NIH Public Access
Author Manuscript
Magn Reson Med. Author manuscript; available in PMC 2016 February 01.

Published in final edited form as:
Magn Reson Med. 2015 February ; 73(2): 828–842. doi:10.1002/mrm.25176.

Robust 4D Flow Denoising Using Divergence-Free Wavelet Transform

Frank Ong[1], **Martin Uecker**[1], **Umar Tariq**[2], **Albert Hsiao**[2], **Marcus T Alley**[2], **Shreyas S. Vasanawala**[2], and **Michael Lustig**[1,*]

[1]Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, California, USA

[2]Department of Radiology, Stanford University, Stanford, California, USA

Abstract

Purpose—To investigate four-dimensional flow denoising using the divergence-free wavelet (DFW) transform and compare its performance with existing techniques.

Theory and Methods—DFW is a vector-wavelet that provides a sparse representation of flow in a generally divergence-free field and can be used to enforce "soft" divergence-free conditions when discretization and partial voluming result in numerical nondivergence-free components. Efficient denoising is achieved by appropriate shrinkage of divergence-free wavelet and nondivergence-free coefficients. *SureShrink* and cycle spinning are investigated to further improve denoising performance.

Results—DFW denoising was compared with existing methods on simulated and phantom data and was shown to yield better noise reduction overall while being robust to segmentation errors. The processing was applied to in vivo data and was demonstrated to improve visualization while preserving quantifications of flow data.

Conclusion—DFW denoising of four-dimensional flow data was shown to reduce noise levels in flow data both quantitatively and visually.

Keywords

four-dimensional flow; wavelet denoising; divergence-free

Introduction

Time-resolved three-dimensional phase-contrast MRI [four-dimensional (4D) flow] is a promising imaging technique that can provide both cardiac anatomy and function in a single acquisition (1). Potential clinical applications of 4D flow were shown in many areas including evaluation of valve-related disease, analysis of dynamic blood flow in the aorta

*Correspondence to: Michael Lustig, 506 Cory Hall University of California, Berkeley, CA 94720. mlustig@eecs.berkeley.edu.

This article was presented in part at the 21st annual meeting of ISMRM, Salt Lake City, Utah, 2013, the annual meeting of ISBI, San Francisco, California, 2013, and the annual meeting of SCMR, San Francisco, California, 2013.

and quantification of cardiac flow using derived parameters such as pressure difference maps and wall sheer stress (2). Although it is now possible to get acquisition times below 5 min, clinical acceptance is limited by its vulnerability to phase errors and issues associated with the interpretation of the vast amount of generated data. In this work, we focus on reducing noise-like phase errors in 4D flow data.

Noise-like phase error in flow data can arise from body noise or hardware imperfection and may be further amplified by high velocity encodes (VENCs) to avoid velocity aliasing when the dynamic range is high. Low velocity-to-noise ratio in 4D flow data often reduces confidence in visualization and lowers the quantification accuracy. Moreover, a common research approach to accelerate 4D flow scan time is to use undersampling methods, such as k-t GRAPPA (3), k-t BLAST, k-t SENSE (4), k-t PCA (5), k-t SPARSE (6), L1-SPIRiT (7), or other parallel imaging and compressed sensing techniques. Although these techniques reduce scan time, this reduction is usually associated with lower signal-to-noise ratio and hence velocity-to-noise ratio. In the case of nonuniform subsampling, artifacts may also appear as noise in velocity data, which can persist in reconstructed data. Hence, an effective noise reduction processing is highly desired for 4D flow data.

To reduce noise or artifacts, several authors have proposed incorporating physical conditions of blood flow in flow data processing (8–11). As blood flow is incompressible and hence divergence-free, noise-like errors can be reduced by suppressing divergent components in flow data. In particular, Song et al. (8) proposed denoising MR flow field by projecting the data onto divergence-free vector fields using the finite difference method (FDM). The projection operation was reduced to an inverse 7-point Laplacian problem, which was solved by a fast Poisson solver using the Fast Fourier Transform. Another recent work by Busch et al. (9) constructed divergence-free flow field by projecting the noisy flow field onto divergence-free radial basis functions (RBF) using iterative least squares. Normalized convolution with an uncertainty map was used to incorporate boundary conditions in the flow field. Both FDM and RBF were shown to be effective as a denoising process for flow imaging (12).

However, one issue with existing projection methods is that they enforce the flow field to be strictly divergence-free, which require accurate segmentation to prevent unwanted boundary effects near edges. In practice, discrete approximation and partial voluming of flow cannot be fully captured by a strict divergence-free representation. This situation often occurs in places near edges of flow, static tissue, or turbulent flow, where discrete representation of flow consists of discontinuities. Strict divergence-free enforcement across these discontinuities may result in significant error propagation throughout the flow field. Although segmentation of flow data can help preventing these effects, accurate segmentations are often hard to obtain in low signal-to-noise ratio data. Errors from segmentation can also contribute to significant divergent components. Hence, a "softer" divergence-free enforcement of flow data is needed to enforce appropriate constraints on different flow regions.

In this work, we present a robust and effective noise reduction processing using the divergence-free wavelet (DFW) transform (13). DFWs were first introduced by Lemarié-

Rieusset (14) to the computational fluid dynamics (CFD) community in the 1980s. Since then, DFWs were investigated in several CFD applications for simulations and flow data compression (15–18). In particular, DFWs were shown to provide a sparse representation for simulated flow data in (17) and were used to separate random flow from actual flow field in (18). These two properties encourage us to apply DFW denoising in the context of 4D flow MRI.

The purpose of this work is to demonstrate the effectiveness and robustness of DFW denoising on flow data through soft-thresholding (19). To further improve denoising performance, we investigate *SureShrink* (20) for selecting appropriate thresholds and cycle-spinning (21) for removing blocking artifacts from wavelet transform.

Theory

Divergence-Free Wavelets

DFWs are vector-wavelets that can separate flow data into divergence-free wavelet and non-divergence-free wavelet (DFW) coefficients. Despite its name, DFW coefficients span both components and hence the entire space of vector fields. By separating flow field into divergence-free and nondivergence-free components, DFW transform offers better decorrelation of flow data than standard separate wavelet transforms and thereby provides better energy compactness or sparsity of flow data (17). Efficient denoising can be achieved by appropriate shrinkage of divergence-free and nondivergence-free coefficients. As the processing is essentially wavelet denoising, DFW denoising inherits advantages of wavelet denoising, including efficient multiscale decompositions, edge preserving transforms, and sparse representation of signals, while amounting to only linear computational complexity.

The construction of DFWs relies on the following proposition that relates two different wavelet functions by differentiation (17):

Proposition— Let $\phi_1(x)$ and $\psi_1(x)$ be a one-dimensional differentiable scaling function and wavelet function, respectively. Then, we can build another one-dimensional scaling function $\phi_0(x)$ and wavelet function $\psi_0(x)$ such that

$$\phi_1'(x) = \phi_0(x) - \phi_0(x-1) \quad \psi_1'(x) = 4\psi_0(x) \qquad [1]$$

One set of wavelets that satisfies the above proposition is the linear spline wavelet and the quadratic spline wavelet shown in Figure 1a. Using the above proposition, DFWs can then be explicitly constructed by combining tensor products of the one dimensional wavelet functions. Specifically, consider the case of two-dimensional for simplicity and let the following functions be a subset of the scaling and wavelet basis functions when applying standard wavelet transform with filters ϕ_0, ψ_0, ϕ_1, and ψ_1 on v_x and v_y separately:

$$\Phi_{vx}(x,y)=\begin{pmatrix}\phi_1(x)\phi_0(y)\\0\end{pmatrix},\quad \Psi_{vx}(x,y)=\begin{pmatrix}\psi_1(x)\psi_0(y)\\0\end{pmatrix}$$
$$\Phi_{vy}(x,y)=\begin{pmatrix}0\\\phi_0(x)\phi_1(y)\end{pmatrix},\quad \Psi_{vy}(x,y)=\begin{pmatrix}0\\\psi_0(x)\psi_1(y)\end{pmatrix} \qquad [2]$$

