The Continuous Boundary Local Fourier Transform

Brons Larson and Naoki Saito

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

ABSTRACT

The Local Fourier Transform (LFT) provides a nice tool for concentrating both a signal and its Fourier transform. But there are certain properties of this algorithm that make it unattractive for various applications. In this paper, some of these disadvantages are explored, and a new approach to localized Fourier analysis is proposed, the *continuous boundary local Fourier transform* (CBLFT), which attempts to correct some of these shortcomings. Results ranging from segmentation to representation cost to compression are also presented.

Keywords: Local Fourier transform, local cosine transform, best basis, sparse representation, segmentation, compression

1. INTRODUCTION

It is a well known fact that the visual world is periodic. From man-made objects such as shingled roofs, brick walls or striped clothing, to natural or biological objects such as trees, grass or blown sand, periodicities abound at all scales. It is also well known from neuroscience that mammalian vision performs a spatial frequency filtering of the visual information in a localized, patch-by-patch manner. The main benefit of this type of representation is that visual information is more efficiently condensed, decreasing the loss of crucial information. Because of this, much recent effort has been devoted to developing mathematical models and transform algorithms that mimic this type of behavior. Some of the more successful approaches have been discrete cosine transform (DCT) used in the JPEG compression standard,[1] discrete wavelet transforms (DWT), wavelet packet transforms (WPT), local trigonometric transforms (LTT),[2] local Fourier transforms (LFT),[2] and the brushlet transform (BT).[3]

The main advantage of these methods is that they provide a good tool for concentrating both a signal and its frequency content. In addition, the LFT and BT are equipped with a phase. But all of these approaches suffer from a crucial problem: how to effectively periodize the initial signal. Without any information outside of the original interval, there lacks an effective way to both treat the boundary and recover the original signal from the transformed coefficients. Furthermore, instability of the folding process on or near signal discontinuities can increase edge effect which translates recursively to all subsequent levels of the hierarchy in the transforms. This can result in inefficient representation, improper segmentation and incorrect analysis of the signal.

It is with this in mind that a new approach to localized analysis is proposed, the Continuous Boundary Local Trigonometric Transform (CBLTT). Closely related to the LTT, it attempts to correct these and other shortcomings of the LTT. The main difference between the LTT and the CBLTT is that the former projects the signal onto smooth overlapping basis functions, whereas the latter decomposes the signal into a set of completely disjoint regions without any overlapping. Each disjoint subspace then undergoes an invertible, nonlinear transformation to reduce edge effect, and it is immediately expanded into an orthonormal basis; e.g., the Fourier basis, the sine/cosine basis, or the wavelet basis, etc.. As a tensor product, it can be applied to multi-dimensional data: signals, images, video. Since this new approach can be used to efficiently segment the frequency domain of a signal, it also gives rise to a new version of the BT.

In order to reduce edge effect and increase numerical stability, many variations of this new scheme have been derived. But due to page limitation, only one version, the *continuous periodic local Fourier transform*, is presented here. A more in-depth analysis of this method and its applications, as well as the other variations can be found in.[4]

Further author information: E-mail: larson@math.ucdavis.edu

2. PROBLEMS WITH THE LOCAL FOURIER TRANSFORM

In this section, some of the shortcomings of the LFT are presented. Although similar problems also affect the LCT, BT, DWT and WPT to varying degrees, only the problems specific to the LFT are illustrated here for simplicity. These descriptions provide motivation for the CBLFT, which is introduced in Section 3 and described in detail. In order to better understand the CBLFT, a few simple operations are needed and are defined next.

2.1. Smooth Orthogonal Periodization

A primary component of the LFT involves smoothly restricting a function to an interval and then periodizing it, thus allowing it to be expanded into the Fourier basis with minimal edge effect. In other words, suppose a function $x(t) \in L^2(\mathbb{R})$ is to be split smoothly into pieces, each of which is supported on an interval I_k (with some overlap), where $\bigcup_{k \in \mathbb{Z}} I_k = \mathbb{R}$, and I_k's are disjoint. Define x_I to be an *I-periodic extension* of $x \in L^2(I)$ such that

$$x_I(t) \triangleq \sum_{k \in \mathbb{Z}} x(t - k|I|). \tag{1}$$

To achieve this goal, Wickerhauser[2] introduced the following *smooth orthogonal periodization* from $L^2_{loc}(\mathbb{R})$ into an I_k-periodic extension of $L^2(I_k)$:

$$T_{I_k} x(t) \triangleq W^*_{I_k} \mathbf{1}_{I_k} U_{I_k} x(t). \tag{2}$$

This operator consists of three operators: U_{I_k} (*unitary folding operator*), $\mathbf{1}_{I_k}$ (restriction operator, i.e., $\mathbf{1}_{I_k} x(t) = x(t)$ if $t \in I_k$, $= 0$ otherwise), and $W^*_{I_k}$ (*periodized unfolding operator*). To define the unitary folding operator U_{I_k} for the interval I_k, let $I_k = (\alpha_k, \alpha_{k+1})$. Then

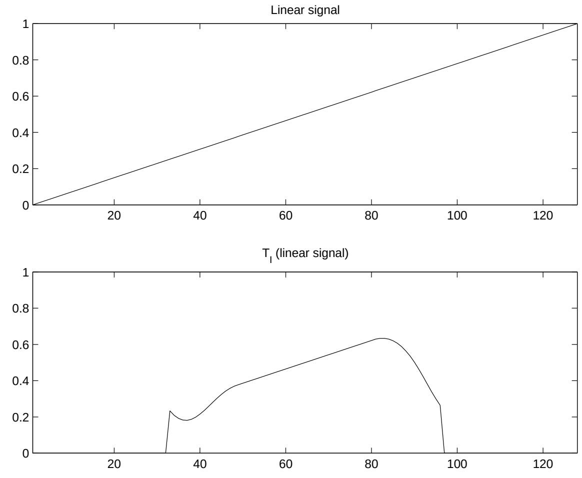

Figure 1. The bottom plot shows the application of $T_{I,k}$ to the linear function $x(t) = t$ which is shown in the top plot. Notice how the signal is nicely restricted and periodized.

