A Framework for Linear Transform Approximation Using Orthogonal Basis Projection

Yinpeng Chen
Arts, Media and Engineering Program, Arizona State University, Tempe, AZ, USA
Email: yinpeng.chen@asu.edu

Hari Sundaram
Arts, Media and Engineering Program, Arizona State University, Tempe, AZ, USA
Email: hari.sundaram@asu.edu

***Abstract*—This paper aims to develop a novel framework to systematically trade-off computational complexity with output distortion in linear multimedia transforms, in an optimal manner. The problem is important in real-time systems where the computational resources available are time-dependent. We solve the real-time adaptation problem by developing an approximate transform framework. There are three key contributions of this paper – (a) a fast basis projection approximation framework that allows us to store *signal independent partial transform* results to be used in real-time, (b) estimating the complexity distortion curve for the linear transform approximation using a given basis projection approximation set and searching for optimal transform approximation which satisfies the complexity constraint with minimum distortion and (c) determining optimal operating points on complexity distortion function and a meta-data embedding algorithm for images that allows for real-time adaptation. We have applied this approach on the FFT approximation for images with excellent results.**

***Index Terms*—linear transform approximation, basis projection, complexity distortion function, metadata encoding**

I. INTRODUCTION

This paper presents a novel framework for developing linear transform approximations that adapt to changing computational resources. The problem is important since in real-time multimedia systems, the computational resources available to content analysis algorithms are not fixed. A *generic* computationally scalable framework for content analysis would be very useful. The problem is made difficult by the observation that the relationship between computational resources and distortion depends on the specific content.

Based on "Basis Projection for Linear Transform Approximation in Real-Time Applications", by Yinpeng Chen and Hari Sundaram which appeared in the Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2006, Toulouse, France, May 2006. © 2006 IEEE.

A. Motivation

The desired approximation framework should be adaptive to the changing computational resources over time. For example, let us assume that we play a video on a PDA. Further assume that another program starts to run in the background. Because the background program takes partial computational resources and the available computational resources for the video player suddenly decrease, this causes delay. Hence, the PDA may not play the video in real time. In Figure 1, the solid curve shows the available computational resources for the video player and the straight line shows the computational complexity of the video decoding and visualization algorithm. We can see that during the time between t_1 and t_2, the available resources for video player are less than the computational complexity of the video decoding and visualization algorithm due to running the background program and the video player cannot operate in real time. Therefore, we seek an approximate transform that is able to gracefully adapt to the resources available, with small error. In Figure 1, the dashed curve represents the time varying computational resource usage by our desired transform.

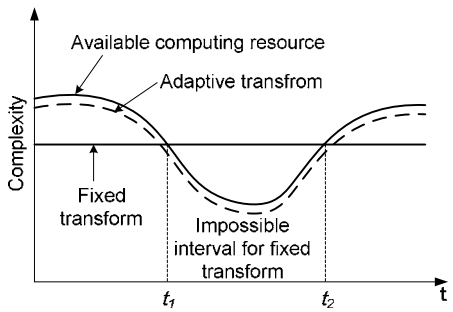

Figure 1. Computational complexity for fixed and adaptive transforms (example: video decoding algorithm that adapts to changing computational resources).

B. Related Work

There has been prior work on fast transform approximation. In [12,20,28], several fast FFT pruning techniques are proposed. They reduced the number of operations by only computing a subset of output coefficients. A FFT approximation technique suitable for on-chip spectral BIST signal generation and analysis is proposed in [9]. In [13,27], the authors propose several algorithms for pruning 1-D DCT. Fast, recursive DCT algorithm based on the sparse factorizations of the DCT matrix are proposed in [3,11,15]. Besides 1D algorithms, two-dimensional DCT algorithms have also been investigated in [2,8,10,18,26]. These fast DCT algorithms require fixed-point multiplications and need 32-bit data bus, which is costly in VLSI implementation and hand-held devices. To solve this problem, Liang [17] proposed fast multiplierless approximations of DCT with lifting scheme.

There has been prior work on adaptation in multimedia. The computational efficient transforms in video coding was proposed in [19,30]. There has been work on complexity-scalable coders by focusing on various aspects [1,4,14,21,23,24]. In the area of adaptive multimedia streaming, [16,32] focused on content adaptive transcoding. Schaar [25] proposed a generic rate-distortion-complexity model that can generate MPEG-21 Digital Item Adaptation (DIA) descriptions for image and video decoding algorithms. In more theoretical work [22] the authors look at properties of approximate transform formalisms and [29] looks at relationship between Kolmogorov complexity and distortion.

However several issues remain – (a) while there has been some success in complexity scalable decoders, there are no formal *generic* adaptation strategies to guide us for other content analysis applications, (b) given a specific transform (say DCT) approximation and distortion, there is no framework that enables us to systematically change the approximation in real-time to take advantage of additional computational resources to minimize distortion.

C. Our approach

There are three key ideas in this paper. First we show that a linear transform can be efficiently approximated with low computational complexity using a basis projection technique. Then we show that for a linear transform, there exists a complexity distortion curve, that is estimated using a basis projection set. Finally we show how optimal operating points from the C-D curve can be added as metadata to the images, with about 1.5% increase in the size of the image. We use the FFT as the linear transform, and apply the polynomial basis projection and Haar wavelet basis projection to approximate FFT for image. We show excellent results on the standard Lena image. This paper builds upon our current work [6].

This paper is organized as follows. In Section II, we define the optimal approximation for a single input and present a basis projection approximation algorithm. We apply the basis projection approximation algorithm on Fast Fourier Transform (FFT) for 8x8 image block by using polynomial basis and Haar wavelet basis in Section III. In Section IV, We define the optimal approximation for input set and present an estimation algorithm by using the basis projection approximation for single input. In Section V, we define the complexity distortion function and basis set complexity distortion function (BSCDF) for linear transform approximation on input set. We also propose a fast algorithm to estimate basis set complexity distortion function (BSCDF) and present how to estimate optimal approximation using BSCDF. In Section VI, we discuss how to encode and decode metadata for resource adaptive approximations in real time. We show the experimental results in Section VII and conclude the paper in Section VIII.

II. Basis Projection Approximation

We now formally define the technical problem of linear transform approximation for a single input, and present our basis projection solution. Let T be the linear transform that we wish to approximate, and let x and y be the input and output vectors respectively (i.e. $y = Tx$). For definiteness, assume that T is an MxM matrix, and x and y are Mx1 dimensional vectors. Let $C(T)$ represent the computational complexity of the transform T. Let $\tilde{T}$ be the approximate transform and $C(\tilde{T})$ be the computational complexity of the approximation. Let the computational resources available to compute the transform be C. Then the optimal approximate transform $\tilde{T}^*_x(C)$ for the input x for available computational resource C is defined as follows:

$$\tilde{T}_x^*(C) \triangleq \arg\min_{\tilde{T}:C(\tilde{T})\le C} \left\| Tx - \tilde{T}x \right\|, \qquad (1)$$

where $||\cdot||$ is the norm operator. The equation indicates that the optimal approximate transform $\tilde{T}^*_x(C)$ minimizes output distortion while satisfying computational complexity constraints C. Note that the optimal approximation transform might not be computable because the size of searching space for all possible approximations is infinite.

In this paper, when we refer to complexity, it is *computational* complexity of the transform. We shall assume that a single real addition, subtraction, or multiplication use equivalent computing costs and they are all considered to cost one operation. This is also true for some of the DSP chips. The case when the costs are different is easily handled by using appropriate weights in the calculations.

