New Flexible Segmentation Technique in Seismic Data Compression Using Local Cosine Transform

Ru-Shan Wu¹ and Yongzhong Wang¹

Institute of Tectonics, University of California, Santa Cruz, CA 95064

ABSTRACT

Best-basis searching algorithm based on binary (in general, M-ary) segmentation was constructed by Coifman and Wickerhauser in 1992 [1] (IEEE Trans. on Information Theory, 38, 713-718). However, there are several problems with the binary scheme. First, the binary segmentation is inflexible in grouping signals along the axis. Secondly, the binary-based segmentation method is very sensitive to time/space shifts of the original signal, such that the resulted best-basis will change a great deal if the signal is shifted by some samples. Thirdly, the reconstruction distortion after compression is relatively strong. In this paper, we design a new flexible segmentation algorithm with arbitrary time/space segmentation resolution which addresses the above-mentioned problems caused by the binary segmentation scheme. This new flexible segmentation algorithm is applied to 2-D seismic data compression with two semi-adaptive schemes: Flexible 2-D time-ALCT (Adapted Local Cosine Transform) and Flexible 2-D space-ALCT. From our numerical tests on both synthetic signals and real seismic data using local cosine transform, the advantages of this new flexible segmentation technique over the binary searching scheme can be easily seen, from overcoming the constraint of dyadic segmentations, reducing time/space-shift sensitivity, less reconstruction distortions to superior performance in seismic data compression.

Keywords: Segmentation, Flexible, Binary Tree, Best-Basis, Local Cosine Transform, Seismic Data Compression

1 INTRODUCTION

Orthogonal transforms play a key role in data compression schemes, such as discrete cosine transform, discrete wavelet transform and adapted wavelet-packet transform. Since seismic signals are always nonstationary in time and/or space, compression schemes with flexible segmentations in time/space to match the characteristics of nonstationary signals are highly desirable. Much progress has been achieved in signal processing and seismic data compression in the applications of (Adapted) Local Cosine Transform (ALCT) [1-9]. However, the existing methods in the literature are all based on dyadic decomposition trees in selecting best-basis. There are several problems for the binary scheme. First, the binary time/space segmentation has no flexibility at all. For example, if the peak of a pulse just rides on the midpoint of the whole signal, the binary segmentation scheme will result in either separating the pulse into two halves from the peak, or keeping the whole signal as only one segment. Undoubtedly, neither segmenting is satisfactory. Secondly, the binary-based segmentation method is very sensitive to time/space shifts of the original signal, such that the resulted best-basis will change a great deal if the signal is shifted by some samples. Due to these reasons, the reconstruction distortion after compression is relatively strong. In this paper, we design a new flexible segmentation algorithm which addresses the above-mentioned
problems caused by the binary segmentation scheme. From our numerical tests on both synthetic signals and real seismic data using local cosine transform, the advantages of this new flexible segmentation technique over the binary searching scheme can be easily seen.

The rest of this paper is organized as follows: In Section II, we will briefly review the binary local cosine basis. The new flexible time/space segmentation algorithm will be described in detail in Section III. In Section IV, some numerical tests on both synthetic signals and real seismic data are given. Finally, we draw the conclusions in Section V.

2 BINARY LOCAL COSINE BASIS

2.1 Local cosine basis

Local cosine bases constructed by Coifman and Meyer [2-3] consist of cosines multiplied by smooth, compactly supported bell functions. These localized cosine functions remain orthogonal and have a small Heisenberg product. The local cosine transform has much in common with the windowed Fourier Transform [10]. However, for the latter, the Balian-Low obstruction prevents the windowed exponential bases from simultaneously being a frame and having finite Heisenberg product. Local cosine bases, on the other hand, overcame this limitation. Suppose a sequence \( \{\alpha_j\} \) be selected to satisfy \( \alpha_j < \alpha_{j+1} \), \( \lim_{j \to \pm \infty} \{\alpha_j\} = \pm \infty \) and there also exists an accompanying sequence \( \{\epsilon_j\} \) such that \( \alpha_j + \epsilon_j \leq \alpha_{j+1} - \epsilon_{j+1} \) for all \( j \in \mathbb{Z} \). Then local cosine functions as

\[
\psi_{jk}(x) = \sqrt{\frac{2}{\alpha_{j+1} - \alpha_j}} b_{\alpha_j, \alpha_{j+1}}(x) \cos \left( \frac{x}{2} \left( k + \frac{1}{2} \right) \right)
\]

