

3D spatio-temporal analysis for compressive sensing in magnetic resonance imaging of the murine cardiac cycle

Brice Hirst^a, Yahong Rosa Zheng^a, Ming Yang^b, and Lixin Ma^b

^aMissouri University of Science and Technology, Department of Electrical and Computer Engineering;

^bUniversity of Missouri, Department of Radiology, Nuclear Science and Engineering Institute, and Harry S. Truman Memorial Veteran's Hospital.

ABSTRACT

This paper explores a three-dimensional compressive sensing (CS) technique for reducing measurement time in magnetic resonance imaging (MRI) of the murine (mouse) cardiac cycle. By randomly undersampling a single 2D slice of a mouse heart at regular time intervals as it expands and contracts through the stages of a heartbeat, a CS reconstruction algorithm can be made to exploit transform sparsity in time as well as space. For the purposes of measuring the left ventricular volume in the mouse heart, this 3D approach offers significant advantages against classical 2D spatial compressive sensing.

Keywords: Compressive sensing, MRI, three-dimensional, random sampling, mouse heart

1. INTRODUCTION

For many imaging methods (such as video recording), data acquisition is fast enough and cheap enough that acquiring fully (i.e. Nyquist rate or greater) sampled sets of data is not a problem. However, taking full sets of measurements with other imaging methods can be costly in several ways. For example, a camera CCD sensor array may be very cheap to produce, but microwave imaging requires expensive sensors which generally take a long period of time to acquire data.¹ In magnetic resonance imaging, an image is created by using radio-frequency (RF) pulses to spin the protons in living tissue at different frequencies and phases, then reading the total RF emissions from all of them combined, creating a frequency-based measurement domain known as “k-space”. A full set of data can take a long time to acquire, which can cause unacceptable motion artifacts in cases such as real-time imaging of the mouse cardiac cycle. By reducing the number of measurements, compressive sensing can reduce either the amount of imaging time or the number of sensors, or both, depending on the imaging method. For preclinical MRI, the prospect of reduced measurement time is of great interest to the field of medicine because:

1. An MRI procedure is expensive. Less measurement time = less money spent.
2. Fast-moving targets (such as mouse hearts) exhibit less severe motion artifacts when measurement times are shorter.
3. A shorter measurement time means less exposure to magnetic & RF energy for the animal under test.
4. Preclinical research can be completed faster when more scans can be made in the same period of time.

Further author information

Brice Hirst - E-mail: bahrkb@mst.edu, Telephone: 1-573-368-9504

Dr. Zheng - E-mail: zhengyr@mst.edu, Telephone: 1-573-341-6632

Ming Yang - E-mail: my5f2@mail.missouri.edu, Telephone: 1-573-814-6000

Dr. Ma - E-mail: mal@health.missouri.edu, Telephone: 1-573-814-6000

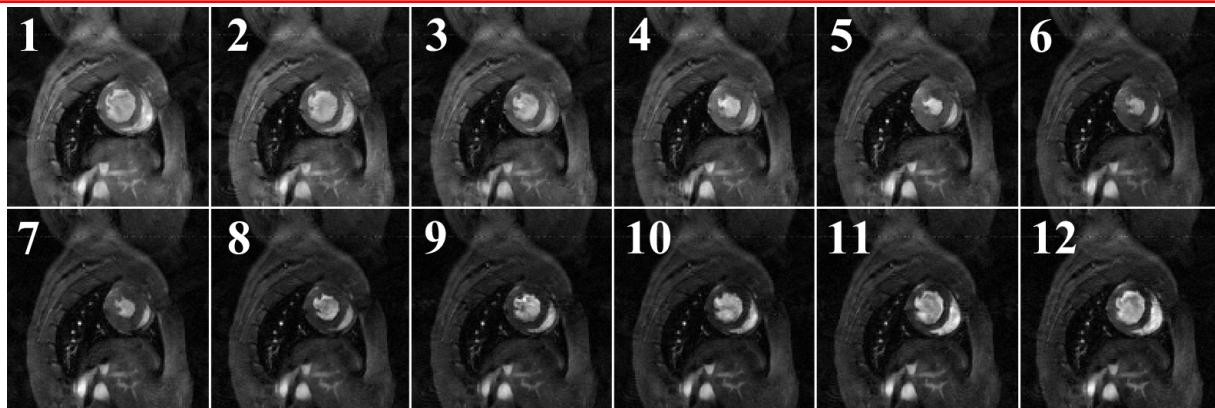


Figure 1. Sample cardiac cycle for a mouse heart. (Frames 1-6) The heart contracts in the systolic phase until it reaches end-of-systolic (minimum) volume. (Frames 7-12) The heart expands in the diastolic phase until the end-of-diastolic (maximum) volume is reached again. The horizontal line of interference near the top of each image was caused by RF noise entering the MRI machine.

An example of a complete cardiac cycle for a mouse is shown in Figure 1, with the important stages labeled. The primary diagnostic goal of these images is to estimate the volume of the left ventricular cavity (the prominent circular white region in each frame of Figure 1) as the heart beats.² A notable feature of this sort of measurement is that the areas of the heart surrounding this cavity do not change much with time; hence, they will be sparse in any time-based transform domain that can exploit this, such as the discrete Fourier transform (DFT) domain. By applying a classical 2D sparsifying transform such as the discrete cosine transform (DCT) or discrete wavelet transform (DWT) in the x-y domain, and subsequently applying a DFT in the time domain, the performance over the entire “video” will be improved after a 3D CS reconstruction is applied as compared to the case when a 2D CS reconstruction is applied to each frame individually.

2. THE COMPRESSIVE SENSING METHOD

The concept of compressive sensing is heavily rooted in information theory. The important “information” in an image can be condensed into fewer samples after a transformation is applied; CS is a method that exploits this “transform sparsity” to capture most of this condensed information directly by sub-Nyquist sampling. Randomness and sparsity play heavy roles in the quality of image reconstruction from this undersampled data in both 2D and 3D methods.