We now propose our *basis projection approximation* technique which is computable and scalable. Using this technique, a linear transform can be efficiently approximated with low computational complexity. We represent the approximation $\tilde{T}$ as a composition of operators: $\tilde{T} = TP$, where P is a linear projection operator (MxM matrix). In order to see how this can be used, let $P = B_k B_k^{\mathrm{T}}$ where B_k includes k orthonormal basis (e.g. Haar wavelet basis). B_k is an Mxk matrix with only k column vectors and the column vectors of B_k are orthonormal. Then the output and the distortion are calculated as follows:

$$\tilde{T}x = TB_k B_k^T x$$
$$\| \tilde{T}x - Tx \| = \| T(I - B_k B_k^T)x \| \quad (2)$$

This decomposition allows us to compute $\tilde{T}x$ into two steps – (a) project x onto a set of basis B_k (i.e. $B^T_k x$), then (b) project the result onto TB_k. The significant advantage is that TB_k is *independent of the input,* and can be pre-computed and stored offline. We only need compute $B^T_k x$ and combine with the stored TB_k matrix during real-time computation. A good choice of basis set B_k, can ensure that overall computational complexity of two step projections is significantly less than the computational complexity of the exact transform $C(T)$. Note that if B_k were an MxM matrix (i.e. k=M) which means B_k is complete, and then there is no approximation error. The general basis projection approximation diagram is shown in Figure 2.

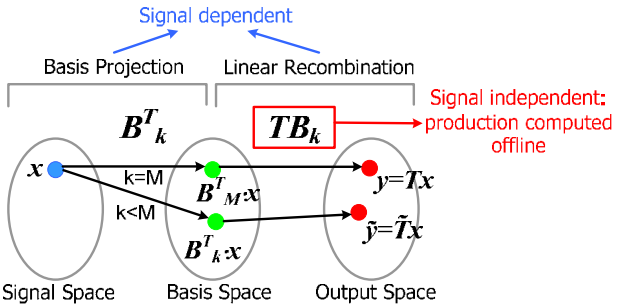

Figure 2: Diagram of basis projection approximation. x is the input, T is the exact transform operator, B_k is the basis set and $\tilde{T}$ is the approximation operator.

In the next section, we shall show how to use this basis projection approximation technique for 2D FFT approximation on 8x8 image block.

III. FFT Approximation

In this section, we apply basis projection approximation technique on the 2D FFT approximation, using two orthogonal basis set (orthogonal polynomial basis and Haar wavelet basis). We chose the FFT as it is a widely used linear transform in multimedia applications. Our approach is *complementary* to aready existing efficient FFT implementations.

We analyze the computational complexity of 2D FFT approximation for an 8x8 image block. Mathematically, an 8x8 image block can be represent as a 64x1 real vector and FFT operator can be represented as a 64x64 complex matrix. Note the exact 2D FFT has a *fixed* computational complexity, over all inputs. It is straightforward to show that the 2D FFT for an 8x8 real block requires 596 operations.

A. FFT approximation using orthogonal polynomial basis

We now propose FFT approximation using 2D separable orthogonal polynomial basis (Legendre polynomials). Table 1 shows these orthogonal polynomial basis functions at three resolutions. Note that higher resolution includes the basis function of lower resolution (i.e. resolution 0, 1, 2 have 1, 3, and 5 basis functions respectively).

Table 1: Orthogonal polynomial basis functions

Resolution	Orthogonal polynomial basis function
0	1
1	x
	y
2	$(3x^2-1)/2$
	$(3y^2-1)/2$

These basis functions can be easily represented using basis matrix B_k. Since we deal with 8x8 image block, B_k is a 64xk matrix. k equals 1, 3 and 5 at resolution J=0,1 and 2 respectively.

The forms of FFT matrices of polynomial basis functions are shown in Figure 3. The approximation using polynomial basis only estimate the FFT coefficients at the first row and the first column. This is because the polynomial basis we use are x-y separate. For example, the basis $(3y^2-1)/2$ only changes along the y direction with no change along the x direction. Thus, in the x direction, the energy of basis $(3y^2-1)/2$ focuses on the zero frequency (ω_x=0). Therefore, the FFT of basis $(3y^2-1)/2$ only has nonzero coefficients on the first column.

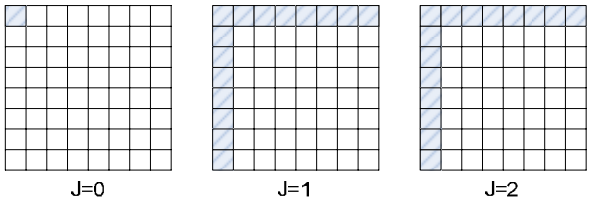

Figure 3: FFT matrix pattern for polynomial basis at resolution J=0,1,2. Shaded regions are the non-zero components and do not indicate the magnitude.

Table 2 shows the computational complexity of FFT approximation using orthogonal polynomial basis (Legendre polynomials). As we discussed in Section II, the computational complexity of FFT approximation using polynomial basis projection includes two parts: (a) Projection of input onto B_k (i.e. $B^T_k x$) and (b) projection of the result on TB_k. The exact FFT transform of polynomial basis TB_k is pre-computed and stored offline.

Table 2: Complexity (number of operations) of FFT approximation using polynomial basis projection

Operation \ Resolution	J=0	J=1	J=2	Exact FFT
Projection onto B_k	63	141	159	
Projection onto TB_k	0	6	28	
Total	63	147	187	596

We can find that the computational complexity of first projection (projection of input x onto the polynomial basis B_k) is much more than the second projection (projection of result on to TB_k). This is because of two reasons. First, most elements in the result of exact FFT on polynomial basis (i.e. TB_k) are zero (shown in Figure 3). The second reason is that the number of basis is small. At resolution J=0,1,2, we only have 1, 3, 5 polynomial basis

(k=1,3,5). This results in the smaller dimension of TB_k (Mxk).

B. FFT approximation using Haar wavelet basis

In this section, we present FFT approximation using Haar wavelet basis projection. The 2D nonstandard Haar wavelet basis decomposition [31] for an 8*8 image block (x) can be represented as:

$$x'_J = c^0_{0,0}\phi\phi^0_{0,0} + \sum_{j=0}^{J-1}\sum_{k=0}^{2^j-1}\sum_{l=0}^{2^j-1}(d^j_{k,l}\phi\psi^j_{k,l} + e^j_{k,l}\psi\phi^j_{k,l} + f^j_{k,l}\psi\psi^j_{k,l}), \quad (3)$$

where x'_J is the Haar wavelet approximation of image block x at the J^{th} resolution, $c^0_{0,0}$ and $\phi\phi^0_{0,0}$ are the scaling coefficient and scaling function respectively, $d^j_{k,l}$ and $\phi\psi^j_{k,l}$ are the $(k,l)^{th}$ horizontal wavelet coefficient and function at the $(j+1)^{th}$ resolution, $e^j_{k,l}$ and $\psi\phi^j_{k,l}$ are the $(k,l)^{th}$ vertical wavelet coefficient and function at the $(j+1)^{th}$ resolution, $f^j_{k,l}$ and $\psi\psi^j_{k,l}$ are the $(k,l)^{th}$ diagonal wavelet coefficient and function at the $(j+1)^{th}$ resolution. The nonstandard Haar wavelet basis is shown in Figure 4. The plus signs and minus signs are +1 and -1 and the blank region is 0. Each wavelet function is an 8x8 matrix. Note the higher resolution basis set includes the basis in the lower resolution. In Figure 4, there are 1, 4, 16 Haar wavelet basis at resolution J=0,1,2 respectively.