(1)

\((j \in \mathbb{Z}, k = 0, 1, 2, \ldots)\) with positive polarity at \( \alpha_j \) and negative polarity at \( \alpha_{j+1} \) form an orthonormal basis for the space \( L^2(R) \). The basis element in Eq. (1) can be characterized by position \( \alpha_j \), interval \( I_j = [\alpha_j, \alpha_{j+1}] \), and frequency/wavenumber index \( k \). \( b = b_{\alpha_j}(x) \) is a bell function which is a smooth function compactly supported in the interval \( [\alpha_j, \alpha_{j+1}] \). This interval contains \( I_j \) which is called its nominal support. We can control the nominal window width by \( I_j \), the left endpoint of the nominal window by position \( \alpha_j \), and the frequency/wavenumber by index \( k \). The properties of the bell function, i.e.,

\[
\begin{align*}
 b_{I_j}(x)^2 + b_{I_j}(2\alpha_j - x)^2 & = 1, x \in [\alpha_j - \epsilon_j, \alpha_j + \epsilon_j], \\
b_{I_j}(x)^2 + b_{I_j}(2\alpha_{j+1} - x)^2 & = 1, x \in [\alpha_{j+1} - \epsilon_{j+1}, \alpha_{j+1} + \epsilon_{j+1}], \\
b_{I_j}(x) & = 1, x \in (\alpha_j + \epsilon_j, \alpha_{j+1} - \epsilon_{j+1}).
\end{align*}
\]

(2)

insure the orthonormality of the bases and make a fast algorithm possible.

Two-dimensional local cosine basis can be generated by the tensor products \( \psi_{j_k}(x) \psi_{k_k}(y) \) and their nominal supports are the Cartesian product rectangles of the nominal supports of the \( x \) and \( y \) factors.

2.2 Adapted binary local cosine basis

For binary decomposition, if we fix the window width \( |I_j| = |\alpha_{j+1} - \alpha_j| \equiv |I| \) as a constant whatever \( j \) is, all wavelets in Eq. (1) can form an orthonormal basis for \( L^2(R) \), this basis is called level 1; then, we select \( |I|/2 \) as the window width, wavelets in Eq. (1) can also form an orthonormal basis, it is called level 2; \( \cdots \); and so on. Thus, a binary-based decomposition tree consists of the bases at different levels. However, not all the bases are efficient at matching a given signal, therefore, we want to pick up the “best-basis” from all the local cosine packets in the binary-tree library based on a cost-functional.

To search for local cosine best-basis, i.e., adaptive local cosine basis, aiming to achieve the best match of the signal, a cost-functional is defined to evaluate the “cost” or “efficiency” of the transformed signal. There are
several kinds of cost-functionals appearing in the literature [11]. Here, we use the so-called Shannon entropy as the cost-functional defined as

\[ E(X) = - \sum_n p_n \log_2(p_n) \]

where \( p_n = |x_n|^2/\|X\|^2 \), \( X \) stands for the signal with the length \( N \), \( x_n \) is the \( n \)-th component of \( X \) and \( E(X) \) denotes the entropy of signal \( X \).

In implementation, the Coifman-Wickerhauser (1992) binary-tree fast algorithm [1] is used to search for the “best-basis” based on the Shannon entropy cost-functional. The main idea of fast algorithm is that the full local cosine tree is pruned recursively at each node by comparing its entropy to the summation of the entropy of its corresponding child nodes,

IF Entropy(parent node) \leq [Entropy(child1)+Entropy(child2)] THEN cut off the child branches.