2.1 Compressive Sensing Background

The premise of compressive sensing centers highly upon the concept of *sparsity*. Consider a signal $\mathbf{x} \in \mathbb{C}^n$, where n is the full number of samples in the signal. A signal is considered to be S -sparse if only S of the samples are significant (i.e. above the noise floor of the image). In general, the lower the ratio S/n , the more compressible a signal is and hence the better quality the reconstruction will be when CS is applied. Now, many signals are not very sparse in their original form, so image reconstruction in CS can make use of a *sparsifying transform*. A sparsifying transform can be defined as an orthogonal basis $\Psi^{n \times n}$ such that:

$$\mathbf{s} = \Psi \mathbf{x}, \quad (1)$$

where $\mathbf{s} \in \mathbb{C}^n$ is the transformed signal. If the basis Ψ is chosen correctly, the resulting signal \mathbf{s} will be much sparser than the original signal \mathbf{x} . The best sparsifying transform Ψ for these purposes is highly dependent on the data; for instance, the JPEG compression scheme uses a block-based discrete cosine transform to sparsify data because its target, photographic images, is well suited to compressibility with this transform.

In standard notation, Ψ is defined as a matrix which, if all of the pixels in the 2D image are arranged into a single dimension to form \mathbf{x} , can be multiplied by \mathbf{x} to find the sparse result \mathbf{s} . Indeed, all sparsifying transforms

can be expressed this way in matrix form if desired, but this is not advised; the main reason is because most common sparsifying transforms are computationally reducible (i.e. they can be computed a special way that takes much less time than a straight matrix multiplication). In this case, matrix multiplications for large values of n are unwieldy and unnecessary; for example, if the DFT is the sparsifying basis, a straight DFT matrix multiplication has a computational complexity of $O(n^2)$, whereas the fast Fourier transform (FFT) method can do it with a complexity of $O(n \log n)$.³ Computational advantages apply to all of the transforms used in this paper; henceforth, Ψ will instead be defined as a function of \mathbf{x} for all subsequent purposes.

Using only a small subset of the k-space data points, it is our goal to approximate the fully sampled MRI image as accurately as possible. Let $\mathbf{x} \in \mathbb{C}^n$ represent the full reconstructed image (in the pixel domain), where the visual representation of the image is the magnitude of the complex data, $|\mathbf{x}|$. Let $\mathbf{y} \in \mathbb{C}^m$ represent the randomly undersampled k-space points. In standard MRI, $m = n$, and so the process of moving from k-space to the image domain is a simple matter of a 2D Fourier Transform on each individual video frame in question. However, when $m \ll n$, the reconstruction necessitates the solution of an underdetermined system for \mathbf{x} . A naive approach would involve simply filling in all of the unsampled k-space points with zeroes to create a full set of k-space data with n points, then performing the Fourier Transform as before. The problem with this is that zero-filling in this manner tends to create coherent aliasing, or noise that looks like multiple periodic superimposed copies of the image.⁴ Using the concept of transform sparsity, we can create a much more accurate image reconstruction using the same points in k-space, provided we know of a transform domain where the image would be relatively sparse. In order to take advantage of this sparsity, we would like to minimize the following equation:

$$\min f(\Psi\mathbf{x}) \text{ subject to } \mathbf{Ax} = \mathbf{y}, \quad (2)$$

where $\mathbf{A}^{m \times n}$ is an undersampled inverse Fourier Transform matrix, and $f(\Psi\mathbf{x})$ is a function that calculates a specific sparsity measure from the sparse representation of \mathbf{x} . The standard sparsity measure used in compressive sensing is the L_1 norm, defined as:

$$\|\mathbf{s}\|_1 = \sum_{i=1}^n |s_i| \text{ for all } \mathbf{s} \in \mathbb{C}^n \quad (3)$$

This sparsity measure, while not as ideal other measures such as the Gini index,⁵ is a good candidate for CS because its minimization is well-defined as a convex optimization problem.⁶ The final equation, then, is:

$$\min \|\Psi\mathbf{x}\|_1 \text{ subject to } \mathbf{Ax} = \mathbf{y} \quad (4)$$

This is a standard basis pursuit problem which is solvable with minimal computational complexity.⁷ However, it does not always result in an optimal reconstruction when noise is involved. Another, more flexible minimization is the constrained L_1/L_2 problem, which is:

$$\min \|\Psi\mathbf{x}\|_1 \text{ subject to } \|\mathbf{Ax} - \mathbf{y}\|_2 \leq \delta \quad (5)$$

Here, the parameter δ can be adjusted to change the trade-off between transform sparsity and the noise floor of the reconstruction. Our experiments indicate that this problem takes longer to solve, but produces more accurate results. With computation power being so cheap (compared to MR imaging time), it is arguably the best choice for this application. It is also possible to include a TV (total variation) penalty if desired, but this will increase computation time. In the experiments performed in this paper, the YALL1 v1.4 ADM minimization algorithm⁸ was used with $\delta = 5 \times 10^{-4}$ and a stopping tolerance of 1×10^{-4} .

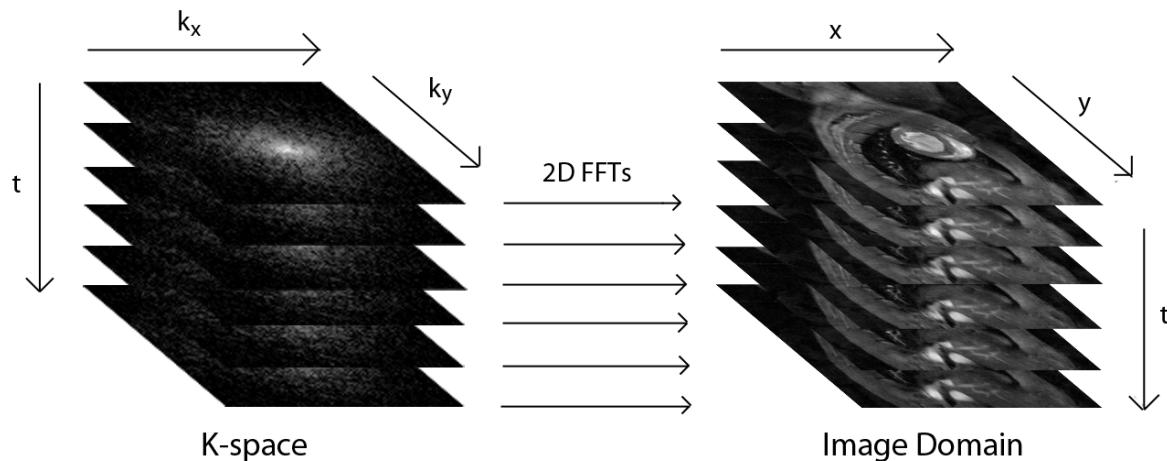


Figure 2. Example of image formation for non-CS mouse heart imaging. An entire set of k-space points is measured for each time period of interest, then a 2D FFT is applied to each frame to recover the image sequence.

