaptive Transform

As described in equation 2, the only concerned errors are reconstruction errors within the image boundary, i.e. errors within the covered region *B*. An equivalent but maybe more efficient approach to finding optimal representation of AS image segments is to perform the optimization in the subspace defined over region *B* only, denoted by S_B. Basically, we project the AS image block and the transform bases 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. But since the subspace varies with the image shape, the optimal transform bases for different image shapes may also vary. This is the reason why we call this approach *shape-adaptive*.

3.1 Shape-Projected Subdomain

Instead of filling data outside the image boundary and applying the full-block rectangular transform, we can focus on the defined image contents only, i.e. P(x,y) values within region *B*. Mathematically, let's define S_R as the linear space spanned over the whole square block *R*, S_B as the subspace spanned over the irregular region *B* only. For example, in figure 5, space S_R has a dimension equal to 16, while the dimension of subspace S_B equals to 4 only. One possible bases for subspace S_B is shown in figure 5(b). Each basis matrix has a single non-zero element only.

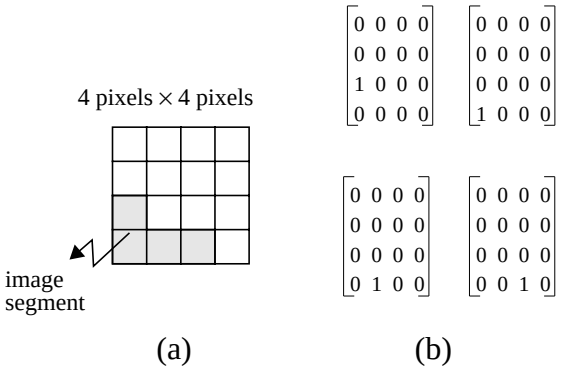

FIGURE 5. (a)An irregular-shape image segment in a 4×4 block area. A canonical bases of the subspace is shown in (b).

Every arbitrarily-shaped image segment, P(x,y), can be considered as a vector in S_B. To represent this vector completely, we can find a 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 whole problem domain now is confined in the subspace S_B only. We don't have to worry about the redundant data outside the image boundary, i.e. vector component outside subspace S_B. If we still want to use traditional block-based transform bases, say f_i (e.g. DCT bases), we can project these basis functions into subspace S_B,

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

and describe vector P(x,y) as a linear combination of $\hat{f}_i$'s. Actually, the above projection is very simple. It just removes the components of f_i outside subspace S_B.[1] Figure 6 illustrates a simple example when the dimension of S_B equals 2

An important issue remains now is how to find 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 very flexible platform to derive new transform bases and evaluate their performance. We will describe some existing approaches and our new proposal in the following subsections.

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

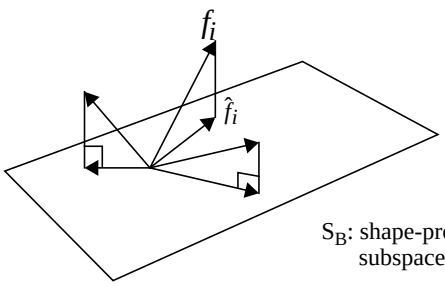

FIGURE 6. Project the image segment and all representation bases into the subspace spanned over the image shape region only.

3.2 Successive Approximation with Fixed Bases in the Full-Block Domain

Using the existing full-block 2D DCT bases to represent the arbitrarily-shaped image segments is still very attractive since existing decoders for rectangular images can be used without any modifications. But as described earlier, the shape-projected DCT bases, $\{\hat{d}_i\}$, are generally not orthogonal and mutually dependent. There are more than one solutions for equation 1 if we use $\{\hat{d}_i\}$ as the representation bases. Instead of finding a fully accurate representation, Kaup and Aach [2] proposed a successive approximation method to calculate the most significant coefficients only. In this section, we first briefly review Kaup and Aach's approach based on our shape-projected subdomain formulation. Then, we apply this 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 errors, we call it a *perfect-reconstruction* (PR) representation. Otherwise, it's called a *non-perfect-reconstruction* (non-PR) representation. If the rank of the representation bases is less than the dimension of the image segment vector, then the PR property cannot be guaranteed. As mentioned earlier, the shape-projected DCT bases $\{\hat{d}_i\}$ form a dependent but complete set of vectors in the shape-projected subspace. We should be able to choose m independent bases out of projected DCT bases to achieve PR, where m is the rank of the shape-projected subspace. But the issue is which bases can produce the best energy compaction.

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

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

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

where r is the residual error during each iteration. Note the same basis function could be chosen repetitively since $\{\hat{d}_i\}$ are not orthogonal. A more important point is that the number of transform coefficients may exceed m (i.e. the image segment size) without achieving the PR property.

