Volumetric Data Reduction in a Compressed Sensing Framework

X. Xu\textsuperscript{1}, E. Sakhaee\textsuperscript{1}, A. Entezari\textsuperscript{1}

\textsuperscript{1}University of Florida, USA

Abstract

In this paper, we investigate compressed sensing principles to devise an in-situ data reduction framework for visualization of volumetric datasets. We exploit the universality of the compressed sensing framework and show that the proposed method offers a refinable data reduction approach for volumetric datasets. The accurate reconstruction is obtained from partial Fourier measurements of the original data that are sensed without any prior knowledge of specific feature domains for the data. Our experiments demonstrate the superiority of surfactlets for efficient representation of volumetric data. Moreover, we establish that the accuracy of reconstruction can further improve once a more effective basis for a sparser representation of the data becomes available.

Categories and Subject Descriptors (according to ACM CCS):
I.4.2 [Image Processing and Computer Vision]: Compression—Approximate methods
I.4.5 [Image Processing and Computer Vision]: Reconstruction—Transform methods
I.4.10 [Image Processing and Computer Vision]: Image Representation—Volumetric

1. Introduction

Shannon’s sampling theorem has been the hallmark of information theory and signal processing and has had an enormous impact on applications. In many acquisition systems, the signal is often pre-filtered and the data is sampled at the Nyquist rate where exact reconstruction is guaranteed by the sampling theorem [Uns00]. The transform coding is involved after the acquisition where sparsity (in the transform-domain) is exploited for “feature”-based compression of the data. For example, wavelet-domain transform coding techniques have been successful in sparse feature-based representation of multidimensional (e.g., image, volumetric, time-varying) data. For images and higher dimensional data, there are several generalizations of wavelets that incorporate geometric structures for sparse feature representation such as curvelets [CDDY06], shearlets [KL12] and surfactlets [LD07].

The research efforts in signal processing are shifting the sample-then-compress paradigm to compressively sample the data. The emerging field of compressed sensing (CS) integrates the sparse transform-coding techniques to the classical sampling theory. Compressed sensing has had a significant impact in scientific and biomedical imaging: modalities such as MR [LDP07, LDSP08] and CT [PSV09] and scientific simulation experiments [Bar11] are adopting concepts from compressed sensing to exploit sub-Nyquist rates for acquisition. While there are significant efforts on 2D imaging (e.g., single-pixel camera project [DDT*08]), sparse representations for 3D volumetric data have not been studied in the context of compressed sensing.

In this paper we study the problem of sparse approximation in the context of volumetric data and propose using the (non-separable) surfactlets as a sparse reconstruction domain for compressed sensing in 3D. The key aspect of the proposed framework is the universality of the sensing process: the sensing is performed without a priori specification of the domain in which the data features are best described. The feature domain only enters the picture at the reconstruction stage. This implies that one can refine the notion of features (via integration of domain knowledge or learning mechanisms) and further improve the accuracy of reconstruction without having to repeat the acquisition process.

The prospects of studying sparse representations for volumetric data are quite attractive. For a dataset of size $N$ with a relatively small number, $k$, of volumetric features— with these features being none other than space-domain features or non-zero wavelet coefficients in the simplest case—the necessary number of samples is $O(k \log N)$: linearly...
proportional to $k$ and logarithmically proportional to the dataset size $N$. The remarkable logarithmic reduction from the “Nyquist-rate” sampling is the significant accomplishment of the compressed sensing paradigm.

The challenge is that reconstruction algorithms, unlike in the classical Shannon sampling theorem, are no longer linear and involve convex optimization and other iterative methods. Given the availability of abundant computing (e.g., multi-core/GPU) resources, non-linear methods may be no longer barriers to reconstruction. Specially considering the simulation environments where the supercomputer’s processing power keeps growing, the increase in dataset sizes is challenged by the I/O bottleneck. The proposed framework provides a smart in-situ data reduction mechanism in which the universality of the sensing process allows for refinements of reconstruction by further improving the definition of features after the data acquisition stage.

2. Merits of sparse modeling for volumetric data

2.1. Volumetric data reduction

With the growing scales of scientific simulations, challenges imposed by the limited performance of I/O modules can only be met by data reduction methodologies that are becoming a necessary pre-processing step [MWYT07]. The usual approach of saving one from every 300 to 500 time steps of the simulated data, or reducing the resolution, for storage results in significant waste of computational results [Ma09].