2.2 2D Application of CS for MRI

Since only a small subset of the measurements are to be taken, proper selection of the samples to measure is of paramount importance. In general, the more “random” the sampling pattern, the more incoherent the undersampling noise will appear, and hence the better the resulting reconstruction will be.⁴ However, when the sampling is done in the frequency domain (as MRI is), there is an additional consideration. It is well known that most natural 2D images have most of the “information” concentrated near the origin in the frequency domain, and MRI images are no exception.⁹ Thus, a balanced approach to undersampling would use a probability distribution that is more likely to pick points closer to the origin than further away; the Gaussian distribution is often used to this effect.² Also, it is important to note that, due to the nature of MRI data acquisition patterns, only sampling in complete lines can provide practical savings in scan time. Though research on new MRI pulse sequences is in progress, the available pulse sequences at the time of this writing cannot sample single points individually. Thus, this paper will only concern itself with line-based k-space sampling.

The process of image formation for ordinary time-varying applications of MRI is shown in Figure 2. A fully-sampled set of k-space data points is acquired at each of several time instances, much like a video camera captures full “frames” of image data at evenly spaced points in time. However, since k-space cannot be sampled fast enough to keep up with a mouse’s rapid heartbeat (300-500 beats per minute), an assumption of periodicity in the cardiac cycle is made (more details in Section 3.1). After all k-space points are measured, a 2D FFT is applied to each frame individually to create the final image sequence.

In classical 2D compressive sensing, each k-space image is treated as a separate reconstruction. As shown in Figure 3, a single set of k-space points is randomly undersampled, then an L_1 minimization is applied using an appropriate 2D sparsifying transform (such as the 2D DCT). This approach is optimal for static images, but for time-varying sequences of images such as those obtained during mouse cardiac imaging, time-domain sparsity can be considered as well, resulting in better overall reconstruction quality for the entire image sequence.

2.3 3D Extension to CS for MRI

In contrast to the 2D approach, our 3D approach treats an entire sequence of k-space images as a single problem, using a 3D sparsifying transform that can take advantage of sparsity in time. By treating the entire “video” as a single optimization problem, the computation time will be increased in comparison to a 2D approach, but the reconstructions will be of far better quality. Figure 3 shows the contrast between 2D and 3D methods; the 3D case considers the entire image sequence as a single L_1 minimization. Treating the entire set of images as a single minimization problem will increase total reconstruction time due to the fact that computation time increases

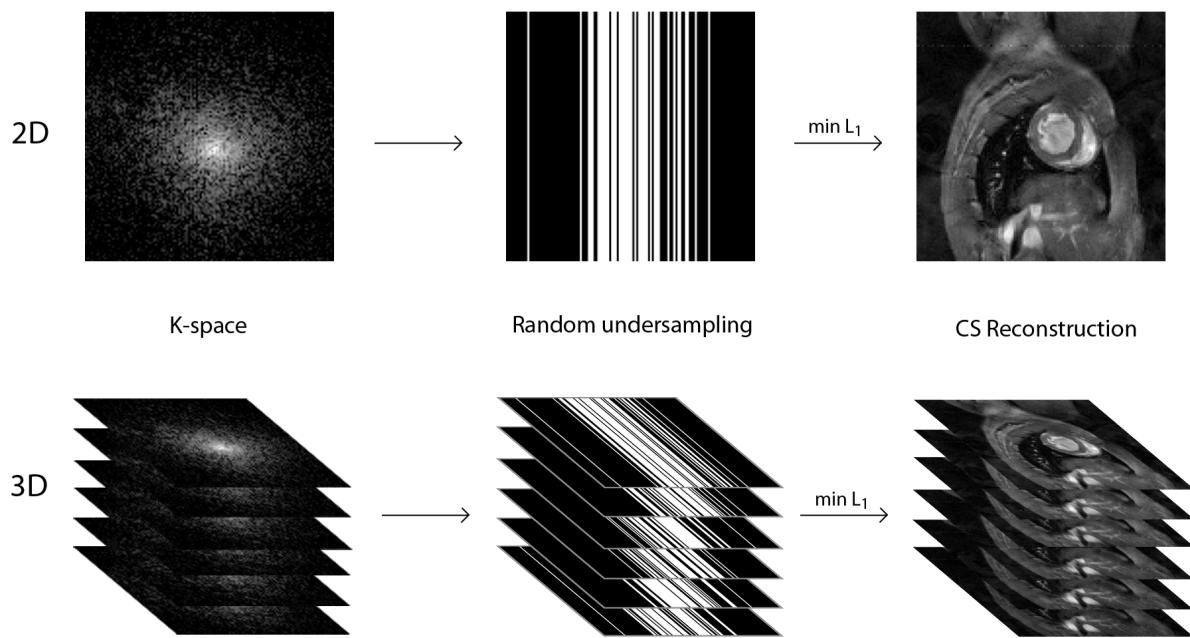


Figure 3. Contrasting 2D and 3D approaches to compressive sensing. The 2D method only performs minimizations on a single frame at a time; hence, it only exploits transform sparsity in space. The 3D method performs a minimization over the entire image sequence at once, so sparsity in time can be exploited as well. In both cases, the random undersampling patterns are shown using black-and-white masks where white pixels indicate k-space points that are sampled and black pixels indicate ones that are not.

more than linearly with problem size (especially when 3D sparsifying transforms are considered), but this is an acceptable trade-off for achieving better image quality.

