

The Polyharmonic Local Sine Transform: A New Tool for Local Image Analysis and Synthesis without Edge Effect^{*}

Naoki Saito * Jean-François Remy **

*Department of Mathematics
University of California
Davis, CA 95616 USA*

Abstract

We introduce a new local sine transform that can completely localize image information both in the space domain and in the spatial frequency domain. This transform, which we shall call the *polyharmonic local sine transform* (PHLST), first segments an image into local pieces using the characteristic functions, then decomposes each piece into two components: the *polyharmonic component* and the *residual*. The polyharmonic component is obtained by solving the elliptic boundary value problem associated with the so-called polyharmonic equation (e.g., Laplace's equation, biharmonic equation, etc.) given the boundary values (the pixel values along the boundary created by the characteristic function). Subsequently this component is subtracted from the original local piece to obtain the residual. Since the boundary values of the residual vanish, its Fourier sine series expansion has quickly decaying coefficients. Consequently, PHLST can distinguish *intrinsic* singularities in the data from the artificial discontinuities created by the local windowing. Combining this ability with the quickly decaying coefficients of the residuals, PHLST is also effective for image approximation, which we demonstrate using both synthetic and real images. In addition, we introduce the *polyharmonic local Fourier transform* (PHLFT) by replacing the Fourier sine series above by the complex Fourier series. With a slight sacrifice of the decay rate of the expansion coefficients, PHLFT allows one to compute local Fourier magnitudes and phases without revealing the edge effect (or Gibbs phenomenon), yet is invertible and useful for various filtering, analysis, and approximation purposes.

Key words: Polyharmonic equation, Laplace equation, local sine transform, local Fourier transform, image approximation

1 Introduction

For smooth periodic signals and images, the conventional Fourier series expansion can give us almost everything we need. Simply manipulating the Fourier coefficients, we can shift them, differentiate/integrate them, and filter/attenuate them at our disposal. We can also measure the smoothness class (the Lipschitz/Hölder exponents) of a signal by checking the rate of the decay of its Fourier coefficients, which is related to sparsity since the faster decay of the expansion coefficients leads to a sparser representation. See the standard references on Fourier series[1–3] for the details. It is very important to realize that we lose everything once we have to deal with *non-periodic* signals. The mismatch caused by the brute-force periodization (i.e., the head and the tail of an input signal do not match) kills every nice property we would have in the case of the smooth periodic signals. For example, the Fourier series expansion coefficients of non-periodic data decay slowly, i.e., $O(\|\mathbf{k}\|^{-1})$ where \mathbf{k} is the frequency index vector, and reveal the Gibbs phenomenon. Dealing with non-periodic signals in a proper manner is of paramount importance in many applications that require analysis of local signal features such as data compression and discrimination. First of all, real signals are often non-periodic. Second, even if the original signal is perfectly periodic, its locally segmented pieces (for local analysis of such pieces) are almost always not periodic. Thus, what we must do is not to create artificial discontinuities due to the local windowing or the brute-force periodization so that we can focus on the analysis of *intrinsic* discontinuities and singularities in the signal if any. Our goal is to develop a transform that can eliminate the interference of the boundary of a local window imposed by a user with the Fourier analysis of the information inside of such a window as much as possible.

We note that the wavelets/wavelet packets do not really solve this problem because in order to use wavelets or wavelet packets one needs to follow one of the following recipes: 1) periodize a signal and apply the standard wavelets/wavelet packets; or 2) design special wavelets near the boundaries (called “wavelets on the interval”) as Cohen, Daubechies, and Vial proposed in [4]; or 3) use the so-called “multiwavelets” developed by Alpert [5], which essentially use segmented orthogonal polynomials. Unfortunately, none of them is really satisfactory for the

* Dedicated to the memory of our fathers, Toshiki Saito and Christian Remy, both of them passed away in 2003. A preliminary version of the material was presented at the SPIE Conference on Wavelets: Applications in Signal and Image Processing X (San Diego, CA, August 2003), and the 2nd International Conference on Computational Harmonic Analysis (Nashville, TN, May 2004).

* Corresponding author.

**Present address: France Telecom, Batiment Equinoxe, 4 avenue du 8 mai 1945, 78284 Guyancourt Cedex FRANCE

Email addresses: saito@math.ucdavis.edu (Naoki Saito),
jeanfrancois.remy@francetelecom.com (Jean-François Remy).

applications we are interested in. The brute-force periodization (1) again creates the mismatch/discontinuity at the end points, and this generates large wavelet/wavelet packet coefficients. The wavelets on the interval (2) are rather complicated to implement in practice, particularly if one wants to use these in the wavelet packet mode or for images. Finally, the multiwavelets (3) are unfortunately not efficient for oscillatory signals such as textured images.

Coifman and Meyer [6] and Malvar [7,8] independently developed the so-called lapped orthogonal transform (LOT) (also known as local trigonometric transform (LTT) or Malvar wavelets) in an attempt to solve the above problems. However, it turned out that they were not completely satisfactory either, particularly for local signal analysis. The primary reason is the lack of the true localization capability of LTT: it cannot split an input signal into a set of non-overlapping (disjoint) regions without the influence of their adjacent regions due to its use of the folding operations. (We shall call this a “crosstalk” problem.) In this respect, the block discrete cosine transform (DCT) used in the JPEG Baseline standard is better because it chops a signal with the characteristic functions, i.e., no influence from the adjacent regions. Chopping the signal with the characteristic functions, however, makes the representation of the signal segment less sparse due to the sharp cutoff at the boundaries of the segment. If the signal segment does not contain any intrinsic discontinuity, the Fourier cosine coefficients decay with the rate $O(\|\mathbf{k}\|^{-2})$, which may not be fast enough, although this is better than the complex Fourier series expansion or Fourier sine series expansion, which provide only $O(\|\mathbf{k}\|^{-1})$ decay. In the case of LTT, we can tune the width of the overlap in the folding operations to get a faster decay rate, but there is another dilemma. The wider the overlap, the better in terms of sparsity, but the worse in terms of crosstalks (i.e., more statistically dependent). Fang and Séré developed the so-called “multiple folding” local trigonometric transforms (MLTT) [9] to overcome one of the problems. In terms of sparsity, MLTT provides a better representation than the conventional LTT. However, in terms of crosstalk problems, MLTT is definitely worse than LTT because MLTT has wider overlaps, especially in the early stage of the hierarchical split, and that creates more and more crosstalks down in the hierarchy. Villemoes proposed a further improvement in [10], which provides us a set of local cosines satisfying the uniform bounds on their time-frequency concentrations, which was not possible by LTT and MLTT. However, this method constrains the arrangement of the dyadic blocks: it only allows each block to have adjacent blocks of either the same size or the twice as large or small in each dimension. In other words, around each block, we cannot place its adjacent blocks whose sizes are very different from it. Thus, this method generates a set of blocks similar to the Whitney decomposition [11]. However, this constraint may reduce the efficiency in approximation and compression. In principle, there should not be such a constraint; we should be able to arrange the blocks with different sizes freely. This constraint is due to the specific requirement of the action regions and bell functions which are the essence of the LTT construction.

All these problems essentially boil down to how to treat the boundaries that define those subintervals and regions. To attack this fundamental problem, we developed a new version of the local Fourier transform called “continuous boundary local Fourier transform” (CBLFT) a few years ago [12,13]. CBLFT does not create any crosstalks among the subintervals and any discontinuities in the signal values at the boundaries of those subintervals. For simplicity, let us describe the one-dimensional version of CBLFT here. It first chops a signal into segments with the characteristic functions. Let $I = [0, 1]$ be one of such segments. Then, let us periodize the segmented signal on I , say, $f(x)$ with period 1. Let us assume that f is continuous on I . At this point, the head and tail of f on I does not match, so the periodized signal has two discontinuities at $x = 0$ and $x = 1$. Consider now this periodized function on the extended interval $[-1, 2]$. We add the constant $f(0) - f(1)$ to this periodized signal only over $[-1, 0]$, and similarly $f(1) - f(0)$ only over $(1, 2]$. This makes this extended signal continuous over $[-1, 2]$ although this is not periodic with period 1 anymore. CBLFT then applies the folding operations at $x = 0$ and $x = 1$ and follows the usual LFT recipe of Wickerhauser [14], [15, Chap. 4]. This procedure guarantees the decay rate of $O(|k|^{-2})$, generates no crosstalk since it only uses the signal information over I , and moreover, allows us to use the complex exponentials instead of real-valued cosine functions, which may be advantageous for certain applications such as computing locally analytic signals. However, CBLFT is not yet satisfactory: it does not deal with discontinuities in the derivatives at $x = 0$ and 1 so that the speed of the decay of the expansion coefficients is not fast enough. Moreover, due to its use of the folding operators, CBLFT is not too effective for analyzing very short segments of a signal.

We shall propose the *Polyharmonic Local Sine Transform* (PHLST) in the next section, which is a fundamentally better transform in the sense that it does not suffer from any of these problems compared to the other methods mentioned above. It can also be generalized easily and hierarchically to higher dimensions. In essence, PHLST first segments an image into local pieces using the characteristic functions, then decomposes each piece into two components: the *polyharmonic component* and the *residual*. The polyharmonic component is obtained by solving the elliptic boundary value problem associated with the so-called polyharmonic equation (e.g., Laplace’s equation, biharmonic equation, etc.) given the boundary values (the pixel values along the boundary created by the characteristic function) possibly with the estimates of normal derivatives at the boundary. Subsequently, this component is subtracted from the original local piece to obtain the residual. Because the boundary values of the residual (possibly with their normal derivatives) vanish, its Fourier sine series expansion has quickly decaying coefficients i.e., $O(\|\mathbf{k}\|^{-3})$ or faster. Using this transform, we can distinguish *intrinsic* singularities in the data from the artificial discontinuities created by the local windowing using the characteristic functions.

The organization of this paper is as follows. We shall describe the basic idea of PHLST in the next section. Section 3 describes important practical information, in

particular, implementation issues. Section 4 reports the results of our simple image approximation experiments using DCT, local cosine transform (LCT; one version of LTT), discrete wavelet transform (DWT), and PHLST. In Section 5, we introduce the polyharmonic local Fourier transform (PHLFT) by replacing the Fourier sine series expansion of v components by the ordinary Fourier series expansion using the complex exponentials. With a slight sacrifice in the decay rate of the expansion coefficients, we gain many nice properties of the complex Fourier series such as the ability to represent local image patches by its Fourier magnitudes and phases without suffering from the Gibbs phenomenon. Finally, we conclude this paper in Section 6 where we shall discuss the open issues and our future plans, as well as the relation of PHLST to the previous works done by other scientists.

2 Our Proposed Method: Polyharmonic Local Sine Transform

Instead of constructing an orthonormal basis, we first segment a signal $f(\mathbf{x})$, $\mathbf{x} \in \mathbb{R}^d$ into local pieces using the characteristic functions, then split each piece into two components: the polyharmonic component and the residual. Let $\Omega \in \mathbb{R}^d$ be the overall domain where the signal is supported, and let $\overline{\Omega} = \cup_{j=1}^J \overline{\Omega}_j$ be a decomposition of the domain Ω into a set of disjoint subdomains $\{\Omega_j\}$. Note that Ω_j 's are open and disjoint whereas $\overline{\Omega}_j$'s are closed and may share the boundaries. Let f_j be the restriction of f to $\overline{\Omega}_j$, i.e., $f_j = \chi_{\overline{\Omega}_j} f$. Let us now consider the local decomposition $f_j = u_j + v_j$ on Ω_j . We shall call u_j the *polyharmonic component* of f_j and v_j the *residual*. Let Δ be the Laplace operator in \mathbb{R}^d . Then the polyharmonic component is obtained by solving the following polyharmonic equation:

$$\Delta^m u_j = 0 \quad \text{in } \Omega_j, \quad m = 1, 2, \dots \quad (1)$$

with given boundary values and normal derivatives

$$\frac{\partial^{q_\ell} u_j}{\partial \nu^{q_\ell}} = \frac{\partial^{q_\ell} f}{\partial \nu^{q_\ell}} \quad \text{on } \partial\Omega_j, \quad \ell = 0, \dots, m-1, \quad (2)$$

where q_ℓ is the order of the normal derivatives yet to be specified. The parameter q_0 is normally set to 0, which means that $u_j = f$ on the boundary $\partial\Omega_j$. These boundary conditions enforce the function values and the normal derivatives of orders q_1, \dots, q_{m-1} of the solution u_j along the boundary $\partial\Omega_j$ to match those of the original signal f over there. If the Ω_j 's are all rectangles (of possibly different sizes), then we take $q_\ell = 2\ell$, i.e., the *even order* normal derivatives. It is not necessary to match the odd order normal derivatives for the rectangular domain case because the Fourier sine series of v_j is equivalent to the Fourier series expansion of the extension of v_j by odd reflection with respect to the boundary $\partial\Omega_j$ and *the continuity of the odd order normal derivatives (up to order $2m-1$) is automatically guaranteed*.

For $m = 1$, Equation (1) becomes the following familiar form:

$$\begin{cases} \Delta u_j = 0 & \text{in } \Omega_j, \\ u_j = f & \text{on } \partial\Omega_j, \end{cases} \quad (3)$$

which is the *Laplace equation with the Dirichlet boundary condition*. For $m = 2$, we have the following *biharmonic equation with the mixed boundary condition*:

$$\begin{cases} \Delta^2 u_j = 0 & \text{in } \Omega_j, \\ u_j = f, \quad \frac{\partial^\ell u_j}{\partial \nu^\ell} = \frac{\partial^\ell f}{\partial \nu^\ell} & \text{on } \partial\Omega_j, \end{cases} \quad (4)$$

where $\ell = 1$ or 2 depending on the shape of Ω_j . For the rectangular-shaped Ω_j , we take $\ell = 2$ as explained above. Note that in 1D ($d = 1$), the solution of (3) is simply a straight line connecting two boundary points of an interval $\overline{\Omega}_j$ whereas that of (4) is a cubic polynomial. However, in higher dimensions ($d \geq 2$), the solutions of (3), (4), and more generally (1) with (2) are not a tensor product of algebraic polynomials in general.

Subtracting such u_j from f_j gives us the residual $v_j = f_j - u_j$ satisfying:

$$\frac{\partial^{q_\ell} v_j}{\partial \nu^{q_\ell}} = 0 \quad \text{on } \partial\Omega_j, \quad \ell = 0, \dots, m-1. \quad (5)$$

Since the values and the normal derivatives of v_j on $\partial\Omega_j$ vanish, its Fourier sine expansion coefficients decay rapidly, i.e., $O(\|\mathbf{k}\|^{-2m-1})$, if there is no other intrinsic singularity in Ω_j . In fact, we have the following theorem.

Theorem 1 *Let Ω_j be a bounded rectangular domain in \mathbb{R}^d , and let $f_j \in C^{2m}(\overline{\Omega}_j)$, but non-periodic. Assume further that $(\partial/\partial x_i)^{2m+1} f$, $i = 1, \dots, d$, exist and are of bounded variation. Furthermore, let $f_j = u_j + v_j$ be the PHLST representation, i.e., the polyharmonic component u_j is the solution of the polyharmonic equation (1) of order m with the boundary condition (2) with $q_\ell = 2\ell$, $\ell = 0, 1, \dots, m-1$, and $v_j = f_j - u_j$ is the residual component. Then, the Fourier sine coefficient $b_{\mathbf{k}}$ of the residual v_j is of $O(\|\mathbf{k}\|^{-2m-1})$ for all $\mathbf{k} \neq 0$, where $\mathbf{k} = (k_1, \dots, k_d) \in \mathbb{Z}_+^d$, and $\|\mathbf{k}\|$ is the usual Euclidean (i.e., ℓ^2) norm of \mathbf{k} .*

The proof of this theorem is in Appendix A. Note that if we employ the straightforward Fourier series expansion of non-periodic f_j by brute-force periodization, the decay rate becomes only $O(\|\mathbf{k}\|^{-1})$ even if $f_j \in C^{2m}(\overline{\Omega}_j)$. If we use the Fourier cosine series expansion of f_j , we can get $O(\|\mathbf{k}\|^{-2})$.