The assumption that most signals can be sparsely modeled in a certain feature domain, is central to data reduction or feature extraction. For example, most images when transformed to the wavelet domain, are representable by a sparse set of coefficients. The sparse representations or extracted features can be useful in recovering the data. In recent years, several compression-domain volume rendering approaches have been proposed [SW03, FM07, GGM12, SIGM*11], where the data decoding is combined with rendering such that data transfer to the GPU is reduced. In these approaches, the data is first sparsely represented, for example, by vector quantization [SW03], transform coding [FM07], dictionary learning [GGM12], or tensor approximation [SIGM*11]. During the rendering stage, only the features are transferred to the GPU and decoded on the fly.

The sparse modeling not only provides a flexible framework for many data processing applications [Ela10], it is also the fundamental building block of the compressed sensing theory discussed in this paper. Unlike the traditional data reduction and compression methodologies where the sparse modeling is considered as a priori, the CS framework only involves the feature domain in the decoding stage. In other words, the compressed sensing provides us a universal data reduction mechanism where we can refine the feature domain after the reduction.

2.2. Universal and reusable sensing

The common assumption that the data is bandlimited or can be sparsely represented by a dictionary is central to data reduction or feature extraction. Once the features are extracted or data is reduced, recovering the original data may not be possible. Besides, the knowledge of which feature domain fits the data may not be available at the in-situ processing stage [MWYT07]. This can be a limitation for exploration applications where the domain experts need to search and revise the features of interest. For such exploration applications, the loss of information during the in-situ data reduction stage means that the costly simulations have to be repeated in order to re-create the original data. This need for repeating simulations can be eliminated in the compressed sensing framework discussed in this paper.

As we will see in Section 3, the advantage of the CS framework is that at the pre-processing stage no prior information is required about the feature space – just the assumption that the data is sparse in some feature space. The information about features only enters the picture at the reconstruction phase. This means that once the data is sensed using the CS framework, we can continue to refine the definition of features through dictionary learning or other sources of domain-knowledge after the data reduction stage. This
flexibility makes our approach suitable for in-situ processing. With a refined notion of features, one only needs to rerun the reconstruction on the “old” CS data without having to access the original data from the high-resolution simulation results. This ‘forward-compatibility’ of the CS data reduction allows the domain-experts to further study the simulations to better define the features of interest in the simulation data. Fig. 1 depicts the proposed framework.

We note that the compressed sensing approach provides us a reusable data reduction mechanism but not a compression algorithm. To develop a data reduction mechanism (e.g., JPEG), further block splitting, quantization, and entropy coding are needed.

2.3. Contributions

Our primary interest is to exploit the universality of compressed sensing for the in-situ data reduction of large-scale volumetric datasets. Although the compressed sensing has been widely applied on image and video processing, it has not been adopted on volumetric – the real 3D data. In this sense, we extend the application of compressed sensing to the field of volumetric data processing. In our experiments, we have applied both the curvelet transform and the surfacelet transform on volumetric data, while the traditional usage of these transforms is mostly with video denoising. We are the first to propose using the surfaces as a sparse reconstruction domain for compressed sensing in 3D. Our experimental results motivate future research on the study of custom-designed sparse representations for large-scale volumetric data as well as efficient implementation of the sparse approximation algorithms.

3. Compressed sensing for volumetric data

We briefly discuss concepts from the compressed sensing (CS) theory that are pertinent to our data reduction framework. The works by Donoho [Don06] and Candès et al. [CT06, CRT06a] can be consulted for a more in-depth presentation. For clarity of presentation we summarize notions used in our discussion in Table 1.

Formally, let \( x \in \mathbb{R}^N \) be a vectorized volume dataset comprising \( N \) voxels. If most of the voxels are zero, then the volume represented by \( x \) is sparse. The sparsity is described by the so-called \( \ell_0 \)-(pseudo) norm, \(|x|_0\), that simply counts the number of non-zero elements in \( x \). For a \( k \)-sparse signal we have \(|x|_0 \leq k \). The sensing mechanism is linear as the measurement vector (i.e., the sample set) \( y \in \mathbb{R}^n \) is linearly related to \( x \) via an \( n \times N \) matrix \( A \):

\[
y = Ax,
\]

with \( n \ll N \). The term “reconstruction” refers to the task of recovering \( x \) from the measurement vector \( y \) through this under-determined system. For a provable accurate recovery the sensing matrix \( A \) needs to satisfy the Restricted Isometry Property (RIP) [CT05]: there exists a restricted isometry constant \( \epsilon \in (0, 1) \) such that

\[
(1 - \epsilon) ||x||^2_2 \leq ||Ax||^2_2 \leq (1 + \epsilon)||x||^2_2, \tag{2}
\]

holds for any \( k \)-sparse vector \( x \), where \( ||.||_2 \) represents a \( \ell_2 \) norm. Intuitively, the restricted isometry implies that in a valid sensing matrix \( A \), every possible set of \( k \) columns of the matrix \( A \) forms an approximately orthogonal set. Matrices that have been proven (probabilistically) to meet RIP include partial Fourier or cosine matrices (i.e., with randomly selected rows from the full discrete Fourier or cosine transform matrix), Gaussian and Bernoulli random matrices (with elements i.i.d. drawn by normal and Bernoulli distributions, respectively) [CT05, CT06].