In our experiment, sets of k-space data consisting of 12 frames of 128×128 pixels were used. Traditional 2D CS would dictate 12 separate reconstructions of size $n = 16834$. We can find the required number of samples m to acquire under line-based random sampling using:

$$m = \text{ceil}\left(\frac{rn}{128}\right) \times 128 \quad (6)$$

where r is the approximate undersampling ratio (i.e. $r = 0.3$ for 30% undersampling) and $\text{ceil}()$ is the operation of rounding up to the nearest integer. In 3D CS, both n and m are simply multiplied by 12, resulting in a problem size of $n = 196608$ and $59904 \leq m \leq 79872$ for sampling rates between 30% and 40%. Using MATLAB with YALL1 v1.4 on a computer with an Intel Core2 Quad Q9550 CPU and 8GB of RAM, 3D CS reconstructions for L1/L2 constrained problems of this size using the parameters listed in Section 2.1 take between 11 and 120 seconds each depending on the sparsifying function Ψ , the sampled MRI data, and the undersampling percentage. Further improvement on these times is possible via code optimization of the various 3D sparsifying functions, and also possibly with a massively-parallel GPU implementation.¹⁰

3. THE EXPERIMENT

The main goal of the proposed experiment is to compare the image quality obtained by 3-dimensional time-space CS reconstructions with that of the classical 2-dimensional approach in the mouse cardiac cycle. To do this, a set of seven image sequences was obtained from the University of Missouri VA Hospital, each of which contains 12 frames that form a single complete murine cardiac cycle (example in Figure 1).

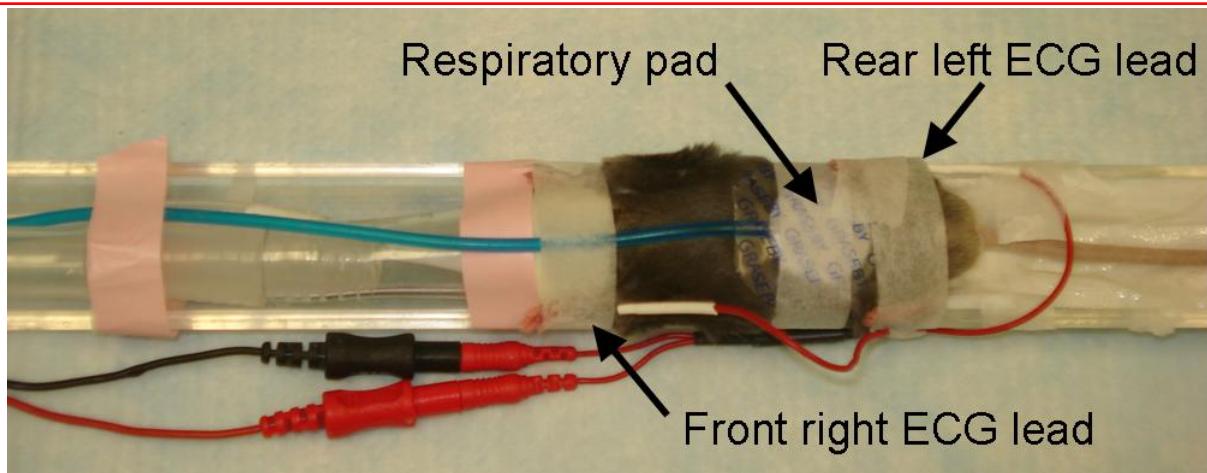


Figure 4. Picture of anesthetized mouse taped in cradle with sensors attached.

3.1 Equipment and Imaging Procedure

During the MRI scans, the animals were anesthetized with around 2% isoflurane in oxygen and their heart rates were maintained around 360 beats per minute. The animals were positioned supine in a cradle, where ECG electrodes were inserted into the right forepaw and left rear paw and a respiration sensor was taped on the chest as shown in Figure 4. The cradle was inserted into a 300 MHz quadrature driven birdcage RF coil with an inner diameter of 38 mm (shown in Figure 5), which was then placed into the center of a Varian Unity Inova 7 Tesla 210 mm horizontal bore MRI system equipped with a 400 mT/m gradient system (example shown in Figure 6). The ECG was used to trigger the MR scan at a particular point in the cardiac cycle (for example, immediately after the R-wave). The animals' body temperature was supported by warm air circulated through the MRI bore, and the ECG and respiratory monitoring/gating as well as the body temperature maintenance were done using a small animal physiological monitoring system.

Scout images at axial, coronal and sagittal planes were obtained at the very beginning of the procedure to localize the animal. Three images were acquired at each plane to confirm the correct positioning of the animal. After that, a multi-slice gradient echo pulse sequence was applied in order to obtain three sagittal images across the heart location based on the scout images. These sagittal images were used for determining the long-axis (coronal) plane of the left ventricle by positioning the slice across from the aortic valve to the apex. The short-axis plane was located by positioning the slice perpendicular to the long-axis plane. Finally, a typical steady state free precession (SSFP) sequence was applied to collect the first read-out line in k-space from each of the 12 frames in one cardiac cycle; this was then repeated for each subsequent line until the full set of measurements was acquired. The TR (repetition time) was calculated in real-time as the RR-delay (the time between R waves in the ECG) divided by 12, and the TE (echo time) was 1.57ms.

Since even a sedated mouse has a heart rate of over 300 beats per minute, there is obviously not enough time to sample anywhere close to a full set of k-space data before the heart moves to the next stage in the cycle. Because of this, the sampling scheme shown in Figure 7 was used to collect fully-sampled sets of k-space data for each of 12 stages in the cardiac cycle that are equally-spaced in time. The reason this works is because of the periodicity of the cycle. Each full heartbeat can be assumed to be identical (or mostly identical) to every other one; in this way, we can collect enough data for a full video in roughly 128 sequential heartbeats from the mouse. True undersampling performed in the pulse sequence can cut this down to 40 or less, resulting in fewer image artifacts due to the fact that not all heartbeats will look exactly identical.