Then using the above functions, we can construct two-dimensional divergence-free scaling and wavelet functions of the following form:

$$\Phi_{\text{divfree}}(x,y)=\begin{pmatrix}\phi_1(x)[\phi_0(y)-\phi_0(y-1)]\\-[\phi_0(x)-\phi_0(x-1)]\phi_1(y)\end{pmatrix}=[\Phi_{vx}(x,y)-\Phi_{vx}(x,y-1)]-[\Phi_{vy}(x,y)-\Phi_{vy}(x-1,y)]$$
$$\Psi_{\text{divfree}}(x,y)=\begin{pmatrix}\psi_1(x)\psi_0(y)\\-\psi_0(x)\psi_1(y)\end{pmatrix}=\Psi_{vx}(x,y)-\Psi_{vy}(x,y) \qquad [3]$$

which can be verified to have zero divergence:

$$\nabla\cdot\Phi_{\text{divfree}}=[\phi_0(x)-\phi_0(x-1)][\phi_0(y)-\phi_0(y-1)]-[\phi_0(x)-\phi_0(x-1)][\phi_0(y)-\phi_0(y-1)]=0$$
$$\nabla\cdot\Psi_{\text{divfree}}=4\psi_0(x)\psi_0(y)-4\psi_0(x)\psi_0(y)=0 \qquad [4]$$

As each DFW basis function can be expressed in terms of separate wavelet basis functions Φ_{vx}, Φ_{vy}, Ψ_{vx} and Ψ_{vy}, computation of DFW coefficients is reduced to a simple linear combination of wavelet coefficients generated by separate wavelet transforms on each velocity component. Nonisotropic resolutions along directions can be compensated by scaling the wavelet coefficients for each direction by its relative resolution (see Appendix). Thus, the procedure for DFW denoising is only different from standard wavelet denoising in that we have to linearly combine wavelet coefficients before and after soft-thresholding. Similar procedure can be extended to the case of three-dimensional and used to generate a complete set of 21 DFW functions. The complete set of linear combination equations is provided in Appendix. Detailed derivation can be found in (17) under the name “isotropic DFW transform.” Examples of two-dimensional slices of DFW basis functions are shown in Figure 1b,c. The entire denoising flow diagram is shown in Figure 1d and achieves linear computational complexity.

With the DFWs, we obtain a sparse representation of flow data. Hence, to effectively denoise flow data, we propose soft-thresholding (19) the wavelet coefficients to promote sparsity in the divergence-free components and enforce “soft” divergence-free constraints in the nondivergence-free components. Instead of eliminating nondivergence-free coefficients, soft-thresholding nondivergence-free coefficients allows the flexibility to adjust the cutoff so that important components, such as those arising near edges, can be captured. This operation is essentially an approximation to an l_1-penalized least squares, which was shown to be more robust to errors near boundaries (22). As wavelet coefficients are separated into divergence-free and nondivergence-free coefficients, two separate thresholds can be chosen for divergence-free and nondivergence-free components, thereby allowing better denoising performance. Moreover, standard wavelet denoising techniques, such as Stein's Unbiased Risk Estimator (SURE)-based threshold selector (20) and cycle spinning (21), can also be used to optimize the performance.

Threshold Selection and Cycle Spinning

To select an appropriate threshold for a given noise level, we consider *SureShrink* (20) as an optimal scheme for minimizing mean square error in the wavelet domain. *SureShrink* was proposed by Donoho and Johnstone as a hybrid scheme that chooses between a SURE-based threshold and a minimax optimal threshold. As minimax threshold tends to oversmooth when applied on image data (23), we use only the SURE-based threshold in *SureShrink* for denoising flow data. Since SURE-based methods assumes white Gaussian noise, flow data is first segmented to remove flow regions with low image magnitude. Noise in the resulting flow data can then be approximated as Gaussian noise with standard deviation VENC/SNR (24). In practice, this mask can be conservatively chosen as DFW thresholding is robust to segmentation errors, which we demonstrate in Results section.

Formally, for each wavelet subband j, let I_j be the index set of subband coefficients corresponding to the segmented data, N_j be the length of the index set I_j, $x_{i,j}$ be the ith subband coefficients in I_j, and σ be the noise standard deviation, *SureShrink* chooses the subband dependent threshold t_j^* as follows (25).

$$t_j^* = \min_t \sigma^2 - \frac{1}{N_j}\left(2\sigma^2 \# \{i: |x_{i,j}| \le t\} - \sum_{i=1}^{N_j} \min\left(|x_{i,j}|, t\right)^2\right) \quad [5]$$

Thresholds for *SureShrink* can be computed with complexity $N \log N$. Detailed derivation can be found in (20) and (25). To robustly estimate σ, median absolute deviation (MAD) can be used, which is given by the formula: $\sigma = 1.4826$ median $(|x_i - \text{median}(x)|)$.

As with standard biorthogonal wavelet denoising, DFW denoising is not translation invariant and suffers from blocking artifacts. To reduce these artifacts, we consider cycle spinning (21) to improve denoising performance.

Methods

The proposed methods were implemented in the programming language C and CUDA (26) with MATLAB (The MathWorks, Natick, MA) MEX wrappers. FDM and RBF were implemented in MATLAB for comparison. For the construction of DFW, linear spline wavelet (Cohen-Daubechies-Feauveau 2.2) was used for ϕ_0 and ψ_0, and quadratic spline wavelet (Cohen-Daubechies-Feauveau 3.1) were used for ϕ_1 and ψ_1, all of which with symmetric boundary extensions. Unless specified, two levels of wavelet decompositions were used for CFD simulation and three levels of wavelet decompositions were used for other experiments. Instead of applying the full cycle spinning, partial cycle spinning was used to reduce complexity. This procedure shifts the input data randomly, applies DFW denoising and averages the results for a few iterations. In all experiments, eight random shifts were used for partial cycle spinning. In the soft-thresholding operation, wavelet coefficients in the coarsest level were left untouched as they were not sparse. *SureShrink* and MAD were used for threshold selection unless specified. For *SureShrink*, MAD was applied on the highest frequency subband of nondivergence-free component to estimate the noise standard deviation because the estimation is more accurate when applied on a sparser subband. In the current implementation of DFW transform, wavelet filters along with the linear combination step for each subband are not normalized. To compensate for the scaling, the normalization factors for each subband were precomputed by applying DFW transform on white Gaussian noise and averaging over each subband. For FDM, first-order finite difference and periodic boundary conditions were used. Velocity data was masked by an image magnitude mask for both DFW and FDM. For RBF, the support of the basis functions was set to be 19 × 19 × 19 for CFD simulation and 9 × 9 × 9 for phantom experiments as they produced low errors. A binary certainty function with a uniform nonzero weight for flow regions was used for normalized convolution for RBF. Iterative least squares in RBF was implemented with LSQR (27) in MATLAB with maximum number of iterations set to 30. All methods and simulations were implemented on a workstation that has dual-socket with six-core Intel Westmere CPUs at 2.67 GHz with 64 GB of system DRAM and an Nvidia GTX580 GPU with 3 GB of high-speed Graphics DRAM.

In the spirit of reproducible research, we provide a software package to reproduce some of the results described in this article. The software can be downloaded from: http://www.eecs.berkeley.edu/~mlustig/Software.html.

CFD Simulation: Comparison with Existing Methods

To compare denoising performances with existing methods, a three-dimensional steady-state flow through a stenosis was simulated using OpenFOAM (28), an open source CFD software package. The tube had an opening of 2-cm diameter with a narrowing of 0.5-cm diameter. Kinematic viscosity was set to be 3.33×10^{-6} m^2/s. A constant flow of 15 cm/s was applied from the top of the tube and gridded flow data with matrix size 480 × 80 × 80 was collected after 100 time-steps (time-step = 1 μs). This particular geometry and time point were chosen because of the resulting detailed flow field. MR data with five-point balanced phase-contrast method (29) was simulated by setting the phase to be linear combinations of flow data following (29). VENC was set to be 229 cm/s. The complex data magnitude in image domain was set to one wherever the velocity field was nonzero. The reference phase was set to be zero.

Different levels of complex Gaussian noise were added to the complex data. RBF, FDM, and DFW without cycle spinning and DFW with partial and full cycle spinning were applied on the noisy data for comparison. Resulting errors before and after processing were averaged over 30 iterations for each noise standard deviation.