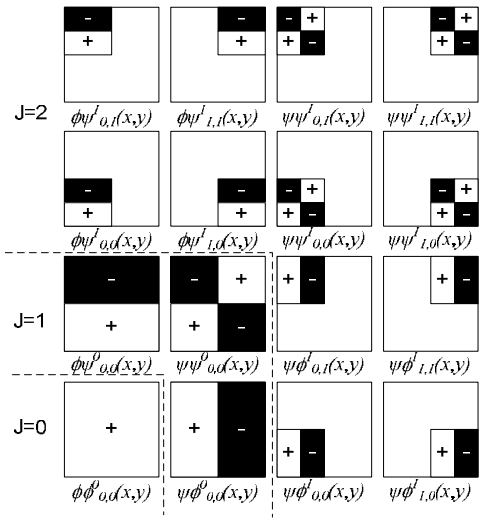

Figure 4: 2D nonstandard Haar wavelet basis

The 2D Haar wavelet basis can be easily represented using basis matrix B_k. B_k is a 64xk matrix, each column is a basis. k equals 1, 4 and 16 at resolution J=0,1 and 2 respectively. The higher resolution basis set includes the basis at the lower resolution. Since Haar wavelet basis are orthogonal, the columns of B_k are orthogonal. We do not consider resolution J=3, because when J=3 the Haar wavelet basis is complete for 8x8 image block and the basis projection approximation is equivalent to the exact FFT. The patterns of FFT matrices (i.e. TB_k) of Haar wavelet basis functions are shown in Figure 5. We find that more non-zero coefficients are introduced as the resolution J increases. This is because the higher resolution Haar wavelet basis contribute to the higher frequency FFT coefficients. At basis resolution J=2, the zero coefficients only lie on the fifth column and the fifth row. These frequencies are related to the Haar wavelet basis that only lie at resolution J=3. Comparing with FFT matrices of polynomial basis functions (ref. Figure 3), we can see FFT matrices of Haar wavelet basis have more nonzero coefficients at resolution J=1 and 2.

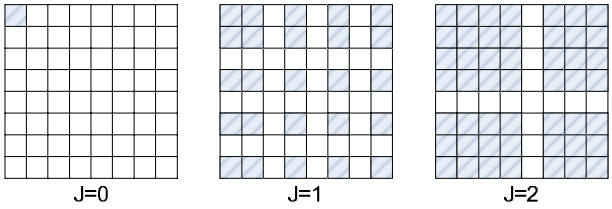

Figure 5: FFT matrix pattern for Haar wavelet basis at resolution J=0,1,2. Shaded regions are the non-zero components and do not indicate the magnitude.

In similar manner with polynomial basis projection, the computation complexity also includes the complexity of projecting input x on B_k and the complexity of projecting $B^T_k x$ on TB_k (ref. (2)). The exact FFT transform of Haar wavelet basis (i.e. TB_k) is pre-computed and stored offline. Table 3 shows the computational complexity of FFT approximation using Haar wavelet basis projection.

Table 3: Computational complexity (number of operations) of FFT approximation using Haar wavelet basis projection

Operation \ Resolution	J=0	J=1	J=2	Exact FFT
Projection onto B_k	63	68	88	
Projection onto TB_k	0	11	95	
Total	63	79	183	596

We can see that as the resolution J increases, complexity of projection of input x onto Haar wavelet basis B_k increases slowly while the complexity of projection of $B^T_k x$ onto TB_k increases rapidly. And at resolution J=2, the projection of $B^T_k x$ onto TB_k requires more complexity than the projection of input x onto Haar wavelet basis B_k. This is because we save more computations in the projection of input x onto the higher resolution basis by using the intermediate results, which are computed during the projection of the input x onto the lower resolution basis.

We now compare the computational complexity of FFT approximation using polynomial basis projection (FFT_{poly}) with FFT approximation using Haar wavelet basis projection (FFT_{Haar}). They have the same complexity at resolution J=0. This is because both approximations only compute the lowest frequency coefficient (top-left coefficient) which requires 63 operations. At resolution J=1, the complexity of FFT_{Haar} is much less than $FFT_{poly.}$ This is because in projection onto B_k, FFT_{Haar} saves more computations due to the intermediate results computed during the projection onto the low resolution basis. At resolution J=2, the complexity of FFT_{Haar} is close to the complexity of FFT_{poly}. Although FFT_{Haar} saves more computations in projection of input x onto basis B_k, it costs more computations in another projection – projection $B_k x$ onto TB_k. This is because the number of basis elements – k increases exponentially (k=$2^{2(J-1)}$) with respect to

resolution J in FFT_{Haar} while k increase linearly (k=2J+1) in FFT_{poly}. At resolution J=2, FFT_{Haar} has 16 basis while FFT_{poly} only has 5 basis.

IV. Linear Transform Approximation for Input Set

In this section, we define the technical problem of linear transform approximation for *input set* and present our solution by using approximation for *single input* (ref. Section II). For example, we assume that we compute the FFT approximation for all 8x8 image blocks of a given image. Each image block is a *single input* and the entire image is the *input set*. The problem is to select proper approximation operator for each image block such that the overall transform computational complexity satisfies the resource complexity constraint and the overall distortion is minimized.

Let us denote the input set (e.g. image) as $\boldsymbol{X}=\{x_i\}$ where x_i is the ith element (e.g. image block). Let $T\boldsymbol{X}=\{Tx_i\}$ be the results of exact transform T for the input set. Let $\tilde{\boldsymbol{T}}=\{\tilde{T}_i\}$ be the approximation set where $\tilde{T}_i$ is the approximation operator for the corresponding input element x_i. Thus the approximation result is $\tilde{\boldsymbol{T}}\boldsymbol{X}=\{\tilde{T}_i x_i\}$. In the case of the FFT approximation for all 8x8 image blocks of an image, T is the FFT operator. Each image block is an input element x_i and the entire image is an input set $\boldsymbol{X}$. The result of exact FFT for image block x_i is Tx_i. The result of approximate FFT of the image is $\tilde{\boldsymbol{T}}\boldsymbol{X}$ ($\tilde{\boldsymbol{T}}\boldsymbol{X}=\{\tilde{T}_i x_i\}$), where $\tilde{T}_i$ is the FFT approximation operator for image block x_i.

For the sake of definiteness, assume that each element of input set (i.e. x_i) is an Mx1 dimensional vector, and its corresponding approximate operator (i.e. $\tilde{T}_i$) is an MxM matrix. In the example of computing approximate FFT of 8x8 image blocks, M equals 64 (each image block can be represented as a 64x1 vector).

We define the computational complexity of approximation set as the average number of operations per input element:

$$C(\tilde{\boldsymbol{T}}) \triangleq \frac{1}{N}\sum_{i=1}^{N} C(\tilde{T}_i), \tag{4}$$

where $C(\tilde{T}_i)$ is the computation complexity of approximation operator $\tilde{T}_i$ for the ith input element, N is the number of input elements.