We call this new way of decomposing a function f into a set of functions $\{f_j = u_j + v_j\}_{j=1}^J$ the *Polyharmonic Local Sine Transform* (PHLST) with polyharmonicity m . For $m = 1$ and 2 , we use the special abbreviations, LLST (Laplace LST) and BLST (Biharmonic LST).

Remark 2 The polyharmonic component u_j is smooth particularly in the middle of the region Ω_j and as \mathbf{x} approaches the boundary $\partial\Omega_j$, $u_j(\mathbf{x})$ approaches $f(\mathbf{x})$. In fact, u_j is also the minimizer of the following optimization problem (see e.g., [16]):

$$\min_{u \in H^m(\Omega_j)} \sum_{|\alpha|=m} \binom{m}{\alpha} \int_{\Omega_j} |D^\alpha u|^2 d\mathbf{x} \quad \text{subject to the boundary condition (2),} \quad (6)$$

where $H^m(\Omega_j)$ is an L^2 -Sobolev space of order m over Ω_j , $\alpha = (\alpha_1, \dots, \alpha_d)$ is a multi-index of a d -dimensional variable with the following definitions: $|\alpha| = \alpha_1 + \dots + \alpha_d$, $\binom{m}{\alpha} = m! / (\alpha_1! \cdots \alpha_d!)$, and $D^\alpha = \partial_1^{\alpha_1} \cdots \partial_d^{\alpha_d}$, where $\partial_i = \partial / \partial x_i$. This means that the polyharmonic component minimizes the roughness measured by the objective function in (6) while satisfying the boundary condition (2). The larger the polyharmonicity m is, the smoother the solution u_j gets.

Remark 3 Because the u_j component becomes very smooth away from the boundary and only carries the low frequency information in the middle of the region as discussed in Remark 2, the frequency content of the v_j component is close to that of f_j , particularly, around the mid to higher frequency band, where textures are well represented. Therefore, the v_j component retains the original texture information almost intact and is more suitable for texture analysis than the original image patch f_j .

Remark 4 Our PHLST belongs to a larger class of models, to which Yves Meyer refers as the $u+v$ models [17]. In a $u+v$ model, the original data f is assumed to be a sum of the two components u and v . The first component u is aimed at modeling the objects present in the data (often they are assumed to be in some specific function spaces such as the Besov spaces or the functions of bounded variation BV) whereas the v term is responsible for texture and noise in the data. In our PHLST model, one can also interpret u as “trend” and v as “fluctuation” of the original data f , which are commonly-used terms in the wavelet literature [18]. Yet another interpretation of this decomposition is the following: the polyharmonic component u represents the part of f that is predictable from the boundary data only, whereas the residual v represents the unpredictable part of f .

3 Practical Aspects of PHLST

If Ω and Ω_j 's are rectangular regions in \mathbb{R}^d and the functions defined over there are sampled on a regular lattice (which is the most efficient case in terms of numerical implementation), we can employ the Discrete Sine Transform (DST) based on the FFT algorithm to rapidly compute both the polyharmonic components and the Fourier sine series expansion of the residual components. In particular, we use the algorithm proposed by Averbuch, Braverman, Israeli, and Vozovoi [19,20], which

seems to us the most natural and practical Laplace/Poisson equation solver on rectangles. This ABIV method gives us more accurate solutions (in fact an arbitrary order of accuracy in principle) than those based on the finite difference (FD) approximation of the Laplace operator followed by FFT, which only gives us the solutions with accuracy $O(h^2)$ or $O(h^4)$, depending on the 5-point FD scheme or the modified 9-point FD scheme is used [21, Sec. 7.1], [22, Chap. 12] (h is, of course, the grid spacing here). Yet, the computational cost of the ABIV algorithm is still $O(N^d \log N)$ where N is the number of grid points in each direction of a rectangular domain. We also note that we use the $(d - 1)$ -dimensional version of PHLST to represent the boundary function $f(x)$ for $x \in \partial\Omega_j$, which can be recursively computed starting from the 1D version. To describe the ABIV method more simply and precisely, let us now consider the LLST decomposition $f(x, y) = u(x, y) + v(x, y)$ on the unit square $\overline{\Omega} = [0, 1]^2 \subset \mathbb{R}^2$. There are several options in the ABIV method, but the simplest and most practical version to compute the solution to (3) that does not require any derivative estimate can be written as follows.

$$u(x, y) = p(x, y) + \sum_{k \geq 1} \left\{ b_k^{(1)} h_k(x, 1 - y) + b_k^{(2)} h_k(y, 1 - x) + b_k^{(3)} h_k(x, y) + b_k^{(4)} h_k(y, x) \right\}, \quad (7)$$

where $p(x, y)$ is a harmonic polynomial that agrees with $f(x, y)$ at the four corner points of the domain, and its simplest form is:

$$p(x, y) = a_3 xy + a_2 x + a_1 y + a_0, \quad (8)$$

where the parameters a_j 's can be easily and uniquely determined using the function values at the four corners. The function $h_k(x, y)$ is defined as:

$$h_k(x, y) \triangleq \sin(\pi kx) \frac{\sinh(\pi ky)}{\sinh(\pi k)}, \quad (9)$$

and $b_k^{(j)}$, $j = 1, 2, 3, 4$, are the k th 1D Fourier sine coefficients of the boundary functions $f(x, 0) - p(x, 0)$, $f(0, y) - p(0, y)$, $f(x, 1) - p(x, 1)$, and $f(1, y) - p(1, y)$, respectively, where $0 \leq x \leq 1$, $0 \leq y \leq 1$. One can easily verify that each sum involving h_k functions in (7) is a solution of the Laplace equation with the nonzero boundary values on only one side of the square. For example, $u^{(1)}(x, y) = \sum_{k \geq 1} b_k^{(1)} h_k(x, 1 - y)$ satisfies the boundary condition:

$$u^{(1)}(x, 0) = f(x, 0) - p(x, 0), \quad u^{(1)}(x, 1) = 0, \quad u^{(1)}(0, y) = 0, \quad u^{(1)}(1, y) = 0.$$

On the other hand, the v component can be written as

$$v(x, y) = \sum_{m \geq 1} \sum_{n \geq 1} \beta_{mn} \sin(m\pi x) \sin(n\pi y) \quad (10)$$

where β_{mn} is the 2D Fourier sine coefficients of $v(x, y) = f(x, y) - u(x, y)$. We also note that we use the gridpoint formulation rather than the midpoint formulation for our discretization. Therefore, we use DST-I rather than the DST-II/III combination

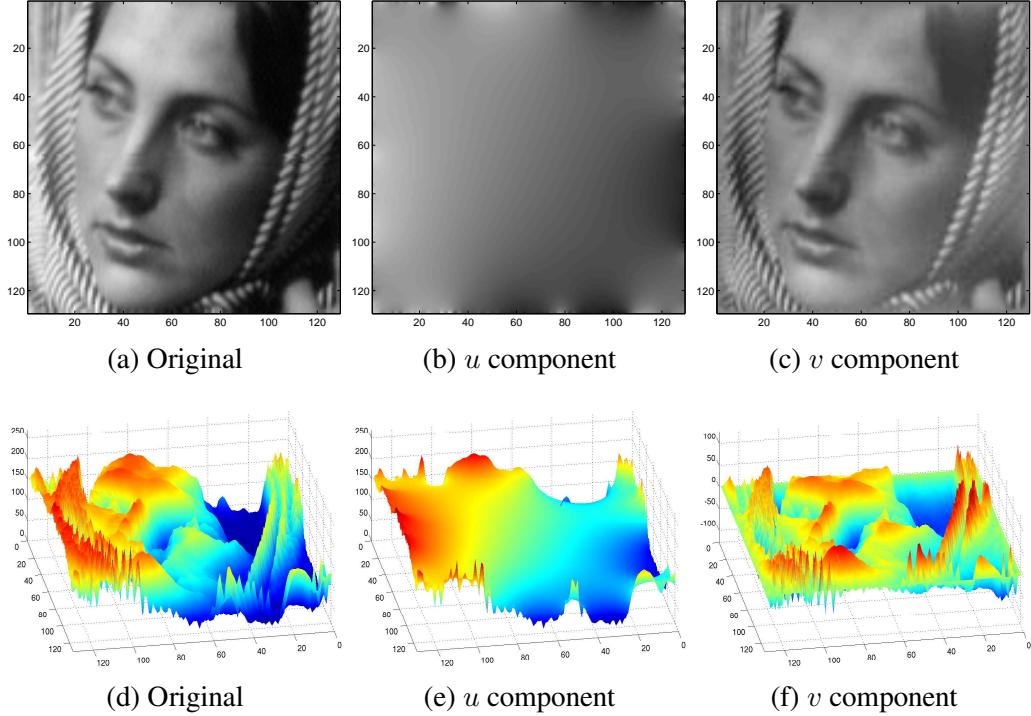


Fig. 1. The global version of LLST applied to the face part (129×129 pixels) of the Barbara image. Note that the display dynamic range of (a) and (b) are set to $[0, 255]$ whereas that of (c) is set to $[-127, 127]$. Figures (d)–(f) are the 3D perspective plots of the top row. It is easier to see in (f) that the overall shading of the face is removed in the v component.

in this paper. See e.g., [15, Chap. 4], [23] for the details of gridpoint vs midpoint formulations.

Let us show a simple example of the effect of removing the polyharmonic component. Figure 1 demonstrates this. We first use the global version of LLST, i.e., no further split of the image. One may wonder what is the difference between the original and the residual v component in this case because they look very similar. We would like to point out that the v component is displayed in a shifted dynamic range (i.e., $[-127, 127]$ instead of $[0, 255]$) to see the details. Moreover, if we examine them carefully, we can observe that the overall shading is gone in the v component. The bottom row of Figure 1 clarifies this by displaying them as 3D perspective plots.

Figure 2 shows the results of LLST decomposition by splitting the original into a set of homogeneous square regions each of which has 17×17 pixels.

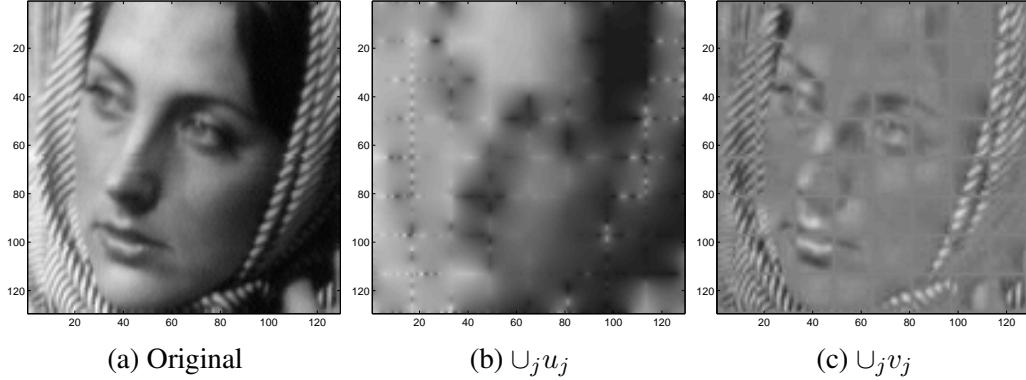


Fig. 2. The hierarchical LLST decomposition of the Barbara face image: (a) Original image; (b) the union of the u_j components $\cup_j u_j$ where each Ω_j is a square of 17×17 pixels; (c) the union of the v_j components $\cup_j v_j$.

4 Image Approximation using PHLST

Since PHLST can represent smooth regions of an image very efficiently (i.e., faster decay of the DST coefficients of the residual components), image approximation and compression are immediate and important applications of PHLST. In the field of image approximation, a great theoretical progress and understanding has been made. Nonlinear approximation of functions belonging to various function spaces, in particular, using the orthonormal wavelet bases or splines are now well understood [24]. However, as we shall discuss in Section 6.1.3, obtaining precise nonlinear approximation results for PHLST remains elusive. Therefore, in this paper, we shall report the results of our numerical experiments using PHLST and demonstrate its superiority over the other representative transforms such as DCT, LCT, and wavelets, which will hopefully illuminate our theoretical investigation.

In the rest of the paper, the dimension is set to $d = 2$, and the domain Ω and its subdomains Ω_j are all rectangles. We shall also limit our investigation to the specific version of PHLST with $m = 1$, i.e., LLST, rather than the PHLST with higher polyharmonicities. We shall report the other settings (e.g., on more general domains, PHLST with $m > 1$, and investigation on PHLST-based image compression with quantization) at a later date. In our experiments, we shall examine how the intrinsic discontinuities and the smoothness and oscillation of the data away from the discontinuities affect the approximation performance.

Let us first describe the details of the LLST representation of a given image for a given split of the domain $\overline{\Omega} = \cup_j \overline{\Omega}_j$. In each subdomain Ω_j , we have an image patch f_j , which is decomposed into u_j and v_j components. Recall that each harmonic component u_j can be completely represented by the boundary values $f_j|_{\partial\Omega_j}$ that in turn can be represented by 1D version of LLST. As (7) shows, the u_j can be further decomposed into the harmonic polynomial p_j (8) and the rest using the h_k functions (9). The p_j component can be completely represented by (or recovered from) the

four corner values of $\bar{\Omega}_j$. The rest of the u_j component is completely determined by 1D DST coefficients of the four boundary functions. Therefore, *we represent the u_j component as the four corner values and the four sets of the 1D DST coefficients of the boundary values.* We note that these four corner points and four sets of the 1D DST coefficients are shared by the adjacent blocks, so we should not duplicate them for storage efficiency. For example, if we split an image of size $(2^M + 1) \times (2^N + 1)$ pixels into a set of homogeneous blocks of size $(2^m + 1) \times (2^n + 1)$ pixels, then the total number of the corner points without duplication is $(2^{M-m} + 1) \times (2^{N-n} + 1)$. Another thing to note here is about the values of these corner points. The range of the corner pixel values are between 0 and 255 for a typical 8 bit gray scale image, which is quite different from that of the DST coefficients. This difference should be taken into account when the image compression including quantization is considered. Once the u_j component is taken care of, the v_j component is simply obtained by 2D DST (more precisely, 2D DST-I) in a straightforward manner.

As for the performance comparison, we compare LLST with: 1) the block DCT-II (abbreviated as “BDCT” below), which is used in the JPEG Baseline standard [23]; 2) LCT, which uses DCT-IV after the folding operations [15, Chap. 4]; and 3) C6 discrete wavelet transform (C6DWT), which is based on the ‘Coiflet’ 6-tap conjugate quadrature filter [25], [26, Chap. 8]. Note that the C6 wavelet functions has two vanishing moments, which means that any locally linear part of an input image generates negligible coefficients. Thus, we believe that it is a fair comparison with LLST since the u_j components of LLST also take care of locally linear parts.