The reconstruction (recovery) mechanism that is associated with this sensing, is non-linear and can be formulated as an optimization problem (also called sparse approximation problem [CDS98]):

\[
\min_{x} ||x||_0 \quad \text{subject to} \quad Ax = y. \tag{3}
\]

When \( n \geq 2k \) this optimization problem exactly recovers the original \( x \) [CRT06a, Don06]. This means that, in theory, we can solve for the under-determined system uniquely by searching for the sparsest \( x \) that satisfies the sensing constraints.

3.1. Sparse approximation

It is demonstrated that solving (3) is an NP-hard problem [CRTV05] for a non-trivial \( A \). However, it turns out that with increasing the number of samples to \( n = \mathcal{O}(k \log N) \) (with \( k = ||x||_0 \)), we can find polynomial-time alternatives [CRT06b]. Two commonly used approaches to the optimization problem (3) are greedy methods and convex relaxation methods. These two approaches have provably correct solutions when \( A \) meets the RIP conditions [TW10].

The idea behind greedy methods (e.g., orthogonal matching pursuit [TG07]) is that they iteratively refine the current

<table>
<thead>
<tr>
<th>Table 1: Notation Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>( N )</td>
</tr>
<tr>
<td>( k )</td>
</tr>
<tr>
<td>( n )</td>
</tr>
<tr>
<td>( \rho )</td>
</tr>
<tr>
<td>( x )</td>
</tr>
<tr>
<td>( y )</td>
</tr>
<tr>
<td>( A )</td>
</tr>
<tr>
<td>( \omega )</td>
</tr>
<tr>
<td>( \Psi )</td>
</tr>
<tr>
<td>( \delta )</td>
</tr>
<tr>
<td>( \mu )</td>
</tr>
</tbody>
</table>

© 2014 The Author(s)
Computer Graphics Forum © 2014 The Eurographics Association and John Wiley & Sons Ltd.
estimate of the sparse signal by modifying one or several coefficients such that the modification yields a better approximation of the signal. The iteration continues until all non-zero coefficients are found or a stopping criterion is reached. Greedy methods are viable for problems where the target signal is extremely sparse. This is due to the fact that greedy methods only bring a limited number of non-zero entries of $x$ at a time. Therefore, the number of iterations increases linearly in terms of the signal density.

It has been proven that optimal or near-optimal solutions to sparse approximation problems can be achieved using convex relaxation methods in a variety of settings [TW10]. Once the measurement matrix $A$ satisfies RIP, the highly non-convex optimization problem (3) can be relaxed to its convex counterpart [CRT06a,Don06]:

$$\min_{x} \|x\|_1 \text{ such that } Ax = y. \quad (4)$$

This convex optimization problem is also known as the Basis Pursuit method [CD98] in the literature. Compared to greedy methods, convex relaxation algorithms usually have better performance (in terms of accuracy and robustness) when the signal is not very sparse or noise is present.

### 3.2. Transform Domain Sparsity

Many signals may not necessarily be sparse in the canonical space domain. However, most of the natural signals are sparse in certain transform domain. For example, wavelets and their generalizations provide generic sparsifying transforms for natural signals. Many transform-coding compression systems such as JPEG 2000 and MPEG actually utilize this fact that wavelet transforms of many signals lead to sparse or compressible representations, that is, most of the transform coefficients are zero or very close to zero. Let $\Psi$ denote the change of basis matrix that relates a signal $x$ to its, say, wavelet representation: $\Psi x = \hat{x}$. Here, the matrix $\Psi$ represents a discrete wavelet transform and $\hat{x}$ denote the wavelet coefficients. We can integrate the sparse transform-domain representation into the CS framework (3) and reformulate the optimization in the transform-domain:

$$\min_{x} \|\Psi x\|_0 \text{ subject to } Ax = y. \quad (5)$$

The convex counterpart of (5) becomes

$$\min_{x} \|\Psi x\|_1 \text{ subject to } Ax = y. \quad (6)$$

We note that the measurement vector $y$ is obtained via $y = Ax$ without knowledge of the transform domain, $\Psi$. In other words, the acquisition is universal: it is performed without knowing which $\Psi$ sparsifies the signal $x$. This implies that whenever we have a better description of features of interest (a new $\Psi'$ through learning, future x-lets or domain knowledge that provides a dictionary of features), we can re-use the same sample set $y$ and replace $\Psi$ in the optimization framework (5), as depicted in Fig. 1. In this paper we demonstrate the suitability of this feature of the CS framework for volumetric data. By choosing suitable transform domain, we can increase the accuracy of reconstruction, or reduce the number of necessary measurements.

### 4. Experiments and discussion

In this section we present our framework of volumetric compressed sensing and discuss its advantages for data reduction. We demonstrate that with a small set of random discrete cosine transform (DCT) measurements, we can recover a volumetric dataset accurately. Our experimental results suggest that the surfacelet transform is particularly efficient for sparse approximation of volumetric datasets.

We also compare our volumetric compressed sensing framework with commonly used data reduction techniques such as run-length encoding and downsampling. We demonstrate that the performance of data reduction and reconstruction accuracy of the proposed framework is comparable to or often better than those of the run-length encoding and downsampling approaches.

The main theme we investigate in our experiments is the notion of universality of sensing, that is, with a sparser representation of the data $x$, we can re-use the sensed data $y$, to further refine the reconstructions—hence improving the accuracy and fidelity of visualization. Unlike other data reduction techniques (e.g., downsampling), the proposed framework provides a flexible solution to data reduction of generic kinds of volumetric datasets.

### 4.1. Sparsifying transform selection

Although they are a common choice for sparse representations, wavelets are in fact only optimal for approximating (multivariate) data with pointwise singularities. Wavelets are unable to handle singularities along curves or surfaces efficiently [KL12]. This is due to the isotropic characteristic of wavelets, that is, they are obtained by isotropically dilating a single or finite set of generating functions. Therefore wavelets lack directional sensitivity, and are unable to detect the geometry (i.e., singularities) of multivariate data well. The difficulty with representing geometric singularities is present in Haar as well as higher-order (e.g., Daubechies) wavelets. For volumetric data, the singularities introduced by surface boundaries are usually present. When using wavelets to sparsely represent volumetric objects, these singularities require many coefficients in the representation to accurately capture them, leading to suboptimal sparse representations.

There are a variety of geometric extensions of wavelets such as curvelets [CDDY06], shearlets [KL12] and surfacelets [LD07], exploiting the geometry of the multivariate data for optimally sparse representations. Unlike conventional wavelets, these so-called X-lets usually allow anisotropic scaling, shearing, and rotation of basis elements.
such that these bases can capture object structures more efficiently. A comprehensive list of these directional decomposition methods is available online [Duv13]. A caveat of using these directional transforms is their redundancy factors (i.e., the number of transformed coefficients over the number of signal elements). A high redundancy factor can make the problem size too large to handle, which becomes especially problematic when processing volumetric data.

In addition to wavelets, we exploit both the curvelet transform and the surfacelet transform for their adaptivity to features in volumetric data and their relatively low redundancy factors. The 3D curvelet transform has a redundancy factor of approximately 25 while the 3D surfacelet transform has a redundancy factor of approximately 4. To reduce the redundancy factor of the 3D curvelet transform, the original paper [CDDY06] suggests to use a mix representation where wavelets (instead of curvelets) are used at the finest scale. The new version of the curvelet transform has a redundancy factor of approximately 5 but its directional selectivity of fine details is significantly reduced. Therefore, the low redundant curvelet transform is more suitable for bandlimited signals [LD07]. In our experiments we have used the low redundant version of the curvelet transform to keep the problem size in a manageable scale.

4.2. Implementation details

For large image and volumetric data, the matrix $A$ (or $\Psi$) is prohibitively large and it cannot be loaded in main memory. Instead, the matrix $A$ can be represented by storing the indices to the random wave modes selected for measurement (i.e., $\Omega$ contains the indices of the random DCT rows). For example, in our implementation of the volumetric compressed sensing in (1), the matrix-vector multiplications (i.e., $Ax$) can be efficiently implemented by applying the (3D) DCT on $x$, retaining the DCT coefficients corresponding to the desired measurement indices $\Omega$. Moreover, when we choose $\Psi$ to be the wavelet (or curvelet, surfacelet) transform matrix, we usually have fast algorithms to perform these transforms and the algorithms may even benefit from parallel computing, matrix-vector multiplications in this case can be computed efficiently. The approach has a very small memory footprint (size of $\Omega$).