Flow Phantom Experiment: Effects of Segmentation Errors

To test the denoising performances on MR flow phantom data and the effect of using an incorrect segmentation of the flow field, a fully sampled 4D flow data was acquired from a pulsatile flow phantom on a 3T GE Scanner with a 32 channel Torso array. The 4D flow acquisition was performed using a spoiled gradient-echo-based sequence with tetrahedral flow encoding. The spatial resolution was 0.86 × 0.86 × 1.30 mm^3. The flip angle was 15° and T_R/T_E was 3.52/1.37 ms. VENC was set to be 150 cm/s. The flow data was corrected for Maxwell phase effects (30), gradient nonlinearity distortions (31), and eddy-current (32). The entire flow data had a matrix size of 134 × 192 × 64 and featured a tube with stenosis on the left and a static flow phantom in the middle. The static region was used to correct for effects from eddy currents. Only the tube with stenosis is shown in most of the following figures. The entire flow phantom is shown in Figure 7.

Complex Gaussian noise was retrospectively added to the acquired data to generate a noisy flow data with peak velocity-to-noise ratio (PVNR) of 33.5 dB. Image magnitude segmentation was obtained by setting an appropriate threshold on the magnitude image. An incorrect image magnitude segmentation was obtained by lowering the threshold. RBF, FDM, and DFW with and without partial cycle spinning were applied on the noisy data with the correct segmentation to test for noise reduction performance and with the incorrect segmentation to test for robustness to boundary errors.

Flow Phantom Experiment: Reduction of Incoherent Artifacts

Reduction of incoherent artifacts from undersampling (33) using DFW was also investigated. k-space data of the flow phantom was first coil-compressed (34) into eight virtual channels. The phase encodes were retrospectively subsampled by 5.4 using a Poisson-disk sampling mask [Fig. 7; (35)]. The same sampling mask was applied on each VENC. ESPIRiT (36) was used to extract sensitivity maps from the calibration region and SENSE (37) was used to reconstruct the flow data. DFW with partial cycle spinning, *SureShrink*, and MAD were applied on the reconstructed flow data. As coherent artifacts in the undersampled data can overwhelm the SURE risk minimizer, DFW denoising with manually tuned thresholds was also applied and compared. For ease of usage, only two global thresholds for divergence-free and nondivergence-free components were manually specified. Flow data reconstructed by ESPIRiT with l_1 regularization was generated for comparison.

In Vivo Data: Visual Improvement and Effect on Quantification

To investigate the effect on flow quantifications, DFW was applied on eight in vivo datasets. In vivo 4D cardiac flow data were acquired in eight pediatric patients with 20 cardiac phases, 122–144 slices and an average spatial resolution of 0.99 × 0.99 × 1.12 mm^3. Four patient data had regurgitant fractions (RF) less than 5% and the other four had RFs greater than 30%. The flow data were acquired on a 1.5T GE Signa Scanner with an eight channel cardiac array. The 4D flow acquisition was performed using a spoiled gradient echo-based sequence with tetrahedral flow encoding and variable density Poisson disk under-sampling. The flip angle was 15° and the average T_R/T_E was 4.94/1.91 ms. VENCs for the studies ranged from 150 to 300 cm/s. The acquisitions were undersampled by about 4 and was reconstructed using L1-SPIRiT, a compressed sensing and parallel imaging reconstruction algorithm (7). Volumetric eddy-current correction was performed on velocity data following (32). Segmentations for flow calculations were done manually on the aorta and pulmonary trunk. Net flow rate(volume/time) and RF(%) were calculated for each segmentation. DFW was applied on reconstructed flow data from each cardiac phase.

Besides *SureShrink*, results from manually chosen thresholds for DFW were also analyzed with two global thresholds on the divergence-free and non-DFW coefficients. DFW was implemented in JCUDA and incorporated into a custom built Java-based flow visualization software package (38). The computation time of DFW denoising with no cycle spinning on a single cardiac phase was less than 1 s. This enabled real-time interactive control of the denoising parameters to improve visual quality and minimize the flow inconsistency between the aorta and pulmonary trunk. Once the thresholds were set, DFW denoisings with partial cycle spinning were applied on all cardiac phases.

The quantifications measured were net flow rate, RFs, and the deviations between systematic and pulmonary flow. Because RFs and net flow rate were measured over segmentations and over time, they were relatively robust to noise and artifacts even before processing and were shown to agree with gold standards. Hence, change in RF and flow rates after denoising were expected to be small to preserve flow quantifications. Deviations between systematic and pulmonary flow were also expected to be small.

In addition, streamline quantification on a particular study was generated using Ensight (CEI, Apex, NC). For qualitative assessment, streamlines were released from a plane placed at the ascending aorta for L1-SPIRiT reconstructed data and the subsequent DFW denoised data. For quantitative streamline metric, particles were emitted from a plane placed at the upper part of the descending aorta and an analysis plane was placed at the lower part of the descending aorta. As most of the flow from the emitter plane should pass through the analysis plane, the percentage of streamlines reaching the analysis plane was used as a metric to quantify the improvement after denoising.

Error Analysis

To quantify errors in experiments, different error metrics were used. PVNR was used to quantify the initial noise level in simulations. Denoising performance was quantified with regard to normalized-root-mean-squared-error (NRMSE) in velocity and in speed, as defined in the standard convention. Average direction error was also considered, which was bounded by 1. All error calculations shown in Result section were done on the entire (correctly) segmented regions.

Formally, let N be the number of segmented voxels and $v_{i,\text{ref}}$ and $v_{i,\text{denoised}}$ be the reference and denoised velocity vectors, respectively, at the ith segmented voxel, we define the error metrics as the following:

$$\text{PVNR} = 20\log_{10}\frac{1}{\text{Velocity NRMSE}}\text{dB}$$

$$\text{Velocity NRMSE} = \frac{1}{\max_i\left(|V_{i,\text{ref}}|\right)}\sqrt{\frac{1}{N}\sum_{i=0}^{N}\left|V_{i,\text{ref}} - V_{i,\text{denoised}}\right|^2}$$

$$\text{Speed NRMSE} = \frac{1}{\max_i\left(|V_{i,\text{ref}}|\right)}\sqrt{\frac{1}{N}\sum_{i=0}^{N}\left(|V_{i,\text{ref}}| - |V_{i,\text{denoised}}|\right)^2} \quad [6]$$

$$\text{Direction Error} = \frac{1}{N}\sum_{i=0}^{N}\left(1 - \frac{|V_{i,\text{ref}} \cdot V_{i,\text{denoised}}|}{|V_{i,\text{ref}}|\,|V_{i,\text{denoised}}|}\right)$$

Results

CFD Simulation: Comparison with Existing Methods

Figure 2 presents the simulation results on a noisy CFD data with 22-dB PVNR. Visually, both DFW and RBF show significant noise reduction in velocity magnitude, whereas FDM shows little improvement over the noisy flow field. Comparing to DFW without cycle spinning, DFW with partial cycle spinning reduces blocking artifacts and improves denoising performance. DFW with two wavelet level decomposition also suppresses more noise in general than DFW with one wavelet level decomposition, but may lose some details as pointed by the white arrows. RBF also loses some details as pointed and has difficulty representing discontinuities in the velocity field near the stenosis. The red arrow points to artifacts produced by RBF. These artifacts persist even when a smaller kernel for RBF is used.

Vector visualization of the same experiment is shown in Figure 3. As in velocity magnitude, both DFW and RBF show significant noise reduction in vector visualization. Visually, their vector representations look very similar to the original flow field. Although FDM shows less improvement than other methods, some noise suppression can still be observed especially near the vortices on both sides.

Figure 4 shows the velocity profiles of different slices before and after DFW denoising with partial cycle spinning. Velocity profiles after DFW processing closely resemble the original velocity field. Discontinuities in velocity data, such as those in slice 2 and 3, can still be captured with DFW even when DFW was applied on the entire flow field. DFW suppresses most of the noise while preserving the shape of each individual velocity direction. For example, in slice 3, DFW preserves small variation in v_y even when the variation in v_x is large.