Note also that if the size of an original image is $(2^M + 1) \times (2^N + 1)$, which is suitable for LLST, then we remove its last column and row to make it suitable for BDCT, LCT, and C6DWT. On the other hand, if an original image is of size $2^M \times 2^N$ pixels, then we duplicate the last column and row to make it suitable for LLST. Note also that the block size $(2^m + 1) \times (2^n + 1)$ pixels used for LLST correspond to $2^m \times 2^n$ pixels for BDCT and LCT.

Now let us describe our approximation strategy. We first retain all the corner pixel values for LLST, all the “DC” components for BDCT and LCT, and all the coefficients of the coarsest scaling function (i.e., the lowest frequency band) for C6DWT. Starting from those number of retained coefficients, we do the “nonlinear approximation,” i.e., we select the coefficient with the largest energy among all the coefficients not yet used, add it to the retained set of coefficients, reconstruct an approximation from this set, and evaluate its quality of approximation. We repeat this process until we use up all the coefficients. The quality of approximation in this paper is measured by *PSNR* (or *peak signal-to-noise ratio*), which is normally considered as a better metric for evaluating image quality than *SNR* (signal-to-noise ratio) and defined as

$$20 \log_{10} \left(\max_{\mathbf{x} \in \Omega} |f(\mathbf{x})| / RMSE \right)$$

where $RMSE$ (the root mean square error) is the absolute ℓ^2 error between the original and the approximation divided by the square root of the number of pixels. The unit of $PSNR$ is decibel (dB). Thus, $PSNR = 0$ dB corresponds to $RMSE = \max |f(\mathbf{x})|$, and $PSNR = 350$ dB corresponds to $RMSE \approx 10^{-17.5} \times \max |f(\mathbf{x})|$, which is essentially the machine precision, i.e., no error in practice. In our experiments, $PSNR$ is computed as a function of the ratio R of the number of the retained coefficients to the total number of pixels of the input image.

4.1 Approximation of Synthetic Smooth Images with and without Discontinuity

In order to demonstrate the power of LLST, we shall conduct simple approximation experiments using synthetic data. We shall first examine the effect of intrinsic discontinuity in otherwise smooth data to the transforms under consideration.

4.1.1 A case without discontinuity

We first synthesize the 2D Gaussian function $\exp(-(x + 1/3)^2 - (y + 1/3)^2)$ over the region $\overline{\Omega} = [-1, 1] \times [-1, 1]$, which does not have any intrinsic discontinuity. We sample this function with rate $\Delta x = \Delta y = 1/128$, yielding discrete data of 257×257 pixels. If $\Omega = \mathbb{R}^2$, then this Gaussian function is in $C^\infty(\Omega)$. However, once we restrict this on a finite rectangle such as $[-1, 1] \times [-1, 1]$, its periodic extension is not in $C^\infty(\mathbb{R}^2)$ anymore, in fact, not even in $C(\mathbb{R}^2)$! This is due to the domain boundary we discussed in Introduction and by all means we need to remove this boundary effect in order to efficiently represent and characterize this dataset. On the other hand, we really do not need to split Ω into a set of smaller segments in this case since there is no intrinsic discontinuities in the data and it is homogeneous and isotropic. In order to confirm this, we also segment Ω into a set of homogeneous blocks of $2^j \times 2^j$ pixels for BDCT and LCT and $(2^j + 1) \times (2^j + 1)$ pixels for LLST where $j = 7, 6, 5$, and check their approximation performance. We can view them as the quadtree-structured splits with the original image as the root node and those homogeneous blocks as the leaf nodes. Then, the maximum depth (or level) J of the quadtree is $J = 1, 2, 3$, respectively. Note that $J = 0$ implies no split of the domain Ω . For C6DWT, to match the number of the coefficients in the lowest frequency band with that of the DC components in the BDCT/LCT representations, the maximum depth of decomposition needs to be $J = 7, 6, 5$, respectively. In general, for an image of $2^N \times 2^N$ pixels, the maximum depth J in the BDCT/LCT tree corresponds to the maximum depth $N - J$ in the wavelet tree.

Figure 3 shows these approximation results. As we have expected, LLST with no segmentation ($J = 0$) outperforms all the other transforms with various depths of decomposition except that it is defeated by C6DWT if the ratio of the retained coefficients R is higher than about 72%. However, such a high ratio range is of no

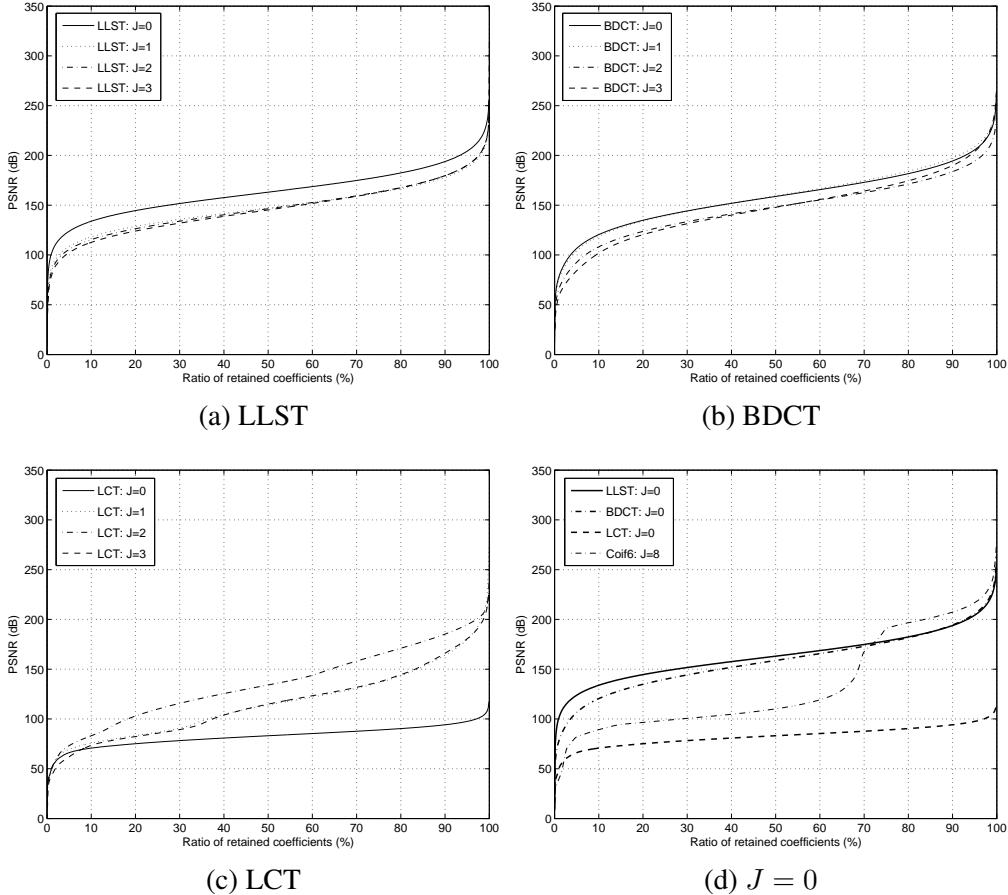


Fig. 3. Quality of approximation of the 2D Gaussian data measured by PSNR values. The index J represents the depth of the quadtree decomposition. Note that $J = 0$ means no splitting (i.e., the original domain is used).

interest for the purpose of efficient approximation. We also observe that the more we subdivide the original domain, the more the performance of LLST and that of BDCT degrade. This superior performance of LLST with $J = 0$ over BDCT with $J = 0$ demonstrates the importance of having the coefficients decaying with $O(\|\mathbf{k}\|^{-3})$ rather than $O(\|\mathbf{k}\|^{-2})$.

The behavior of LCT is markedly different from that of LLST and BDCT. This is due to the folding operations and the use of DCT-IV as the underlying base transform in LCT [15, Chap. 4]. In essence, the folding operations split the data on the original domain into blocks smoothly using the appropriate bell function so that the data on each block become suitable for DCT-IV. In fact, the folding operations make the data locally even at the upper and left boundary and locally odd at the lower and right boundary of each block since each DCT-IV basis function behaves like cosines at the upper and left boundary and like sines at the lower and right boundary. These folding operations, however, need the data outside of the blocks. This means that we have a trouble at the blocks that share the boundary of the original domain because we do not know the data outside of the original domain.

In particular, we cannot apply the folding operation with $J = 0$, unless we can estimate a smooth extension of the data from the original domain. Note also that extending the input data by the brute-force periodization and then performing the folding do not solve the problem at all: such an operation does not match the polarity of the data at the domain boundary to what DCT-IV prefers. For simplicity, we use the so-called “free” boundary condition in our experiments. We do not assume any data outside of the domain, and do not perform any folding at the boundaries of the original domain. Therefore, LCT with $J = 0$ is equivalent to applying DCT-IV to the original image. Consequently, it senses the discontinuity at the lower and right boundaries unlike the simple DCT-II that views the data outside as the even-reflected version of the data inside. Thus, the LCT coefficients for $J = 0$ only decay as $O(\|\mathbf{k}\|^{-1})$, and that is the reason why LCT with $J = 0$ performs poorly. This is clearly worse than simply using DCT-II, and that is the main reason why DCT-IV is not used in the JPEG Baseline standard. On the other hand, if we use the smaller blocks such as the $J = 3$ case, then the influence of the boundary effect of the original domain is localized to the out-most blocks sharing the boundary of the original domain. However, the width of the bell function gets too short and the decay rate of the LCT coefficients becomes slow. That is the reason why LCT with $J = 2$ performs better than that with $J = 3$. See also [12,13] for the details of the boundary effects due to the folding operations.

As for the performance of C6DWT, we only plot $J = 8$ case in Figure 3 (d) because the PSNR curves for the other values $J = 7, 6, 5$ are almost identical to that for $J = 8$ with very small differences in the low ratio region. This is due to the fact that the higher frequency finer scale coefficients are identical for these values of J . C6DWT performs reasonable for this dataset, but it is beaten by LLST and BDCT except at the very high ratios such as the region of $R \geq 72\%$ as mentioned above. Note also that this implementation of C6DWT assumes the periodic boundary condition at the domain boundary. Therefore, it too views the domain boundary as discontinuity in general.

Remark 5 *The above observations on LCT and C6DWT suggest that removing the u component for $J = 0$ from the original data would help these transforms. In fact, if we remove the boundary by subtracting the u component, the performance of both LCT and C6DWT improves. In particular, for wavelets, this can be an alternative to “wavelets on the intervals” [4] provided that the pixels along the domain boundary (or their transformed version) are retained.*

4.1.2 A case with discontinuity

Let us now introduce simple discontinuity in this dataset, and examine how the approximation performance changes by that. We multiply a scalar factor 2 to the circular region satisfying $\{(x, y) | (x + 1/3)^2 + (y + 1/3)^2 < 0.2\} \subset \Omega$.

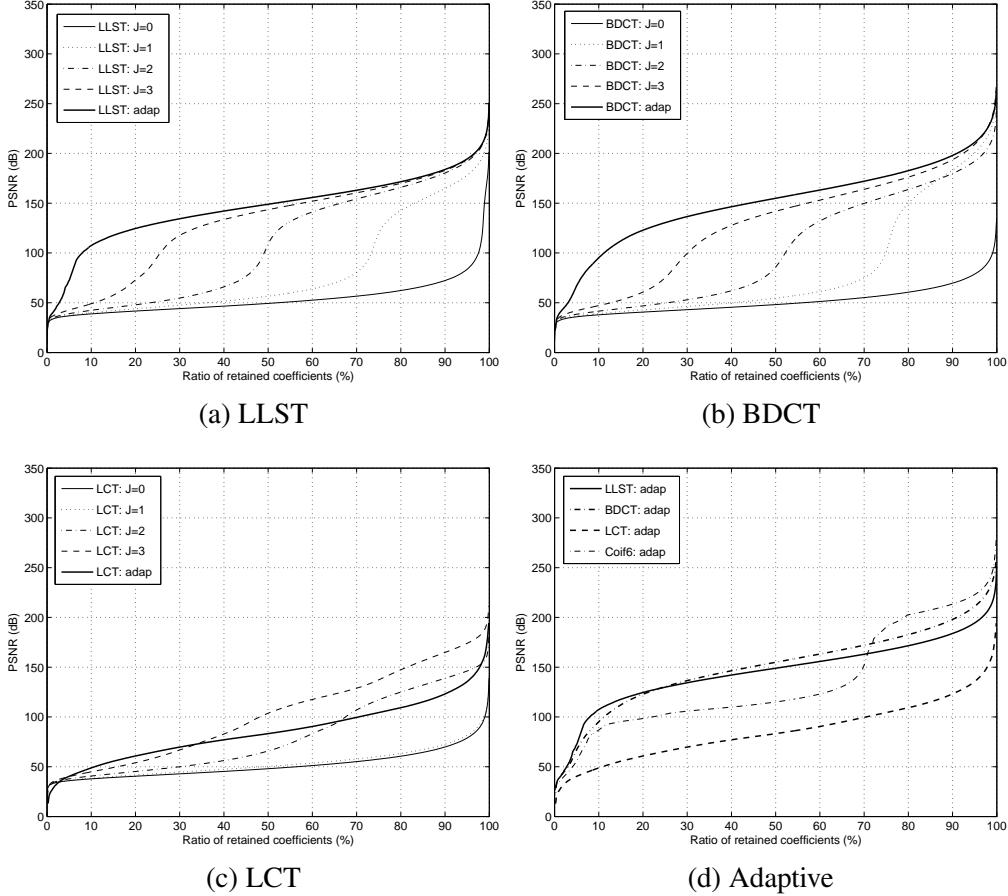


Fig. 4. Quality of approximation of the Gaussian data with circular discontinuity measured by PSNR values.

Figure 4 shows these approximation results. Due to the discontinuity, the global methods ($J = 0$) are now inferior to the segmented cases for LLST, BDCT, and LCT. In fact, from Figure 4 (a) and (b), we can see that the performance steadily improves as J increases. This is obvious since the discontinuity affects all the coefficients for $J = 0$ and makes their decay as $O(\|\mathbf{k}\|^{-1})$. On the other hand, for $J = 3$, the effect of the discontinuity is localized to the blocks where the discontinuity passes through. Note that in the continuum, whether covering the discontinuity by a large block or by a set of small or narrow blocks does not matter since the discontinuity results in the countable number of slowly decaying coefficients anyway. However, we are dealing with the finite dimensional sampled data. Consequently, localizing the discontinuity with a set of small size blocks reduces the number of slowly decaying coefficients.