In the beginning, a full binary-based decomposition tree with a preset maximum decomposition level is produced. Then, the pruning procedure starts from the leaf nodes and proceeds toward the root. At the end of this procedure, an optimal pruned tree is obtained for the given signal, i.e., an adaptive binary local cosine basis is obtained.

3 NEW FLEXIBLE SEGMENTATION ALGORITHM

As shown by the example in Section I, the severe limitation of the binary decomposition tree is its binary decomposition. To overcome the binary segmentation constraint, we now design a flexible segmentation algorithm.

Let \( L \) stand for the time/space segmentation resolution, i.e., the length of the finest segment (cell) we want to have, for a 1-D signal with length \( N \), we always assume that \( N \) is a multiple of \( L \), say, \( N = KL \), where \( K \) is the total number of finest segments. In our segmentation algorithm, we choose the Shannon Entropy as the cost-functional.

Starting with the uniform finest segmentation, we adopt a left-to-right merging process to optimize the segmentation which doesn’t suffer from the binary tree restriction. For each possible merge, we compare the cost of the merged entity (segment) with the total cost, i.e., the sum of costs of the two separate entities (segments). If the cost of the merged segment is smaller than the total cost of two separate segments, the merge is approved, and the merged segment will be treated as one entity in the next possible merge. Otherwise, if the opposite is true, the merge will be abandoned. Since the merge is only possible between neighboring segments, the segment on the left after disapproval of a merging will be dropped from the list of merging candidates. The new merging process will start from the segment on the right and will never look back to the left. In other words, we will put a termination node at the right endpoint of the left segment, signaling the termination of an old merging process and the beginning of a new one.

At the end of the process, we obtain an optimized segmentation by retrieving all the termination nodes. Note that, in this segmentation, the minimum length of segments is the preset resolution, i.e., the cell length; while the maximum length of segments has no limit, and can be the whole length of signal in the extreme case. The binary segmentation process starts from the longest segment, i.e., the whole signal, and proceeds by dividing the segment in the middle, while our segmentation starts from the finest segments (cells) and achieves the flexible segmentation by a neighbor merging process. Many drawbacks of the binary segmentation process have been avoided by the new method.

For the 2-D case, based on the BINARY 2-D semi-ALCT scheme proposed by Wang and Wu (1999) in their paper [9], we can parallelly construct the FLEXIBLE 2-D semi-ALCT algorithm. It includes two semi-adaptive
schemes, i.e., FLEXIBLE 2-D time-ALCT and FLEXIBLE 2-D space-ALCT. For the FLEXIBLE 2-D time-ALCT, it is uniform in space direction, and adaptive in time direction within each fixed strip along the time direction. Its adaptability is accomplished by the above-mentioned 1-D new flexible segmentation algorithm. For the FLEXIBLE 2-D space-ALCT, the segmentation is uniform in time direction, but adaptive in space direction.

4 NUMERICAL EXPERIMENTS

In this section, our proposed new flexible segmentation algorithm will be tested on both synthetic signals and real seismic data. The tests will be done on the following four aspects: 1) No restriction on segmentation from the binary tree structure; 2) Much reduced time/space-shift sensitivity; 3) Less reconstruction distortion after compression; 4) Superior performance in seismic data compression.

In all the tests on synthetic signals, we always choose $N = 1024$ and time segmentation resolution, i.e., the smallest time segment, $L = 32$ and thus $K = 32$, although arbitrary time resolution can be used in our new FLEXIBLE segmentation algorithm. For the sake of comparison, in the BINARY tree algorithm, the maximum decomposition level (depth) is preset at 5 for synthetic signals so that the smallest time segment is also 32.