$$U_{I_k} x(t) \triangleq U(r_k, \alpha_k, \epsilon_k) U(r_{k+1}, \alpha_{k+1}, \epsilon_{k+1}) x(t), \tag{3}$$

where $U(r, \alpha, \epsilon)$ is a unitary folding operator associated with the *action region* $t \in (\alpha - \epsilon, \alpha + \epsilon)$ and is defined as

$$U(r, \alpha, \epsilon) x(t) \triangleq \begin{cases} r\left(\frac{t-\alpha}{\epsilon}\right) x(t) + r\left(\frac{\alpha-t}{\epsilon}\right) x(2\alpha - t) & \text{if } \alpha < t < \alpha + \epsilon, \\ r\left(\frac{\alpha-t}{\epsilon}\right) x(t) - r\left(\frac{t-\alpha}{\epsilon}\right) x(2\alpha - t) & \text{if } \alpha - \epsilon < t < \alpha, \\ x(t) & \text{otherwise.} \end{cases} \tag{4}$$

The function $r(t)$ above is called a *rising cutoff function*, which is a smooth version (e.g., $r \in C^d(\mathbb{R})$ with $d \in \mathbb{N}$) of the Heaviside step function satisfying the following condition:

$$|r(t)|^2 + |r(-t)|^2 = 1 \quad \text{for all } t \in \mathbb{R}, \quad \text{and} \quad r(t) = \begin{cases} 0 & \text{if } t \leq -1, \\ 1 & \text{if } t \geq 1. \end{cases} \tag{5}$$

A typical example of $C^1(\mathbb{R})$ is the following iterated sine function which will be employed throughout the remainder of this paper:

$$r(t) = \begin{cases} 0, & \text{if } t \leq -1, \\ \sin[\frac{\pi}{4}(1 + \sin\frac{\pi}{2}t)], & \text{if } |t| < 1, \\ 1 & \text{if } t \geq 1. \end{cases} \tag{6}$$

The *periodized unfolding operator* $W^*_{I_k}$ in Equation (2) is defined as:

$$W^*_{I_k}x(t) = W^*(r_k, I_k, \epsilon_k)x(t) = \begin{cases} r_k(\frac{t-\alpha}{\epsilon_k})x(t) - r_k(\frac{\alpha_k - t}{\epsilon_k})x(\alpha_k + \alpha_{k+1} - t) & \text{if } \alpha_k < t < \alpha_k + \epsilon_k, \\ r_k(\frac{\alpha_{k+1} - t}{\epsilon_k})x(t) + r_k(\frac{t - \alpha_{k+1}}{\epsilon_k})x(\alpha_k + \alpha_{k+1} - t) & \text{if } \alpha_{k+1} - \epsilon_k < t < \alpha_{k+1}, \\ x(t) & \text{otherwise.} \end{cases} \tag{7}$$

An important feature of T_{I_k} is the fact that it preserves the smoothness of a smooth function; that is, if $x \in C^d(\mathbb{R})$, then $T_{I_k}x$ is an I_k-periodic extension that also belongs to $C^d(\mathbb{R})$. Figure 1 shows the application of T_{I_k} to a linear function $x(t) = t$.

2.2. Instability of the Folding Process

One of the main drawbacks of the LFT derives from the instability of the folding procedure on or near a jump in the signal. Discontinuities at folding locations cause improper boundary conditions within a subspace, often times resulting in increased representation cost. Furthermore, once a subspace fails to achieve the proper boundary conditions, the problem repeats itself recursively in children subspaces, resulting in improper or unpredictable partitioning (see Figure 2). In other words, it is the folding operation itself, rather than the dataset, that is the dominating factor in the resulting basis partition pattern and overall representation cost. In addition, since there does not exist an attractive, invertible folding process for the top level, then this problem is present in every subspace and every level.

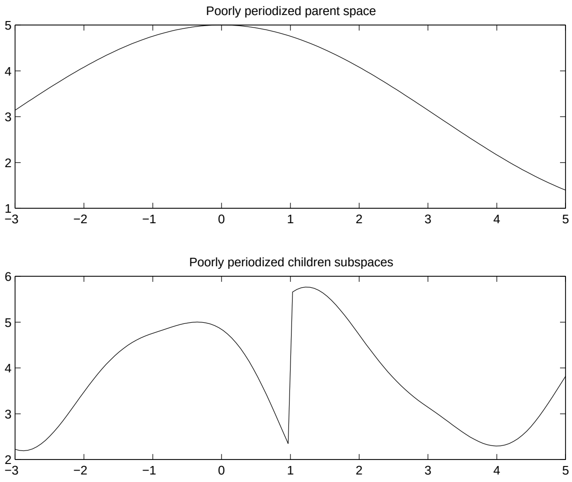

Figure 2. If the parent space is not properly periodized (top figure), then the children subspaces will not properly periodize (bottom figure). This error propagates recursively to lower levels of the decomposition.

Evidence of this behavior can also be seen when switching between cosine and sine polarities (see Wickerhauser[2] for details on polarity). Using the same dataset, completely different basis partition patterns arise when the polarity is switched. Figure 3 illustrates this fact. The folding operation seems to be the determining factor in the choice of basis (or partitioning), rather than the underlying structure in the image.

2.3. Mixing Information Across Boundaries

Another problem arises because information within subspaces is folded across boundaries; that is, basis functions have global support. Although this can produce a desirable effect when reconstructing compressed signals by reducing the blocking effect between subspace boundaries, it also can be detrimental for various applications such as texture

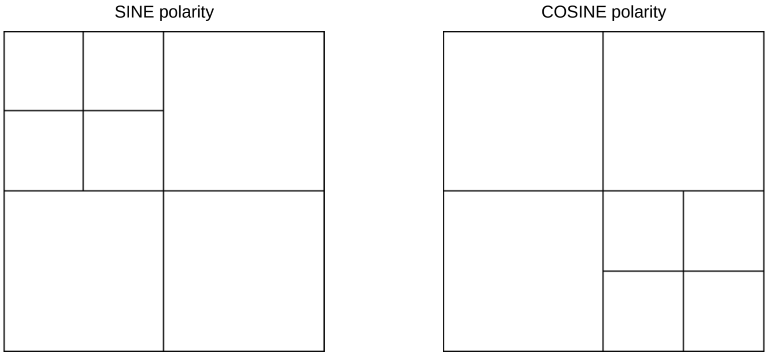

Figure 3. Basis partition patterns of the same image using the real valued local Fourier transform. The pattern on the left was chosen when sine polarity was used, and the pattern on the right was chosen when cosine polarity was used.