The optimal approximation $\tilde{\boldsymbol{T}}_{\boldsymbol{X}}^{*}(C)$ for an input set $\boldsymbol{X}$ for the linear transform T under a computational complexity constraint C is defined as follows:

$$\tilde{\boldsymbol{T}}_{\boldsymbol{X}}^{*}(C) \triangleq \arg\min_{\tilde{\boldsymbol{T}}: C(\tilde{\boldsymbol{T}})\le C} \frac{1}{N}\sum_{i=1}^{N} \| Tx_i - \tilde{T}_i x_i \|, \tag{5}$$

where $C(\tilde{\boldsymbol{T}})$ is the computational complexity of approximation set $\tilde{\boldsymbol{T}}$ (ref. (4)) and N is the cardinality of the input set $\boldsymbol{X}$ ($|\boldsymbol{X}|=N$). The equation indicates that the optimal approximation is the combination of approximation operators with minimum average output distortion while satisfying computational complexity constraint C. The optimal approximation $\tilde{\boldsymbol{T}}_{\boldsymbol{X}}^{*}(C)$ is related to the computational complexity constraint C and the input set $\boldsymbol{X}$. Note that $\tilde{\boldsymbol{T}}_{\boldsymbol{X}}^{*}(C)$ may not be unique. $\boldsymbol{T}_{\boldsymbol{X}}^{*}(C)$ may not be computable because the size of searching space for all possible approximations is infinite.

We now present our estimation solution. We simplify the problem by adding a constraint that each approximation element $\tilde{T}_i$ belongs to a given basis projection approximation set (e.g. Haar) $\mathbf{B}$ ($\mathbf{B}=\{b_j\}$). Each element of $\mathbf{B}$ is a basis projection approximation operator (ref. Section II). The optimal approximation set $\tilde{\boldsymbol{T}}_{\boldsymbol{X}}^{*}(C|\mathbf{B})$ under the constraint $\mathbf{B}$ is defined as follows:

$$\tilde{\boldsymbol{T}}_{\boldsymbol{X}}^{*}(C\,|\,\mathbf{B}) \triangleq \arg\min_{\tilde{\boldsymbol{T}}=\{\tilde{T}_i\}:\tilde{T}_i\in\mathbf{B}, C(\tilde{\boldsymbol{T}})\le C} \frac{1}{N}\sum_{i=1}^{N} \| Tx_i - \tilde{T}_i x_i \|. \tag{6}$$

In the following two sections, we shall show how to search for the optimal approximation set $\tilde{\boldsymbol{T}}_{\boldsymbol{X}}^{*}(C|\mathbf{B})$ under different complexity constraints C and how to do adaptive approximations in real-time applications.

V. Complexity Distortion Function

In this section, we will (a) establish a theoretical framework for the complexity distortion function $C(D)$ for linear transform approximation, and (b) obtain a useful estimate of the C-D function using basis set complexity distortion function (BSCDF).

The complexity distortion function $C(D)$ of a linear transform approximation for an input set $\boldsymbol{X}$ provides the lower bound of distortion for optimal approximation (ref. (5)) under any complexity constraint C. The BSCDF provides the lower bound of distortion for optimal constraint approximation (ref. (6)).

A. Complexity Distortion Function Definition

We now discuss complexity distortion function of linear transform approximation for input set $\boldsymbol{X}$. We are motivated by the well established definitions from rate distortion theory [7], to define the relationship between the computational complexity and distortion. For the sake of definiteness, let us assume that the input set $\boldsymbol{X}$ is an image and the linear transform T is the FFT. It is easy to see that these results generalize to arbitrary input and linear transforms.

Let $\boldsymbol{X}$ be an image which is divided into N blocks ($\boldsymbol{X}=\{x_i,\ i=1,..,N\}$). x_i is the ith image block with size 8x8. The computational complexity of FFT approximation of $\boldsymbol{X}$ (i.e. $C(\tilde{\boldsymbol{T}})$) is defined as the average number of operations per block (ref. (4)). The *distortion* $D_{\boldsymbol{X}}(\tilde{\boldsymbol{T}})$ due to the FFT approximation set $\tilde{\boldsymbol{T}}=\{\tilde{T}_i\}$ for an image $\boldsymbol{X}=\{x_i\}$ is defined as follows:

$$D_{\boldsymbol{X}}(\tilde{\boldsymbol{T}}) = \frac{1}{N}\sum_{i=1}^{N} D_{x_i}(\tilde{T}_i) = \frac{1}{N}\sum_{i=1}^{N} d(Tx_i, \tilde{T}_i x_i), \tag{7}$$

where $\tilde{\boldsymbol{T}}$ is the FFT approximation set ($\tilde{\boldsymbol{T}}=\{\tilde{T}_i\}$, i=1,…,N), each element $\tilde{T}_i$ is the FFT approximation operator for the corresponding image block x_i, $D_{xi}(\tilde{T}_i)$ is the distortion due to the approximation $\tilde{T}_i$ for the ith image block (i.e. x_i), and $d(\cdot)$ is the distortion measure. We use root mean square (RMS) as the distortion measure. The *complexity distortion region* $R_{\boldsymbol{X}}(C,D)$ for an image $\boldsymbol{X}$ for the FFT approximation is the closure of the set of achievable complexity distortion pairs (C,D).

Definition: The *complexity distortion function* $C^T_{\boldsymbol{X}}(D)$ for an image $\boldsymbol{X}$, for the approximation of linear transform T is defined as the infimum of all complexities C such that (C,D) is in the achievable complexity distortion region for a given distortion D.

$$C^T_{\boldsymbol{X}}(D) = \inf_{\tilde{\boldsymbol{T}}: D_{\boldsymbol{X}}(\tilde{\boldsymbol{T}}) \le D} C(\tilde{\boldsymbol{T}}), \tag{8}$$

where $C(\tilde{\boldsymbol{T}})$ and $D_{\boldsymbol{X}}(\tilde{\boldsymbol{T}})$ are the computational complexity (ref. eq.(4)) and distortion (ref. eq.(7)) of FFT approximation $\tilde{\boldsymbol{T}}$ for the image $\boldsymbol{X}$ respectively. Since $C^T_{\boldsymbol{X}}(D)$ is the minimum complexity over increasingly larger sets as D increases, $C^T_{\boldsymbol{X}}(D)$ is non-increasing in D. In [5], we prove that $C^T_{\boldsymbol{X}}(D)$ is convex.

In the similar manner to rate distortion theory [7], the complexity distortion function can be defined in another way as *distortion complexity function* $D^T_{\boldsymbol{X}}(C)$. $D^T_{\boldsymbol{X}}(C)$ is equivalent to the complexity distortion function $C^T_{\boldsymbol{X}}(D)$. It is straightforward to prove the equivalence. The $D^T_{\boldsymbol{X}}(C)$ provides the lower bound of distortion for the optimal approximation (ref. eq.(5)) under any complexity constraint C. Note that the $C^T_{\boldsymbol{X}}(D)$ (or $D^T_{\boldsymbol{X}}(C)$) might not be computable or achievable.

B. Basis Set Complexity Distortion Function (BSCDF)

We now propose the use of a basis projection approximation set (e.g. polynomial / Haar) to estimate the complexity distortion function $C(D)$. A *basis projection approximation set* leads to a *Basis Set Complexity Distortion Function* (BSCDF). The BSCDF is the complexity distortion function based on the condition that any FFT approximation operator of image block x_i (i.e. $\tilde{T}_i$) must belong to a given basis projection approximation set $\mathbf{B}$ (i.e. $\tilde{T}_i \in \mathbf{B}$). The *Basis Set Complexity Distortion Function* (BSCDF) $C^T_{\boldsymbol{X}}(D|\mathbf{B})$ for an image $\boldsymbol{X}$, for the FFT approximation using basis projection approximation set $\mathbf{B}$ is defined as follows:

$$C^T_{\boldsymbol{X}}(D \mid \mathbf{B}) = \inf_{\tilde{\boldsymbol{T}}: \tilde{T}_i \in \mathbf{B},\, D_{\boldsymbol{X}}(\tilde{\boldsymbol{T}}) \le D} C(\tilde{\boldsymbol{T}}), \tag{9}$$

where $C(\tilde{\boldsymbol{T}})$ and $D_{\boldsymbol{X}}(\tilde{\boldsymbol{T}})$ are the computational complexity (ref. (4)) and distortion (ref. (7)) of the FFT approximation $\tilde{\boldsymbol{T}}$ for the image $\boldsymbol{X}$ respectively. Each element of $\tilde{\boldsymbol{T}}$ (i.e. $\tilde{T}_i$), that is the FFT approximation operator of the corresponding image block x_i, belongs to approximation set $\mathbf{B}$. In similar manner to distortion complexity function $D^T_{\boldsymbol{X}}(C)$, we can define the basis set distortion complexity function (BSDCF) $D^T_{\boldsymbol{X}}(C|\mathbf{B})$, which is equivalent to the BSCDF.