Quantitative error plots over a range of PVNRs for CFD simulations are also shown in Figure 4. DFW with full cycle spinning outperforms other methods in all three error criteria. DFW with partial cycle spinning comes close as second. RBF is third in most PVNRs but loses to DFW without cycle spinning in both velocity and speed NRMSE for high PVNRs. FDM is consistently behind other methods, but has lower errors than noisy data. As expected, at PVNR equals 22 dB, the quantitative errors for each method match their visual quality in Figures 2 and 3.

Flow Phantom Experiment: Effects of Segmentation Errors

Figure 5 shows the results of denoising a noisy flow phantom with PVNR of 33.5 dB. Similar to the CFD simulations, both DFW and RBF show significant noise reduction in velocity magnitude, whereas FDM shows only small improvement over noisy data. DFW is shown to reduce noise in the static flow regions while preserving details in velocity magnitude. In general, RBF provides a smoother representation of velocity magnitude and may present a better performance visually compared to DFW. However, the error maps show that some details in the original flow field are blurred after RBF processing. In addition, while artifacts in RBF are not as prominent as in the CFD simulation, some ringing artifacts still show up near the stenosis as pointed by the white arrow.

Quantitatively, velocity NRMSE, speed NRMSE, and direction error for the noisy flow phantom are 2.12%, 1.23%, and 0.00807, respectively. For DFW without cycle spinning, they are 1.69%, 1.02%, and 0.00598, respectively. For DFW with cycle spinning, they are 1.50%, 0.907%, and 0.00481, respectively. For FDM, they are 3.37%, 1.97%, and 0.0207, respectively. For RBF, they are 1.60%, 1.05%, and 0.00727, respectively. DFW with cycle spinning achieves the lowest errors in all three error criteria, with DFW without cycle spinning and RBF competing for second. FDM has higher errors than the noisy flow field has, which may due to errors in the acquired flow data.

To test for the robustness of denoising methods, an incorrect segmentation mask was chosen by lowering the threshold on image magnitude. Because of partial voluming, regions outside the actual flow region can be included and result in significant discontinuities near edges. Figure 6 shows the results of denoising the same noisy flow phantom with the incorrectly chosen segmentation. Visually, DFW is largely unaffected by the change in segmentation and produces similar results as before. RBF shows significant distortions near edges and errors propagating throughout the field. FDM shows less distortions, but they can still be observed in the zoomed in portion.

Quantitatively, velocity NRMSE, speed NRMSE, and direction error for the noisy flow phantom within the correctly masked data are 2.12%, 1.23%, and 0.00807, respectively. For DFW without cycle spinning, they are 1.72%, 1.03%, and 0.00691, respectively. For DFW with cycle spinning, they are 1.39%, 0.849%, and 0.00470, respectively. For FDM, they are 3.80%, 2.21%, and 0.0262, respectively. For RBF, they are 3.17%, 1.89%, and 0.00214, respectively. Compared to the results with the correct segmentation, errors for all methods, except for DFW with cycle spinning, go up. Errors for RBF and FDM increase significantly, which confirms with the visual quality. Both the visual quality and error quantities show that DFW is robust to segmentation errors. However, with the coarsely chosen mask, errors near edges for DFW increase slightly compared to the correct mask, indicating that a better mask still leads to better performance in general.

Flow Phantom Experiment: Reduction of Incoherent Artifacts

To test for the artifact reduction performance for DFW, k-space data was retrospectively undersampled using a 5.4-fold Poisson-disk sampling mask on Figure 7. The figure presents the result of denoising reconstructed velocity field using DFW with *SureShrink* and manually chosen threshold. Visually, DFW with *SureShrink* reduces some artifacts, but is overly conservative as some of the incoherent artifacts can still be observed. With more aggressively chosen thresholds, DFW suppresses most of the artifacts and improves the performance significantly. Quantitatively, velocity NRMSE, speed NRMSE, and direction error for the reconstructed flow phantom are 3.36%, 1.94%, and 0.0249, respectively. For DFW with *SureShrink*, they are 2.99%, 1.75%, and 0.0215, respectively. For DFW with manually chosen thresholds, they are 2.35%, 1.59%, and 0.0154, respectively. The errors for DFW with *SureShrink* decrease slightly compared to the noisy data while DFW with manually chosen thresholds further suppresses the errors. In comparison, ESPIRiT with spatial wavelet l_1 regularization is also shown in Figure 7 and recovers almost exactly the original flow field, showing that denoising by itself cannot replace the entire compressed sensing reconstruction.

In Vivo Data: Visual Improvement and Effects on Quantification

Table 1 shows the quantitative results before and after applying DFW with *SureShrink* and manually chosen thresholds. For DFW with *SureShrink*, the mean percentage change in flow rate and mean change in RF after denoising were small for both groups with RF < 5% and RF > 30%. The minor change in quantifications suggests that *SureShrink* does not distort flow quantifications. After applying DFW with *SureShrink*, mean of ($Q_p - Q_s$) stays close to zero, indicating that the bias is small. Standard deviation of ($Q_p - Q_s$) is observed to decrease after DFW denoising, suggesting that DFW with *SureShrink* improves flow consistency across patient data.

Thresholds that were manually chosen based on visual quality were compared to *SureShrink*. In general, the manually chosen thresholds were greater than the *SureShrink* thresholds. One of the patient data with RF > 30% near the aorta is shown in Figure 8. Visually, *SureShrink* thresholding reduces the noise level slightly when compared to the original data. For manually chosen thresholds, three levels of thresholds and their corresponding positions on the L-curve are shown in the same figure to demonstrate the tradeoffs of choosing the thresholds. White arrows point to details that are lost when a high threshold was applied during denoising.

Vector visualization and streamline visualization of DFW denoising are shown in Figure 9. In the vector visualization panel, results from thresholding only the non-DFW coefficients and thresholding both divergence-free wavelet and non-DFW coefficients are compared. With only non-DFW coefficients thresholded, the flow vectors are more aligned and the global noise level decreases slightly. With both divergence-free wavelet and non-DFW coefficients thresholded, the global noise level is significantly reduced and flow patterns become cleaner. In streamline visualization, the DFW denoised flow shows more coherent streamlines when compared to the original streamlines. Red arrows point to streamlines that flow outside of the anatomy for the L1-SPIRiT reconstructed data but remains inside of the anatomy for DFW. Quantitatively, for the streamlines released from the descending aorta, 21.9% of the particles emitted from the emitter plane reach the analysis plane for the L1-SPIRiT data, whereas 33.6% of them reach the analysis plane for the DFW denoised data, showing that DFW can improve streamline lengths.

Computation Time

All simulations were run on the same workstation with configuration described in Methods section. In the MATLAB implementation on the CFD data (matrix size 480 × 80 × 80), DFW (1 cycle) took 20–40 s, RBF took 10–15 min, and FDM took about 1 s. In general, as FDM and DFW are both noniterative, they are significantly faster than RBF. In C implementation with no parallelization, DFW (1 cycle) took about 10 s and in CUDA implementation, it took less than 1 s and was dominated by memory transfer from CPU to GPU.

Discussion

Performance of DFW Denoising and Existing Methods

In our simulations and experiments, we have shown that soft divergence-free enforcement through DFW transform leads to a better denoising performance. Although enforcing divergence-free conditions on flow can suppress noise in general, eliminating nondivergence-free components in the flow field can contribute to significant error propagating throughout the field. In particular, we have shown that in two experiments, sharp transition near stenosis (Fig. 2) or segmentation errors (Fig. 6) can result in prominent nondivergence-free components in flow field. Both strict divergence-free enforcements using FDM or RBF generated artifacts. Since DFW denoising enforces divergence-free constraints through the soft-thresholding operation, significant nondivergence-free components were preserved and hence did not distort the flow field in those experiments.