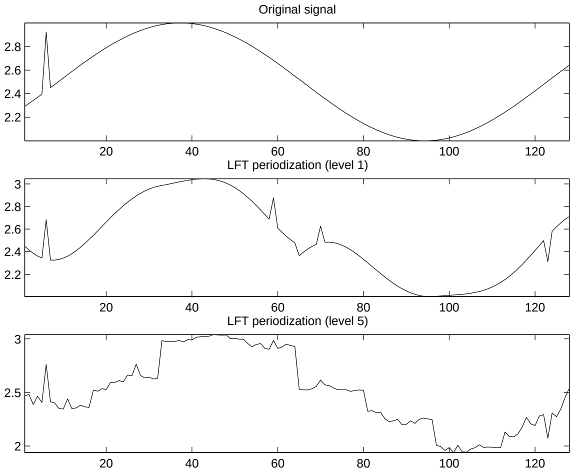

Figure 4. LFT periodization causes the sharp bump in the parent space to be mixed into many locations of the children subspaces.

segmentation and signal analysis. This becomes particularly problematic for the LFT and BT since they require the additional "periodized unfolding" operations (equation (7)) which are not needed for the LCT.

As can be seen in Figure 4, a structure such as the sharp bump which is localized to one region of the parent space, is mixed within all children subspaces when using LFT periodization. By the time that bottom level of the decomposition is reached, the bump is mixed to every region of the signal; in essence, almost all recognizable structure has been lost due to the mixing of information across subspace boundaries.

In response to the problems described above, an approach was devised to remedy these and other shortcomings. In particular, the algorithm satisfies the following constraints:

- It is invertible,
- It operates on all subspaces including the top level,
- It is stable, preserving continuity at the subspace boundaries,
- It is computationally efficient,
- It is an isometry for use in the best basis algorithm.

One possible solution is presented in the next section.

3. THE CONTINUOUS BOUNDARY LOCAL FOURIER TRANSFORM

This section describes one version of the CBLFT. Although there are many different variations of the algorithm, each with its own strengths and weaknesses, only one which is based on a continuous periodic extension is presented here due to page limitation. The other forms of the CBLFT are detailed and analyzed in.[4]

3.1. The Continuous Periodic Local Fourier Transform

The basic approach is to force continuity at the boundary of each subspace by creating an artificial extension to be used for folding. The main problem is that an invertible transformation is often difficult to find. Some simple ideas include even and odd extensions. They suffer from a host of problems, though, which are described in.[4] The following scheme was devised to satisfy the above conditions, while also solving the information mixing problem described in section 2.3. The idea is to use a periodic extension of each subspace, while also forcing continuity at the boundary. Figure 5 illustrates the approach. To begin, the right hand side of the signal is extended in a continuous periodic fashion, and folding is performed at the right hand boundary. The extension is discarded and the energy of the folded right half is computed to find how much energy has been lost or gained. This is then repeated for the opposite side, but with a temporary shift in both the signal and extension prior to folding. The value of the shift, which depends on the signal, can be precomputed and is used to preserve the isometry. Immediately afterwards, periodized unfolding is performed to periodize the subspace, as well as undo any mixing which occurred during the folding procedure. Figure 6 shows the results of this periodization process.

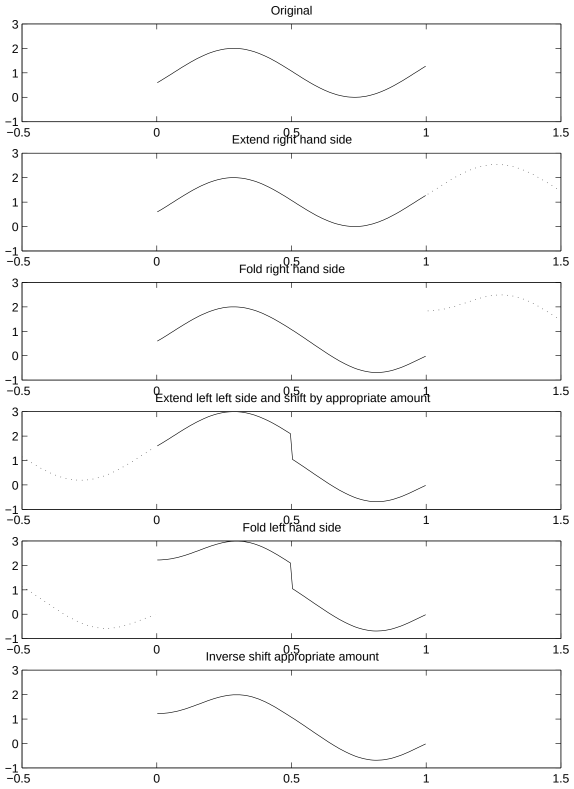
Original
Extend right hand side
Fold right hand side
Extend left left side and shift by appropriate amount
Fold left hand side
Inverse shift appropriate amount

Figure 5. Isometric folding using a continuous periodic extension. Notice that both the left and right hand sides are extended in a continuous periodic manner, but the left hand side is temporarily shifted prior to folding in order to preserve the isometry.

Notice that the continuous periodic extension successfully periodizes the two children subspace, while also minimizing the mixing of information. It should be noted that this method can be used for any subspace, including the top level. And because of the isometry, it can be cast in a best basis setting to perform a stable adaptive analysis of a signal.