This observation leads us to the following expectation: grabbing smooth regions by largest possible blocks and using narrow blocks around the singularity should give us more efficient approximation than splitting the domain into a set of homogeneous blocks. There is no need to divide a smooth region into a set of smaller blocks, and in fact, that is wasteful due to the storage of the corner and bound-

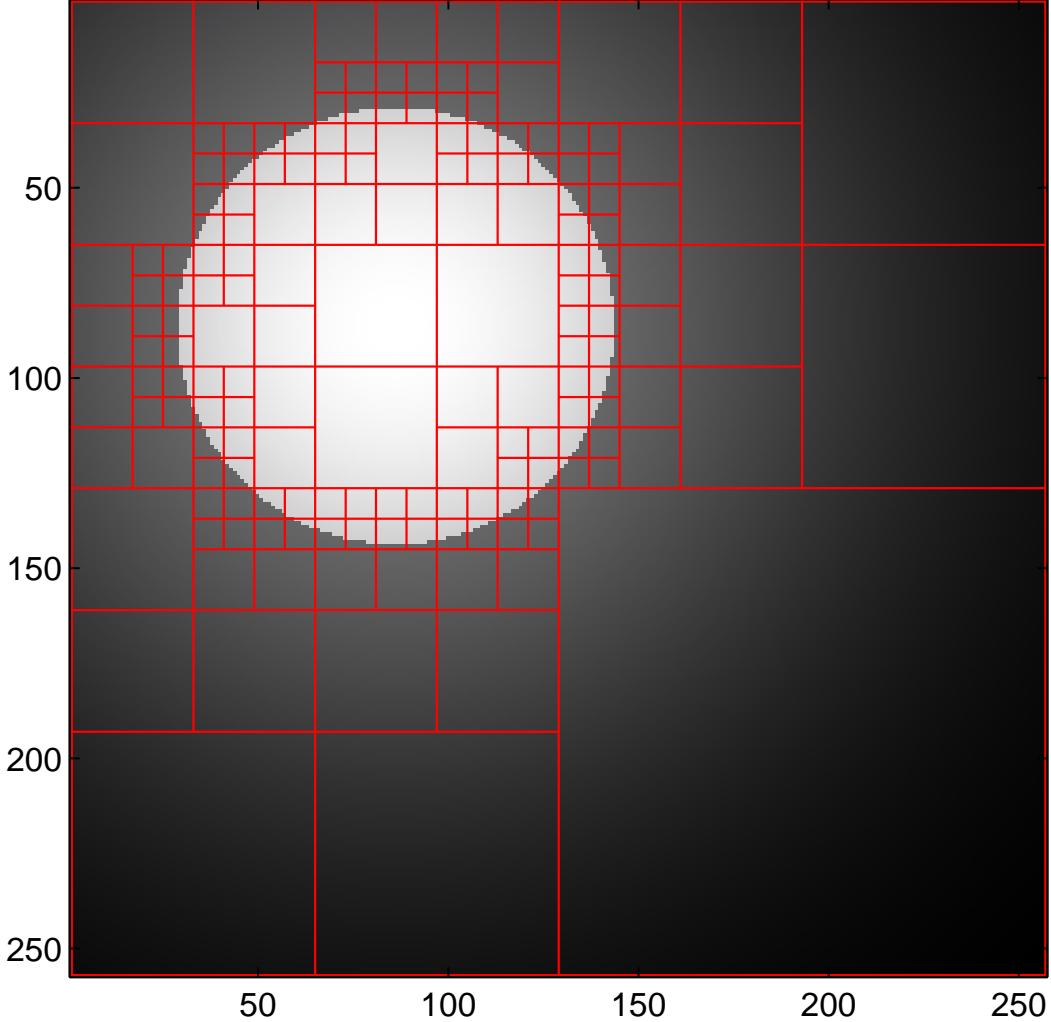


Fig. 5. Adaptive dyadic segmentation of the 2D Gaussian data with circular discontinuity, which honors the geometry of the discontinuity.

ary information in each block as demonstrated earlier. Therefore, we examine the power of the adaptive split of the domain honoring the geometry of (or the singularity in) the data. Figure 5 shows such an adaptive segmentation: smaller blocks around the singularity and progressively larger blocks away from the singularity, which is somewhat similar to the Whitney decomposition [11]. This particular segmentation has been done manually using our Java-based “Interactive Picture Partition” (IPP) program developed by Gary Zhong. We are currently developing an automatic adaptive segmentation algorithm using PHLST, and will report the details at a later date. Figure 4 also shows the performance of the approximations based on this adaptive segmentation. Of course, we use the same segmentation for LLST, BDCT, and LCT. As for adaptive version of C6DWT, we compute the so-called *best sparsifying basis* (BSB) via ℓ^1 norm minimization [27,28] from the C6 wavelet packet tree, and approximate by sorting out the BSB coefficients. However, our numerical computation shows that the BSB for this dataset turns out to be the standard wavelet decomposition. This confirms the well-known fact that the

wavelet transform performs well on piecewise smooth data. As we can easily see, the adaptive segmentation improves the performance of LLST and BDCT. Without adaptive segmentation, C6DWT performed best for $R \leq 25\%$ and $R \geq 72\%$, which can be seen by comparing the curve of LLST with $J = 3$ in subplot (a) with that of C6BSB ('Coif6: adap') in subplot (d) in Figure 4.

With adaptive segmentation, however, LLST outperformed both BDCT and C6BSB for $R \leq 25\%$, which is the range of our interest.

Interestingly enough, the adaptive segmentation does not improve the performance of LCT so much from that of nonadaptive homogeneous split; it is slightly better only in the range $R \in [0, 30]$. The reason is that the adaptive segmentation uses many small blocks of size 8×8 pixels, as shown in Figure 5. The version of LCT in our experiments uses the so-called "fixed" folding whose bell function width is completely determined by the smallest block size [15, Chap. 4]. Since the same bell function is also used for processing larger blocks, LCT loses its effectiveness as we discussed in Introduction and Section 4.1.1. To avoid such a problem, one needs to use multiple folding LCT of Fang and Séré [9], or better yet, the time-frequency LCT of Villemoes [10], which unfortunately constrains possible segmentation patterns as we discussed in Introduction. We shall not pursue these alternative versions of LCT here.

4.2 Approximation of Synthetic Oscillatory Images with and without Discontinuity

We shall now examine the performance of these transforms on an oscillatory function, in fact, a Gabor (i.e., a modulated Gaussian) function, with and without discontinuity as in the simple Gaussian case.

4.2.1 A case without discontinuity

We generate the data according to the formula: $f(x, y) = \exp(-(x + 1/3)^2 - (y + 1/3)^2) \cdot \sin(5\pi(2x + 3y))$ where $(x, y) \in [0, 1]^2$, and sample this with the same rate as before.

Figure 6 shows the approximation results. Again, in principle, there should not be any gain by splitting the data into a set of smaller blocks because the data do not have any intrinsic singularity. In fact, both LLST and BDCT, the $J = 0$ case outperforms the other values of J , and LLST with $J = 0$ remains as the best performer. Most obvious difference from the simple Gaussian case is the degradation of the performance of C6DWT, which confirms the well-known fact that the wavelet is not efficient for handling oscillatory data and textures. Another interesting observation is that LCT with $J = 2, 3$ performs quite well for this dataset. In fact, LCT with

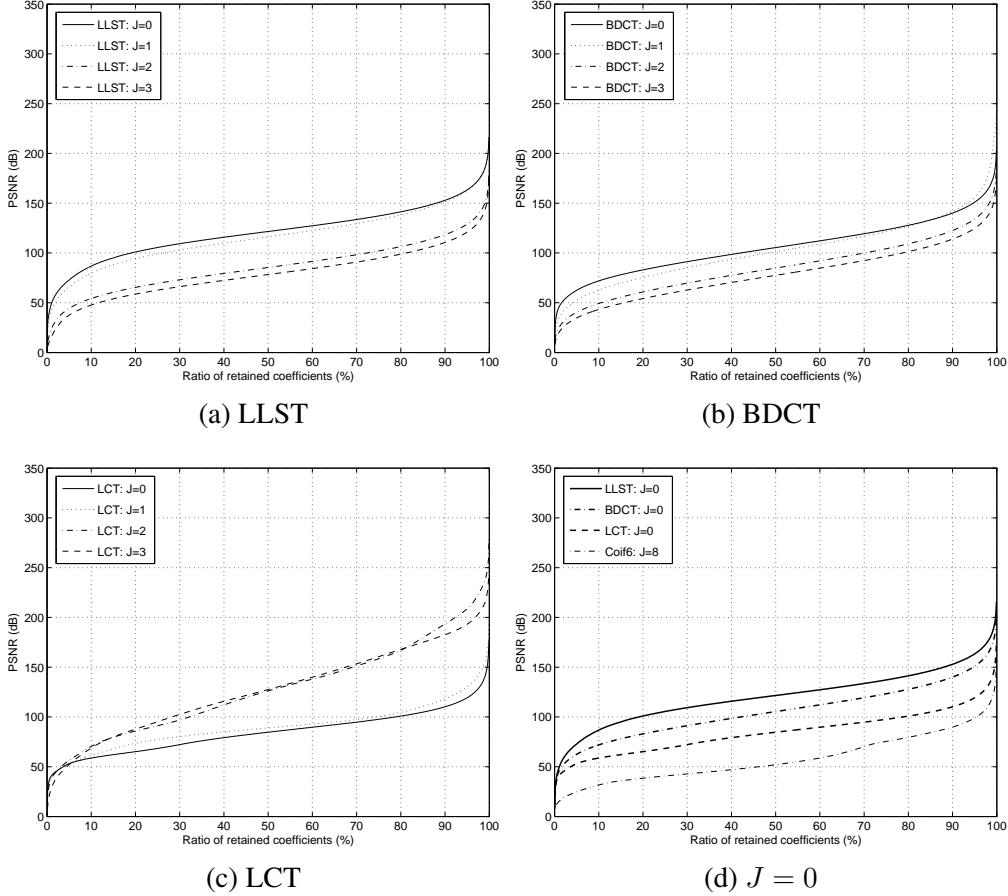


Fig. 6. Quality of approximation of the 2D Gabor (i.e., modulated Gaussian) data measured by PSNR values.

$J = 2, 3$ defeats LLST with $J = 0$ for the range $R > 42\%$; compare Figure 6 (a) and (c). However, the range of our interest for efficient approximation, LLST with $J = 0$ still performs the best.

4.2.2 A case with discontinuity

Finally, we shall examine the toughest scenario so far: the Gabor function with the same circular discontinuity as the one used in Section 4.1.2. We shall not only use the homogeneous splits, but also the same segmentation pattern shown in Figure 5 for the adaptive segmentation. We shall also compute the BSB from the C6 wavelet packet tree.

Figure 7 shows the approximation results. The performance of the approximation are degraded for all the transforms compared to the other scenarios considered so far, as expected. However, the overall best performer is LLST with adaptive segmentation, except for the range $R < 6\%$ where BDCT with adaptive segmentation performs slightly better. LCT with adaptive segmentation does not perform better

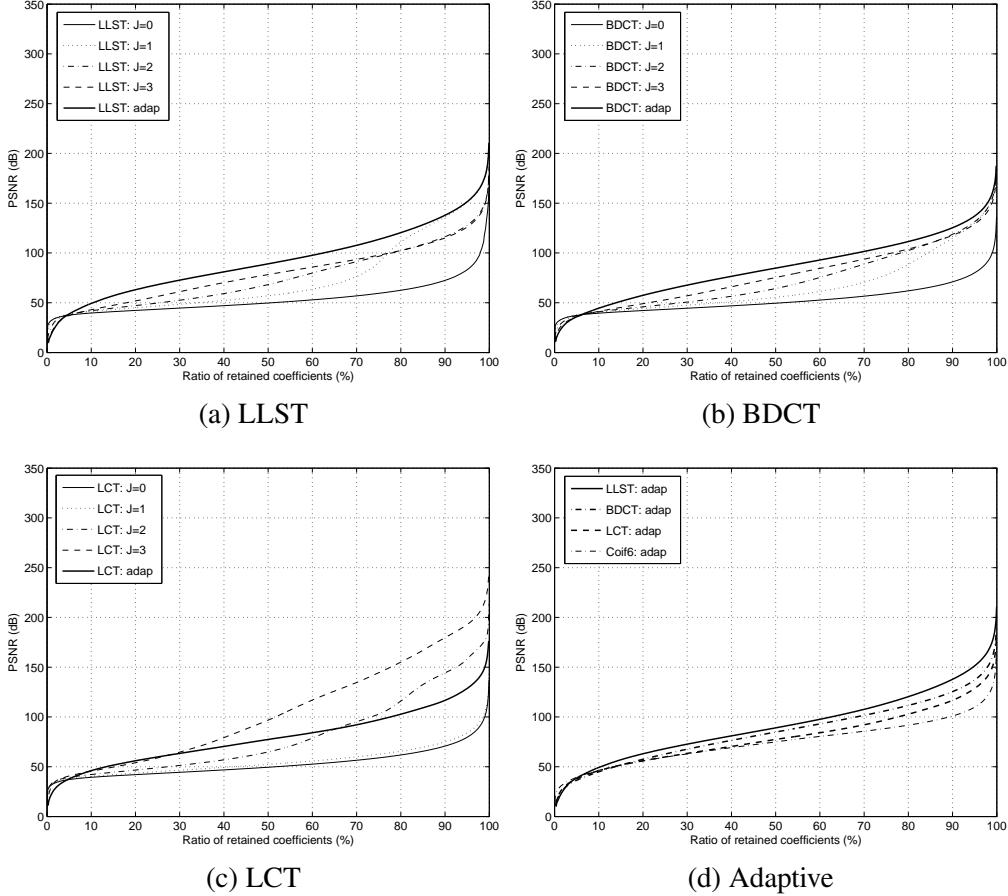


Fig. 7. Quality of approximation of the 2D Gabor data with circular discontinuity measured by PSNR values.

than LCT with $J = 3$ as in the case of the Gaussian with discontinuity for the same reason: the width of the bell function is too narrow due to the smallest block size in this adaptive segmentation. We note that C6BSB in this case tries to subdivide the high frequency bands and performs better than the standard wavelet transform C6DWT unlike the case of Section 4.1.2 where C6BSB is in fact C6DWT.

In summary, our experiments with these synthetic images have shown that LLST is: 1) the best overall performer in the low ratio region of our interest; and 2) most effective if the adaptive segmentation of the domain honoring the geometry of the singularity in the data is used.

4.3 *Barbara Face Image*

Finally, we shall examine the approximation performance of various transforms using a real image. We shall use a face part of the famous image ‘‘Barbara,’’ shown in Figure 1 (a), which has been frequently used for image approximation and com-

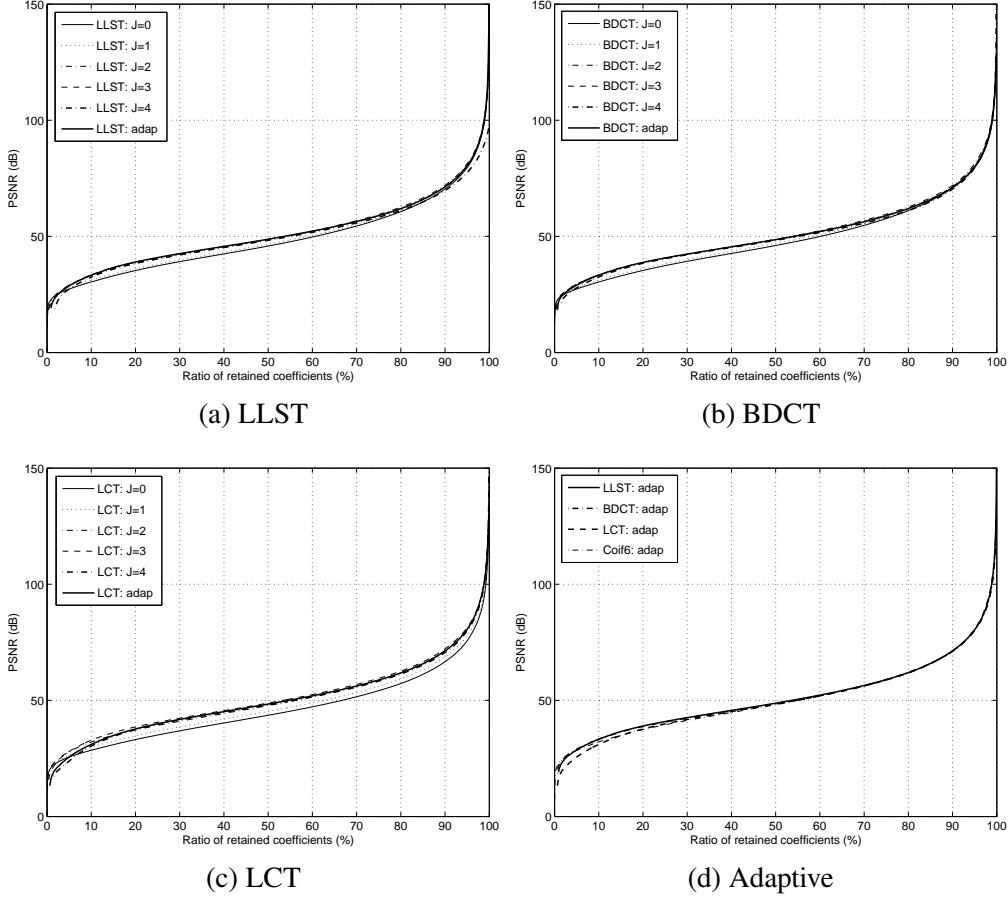


Fig. 8. Quality of approximation of the Barbara face image measured by PSNR values.

pression tests. The size of this face part of the Barbara image is 129×129 pixels. This is much more difficult to approximate than the simple Gaussian/Gabor datasets in the previous subsections because it contains a variety of regions: smooth regions (forehead and cheeks); more complicated regions (eyes, nose, mouth); and regions with oscillatory patterns (scarf). In this experiment, we shall use five different homogeneous splits $J = 0, 1, 2, 3, 4$. Note that the $J = 4$ case corresponds to the block size of 8×8 pixels, i.e., the standard JPEG-DCT scenario. We shall also examine the adaptive segmentation.