In this paper, we present two available basis projection approximation sets for the FFT approximation: (a) Haar wavelet basis projection set $\mathbf{B^H}$ and (b) Orthogonal polynomial basis projection set $\mathbf{B^P}$. Both sets have four elements ($\mathbf{B^H}=\{b^H_j,\ j=0,1,2,3\}$, $\mathbf{B^P}=\{b^P_j,\ j=0,1,2,3\}$) where b^H_0, b^H_1, b^H_2 are FFT approximation using Haar wavelet basis projection at resolution J=0,1,2, b^P_0, b^P_1, b^P_2 are FFT approximation using orthogonal polynomial basis projection at resolution J=0,1,2, b^H_3 and b^P_3 are exact FFT. If the image size is large, the computation of BSCDF is very expensive even if we only use $\mathbf{B^H}$ or $\mathbf{B^P}$ which has four operators. The number of possible approximation permutation of all image blocks is N^4 where N is the number of image blocks. Hence, we present a fast step-wise algorithm to estimate BSCDF using Haar wavelet basis projection set $\mathbf{B^H}$. This procedure also holds true for other basis projections such as $\mathbf{B^P}$.

The step-wise algorithm starts from the *(C,D)* pair such that exact FFT (i.e. b^H_3) is used for all image blocks and ends with *(C,D)* pair where FFT approximation for all image blocks are based on Haar wavelet basis projection approximation at resolution J=0 (i.e. b^H_0). At each step, we first select the image block using FFT approximation operator b^H_1, b^H_2 or b^H_3 such that changing its approximation to the lower complexity approximation in $\mathbf{B^H}$ (decrease index number by 1 e.g. $b^H_3 \rightarrow b^H_2$ or $b^H_2 \rightarrow b^H_1$ or $b^H_1 \rightarrow b^H_0$) introduces the maximum ratio between complexity decrement and distortion increment ($|\Delta C/\Delta D|$), i.e. we look for the block that maximize the rate of change, thereby intuitively being close to the tangent to the C-D lower bound. Then we change the approximation operator of the selected block to the lower complexity approximation (decreasing operator index by 1) and get a new *(C,D)* pair. Hence, we will obtain a C-D curve estimation by repeating this procedure. Our fast stepwise algorithm only generates $3N$-1 C-D pairs. Comparing with searching for exact BSCDF which search for lower bound from N^4 possible C-D pairs, our algorithm saves a lot of computations. However, our algorithm only introduces a small distortion. The estimation results are shown in Figure 7. We can see that the exact BSCDF and estimation are very close for both polynomial basis approximation and Haar wavelet basis approximation.

In our fast stepwise algorithm, at each step, we obtain a approximation index list $\boldsymbol{L}=\{\boldsymbol{L}(i),\ i=1,\ldots,N\}$, each element $\boldsymbol{L}(i)$ indicates the index of approximation operator for image block x_i. For example, at the beginning when all image block use exact FFT b^H_3, l_i=3, i=1,…,N. Using approximation index list, we can estimate the optimal constraint approximation set $\tilde{\boldsymbol{T}}^*_{\boldsymbol{X}}(C|\mathbf{B^H})$ (ref. (6)). Let us denote the overall complexity and approximation index list at the k^{th} step as C_k and $\boldsymbol{L}_k$. The optimal constraint approximation set is represented as follows:

$$\tilde{\boldsymbol{T}}_{\boldsymbol{X}}(C_k \mid \mathbf{B^H}) = \{\tilde{T}_i \mid \tilde{T}_i = b^H_{\boldsymbol{L}_k(i)},\ i = 1,\ldots,N\}. \tag{10}$$

VI. RESOURCE ADAPTIVE APPROXIMATIONS

In this section, we shall discuss how to do resource adaptive FFT approximation in real applications. The idea is we select several operating points or (C, D) pairs along the estimation of basic set complexity distortion curve (BSCDF) and save the corresponding (C, D) values and approximation index list (ref. Section V.B) in the image metadata during the encoding. In the real FFT approximation, we select an appropriate operating point which guarantees that the complexity constraint is satisfied and use the corresponding approximation index list to do the FFT approximation.

We use operating points instead of the whole BSCDF curve because of two reasons. First, the computation for

estimating BSCDF is relatively expensive comparing with the complexity of transform approximation (e.g. FFT approximation). The second reason is that storing the entire BSCDF in the metadata as results in a huge image file. Therefore, we pre-compute the BSCDF *offline* using fast stepwise algorithm (ref. Section V.B), select optimal operating points on the estimated BSCDF and storing these operating points (C_k, D_k) and their corresponding approximation index list $\boldsymbol{L_k}$ in the metadata of image.

A. Operating point selection

We now show how to approximate the C-D curve, using K operating points on the curve. We use our step-wise algorithm (see Section V.B), to estimate the C-D curve. For the sake of simplicity, we use the distortion complexity function $D(C)$ to denote C-D curve in this section. Note that distortion complexity function is equivalent to the complexity distortion function $C(D)$ (ref. Section V.A).

Let us denote the $K+2$ operating points (K operating points on $D(C)$ and two end points of $D(C)$) as (C_k, D_k), $k=0,...,K+1$, $C_0 \geq \ldots \geq C_{K+1}$, $D_0 \leq \ldots \leq D_{K+1}$. ($C_0$, D_0) and (C_{K+1},D_{K+1}) are two ending points. Since the two ending points (C_0, D_0) and (C_{K+1},D_{K+1}) are fixed when the distortion complexity function $D(C)$ is given, we only need to find the optimal K operating points on $D(C)$. Figure 6 shows a distortion complexity function with 5 (K=3) operating points.

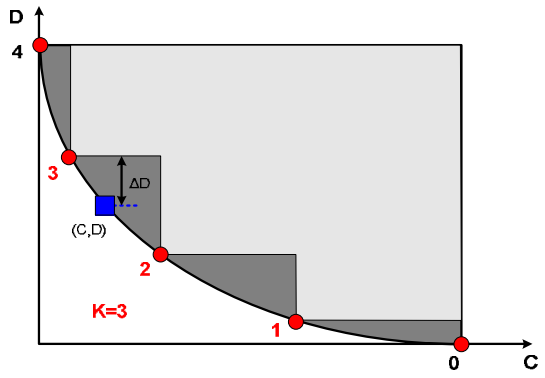

Figure 6: Five operating points (K=3) on distortion complexity curve. Points with label 1,2,3 are operating points on $D(C)$ and points with label 0, 4 are two end points.