The acquisition and reconstruction frameworks described in Section 3 are implemented in C++ and MATLAB for our experiments on volumetric datasets. In our experiments, we have used FFTW library [FJ13] to perform the parallel DCT transform on the volumetric data. During the acquisition step, after applying DCT on the ground truth dataset, we keep a small number of base (DC) frequencies (less than 0.5% of the coefficients) and randomly pick the rest of measurements, thus obtain a measurement vector $y$ of length $n$. We define the sampling rate as $\rho := n/N$, the number of kept measurements over the total number of voxels of a dataset.

We examined the reconstruction process for two cases: a) the features are in the canonical domain (i.e., the dataset is sparse) where the reconstruction follows (3); and b) the features are in a transform domain where the reconstruction follows (5). Experiments involving sparsity in transform domains are carried out in the Haar wavelet domain, the curvelet domain and the surfacelet domain. We have implemented the Haar wavelet transform in an efficient and parallel manner (using OpenMP [Boa13]) and adopted the CurveLab toolbox [CDDY07] and the Surfacelet toolbox [Lut08] to perform the curvelet transform and the surfacelet transform, respectively. We note that the curvelet and surfacelet transforms provided by the toolboxes do not run in parallel. Therefore, performing the curvelet transform or the surfacelet transform takes significantly longer time than performing the wavelet transform (see Section 4.4).

The basis pursuit solver NESTA [BB11], which solves the problem of the form (6), is chosen for sparse approximation in our experiments. It is a first-order method and uses a smoothing technique that replaces the non-smooth $\ell_1$ norm with a smooth term [BBC11]. The smoothed version of the $\ell_1$ norm involves a smoothness parameter $\mu$. In general, $\mu$ should be small for high accuracy or large for faster performance; when $\mu = 0$, no smoothing is applied. A suitable value of $\mu$ balances the trade-off between recovery accuracy and time, which can be chosen by performing some trial experiments. The actual values of $\mu$ used in our reported results are specified in the following sections. The stopping criterion of the solver is controlled by a tolerance parameter $\delta$, which enforces the fidelity of the solution. We set a small tolerance value ($\delta = 1 \times 10^{-5}$) in our experiments.

We also compare the proposed compressed sensing (CS) framework with run-length encoding (RLE) and downsampling (DS), both of which are implemented in MATLAB. In the case of downsampling, we reduced the sampling rate by a factor of two in each dimension, resulting in a downsampling rate of $\rho = 12.5\%$. We used linear as well as cubic splines for the choice of interpolation filters for downsampling, which are denoted by “DS-linear” and “DS-cubic”, respectively in our experiments. The effective sampling rate of RLE depends on the dataset.

The datasets examined in our experiments include Aneurysm, Hydrogen, Supernova and Head Aneurysm. The Supernova datasets have been examined at multiple time steps. Our experiments are carried out on a workstation with AMD Opteron(TM) Processor 6274 2.20GHz CPU and 16GB main memory. The maximum number of parallel CPU threads allowed in our experiments is 32. The accuracy of reconstruction is evaluated numerically in terms of the Signal to Noise Ratio (SNR) which is measured in logarithmic scale (dB) over the entire volumetric data. We also provide volume rendering images of both ground truth and recovered data for visual comparison.
4.3. Sparse dataset – Aneurysm

The Aneurysm dataset has been sampled at a resolution of 256 × 256 × 256, of which only 1% of data elements are non-zero. RLE of the Aneurysm dataset yields a sampling rate of ρ = 2.41%. Fig. 2 shows the accuracy of the sparse approximation with different sampling rates, and the time used for recovery of the Aneurysm dataset (μ = 5 × 10^{-3} for the solver). We can see that increasing the sampling rate improves the reconstruction accuracy; however, the improvements become less significant at higher sampling rates. With a sampling rate as low as ρ = 2.5%, sparse approximation can achieve a reconstruction with SNR 42.27 dB. As the number of measurements increases, the reconstruction time decreases due to faster convergence of the solver. When the sampling rate increases to ρ = 12.5% (not shown in the plot), sparse approximation can yield an SNR of 78.32 dB in less than 2 minutes.

Fig. 3 shows the volume rendering images, as well as SNR’s, of the ground truth Aneurysm data and the recovered data from CS measurements and the downsampled dataset of size 128 × 128 × 128. With the same sampling rate as RLE, sparse approximation can result in reconstruction with SNR = 36.35 dB. In contrast, downsampling yields a much higher sampling rate and interpolating the downsampled data leads to poor recovery results. For example, some vessels are missing (e.g., the dotted areas) in the interpolated datasets.