One way to guarantee PR in the above iterative algorithm is to accumulate the chosen bases in each iteration and project the image segment to the whole set of chosen bases, not only a single basis function. Namely, the following operations are performed during each iteration.

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

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

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

where $D_{opt}(\text{n})$ is the set of chosen basis functions up to the current iteration. During each iteration, we keep previously chosen set of bases from last iteration and add an additional basis to minimize the residual error (i.e. maximizing the projection). The rank of the representation bases is incremented by one in each iteration. Note, during each iteration, we need to project the image segment vector to every possible set of bases, each of which requires solving a complete linear equation system. This computation overhead is quite significant.

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

$$\hat{d}_i(\text{n}+1) = \hat{d}_i(\text{n}) - \text{Project}(\hat{d}_i(\text{n}), \hat{d}_{opt}(\text{n})) \ ,$$

$$\text{for all } \hat{d}_i \notin D_{opt} \qquad \text{(EQ 10)}$$

$$r(\text{n+1}) = r(\text{n}) - \text{Project}(r(\text{n}),\ \hat{d}_{\text{opt}}(\text{n})) \qquad \text{(EQ 11)}$$

where $\hat{d}_{\text{opt}}(\text{n})$ is the new basis added to the chosen set in iteration n. Essentially, we reduce the dimension of the residual error and remaining basis functions one by one successively. During each iteration, since all remaining unchosen bases are orthogonal to the chosen set, D_{opt}, the newly added optimal basis is simply the one with the largest projection of the residual vector. The complex process of iteratively solving a complete linear equation system in equation 8 is avoided.

This approximation algorithm can guarantee the PR property after m steps, since only independent bases are chosen. Also, the prediction error decreases faster than the above non-PR approximation method at some cost of computational overhead.

3.2.2 Constant Rate vs. Constant Quality

The above successive approximation algorithm increases the number of coefficients and thus reduces the residual error successively. As discussed, the residual error will always decrease 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 done by limiting the number of coefficients required. Quality control can be done by measuring the final residual energy. And computational complexity depends on the number of iterations performed. These controls will be 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 rate. Small coefficients may be truncated to zero after quantization. Thus, after quantization, the direct 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 and thus will not increase the recovered image quality level. Second, existing coefficients maybe changed when new coefficients are added (particularly for the PR approximation technique). These changes may cause the quantized approximation more distant from the perfect representation and thus increase the prediction error. Figure 7 shows the peak signal-to-noise ratio (PSNR) of a simple image segment when the successive approximation proceeds. The PSNR after quantization begins to drop after 4 iterations. One way to avoid this problem is to integrate the quantization into the optimization process, namely change equation 8 to the following

$$\text{Project}(r_0, D_{opt}(\text{n})) = \max_i (\text{Quantz}(\text{Project}(r_0, D_i(\text{n})))) \qquad \text{(EQ 12)}$$

In other words, we choose the bases with the largest projection *after* quantization. This definitely will increase the computational complexity, but the recovered image quality, as shown in figure 7, becomes non-decreasing and generally higher than that obtained from the original approach. Another simple way to avoid this quality decline due to over-iteration is to stop iteration when quality after quantization begins to drop or already reach a preset quality goal.

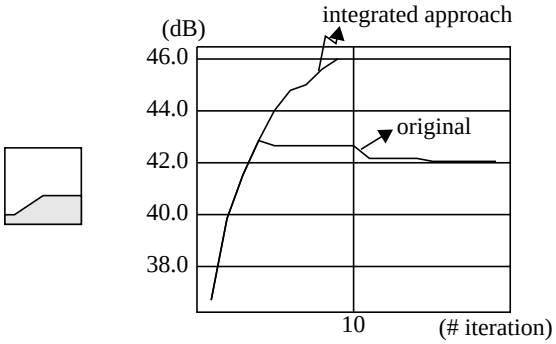

FIGURE 7. A simple image segment and its PSNR in each iteration of the PR successive approximation coding algorithm. The original method finds the minimal residual error *before* quantization, while the integrated method finds the minimal residual errors *after* quantization. We use uniform quantization here.

3.3 Adaptive Transform Bases

Intuitively, the spatial statistics of an AS image segment will vary with its irregular shape, thus will require different optimal transform bases. For example, a single image line may prefer a 1D DCT bases, while a square image block may prefer a 2D DCT bases. 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 8, the shape information is also available at the receiver and thus correct transform bases can be used to reconstruct the original image signal.

