DISTRIBUTED MULTISCALE DATA ANALYSIS AND PROCESSING FOR SENSOR NETWORKS

Raymond Wagner, Shriram Sarvotham, Hyeokho Choi, Richard Baraniuk

Rice University
Department of Electrical and Computer Engineering
Houston TX, USA

ABSTRACT

While multiresolution data analysis, processing, and compression hold considerable promise for sensor network applications, progress has been confounded by two factors. First, typical sensor data are irregularly spaced, which is incompatible with standard wavelet techniques. Second, the communication overhead of multiscale algorithms can become prohibitive. In this paper, we take a first step in addressing both shortcomings by introducing two new distributed multiresolution transforms. Our irregularly sampled Haar wavelet pyramid and telescoping Haar orthonormal wavelet basis provide efficient piecewise-constant approximations of sensor data. We illustrate with examples from distributed data compression and in-network wavelet de-noising.

1. INTRODUCTION

In the recent call-to-arms found in [1], the authors emphasize that spatially irregular data sampling is an inescapable reality when considering real-world sensor network deployments. Using the example of compression of a sensor network measurement field for transmission to a single, external sink via a tree-like routing structure, they emphasize that traditional regular 2-D signal processing schemes simply do not translate to the irregular setting. Observing that much of sensor network signal processing literature assumes regularity of sensor placement, the authors motivate the need for newer, irregularity-tolerant solutions for the sensor network setting.

To that end, we propose what is to our knowledge the first distributed, irregular wavelet transform for sensor networks. Employing successive piecewise constant approximations inspired by the Haar wavelet in the regular setting, our transform integrates seamlessly with existing hierarchical routing schemes to align its data flow with efficient communication paths in the network. We present two variants on our technique and outline exactly how to implement each in a real sensor network setting. Finally, using our transform, we specify protocols for distributed solutions to both the data compression and transmission problem discussed above and the problem of de-noising sensor measurements within the network.

A good deal of prior work has addressed wavelet-based processing in sensor networks, but none has yet to resolve the difficulties of working with irregularly-spaced data while appreciating the need to minimize communication overhead to reduce network power consumption. Dimensions [2, 3] proposes an in-network multiscale wavelet transform and hierarchical coefficient routing with its wavRoute protocol but assumes sensor measurements lie on a square grid. Similarly, Wisden [4] also assumes a square grid and application of regular wavelets for compression purposes, as do [5] and [6]. Finally, Fractional Cascading [7] assigns to a sensor a view of the entire network which decays in resolution as the network becomes more distant from the sensor. While this approach does not assume regular sensor placement and is useful for answering measurement range queries, it does not have the broad applicability of a wavelet transform to a variety of signal processing applications.

In specifying our transform algorithms, we assume only that each sensor knows its location and shares this information with a data sink and that an efficient multiscale routing hierarchy is already in place. A number of algorithms for sensor localization address the former problem (see [8]), and both wavRoute mentioned above and COMPASS [9] provide examples of the latter. COMPASS, in particular, creates a hierarchical routing scheme by clustering nodes, electing clusterheads, and iteratively clustering the clusterheads. Under this paradigm, local communication within clusters and up and down the routing hierarchy are less expensive than communication across cluster boundaries. Adopting the routing hierarchy as our multiscale decomposition hierarchy allows us to tailor data flows to match the economics of the routing protocol.

In Section 2 we review challenges of irregular data sampling and motivate the need for wavelet transforms tailored to irregular data. Section 3 provides the mathematical foundations of the two proposed wavelet transforms, beginning with the multiscale dataflow model induced by the routing topology. The details of the two transforms, one a tight frame pyramid and the other a complete orthonormal basis, follow. Section 4 addresses details of implementing these transforms, and Section 5 provides protocols for both distributed data compression/ transmission and distributed de-noising of sensor measurements. Section 6 demonstrates the aforementioned applications in a simulated sensor network setting, and Section 7 concludes with a discussion of future directions for such irregular wavelet transforms.

2. CHALLENGES OF IRREGULAR DATA SAMPLING

A large body of elegant wavelet theory suited to regularly-spaced data has emerged in recent decades, but application of wavelet analysis to irregularly spaced samples is a relatively new endeavor. The Fourier mathematics underlying regular wavelets no longer apply in the regular setting, and second-generation wavelet the-

This work was supported by NSF, AFOSR, ONR, and the Texas Instruments Leadership University Program. Email: {rwagner,shri,choi,richb}@rice.edu. Web: dsp.rice.edu.

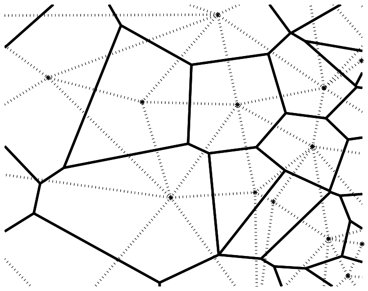

Fig. 1. Voronoi polygons (solid) bounding a field of sensors connected by edges in a Delaunay triangulation (dashed).