RBF, in general, performs exceptionally well when the flow field is smooth and the uncertainty map is correctly chosen. However, compared to FDM, RBF is more sensitive to nondivergence-free components and creates more prominent artifacts. This is due to the larger kernel size of RBF. Although such artifacts can be reduced when a smaller kernel size is chosen, they can never be eliminated as they also appear in FDM, which has kernel size 3 × 3 × 3 for the Laplacian operator. Because a larger kernel size results in smaller errors in our simulations, the current kernel sizes are used instead. Conversely, because FDM has the smallest kernel size, FDM consistently performed the worst out of the three denoising schemes implemented. Although using a higher order finite difference and incorporating a smoothness penalty can improve its denoising performance (11), the inherent problem of imposing strict divergence-free conditions is not solved.

Although soft-thresholding allows DFW denoising to impose soft divergence-free constraints, it can also result in blocking artifacts and lead to a worse reconstruction as shown in Figure 2. Hence, with cycle spinning, DFW has consistently shown to improve denoising performance. As doing full cycle spinning often requires 64 or more wavelet transforms, we opt for partial cycle spinning with eight random shifts and have shown in the CFD simulation that its performance is close to the performance of full cycle spinning.

Together with partial cycle spinning, DFW denoising outperforms other denoising schemes overall both quantitatively and visually.

In our experiments with retrospectively added Gaussian noise, *SureShrink* in general has picked appropriate thresholds that produce data with good visual quality. Its denoising performance was verified by the CFD simulation and flow phantom experiment. However, with undersampled data, we have shown that *SureShrink* can be overly conservative and hence manually tuned thresholds may be required. This underperformance of *SureShrink* on undersampled data may be caused by the coherent artifacts in the flow field, which can overwhelm the SURE risk minimizer. However, we emphasize that when underperforming, *SureShrink* is often overly conservative and hence can be acted as a baseline for fine tuning the thresholds. In vivo studies have also indicated that *SureShrink* does not distort flow quantifications as the net flow rate and RF did not change drastically. Although the changes were not large, the decreased deviation between Q_p and Q_s suggests that DFW with *SureShrink* improves the flow consistency across the field.

Applicability of Divergence-Free Denoising

We also highlight the flexibility of DFW denoising, offering both high-level automated denoising and low level manual adjustments. At a high level, MAD and *SureShrink* simplify the denoising process and effectively reduce the input parameters to be the number of wavelet levels and the image mask. Since DFW denoising is robust toward segmentation errors, both parameters can be easily obtained and provide good denoising performance in general.

Conversely, as *SureShrink* only minimizes the mean squared error, a better threshold can be fine-tuned for specific needs. For example, a smaller threshold may be chosen to preserve the details in the CFD experiment (Fig. 2) or a higher threshold may be chosen for denoising undersampled data with non-Gaussian artifacts (Fig. 7). Since the fast computation of DFW transform allows users to pick a threshold with instant feedback, it enables fine tuning of the parameters. The robustness of DFW transform and the ability to fine tune parameters suggest that DFW denoising can be safely applied to clinical data.

Further Improvement

Although *SureShrink* produces thresholds that have small mean squared errors in simulations, it can be suboptimal as compactly supported DFWs are biorthogonal (15). Biorthogonality of DFWs implies that the transform does not preserve noise statistics, so minimizing errors in the wavelet domain does not minimize errors in the image domain directly. However, spline wavelets are nearly orthogonal and the constructed DFWs are also close to orthogonal as shown in (18). When choosing the optimal threshold for *SureShrink*, we make the approximation that the DFWs are approximately orthogonal for computational efficiency. For true optimal selection of thresholds, the minimization for *SureShrink* should be solved in the image domain instead, which involves multiple wavelet transforms (23). Moreover, more advanced threshold selectors, such as SURE-LET (23), can be used instead of *SureShrink* and may offer a better performance. Jointly estimating thresholds from different shifts or undecimated wavelet transform can also improve the denoising performance (23) but can increase the computation substantially.

Conclusions

In this article, 4D flow denoising using DFW is shown to be effective, while being robust to discontinuities in flow. We have shown that combining DFW transform with *SureShrink* and partial cycle spinning results in better denoising performance in general. Our in vivo experiments also suggest that DFW denoising can be safe for quantification purposes, especially when fast computation of DFW denoising allows the user fine tuning the level of denoising interactively. When compared to existing methods, DFW enables “softer” enforcement of divergence-free constraints, thereby providing a more robust denoising performance overall.

Acknowledgments

NIH: Grant number: P41RR09784, R01EB009690; Grant sponsor: American Heart Association; Grant number: 12BGIA9660006; Grant sponsor: The Sloan Research Fellowship.

Appendix: Construction of DFW Coefficients

The following describes the construction of all DFW coefficients following (17). For each wavelet decomposition level, we define:

- $(\varphi_1(x), \Psi_1(x))$ and $(\varphi_0(x), \Psi_0(x))$ be the scaling and wavelet function pairs described in Theory section.
- $s = (i, j, k)$ be the indices for each subband in a three dimensional wavelet transform, with i, j and k equal to 1 when the subband is projected on ψ along x, y, and z directions, respectively, and 0 when the subband is projected on ϕ along x, y, and z directions, respectively.
- d_{vx}^s be the wavelet coefficients of v_x with ϕ_1/ψ_1 applied along the x direction and ϕ_0/ψ_0 applied along the y and z directions on subband s. For example, $d_{vx}^{(0,1,0)}$ corresponds to the wavelet coefficients of v_x with ϕ_1 applied along x direction, ψ_0 applied along y direction, and ϕ_0 applied along z direction.
- d_{vy}^s be the wavelet coefficients of v_y with ϕ_1/ψ_1 applied along the y direction and ϕ_0/ψ_0 applied along x and z directions on subband s.
- d_{vz}^s be the wavelet coefficients of v_z with ϕ_1/ψ_1 applied along z direction and ϕ_0/ψ_0 applied along x and y directions on subband s.
- d_{df1}^s and d_{df2}^s be the divergence-free component of DFW coefficients and d_{n}^s be the nondivergence-free component of DFW coefficients.
- Then, the construction of DFW coefficients are given by:

$$
\begin{aligned}
d_{\mathrm{df1}}^{(1,0,0)}(x,y,z) &= d_{vy}^{(1,0,0)}(x,y,z)\\
d_{\mathrm{df2}}^{(1,0,0)}(x,y,z) &= d_{vz}^{(1,0,0)}(x,y,z)\\
d_{\mathrm{n}}^{(1,0,0)}(x,y,z) &= d_{vx}^{(1,0,0)}(x,y,z) + \frac{1}{4}\left(d_{vy}^{(1,0,0)}(x,y,z) - d_{vy}^{(1,0,0)}(x,y-1,z)\right) + \frac{1}{4}\left(d_{vz}^{(1,0,0)}(x,y,z) - d_{vz}^{(1,0,0)}(x,y,z-1)\right)\\
d_{\mathrm{df1}}^{(0,1,0)}(x,y,z) &= d_{vx}^{(0,1,0)}(x,y,z)\\
d_{\mathrm{df2}}^{(0,1,0)}(x,y,z) &= d_{vz}^{(0,1,0)}(x,y,z)\\
d_{\mathrm{n}}^{(0,1,0)}(x,y,z) &= d_{vy}^{(0,1,0)}(x,y,z) + \frac{1}{4}\left(d_{vx}^{(0,1,0)}(x,y,z) - d_{vx}^{(0,1,0)}(x-1,y,z)\right) + \frac{1}{4}\left(d_{vz}^{(0,1,0)}(x,y,z) - d_{vz}^{(0,1,0)}(x,y,z-1)\right)\\
d_{\mathrm{df1}}^{(0,0,1)}(x,y,z) &= d_{vx}^{(0,0,1)}(x,y,z)\\
d_{\mathrm{df2}}^{(0,0,1)}(x,y,z) &= d_{vy}^{(0,0,1)}(x,y,z)\\
d_{\mathrm{n}}^{(0,0,1)}(x,y,z) &= d_{vz}^{(0,0,1)}(x,y,z) + \frac{1}{4}\left(d_{vx}^{(0,0,1)}(x,y,z) - d_{vx}^{(0,0,1)}(x-1,y,z)\right) + \frac{1}{4}\left(d_{vy}^{(0,0,1)}(x,y,z) - d_{vy}^{(0,1,0)}(x,y-1,z)\right)\\
d_{\mathrm{df1}}^{(1,1,0)}(x,y,z) &= \frac{1}{2}\left(d_{vx}^{(1,1,0)}(x,y,z) - d_{vy}^{(1,1,0)}(x,y,z)\right)\\
d_{\mathrm{df2}}^{(1,1,0)}(x,y,z) &= d_{vz}^{(1,1,0)}(x,y,z)\\
d_{\mathrm{n}}^{(1,1,0)}(x,y,z) &= \frac{1}{2}\left(d_{vx}^{(1,1,0)}(x,y,z) + d_{vy}^{(1,1,0)}(x,y,z)\right) + \frac{1}{8}\left(d_{vz}^{(1,1,0)}(x,y,z) - d_{vz}^{(1,1,0)}(x,y,z-1)\right)\\
d_{\mathrm{df1}}^{(1,0,1)}(x,y,z) &= \frac{1}{2}\left(d_{vx}^{(1,0,1)}(x,y,z) - d_{vz}^{(1,0,1)}(x,y,z)\right)\\
d_{\mathrm{df2}}^{(1,0,1)}(x,y,z) &= d_{vy}^{(1,0,1)}(x,y,z)\\
d_{\mathrm{n}}^{(1,0,1)}(x,y,z) &= \frac{1}{2}\left(d_{vx}^{(1,0,1)}(x,y,z) + d_{vz}^{(1,0,1)}(x,y,z)\right) + \frac{1}{8}\left(d_{vy}^{(1,0,1)}(x,y,z) - d_{vz}^{(1,0,1)}(x,y-1,z)\right)\\
d_{\mathrm{df1}}^{(0,1,1)}(x,y,z) &= \frac{1}{2}\left(d_{vy}^{(0,1,1)}(x,y,z) - d_{vz}^{(0,1,1)}(x,y,z)\right)\\
d_{\mathrm{df2}}^{(0,1,1)}(x,y,z) &= d_{vy}^{(0,1,1)}(x,y,z)\\
d_{\mathrm{n}}^{(0,1,1)}(x,y,z) &= \frac{1}{2}\left(d_{vy}^{(0,1,1)}(x,y,z) + d_{vz}^{(0,1,1)}(x,y,z)\right) + \frac{1}{8}\left(d_{vx}^{(0,1,1)}(x,y,z) - d_{vx}^{(0,1,1)}(x-1,y,z)\right)\\
d_{\mathrm{df1}}^{(1,1,1)}(x,y,z) &= \frac{1}{3}\left(-2d_{vx}^{(1,1,1)}(x,y,z) + d_{vy}^{(1,1,1)}(x,y,z) + d_{vz}^{(1,1,1)}(x,y,z)\right)\\
d_{\mathrm{df2}}^{(1,1,1)}(x,y,z) &= \frac{1}{3}\left(-d_{vx}^{(1,1,1)}(x,y,z) + 2d_{vy}^{(1,1,1)}(x,y,z) - d_{vz}^{(1,1,1)}(x,y,z)\right)\\
d_{\mathrm{n}}^{(1,1,1)}(x,y,z) &= \frac{1}{3}\left(d_{vx}^{(1,1,1)}(x,y,z) + d_{vy}^{(1,1,1)}(x,y,z) + d_{vz}^{(1,1,1)}(x,y,z)\right)
\end{aligned}
\qquad [A1]
$$

And, the inverse transform of the DFW coefficients are given by

$$d_{vx}^{(1,0,0)}(x,y,z) = d_{\mathrm{n}}^{(1,0,0)}(x,y,z) - \frac{1}{4}(d_{\mathrm{df1}}^{(1,0,0)}(x,y,z) - d_{\mathrm{df1}}^{(1,0,0)}(x,y-1,z)) - \frac{1}{4}(d_{\mathrm{df2}}^{(1,0,0)}(x,y,z) - d_{\mathrm{df2}}^{(1,0,0)}(x,y,z-1))$$

$$d_{vy}^{(1,0,0)}(x,y,z) = d_{\mathrm{df1}}^{(1,0,0)}(x,y,z)$$

$$d_{vz}^{(1,0,0)}(x,y,z) = d_{\mathrm{df2}}^{(1,0,0)}(x,y,z)$$

$$d_{vx}^{(0,1,0)}(x,y,z) = d_{\mathrm{df1}}^{(0,1,0)}(x,y,z)$$

$$d_{vy}^{(0,1,0)}(x,y,z) = d_{\mathrm{n}}^{(0,1,0)}(x,y,z) - \frac{1}{4}(d_{\mathrm{df1}}^{(0,1,0)}(x,y,z) - d_{\mathrm{df1}}^{(0,1,0)}(x-1,y,z)) - \frac{1}{4}(d_{\mathrm{df2}}^{(0,1,0)}(x,y,z) - d_{\mathrm{df2}}^{(0,1,0)}(x,y,z-1))$$

$$d_{vz}^{(0,1,0)}(x,y,z) = d_{vz}^{(0,1,0)}(x,y,z)$$

$$d_{vx}^{(0,0,1)}(x,y,z) = d_{\mathrm{df1}}^{(0,0,1)}(x,y,z)$$

$$d_{vy}^{(0,0,1)}(x,y,z) = d_{\mathrm{df2}}^{(0,0,1)}(x,y,z)$$

$$d_{vz}^{(0,0,1)}(x,y,z) = d_{\mathrm{n}}^{(1,0,0)}(x,y,z) - \frac{1}{4}(d_{\mathrm{df1}}^{(0,0,1)}(x,y,z) - d_{\mathrm{df1}}^{(0,0,1)}(x-1,y,z)) - \frac{1}{4}(d_{\mathrm{df2}}^{(0,0,1)}(x,y,z) - d_{\mathrm{df2}}^{(0,1,0)}(x,y-1,z))$$

$$d_{vx}^{(1,1,0)}(x,y,z) = d_{\mathrm{df1}}^{(1,1,0)}(x,y,z) + d_{\mathrm{n}}^{(1,1,0)}(x,y,z) - \frac{1}{8}(d_{\mathrm{df2}}^{(1,1,0)}(x,y,z) - d_{\mathrm{df2}}^{(1,1,0)}(x,y,z-1))$$

$$d_{vy}^{(1,1,0)}(x,y,z) = -d_{\mathrm{df1}}^{(1,1,0)}(x,y,z) + d_{\mathrm{n}}^{(1,1,0)}(x,y,z) - \frac{1}{8}(d_{\mathrm{df2}}^{(1,1,0)}(x,y,z) - d_{\mathrm{df2}}^{(1,1,0)}(x,y,z-1)) \quad \text{[A2]}$$

$$d_{vz}^{(1,1,0)}(x,y,z) = d_{\mathrm{df2}}^{(1,1,0)}(x,y,z)$$

$$d_{vx}^{(1,0,1)}(x,y,z) = d_{\mathrm{df1}}^{(1,0,1)}(x,y,z) + d_{\mathrm{n}}^{(1,0,1)}(x,y,z) - \frac{1}{8}(d_{\mathrm{df2}}^{(1,0,1)}(x,y,z) - d_{\mathrm{df2}}^{(1,0,1)}(x,y-1,z))$$

$$d_{vy}^{(1,0,1)}(x,y,z) = d_{\mathrm{df2}}^{(1,0,1)}(x,y,z)$$

$$d_{vz}^{(1,0,1)}(x,y,z) = -d_{\mathrm{df1}}^{(1,0,1)}(x,y,z) + d_{\mathrm{n}}^{(1,0,1)}(x,y,z) - \frac{1}{8}(d_{\mathrm{df2}}^{(1,0,1)}(x,y,z) - d_{\mathrm{df2}}^{(1,0,1)}(x,y-1,z))$$

$$d_{vx}^{(0,1,1)}(x,y,z) = d_{\mathrm{df2}}^{(0,1,1)}(x,y,z)$$

$$d_{vy}^{(0,1,1)}(x,y,z) = d_{\mathrm{df1}}^{(0,1,1)}(x,y,z) + d_{\mathrm{n}}^{(0,1,1)}(x,y,z) - \frac{1}{8}(d_{\mathrm{df2}}^{(0,1,1)}(x,y,z) - d_{\mathrm{df2}}^{(0,1,1)}(x-1,y,z))$$

$$d_{vz}^{(0,1,1)}(x,y,z) = -d_{\mathrm{df1}}^{(0,1,1)}(x,y,z) + d_{\mathrm{n}}^{(0,1,1)}(x,y,z) - \frac{1}{8}(d_{\mathrm{df2}}^{(0,1,1)}(x,y,z) - d_{\mathrm{df2}}^{(0,1,1)}(x-1,y,z))$$

$$d_{vx}^{(1,1,1)}(x,y,z) = -d_{\mathrm{df1}}^{(1,1,1)}(x,y,z) + d_{\mathrm{n}}^{(1,1,1)}(x,y,z)$$

$$d_{vy}^{(1,1,1)}(x,y,z) = d_{\mathrm{df2}}^{(1,1,1)}(x,y,z) + d_{\mathrm{n}}^{(1,1,1)}(x,y,z)$$

$$d_{vz}^{(1,1,1)}(x,y,z) = -(d_{\mathrm{df1}}^{(1,1,1)}(x,y,z) + d_{\mathrm{df2}}^{(1,1,1)}(x,y,z) + d_{\mathrm{n}}^{(1,1,1)}(x,y,z))$$