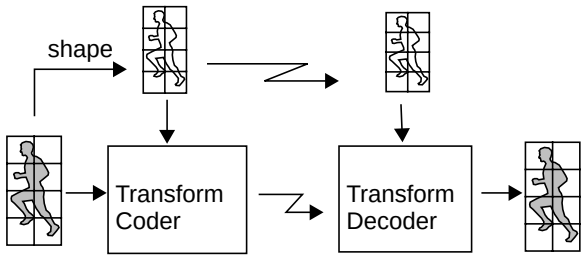

FIGURE 8. Use the shape information to assist in choosing the optimal transform bases. The shape information is also available at the receiver and thus correct transform bases can be used to reconstruct the original image.

3.3.1 Orthogonal Transform

The computations for finding the coefficients in the linear representation of equation 1 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 to each other, then we need to solve a complete linear equation system. Also, using orthogonal basis functions usually implies good energy decoupling in the transform spectrum.

One easy way to construct an orthogonal transform bases is to reshape the arbitrarily-shaped image segment into a 1D array and apply the 1D DCT bases. DCT is known to be close to the optimal Karhunen-Loeve Transform (KLT) if the image contents has high spatial correlation. But except the single-line image, most arbitrarily-shaped image segments usually do not have exact 1D spatial correlations. Furthermore, the dimension of the 1D DCT bases 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 Schmidt algorithm, as proposed in [1]. The Schmidt algorithm can extract an orthogonal subset of functions out of a larger set of arbitrary functions. One possible initial seed set of functions for the Schmidt algorithm is the traditional 2D DCT bases. Suppose the dimension of the full block is n and the dimension of subspace S_B is m (m<=n). Let d_i represent the original DCT bases, and $\hat{d}_i$'s represent their projected version in the subspace S_B. It is easy to show that rank($\{d_i\}$)=n, rank($\{\hat{d}_i\}$)=m, and $\{\hat{d}_i\}$ are mutually dependent if m<n. Actually, in the Schmidt algorithm, we still have a great flexibility in choosing different orthogonal subset from a larger set of func-

tions. In later simulations, we start from the DCT bases with the smallest zonal order. The final choice of orthogonal bases will depend on the input image shape.

3.3.2 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 [11]. We propose a new transform bases based on the above 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×m variance-covariance matrix, C, with

$$C_{ij} = (\lambda_1)^{|k-l|} \cdot (\lambda_2)^{|p-q|} \qquad \text{(EQ 13)}$$

where λ_1 and λ_2 are correlation coefficients in x and y direction, 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 9 shows an example of a 4-pixel segment in a 4×4 image block. For simplicity, we assume that λ_1 equals λ_2 in later simulations.

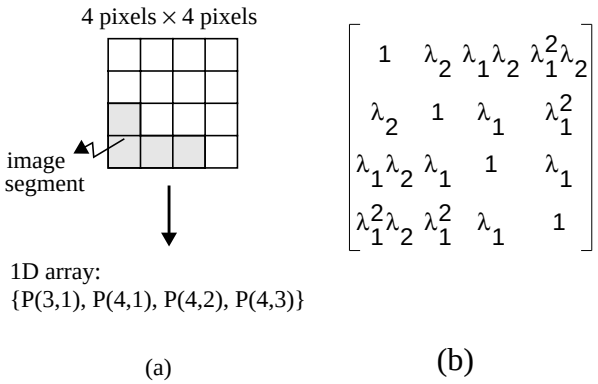

FIGURE 9. (a)Reshape the image segment into a 1D array and (b)construct its variance-covariance matrix based on the 1st-order Markovian model.

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 bases in the subspace, S_B, as long as λ_1 and λ_2 are less than 1. Hopefully, these KLT-like transform bases can encode AS image segments as well as the DCT bases do to 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 2 (miss USA) as a test case to simulate the performance of various transform coding schemes described in this paper. Only the boundary blocks (8 pixels × 8 pixels each) with irregular shape are used. As discussed in section 2, it's very difficult to have a quantitative measure of energy compactness of a transform spectrum. Here, we use a uniform quantizer with various quantization steps to approximate the distortion/rate curve 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 10.