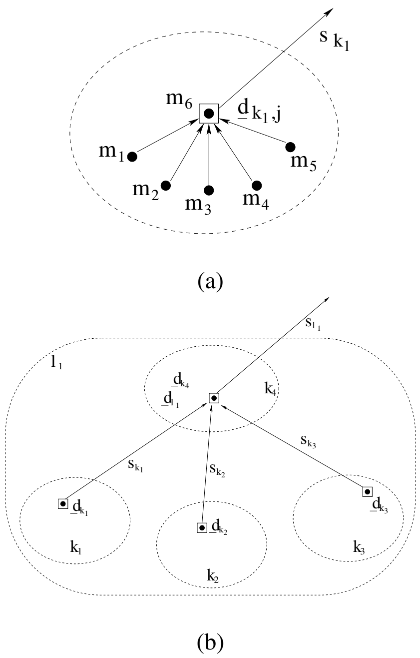

Fig. 2. Data flow (a) within a single cluster k_1 and (b) from clusterheads in k_1 through k_3 to the supercluster l_1 head in cluster k_4.

ory such as the lifting scheme [10] has risen to replace traditional techniques. Extending these ideas to two dimensions proves an ever greater challenge. The pyramid-based approach of [11] and the non-redundant technique given in [12] represent some of the first sophisticated attempts at tackling this problem. Unfortunately, such approaches are typically intended to operate in a centralized fashion on an entire dataset, so as such they are not directly applicable to the sensor network problem. For example, they typically assume complete freedom to construct their own multiscale analysis hierarchies, and as stated in Section 1, communication efficiency suggests that transforms must direct their flows along routing hierarchies formed with respect to physical constraints such as sensor placement and wireless channel quality.

Rather than attempting to directly apply these ideas, we take inspiration from the approaches contained therein. Varying sensor density poses one of the key challenges in computing an irregular transform, as measurements from nodes in more dense areas of the graph will appear to have greater "importance" than those from less dense regions. To counter this problem, we adopt the technique in [12] of assigning weight to sensor measurements based on the area of a Voronoi polygon [13] around the sensor. Figure 1 illustrates a set of Voronoi tiles drawn as solid lines around a set of sensor locations. The figure also shows a Delaunay triangulation of the sensor locations drawn in dashed lines. Voronoi tesselations are in fact geometric duels of Delaunay triangulations, and a sensor can compute the area of its surrounding Voronoi polygon merely by knowing the edges for which it is an endpoint in the Delaunay triangulation. Delaunay triangulations can be computed in a distributed fashion within the sensor network, as will be discussed in Section 4.

3. MULTISCALE TRANSFORMS WITH IRREGULAR SAMPLES

We now present the mathematical details behind our proposed transforms. First, we discuss the interplay between network routing and multiscale data decomposition. Then, we delve into the details behind our two proposed transforms, one a tight-frame Haar pyramid and the other an orthonormal basis Haar "telescoping" transform.

3.1. Hierarchical Routing and Data Decomposition

In Section 1 we introduced the idea of matching the transform hierarchy to the routing hierarchy. Such a hierarchical routing topology will typically consist of clusters of sensors, one of which is designated as the clusterhead. Figure 2(a) describes the scenario in a typical 6-node cluster, labeled k_1. Under the assumed routing economics, local communications within a cluster are relatively cheap, so each sensor m_1 through m_5 sends it measurement to the clusterhead m_6 for processing. Using the set $\underline{m}$ of measurements, the clusterhead computes two quantities. A *scaling coefficient* s_{k_1} describes some average behavior over the cluster, while a set $\underline{d_{k_1}}$ of *wavelet coefficients* encode deviations of the measurements from the average value.

The clusterhead stores wavelet coefficients $\underline{d_{k_1}}$, but the scaling coefficient s_{k_1} flow up to the next level of the transform/routing hierarchy, as show in Figure 2(b). Along with cluster k_1 from Figure 2(a), clusters k_2 through k_4 are grouped together to form a *supercluster* l_1, whose clusterhead coincides with that of cluster k_4. Scaling coefficients s_{k_1} through s_{k_3} are passed to cluster k_4, which combines them with its own s_{k_4} as the input to the next level of multiresolution analysis. A scaling coefficient s_{l_1} for the super cluster is generated along with a set $\underline{d_{l_1}}$ of wavelet coefficients. Note that the node acting as the super clusterhead performs double duty. As the clusterhead of cluster k_4, it stores the set $\underline{d_{k_4}}$ of level-k wavelet coefficients, and as the clusterhead for the supercluster l_1, it maintains the set $\underline{d_{l_1}}$ of level-l wavelet coefficients. The scaling coefficient s_{l_1} for the supercluster is then passed up the hierarchy to participate in subsequent analysis. The iterative process terminates when one supercluster spans the entire network and a scaling coefficient for a single root node is computed. This root scaling value, which describes the average behavior over the whole network, and the entire set of wavelet coefficients at each level represent the completed multiresolution analysis of the sensor field. Note that, under this model, the hierarchical wavelet decomposition matches *exactly* with the routing topology, grouping data arbitrarily as dictated by routing concerns. As such, our proposed techniques are extraordinarily flexible in the realm of irregular wavelet transforms.

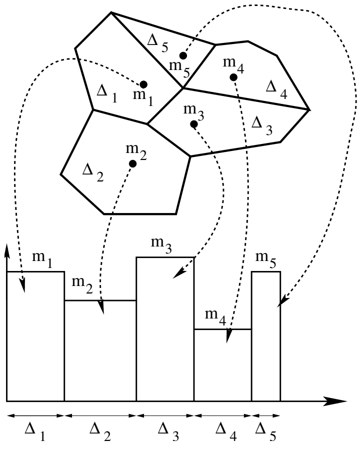

Fig. 3. Conceptual mapping of piecewise-constant function over 2-D intervals of area Δ_i to piecewise-constant 1-D function over intervals of width Δ_i

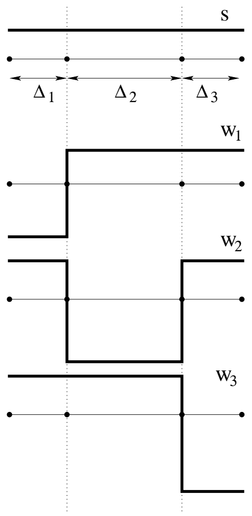

Fig. 4. Tight frame analysis functions for a cluster of 5 nodes, including one scaling function S and 3 wavelet functions W_1 through W_3

