Resampling 4-D Images Using Adaptive Filtering

Alexander Andreopoulos and John K. Tsotsos
Department of Computer Science and Engineering
Centre for Vision Research
York University
Toronto Ontario, M3J 1P3, Canada
{alekos, tsotsos}@cs.yorku.ca

Abstract

We present an adaptive filtering based methodology for resampling 3-D time series images using an extension of the method presented by Westin in [11]. We simultaneously reduce the artifacts due to image noise and resample the data on a finer grid along the time dimension. This provides a methodology for obtaining high quality image resampling without the disadvantages of staircase artifacts created by more common interpolation methods such as linear interpolation. We present qualitative results of the algorithm on a data set of 4-D cardiac MRI. This is a useful approach for any situation where we have a data set of 4-D images needing to be resampled.

1. Introduction

In 2001, Cardiovascular Disease (CVD) contributed to almost one third of global deaths. CVD is among the leading causes of death in the developed world and by 2010, CVD is estimated to be the leading cause of death in developing countries [1]. As a result, three dimensional (3-D) and four dimensional (4-D) imaging of the heart (providing a 3-D time series of the cardiac cycle) using imaging modalities such as Magnetic Resonance Imaging (MRI) and Ultrasound, constitute a rapidly developing area of research in medical imaging [3]. Automated methods for extracting clinically useful information from these images are highly desirable. This has led to a vast amount of 4-D data needing to be processed.

One of the ultimate goals of research in medical imaging, is to build a reliable 4-D model of the heart for automated segmentation [9]. By incorporating prior knowledge about the typical deformation of the heart across the cardiac cycle, it should be capable of providing a robust methodology for segmenting the heart from a sequence of noisy volumetric images. This would lead to a reliable method for extracting useful indicators of the heart function such as the Ejection Fraction ratio (EF ratio).

When determining the EF ratio using MRI, we need to get the temporal boundaries of end systole and end diastole correct. The usual time sampling during the MRI image acquisition process is not set up to be optimal for this. An interpolation algorithm can be used to resample the data on a finer temporal grid, and detect the most likely time when end systole or end diastole occurs. The interpolated volumetric image depicting end systole or end diastole can then be used for the segmentation and EF ratio estimation. The need for high quality interpolation algorithms in 4-D, becomes apparent. Moreover, we often encounter 4-D MRI data of different time resolution. Many 3-D and 4-D cardiac segmentation algorithms (such as Active Appearance Models [8, 9]) sample the texture of the 3-D time series using a constant grid. If our data set consists of data with varying temporal resolution, we have to use interpolation algorithms to bring all the training data to a common temporal resolution.

For example in [2] the authors define Active Appearance Motion Models (AAMMs) as 2-D AAMs which incorporate a third temporal dimension and are used for robustly segmenting time varying cardiac data. The fact that not all training images contain the same number of time frames, leads the authors to using linear interpolation to estimate a common number of frames for all the training images and for detecting the exact moment when end diastole and end systole occur. All these reasons motivate the need for high quality image resampling.

A common problem with many methods for resampling images is that they tend to introduce staircase artifacts which in turn can lead to bad segmentations, especially in cases using thresholding techniques for classifying image structures. [11] presented a method which provides a good solution to the problem of staircase artifact reduction and high quality resampling of 3-D volumes.

In this paper we present an extension of [11] to 4-D and show how it can be used for resampling 3-D time series cardiac MRI. We provide qualitative comparisons of the algorithm and demonstrate its high quality resampling. Straight-

forward extensions of this algorithm can lead to a resampling of the 4-D image along any of the four dimensions. We accomplish high quality resampling using a combination of shifted lowpass and highpass filters. We use orientation estimation techniques to adaptively control the highpass filter and increase its weight in areas of strong orientation. This results in a resampling of the image, with a simultaneous reduction in image noise, and with minimal blurring of important image structures.

We begin with a review of the concepts of orientation representation using tensors, as introduced by Knutsson [7]. We then discuss the use of quadrature filters for adaptive filtering. We then show how all these ideas are used to give us an adaptive filtering based resampling method [4]. We end by giving qualitative evidence of our algorithm's ability to provide high quality resampling.

In 2-D, the smallest element of an image is called a pixel, in 3-D it is called a voxel, and by convention in this paper, due to lack of available terminology, we will be referring to the smallest element of a 4-D image as a toxel.

2. Orientation

The concept of orientation is fundamental in both computer and biological vision [4, 5]. The need for a good method of representing a local neighborhood's orientation led Knutsson to formulate a set of characteristics that such a representation should adhere to, which we discuss below.