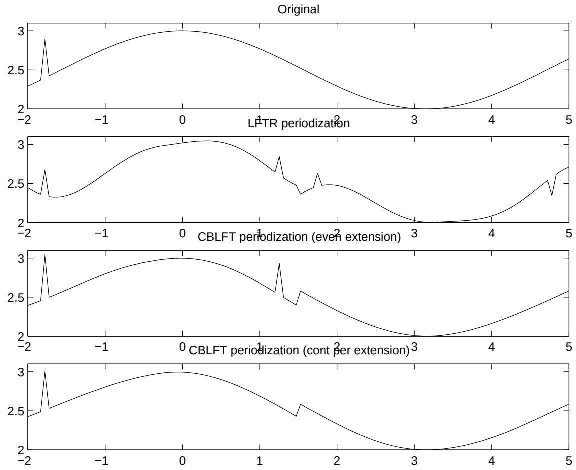

Figure 6. A comparison of various methods of periodization. The top figure shows the original function with a sharp bump at one location. The second figure shows the effects of splitting the space into two children via the local Fourier transform. The third figure shows splitting via an even extension (described in[4]). And the fourth figure shows the results of splitting the space using the continuous periodic extension.

3.2. Continuous Periodic Extension and Folding

Mathematically, the approach for folding is formulated as follows. Let x_n be a function of $N+1$ discrete points indexed from 0 to N. Let $x_0 \neq x_N$, i.e., there a discontinuity at the boundary if the function is periodized. For simplicity, let $R =$ *the radius of the rising cutoff function* such that $R \leq \frac{N+1}{2}$, and let $n = 0, \ldots, R-1$ in all of the following formulas.

Then one version of the discrete continuous periodic extension can be constructed according to Figure 7 using *midpoint folding* (see[2]). With this arrangement, the values of the extension to the left of x_0 are given by

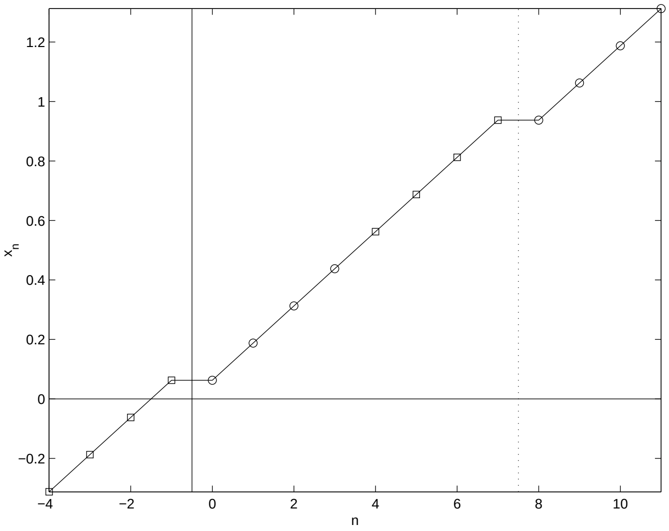

Figure 7. Example of a continuous periodic extension of a linear function with midpoint folding.[2] In this example, $N = 7$.

$$x_{-n-1} = x_{N-n} + x_0 - x_N \tag{8}$$

and the values of the extension to the right of x_N are

$$x_{N+n+1} = x_n + x_N - x_0. \tag{9}$$

Using Equation (4), folding at the left hand edge is defined as

$$\begin{aligned} \tilde{x}_n &= r_n x_n + r_{-n-1} x_{-n-1} \\ &= r_n x_n + r_{-n-1}(x_{N-n} + x_0 - x_N) \end{aligned} \tag{10}$$

and at the right hand edge as

$$\begin{aligned} \tilde{x}_{N-n} &= r_n x_{N-n} - r_{-n-1} x_{N+n+1} \\ &= r_n x_{N-n} - r_{-n-1}(x_n + x_N - x_0). \end{aligned} \tag{11}$$

Now, for the isometric version, the generalized isometric folding formula for the left hand side can be found using Equation (10) and applying the shift operation with shift, $s = s(x_n)$, to get

$$\begin{aligned} \tilde{x}_{s,n} &= r_n(x_n + s) + r_{-n-1}(x_{-n-1} + s) - s \\ &= r_n x_n + r_{-n-1} x_{-n-1} + s(r_n + r_{-n-1} - 1) \\ &= \underbrace{r_n x_n + r_{-n-1}(x_{N-n} + x_0 - x_N)}_{\tilde{x}_n} + s\,\underbrace{(r_n + r_{-n-1} - 1)}_{\lambda_n}. \end{aligned} \tag{12}$$

where $\tilde{x}_{s,n}$ stands for the folded result with shift. Thus, this is just the standard folding operation, with the added term, $s\lambda_n$.

With isometric folding well defined, a formula for finding an explicit shift value, s, can be easily derived by using the above notation along with the definition of an isometry to get

$$\begin{aligned} 0 &= \sum_{n=0}^{R-1} \left[(\tilde{x}_n + s\lambda_n)^2 + \tilde{x}_{N-n}^2 \right] - \sum_{n=0}^{R-1} \left(x_n^2 + x_{N-n}^2 \right) \\ \Leftrightarrow 0 &= \sum_{n=0}^{R-1} \left(\tilde{x}_n^2 + 2s\tilde{x}_n\lambda_n + s^2\lambda_n^2 + \tilde{x}_{N-n}^2 - x_n^2 - x_{N-n}^2 \right) \\ \Leftrightarrow 0 &= s^2 \sum_{n=0}^{R-1} \lambda_n^2 + 2s \sum_{n=0}^{R-1} \tilde{x}_n\lambda_n + \sum_{n=0}^{R-1} \left(\tilde{x}_n^2 + \tilde{x}_{N-n}^2 - x_n^2 - x_{N-n}^2 \right) \\ \Leftrightarrow s &= \frac{-\sum_{n=0}^{R-1} \tilde{x}_n\lambda_n \pm \sqrt{\left(\sum_{n=0}^{R-1} \tilde{x}_n\lambda_n \right)^2 - \sum_{n=0}^{R-1} \lambda_n^2 \cdot \sum_{n=0}^{R-1} \left(\tilde{x}_n^2 + \tilde{x}_{N-n}^2 - x_n^2 - x_{N-n}^2 \right)}}{\sum_{n=0}^{R-1} \lambda_n^2}. \end{aligned} \tag{13}$$

It should be noted that most of the values in Equation (25) are constants that can be precomputed only once; hence, there is not a lot of computational overhead involved with this isometric operator.