3.2 2D and 3D Sparsifying Transforms

In CS, common 2D sparsifying transforms include the identity transform, the DFT, the DCT, and the DWT. The definitions of these are detailed in Table 1. For simplicity of analysis, the wavelet transform variants use periodic

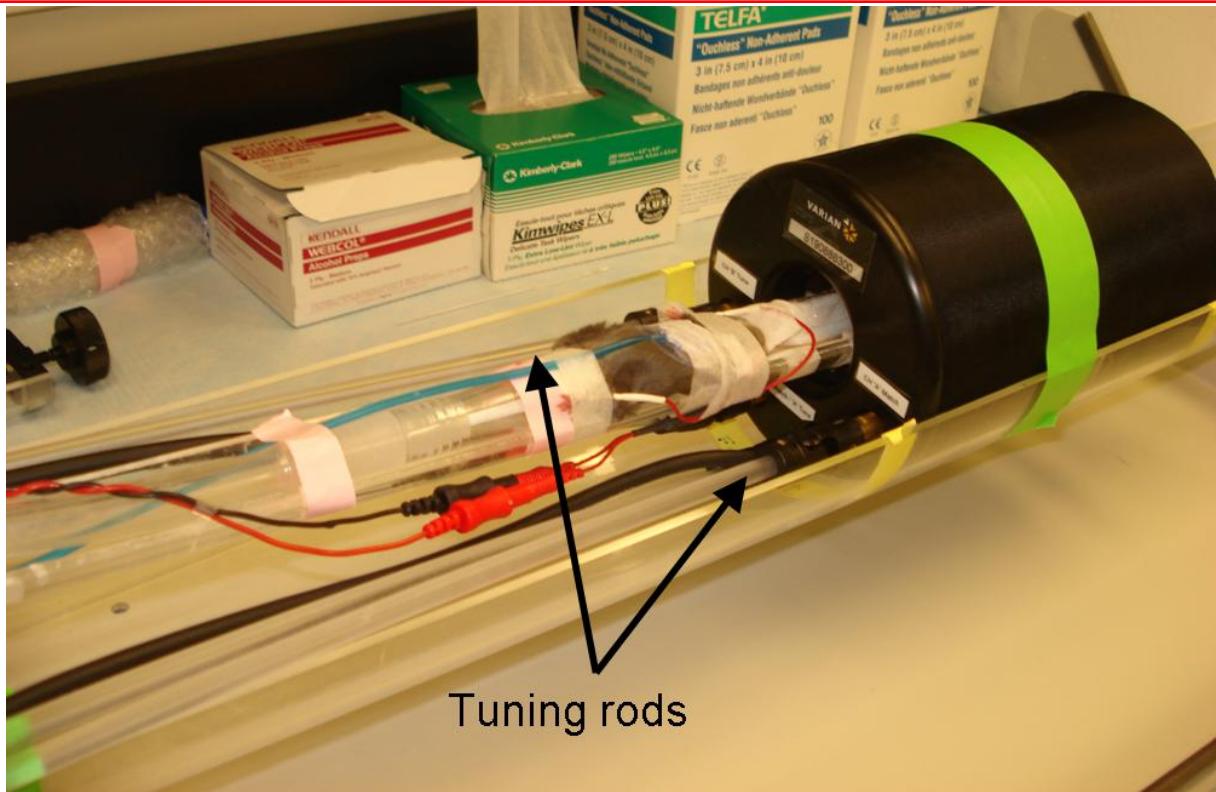


Figure 5. Picture of cradle being inserted into RF coil.

extension (i.e. the filters use circular convolution) to keep the size of the image unchanged upon transformation. Also worthy of mention is that, for the DCT, it is common to break the image up into a number of identically sized blocks before applying a 2D DCT to each block individually. For the DWT, it is common to repeatedly take the upper-left section and decompose it in order to increase sparsity. In this paper, an 8×8 pixel block DCT is used, and each DWT decomposition is performed to as many levels as MATLAB will allow.

In order to extend our analysis to three dimensions, it is necessary to apply a sparsifying transform that also extends to the time domain. The FFT was chosen as the most suitable transform for this purpose because of its computational simplicity in MATLAB as well as its propensity to condense the information from unchanging pixels into just one significant coefficient (the DC term). The full composite transform, then, consists of applying one of the 2D transforms from Table 1 to each frame of the video individually, then applying a 1D FFT along every pixel in the video as it changes with time. This process is shown in Figure 8 using the 8×8 pixel block DCT as an example of a 2D transform.

3.3 Analysis Method

The analysis method can be represented in pseudo-code as follows:

```
For undersampling percentages of x = 30%, 35%, and 40%, do these steps:  
  For each of the 7 videos of interest, do these steps:  
    Find visual reconstructions for the fully sampled k-space data  
    For each of 3 random trials, do these steps:  
      Use Gaussian line-based random undersampling to pick x% of the k-space points  
      For each of 10 sparsifying transforms, do these steps:  
        Perform constrained L1/L2 minimization with YALL1 to get the CS reconstruction  
        Calculate PSNR error metric of undersampled reconstruction  
  
The 10 sparsifying transforms applied in the x-y dimensions include:  
Identity transform  
2D DFT
```

Please verify that (1) all pages are present, (2) all figures are correct, (3) all fonts and special characters are correct, and (4) all text and figures fit within the red margin lines shown on this review document. Complete formatting information is available at <http://SPIE.org/manuscripts>

Return to the Manage Active Submissions page at <http://spie.org/app/submissions/tasks.aspx> and approve or disapprove this submission. Your manuscript will not be published without this approval. Please contact author_help@spie.org with any questions or concerns.



Figure 6. Example of full MRI system used in experiments.

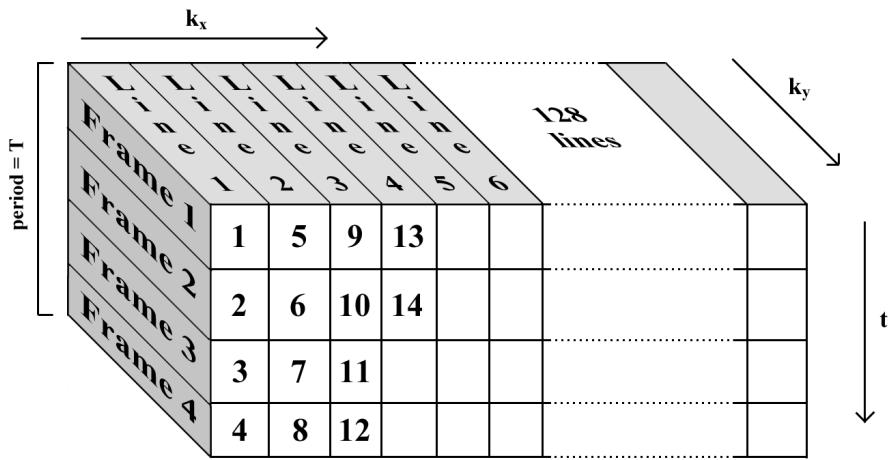


Figure 7. Example sampling method for acquiring a full video of a mouse heartbeat consisting of 4 frames. Starting with the leftmost line of k-space, the same line is measured 4 times in succession as the heart cycle progresses in time. After one full period T (i.e. one full heartbeat), the machine begins measuring the second k-space line as the heart goes through another full cycle. At the end, 128 lines will have been measured each of 4 times, taking a total of 128 full heartbeats to complete. Note that our experimental procedure divides the heart cycle into 12 frames instead of the 4 shown here.