From our experiments on the Aneurysm dataset, we can observe that RLE is extremely efficient for data compression of truly sparse, noise free, datasets. However, RLE is only suitable for datasets with large homogeneous areas and fails to compress data with complex structures and is susceptible to noise. Meanwhile, the proposed framework provides a universal sampling process for various kinds of datasets. With a small number of compressive measurements, we can accurately recover the dataset. We will examine a high resolution aneurysm dataset in Section 4.5.

4.4. Transform domain sparsity

We now experiment with datasets that are non-sparse in the canonical space domain but (approximately) sparse in certain transform domain using the approach discussed in Section 3.2. Since the voxel value of the dataset varies, RLE fails to compress the dataset (ρ ≈ 200%) and is not a suitable choice for data reduction here. We examine the wavelet domain, as the common choice for sparse representation. Moreover, to demonstrate the universality of the sensing process, we also examine the reconstruction in the curvelet domain and the surfacelet domain, and establish their utility for more accurate reconstruction from the same measurements that were provided for the wavelet framework.

4.4.1. Hydrogen

The ground truth of the Hydrogen dataset has a resolution of 128 × 128 × 128. Fig. 4 plots the sparse approximation results of the Hydrogen dataset exploiting the wavelet domain, the curvelet domain and surfacelet domain for different sampling rates. The results shown in Fig. 4 are obtained from NESTA with μ = 1 × 10^{-2}. From the plot, we can see that as the sampling rate increases, the reconstruction accuracy improves. We observe that for various ranges of sampling rates, changing the sparsifying domain can improve the sparse approximation performance. Surfacelets yield the most accurate reconstruction among the three domains used.

Fig. 5 shows the volume rendering images of the sparse approximation datasets for the case of ρ = 12.5% as well as
the datasets reconstructed from interpolation of the downsampled dataset. We see that the downsampling works well for the Hydrogen dataset. The sparse approximation using wavelets does not return a faithful reconstruction. Using curvelets, the sparse approximation improves. When choosing surfaclets for the sparse approximation, high accurate recoveries are achieved. We also see that the sparse approximation tends to smooth the dataset because the sparse approximation recovers the most significant (usually low frequency) coefficients while leaves other coefficients (usually high frequency) as zeros.

The average time used for sparse approximation shown in Fig. 4 is 1.5 minutes when using wavelets, 6 minutes when using surfaclets, and 23 minutes when using curvelets. Reconstructions using curvelets and surfaclets are slower than using wavelets is due to the non-parallel implementation. For example, performing a pair of forward and backward wavelet, surfaclet and curvelet transforms on the Hydrogen dataset requires about 0.06 seconds, 1.6 seconds and 10 seconds, respectively.

4.4.2. Supernova

The Supernova datasets include 60 time steps (ranging from 1295 to 1354). Each time step of the ground truth data has a resolution of \(216 \times 216 \times 216\). Table 2 illustrates results (SNR and timing) of the sparse approximation \((\mu = 5 \times 10^{-4})\) of the Supernova datasets of time steps 1305 and 1345. Fig. 6 shows volume rendering images of the sparse approximated datasets at time step 1345 when the sampling rate is \(\rho = 12.5\%\). The interpolation results from the downsampled dataset are also shown in Fig. 6.

We see from the table that in most cases the sparse approximation with surfaclets yields the highest SNR (i.e., on a logarithmic scale). However, the SNR difference between wavelets and surfaclets on sparse approximation drops as the number of measurements increases. For time step 1305, sparse approximation using wavelets even yields higher SNR’s than sparse approximation using surfaclets when the sampling rates are above 35%. This indicates that as long as a sufficient number of measurements are provided, sparse approximation using wavelets can also return accurate reconstructions.

For the Supernova datasets the performance of curvelets for sparse approximation does not necessarily better than that of wavelets. This is because the Supernova datasets present more details and the (low redundant) curvelet transform used is unable to capture the directional information of fine details efficiently. The reconstruction accuracy of the sparse approximation may be improved by setting smaller \(\mu\) values but with increasing reconstruction time.

Sparse approximation using curvelets or surfaclets takes much more time than sparse approximation using wavelets, which is caused by the slowness of the curvelet transform and the surfaclet transform. Performing a pair of forward and backward wavelet, surfaclet and curvelet transforms on one Supernova dataset requires about 0.6 seconds, 9.3 seconds and 39 seconds, respectively.