3.3. Continuous Periodic Extension and Unfolding/Inversion

Solving for x in Equations (10) and (11) yields the following formula for inverse folding at the left hand side

$$x_n = \left[\tilde{x}_n - r_{-n-1}(x_{N-n} + x_0 - x_N) \right] / r_n \tag{14}$$

and at the right hand side

$$x_{N-n} = \left[\tilde{x}_{N-n} + r_{-n-1}(x_n + x_N - x_0) \right] / r_n. \tag{15}$$

Now, similar to many of the methods described in,[4] this approach yields implicit inversion formulas. Represented as a linear system, it can be written as

$$\begin{bmatrix} r_0 + r_{-1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ r_{-2} & r_1 & 0 & 0 & 0 & 0 & r_{-2} & -r_{-2} \\ \downarrow & 0 & \searrow & 0 & 0 & \swarrow & 0 & \downarrow \\ r_{-\frac{N+1}{2}} & 0 & 0 & r_{\frac{N-1}{2}} & r_{-\frac{N+1}{2}} & 0 & 0 & -r_{-\frac{N+1}{2}} \\ r_{-\frac{N+1}{2}} & 0 & 0 & -r_{\frac{N+1}{2}} & r_{\frac{N-1}{2}} & 0 & 0 & -r_{-\frac{N+1}{2}} \\ \downarrow & 0 & \swarrow & 0 & 0 & \searrow & 0 & \downarrow \\ r_{-2} & -r_{-2} & 0 & 0 & 0 & 0 & r_1 & -r_{-2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & r_0 - r_{-1} \end{bmatrix} x = \tilde{x}. \tag{16}$$

In this system, it is assumed that $R = \frac{N+1}{2}$, although this is not necessary.

In order to invert this, sparse matrix routines can be employed, but for computational speed it is better to exploit the structure of the array and find an explicit solution. To do this, it is necessary to recover x_0 and x_N (see Equations (10) and (11)). They are easily found from the first and last rows of Equation (16) to be

$$x_0 = \frac{\tilde{x}_0}{r_0 + r_{-1}} \tag{17}$$

and

$$x_N = \frac{\tilde{x}_N}{r_0 - r_{-1}}. \tag{18}$$

Using these results, and exploiting the symmetry in Equation (16), generalized inversion formulas for $x_1, \dots, x_{N-1}$ are found by simultaneously solving a system of equations derived from rows n and $N-n$ for $n = 1, \dots, R-2$ in Equation (16). These two equations have two unknowns, and so the unique solution is found to be

$$\begin{aligned} x_n &= [\tilde{x}_n - r_{-n-1}(x_{N-n} + x_0 - x_N)]/r_n \\ &= (\tilde{x}_n - r_{-n-1}\{[\tilde{x}_{N-n} + r_{-n-1}(x_n + x_N - x_0)]/r_n + x_0 - x_N\})/r_n \\ &= [r_n\tilde{x}_n - r_{-n-1}\tilde{x}_{N-n} + r_{-n-1}(r_{-n-1} - r_n)(x_0 - x_N)]/\underbrace{(r_n^2 + r_{-n-1}^2)}_{=1} \end{aligned} \tag{19}$$

and

$$\begin{aligned} x_{N-n} &= [\tilde{x}_{N-n} + r_{-n-1}(x_n + x_N - x_0)]/r_n \\ &= (\tilde{x}_{N-n} + r_{-n-1}\{[\tilde{x}_n - r_{-n-1}(x_{N-n} + x_0 - x_N)]/r_n + x_N - x_0\})/r_n \\ &= [r_n\tilde{x}_{N-n} + r_{-n-1}\tilde{x}_n - r_{-n-1}(r_{-n-1} + r_n)(x_0 - x_N)]/\underbrace{(r_n^2 + r_{-n-1}^2)}_{=1}. \end{aligned} \tag{20}$$

For the isometry, the generalized isometric inversion formula for the left hand side is found using Equation (12) to be

$$x_n = [\tilde{x}_{s,n} - r_{-n-1}(x_{N-n} + x_0 - x_N)]/r_n - s\lambda_n/r_n. \tag{21}$$

Adding this to Equations (17) , (19) and (20) yields the following isometric inversion formulas

$$x_0 = \underbrace{\frac{\tilde{x}_{s,0}}{r_0 + r_{-1}}}_{z_0} - s\underbrace{\frac{\lambda_0}{r_0 + r_{-1}}}_{\alpha_0} \tag{22}$$

$$x_n = \underbrace{r_n\tilde{x}_{s,n} - r_{-n-1}\tilde{x}_{s,N-n} + r_{-n-1}(r_{-n-1} - r_n)\left(\frac{\tilde{x}_{s,0}}{r_0 + r_{-1}} - \frac{\tilde{x}_{s,N}}{r_0 - r_{-1}}\right)}_{z_n} - s\underbrace{\left(r_n\lambda_n + r_{-n-1}(r_{-n-1} - r_n)\frac{\lambda_0}{r_0 + r_{-1}}\right)}_{\alpha_n} \tag{23}$$

and

$$x_{N-n} = \underbrace{r_n\tilde{x}_{s,N-n} + r_{-n-1}\tilde{x}_{s,n} - r_{-n-1}(r_{-n-1} + r_n)\left(\frac{\tilde{x}_{s,0}}{r_0 + r_{-1}} - \frac{\tilde{x}_{s,N}}{r_0 - r_{-1}}\right)}_{z_{N-n}} - s\underbrace{\left[r_{-n-1}\left(\lambda_n + (r_{-n-1} - r_n)\frac{\lambda_0}{r_0 + r_{-1}}\right)\right]}_{\beta_n} \tag{24}$$

for $n = 1, \dots, R-1$. Notice that is just the standard inversion operation of Equations (19) and (20) applied to $\tilde{x}_{s,n}$, with the added terms α_n and β_n (also note that $\beta_0 = 0$).

Now that the inversion formulas are well defined, they too need to be used in a formula for finding an explicit shift value, s. Using the above notation along with the definition of an isometry yields

$$s = \frac{\sum_{n=0}^{R-1}(\alpha_n z_n + \beta_n z_{N-n}) \pm \sqrt{\left(\sum_{n=0}^{R-1}(\alpha_n z_n + \beta_n z_{N-n})\right)^2 - \sum_{n=0}^{R-1}(\alpha_n^2 + \beta_n^2)\cdot\sum_{n=0}^{R-1}\left(z_n^2 + z_{N-n}^2 - \tilde{x}_{s,n}^2 - \tilde{x}_{s,N-n}^2\right)}}{\sum_{n=0}^{R-1}(\alpha_n^2 + \beta_n^2)}. \tag{25}$$

As stated above, most of the values in Equation (25) are constants that can be precomputed only once; hence, there is not a lot of computational overhead involved with this isometric operator.