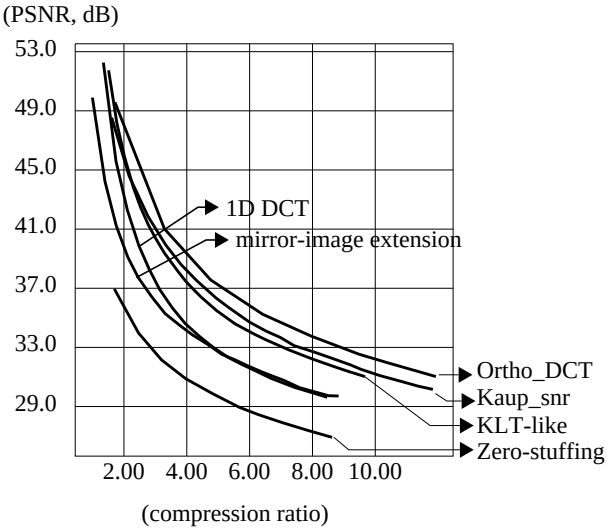

FIGURE 10. Rate/Distortion curves for various transform coding schemes for the image segment shown in figure 2 by using uniform quantizers. (Kaup_snr represents the Kaup & Aach's successive algorithm which iterates until the PSNR before quantization exceeds 50 dB.)

There are basically three different groups of coding schemes in figure 10. Algorithms in the first group use adaptive transform bases. They include 1D DCT (section 3.3.1), the proposed KLT-like transform bases (section 3.3.2), and DCT-based orthogonal transform bases proposed by Gilge *et al.* (section 3.3.1). These algorithms change the transform bases when the image segment shape changes. They are orthogonal and complete set in the shape-projected subspace S_B. Therefore, perfect reconstruction property is assured if without quantization. At the decoders, the adaptive transform bases can be recalculated in the real time, or pre-calculated and stored in the memory in advance. However, the required memory size could be large due to the large variety of possible shapes.

The second group of algorithms are modified versions of successive approximation proposed by Kaup and Aach [2]. As discussed in section 3.2.2, the iteration process can be based on the output quality or rate constraints. For example, the quality-based scheme may iterate until the PSNR reaches 50 dB. The rate-based scheme may iterate until the number of transform coefficients exceeds 25% of the segment size. In average, for the same performance level, the quality-based schemes need fewer iterations than the rate-based schemes. The reason is because that the

quality-based schemes can adapt to the local activity of each individual image block and spend more computations in busy image blocks than flat ones. The overhead of this successive approximation algorithm is in the encoders only. Existing decoders can be used to reconstruct the image segment without any modifications. Note, this group of algorithms can achieve perfect reconstruction (PR) at some cost of extra computations, as discussed in section 3.2.1. 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. After augmentation, the image segments are treated as the regular rectangular image blocks. No overheads are introduced and the perfect reconstruction is assured.

From the R/D curves shown in figure 10, we can see that adaptive-basis schemes (the 1st group) and iterative schemes (the 2nd group) outperform the most straightforward scheme (i.e. zero-stuffing) by a quality gap of 5-10 dB. The only exception is the 1D DCT, which suffers a lower performance (about 3-4 dB difference) at high compression rates compared to other complicated schemes. This is quite reasonable since an arbitrarily-shaped 2D image segment usually does not have 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. But a large number of iteration (20 iterations in average) is still required for the iterative method to achieve the performance shown in figure 10. However, 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 compared to the zero-stuffing method without any significant overheads.

Table 1 lists some major characteristics and compression performance of these coding algorithms. This comparison should be very 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 image quality. In addition, both adaptive and iterative algorithms need to change the codec hardwares, but the mirror-image extension technique is compatible with existing hardwares.

5.0 Conclusions and Future Works

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 extend the irregular image segments into a full block image. In the latter, we project the image segment and all basis functions into the subspace spanned over the image segment only. 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 transform bases based on the implication of optimal KLT transform.

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. Currently, we are also studying efficient techniques to represent the irregular shapes with a joint consideration of the anti-aliasing operation (as shown in figure 1). Our goal is to design a software to support creation, compression, and manipulation (like scaling, translation and rota-

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

	Transform Bases	iterative computations	Perfect Reconstruction (PR)	Compression Gain cp. to zero-stuffing
Orthogonal_DCT	adaptive, orthogonal[a]		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 (20 iterations for the R/D curve of fig. 10)	Possible	5-10 dB
Mirror-image extension	static, orthogonal		Yes	2.7-4 dB
Zero-stuffing	static, orthogonal		Yes	—

a. orthogonal with respect to the shape-projected subspace

tion) of AS image objects. The envisioned application is desktop multimedia editing.

6.0 References

[1] 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.

[2] 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.

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

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

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

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

[7] Clarke, R.J., “Transform Coding of Images,” Academic Press, 1985.

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

[9] Standard Draft, JPEG-9-R7, Feb. 1991

[10] Standard Draft, MPEG Video Committee Draft, MPEG 90/176 Rev. 2, Dec. 1990.

[11] Jain, A., “Fundamentals of Digital Image Processing,” Prentice-Hall Inc., 1989.

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

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