2.1. Simple Neighborhoods and Orientation

In 2-D, a simple neighborhood is a noise free image region containing parallel edges and/or lines. To extend this definition of a simple neighborhood to higher dimensions we need to give a more formal definition. For an arbitrary non-constant function $g : \Re \rightarrow \Re$ define $f : \Re^n \rightarrow \Re$ by

$$f(\xi) = g(\xi^T \hat{\mathbf{x}}) \tag{1}$$

where $\|\hat{\mathbf{x}}\| = 1$. This defines a simple n-dimensional neighborhood. Notice that $f(\xi)$ is constant for all ξ such that $\xi^T \hat{\mathbf{x}}$ is equal to the same constant. In 2-D for example, this defines an image $f(\xi)$ containing parallel lines and edges, whose amplitude and frequency depend on g. Even though it is quite unlikely that any non-artificial image is of the form $f(\xi)$, this formalization of a simple neighborhood helps with the analysis we will perform later. We now need to find a way to represent this local neigborhood's orientation. We could use $\hat{\mathbf{x}}$ for this (a unit vector pointing in the intensity gradient direction), but we run into the problem that if we use $-\hat{\mathbf{x}}$ to represent this particular orientation and flip g around the y-axis, we have the same neighborhood. In other words this definition of orientation is dependent on the sign of $\hat{\mathbf{x}}$ and on g. We want a representation of orientation which is independent of g and the sign of $\hat{\mathbf{x}}$. For this we will use symmetric matrices, sometimes also referred to as second order tensors.

2.2. Representation of Simple Neighborhoods

By the discussion above we see that our representation of orientation in n dimensional space needs to satisfy certain requirements. In [4, 7] the above requirements are formalized as the uniqueness, invariance and equivariance requirements, which we explain below. It is shown that the space of $n \times n$ symmetric matrices satisfy the above requirements, and can be used to represent any orientation of a simple neighborhood.

The uniqueness requirement states that the orientations implied by $\hat{\mathbf{x}}$ and $-\hat{\mathbf{x}}$ must have the same representation. It turns out that the symmetric matrix $\mathbf{A} = \lambda \hat{\mathbf{x}} \hat{\mathbf{x}}^T$ can be used to represent this orientation. It is easy to see that $\mathbf{A}$ does not depend on the sign of $\hat{\mathbf{x}}$ so it satisfies the uniqueness requirement (where λ is a positive eigenvalue of $\mathbf{A}$, representing the frequency characteristics of the neighborhood).

The invariance requirement is that $\hat{\mathbf{A}} = \frac{\mathbf{A}}{\|\mathbf{A}\|}$ must not depend on g. In other words the orientation must not change as we change g. This is trivially true once we know either $\hat{\mathbf{x}}$ or $-\hat{\mathbf{x}}$ (ie: once we know the orientation). In practice we estimate the orientation using quadrature filters (which we discuss below, as well as in Appendix A), since they are phase invariant for sinusoidal g, and, therefore, provide us with a simple way of detecting edges and lines, that is not as sensitive to the function g. They also provide the means of estimating a value for λ.

The equivariance requirement is that small changes in the orientation should lead to small changes in $\mathbf{A}$. More formally, this can be defined by requiring that if we let $\mathbf{y} = \hat{\mathbf{x}} + \epsilon \mathbf{v}$ and $\mathbf{B} = \mathbf{y}^T \mathbf{y}$ then $lim_{\epsilon \rightarrow 0} \frac{\mathbf{B}-\mathbf{A}}{\epsilon}$ must exist [4].

3. Quadrature Filters

A quadrature filter is a filter q whose fourier transform Q is real valued and satisfies

$$Q(\mathbf{u}) = 0 \text{ if } \mathbf{u}^T \hat{\mathbf{n}}_k < 0 \tag{2}$$

where $\hat{\mathbf{n}}_k$ denotes the filter orientation. Quadrature filters are useful because their magnitude is phase invariant when applied to sinusoidal signals with a single frequency, thus providing a methodology of detecting edges and lines. In practice it is often useful to represent such a quadrature filter Q by a spherically separable lognormal filter for its nonzero values, given by

$$Q(\mathbf{u}) = R(\rho) D_k(\hat{\mathbf{u}}) \tag{3}$$

where

$$D_k(\hat{\mathbf{u}}) = \begin{cases} 0 & \text{if } \hat{\mathbf{u}}^T \hat{\mathbf{n}}_k < 0 \\ (\hat{\mathbf{u}}^T \hat{\mathbf{n}}_k)^2 & \text{otherwise} \end{cases} \tag{4}$$

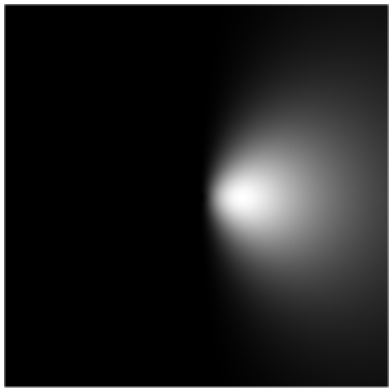

Figure 1. Lognormal 2-D quadrature filter with orientation $\mathbf{n_k} = (1, 0), \rho_i = 0.7, B = 3$.