3.4. Existence of a Real Valued Shift

In light of Equation (13), it is important to know whether a real valued isometric shift always exists. The answer is no, even for nonnegative functions. A simple counterexample is $x = (1, 0, \dots, 0)$ where $N = 15$ and $R = 8$. The problem arises because the right hand extension is nonpositive, causing the energy of the folded right hand side to *increase*; that is, $\sum_{n=0}^{R-1} \left(\tilde{x}_{N-n}^2 - x_{N-n}^2\right) \geq 0$. In response to this result, it is natural to wonder whether it is possible to place a condition on x to cause Equation (13) to always be satisfied with a real valued s. The following theorem answers this question.

THEOREM 3.1. *[]For all real valued functions $x \in [0,1]$ and rising cutoff fuctions* (6)*, there exists a value $C \in \mathbb{R}$ such that for $y = x + C$ and $C \geq 0$, there exists a real valued shift, $s = s(y)$, which is the solution to Equation* (13)*.*

Proof. (*i*) $\sum_{n=0}^{R} \lambda_n^2 \neq 0$; i.e., the denominator of Equation 13 is nonzero. For this to be true, the following equivalent relationships need to be satisfied.

$$\begin{array}{rrcll}
 & \sum_{n=0}^{R-1} \lambda_n^2 & \neq & 0 & \\
\Leftrightarrow & \lambda_n & \neq & 0 & \text{for some } n \in [0, R-1] \\
\Leftrightarrow & r_n + r_{N-n} & \neq & 1 & \text{for some } n \in [0, R-1] \\
\Leftrightarrow & r_n + \sqrt{1 - r_n^2} & \neq & 1 & \text{for some } n \in [0, R-1] \\
\Leftrightarrow & \sqrt{1 - r_n^2} & \neq & 1 - r_n & \text{for some } n \in [0, R-1] \\
\Leftrightarrow & 1 - r_n^2 & \neq & 1 - 2r_n + r_n^2 & \text{for some } n \in [0, R-1] \\
\Leftrightarrow & 2r_n^2 - 2r_n & \neq & 0 & \text{for some } n \in [0, R-1] \\
\Leftrightarrow & r_n(r_n - 1) & \neq & 0 & \text{for some } n \in [0, R-1] (\star)
\end{array}$$

Now, using the fact that $\frac{1}{\sqrt{2}} < r_n < 1 \; \forall \; n \in [0, R-1]$, then $r_n(r_n - 1) < 0 \; \forall \; n \in [0, R-1]$ and $(\star)$ is satisfied;

(*ii*) $\left(\sum_{n=0}^{R-1} \tilde{y}_n \lambda_n\right)^2 - \sum_{n=0}^{R-1} \lambda_n^2 \cdot \sum_{n=0}^{R-1} \left(\tilde{y}_n^2 + \tilde{y}_{N-n}^2 - y_n^2 - y_{N-n}^2\right) \geq 0 \; (*)$; i.e., the determinant in Equation (13) is nonnegative.

To begin, it is clear that $(*)$ will always be true as long as $\sum_{n=0}^{R-1} \left(\tilde{y}_n^2 + \tilde{y}_{N-n}^2 - y_n^2 - y_{N-n}^2\right) \leq 0$. So it is necessary to understand when this occurs. To do this, consider the left and right parts separately. Starting with the right side, notice that $\sum_{n=0}^{R-1} \left(\tilde{y}_{N-n}^2 - y_{N-n}^2\right)$ will be maximized when y_{N-n}^2 is a minimum, and the right hand extension is also minimized. This occurs when $y_0 = C + 1, y_n = C$ for $n = 1, \dots, R-1$, and $y_{N-n} = C$ for $n = 0, \dots, R-1$. Using this yields

$$\begin{aligned}
0 &= \sum_{n=0}^{R-1} \left(\tilde{y}_{N-n}^2 - y_{N-n}^2\right) \\
\Leftrightarrow 0 &= \sum_{n=0}^{R-1} \left[r_n y_{N-n} - r_{-n-1}(y_n + y_N - y_0)\right]^2 - \sum_{n=0}^{R-1} y_{N-n}^2 \\
\Leftrightarrow 0 &= C^2 (r_0 - r_{-1})^2 + \sum_{n=1}^{R-1} \left[r_n - (C-1) r_{-n-1}\right]^2 - C^2 R \\
\Leftrightarrow 0 &= C^2 \sum_{n=0}^{R-1} \left[(r_n - r_{-n-1})^2 - 1\right] + 2C \sum_{n=1}^{R-1} \left[r_{-n-1}(r_n - r_{-n-1})\right] + \sum_{n=1}^{R-1} r_{-n-1}^2 \\
\Leftrightarrow C &= \frac{-\sum_{n=1}^{R-1} r_{-n-1}(r_n - r_{-n-1}) \pm \sqrt{\left[\sum_{n=1}^{R-1} r_{-n-1}(r_n - r_{-n-1})\right]^2 - \sum_{n=0}^{R-1}\left[(r_n - r_{-n-1})^2 - 1\right] \cdot \sum_{n=1}^{R-1} r_{-n-1}^2}}{\sum_{n=0}^{R-1}\left[(r_n - r_{-n-1})^2 - 1\right]}. \qquad (26)
\end{aligned}$$