Given this transform model, we can present the details of our two proposed transforms. Both are based on piecewise-constant approximations as seen in the well-known Haar wavelet in the regular setting by the. We adopt a similar piecewise-constant model for the data gathered in the network, assuming that measurements hold constant over the Voronoi polygons surrounding each sensor, as described in Section 2. Section 3.2 details a pyramid-like transform whose analysis and synthesis functions are remarkably easy to implement but lead to a slightly redundant representation, computing 1 scaling and N wavelet coefficients for a cluster of N sensors. Section 3.3 presents a more advanced transform which employs a complete orthonormal basis expansion of sensor measurements, achieving a non-redundant coefficient representation (N coefficients for an N-sensor cluster). Providing a sparser representation of the sensor field, this transform nonetheless carries a slightly higher cost in terms of computational cost and ease of implementation.

3.2. Tight-Frame Haar Pyramid

The first transform, originally introduced in [14], represents an extraordinarily intuitive and easy-to-implement decomposition of sensor data into wavelet and scaling coefficients. Harkening back to the pyramid image coders which preceded modern wavelet coders [15], the transform passes a smoothed average to the next level of decomposition via the scaling coefficient s and leaves behind as many detail coefficients $\underline{d}$ as there are sensors in a cluster — in essence, a redundant transform with the slightest possible redundancy.

We now describe how to calculate the transform coefficients given an N-length vector $\underline{m}$ of measurements for a cluster of N sensors. We assume that each sensor in the cluster knows it's support size as defined by the Voronoi polygon surrounding the sensor. These areas, $\{\Delta_1, \Delta_2, \ldots, \Delta_N\}$, can be computed in a distributed fashion, as detailed in Section 4. Given the set of areas and measurements, we can conceptually map the 2-D transform problem to 1-D as shown in Figure 3, where sensor measurements are taken to be constant over intervals whose lengths correspond to sensors' Voronoi region areas. Note that this mapping is purely for illustrative purposes and that all subsequent integrations in fact involve piecewise-constant functions over 2-D regions.

Given the set $\{\Delta_j\}_{j=1}^N$ of areas, we form the $(N+1) \times N$ matrix $\mathbf{K}$, given by

$$\mathbf{K} = \begin{pmatrix} k_0 & k_0 & k_0 & \cdots & k_0 \\ k_1^{'} & k_1 & k_1 & \cdots & k_1 \\ k_2 & k_2^{'} & k_2 & \cdots & k_2 \\ k_3 & k_3 & k_3^{'} & \cdots & k_3 \\ \vdots & \vdots & \vdots & & \vdots \\ k_N & k_N & k_N & \cdots & k_N^{'} \end{pmatrix}, \tag{1}$$

where the total area over the cluster is given as $\Delta_{tot} = \sum_{i=1}^N \Delta_i$. The bulk of the entries of $\mathbf{K}$ are defined as

$$k_i = \begin{cases} \frac{1}{\sqrt{\Delta_{tot}}}, & i = 0 \\ \\ \frac{\sqrt{\Delta_i}}{\Delta_{tot}}, & i > 0 \end{cases} \tag{2}$$

while the first sub-diagonal entries are given by

$$k_i^{'} = -\frac{k_i(\Delta_{tot} - \Delta_i)}{\Delta_i}, \quad i > 0. \tag{3}$$

Taken together, the entries of $\mathbf{K}$ define the basis functions given in Figure 4. The first row of $\mathbf{K}$ represents the constant-valued scaling function, while each $(i,j)^{th}$ element in the remaining rows gives the value of wavelet function $i-1$ over the Voronoi cell surrounding sensor j

Given the matrix $\mathbf{K}$ describing the set of basis functions, all that is left to do is simply multiply against the sensor measurements and integrate over the set of support areas. Defining a an N-by-N matrix $\mathbf{\Delta}$ as a diagonal matrix of the areas $\{\Delta_j\}_{j=1}^N$, we form the analysis matrix as

$$\mathbf{T}_A = \mathbf{K}\mathbf{\Delta}. \tag{4}$$

Using the analysis matrix, we compute the $(N+1) \times 1$ vector of transform coefficients $\underline{c}$ as

$$\underline{c} = \mathbf{T_A}\underline{m}, \tag{5}$$

where the first element of $\underline{c}$ is the scaling coefficient s and the remaining N elements give the wavelet detail coefficients $\underline{d}$. Defining the synthesis matrix as

$$\mathbf{T}_S = \mathbf{\Delta}^{-1}\mathbf{T}_A^T, \tag{6}$$

we recover the set of measurements from the coefficients as

$$\underline{m} = \mathbf{T_S}\underline{c}. \tag{7}$$

As stated previously, the analysis and synthesis functions form a tight frame. In fact, they form a special class of tight frame known as a *Parseval tight frame* [16]. Given this membership, we conclude that, even though the frame is redundant, we have an equivalence between energy in the signal domain and energy in the coefficient domain:

$$E = \sum_{i=0}^{N} c_i^2, \tag{8}$$

where E is calculated by squaring the piecewise-continuous signal given in Figure 3 and integrating it over its support regions. This equivalence implies that, were we to selectively discard q wavelet coefficients, the energy $\widehat{E}$ of the resultant signal would be maximized by discarding the smallest q coefficients — exactly the procedure we turn to for the compression application discussed in Section 5.1. Moreover, given properties of the transform, we can bound the difference in energies between the reconstructed and original signals as

$$\sum_{i=1}^{q} c_i^2 \leq (E - \hat{E}) \leq 2(\sum_{i=1}^{q} c_i^2), \tag{9}$$

ensuring that the approximated signal will not blow up when the inverse transform is applied — a useful sanity check when working with tight frames, whose redundancy can induce problems in such situations. For a complete proof of the transform's Parseval tight frame properties and error bounds, we refer the reader to [17].