4.1 No binary constraint

To highlight the binary restriction of the BINARY tree algorithm, and to demonstrate the advantages of the new FLEXIBLE segmentation algorithm, we choose as our input signal a Ricker wavelet. In Fig. 1, the support of the Ricker wavelet is 21 from $t = 502$ to $t = 522$, and the peak of this wavelet just rides on the midpoint ($t = 512$) of the whole signal, a possible segmentation point for both the BINARY tree algorithm and the new FLEXIBLE segmentation algorithm. Fig. 1(a) shows the best segmentation result using the BINARY tree algorithm. We see that an intact impulse is unreasonably separated from the peak. Furthermore, to localize this impulse, many segments, which should not have existed, occur. In fact, because of the binary restriction, the impulse is either split into two halves from the peak, or kept as only one segment for the whole signal. Obviously, neither segmenting result is satisfactory. Undoubtedly, this kind of drawback is inherent in the BINARY tree algorithm. By contrast, in Fig. 1(b), although $t = 512$ is still a possible segmentation point for the new FLEXIBLE segmentation algorithm, the new scheme doesn’t separate the impulse from the peak any more and correctly localize the impulse without any abundant segments.

Another example is demonstrated in Fig. 2. The synthetic signal consists of a blocked sinusoid (from $t = 1$ to $t = 128$) and a Gaussian pulse (from $t = 805$ to $t = 815$, the peak is at $t = 810$). Note that the Gaussian pulse is not situated at any dividing points for the binary tree algorithm, which is different from the case in Fig. 1. This signal is designed to further test the restriction of the binary tree algorithm. The best segmentation from the BINARY tree algorithm is plotted in Fig. 2(a). Although the pulse is correctly segmented, an overabundance of segments occurs. Moreover, a complete sinusoid is unreasonably split into three parts. In contrast, in Fig. 2(b), the new FLEXIBLE segmentation algorithm not only separates the sinusoid from the rest of the signal but also finds the pulse. A careful examination of Fig. 2(b) can explain why the termination node for the sinusoid is not at $t = 128$, but instead at $t = 160$. This is due to the overlapping between segments. In fact, in the implementation of this new algorithm, the support of the window for the segment $[128, 160]$ is $[112, 176]$ (The overlap radius is 16 samples), hence, the folded signal in the segment $[128, 160]$ does not consist of all zeroes.

4.2 Much reduced time/space-shift sensitivity

Let the signal in Fig. 1 be shifted by 64 samples, twice the segmentation resolution $L$ ($L = 32$). Fig. 3(a) shows us the segmentation result obtained by the BINARY tree algorithm, which has changed a great deal compared with Fig. 1(a). This fact clearly illustrates the segmentation given by the BINARY tree algorithm is not shift invariant. However, the segmentation result in Fig. 3(b) generated by the new FLEXIBLE segmentation algorithm shows its shift invariant. Of course, we can not say that the new scheme completely eliminate the
Figure 1: Comparison of segmentation results from the BINARY tree algorithm and the new FLEXIBLE segmentation algorithm. The synthetic signal is a *Ricker* wavelet. (a) Time segmentation given by the BINARY tree algorithm, $N = 1024$, the maximum preset decomposition level is 5. (b) Segmentation obtained from the new FLEXIBLE segmentation algorithm, $N = 1024$, time segmentation resolution $L = 32$. Obviously, the *Ricker* wavelet should not be separated from the peak, therefore, only the new FLEXIBLE algorithm correctly localizes the impulse.