Figure 8 shows our results. The zoomed-up versions in the range $R \in [0, 20]$ are shown in Figure 9. We use the IPP program as before to generate the adaptive segmentation for LLST, BDCT, and LCT, which is shown in Figure 10. From these figures, we observe the following:

- The performance of LLST and BDCT with the adaptive segmentation and that of the homogeneous splits with $J = 2, 3$ are virtually the same.
- LLST and BDCT (with the adaptive segmentation or the homogeneous splits with $J = 2, 3$) outperform C6BSB except in the range $R \in [0, 5]$.
- The PSNR values of C6BSB in the range of $R \in [2, 20]$ are 1 dB higher than

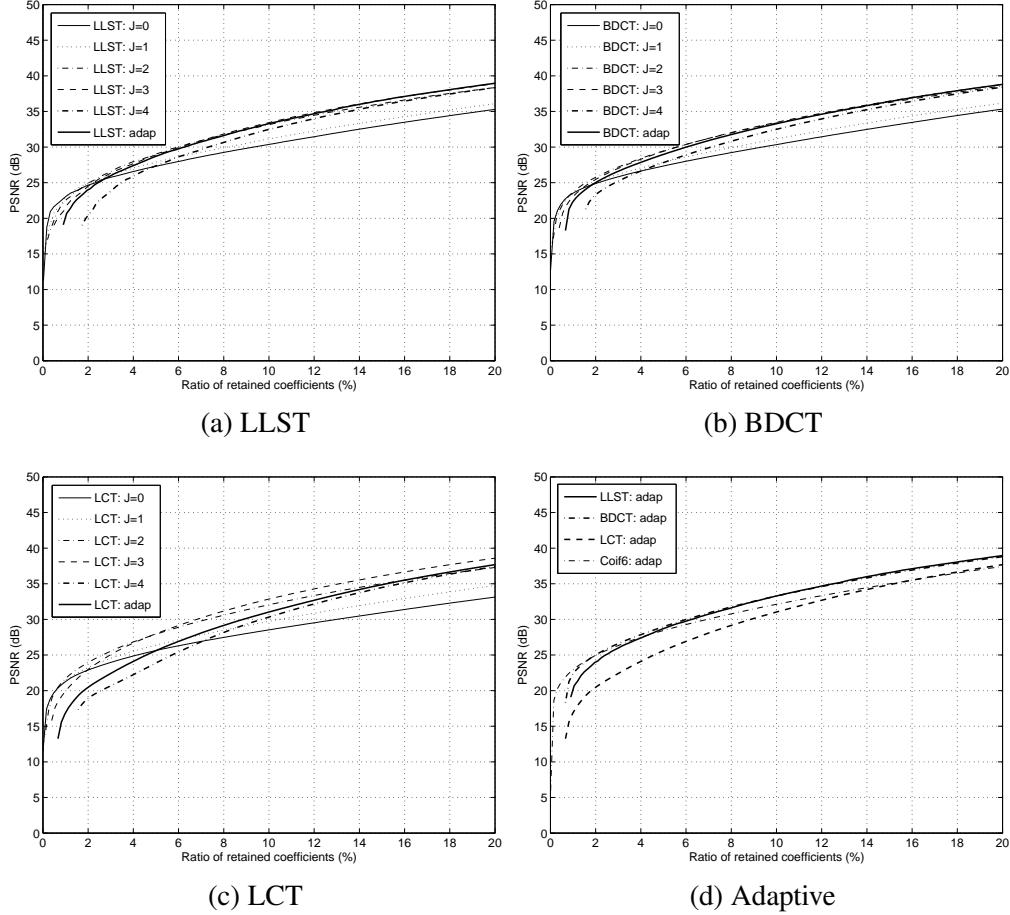


Fig. 9. Zoomed up versions of Figure 8 when 0% to 20% of the original coefficients are retained.

those of C6DWT with $J = 7$ (but not shown in the plots).

- LCT with adaptive segmentation is defeated by LCT with $J = 2, 3$, especially in the low range $R \leq 20\%$, by exactly the same reason as the Gaussian/Gabor cases.
- BDCT with $J = 4$ is defeated by BDCT with $J = 3$. This shows the non-optimality (at least for this image) of the strategy of homogeneously splitting a target image into a set of blocks of size 8×8 pixels, which is adopted in the JPEG Baseline standard.
- Advantage of the adaptive segmentation is less obvious except the C6BSB case for this particular image.

From these observations, unlike the synthetic datasets in Sections 4.1 and 4.2, we can claim neither the superiority of the adaptive segmentation over the homogeneous splits nor that of LLST over the other transforms, in particular, over BDCT *for this particular image and in terms of the PSNR values*. We shall therefore compare the perceptual quality of these approximations further by displaying the reconstructed images. Figure 11 shows the approximations using the top 5 % of the coefficients by these transforms with the adaptive segmentation. Figure 12 shows the

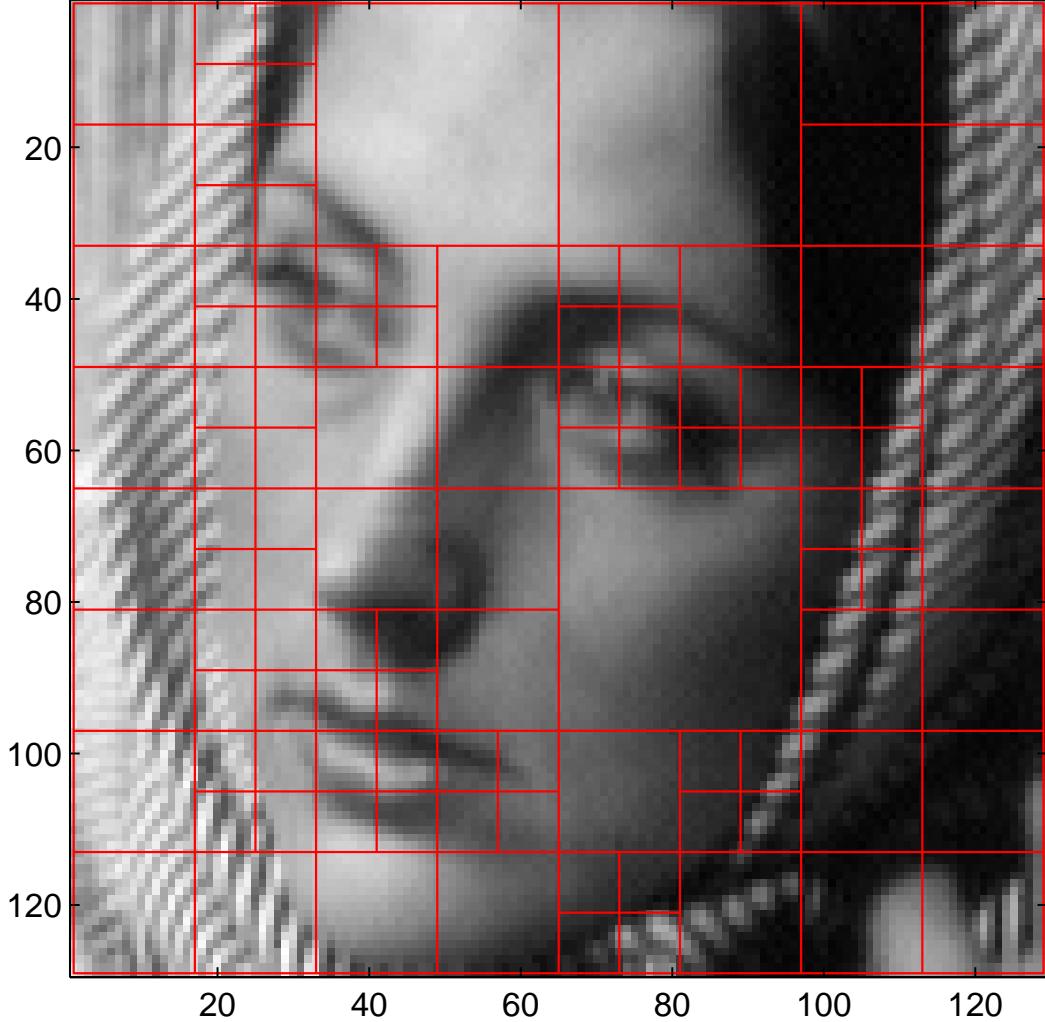


Fig. 10. Adaptive dyadic segmentation of the Barbara face image performed interactively using the Java-based IPP program.

zoomed-up versions to see the quality of those approximations in details. It is clear that the visual quality of the LLST approximation is closest to the original among these transforms. The BDCT approximation reveals the annoying blocking artifact. The LCT approximation is the worst due to the the artifact generated by the folding procedure. The C6BSB approximation is perhaps the next best, but the forehead and cheek portions are less smooth than the LLST approximation, and the stripes of the scarf, especially those in the righthand side, are not well approximated. These experiments suggest that PSNR is not necessarily the best metric for judging the quality of displayed images. In the near future, we plan to use the more perceptually correct metric such as *Mean Structural Similarity Index* (MSSIM) [29], which is based on the comparison of the local patterns of pixel intensities normalized for luminance and contrast.

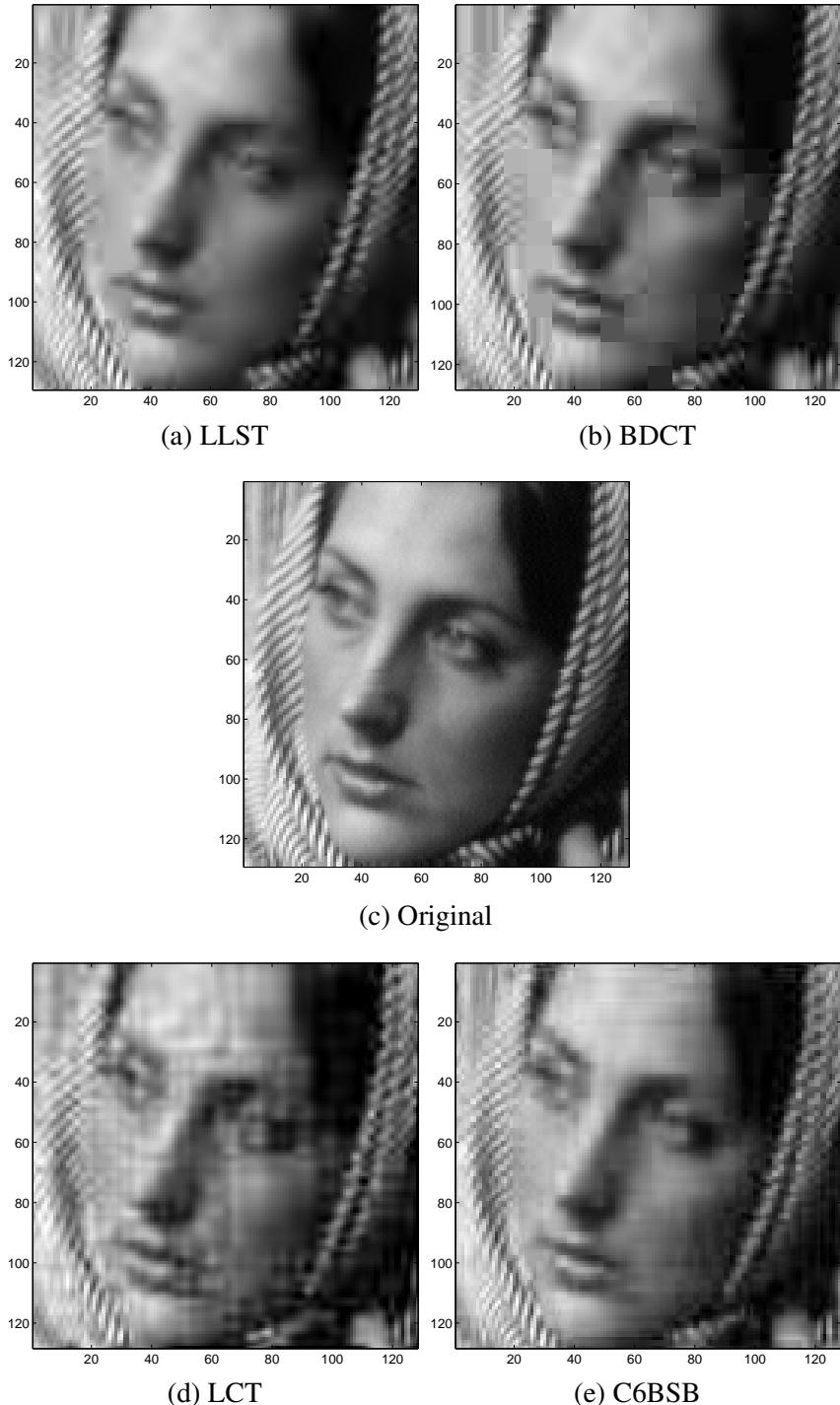


Fig. 11. Approximation of the Barbara face image by LLST, BDCT, LCT, and C6BSB using top 5% of the representations. The adaptive segmentation shown in Figure 10 was used for LLST, BDCT, and LCT. C6BSB was computed from the C6 wavelet packet tree.