Some values of C computed from Equation (26) are shown in Figure 8. As can be seen, as R increases, C converges to a number close to 0.8019. Thus, for any $x \in [0,1]$, and for all values of $C > 0.8019$, $\sum_{n=0}^{R-1} \left(\tilde{y}_{N-n}^2 - y_{N-n}^2\right) < 0$. In a similar manner, the above argument can be repeated for the left hand side, but using the function that maximizes $\sum_{n=0}^{R-1} \left(\tilde{y}_n^2 - y_n^2\right)$, namely $y_n = C + 1$ for $n = 0, \dots, R-1$, $y_{N-n} = C + 1$ for $n = 1, \dots, R-1$, and $y_N = C$. The results are

$$C = \frac{-\sum_{n=1}^{R-1} r_{-n-1}(r_n + r_{-n-1}) \pm \sqrt{\left[\sum_{n=1}^{R-1} r_{-n-1}(r_n + r_{-n-1})\right]^2 - \sum_{n=0}^{R-1}\left[(r_n + r_{-n-1})^2 - 1\right] \cdot \sum_{n=1}^{R-1} r_{-n-1}^2}}{\sum_{n=0}^{R-1}\left[(r_n + r_{-n-1})^2 - 1\right]} - 1. \qquad (27)$$

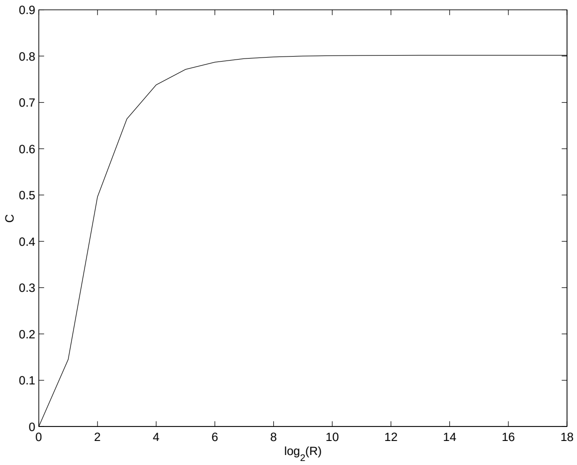

Figure 8. Values of C found using Equation (26) with $2^0 \leq R \leq 2^{18}$.

A plot of various values of C found using Equation (27) is shown in Figure 9. As R increases, C tends towards -1.1981. What this means is that the left hand side can always be shifted so that its energy before and after folding can be the same.

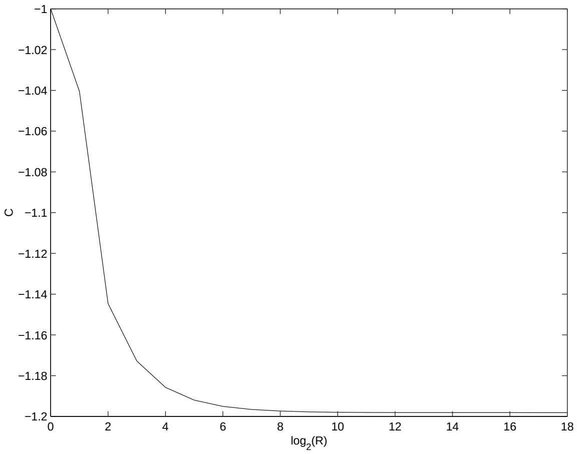

Figure 9. Values of C found using Equation (27) with $2^0 \leq R \leq 2^{18}$.

Thus, as long as $C > 0.8019$, then a shift, $s(y)$, will always exist. And the lowest value that the left hand side will ever be shifted is -1.1981. □

4. RESULTS

Results, ranging from segmentation to representation cost and compression are presented here. First, take Figure 10, which shows a comparison between the standard LFT and the CPLFT when *sparsity* among the transformed coefficients, $\hat{x}$, is optimized. It should be noted that $\lim_{p\downarrow 0} = ||\hat{x}||_p^p = \lim_{p\downarrow 0} \sum_i |\hat{x}_i|^p = ||\hat{x}||_0 = \#\{i : \hat{x}_i \neq 0\}$. Hence, the sparsity measure ℓ^p with $0 < p \leq 1$ is employed as a stable approximation of true sparsity. (See[5,6] for more about such sparsity measures). As can be seen in plots (a) and (b), CBLFT segments the signal into regions that match the structure of the underlying signal, whereas the LFT does not. In terms of cost, the sparsity measure is $\ell^{0.1} = 91.97$ for LFT, and $\ell^{0.1} = 76.43$ for CBLFT. This affects the compression results as can be seen in plots (c) and (d), which show the reconstructed signals after thresholding half of the transform coefficients. As can be seen, LFT produces some undesirable oscillations near the subspace boundaries. This artifact is not as prevalent in the CBLFT reconstruction.

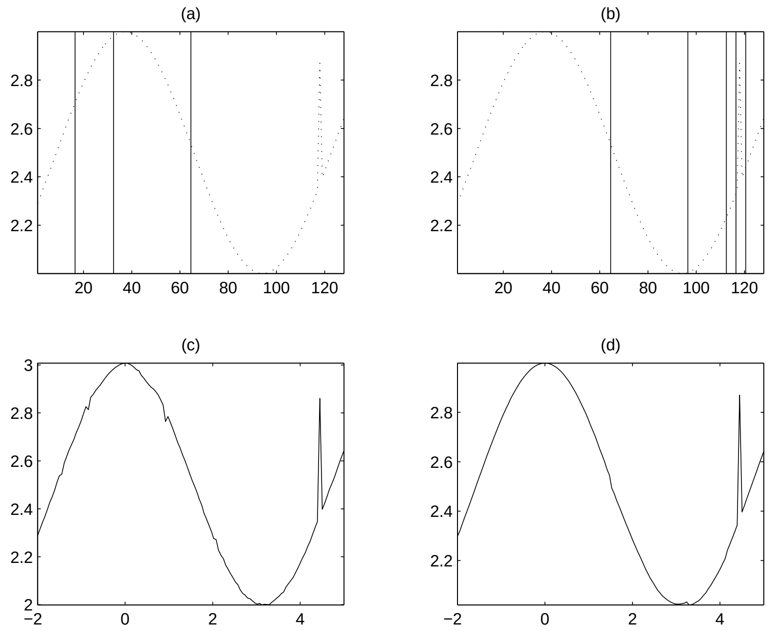

Figure 10. Sparsest basis partition pattern using (a) LFT with $\ell^{0.1}$, and (b) CBLFT. Compression results, 2:1, using (c) LFT, and (d) CBLFT.

Another important property of CBLFT is the stability of partition patterns under various conditions. For example, Figure 11 shows the results of using different sparsity measures, ℓ^p, as well as shifting an image. As can be seen, the partition patterns tend to be more stable for CBLFT. This is not the case when using LFT, LCT or BT.