From the images shown in Fig. 6, we can observe an overall accuracy of the sparse approximation with these three transform domains. However, obvious artifacts exist in the dataset recovered with wavelets and the inside red structures...
Table 2: The accuracy (in dB, on the left) and recovery time (in minute, on the right) of the sparse approximation of the Supernova datasets (time steps 1305 and 1345) using wavelets (W.), curvelets (C.) and surfacelets (S.).

<table>
<thead>
<tr>
<th>ρ</th>
<th>10%</th>
<th>12.5%</th>
<th>15%</th>
<th>20%</th>
<th>25%</th>
<th>30%</th>
<th>35%</th>
<th>10%</th>
<th>12.5%</th>
<th>15%</th>
<th>20%</th>
<th>25%</th>
<th>30%</th>
<th>35%</th>
</tr>
</thead>
<tbody>
<tr>
<td>1305</td>
<td>W.</td>
<td>21.27</td>
<td>22.61</td>
<td>24.05</td>
<td>26.67</td>
<td>29.20</td>
<td>31.51</td>
<td>33.59</td>
<td>3.3</td>
<td>3.1</td>
<td>3.2</td>
<td>3.0</td>
<td>2.9</td>
<td>2.8</td>
</tr>
<tr>
<td></td>
<td>C.</td>
<td>21.09</td>
<td>21.79</td>
<td>22.42</td>
<td>23.61</td>
<td>24.69</td>
<td>25.75</td>
<td>26.77</td>
<td>60.8</td>
<td>61.2</td>
<td>61.2</td>
<td>60.1</td>
<td>58.7</td>
<td>58.1</td>
</tr>
<tr>
<td></td>
<td>S.</td>
<td>25.11</td>
<td>26.08</td>
<td>26.98</td>
<td>28.69</td>
<td>30.30</td>
<td>31.84</td>
<td>33.32</td>
<td>26.7</td>
<td>25.0</td>
<td>23.7</td>
<td>22.6</td>
<td>21.8</td>
<td>21.0</td>
</tr>
<tr>
<td>1345</td>
<td>W.</td>
<td>20.69</td>
<td>21.75</td>
<td>22.94</td>
<td>25.14</td>
<td>27.23</td>
<td>29.22</td>
<td>31.10</td>
<td>3.3</td>
<td>3.1</td>
<td>2.9</td>
<td>2.8</td>
<td>2.7</td>
<td>2.9</td>
</tr>
<tr>
<td></td>
<td>C.</td>
<td>22.23</td>
<td>22.88</td>
<td>23.47</td>
<td>24.56</td>
<td>25.55</td>
<td>26.53</td>
<td>27.47</td>
<td>60.8</td>
<td>60.9</td>
<td>61.0</td>
<td>60.4</td>
<td>59.0</td>
<td>57.7</td>
</tr>
<tr>
<td></td>
<td>S.</td>
<td>25.74</td>
<td>26.58</td>
<td>27.35</td>
<td>28.79</td>
<td>30.16</td>
<td>31.52</td>
<td>32.84</td>
<td>25.2</td>
<td>25.4</td>
<td>24.2</td>
<td>23.1</td>
<td>22.1</td>
<td>21.2</td>
</tr>
</tbody>
</table>

are missed in the dataset recovered with curvelets, while reconstruction using surfacelets generates clearer and more accurate result. The sparse approximation using wavelets or curvelets does not perform better than interpolation from downsampled dataset. However, by exploiting the surfacelet transform domain, we can improve the reconstruction by reusing the existing measurements and achieve higher recovery accuracy than the interpolation methods. We will examine a high resolution supernova dataset in Section 4.5.

Our experiments on Hydrogen and Supernova show that it is possible to improve the sparse approximation by exploiting more effective sparsifying transforms while re-using the same measurements. We note that after reducing the sampling rate to extremely low levels, we may not achieve an accurate reconstruction from the sparse approximation. The required number of measurements for accurate reconstruction depends on how we can sparsely represent the data. The sparser we can represent the data, the fewer measurements are required for accurate recovery. In other words, while re-using the existing measurements, the sparser we can represent the data, the more accurate reconstruction we can achieve. To obtain an accurate reconstruction from a very low sampling rate, optimal sparse representations of the data are desired. Such optimal sparse representations are usually not available during the in-situ data reduction stage, but may be obtained/refined with the domain knowledge at a later stage. The compressed sensing provides us such a refinable reconstruction framework.