Figure 2: Comparison for a synthetic signal consisting of a sinusoid and a *Gaussian* pulse. (a) Time segmentation given by the BINARY tree algorithm, $N = 1024$, the maximum preset decomposition level is 5. Although the pulse is correctly identified, an overabundance of segments occurs. Moreover, a complete sinusoid is unreasonably split into three parts. (b) Segmentation obtained by the new FLEXIBLE segmentation algorithm, $N = 1024$, time segmentation resolution $L = 32$. The new FLEXIBLE segmentation algorithm not only separates the sinusoid from the rest of the signal but also accurately identifies the pulse.
problem of sensitivity to time/space shifts. This can be explained through the following demonstration. For example, if the signal in Fig. 1 is shifted by 80 samples, which are not the multiples of resolution $L$, then the segmentation may be different from the ones in Fig. 1(b) and Fig. 3(b). However, by the new FLEXIBLE segmentation algorithm, we do alleviate the problem of time/space-shift sensitivity. When the time/space shift is a multiple of segmentation resolution $L$, the segmentation by the new FLEXIBLE algorithm is shift invariant.

![Figure 3](image-url)

**Figure 3:** Comparison of time/space-shift sensitivity between the two segmentation schemes. A Ricker wavelet is shifted by 64 samples of the one in Fig. 1(a), twice the segmentation resolution $L$. (a) Segmentation by the BINARY tree algorithm; (b) Segmentation by the new FLEXIBLE algorithm. Compared to Fig. 1, segmentation in Fig. 3(a) changes a great deal, whereas the one in Fig. 3(b) is shift invariant.

### 4.3 Less reconstruction distortion after compression

To demonstrate the relatively weak reconstruction distortion by the new FLEXIBLE segmentation algorithm, we select a 1-D signal from a migration operator in the homogeneous medium ($velocity = 2000m/s$, $frequency = 5.9Hz$, $dx = dz = 6.25m$). Fig. 4 shows the segmentation results obtained by the BINARY tree algorithm and the FLEXIBLE segmentation scheme. Obviously, the signal should not be separated from the peak. Fig. 5 gives us the original signal (solid line) and the reconstructed signal (dotted line) from the compressed ALCT coefficients with the threshold=1% of the maximum absolute coefficient. Fig. 5(a) is from the best segmentation by the BINARY tree algorithm (see Fig. 4(a)), whereas Fig. 5(b) from the adapted segmentation by the new FLEXIBLE segmentation algorithm (see Fig. 4(b)). The results from the compressed ALCT coefficients with the threshold=0.5% and 0.1% of the maximum absolute coefficient are plotted in Fig. 6 and Fig. 7, respectively. Fig. 6(a) and Fig. 7(a) are from the BINARY tree algorithm, and Fig. 6(b) and Fig. 7(b) are from the new FLEXIBLE algorithm. From Figs. (5)-(7), we see that the new FLEXIBLE segmentation algorithm has the advantage of less reconstruction distortion. For example, in Fig. 7(b), the new method already accurately reconstructs the original signal, but, there still exists information loss in the left part of Fig. 7(a).

### 4.4 Superior performance in seismic data compression

In this subsection, we will demonstrate the superior performance of the new FLEXIBLE segmentation algorithm in seismic data compression compared with the BINARY tree algorithm.

Fig. 8(a)-(b) show us the segmentation results for a real 1-D seismic signal by the BINARY tree algorithm and the new FLEXIBLE algorithm, respectively. Here, $N = 1024$, $L = 64$, and the preset maximum decomposition level is 4. As seen from Fig. 8(a), the segmentation point $t = 832$ by the BINARY tree algorithm is unreasonable, whereas Fig. 8(b) with the new FLEXIBLE algorithm gives the correct partition.

In our past work [9], we have proposed two 2-D compression schemes based on the BINARY tree algorithm,
Figure 4: Comparison of segmentation results from the BINARY tree algorithm and the new FLEXIBLE segmentation algorithm. The signal is selected from a migration operator. (a) Segmentation given by the BINARY tree algorithm, $N = 1024$, the maximum preset decomposition level is 4. (b) Segmentation obtained from the new FLEXIBLE segmentation algorithm, $N = 1024$, segmentation resolution $L = 64$. Obviously, the signal should not be separated from the peak (Fig. 4(a)).