Clearly, the tight frame pyramid transform is extremely easy to implement. Given a set $\{\Delta_j\}_{j=1}^N$ of sensor support areas, computing the wavelet decomposition amounts to multiplying the sensor measurements against $\mathbf{T_A}$, which depends entirely on the $\{\Delta_j\}_{j=1}^N$. Unfortunately, this does entail suffering redundancy in the coefficient representation — a problem we can rectify with a second, more computationally intensive piecewise-constant transform.

3.3. Orthonormal Basis Haar Telescope

Clearly, a non-redundant, distributed wavelet transform for sensor networks is superior from a data-representation standpoint. In this section we present a technique which appeals to classic Haar wavelets in the 1-D setting and forms a set of basis functions which by construction are orthonormal and span the piecewise-constant measurements in a cluster of sensors. Figure 5(a) recalls the regular Haar scaling and wavelet functions in 1-D. Applied to data which is piecewise constant over the intervals $[0, 0.5]$ and $[0.5, 1]$, the scaling function merely sums measurements in the two regions while the wavelet function differences those measurements. The two functions are each unit norm, mutually orthogonal (their product integrates to zero), and span the entire interval (any function

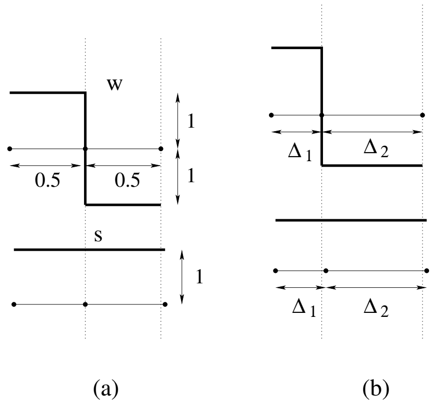

Fig. 5. (a) Regular, 1-D Haar wavelet (W) and scaling (S) basis functions and (b) corresponding matched pair telescope basis functions.

which is piecewise constant over $[0, 0.5]$ and $[0.5, 1]$ can be formed as a linear combination of the Haar scaling and wavelet functions).

By grouping sensors in our irregular setting into pairs, we can perform a decomposition mirroring that of the classic Haar case, though we must take into account the support sizes of our sensor measurements when doing so. Appealing to the 1-D visualization of Section 3, a conceptual mapping of sensor measurements to 1-D intervals places paired sensors in adjacent positions on the line so that pairwise basis functions such as those shown in Figure 5(b) apply over the pair of intervals. Again we stress that this mapping is merely for visualization purposes — the pertinent information is contained in adjacent sensor pairings, which we describe how to generate later in the section.

To each pair of sensors, we apply the set of scaling and wavelet coefficients given in the figure, where $h_1 = \sqrt{\frac{\Delta_2}{\Delta_1(\Delta 1+\Delta_2)}}$, $h_2 = -\sqrt{\frac{\Delta_1}{\Delta_2(\Delta 1+\Delta_2)}}$, and $h_1 = \sqrt{\frac{1}{\Delta 1+\Delta_2}}$. Thus, for a given cluster, we have that

$$\begin{aligned} s &= (m_1\Delta_1 + m_2\Delta_2)\sqrt{\tfrac{1}{\Delta 1+\Delta_2}} \\ d &= m_1\Delta_1\sqrt{\tfrac{\Delta_2}{\Delta_1(\Delta 1+\Delta_2)}} + -m_2\Delta_2\sqrt{\tfrac{\Delta_1}{\Delta_2(\Delta 1+\Delta_2)}}. \end{aligned} \tag{10}$$

Using the h_0, h_1, and h_2 expressions, it is easily verified that the pair of basis functions are mutually orthogonal, have unit-norm, and span the space of functions constant over the Δ_1 and Δ_2 intervals.

Pairing measurements to generate wavelet and scaling coefficients according to Equation 10 gives us roughly $N/2$ scaling coefficients over the cluster of N sensors. Recall, though, that to fit into the model of Figure 2(a) we must describe the whole cluster with a single scaling value for transport to the next layer of the hierarchy. To do so, we merely iterate the pairing and transform process, operating now on scaling coefficients whose supports spans those of the pairs from which they were derived. This process effectively turns the arbitrary, $(N-1)$–to–1 single-depth tree provided by the routing hierarchy into a *virtual binary tree* as shown in Figure 6. We refer to this structure as a **telescope tree** since it "stretches" the cluster into a deeper tree. Note that the tree

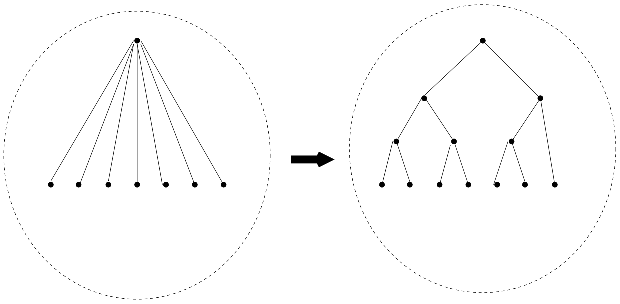

Fig. 6. Generating a virtual telescoping tree from the routing hierarchy cluster specification.