To compensate for nonisotropic resolutions, the wavelet coefficients d_{vx}^{s}, d_{vy}^{s}, and d_{vz}^{s} are scaled by $1/r_x$, $1/r_y$, and $1/r_z$, respectively, before the linear combination stage, where r_x, r_y, and r_z are the relative resolutions along each direction. The inverse is simply scaling the wavelet coefficients by r_x, r_y, and r_z.

References

1. Markl M, Chan FP, Alley MT, et al. Time-resolved three-dimensional phase-contrast MRI. J Magn Reson Imaging. 2003; 17:499–506. [PubMed: 12655592]

2. Markl M, Frydrychowicz A, Kozerke S, Hope M, Wieben O. 4D flow MRI. J Magn Reson Imaging. 2012; 36:1015–1036. [PubMed: 23090914]
3. Huang F, Akao J, Vijayakumar S, Duensing GR, Limkeman M. k-t GRAPPA: a k-space implementation for dynamic MRI with high reduction factor. Magn Reson Med. 2005; 54:1172–1184. [PubMed: 16193468]
4. Tsao J, Boesiger P, Pruessmann KP. k-t BLAST and k-t SENSE: dynamic MRI with high frame rate exploiting spatiotemporal correlations. Magn Reson Med. 2003; 50:1031–1042. [PubMed: 14587014]
5. Pedersen H, Kozerke S, Ringgaard S, Nehrke K, Kim WY. k-t PCA: temporally constrained k-t BLAST reconstruction using principal component analysis. Magn Reson Med. 2009; 62:706–716. [PubMed: 19585603]
6. Lustig, M.; Santos, JM.; Donoho, DL.; Pauly, JM. kt SPARSE: high frame rate dynamic MRI exploiting spatio-temporal sparsity. Proceedings of the 13th Annual Meeting of ISMRM; Seattle, USA. 2006. p. 2420
7. Lustig M, Pauly JM. SPIRiT: iterative self-consistent parallel imaging reconstruction from arbitrary k-space. Magn Reson Med. 2010; 64:457–471. [PubMed: 20665790]
8. Song SM, Napel S, Glover GH, Pelc NJ. Noise reduction in three-dimensional phase-contrast MR velocity measurementsl. J Magn Reson Imaging. 1993; 3:587–596. [PubMed: 8347951]
9. Busch J, Giese D, Wissmann L, Kozerke S. Reconstruction of divergence-free velocity fields from cine 3D phase-contrast flow measurements. Magn Reson Med. 2013; 69:200–210. [PubMed: 22411739]
10. Skrinjar, O.; Bistoquet, A.; Oshinski, J.; Sundareswaran, K.; Frakes, D.; Yoganathan, A. A divergence-free vector field model for imaging applications. IEEE International Symposium on Biomedical Imaging: From Nano to Macro, 2009 (ISBI'09); Boston, Massachusetts, USA. 2009. p. 891-894.
11. Fatouraee N, Amini AA. Regularization of flow streamlines in multi-slice phase-contrast MR imaging. IEEE Trans Med Imaging. 2003; 22:699–709. [PubMed: 12872945]
12. Leocher, M.; Kecskemeti, S.; Johnson, K.; Turski, P.; Wieben, O. Evaluation of divergence-free correction algorithms in high resolution 4-D flow images of cranial vasculature. Proceedings of the 20th Annual Meeting of ISMRM; Melbourne, Australia. 2012. p. 1245
13. Ong, F.; Uecker, M.; Tariq, U.; Hsiao, A.; Alley, MT.; Vasanawala, SS.; Lustig, M. Improved visualization and quantification of 4D flow MRI data using divergence-freewavelet denoising. IEEE 10th International Symposium on Biomedical Imaging (ISBI); 2013; USA. 2013. p. 1186-1189.
14. Lemarié-Rieusset PG. Analyses multi-résolutions non orthogonales, commutation entre projecteurs et derivation et ondelettes vecteurs à divergence nulle. Revista Mathemática Iberoamericana. 1992; 8:221–238.
15. Urban K. On divergence-free wavelets. Adv Comput Math. 1995; 4:51–81.
16. Albukrek CM, Urban K, Rempfer D, Lumley JL. Divergence-free wavelet analysis of turbulent flows. J Sci Comput. 2002; 17:49–66.
17. Deriaz E, Perrier V. Towards a divergence-free wavelet method for the simulation of 2D/3D turbulent flows. J Turbul. 2006; 7:1–37.
18. Deriaz, E.; Domingues, MO.; Perrier, V.; Schneider, K.; Farge, M. Divergence-free wavelets for coherent vortex extraction in 3D homogeneous isotropic turbulence. European Series in Applied and Industrial Mathematics: Proceedings; Marseille, France. 2007. p. 146-163.
19. Donoho DL, Johnstone JM. Ideal spatial adaptation by wavelet shrinkage. Biometrika. 1994; 81:425–455.
20. Donoho DL, Johnstone IM. Adapting to unknown smoothness via wavelet shrinkage. J Am Stat Assoc. 1995; 90:1200–1224.
21. Coifman, RR.; Donoho, DL. Translation-invariant de-noising. New York: Springer; 1995.
22. Tafti PD, Unser M. On regularized reconstruction of vector fields. IEEE Trans Image Process. 2011; 20:3163–3178. [PubMed: 21659026]
23. Blu T, Luisier F. The SURE-LET approach to image denoising. IEEE Trans Image Process. 2007; 16:2778–2786. [PubMed: 17990754]