When the available complexity C is in the interval [C_k, C_{k-1}), the operating point (C_k,D_k) is selected because it introduces minimum distortion amongst all operating points while satisfying the complexity constraint ($C_k \leq C$). The introduced distortion is D_k-$D(C)$. The average distortion over the interval [C_k, C_{k-1}) is represented as:

$$\Delta D_k = \int_{C_k}^{C_{k-1}} p(C)[D_k - D(C)]dC \,, \qquad (11)$$

where $p(C)$ is the pdf of complexity constraint and $D(C)$ is the inverse function of $C(D)$. Thus, the overall introduced distortion is:

$$\Delta D = \sum_{k=1}^{K+1} \Delta D_k = \sum_{k=1}^{K+1} \int_{C_k}^{C_{k-1}} p(C)[D_k - D(C)]dC \,. \qquad (12)$$

Therefore, the optimal selection is choosing the K operating points with the minimum introduced distortion. In [5] we prove that when $K=1$, the optimal single operating point for a C-D curve starting from (C_0,D_0) to (C_2,D_2) ($C_0 > C_2$) is:

$$\begin{aligned} (C_1^*, D_1^*) &= \arg\max_{(c,d)\in\xi}[(D_2 - d)\cdot(f(C_0) - f(c))] \\ f(c) &= \int_{-\infty}^{c} p(C)dC \end{aligned} \,, \qquad (13)$$

where ξ is the set of all (C,D) pairs on the $C(D)$ curve.

We show in [5] that if $\{(C_k,D_k), k=1,...,K\}$ are optimal K operating points of C-D curve starting from (C_0,D_0) to (C_{K+1},D_{K+1}) and $C_0 \geq \ldots \geq C_{K+1}$, $D_0 \leq \ldots \leq D_{K+1}$, ($C_k$,$D_k$) is the optimal single operating point of the sub-curve of $C(D)$ which starts from (C_{k-1},D_{k-1}) to (C_{k+1},D_{k+1}). Thus, we can use an iterative algorithm to obtain the K operating points. In the algorithm, we initially select K operating points randomly and order them in complexity value. At each iteration, we update (C_k,D_k) $k=1,..K$ with the optimal single operating point (ref. eq. (13)) of the sub-curve between (C_{k-1},D_{k-1}) to (C_{k+1},D_{k+1}) (assuming that $p(C)$ is uniform). Finally, $\{(C_k,D_k)\}$ will converge to the optimal operating points.

B. Encoding and using metadata

We now show how to compute metadata using K operating points. In the image encoding phrase, we compute the estimation of basis set complexity distortion curve (BSCDF) for Haar and polynomial basis set (i.e. $\mathbf{B^H}$ and $\mathbf{B^P}$) for the image and select the optimal one. Second, we select the K operating points along the C-D curve of selected basis set. Thus, with the two ending points of the C-D curve, we have $K+2$ operating points denoted as (C_k,D_k), $k=0,...,K+1$, $C_0 \geq \ldots \geq C_{K+1}$, $D_0 \leq \ldots \leq D_{K+1}$. Obviously, ($C_0$,$D_0$) and ($C_{K+1}$,$D_{K+1}$) are the two ending points of the C-D curve. Finally, we save three things in the image metadata: (a) basis bit: (0: polynomial basis and 1: Haar wavelet basis), (b) $K+2$ (C,D) values: (each is a 16 bit float) and (c) K approximation index lists $\boldsymbol{L_k}$, $k=1,...K$: $\boldsymbol{L_k}$ is corresponding with the (C_k,D_k) with size N where N is the number of 8*8 image blocks in the image. $\boldsymbol{L_k}(i)$ is approximation operator index for the image block x_i. For example, assuming $\boldsymbol{L_k}(i)$=2 and we use Haar wavelet basis set $\mathbf{B^H}$ to approximate FFT, the approximation operator for image block x_i is FFT approximation using Haar wavelet basis projection at resolution 2 (i.e. $b^H{}_2$). In this paper, both polynomial basis set $\mathbf{B^P}$ and Haar wavelet basis set $\mathbf{B^H}$ have four elements. Hence, $\boldsymbol{L_k}(i)$ needs 2 bits. Since the approximation index list corresponding with (C_0,D_0) and (C_{K+1},D_{K+1}) are vectors whose elements are all three and zero respectively. We need not save them in the metadata. Therefore, the size of metadata is $2KN+32(K+2)+1$ bits. When we set $K=4$, metadata size is about 1/64 of the gray level image.

In the FFT approximation phrase, the image and computational complexity constraint C are given. We first read the basis bit and select corresponding basis set. Then we select C_k such that $C_{k-1} > C \geq C_k$ from the list of complexity values saved in the image metadata. Finally, we use $\boldsymbol{L_k}$ which is corresponding to C_i to do FFT

approximation. The complexity of this approximation is guaranteed to be less than the complexity constraint.

VII. EXPERIMENTAL RESULTS

In this section, we present our experimental results for (a) estimating the basis set complexity distortion function (BSCDF), (b) comparing FFT approximation using Haar wavelet basis and polynomial basis and (c) optimal operating point selection. We used a well known image – Lena at resolution 256x256 and 64x64 to test our framework.

A. Estimation of BSCDF

We now present our experimental results for estimating the BSCDF (ref. Section V.B). We select Lena image with resolution 64x64 as the input set $\boldsymbol{X}$ that contains 64 8x8 image blocks ($|\boldsymbol{X}|$=64). We select the resolution 64x64 rather than 256x256 here because the computational complexity of searching *exact* BSCDF (ref. Section V.B) increases exponentially with the number of image blocks. The number of achievable C-D pairs for Lena 256x256 and Lena 64x64 are 1024^4 and 64^4 respectively. Therefore computing the *exact* CCDF for Lena 256x256 is very expensive.

We use two basis projection approximation sets for the FFT approximation: (a) Haar wavelet basis projection set $\mathbf{B^H}$ and (b) Orthogonal polynomial basis projection set $\mathbf{B^P}$. Both sets have four elements ($\mathbf{B^H}=\{b^H_j,\ j=0,1,2,3\}$, $\mathbf{B^P}=\{b^P_j,\ j=0,1,2,3\}$) where b^H_0, b^H_1, b^H_2 are FFT approximation using Haar wavelet basis projection at resolution J=0,1,2, b^P_0, b^P_1, b^P_2 are FFT approximation using orthogonal polynomial basis projection at resolution J=0,1,2, b^H_3 and b^P_3 are exact FFT.

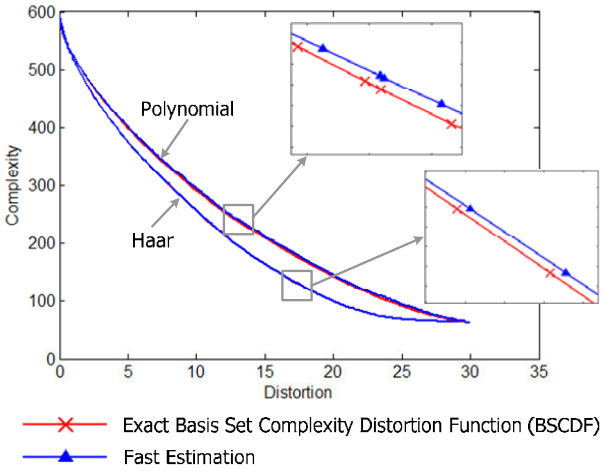

Figure 7. BSCDF estimation results for Lena image 64x64 for FFT approximation using polynomial basis set $\mathbf{B^P}$ and Haar wavelet basis set $\mathbf{B^H}$.