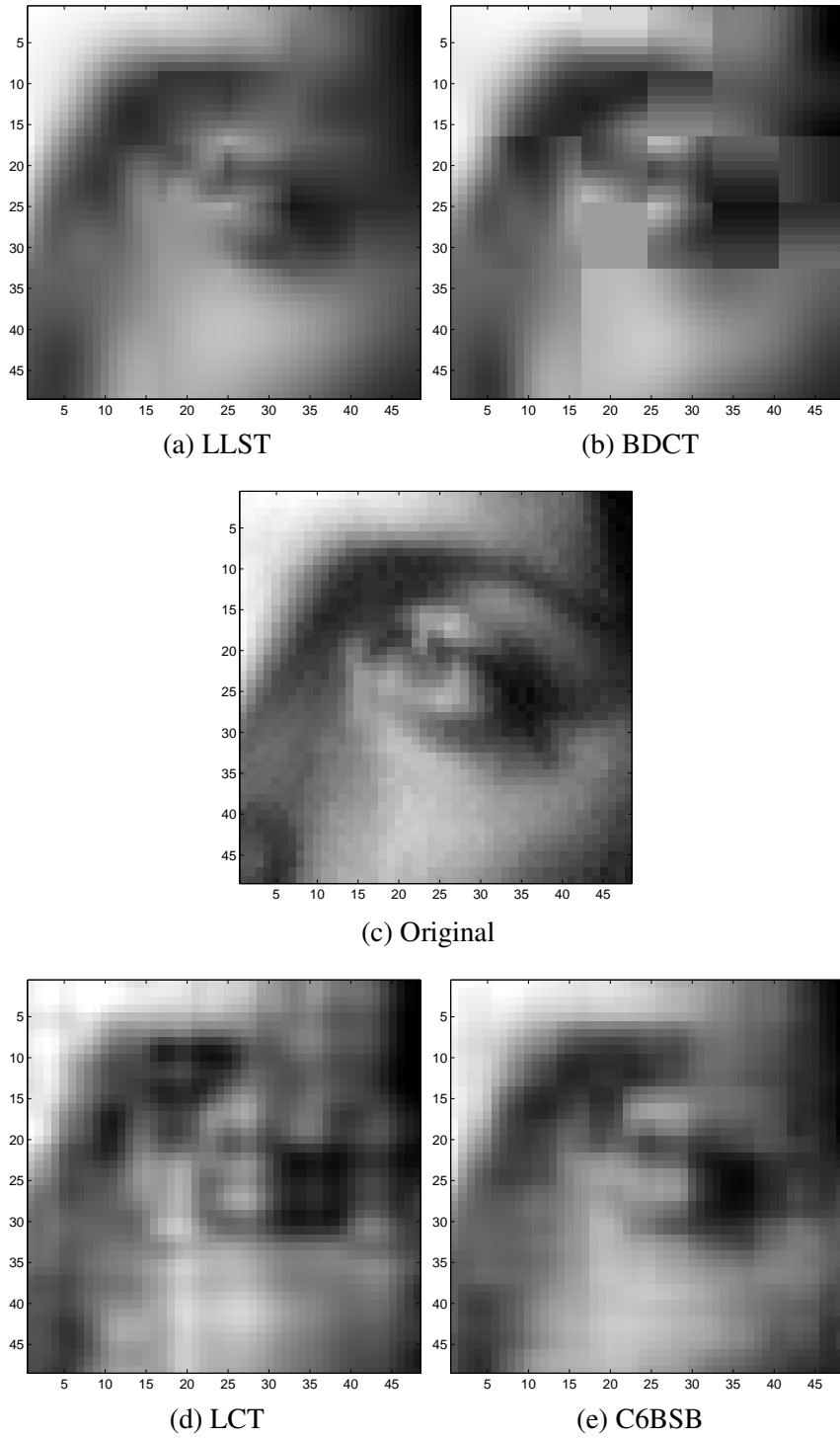


Fig. 12. Zoomed-up versions of Figure 11

5 Polyharmonic Local Fourier Transform: Use of the Complex Exponentials

As we discussed in Section 2, the Fourier sine coefficients of the v_j component of the PHLST decomposition posed as (1), (2), and (5) have the good decay rate, i.e.,

$O(\|\mathbf{k}\|^{-2m-1})$, $m = 1, 2, \dots$. The main motivation of using the sine functions was the extra gain in the decay speed of the expansion coefficients via odd reflection at the boundary of the domain, which in turn is the consequence of the vanishing boundary values of v_j . However, it is well understood now that the *oriented patterns* are not efficiently captured and represented by using only sines (or only cosines). It is much more efficient to use *complex exponentials* for representing oriented patterns than sines alone. Because the boundary values of v_j vanish, it is possible to use the *complex exponentials* instead of sines, i.e., the ordinary Fourier series expansion instead of the Fourier sine series expansion, if one can tolerate slower decay rate of the coefficients. This slower decay is due to the periodization of the v_j component without the odd reflection. We shall call this the *Polyharmonic Local Fourier Transform* (PHLFT). More precisely, let Ω_j be a box of the form $\Omega_j = \mathbf{x}_0^j + \prod_{k=1}^d (a_k^j, b_k^j)$, where \mathbf{x}_0^j is the vertex of the box closest to the origin (i.e., the lower left corner in the case of 2D), and $\ell_k^j \triangleq b_k^j - a_k^j$ is the length of the k th side of this box. Furthermore, let us assume that f is locally C^{2m} on $\overline{\Omega}_j$ with $m \geq 1$. If we use the following boundary condition instead of (2) to solve for the u_j component (1):

$$\frac{\partial^\ell u_j}{\partial \nu^\ell} = \frac{\partial^\ell f}{\partial \nu^\ell} \quad \text{on } \partial\Omega_j, \quad \ell = 0, \dots, m-1, \quad (11)$$

then the resulting v_j component can be extended periodically and becomes $C^{m-1}(\mathbb{R}^d)$. Therefore, we can expand v_j into the Fourier series:

$$v_j(\mathbf{x}) \sim \sum_{\mathbf{k} \in \mathbb{Z}^d} c_{\mathbf{k}} e^{2\pi i \langle \mathbf{k}, L_{\Omega_j}^{-1}(\mathbf{x} - \mathbf{x}_0^j) \rangle}, \quad \mathbf{x} \in \Omega_j,$$

where $L_{\Omega_j} = \text{diag}(\ell_1^j, \dots, \ell_d^j)$, with the coefficients $|c_{\mathbf{k}}|$ decaying as $O(\|\mathbf{k}\|^{-m-1})$; see Theorem 7 in Appendix A. Accordingly, the case with $m = 1$, which we shall call *Laplace Local Fourier Transform* (LLFT), has the decay of order $O(\|\mathbf{k}\|^{-2})$, which is the same as BDCT. But we gain the ability to easily detect and interpret oriented patterns instead, which is difficult with BDCT or LLST. Figure 13 demonstrates this point. If one uses the block Discrete Fourier Transform (BDFT) without removing the u_j components, the DFT coefficients are all contaminated by the Gibbs phenomenon due to the block boundaries as shown in Figure 13 (b). If one uses LLST or BDCT, it is difficult to decipher the orientation information from the coefficients; see Figure 13 (c), (d). On the other hand, LLFT clearly reveals the orientation information as shown in Figure 13 (a). We note that, in practice, one only needs to store the LLFT coefficients in the upper half plane in each block thanks to their symmetry, assuming that an input image is real-valued. LLFT also permits us to compute locally analytic signal, instantaneous frequency, etc., which we will explore at a later date.

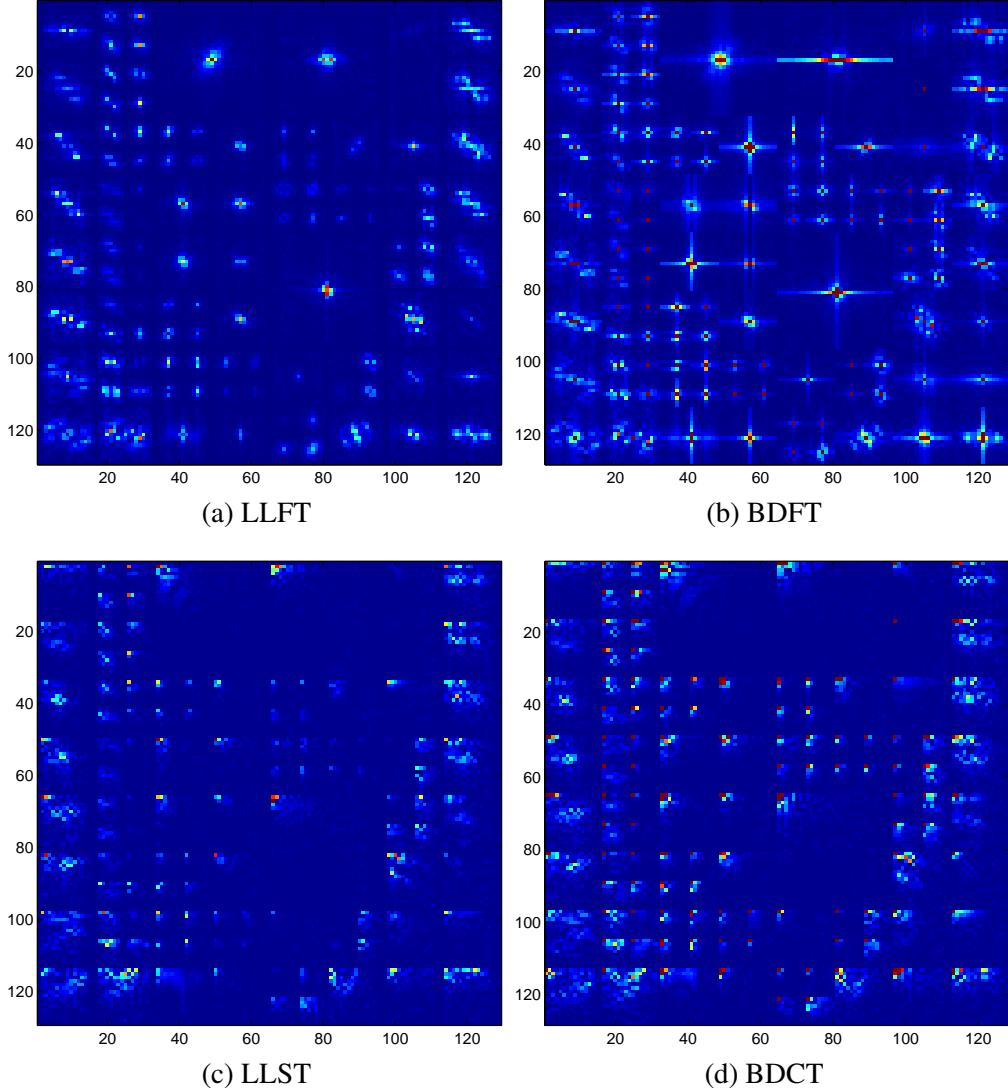


Fig. 13. Magnitude of the coefficients in LLFT, BDFT, LLST, and BDCT of the Barbara face image with the adaptive segmentation of Figure 10. Note that the corner pixel values and 1D DST coefficients of the boundaries are not shown in LLFT and LLST here.

6 Discussion

In this section, we shall first discuss issues that are left open in PHLST. Then we shall describe possible extensions and applications using PHLST and its relatives. We shall also discuss the related concepts and methods proposed previously by other researchers and conclude this article by describing the difference of their work from ours.

6.1 Open Issues

6.1.1 Compression with quantization

A strategy for lossy image compression using PHLST remains an open issue. Unlike the image approximation, which simply keeps a subset of the exact expansion coefficients and corner pixel values as we discussed in Section 4, lossy image compression truly reduces the number of “bits” required to represent a given image approximately and converts it to a stream of bits via the process of *quantization*. In the JPEG Baseline standard [30,31], the DCT-II coefficients in each block of 8×8 pixels are divided by the entries of the so-called Luminance Quantization Table (LQT) and further rounded into the nearest integers. This quantization process tends to suppress the higher frequency coefficients than the lower frequency coefficients so that this process can result in many zeros (i.e., more compression) without sacrificing visual fidelity too much. In fact, LQT was determined by psychophysical experiments on the visibility of the DCT-II basis vectors [30, Sec. 4.1.3] for the square block of 8×8 pixels. The Baseline method further uses the zigzag search of the quantized coefficients in each block to record the end of the block position instead of recording all the subsequent zeros in the block. Finally, this quantized sequence (integers) are encoded by some entropy coder (e.g., the Huffman coder or the arithmetic coder) for further compression in a “lossless” manner.

Unfortunately, we cannot directly use this LQT adopted in the JPEG Baseline standard for PHLST because it is optimized only for DCT-II of 8×8 pixel blocks, and not optimal for any other settings. In fact, it is not optimal even for DCT-II if the size of a block is different from 8×8 pixels. Therefore, if we wish to follow this approach, we would need to calibrate the LQT specifically for DST used in PHLST for various pixel sizes such as 9×9 , 17×17 , 33×33 , etc. It would be too costly (in human resources) to conduct such psychophysical experiments. Also, the PHLST representation consists of three different items in each block: four corner pixel values, the 1D DST coefficients of the block boundary; and the 2D DST coefficients of the residual component. We need to examine whether these three components should be quantized separately or not.

We are currently investigating simpler uniform quantizers with “dead zone” (a special bin covering the zero value). This dead zone can get rid of noise in input data if its width is appropriately chosen because any coefficients within this zone are quantized to zero, i.e., killed. Thus, the width of the dead zone plays the role of the threshold in denoising algorithms. The subtle and interesting issue here is how to choose the width of the dead zone as well as that of the uniform bins outside of the dead zone, which determines the resolution of the quantization. Yet another open issue is that of encoding after quantization. Just like the JPEG Baseline standard, it is possible to use the zigzag search to detect the trailing zeros in each block, and encode the resulting quantized sequences via Huffman coder. However, it still

remains to be investigated whether such an approach is most suitable for PHLST.

6.1.2 Segmentation and denoising

Another open problem is how to automatically and adaptively segment a given input image to achieve more efficient approximation and compression. Note that what we mean segmentation here is different from usual image segmentation studied in the image analysis and computer vision literature where one wishes to faithfully segment an image into a set of objects or domains whose boundaries are general smooth curves. What we want to do here is to simply split an image into a set of adaptive dyadic blocks honoring the geometric structure of input data as much as possible. For example, we wish to obtain segmentations like those shown in Figures 5, 10 completely automatically rather than manually using an interactive program like IPP. We already demonstrated that such adaptive dyadic splits help image approximation in Section 4. We need to confirm that the same is true for image compression, which necessarily includes the cost of recording the geometry of adaptive splits.

We are currently investigating a hierarchical and recursive split-and-merge strategy similar to the best-basis algorithm of Coifman and Wickerhauser [32] in spirit. Clearly, the key is what criterion we should employ to judge whether smaller blocks should be merged into larger blocks. One of the most promising and natural criteria is the *Minimum Description Length* (MDL) criterion [33–36]. Viewing each split pattern of an image in PHLST as a *model*, the MDL criterion suggests that we should choose among many possible models the one giving the shortest description (counted in bits) of the data *and* the model. This leads to the minimization of the infidelity of the model to the data and the complexity of the model itself in a balanced manner. In the case of PHLST, the MDL criterion is particularly appealing since PHLST does not provide an orthonormal basis. Instead, it provides “nonhomogeneous” representations using the corner pixels, 1D DST coefficients of the boundaries, and the 2D DST coefficients of the residuals, as explained in Section 4. Thus, the sparsity criterion using ℓ^p norm with $0 < p \leq 1$ makes less sense contrary to the case of the libraries of orthonormal bases such as local cosines and wavelet packets. It makes more sense to count the description length in bits needed to encode such nonhomogeneous representations.

There are several additional advantages to use the MDL criterion here. First, it forces us to incorporate a quantization procedure to convert the representation into a sequence of bits. Secondly, it can achieve *simultaneous compression and denoising* of input data. Similarly to the case of wavelets and the libraries of orthonormal bases shown by Saito [36], Chang et al. [37], and Hansen and Yu [38] among others, we expect that the MDL-based method will achieve the desired results for PHLST. Finally, the MDL formulation allows us to incorporate easily various noise models and prior models for distribution of the expansion coefficients: it can automatically

select the most suitable noise model and coefficient distribution models out of many competing models. This is advantageous since it is well known that the Laplace (or double exponential) distribution or the generalized Gaussian distribution are more suitable to model the wavelet coefficients [38] and DCT coefficients [39] of natural images than the Gaussian distribution. The DST coefficients in PHLST are expected to behave in a similar manner.