Figure 5: Comparison of reconstruction distortions between the two segmentation schemes. The threshold of compression is 1% of the maximum absolute coefficient. (a) By the BINARY tree algorithm; (b) By the new FLEXIBLE segmentation algorithm.
Figure 6: Same as in Fig. 5 but with a threshold of 0.5%.

Figure 7: Same as in Fig. 5 but with a threshold of 0.1%. As can be seen, the new method already accurately reconstructs the original signal, but, there still exists information loss in the left part of the reconstructed signal using the BINARY segmentation algorithm (Fig. 7(a)).
Figure 8: Segmentation test on a real seismic signal. (a) Segmentation given by the BINARY tree algorithm. Here, the preset maximum decomposition level is 4 and \( N = 1024 \); (b) Segmentation results by the new FLEXIBLE algorithm, \( N = 1024, L = 64 \). Obviously, in Fig. 8(a), \( t = 832 \) is an unreasonable segmentation point, whereas Fig. 8(b) gives the correct segmentation.

Figure 9: Compression performance comparison between BINARY 2-D space-ALCT and FLEXIBLE 2-D space-ALCT. The original SEG-EAGE data size is 626 samples in time direction and 1290 samples in space direction. For the binary method, it is extended to \( 1024 \times 2048 \). \((32,32)\) is the minimum \((\text{time}, \text{space})\) window size. In contrast, for the flexible method, it is only extended to \( 630 \times 1290 \), and \((30,30)\) is the \((\text{time}, \text{space})\) segmentation resolution. Obviously, better compression performance is achieved by the new FLEXIBLE segmentation method.
i.e., 2-D time-ALCT and 2-D space-ALCT, and applied them to the compression of a subset of the SEG-EAGE salt data set, a synthetic zero-offset data from the salt model using a finite-difference exploding reflector modeling algorithm, generated at AMOCO. We concluded that (32, 32) or (32, 64) minimum (time, space) window size can generate the best compression result for the data set. In this paper, using the new FLEXIBLE segmentation algorithm, we also tested the two 2-D compression schemes, i.e., FLEXIBLE 2-D time-ALCT and FLEXIBLE 2-D space-ALCT (see Section III). Fig. 9 is the compression performance comparison between BINARY 2-D space-ALCT and FLEXIBLE 2-D space-ALCT for the salt data. We can see that (30, 30) (time, space) segmentation resolution for the new FLEXIBLE algorithm can even provide better compression performance than the (32, 32) BINARY 2-D space-ALCT, which is already the best result in binary schemes for the data set. Here, the SNR is defined as follows,

$$SNR = 10 \log_{10} \left( \frac{\sum |c_k|^2}{\sum |\epsilon_k|^2} \right)$$

where $c_k$ is the coefficient above the threshold in the absolute value sense and thus being retained, and $\epsilon_k$ is the coefficient discarded.

5 CONCLUSIONS

In this paper, we proposed a new FLEXIBLE segmentation algorithm with arbitrary time/space segmentation resolution. Compared with the BINARY best-basis searching algorithm, the new algorithm has no binary constraint and thus is much more flexible and less sensitive to time/space shifts, therefore, it has less reconstruction distortion and superior performance in the applications to real seismic signal processing. These advantages have been verified by the numerical tests on both synthetic and real seismic signals. This new FLEXIBLE segmentation algorithm has been applied to 2-D seismic data compression with two semi-adaptive schemes: FLEXIBLE 2-D time-ALCT and FLEXIBLE 2-D space-ALCT. 2-D numerical tests have been performed on the SEG-EAGE synthetic salt data subset using the FLEXIBLE 2-D space-ALCT. Superior compression performance for seismic data is achieved.

ACKNOWLEDGMENTS

This research is supported by the WTOPI (Wavelet Transform On Propagation and Imaging for seismic exploration) Consortium at University of California, Santa Cruz. The authors would like to thank the sponsors of the consortium. The facilities have been supported in part by the W. M. Keck Foundation. Contribution number 401 of the Institute of Tectonics, University of California, Santa Cruz.

REFERENCES