Figure 12 shows the effect that image rotation has on the basis partition pattern. Notice that the partition pattern tends to rotate along with the rotation of the image. None of these results were witnessed when the test was repeated using LFT, LCT or BT.

5. CONCLUSION

Although there is a lot more to be learned about this approach, preliminary results seem promising. The overall ability to minimize the mixing of information within subspaces while still reducing edge effect lends itself nicely to signal and image analysis. The stability of the algorithm under various sparsity measures, as well as the robustness under shifts and rotations shows promise for certain applications such as textured segmentation. In addition, numerous variations of this idea have been proposed and analyzed in depth and will be presented in future papers.

REFERENCES

1. K. Sayood, *Introduction to Data Compression*, Morgan Kaufman Publishers, Inc., San Francisco, CA, 2nd ed., 2000.
2. M. V. Wickerhauser, *Adapted Wavelet Analysis from Theory to Software*, A K Peters, Ltd., Wellesley, MA, 1994. with diskette.
3. F. G. Meyer and R. R. Coifman, “Brushlets: a tool for directional image analysis and image compression,” *Appl. Comput. Harmonic Anal.* **4**, pp. 147–187, 1997.
4. B. Larson, *The Continuous Boundary Local Fourier Transform.* PhD thesis, University of California, Davis, 2001.
5. N. Saito, B. M. Larson, and B. Bénichou, “Sparsity and statistical independence from a best-basis viewpoint,” in *Wavelet Applications in Signal and Image Processing VIII*, A. Aldroubi, A. F. Laine, and M. A. Unser, eds., vol. Proc. SPIE 4119, pp. 474–486, 2000. Invited paper.
6. B. Bénichou and N. Saito, “Sparsity vs. statistical independence in adaptive signal representations: A case study of the spike process,” *Ann. Inst. Statist. Math.* , 2001. submitted.

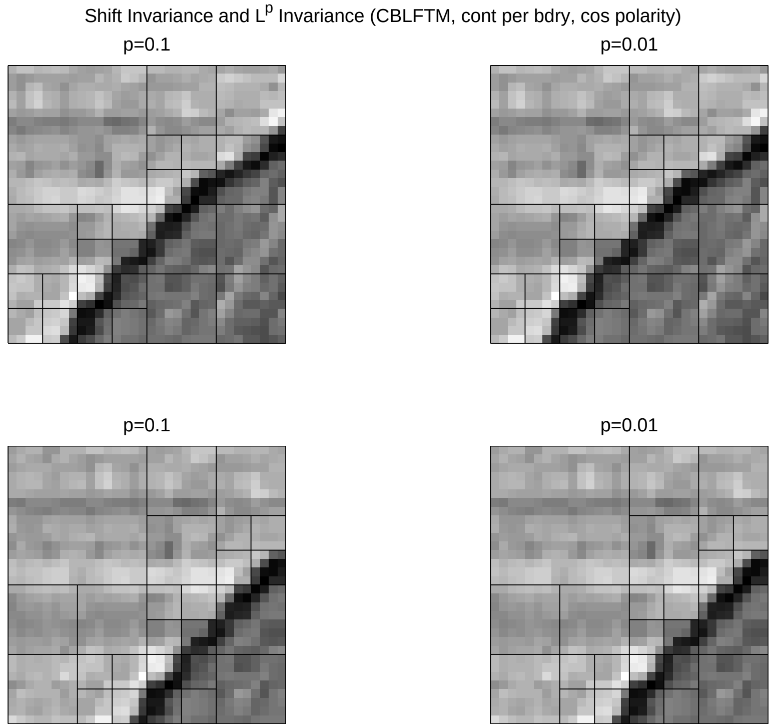
Shift Invariance and L^p Invariance (CBLFTM, cont per bdry, cos polarity)
p=0.1
p=0.01
p=0.1
p=0.01

Figure 11. Comparing the left two plots to the right two plots shows the robustness and stability of the partition pattern under two different sparsity measures ($\ell^{0.1}$ versus $\ell^{0.01}$) using CBLFT. Comparing the top two plots to the bottom two plots shows the robust and stable partition pattern under a shift using CBLFT.

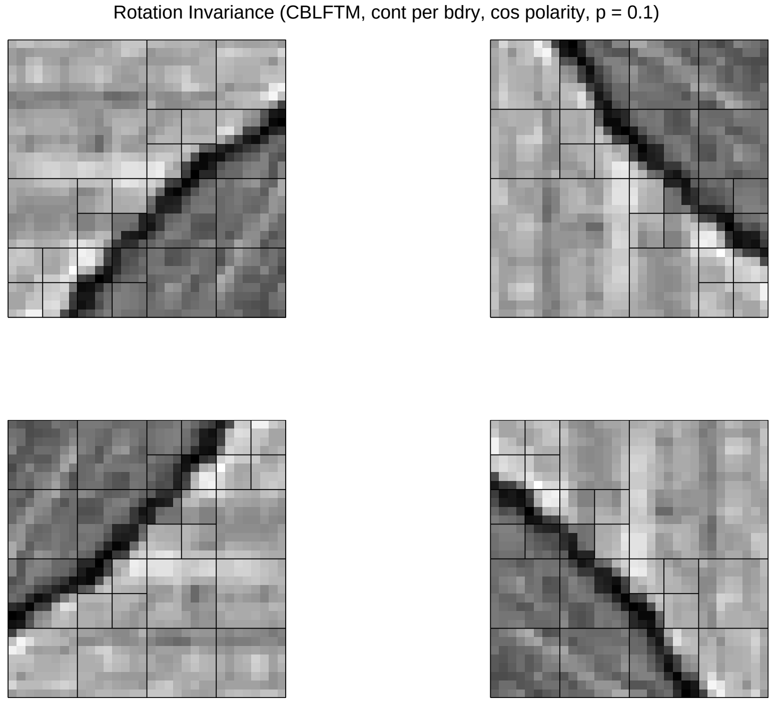
Rotation Invariance (CBLFTM, cont per bdry, cos polarity, p = 0.1)

Figure 12. Robust partition patterns under a rotation of the image using CBLFT.