8669 - 79 V. 1 (p.8 of 13) / Color: No / Format: Letter / Date: 12/1/2012 8:05:35 AM

SPIE USE: DB Check, Prod Check, Notes:

Table 1. Common 2D sparsifying transforms. In all cases, the transformed image $s \in \mathbb{C}^{m \times n}$ is a function Ψ of the input image $x \in \mathbb{C}^{m \times n}$, and j is the imaginary unit. The primary goal of these transforms is to increase the sparsity of s relative to x .

Transform	Description
Identity	The identity transform $s = x$
DFT2	2D Discrete Fourier Transform $s_{u,v} = \sum_{i=0}^{m-1} \sum_{k=0}^{n-1} x_{i,k} e^{-j2\pi(\frac{iu}{m} + \frac{kv}{n})} \quad \text{for } 0 \leq u \leq m-1, 0 \leq v \leq n-1$
DCT2	2D Discrete Cosine Transform $s_{u,v} = \sum_{i=0}^{m-1} \sum_{k=0}^{n-1} x_{i,k} \cos[\frac{\pi}{m}(i+0.5)u] \cos[\frac{\pi}{n}(k+0.5)v] \quad \text{for } 0 \leq u \leq m-1, 0 \leq v \leq n-1$
DWT2	2D Discrete Wavelet Transform Let g_L be a wavelet low-pass filter and g_H be a wavelet high-pass filter Filter x by g_L in both the x-direction and y-direction, then downsample by a factor of 2 in both dimensions to get section LL Use g_L in the x-direction and g_H in the y-direction, then downsample to get section LH Use g_H in the x-direction and g_L in the y-direction, then downsample to get section HL Use g_H in both the x-direction and y-direction, then downsample to get section HH

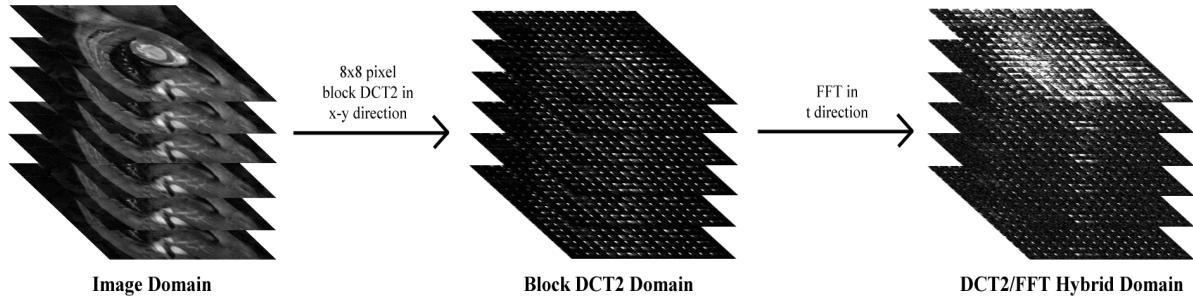


Figure 8. Example of a composite 3D sparsifying transform. First, a 2D DCT is applied on every individual 8×8 block of pixels in each frame of the video. Second, an FFT is applied along the time dimension to condense information into the first frame, which now contains all of the time-based DC terms.

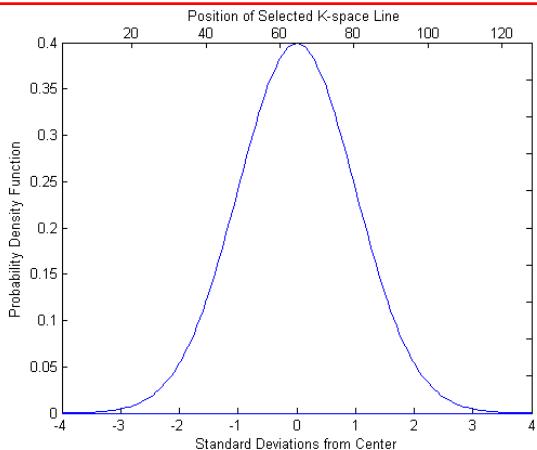


Figure 9. Gaussian PDF with $\sigma = 16$ pixels for random undersampling of a 128 line k-space grid. The mean of the distribution corresponds to line 65, which is always sampled even if the random sampling process does not select it. Lines will be sampled from this curve until the required percentage of coefficients is reached. This process is repeated independently for every frame of 2D k-space data to form the final 3D sampling pattern.

```
Full 2D DCT
8 x 8 pixel block 2D DCT
2D DWT with Haar wavelet
2D DWT with Daubechies wavelet of order 10
2D DWT with Symlet wavelet of order 2
2D DWT with Symlet wavelet of order 5
2D DWT with Coiflet wavelet of order 2
2D DWT with Coiflet wavelet of order 5.
```

The DFT is always the transform applied in the time dimension when 3D reconstruction is attempted.

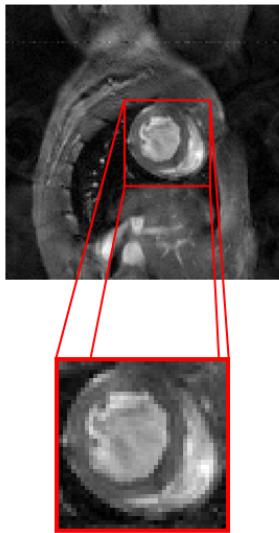
Because of variance in the random sampling patterns, it is necessary to repeat each reconstruction for a total of 3 random trials, each time picking a different sampling mask. The line-based sampling mask was drawn from a Gaussian probability distribution with a standard deviation of 16 pixels, with the center line in k-space as the mean. As an exception, this center line is always chosen for sampling by the program regardless of the probability distribution since it is so important. If a random result ends up choosing a line that either has already been chosen or is located past the image boundary, that result is discarded and a new line is chosen to replace it. Figure 9 demonstrates the relative probability of any given line being chosen at a point during random sampling.