4.5. Large-scale volumetric datasets

As the size of datasets grows, applying compressed sensing and then sparse approximation on the whole dataset is not a wise choice due to the increasing problem size. A tiling strategy, as adopted in the JPEG 2000 standard, may be used. For a large-scale volumetric dataset, we can partition the dataset into non-overlapping blocks of the same size. For each block (i.e., a sub-volume), we can obtain the compressive measurements separately. At the reconstruction stage, the sub-volumes can be recovered in parallel since the sparse approximation of each sub-volume is independent. In this section, we demonstrate that applying such a strategy to the proposed framework is feasible. Sophisticated partitioning strategies such as the one with overlapping areas may be applied to reduce potential artifacts at the tile borders, but we leave it as our future work.

We consider the Head Aneurysm dataset of size 512 × 512 × 512. The volume is divided into eight sub-volumes and each sub-volume has a resolution of 256 × 256 × 256. For each sub-volume, we maintain a CS sampling rate of ρ = 12.5% such that we also have an overall sampling rate of ρ = 12.5% for the whole (combined) volume. The sparse approximation of each sub-volume is done in parallel. All (sub) sparse approximation process have the same parameter setting (µ = 5 × 10⁻³). The eight (sub) sparse approximation processes run simultaneously, and each process is allowed to fork at most eight parallel CPU threads. Note that since the

![Figure 6: Sparse approximation and interpolation of the Supernova dataset (ρ = 12.5%).](image-url)
dataset is sparse by itself, no sparsifying transform is applied for the reconstruction.

The sparse approximation of the Head Aneurysm dataset finishes in 5.5 minutes and yields an SNR of 76.56 dB. Fig. 7 compares the rendering images and SNR’s of the recovered datasets from the sparse approximation and the cubic interpolation of the downsampled dataset. We observe visually and numerically, the sparse approximation outperforms the interpolation method. The sparse approximation leads to an almost identical image as the ground truth image.

Similar experiments have also been conducted on the high resolution Supernova datasets, where the resolution is $432 \times 432 \times 432$ and each time step is partitioned into eight blocks of equal size. Fig. 8 compares the recovered datasets (time step 1345) from the sparse approximation ($\mu = 1 \times 10^{-4}$) using surfaclets and the cubic interpolation of the downsampled dataset. The sparse approximation in this case takes 40 minutes. We observe the sparse approximation outperforms the interpolation method in terms of accuracy. The accuracy of the sparse approximation can be further improved once sparser representations of the dataset are discovered. Note that we have also conducted the experiment ($\rho = 12.5\%$) on the entire time step 1345 (i.e., without tiling); the sparse approximation using surfaclets takes 2.3 hours and yields an SNR of 34.12 dB.

The proposed framework, at the current stage, is limited by the long recovery time. This is due to the high complexity of the non-linear sparse approximation algorithms as well as inefficient implementation of the sparsifying transforms. However, such computational costs can be mostly mitigated by the availability of parallel computing resources such as GPUs. By profiling of our CPU implementation, we observed that, on average, more than 90% of the time was simply spent on performing transformations (i.e., FFT, X-let transforms). Efficient implementations of these transforms that utilize GPU computing have been proposed in recent years [GLD+08, FBFU10, MSMR13]. These implementations allow reducing the recovery time by orders of magnitude. Moreover, new FFT algorithms that exploit the sparsity of the signal (e.g., sparse FFT [HIKP12]) reduce the computational complexity of this transform (e.g., from $O(N \log N)$ to $O(k \log N)$ for a $k$-sparse signal). To translate the advantages of the proposed framework into practical setting, in our future work, we will investigate its GPU implementation.

5. Conclusions

In this paper, we apply the compressed sensing to the field of data reduction for volumetric datasets. Acquiring only a few random measurements of the signal as a part of in-situ processing leads to a significant decrease in the data storage and I/O requirements. The acquisition is universal, that is, the sensing process requires no prior knowledge of features in the signal. Using the compressed measurements we can recover the original data with high accuracy. We build a case for and demonstrate the significance of sparsity in volumetric data processing. Specifically, we show the utility of surfaclets for sparse modeling of volumetric data. Our results motivate future research on the study of custom-designed sparse representations and sparse approximation algorithms for large-scale volumetric data.

Acknowledgment

The Aneurysm, Hydrogen, and Head Aneurysm datasets are courtesy of http://www.volvis.org. The Supernova datasets are courtesy of http://vis.cs.ucdavis.edu/VisFiles/index.php. This work is supported, in part, by the ONR grant N000141210862 and the NSF grant CCF-1018149.
References