6.1.3 Nonlinear approximation with PHLST

Another open issue is to obtain precise nonlinear approximation results using PHLST. Such a task remains elusive compared to the case of the orthonormal wavelets or splines. Let us briefly explain why this is so. First, let us introduce some important concepts of the nonlinear approximation. Let X be some function space (e.g., the Besov space) defined on a domain $\Omega \subset \mathbb{R}^d$, and let $\Sigma_n \subset X$ be a nonlinear manifold so that any function belonging to Σ_n can be exactly representable by at most n parameters. Examples of such Σ_n include a set of functions that can be exactly specified by at most n basis functions selected from a basis set for X and a set of piecewise constant functions with n knots or breakpoints for $d = 1$. We now define the n -term approximation error of $f \in X \subset L^p(\Omega)$ as follows:

$$\sigma_n(f)_p \triangleq \inf_{g \in \Sigma_n} \|f - g\|_p,$$

where $1 \leq p \leq \infty$. In the nonlinear approximation setting, the goodness of Σ_n is measured by the decay rate of the worst case n -term approximation error as n increases:

$$\sigma_n(X)_p \triangleq \sup_{f \in X} \sigma_n(f)_p \leq Cn^{-\alpha}, \quad (12)$$

where $\alpha > 0$. Clearly, the larger the value of α , the better the manifold Σ_n . The primary difficulty in obtaining n -term approximation results for PHLST lies in the strategy to optimally distribute the resources of n terms (or numbers) to the geometry information $\{\Omega_j\}$, the polyharmonic components $\{u_j\}$, and the residuals $\{v_j\}$. Even with a homogeneous split of Ω , it is challenging to derive an optimal allocation of n terms into the $\{u_j\}$ and $\{v_j\}$ components. Another difficulty, which is clearly related to the primary one above, is to determine a good function space X that allows us to obtain the best possible n -term approximation results, i.e., the largest $\alpha > 0$ in (12). We need to start attacking a simpler case of 1D functions here since DeVore in his plenary lecture [40] pointed out that obtaining nonlinear approximation results for an even simpler “Spline-Fourier” method with $d = 1$ (where a target function is segmented into several pieces brutally and each piece is approximated by the truncated Fourier series) is still open. On the other hand, it is encouraging to know that if $\Omega = [0, 1]^d$ without any further split, then the precise nonlinear approximation results are known [41].

6.2 Extensions and Applications

In this subsection, we shall list possible extensions and applications of PHLST and its relatives.

6.2.1 PHLST/PHLFT in the frequency domain

Nothing prevents us from operating PHLST/PHLFT in the frequency domain. In fact, this strategy may be an excellent alternative to the wavelet packets and the brushlets [42]. The frequency support of each wavelet packet basis function is not completely localized except for the Shannon wavelet packets. On the other hand, each of the Shannon wavelet packets has unbounded support in the spatial domain, which makes them less useful. The frequency support of each brushlet basis functions is more localized than that of the wavelet packets, but still overlaps with that of the adjacent brushlets due to the use of the folding operations in the frequency domain. The PHLST/PHLFT in the frequency domain may overcome these shortcomings of the wavelet packets and the brushlets. We note that it is important to subtract the global u component from the original image f before converting it to the frequency domain. Otherwise, sharp horizontal and vertical lines crossing at the origin (the DC component) are generated due to the Gibbs phenomenon from the domain boundary, which interfere our analysis of the frequency information of the image. This is evident from Figure 13 (b), which shows the local version of this annoying interference.

6.2.2 Use of the Neumann boundary condition

It is possible to use the *Neumann boundary condition* instead of the Dirichlet condition in (3). This will give us the $O(\|\mathbf{k}\|^{-4})$ decay rate of the v_j components instead of $O(\|\mathbf{k}\|^{-3})$ in the case of the Dirichlet boundary condition. To do so, we need to modify (3); we shall solve the following *Poisson equation* with the Neumann boundary condition:

$$\begin{cases} \Delta u_j = K_j & \text{in } \Omega_j \\ \frac{\partial u_j}{\partial \nu} = \frac{\partial f_j}{\partial \nu} & \text{on } \partial \Omega_j \end{cases}, \quad (13)$$

where K_j is a constant that needs to be computed as follows:

$$K_j = \frac{1}{|\Omega_j|} \int_{\partial \Omega_j} \frac{\partial f_j}{\partial \nu} ds,$$

which is the boundary integral of $\partial f_j / \partial \nu$ normalized by $|\Omega_j|$ (the volume of the domain). This constant K_j is necessary for (13) to have a unique solution (modulo an additive constant); see e.g., [43, page 84]. The solution u_j is not just a function having the same normal derivative at the boundary with the original function f_j ;

one can show that it is *the minimizer of the total squared curvature integral* on the domain Ω_j . Once we get the solution u_j , the residual function $v_j = f_j - u_j$ clearly satisfies $\partial v_j / \partial \nu = 0$ on $\partial \Omega_j$. Assuming that $f_j \in C^2(\overline{\Omega}_j)$, we can show that the residual v_j has at least C^2 smoothness across the block boundary when it is extended by the even reflection. Therefore, if we use the Fourier *cosine* series expansion of v_j we get the coefficients with decay rate $O(\|\mathbf{k}\|^{-4})$. See also Theorem 7 in Appendix A. We naturally call this version of the transform the *Polyharmonic Local Cosine Transform* (PHLCT). However, this method of course requires to estimate the first order normal derivative of data at the boundary. We are currently investigating this issue with our collaborator, Katsu Yamatani, and getting encouraging results [44].

6.2.3 Extension to higher orders of polyharmonicity

We believe that PHLST with polyharmonicity $m > 2$ is not practical considering the need to estimate $\partial^{2(m-1)} f_j / \partial \nu^{2(m-1)}$ in (2) and the numerical sensitivity to compute these high order derivatives. However, the case of $m = 2$, i.e., BLST (biharmonic local sine transform), remains to be investigated; we are interested in knowing whether BLST is of practical interest for image approximation and compression or not. In theory, the Fourier sine coefficients of the v_j components in BLST should decay as $O(\|\mathbf{k}\|^{-5})$ instead of $O(\|\mathbf{k}\|^{-3})$ in LLST. However, the effectiveness of BLST still hinges on the accuracy of the second order normal derivative estimates at the boundary. Thus developing a robust method to estimate those derivatives from noisy data will be an challenging but important project. Note that such a method will also be useful for other purposes, for example, edge detection.

6.2.4 Extension to higher dimensions

It is important to realize that the information contained in the boundary becomes more and more meaningful as the dimension increases. For example, in 1D, the boundary of an interval simply consists of two endpoints. In 2D, the boundary of a domain becomes a 1D curve whereas in 3D, this becomes a 2D surface. If the data is supported in a 3D cube, then the boundary consists of six faces of that cube. It is also important to recognize that the boundary structure is *recursively organized*. For example, the boundary of each 2D face of a 3D cube consists of four 1D edges, and the boundary of each 1D edge in turn consists of two corners. We can take advantage of this recursive structure: we can process each component, i.e., edges, faces, bodies, etc., by PHLST of the corresponding dimensions. The simple, hierarchical, and recursive algorithmic structure of PHLST can be carried over to higher dimensions in a straightforward manner. In particular, PHLST for 3D datasets is promising for efficient approximation, compression, and local feature computation for 3D medical images and 3D geophysical data, which we are currently investigating in collaboration with our graduate student Noel Smith [45].

6.2.5 Other potential applications

Because of its ability to completely localize the analysis of an input image and sparsify its representation equipped with a fast algorithm, there are many potential applications of PHLST. For example, its application to image zooming and interpolation is promising since PHLST does not suffer from the boundary effect/the Gibbs phenomenon, which is often a nuisance in image zooming. Commonly used image zooming/interpolation techniques such as pixel replication, bilinear interpolation, bi-cubic interpolation have all drawbacks, especially with large magnifications. The pixel replication is computationally most efficient, but generates annoying blocky zoomed images. The bilinear interpolation generates superfluous horizontal and vertical artifacts, and this is also true for the bi-cubic interpolation to a lesser extent. One can use DCT for image zooming, but again one may see the artifacts generated by the even reflection at the boundaries. Sinc (or band-limited) interpolation is theoretically ideal but its straightforward implementation does not work well due to the Gibbs phenomenon.

Another important application of PHLST is local feature computation. PHLST allows us to evaluate and compute various attributes (e.g., directional derivatives) at *any point* in the domain Ω_j even if we start from discrete samples. See Equations (7)–(10). We are currently investigating the interpolation of image values along a given *curve* as well as directional derivatives along such a curve, which is not easy using the conventional methods.

6.3 Relation to the Other Work

Retrospectively speaking, our work is related to many previous works done by other scientists. However, we did not find exactly the same strategy as ours in the other works. As early as 1938, Cornelius Lanczos [46] suggested that “denoising” and interpolation of digital signals sampled on a *equispaced* grid over a *finite interval* should be very nicely dealt with by trigonometric polynomials once the edge effect is taken care of. This was only an abstract given at the AMS conference. Somehow, a full version of the paper did not appear until 1952 [47]. He first used the line removal idea [46] corresponding to the 1D LLST in our case, and then proposed to remove a higher order polynomials [47,48]. As far as we know, Lanczos is one of the first scientists who recognized the importance of representing and interpolating data sampled at equispaced grid points using a combination of trigonometric polynomials and algebraic polynomials. He clearly understood that only using algebraic polynomials leads to the infamous Runge phenomenon, and only using trigonometric polynomial leads to the Gibbs phenomenon. Zygmund had a similar idea (i.e., removal of singularities) for different purposes in 1935 or earlier [49, Sec. 2.13]. See also [2, Sec. 2.2], [50, Chap. 4], [51, Chap. 2], and [48, Chap. 2]. The book of Kantrovich and V. I. Krylov [52] cites an early attempt of extending a function

supported on $[0, 1]$ to $[-1, 1]$ smoothly and periodically (with period 2) by Maliev [53,54]. Smirnov [55] also cites the work of A. N. Krylov for speeding up the decay of Fourier coefficients of a compactly supported function using the line and polynomial removal.

In 1990, Madych and Nelson [56,57] introduced the so-called “polyharmonic cardinal splines”, which uses the polyharmonic equations to interpolate the data given on the lattice \mathbb{Z}^d in \mathbb{R}^d . Their main concern, however, is the interpolation, and they are not concerned with the residuals at all.

The “polysplines” proposed by Kounchev [58,59] are also related to our PHLST. In his case, though, he used a sequence of decreasing subdomains $\Omega = \Omega_0 \supset \Omega_1 \supset \dots$ instead of the disjoint subdomains of our case. This led him to develop a new version of multidimensional wavelets. Again, he is not concerned with the residuals.

All of the above scientists except Lanczos only focused on the u component. As far as we know, Lanczos was the only one who seriously considered the residual v , but explored his idea neither for higher dimensions nor for multiscale setting.

Acknowledgements

This research was partially supported by the ONR grant N00014-00-1-0469 and the NSF grant DMS-0410406. We thank Prof. Gregory Beylkin, University of Colorado at Boulder for fruitful discussions on the numerical issues of the Laplace equation solvers, Prof. Grant Welland, University of Missouri at St. Louis for checking the proof of Theorem 7, and Prof. Raphy Coifman, Yale University for his warm encouragement.

References

- [1] G. H. Hardy, W. W. Rogosinski, Fourier Series, 3rd Edition, Cambridge Univ. Press, 1956, republished by Dover Publications, Inc. in 1999.
- [2] A. Zygmund, Trigonometric Series, 3rd Edition, Cambridge Mathematical Library, Cambridge Univ. Press, 2003, volumes I & II combined.
- [3] M. A. Pinsky, Introduction to Fourier Analysis and Wavelets, Brooks/Cole, 2002.
- [4] A. Cohen, I. Daubechies, P. Vial, Wavelets on the interval and fast wavelet transforms, Appl. Comput. Harmonic Anal. 1 (1) (1993) 54–81.
- [5] B. K. Alpert, A class of bases in L^2 for the sparse representation of integral operators, SIAM J. Math. Anal. 24 (1) (1993) 246–262.

- [6] R. R. Coifman, Y. Meyer, Remarques sur l'analyse de Fourier à fenêtre, *Comptes Rendus Acad. Sci. Paris, Série I* 312 (1991) 259–261.
- [7] H. S. Malvar, The LOT: transform coding without blocking effects, *IEEE Trans. Acoust., Speech, Signal Process.* 37 (1989) 553–559.
- [8] H. S. Malvar, Lapped transforms for efficient transform/subband coding, *IEEE Trans. Acoust., Speech, Signal Process.* 38 (1990) 969–978.
- [9] X. Fang, E. Séré, Adapted multiple folding local trigonometric transforms and wavelet packets, *Appl. Comput. Harmonic Anal.* 1 (1994) 169–179.
- [10] L. F. Villemoes, Adapted bases of time-frequency local cosines, *Applied and Computational Harmonic Analysis* 10 (2001) 139–162.
- [11] H. Whitney, Analytic extensions of differentiable functions defined in closed sets, *Trans. Amer. Math. Soc.* 36 (1) (1934) 63–89.
- [12] B. Larson, N. Saito, The continuous boundary local Fourier transform, in: A. F. Laine, M. A. Unser, A. Aldroubi (Eds.), *Wavelets IX*, Vol. Proc. SPIE 4478, 2001, pp. 415–426.
- [13] B. Larson, The continuous boundary local trigonometric transform, Ph.D. thesis, Dept. Math., Univ. California, Davis (2002).
- [14] M. V. Wickerhauser, Smooth localized orthonormal bases, *Comptes Rendus Acad. Sci. Paris, Série I* 316 (1993) 423–427.
- [15] M. V. Wickerhauser, *Adapted Wavelet Analysis from Theory to Software*, A K Peters, Ltd., Wellesley, MA, 1994, with diskette.
- [16] G. Wahba, *Spline Models for Observational Data*, Vol. 59 of CBMS-NSF Regional Conference Series in Applied Mathematics, SIAM, Philadelphia, 1990.
- [17] Y. Meyer, *Oscillating Patterns in Image Processing and Nonlinear Evolution Equations*, Vol. 22 of University Lecture Series, AMS, Providence, RI, 2001.
- [18] S. Jaffard, Y. Meyer, R. D. Ryan, *Wavelets: Tools for Science & Technology*, SIAM, Philadelphia, PA, 2001.
- [19] A. Averbach, M. Israeli, L. Vozovoi, A fast Poisson solver of arbitrary order accuracy in rectangular regions, *SIAM J. Sci. Comput.* 19 (3) (1998) 933–952.
- [20] E. Braverman, M. Israeli, A. Averbach, L. Vozovoi, A fast 3D Poisson solver of arbitrary order accuracy, *J. Comput. Phys.* 144 (1998) 109–136.
- [21] W. L. Briggs, V. E. Henson, *The DFT: An Owner's Manual for the Discrete Fourier Transform*, SIAM, Philadelphia, PA, 1995.
- [22] A. Iserles, *A First Course in the Numerical Analysis of Differential Equations*, Cambridge Texts in Applied Mathematics, Cambridge Univ. Press, New York, 1996.
- [23] G. Strang, The discrete cosine transform, *SIAM Review* 41 (1) (1999) 135–147.