Figure 7 shows the exact basis set complexity distortion curve and the estimation using step-wise algorithm for polynomial basis set $\mathbf{B^P}$ and Haar wavelet basis set $\mathbf{B^H}$. We can see that two BSCDFs are non-increasing convex curves and the estimation results are very close to the exact BSCDFs. We use relative distortion difference to evaluate our fast estimation algorithm. The relative distortion difference τ is defined as follows:

$$\tau = \frac{\int_{C_0}^{C(T)} [\tilde{D}_X^T(C \mid \mathbf{B}) - D_X^T(C \mid \mathbf{B})] dC}{\int_{C_0}^{C(T)} D_X^T(C \mid \mathbf{B}) dC}, \tag{14}$$

where $D^T_X(C|\mathbf{B})$ and $\tilde{D}^T_X(C|\mathbf{B})$ are exact BSDCF and estimated BSDCF based on basis projection approximation set $\mathbf{B}$ respectively, $C(T)$ is the complexity of exact FFT, C_0 is the complexity to compute the lowest FFT coefficient ($C_0=C(b^H_0)= C(b^P_0)$=63 operations). The relative distortion difference for BSCDF estimation based on polynomial basis and Haar wavelet basis are 1.4% and 0.039% respectively. This shows that our BSCDF estimation algorithm is excellent.

B. Comapring FFT approximation using polynomail basis set and Haar wavelet basis set

We compare the FFT approximation using polynomial basis set $\mathbf{B^P}$ and Haar wavelet basis set $\mathbf{B^H}$ using estimation of basis set complexity distortion function BSCDF. The lower the BSCDF, the better the basis set. We can see FFT approximation using Haar wavelet basis $\mathbf{B^H}$ has lower BSCDF for both Lena image 64x64 (Figure 7) and Lena image 256x256 (Figure 8). We use relative distortion difference to qualify the difference between two basis sets. The relative distortion difference $\Delta_X(\mathbf{B^P}, \mathbf{B^H})$ between FFT approximation using $\mathbf{B^P}$ and $\mathbf{B^H}$ for image $\boldsymbol{X}$ is defined as follows:

$$\Delta_X(\mathbf{B^P}, \mathbf{B^H}) = \frac{\int_{C_0}^{C(T)} |\tilde{D}_X^T(C \mid \mathbf{B^P}) - \tilde{D}_X^T(C \mid \mathbf{B^H})| \, dC}{\min(\int_{C_0}^{C(T)} \tilde{D}_X^T(C \mid \mathbf{B^P}) dC, \int_{C_0}^{C(T)} \tilde{D}_X^T(C \mid \mathbf{B^H}) dC)}, \tag{15}$$

where $\tilde{D}^T_X(C|\mathbf{B^P})$ and $\tilde{D}^T_X(C|\mathbf{B^P})$ are estimation of BSDCF for image $\boldsymbol{X}$ for FFT approximation using polynomial basis set $\mathbf{B^P}$ and Haar wavelet basis $\mathbf{B^H}$ respectively. The relative distortion difference $\Delta_X(\mathbf{B^P}, \mathbf{B^H})$ for Lena image 64x64 and Lena image 256x256 are 21.2% and 15.1% respectively.

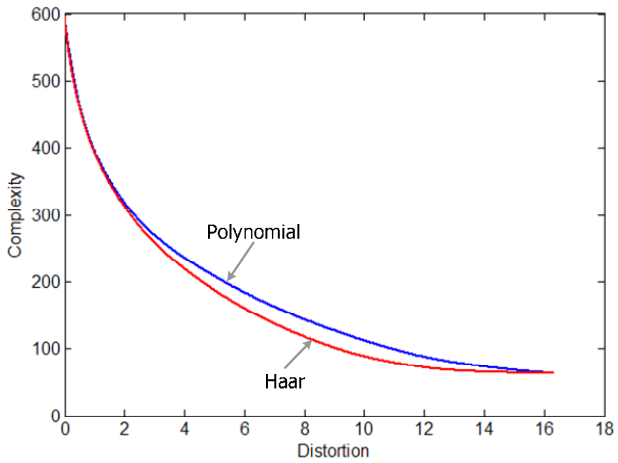

Figure 8. Estimation of basis set complexity distortion function (BSCDF) for Lena image 256x256 for FFT approximation using polynomial basis set $\mathbf{B^P}$ and Haar wavelet basis set $\mathbf{B^H}$.

The $\mathbf{B^H}$ is better than $\mathbf{B^P}$ because Haar wavelet basis projection approximates more FFT coefficients (higher

frequency) than polynomial basis projection (ref. Figure 3 and Figure 5) which results in smaller distortion. Additionally Haar wavelet basis projection (ref. Table 3) requires even less complexity than polynomial basis projection (ref. Table 2) at basis resolution J=1,2 for FFT approximation.

C. Operating point selection

We now present the experimental results for optimal operating point selection (ref. Section VI.A). Figure 9 shows the optimal operating point selection (*K=4*) results on the estimation of BSCDF for Lena image 256x256 for FFT approximation using Haar wavelet basis set $\mathbf{B}^{\mathbf{H}}$. The optimal operating points are selected using our iterative algorithm. For each operating point, we also show the corresponding recovered image by using exact inverse FFT (IFFT). The triangles in the figure are the two end points and the stars are the four optimal operating points.

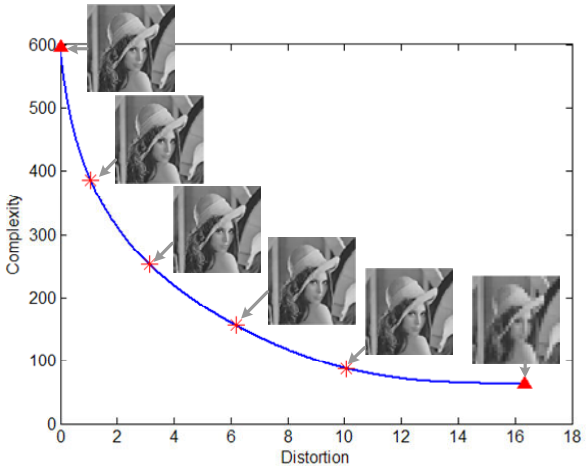

Figure 9. Optimal operating point selection (*K*=4) on the BSCDF for Lena image 256x256 for FFT approximation using Haar wavelet basis set $\mathbf{B}^{\mathbf{H}}$.

VIII. Conclusion

In this paper, we have attempted to create a systematic framework for linear transform approximation using orthogonal basis projection technique. There were three key ideas – (a) orthonormal basis functions to approximate the input, (b) we showed the existence of a convex complexity-distortion curve, and showed how to approximate the curve given a specific basis set, and (c) we finally showed how to compute operating points on the C-D curve, and embed metadata in the image. Our approach is generic, and applies to any linear transform.

Our experimental results on the Lena image are excellent. They showed (a) that our BSCDF estimation algorithm is close to the exact BSCDF. The relative error are 1.4% and 0.039% for polynomial basis and Haar wavelet basis respectively, (b) Haar wavelet basis set has better result than polynomial basis set for FFT approximation for Lena image and (c) We finally showed the results of optimal operating point selection.

This result can be extended in many directions. We plan to combine existing pruning techniques with our basis projection techniques. We also plan to combine different basis projection techniques (e.g. Haar and polynomials), for more efficient basis approximation. We also plan to extend our framework to inverse FFT (IFFT) approximations.

Acknowledgment

This research work is supported by the NSF Grant: NSF 03-08268: “Development Of Quality-Adaptive Media-Flow Architectures To Support Sensor Data Management”.

References

[1] F. Argenti, F. D. Taglia and E. D. Re (2002). "Audio decoding with frequency and compleixty scalability". *Proc. Inst. Elect. Eng., Vis., Imag, Signal Process.* **149**(3): 152-158.