One more consideration to note is that randomly chosen sampling masks may still not perform well, because it is very much possible to “randomly” generate a mask that happens to have a measure of coherence by chance. In order to ensure the most incoherent sampling mask possible, we can generate several masks and choose the one with the least interference in the point spread function (PSF) as proposed by Lustig et al.⁴ In our experiment, 100 different masks were generated each time a new sampling pattern was called for, and the one with the lowest peak PSF out of those was chosen for use.

3.4 Performance Metrics and Image Examples

In order to judge the success of our proposed CS method, and also to discern the best 2D sparsifying transform, it is necessary to use a performance metric. In theory, the transform that results in the sparsest representation of the signal will create the highest quality image upon reconstruction, so a sparsity measure such as the Gini Index could be used. However, since all of the fully-sampled images are at our disposal, a more practical approach is to simply take the error of the undersampled reconstructions when they are compared to the fully sampled ones. The peak signal-to-noise ratio (PSNR) is a fairly standard measure of image quality that takes into account the average error, or “noise”, of every pixel in the image. It is defined as:

Fully-Sampled Reconstruction



Close-up of Left Ventricle

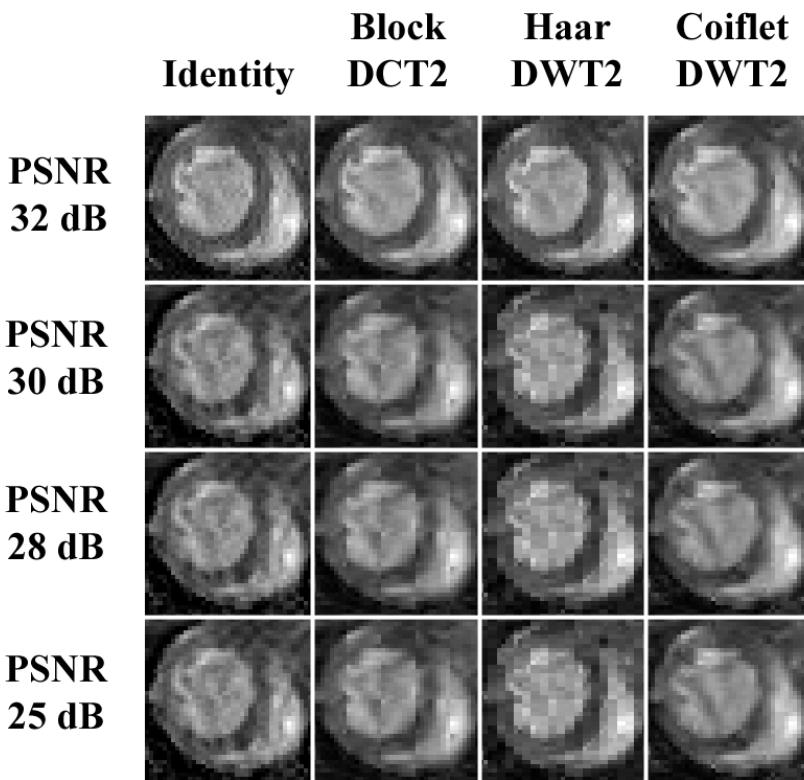


Figure 10. Examples of PSNR ratings for various 2D CS reconstructions. Sparsifying transforms include the identity transform, the 2D DCT performed upon each 8×8 pixel block, the 2D DWT with the Haar wavelet, and the 2D DWT with the third-order coiflet wavelet. The full k-space data set from which samples are drawn is the same for all reconstructions. Images with PSNR below 25 dB are mostly too distorted to be useful, so they are not shown.

$$PSNR = 10\log_{10} \left(\frac{n \max(\mathbf{x})^2}{\sum_{i=0}^{n-1} (x_i - \hat{x}_i)^2} \right) \quad \text{for all } \mathbf{x}, \hat{\mathbf{x}} \in \mathbb{R}^n, \quad (7)$$

where $\max(\mathbf{x})$ is the maximum possible intensity of any single point in the image, \mathbf{x} is the original image, and $\hat{\mathbf{x}}$ is the “estimate” of that image; in our case, it is the reconstruction. This metric can be easily applied in our case, and since it is so widely used as a measure of image quality, it also provides a way to compare our results to those of other MRI CS studies. For this reason, and others discussed by Huynh-Thu and Ghanbari,¹¹ the PSNR was chosen as the main performance metric for this study.

Figure 10 provides a demonstration of the various PSNR levels for CS reconstructions with different transforms. These images were generated by performing 2D reconstructions and adjusting the undersampling rate for each transform until the desired PSNR level was reached. The left side of the figure contains the fully sampled image for reference, and the right side shows the artifacts of CS undersampling on the left ventricle (of which estimating the volume is the main objective for this type of imaging) as the PSNR drops. The identity transform tends to cause moderate-to-severe aliasing artifacts as the sampling rate goes down, whereas the 8×8 pixel block DCT presents the classic “block artifact” seen in heavy JPEG compression. On the other hand, when certain wavelet transforms are used, they tend to exhibit noise that is less disruptive in nature, and as a result tend to require fewer samples to achieve the same subjective image quality as other transforms.

Table 2. PSNR results for each sparsifying transform over all images for 30%, 35%, and 40% undersampling. The best-performing transform for each case is highlighted. The 3D method significantly increases image quality compared to the 2D method even at lower sampling rates.