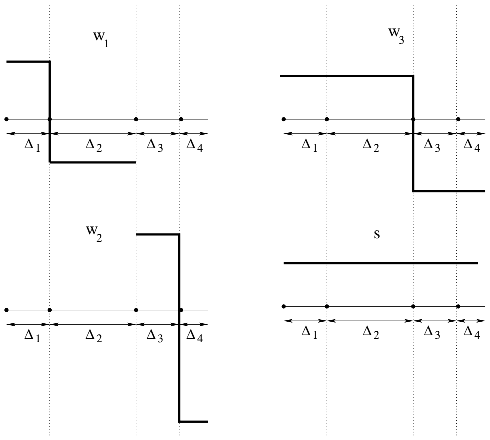

Fig. 7. Comparing a pair of disjoint wavelet functions W_1 and W_2 with the wavelet function W_3 and scaling function S spanning the combined support at the next level of telescope hierarchy.

is, in fact, virtual, as the entire transform is calculated by the clusterhead given the measurements and supports for all sensors in the cluster, as described in the introduction to Section 3.

Orthonormality of the basis functions holds over levels of the telescope tree, as illustrated in Figure 7. W_1 and W_2 in the figure describe wavelet functions in separate pairs over the intervals $\{\Delta_1, \Delta_2\}$ and $\{\Delta_3, \Delta_4\}$ on the same level of the tree. Clearly, these functions are orthogonal given their disjoint support intervals. W_3 and S describe the wavelet and scaling functions at the next level of the hierarchy, which pairs the scaling coefficient over $\{\Delta_1, \Delta_2\}$ with that over $\{\Delta_3, \Delta_4\}$. It is not difficult to see that both W_1 and W_2 at the lower level of the hierarchy are orthogonal to the next-level functions. By construction, all wavelet functions integrate to zero against a constant value. As W_1 and W_2 both reside in regions spanned by constants in W_3 and S, their inner products with those functions are zero, ensuring orthonormality of the complete set of wavelet functions and the scaling function employed in the telescoped decomposition.

Given pairings that form the binary tree in Figure 6 and repeated application of Equation 10 to the pairs, we can form a set of $N-1$ wavelets and a scaling coefficient for a cluster of N sensors. Moreover, as we have a complete orthonormal basis across the cluster, the relation in Equation 8 and approximation energy error is fixed at

$$E - \hat{E} = \sum_{i=1}^{q} c_i^2, \tag{11}$$

where i again indexes q smallest-magnitude wavelet coefficients, which have been set to zero prior to reconstruction.

Computational complexity in this telescoping transform clearly resides in the pairing scheme used. We appeal to a technique known as Perfect Matching [18], which pairs points so that some objective function is minimized. As clusters may have an odd number of nodes, as illustrated in Figure 6, care is needed to specify the non-paired node, which retains its value and support into the next round of matching. We discuss the details of matching, as well as a variety of other practical concerns, in Section 4.

4. IMPLEMENTATION DETAILS

Given both the tight-frame pyramid algorithm described in Section 3.2 and the orthonormal basis algorithm of Section 3.3, we now have two methods for computing piecewise-constant distributed wavelet transforms for sensor networks. In this section we discuss implementation issues of the two transforms, distinguishing the former from the latter. But before addressing their differences, we begin with a common requirement of both.

As discussed in Section 2, the reality of irregular sampling requires that we appropriately weight sensor measurements to account for non-uniform sensor densities. This motivates us to calculate the area of the Voronoi polygon surrounding each sensor, giving a useful measure of the support-size of each sensor value. As the transform progresses up levels in the hierarchy, support areas are merely aggregated and applied to the scaling coefficients which summarize finer-level measurements. Clearly, all higher-level area information can flow up the hierarchy, leading to efficient aggregation. Thus, the real challenge becomes efficiently computing the base-level Voroni cell areas.

Fortunately, efficient, distributed algorithms already exist for computing a Delaunay triangulation of sensors. Recall from Section 2 that the Delaunay triangulation is the geometric dual of the Voronoi polygon. Given the set of Delaunay edges to which it belongs, a sensor can easily compute the area of its Voronoi cell by simple geometric construction. A method for computing the triangulation of an n-node network with an $O(n \lg n)$ communication cost is given in [19], which constructs a subset of the Voronoi graph having all but the largest edges in the triangulation. Edges not found by the distributed algorithm can conceivably be sent from outside the network, as the network user is assumed to know all sensor locations and can easily identify these edges. As area calculation need happen only once, this extra communication overhead should have a small amortized cost.

Once we know support sizes, we can implement each transform as detailed in Algorithms 1 and 2, which respectively provide pseudocode for the pyramid and telescope transforms within a cluster. Note the disparity in complexity between the two. The pyramid transform merely computes the analysis matrix specified by Equation 4 given the set of support areas and multiplies the matrix against the value set for the cluster, passing the resultant scaling coefficient and aggregate area up the hierarchy.