[2] S. C. Chan and K. L. Ho (1991). "A new two-dimensional fast cosine transform algorithm". *IEEE Transaction on Acoust., Speech, Signal Processing* **39**: 481-485.

[3] W. Chen, C. H. Smith and S. D. Fralick (1977). "A fast computational algorithm for the discrete cosine transform". *IEEE Trans. Commun.* **COMM-25**: 1004-1009.

[4] Y. Chen, Z. Zhong and T.-H. Lan (2002). "Regulated complexity scalable MPEG-2 video decoding for media processors", *Circuits and Systems for Video Technology, IEEE Transactions on*, 678-687,

[5] Y. Chen and H. Sundaram (2005). "Approximate Linear Transforms for Real Time Applications". Arts Media and Engineering Program, Arizona State University, AME-TR-2005-17, 2005.

[6] Y. Chen and H. Sundaram (2006). "Basis Projection for Linear Transform Approximation in Real-Time Applications", *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Toulouse, France, May 2006.

[7] T. M. Cover and J. A. Thomas (1991). "Elements of information theory". *Wiley series in telecommunications*. New York, Wiley**:** xxii, 542.

[8] P. Duhamel and C. Guillemot (1990). "Polynomial transform computation of the 2-D DCT", *IEEE Int'l Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, 1515-1518, 1990.

[9] J. M. Emmert, J. A. Cheatham, B. Jagannathan, et al. (2003). "An FFT approximation technique suitable for on-chip generation and analysis of sinusoidal signals", *IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems*, 361-368, 3-5 Nov.

[10] E. Feig and S. Winograd (1992). "Fast algorithm for the discrete cosine transfom". *IEEE. Trans. on Signal Proc.* **40**(9): 2174-2193.

[11] H. S. Hou (1987). "A fast recursive algorithm for computing the discrete cosine transform". *IEEE Transaction on Acoust., Speech, Signal Processing* **ASSP-35**: 1455-1461.

[12] Z. Hu and H. Wan (2005). "A novel generic fast Fourier transform pruning technique and complexity analysis". *IEEE Trans. Signal Processing* **53**(1): 274-282.

[13] Y. Huang, J. Wu and C. Chang (2000). "A generalized output pruning algorithm for matrix-vector multiplication and its application to computing pruning discrete cosine transform". *IEEE Trans. Signal Processing* **48**(2): 561-563.

[14] I. R. ISMAEIL, A. DOCEF, F. KOSSENTINI, et al. (2001). "A computation-distortion optimized framework for efficient DCT-based video coding". *IEEE Transaction on Multimedia* **3**(3): 298-310.

[15] B. G. LEE (1984). "A new algorithm to compute the discrete cosine transform". *IEEE Transaction on Acoust., Speech, Signal Processing* **ASSP-32**: 1243-1245.

[16] C. S. LI, R. MOHAN and J. R. SMITH (1998). "Multimedia content description in the InfoPyramid", *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech, and Signal Processing*, Seattle WA, USA, 3789-3792, May 1998.

[17] J. LIANG and T. D. TRAN (2001). "Fast multiplierless approximations of the DCT with the lifting scheme". *IEEE Transactions on Signal Processing* **49**(12): 3032-3044.

[18] J. MAKHOUL (1980). "A fast cosine transform in one and two dimentions". *IEEE Transaction on Acoust., Speech, Signal Processing* **ASSP-28**: 27-34.

[19] H. MALVAR, A. HALLAPURO, M. KARCZEWICZ, et al. (2003). "Low-complexity transform and quantization in H.264/AVC". *IEEE Trans. CVST* **13**: 598-603.

[20] J. D. MARKER (1971). "FFT pruning". *IEEE Trans. Audio and Electroacoust* **19**(4): 305-311.

[21] M. MATTAVELLI, S. BRUNETTON and D. MLYNEK (1997). "Computational graceful degradation for video sequence decoding", *Proceedings, International Conference on Image Processing*, 330-333,

[22] S. H. NAWAB, A. V. OPPENHEIM, A. P. CHANDRAKASAN, et al. (1997). "Approximate Signal Processing". *The Journal of VLSI Signal Processing-Systems for Signal, Image, and Video Technology*. **15:** 177-200.

[23] W. PAN and A. ORTEGA (2000). "Complexity-scalable transform coding using variable complexity algorithms", *Proc. IEEE Data Compression Conf. (DCC-00)*, 263-272, Mar. 2000.

[24] M. V. D. SCHAAR and P. H. N. D. WITH (2000). "Near-lossless complexity-scalable embedded compression algorithm for cost reduction in DTV receivers". *IEEE Trans. Consumer Electron.* **46**(4): 923-933.

[25] M. V. D. SCHAAR and Y. ANDREOPOULOS (2005). "Rate-Distortion-Complexity Modeling for network and receiver aware adaptation". *IEEE Transaction on Multimedia* **7**(3): 471-479.

[26] A. SILVA and A. NAVARRO (2005). "Fast 8x8 DCT Pruning Algorithm", *ICIP 2005. IEEE International Conference on Image Processing*, 317-320, 11-14 Sept. 2005.

[27] A. SKODRAS (1994). "Fast discrete cosine transform pruning". *IEEE Trans. Signal Processing* **42**(7): 1833-1837.

[28] H. V. SORENSEN and C. S. BURRUS (1993). "Efficient computation of the DFT with only a subset of input or output points". *IEEE. Trans. on Signal Proc.* **41**(3): 1184-2000.

[29] D. M. SOW and A. ELEFTHERIADIS (2003). "Complexity distortion theory". *Information Theory, IEEE Transactions on*. **49:** 604-608.

[30] S. SRINIVASAN and S. REGUNATHAN (2005). "Computationally efficient transforms for video coding", *IEEE International Conference on Image Processing (ICIP 05)*, 11-14 Sept. 2005.

[31] E. J. STOLLNITZ, T. D. DEROSE and D. H. SALESIN (1995). "Wavelets for computer graphics: A primer, part 1". *IEEE Computer Graphics and Applications* **15**(3): 76-84.

[32] Y. WANG, J.-G. KIM and S.-F. CHANG (2003). "Content-Based Utility Function Prediction for Real-Time MPEG-4 Video Transcoding", *IEEE Conference on Image Processing*, Barcelona, Spain, Sep. 2003.

Yinpeng Chen is currently a Ph.D. student in the Department of Electrical Engineering at Arizona State University. His concentration is Arts, Media and Engineering. He received his M.S. and B.S. degree in Electrical Engineering from Tsinghua University, Beijing, China in 2003 and 2000 respectively. His research interests include resource adaptive computing framework, media adaptation in biofeedback system, pattern recognition and computer vision. Mr. Chen is a student member of IEEE. He received the best demo award in ACM Multimedia 2006.

Hari Sundaram is currently an assistant professor, at Arizona State University. This is a joint appointment with the department of Computer science and the Arts Media and Engineering program. He received his Ph.D. from the Department of Electrical Engineering at Columbia University in 2002. He received his MS degree in Electrical Engineering from SUNY Stony Brook 1995 and a B.Tech in Electrical Engineering from Indian Institute of Technology, Delhi in 1993.

His research group works on developing computational models and systems for situated communication. There are two complementary directions – (a) designing intelligent multimedia environments that exist as part of our physical world (b) developing new algorithms and systems to understand the media artifacts resulting from human activity. Prof. Sundaram's research has won several awards — including best paper awards from ACM and the IEEE and a best demo award at ACM multimedia 2006. He is an active participant in the Multimedia community — he is an associate editor for ACM Transactions on Multimedia Computing, Communications and Applications (TOMCCAP), as well as the IEEE Signal Processing magazine.