2D Sparsifying Transform	PSNR (dB) for 2D CS			PSNR (dB) for 3D CS		
	30%	35%	40%	30%	35%	40%
Identity	24.78	25.97	27.08	31.61	32.45	33.28
DFT2	24.26	25.70	27.32	29.81	31.26	32.58
DCT2, 8x8 pixel blocks	26.12	27.41	28.67	31.78	32.93	33.68
DCT2, full image	23.63	25.15	26.71	30.76	32.00	33.11
DWT2, Haar wavelet	25.68	26.81	28.02	31.27	32.22	33.20
DWT2, Daubechies order 10	26.09	27.48	28.76	32.24	33.14	34.07
DWT2, Symlet order 2	26.20	27.39	28.63	31.64	32.46	33.50
DWT2, Symlet order 5	26.33	27.64	28.92	32.25	33.25	34.24
DWT2, Coiflet order 2	26.34	27.68	29.00	32.29	33.30	34.19
DWT2, Coiflet order 5	26.15	27.64	29.04	32.36	33.26	34.14

3.5 PSNR Results for 2D and 3D Compressive Sensing on Mouse Hearts

To obtain a good idea of the significance of 3D reconstruction, it is necessary to compare it to the 2D case. Upon running the analysis detailed in Section 3.3 and obtaining the PSNR metrics, we can deduce the general performance for a particular undersampling rate by averaging the obtained PSNR over all images for each transform in turn. This analysis was performed for the following two cases:

1. All 84 frames (12 frames in each of 7 videos) were considered as independent images, and classical 2D CS reconstruction was performed on all of them using each of 10 different 2D sparsifying transforms.
2. Each of the 7 videos was considered as a single 3D CS reconstruction. The same 2D transforms as the first step were applied to each image frame, but an FFT was performed in the time direction as well before performing an L_1 minimization on the whole video at once.

The final results for 2D and 3D CS at undersampling rates of 30%, 35%, and 40% are shown in Table 2. Under 2D reconstruction, the images exhibit a best-case PSNR of between 24 and 29 dB depending on the chosen transform and the number of k-space lines sampled. Referring back to Figure 10, the area of the image corresponding to the left ventricle is somewhat distorted in this range, though it is still possible to roughly estimate the volume. In order to obtain a more acceptable PSNR, higher undersampling percentages would be required under 2D compressive sensing. However, the 3D reconstructions offer an average PSNR of over 32 dB when the best transform is used, even with only 30% undersampling. The PSNR for 30% undersampling (which is a standard target undersampling rate in MRI CS) experiences an average increase of around 6 dB, which is a major improvement.

In terms of PSNR, the best performing 2D transforms were the higher-order wavelets, but wavelet transforms also offer another advantage over the DCT block transforms that is not reflected in the PSNR. As Figure 10 shows, the block artifacts in the DCT distort the shape and size of the left ventricular cavity more than the time-frequency artifacts in the wavelet transforms, and so the volume measurement will be impacted negatively. Clearly, the high-order symlet and coiflet wavelets should be used whenever it is computationally feasible in order to obtain the best image quality.

4. CONCLUSIONS

The analyses performed in this paper demonstrate that exploiting time sparsity in murine cardiac MRI with compressive sensing offers a significant improvement to image quality. Furthermore, since the reduced measurement time will reduce motion artifacts from the heartbeat, the results in practice should be even more impressive than those predicted by the experiment performed here, given that we simply discarded points from sets of fully (and thus slowly) sampled data.

Warranting future exploration is the use of sparsifying transforms other than the FFT in the time domain for better sparsification. Expanding the dictionary of 2D transforms (adding more high order wavelets in particular) would also allow the algorithm to explore more possibilities for 2D transform sparsity, further improving performance. Finally, as proposed by Zonoobi et al.,⁵ using a stochastic algorithm to minimize the Gini index rather than the L_1 norm could provide even greater reconstruction accuracy, as the Gini index is usually a much better measure of sparsity than the L_1 norm.

ACKNOWLEDGMENTS

Special thanks are due to Ming Yang and Dr. Lixin Ma at the Veteran Affairs Hospital and the University of Missouri - Columbia, who provided all of the raw MRI data for this study. Also due thanks are Hamed Kajbaf and Zengli Yang, who contributed knowledge of CS methods as well as pieces of MATLAB code, and Ken Hanson, for providing the L^AT_EX template for SPIE manuscripts. This work was financially supported by the University of Missouri Research Board Fund.

REFERENCES

- [1] Kajbaf, H., T. Case, J., Zheng, Y. R., Kharkovsky, S., and Zoughi, R., "Quantitative and Qualitative Comparison of SAR Images from Incomplete Measurements Using Compressed Sensing and Nonuniform FFT," *RADAR Conference IEEE 2011*, 592–596 (2011).
- [2] Wech, T., Lemke, A., and Medway, D., "Accelerating Cine-MR Imaging in Mouse Hearts Using Compressed Sensing," *Journal of Magnetic Resonance Imaging* **34**(5), 1072–1079 (2011).
- [3] Cooley, J. W. and Tukey, J. W., "An Algorithm for the Machine Calculation of Complex Fourier Series," *Mathematics of Computation* **19**(90), 297–301 (1965).
- [4] Lustig, M., Donoho, D. L., and Pauly, J. M., "Sparse MRI: The Application of Compressed Sensing for Rapid MR Imaging," *Magnetic Resonance in Medicine* **58**, 1182–1195 (2007).
- [5] Zonoobi, D., Kassim, A. A., and Venkatesh, Y. V., "Gini Index as Sparsity Measure for Signal Reconstruction from Compressive Samples," *IEEE Journal of Selected Topics in Signal Processing* **5**(5), 927–932 (2011).
- [6] Donoho, D. L., "Compressed Sensing," *IEEE Transactions on Information Theory* **52**(4), 1289–1306 (2006).
- [7] Chen, S. S., Donoho, D. L., and Saunders, M. A., "Atomic Decomposition by Basis Pursuit," *SIAM Review* **43**(1), 129–159 (2001).
- [8] Zhang, Y., Yang, J., and Yin, W., *Users Guide for YALL1: Your ALgorithms for L1 Optimization*. http://www.caam.rice.edu/~optimization/L1/YALL1/User_Guide/YALL1v1.0_User_Guide.pdf.
- [9] Fuderer, M., "The Information Content of MR Images," *IEEE Transactions on Medical Imaging* **7**(4), 368–380 (1988).
- [10] Andrecut, M., "Fast GPU Implementation of Sparse Signal Recovery from Random Projections," *Engineering Letters* **17**(3) (2009).
- [11] Huynh-Thu, Q. and Ghanbari, M., "Scope of validity of PSNR in image/video quality assessment," *Electronics Letters* **44**(13) (2008).