The telescope transform, which yields a superior, non-redundant wavelet expansion, requires a bit more computation. Each level of the telescope tree within a cluster requires identification of value pairs. To compute these pairings, we appeal to Perfect Matching

Algorithm 1 PyramidXfm(Values, Areas)

form analysis matrix $\boldsymbol{T}_A$ (Eqn. 4) using Areas
Coefficients $\leftarrow \mathbf{T}_A Values$
RETURN Coefficients(1), sum(Areas)

Algorithm 2 TelescopeXfm(Values, Areas)

if size of Values is 1 **then**
  RETURN Values, Areas
end if
newValues $\leftarrow \emptyset$, newAreas $\leftarrow \emptyset$
Pairs $\leftarrow$ perfectMatch(Areas)
for each pair $\in$ Pairs **do**
  compute pair scaling/wavelet coefficients (Eqn. 10)
  insert scaling coefficient in newValues
  insert sum of pair areas in newAreas
end for
if number of values is odd **then**
  insert leftover value in newValues
  insert leftover area in newAreas
end if
TelescopeXfm(newValues, newAreas)

as described in [18]. The Perfect Matching algorithm computes the lowest-cost match between points in a set given a cost function. For the telescoping tree, we choose distance between paired sensors as our minimizing metric and constrain chosen edges to correspond to edges in the Delaunay triangulation (guaranteeing merged cells share a common boundary). As Voronoi cells merge into super-cells, we define the distance between the super-cells to be the minimum distance between centers of their member Voronoi polygons. This distance-based matching requirement insures that difference coefficients remain as small as possible by exploiting the similarity in spatially proximate measurements from smooth fields.

Though the Perfect Matching requirement does add computational and implementation complexity to the transform computed at a cluster, fast algorithms exist for computing the match [20]. Additionally, the telescope tree within a cluster need only be computed once and stored, provided nodes remain stationary and do not drop out of the network. Such an approach also opens the door to computing all pairings outside the network and flowing the match information down the hierarchy to clusters. Such an approach would entail far more communication, but the amortized cost could potentially be tolerable when spread over many subsequent transform operations.

Now that we have completely described our proposed transforms and their implementation issues, we proceed in Section 5 to outline how transform coefficients can be applied to a pair of pertinent signal processing problems.

5. APPLICATION OF TRANSFORM DATA

Transmitting a compressed version of the entire measurement field to an exterior sink represents an expected sensor network task, as suggested in [1]. Removing noise in a distributed fashion from sensor measurements polluted by additive IID processes will also find much use in sensor networks, especially as a pre-processing step for more advanced distributed algorithms. We describe below algorithms for doing so, given wavelet coefficients calculated as

Algorithm 3 xmtCompressed(ClsthdID,Threshold)

CandidateCoeffs $\leftarrow$ Coefficients(ClsthdID)
Coeffs2xmt $\leftarrow \emptyset$, ID2xmt $\leftarrow \emptyset$
for each coefficient $\in$ CandidateCoeffs **do**
  if *(magnitude of coefficient)* $<$ Threshold **then**
    insert coefficient into Coeffs2xmt, node ID into ID2xmt
  end if
end for
send Coeffs2xmt and ID2xmt up the hierarchy to the root
for** each node in cluster **do
  xmtCompressed(nodeID,Threshold)
end for

described above.

5.1. Distributed Compression and Transmission

Sections 3.2 and 3.3 appeal to Equation 8 to motivate thresholding of wavelet coefficients as an optimal procedure for approximating the function they represent. As reconstruction error energy equals that of discarded wavelet coefficients, discarding smallest coefficients first leads to an optimal approximation, given that only a specified number of coefficients can be retained.

Thus, given a magnitude threshold below which wavelet coefficients are presumed insignificant, compression and transmission of the measurement field proceeds as described in Algorithm 3, which is called on the root node.

Given a threshold, a clusterhead sends up the hierarchy significant wavelet coefficients along with node identifiers indicating which nodes in the transform generated the coefficients. It then passes the threshold down the hierarchy to child nodes in the cluster.

5.2. Distributed Measurement De-Noising

Noise in sensor measurements manifests itself as small wavelet coefficients. Therefore, de-noising proceeds much as compression, where wavelet coefficients are compared against a threshold. An important difference arises, though. IID sensor noise applies only to individual sensor measurements and not to their entire support areas. As wavelet coefficients are derived with respect to continuous functions over the plane, we must tailor the threshold to each coefficient.

We inject an estimate $\widehat{\sigma_n}$ of the noise variance into the network. Considering the telescoped transform and a single cluster, we fashion a threshold for each wavelet coefficient in the telescope tree as follows. Where $\{\Delta_i\}_{i=1}^{u}$ and $\{\Delta_i\}_{i=u+1}^{v}$ describe the v base-level Voronoi cells aggregated into the positive and negative support regions of the functions given in Figure 5(b), the threshold t for the wavelet coefficient generated from those basis pairs is computed as

$$t = \sigma_n \sqrt{h_1^2 \sum_{i=1}^{u} \Delta_i^2 + h_2^2 \sum_{j=u+1}^{v} \Delta_j^2} \qquad (12)$$

Note that this requires knowledge at each transform level of the base-level Voronoi cells spanned by the transform function supports. Such information can be computed in a single bottom-up pass along the routing hierarchy and stored for future reference.

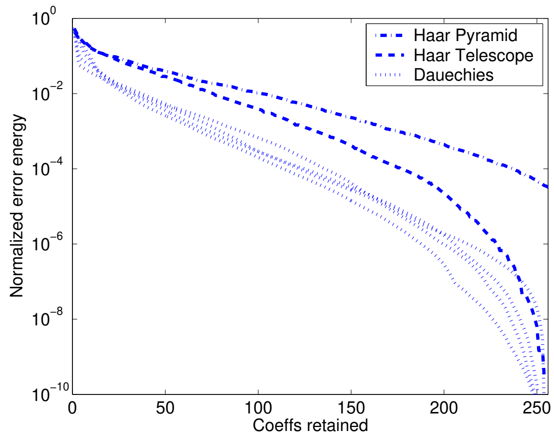

Fig. 8. Comparison of Haar Pyramid and Haar Telescope to D-2, D-4, D-6, and D-8 wavelets in a regular setting.