- [24] R. A. DeVore, Nonlinear approximation, in: *Acta Numerica*, Cambridge Univ. Press, 1998, pp. 51–150.
- [25] I. Daubechies, Orthonormal bases of compactly supported wavelets II. Variations on a theme, *SIAM J. Math. Anal.* 24 (2) (1993) 499–519.
- [26] I. Daubechies, *Ten Lectures on Wavelets*, Vol. 61 of CBMS-NSF Regional Conference Series in Applied Mathematics, SIAM, Philadelphia, PA, 1992.
- [27] B. Bénichou, N. Saito, Sparsity vs. statistical independence in adaptive signal representations: A case study of the spike process, in: G. V. Welland (Ed.), *Beyond Wavelets*, Vol. 10 of Studies in Computational Mathematics, Academic Press, San Diego, CA, 2003, Ch. 9, pp. 225–257.
- [28] N. Saito, The generalized spike process, sparsity, and statistical independence, in: D. Rockmore, J. D. Healy (Eds.), *Modern Signal Processing*, Vol. 46 of MSRI Publications, Cambridge University Press, 2004, pp. 317–340.
- [29] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, Image quality assessment: From error measurement to structural similarity, *IEEE Trans. Image Process.* 13 (4) (2004) 600–613.
- [30] W. B. Pennebaker, J. L. Mitchell, *JPEG Still Image Data Compression Standard*, Van Nostrand Reinhold, New York, 1993.
- [31] G. K. Wallace, The JPEG still picture compression standard, *IEEE Trans. Consumer Electronics* 38 (1) (1992) xviii–xxxiv.
- [32] R. R. Coifman, M. V. Wickerhauser, Entropy-based algorithms for best basis selection, *IEEE Trans. Inform. Theory* 38 (2) (1992) 713–719.
- [33] J. Rissanen, *Stochastic Complexity in Statistical Inquiry*, World Scientific, Singapore, 1989.
- [34] M. H. Hansen, B. Yu, Model selection and the principle of minimum description length, *J. Amer. Statist. Assoc.* 96 (454) (2001) 746–774.
- [35] Y. G. Leclerc, Constructing simple stable descriptions for image partitioning, *Intern. J. Computer Vision* 3 (1989) 73–102.
- [36] N. Saito, Simultaneous noise suppression and signal compression using a library of orthonormal bases and the minimum description length criterion, in: E. Foufoula-Georgiou, P. Kumar (Eds.), *Wavelets in Geophysics*, Academic Press, San Diego, CA, 1994, Ch. XI, pp. 299–324.
- [37] S. G. Chang, B. Yu, M. Vetterli, Adaptive wavelet thresholding for image denoising and compression, *IEEE Trans. Image Process.* 9 (9) (2000) 1532–1546.
- [38] M. Hansen, B. Yu, Wavelet thresholding via MDL: simultaneous denoising and compression, *IEEE Trans. Inform. Theory* 46 (5) (2000) 1778–1788.
- [39] E. Y. Lam, J. W. Goodman, A mathematical analysis of the DCT coefficient distributions for images, *IEEE Trans. Image Process.* 9 (10) (2000) 1661–1666.

- [40] R. A. DeVore, Wavelets and approximation, Presented at the SPIE Conference on Wavelets X (Aug. 2003).
- [41] R. A. DeVore, V. N. Temlyakov, Nonlinear approximation by trigonometric sums, *J. Fourier Anal. Appl.* 2 (1) (1995) 29–48.
- [42] F. G. Meyer, R. R. Coifman, Brushlets: a tool for directional image analysis and image compression, *Appl. Comput. Harmonic Anal.* 4 (1997) 147–187.
- [43] G. B. Folland, *Introduction to Partial Differential Equations*, 2nd Edition, Princeton Univ. Press, 1995.
- [44] K. Yamatani, N. Saito, Improvement of DCT-based compression algorithms using Poisson's equation, Tech. rep., Dept. Math., Univ. California, Davis, in preparation (2005).
- [45] N. Smith, The higher dimensional extension of the polyharmonic local trigonometric transforms, Ph.D. thesis, Dept. Math., Univ. California, Davis, in preparation (2005).
- [46] C. Lanczos, A simple interpolation method for the representation of rugged curves, Abstract of contributed paper given at the November Meeting of the AMS in Cleveland, OH, Nov. 25–26, 1938.
- [47] C. Lanczos, Smoothing of noisy data by trigonometric truncation, in: National Bureau of Standards Report, no. 1582, 1952, pp. 1–16.
- [48] C. Lanczos, *Discourse on Fourier Series*, Hafner Publishing Co., New York, 1966.
- [49] A. Zygmund, *Trigonometrical Series*, Chelsea Publishing Co., 1952.
- [50] C. Lanczos, *Applied Analysis*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1956, republished by Dover Publications, Inc. in 1988.
- [51] C. Lanczos, *Linear Differential Operators*, D. Van Nostrand Co., New York, 1961, republished by Dover Publications, Inc. in 1997.
- [52] L. V. Kantrovich, V. I. Krylov, *Approximate Methods of Higher Analysis*, Interscience Publishers, Inc., New York, 1958.
- [53] A. S. Maliev, Fourier series of heightened convergence for functions defined in a given interval, *Izv. AN SSSR OMEN* 10 (1932) 1437–1450.
- [54] A. S. Maliev, On the expansion in Fourier series of heightened convergence of functions defined in a given interval, *Izv. AN SSSR OMEN* 8 (1933) 1113–1120, (Russian).
- [55] V. I. Smirnov, *A Course of Higher Mathematics*, Vol. II, Pergamon Press, New York, 1964, translated by D. E. Brown.
- [56] W. R. Madych, S. A. Nelson, Polyharmonic cardinal splines, *J. Approx. Theory* 60 (2) (1990) 141–156.
- [57] W. R. Madych, S. A. Nelson, Polyharmonic cardinal splines: A minimization property, *J. Approx. Theory* 63 (3) (1990) 303–320.

- [58] O. I. Kounchev, Minimizing the Laplacian of a function squared with prescribed values on interior boundaries–theory of polysplines, Trans. Amer. Math. Soc. 350 (5) (1998) 2105–2128.
- [59] O. Kounchev, Multivariate Polysplines: Applications to Numerical and Wavelet Analysis, Academic Press, San Diego, CA, 2001.
- [60] M. Taibleson, Fourier coefficients of functions of bounded variation, Proc. Amer. Math. Soc. 18 (4) (1967) 766.
- [61] T. W. Körner, Fourier Analysis, Cambridge Univ. Press, 1988.

A Proof of Theorem 1

First we prove the following auxiliary theorems:

Theorem 6 *Let \mathbb{T} be a torus (i.e., $[0, 1]$), and $f \in C^m(\mathbb{T})$ and periodic. Furthermore, let us assume that $f^{(m+1)}$ exists and in $BV[0, 1]$. Then its Fourier coefficient $c_k = \hat{f}(k)$ decays as $O(|k|^{-m-2})$.*

PROOF. We use the periodicity of f , i.e., $f^{(\ell)}(0) = f^{(\ell)}(1)$, $\ell = 0, 1, \dots, m$, and integration by parts.

$$\begin{aligned}
\hat{f}(k) &= \int_0^1 f(x) e^{-2\pi ikx} dx \\
&= \frac{e^{-2\pi ikx}}{-2\pi ik} f(x) \Big|_0^1 + \frac{1}{2\pi ik} \int_0^1 f'(x) e^{-2\pi ikx} dx \\
&= \frac{e^{-2\pi ikx}}{-(2\pi ik)^2} f'(x) \Big|_0^1 + \frac{1}{(2\pi ik)^2} \int_0^1 f''(x) e^{-2\pi ikx} dx \\
&= \dots \\
&= \frac{e^{-2\pi ikx}}{-(2\pi ik)^{m+1}} f^{(m)}(x) \Big|_0^1 + \frac{1}{(2\pi ik)^{m+1}} \int_0^1 f^{(m+1)}(x) e^{-2\pi ikx} dx \\
&= \frac{1}{(2\pi ik)^{m+1}} \int_0^1 f^{(m+1)}(x) e^{-2\pi ikx} dx
\end{aligned}$$

By assumption, $f^{(m+1)} \in BV[0, 1]$, so we can use the theorem of Taibleson [60] to get:

$$|\hat{f}(k)| \leq \frac{V(f^{(m+1)})}{(2\pi)^{m+1} |k|^{m+2}},$$

where $V(\cdot)$ is the total variation of the argument over $[0, 1]$. This implies that $\hat{f}(k) = O(|k|^{-m-2})$. \square

Theorem 7 Let \mathbb{T}^d be a d -dimensional torus (i.e., $[0, 1]^d$), and $f \in C^m(\mathbb{T}^d)$ and periodic. Furthermore, let us assume that $\partial_j^{m+1} f$ exists and in $BV[0, 1]$, $j = 1, \dots, d$. Then its Fourier coefficient $c_{\mathbf{k}} = \hat{f}(\mathbf{k})$ decays as $O(\|\mathbf{k}\|^{-m-2})$.

PROOF. We proceed essentially the same way as Körner [61, page 409]. We consider each variable x_j in turn. Since f as a function of x_j (with all the other variables fixed) belongs to $C^m(\mathbb{T})$ and $\partial_j^{m+1} f \in BV[0, 1]$, we can use Theorem 6 as:

$$\left| \int_{\mathbb{T}} f(x_1, \dots, x_j, \dots, x_d) e^{-2\pi i k_j x_j} dx_j \right| \leq \frac{C_j}{|k_j|^{m+2}}, \quad (\text{A.1})$$

where $C_j > 0$ is a constant depending on j and k_j is a j th frequency index (nonzero integer). Then,

$$\begin{aligned} |\hat{f}(\mathbf{k})| &= \left| \int_{\mathbb{T}} \cdots \int_{\mathbb{T}} f(x_1, \dots, x_d) e^{-2\pi i \sum_j k_j x_j} dx_1 \cdots dx_d \right| \\ &\leq \int_{\mathbb{T}} \cdots \int_{\mathbb{T}} \left| \int_{\mathbb{T}} f(x_1, \dots, x_d) e^{-2\pi i k_j x_j} dx_j e^{-2\pi i \sum_{\ell \neq j} k_{\ell} x_{\ell}} \right| dx_1 \cdots dx_{j-1} dx_{j+1} \cdots dx_d \\ &\leq \int_{\mathbb{T}} \cdots \int_{\mathbb{T}} \frac{C_j}{|k_j|^{m+2}} dx_1 \cdots dx_{j-1} dx_{j+1} \cdots dx_d \\ &= \frac{C_j}{|k_j|^{m+2}}. \end{aligned}$$

Therefore, we have the following estimate:

$$|\hat{f}(\mathbf{k})| \leq \frac{C_j}{|k_j|^{m+2}} \quad j = 1, \dots, d, k_j \neq 0. \quad (\text{A.2})$$

Thus, if $\mathbf{k} \neq 0$, then

$$\begin{aligned} |\hat{f}(\mathbf{k})| &\leq \min_{1 \leq j \leq d} C_j |k_j|^{-m-2} \\ &\leq \left(\max_{1 \leq j \leq d} C_j \right) \cdot \left(\min_{1 \leq j \leq d} |k_j|^{-m-2} \right) \\ &= \left(\max_{1 \leq j \leq d} C_j \right) \cdot \left(\max_{1 \leq j \leq d} |k_j| \right)^{-m-2} \\ &\leq \left(\max_{1 \leq j \leq d} C_j \right) \cdot \left(\|\mathbf{k}\| / \sqrt{d} \right)^{-m-2} \\ &= \frac{C}{\|\mathbf{k}\|^{m+2}}, \end{aligned}$$

where $C = \max_{1 \leq j \leq d} C_j \cdot d^{1+m/2} > 0$. This implies the conclusion of the theorem. \square

Now, we prove Theorem 1.

PROOF. In this proof, for the notational simplicity, we use u, v, Ω instead of u_j, v_j, Ω_j , which will not generate any confusion. Since Ω is a rectangle in \mathbb{R}^d , it is easy to convert it to a unit cube $\mathbb{T}^d = [0, 1]^d$ by rescaling and translation, which only changes the constants in the following argument. Therefore, we assume that $\Omega = \mathbb{T}^d$ below. Let us now consider the residual $v(\mathbf{x})$ over Ω . Let $\mathbf{x} = (x_1, \dots, x_d) \in [0, 1]^d$, and $\boldsymbol{\sigma} = (\sigma_1, \dots, \sigma_d)$ be a d -dimensional vector whose entries consist of only 1 or -1 . Now, the odd extension $\tilde{v}(\mathbf{x})$ of $v(\mathbf{x})$ to the extended cube $\tilde{\Omega} \triangleq [-1, 1]^d$ can be defined as follows:

$$\tilde{v}(\sigma_1 \cdot x_1, \dots, \sigma_d \cdot x_d) \triangleq \left(\prod_{k=1}^d \sigma_k \right) \cdot v(x_1, \dots, x_d), \quad (x_1, \dots, x_d) \in [0, 1]^d.$$

The boundary conditions of $u(\mathbf{x})$ are:

$$\begin{aligned} u|_{\partial\Omega} &= f|_{\partial\Omega}, \\ \frac{\partial^2 u}{\partial \nu^2}|_{\partial\Omega} &= \frac{\partial^2 f}{\partial \nu^2}|_{\partial\Omega}, \\ &\vdots \\ \frac{\partial^{2m-2} u}{\partial \nu^{2m-2}}|_{\partial\Omega} &= \frac{\partial^{2m-2} f}{\partial \nu^{2m-2}}|_{\partial\Omega}. \end{aligned}$$

Therefore, these imply the following in terms of \tilde{v} :

$$\tilde{v}|_{\partial\Omega} = 0, \tag{A.3}$$

$$\partial_j^{2k} \tilde{v}(x_1, \dots, x_{j-1}, -1, x_{j+1}, \dots, x_d) = \partial_j^{2k} \tilde{v}(x_1, \dots, x_{j-1}, 1, x_{j+1}, \dots, x_d) = 0, \tag{A.4}$$

$$\partial_j^{2k} \tilde{v}(x_1, \dots, x_{j-1}, 0-, x_{j+1}, \dots, x_d) = \partial_j^{2k} \tilde{v}(x_1, \dots, x_{j-1}, 0+, x_{j+1}, \dots, x_d) = 0, \tag{A.5}$$

$$\partial_j^{2k+1} \tilde{v}(x_1, \dots, x_{j-1}, -1, x_{j+1}, \dots, x_d) = \partial_j^{2k+1} \tilde{v}(x_1, \dots, x_{j-1}, 1, x_{j+1}, \dots, x_d), \tag{A.6}$$

$$\partial_j^{2k+1} \tilde{v}(x_1, \dots, x_{j-1}, 0-, x_{j+1}, \dots, x_d) = \partial_j^{2k+1} \tilde{v}(x_1, \dots, x_{j-1}, 0+, x_{j+1}, \dots, x_d), \tag{A.7}$$

where $j = 1, \dots, d$ and $k = 0, \dots, m-1$. Note that $k = 0$ in Equation (A.4) means $\tilde{v}|_{\partial\tilde{\Omega}} = 0$, which is different from (A.3). Equations (A.3)–(A.7) imply that $\tilde{v} \in C^{2m-1}(\tilde{\Omega})$ and *periodic*. Therefore, we can invoke Theorem 7 with m replaced by $2m-1$ so that we get:

$$\hat{f}(\mathbf{k}) = O\left(\|\mathbf{k}\|^{-2m-1}\right) \quad \text{for } \mathbf{k} \neq \mathbf{0}. \quad \square$$