24. Gudbjartsson H, Patz S. The Rician distribution of noisy MRI data. Magn Reson Med. 2005; 34:910–914. [PubMed: 8598820]
25. Luisier F, Blu T, Unser M. A new sure approach to image denoising: interscale orthonormal wavelet thresholding. IEEE Trans Image Process. 2007; 16:593–606. [PubMed: 17357721]
26. Nickolls J, Buck I, Garland M, Skadron K. Scalable parallel programming with CUDA. Queue. 2008; 6:40–53.
27. Paige CC, Saunders MA. LSQR: an algorithm for sparse linear equations and sparse least squares. ACM Trans Math Softw. 1982; 8:43–71.
28. The OpenFOAMV® Foundation. Accessed on October 3, 2013.
29. Johnson KM, Markl M. Improved SNR in phase contrast velocimetry with five-point balanced flow encoding. Magn Reson Med. 2010; 63:349–355. [PubMed: 20099326]
30. Bernstein MA, Zhou XJ, Polzin JA, King KF, Ganin A, Pelc NJ, Glover GH. Concomitant gradient terms in phase contrast MR: analysis and correction. Magn Reson Med. 1998; 39:300–308. [PubMed: 9469714]
31. Markl M, Bammer R, Alley MT, Elkins CJ, Draney MT, Barnett A, Moseley ME, Glover GH, Pelc NJ. Generalized reconstruction of phase contrast MRI: analysis and correction of the effect of gradient field distortions. Magn Reson Med. 2003; 50:791–801. [PubMed: 14523966]
32. Walker PG, Cranney GB, Scheidegger MB, Waseleski G, Pohost GM, Yoganathan AP. Semiautomated method for noise reduction and background phase error correction in MR phase velocity data. J Magn Reson Imaging. 2005; 3:521–530. [PubMed: 8324312]
33. Lustig M, Donoho D, Pauly JM. Sparse MRI: the application of compressed sensing for rapid MR imaging. Magn Reson Med. 2007; 58:1182–1195. [PubMed: 17969013]
34. Zhang T, Pauly JM, Vasanawala SS, Lustig M. Coil compression for accelerated imaging with Cartesian sampling. Magn Reson Med. 2013; 69:571–582. [PubMed: 22488589]
35. Vasanawala, SS.; Murphy, MJ.; Alley, MT.; Lai, P.; Keutzer, K.; Pauly, JM.; Lustig, M. Practical parallel imaging compressed sensing MRI: summary of two years of experience in accelerating body MRI of pediatric patients. 2011 IEEE International Symposium on Biomedical Imaging: From Nano to Macro; Chicago, Illinois, USA. 2011. p. 1039-1043.
36. Uecker M, Lai P, Murphy MJ, Virtue P, Elad M, Pauly JM, Vasanawala SS, Lustig M. ESPIRiT-an eigenvalue approach to autocalibrating parallel MRI: where SENSE meets GRAPPA. Magn Reson Med. 2014; 71:990–1001.
37. Pruessmann KP, Weiger M, Scheidegger MB, Boesiger P. SENSE: sensitivity encoding for fast MRI. Magn Reson Med. 1999; 42:952–962. [PubMed: 10542355]
38. Hsiao A, Lustig M, Alley MT, Murphy MJ, Vasanawala SS. Evaluation of valvular insufficiency and shunts with parallel-imaging compressed-sensing 4D phase-contrast MR imaging with stereoscopic 3d velocity-fusion volume-rendered visualization. Radiology. 2012; 265:87–95. [PubMed: 22923717]

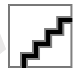

Fig. 1.
Visualization of DFW basis functions. **a**: Linear and quadratic spline scaling function $\phi(x)$ and wavelet function $\psi(x)$ that are used to construct DFWs. ϕ_0, ψ_0 and ϕ_1, ψ_1 are related by differentiation, thereby enabling the construction of DFWs. **b**: Examples of divergence-free components of DFW basis functions. **c**: Examples of nondivergence-free components of DFW basis functions. **d**: Flow diagram of DFW denoising. The entire procedure consists of applying separate wavelet transforms on each velocity component and linearly combine the coefficients, which achieves linear computational complexity overall. (FWT: forward wavelet transform, IWT: inverse wavelet transform, wc: wavelet coefficient, df: divergence-free, n: nondivergence-free). [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

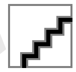

Fig. 2.
Velocity magnitude of simulation results on noisy CFD data with PVNR = 22 dB along with error magnitude maps. From left to right: original CFD data, noisy CFD data, DFW with two wavelet level decomposition, DFW with two wavelet level decomposition and partial cycle spinning, DFW with one wavelet level decomposition and partial cycle spinning, FDM and RBF. Both DFW and RBF show significant noise reduction in velocity magnitude, whereas FDM shows only marginal improvement. White arrows point to details that may be lost during denoising. Red arrow points to artifacts created by RBF. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

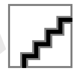

Fig. 3.
Vector visualization of simulation results on noisy CFD data with PVNR = 22 dB. Top row: Original CFD data, noisy CFD data, and DFW. Bottom row: DFW with partial cycle spinning, FDM and RBF. As in velocity magnitude, both DFW and RBF show significant noise reduction in vector visualization. Although FDM shows less improvement than other methods, it shows some noise suppression especially near the vortices on both sides. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

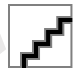

Fig. 4.

Top row: Velocity profiles for original, noisy, and DFW with partial cycle-spinning at three different slices for CFD data with PVNR = 22 dB. Bottom row: Simulation error statistics over a range of PVNRs of noisy CFD data. Comparisons are made between noisy CFD data, DFW, DFW with partial cycle spinning, DFW with full cycle spinning, FDM, and RBF. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

NIH-PA Author Manuscript

NIH-PA Author Manuscript

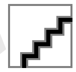

Fig. 5.

Velocity magnitude of results on phantom data with PVNR = 33.5 dB along with error magnitude maps. From left to right: original phantom data, noisy phantom data, DFW with partial cycle spinning, FDM, and RBF. White arrow points to artifact created in RBF denoising. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

Fig. 6.

Velocity magnitude of denoising results on phantom data with a coarsely chosen segmentation with PSNR = 33.5 dB along with error magnitude maps calculated within the correct segmentation. From left to right: Original phantom data, noisy phantom data, DFW, DFW with partial cycle spinning, FDM, and RBF. Both FDM and RBF show distortion, whereas DFW is largely unaffected by the change in segmentation. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

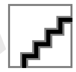

Fig. 7.

Velocity magnitude of DFW denoising on reconstructed undersampled data. From left to right: Original flow data, ESPIRiT reconstructed flow data, DFW with *SureShrink*, DFW with manually chosen thresholds, ESPIRiT with l_1 spatial wavelet regularization and the 5.4-fold Poisson-disk sampling mask used in simulation. The flow phantom k-space was retrospectively subsampled by 5.4 using the same sampling mask for each VENC. ESPIRiT with and without l_1 regularization were used afterward. Applying DFW with *SureShrink* reduces some incoherent artifacts in the reconstructed data, but is overly conservative. DFW with more aggressively chosen thresholds improves the performance significantly. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

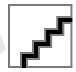

Fig. 8.
Velocity magnitude of DFW denoising results on L1-SPIRiT reconstructed in vivo data. Top row: SPIRiT reconstructed data with l_1 spatial wavelet regularization, DFW denoising with *SureShrink* on the reconstructed data and L-curve for DFW with manually specified thresholds. Bottom row: Results from DFW with low, medium, and high manual thresholds. White arrows point to details that are lost when a high threshold is applied. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

Fig. 9.

Left: Vector visualization of DFW denoising with manually chosen threshold on L1-SPIRiT reconstructed in vivo data. Thresholding only the non-DFW coefficients results in a more aligned flow field, while thresholding both divergence-free and non-DFW coefficients results in a cleaner flow field. Right: The top row shows streamlines released from the ascending aorta, whereas the bottom row shows streamlines released from the descending aorta, both comparing between L1-SPIRiT reconstructed data and DFW denoising with manually chosen thresholds. Red arrows point to streamlines that flow outside of the anatomy for the L1-SPIRiT reconstructed data but remains inside of the anatomy for DFW.

Table 1

Quantitative Results Across Eight Patient Data (Four with Regurgitant Fraction Less Than 5% and Another Four with Regurgitant Fraction Greater than 30%) Before and After Applying DFW with *SureShrink* and Manually Chosen Thresholds

Measurements (mean ± std)	Original	DFW *SureShrink*	DFW manual
Regurgitant fraction < 5%			
Flow rate (L/min)	2.934 ± 0.304	2.913 ± 0.302	3.021 ± 0.331
Percentage change in flow rate (%)		-0.1 ± 0.6	3.0 ± 3.1
Regurgitant fraction (%)	1.542 ± 1.284	1.375 ± 1.076	0.917 ± 1.330
Change in regurgitant fraction (%)		−0.167 ± 0.252	−0.625 ± 0.452
$(Q_p - Q_s)$ (L/min)	−0.019 ± 0.312	−0.031 ± 0.291	−0.034 ± 0.288
Regurgitant fraction >30%			
Flow rate (L/min)	2.056 ± 0.451	2.063± 0.444	2.171 ± 0.463
Percentage change in flow rate (%)		0.4 ± 1.3	6.0 ± 9.2
Regurgitant fraction (%)	19.417 ±21.786	19.333 ± 21.676	17.458 ± 20.002
Change in regurgitant fraction (%)		−0.083 ± 0.427	−1.958 ± 2.930
$(Q_p - Q_s)$ (L/min)	−0.022 ± 0.378	−0.010 ± 0.313	−0.007 ± 0.082