Now, given that appropriate thresholds can be applied to each wavelet coefficient, de-noising proceeds as follows. Starting at the hierarchy root, we threshold wavelet coefficients, compute inverse transforms, and pass reconstructed values down to child clusters as scaling values for subsequent inverse transforms. This process iterates top-down until the base-level values are recovered. These then represent de-noised versions of original sensor measurements.

6. EXAMPLES

We now proceed to demonstrate the effectiveness of our proposed transforms on a variety of simulated sensor measurement fields. We show results for both compression and de-noising, but first we provide a comparison of our technique, operating in the regular setting, to a standard regular wavelet transform.

6.1. Comparison to Wavelets on Regular Grids

As a brief sanity check, we compare the performance of our proposed transforms in the regular setting to transforms using regular Daubechies-2, Daubechies-4, Daubechies-6, and Daubechies-8 wavelets. For all transforms, we plot reconstruction error versus numbers of coefficients retained with increasing fidelity. For this setting, the sensor network consists of a regular square grid of 256 sensors. We allow sensors to cluster randomly and evaluate the resultant reconstructions against those given by the Daubechies wavelets operating on the entire square grid without clustering. Figure 8 illustrates the relative performances. Though both the Haar Pyramid and Haar Telescope transforms are out-performed by the Daubechies wavelets (as expected), their performance is clearly comparable. Thus, the transforms are capable of performing well in both the regular and irregular domains — a claim the Daubechies wavelets cannot make.

6.2. Compression and Transmission of Network Data

Now, we move back into the irregular domain for an examination of the relative performances of the Haar Pyramid and Haar Telescope transforms. Figures 9 (a), (b), and (c) give the three simulated sensor measurement fields used in the study. The first consists of a piecewise-planar field with discontinuities in its left and right corners, the second of a smooth quadratic, and the third of a noisy quadratic with a discontinuity. Relative performances of the Haar Pyramid and Telescope transforms are given in Figures 9 (d), (e), and (f), with the non-redundant Haar Telescope outperforming the easier-to-implement Haar Pyramid, as expected. Finally Figures 9 (g), (h), and (i) show the fields reconstructed from the Haar Telescope function by retaining 250 of the 2,500 coefficients. Clearly, major features of the fields are retained even in the presence of such heavy compression

6.3. In-Network De-noising of Measurements

To evaluate the de-noising operation, we took samples from the noisy quadratic function of Figure 9(c). Over 60 trials we realized a 5.48 dB improvement in signal-to-noise ratio using soft thresholding.

7. CONCLUSION

In conclusion, we have successfully demonstrated novel irregular wavelet transforms which are ideally suited to the setting of sensor networks. Providing transforms which are both implementable and practically valuable, we have demonstrated their utility in the applications of distributed data compression and distributed de-noising. To extend this work, we intend to pursue higher-order approximations, enabling superior data compression.

8. REFERENCES

[1] D. Ganesan, S. Ratnasamy, H. Wang, and D. Estrin, "Coping with irregular spatio-temporal sampling in sensor networks," *SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 1, pp. 125–130, 2004.

[2] D. Ganesan, D. Estrin, and J. Heidemann, "DIMENSIONS: Why do we need a new data handling architecture for sensor networks?," *SIGCOMM Comput. Commun. Rev.*, vol. 33, no. 1, pp. 143–148, 2003.

[3] D. Ganesan, B. Greenstein, D. Perelyubskiy, D. Estrin, and J. Heidemann, "An evaluation of multi-resolution storage for sensor networks," in *Proc. ACM SenSys Conference*, Los Angeles, California, USA, November 2003, pp. 89–102.

[4] N. Xu, S. Rangwala, K. Chintalapudi, D. Ganesan, A. Broad, R. Govindan, and D. Estrin, "A wireless sensor network for structural monitoring," in *Proc. ACM SenSys Conference*, November 2004.

[5] S.D. Servetto, "Sensing Lena - Massively distributed compression of sensor images," in *Proc. IEEE Int. Conf. on Image Proc. (ICIP)*, September 2003, pp. I–613—I–616.

[6] A. Ciancio and A. Ortega, "A distributed wavelet compression algorithm for wireless sensor networks using lifting," in *Proc. IEEE Int. Conf. on Acoustic and Speech Sig. Proc. (ICASSP)*, May 2004, pp. IV–633—IV–636.

[7] J. Gao, L.J. Guibas, J. Hershberger, and L. Zhang, "Fractionally cascaded information in a sensor network," in *Proc. Third Int. Symp. Inf. Proc. in Sensor Networks (IPSN)*, Berkeley, California, USA, 2004, pp. 311–319.

[8] Lingxuan Hu and David Evans, "Localization for mobile sensor networks," in *Proc. 10th Ann. Int. Conf. on Mobile Comp. and Net. (MobiCom)*, 2004, pp. 45–57.

[9] "COMPASS project website," `http://compass.cs.rice.edu`, Rice Univeristy (Houston, TX).

[10] W. Sweldens, "The lifting scheme: A construction of second generation wavelets," *SIAM J. on Math. Anal.*, vol. 29, no. 2, pp. 511–546, March 1998.

[11] I. Daubechies, I. Guskov, P. Schröder, and W. Sweldens, "Wavelets on irregular point sets," *Phil. Trans. R. Soc. Lond. A*, vol. 357, no. 1760, pp. 2397–2413, 1999.

[12] V. Delouille, M. Jansen, and R. von Sachs, "Second generation wavelet methods for denoising of irregularly spaced data in two dimensions," Tech. Rep., Institut de Statistique, Université Catholique de Louvain, Belgium, January 2003.

[13] A. Okabe, B. Boots, K. Sugihara, and S.N. Chiu, *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*, Wiley, second edition, 2000.

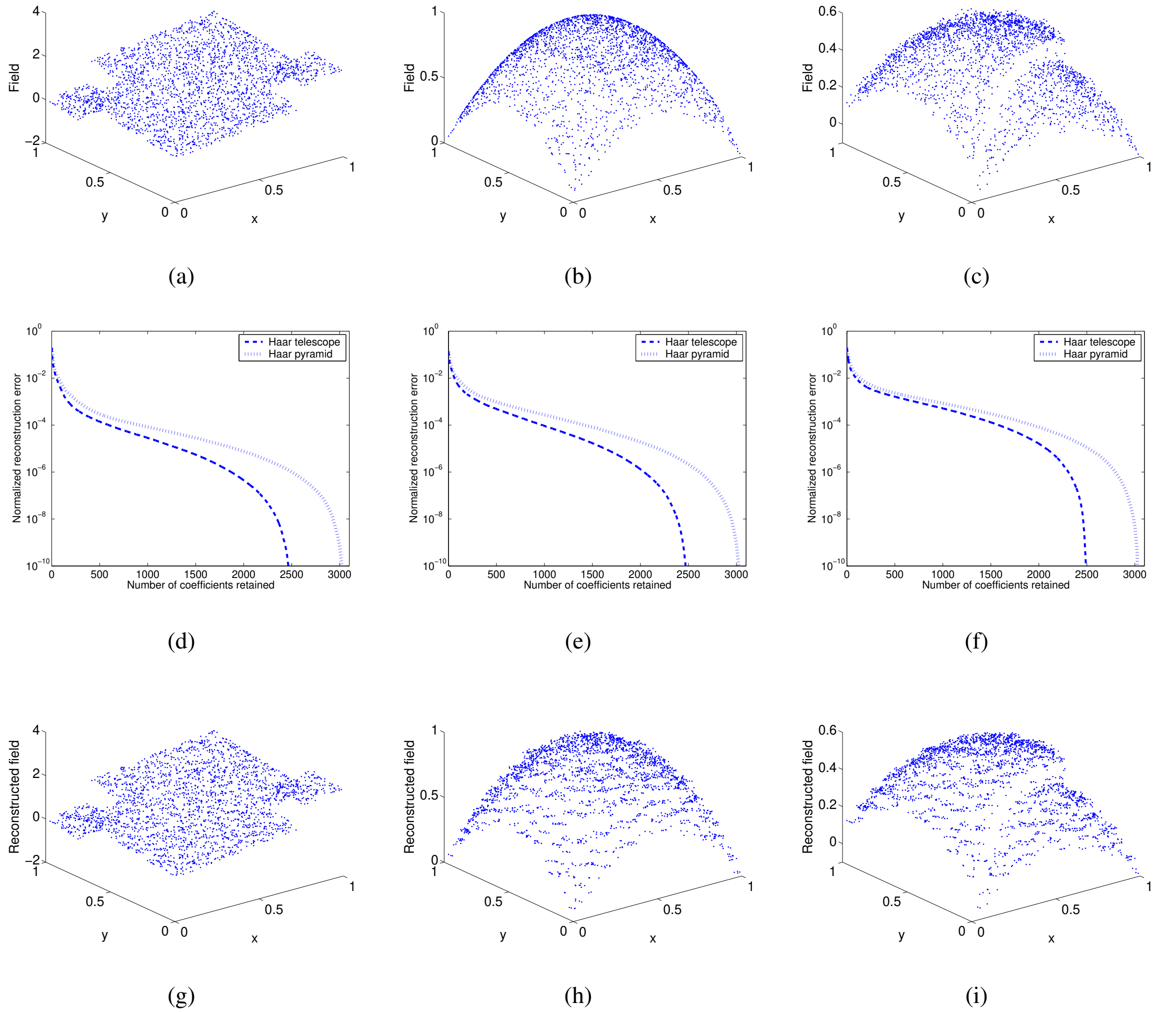

Fig. 9. Sample piecewise-constant (with a discontinuity) (a), quadratic (b), and noisy quadratic (with a discontinuity) (c) fields; approximation-error versus coefficient count curves (d),(e),(f); representative reconstructed fields (g), (h), (i)

[14] R. Wagner, S. Sarvotham, and R. Baraniuk, "A multiscale data representation for distributed sensor networks," in *IEEE Int. Conf. on Acoustics, Speech, and Sig. Proc. (ICASSP)*, March 2005, Accepted for publication.

[15] P.J. Burt and E.H. Adelson, "The Laplacian pyramid as a compact image code," *IEEE Trans. Comm.*, vol. 31, no. 4, pp. 532–540, April 1983.

[16] P.G. Casazza and J. Kovacevic, "Equal-norm tight frames with erasures," *Advances in Comp. Math.*, vol. 18, no. 2-4, pp. 387–430, February 2003.

[17] S. Sarvotham, R. Wagner, and R. Baraniuk, "A multiscale data representation for distributed sensor networks: Proofs of basis characteristics and error bounds," Tech. Rep., Rice University (Houston, TX), `http://cmc.rice.edu/docs/docinfo.aspx?doc=Sar2004Sep9AMultiscal`, September 2004.

[18] J. Edmonds, "Paths, trees, and flowers," *Canadian J. of Math.*, vol. 17, no. 3, pp. 449–467, 1965.

[19] F. Araújo and L. Rodrigues, "Fast localized delaunay triangulation," in *Proc. 8th Int. Conf. on Principles of Dist. Sys. (OPODIS)*, Grenoble, France, December 2004, (to appear).

[20] W. Cook and A. Rohe, "Computing minimum-weight perfect matchings," *INFORMS Journal on Computing*, vol. 11, pp. 138–148, 1999.