and

$$R(\rho) = e^{-\frac{4}{B^2 ln2} ln^2(\frac{\rho}{\rho_i})}$$

with $\rho = \|\mathbf{u}\|$, $\hat{\mathbf{u}} = \frac{\mathbf{u}}{\|\mathbf{u}\|}$, and B, ρ_i are parameters controlling the bandwidth and center frequency of the filter respectively (see figure 1). For the case of a simple neighborhood, and when $g(x)$ is a sinusoidal function with a single frequency ω, it can be shown that quadrature filters are invariant with respect to the phase (see Appendix A for a proof adapted from [6]). They provide us with a way of estimating the orientation of a simple neighborhood in n-dimensions, regardless of whether it represents an edge or a line in 2-D, or some other structure in higher dimensions. This is the main reason why they are used for orientation estimation. However, even in the case when $g(x)$ is not sinusoidal, quadrature filters are still useful - even though the resulting filter output intensity will no longer be phase invariant - since they allow us to estimate the dominant orientation through the eigenvectors corresponding to the largest eigenvalue, by the method described below in section 4. From the appendix we can see that for the case of spherically separable filters the output magnitude of a filter with direction $\hat{\mathbf{n}}_k$ convolved with a simple neighborhood f, is equal to

$$\frac{1}{2} R(\omega)[D_k(-\hat{\mathbf{x}}) + D_k(\hat{\mathbf{x}})].$$

where ω is the frequency of the sinusoidal signal g. In the next section we will see how this helps us in estimating the orientation matrices for 4-D images.

4. Dual Bases and Frames

Assume we are given an $n \times n$ symmetric matrix $\mathbf{T} = \hat{\mathbf{x}}\hat{\mathbf{x}}^{\mathbf{T}}$. Then it can be shown that if $\{\hat{\mathbf{N}}_k\}_k$ and $\{\tilde{\mathbf{N}}_k\}_k$ form a dual basis pair of the set of $n \times n$ real symmetric matrices, $\mathbf{T}$ can be expressed as:

$$\mathbf{T} = \sum_k < \mathbf{T}, \hat{\mathbf{N}}_k > \tilde{\mathbf{N}}_k \tag{5}$$

where the norm is

$$< \mathbf{T}, \hat{\mathbf{N}}_k >= \sum_{i,j} \hat{\mathbf{N}}_k^{ij} \mathbf{T}^{ij} \tag{6}$$

with superscript ij indicating the ij^{th} element of the respective matrix.

By definition $\{\hat{\mathbf{N}}_k\}_k$ is a frame of the space $\mathbf{M}$ of $n \times n$ symmetric matrices if there exist constants $A > 0$,$B > 0$ such that for all $\mathbf{T} \in \mathbf{M}$

$$A\|\mathbf{T}\|^2 \leq \sum_k | < \mathbf{T}, \hat{\mathbf{N}}_k > |^2 \leq B\|\mathbf{T}\|^2. \tag{7}$$

Another way to phrase this for our purposes is that if we remove the condition that $\{\hat{\mathbf{N}}_k\}_k$ is an orthonormal basis of the space of $n \times n$ symmetric matrices and assume only that it spans the space of $n \times n$ symmetric matrices, then this basis is a frame. We will not go into many technical details but for the purposes of this paper, its dual frame is a frame $\{\tilde{\mathbf{N}}_k\}_k$ which satisfies (5). It is cited in the literature that for estimating orientation in 4-D it is preferable to use a frame representation of the space of 4×4 symmetric matrices such as the one we give in Appendix B, since this tends to give better results numerically [4, 6]. We will be using this basis in our paper.

In any case, it can be shown that

$$\mathbf{T} = \sum_k q_k \tilde{\mathbf{N}}_k \tag{8}$$

where $q_k = (\hat{\mathbf{x}}^T \hat{\mathbf{n}}_k)^2$, and the $\hat{\mathbf{n}}_k$ are the set of orientation vectors corresponding to the biorthogonal basis $\{\hat{\mathbf{N}}_k\}_k$ (given in appendix B for the 4-D case), and indicate the filter orientations of a set of quadrature filters and high pass filters we will be using later. But notice, from the previous section on quadrature filters and from appendix A, that for a simple neighborhood, the output magnitude of a lognormal quadrature filter (3) in direction $\hat{\mathbf{n}}_k$ is $p_k = \lambda q_k$ where $\lambda = \frac{1}{2} R(\omega)$ is independent of the filter orientation $\hat{\mathbf{n}}_k$ and only depends on the radial distribution of the simple neighborhood's frequency ω. λ is used to represent the degree to which the filter matches the frequency properties of the simple neighborhood.

5. Rank Interpretation

Notice that the matrix $\mathbf{A}$ above, is a symmetric matrix of rank 1. It is known from linear algebra, that any symmetric matrix $\mathbf{T}$ can be expressed as $\mathbf{T} = \lambda_1 \mathbf{e}_1 \mathbf{e}_1^T + ... + \lambda_n \mathbf{e}_n \mathbf{e}_n^T$ where $\lambda_1, ..., \lambda_n$ indicate the eigenvalues of $\mathbf{T}$, $\mathbf{e}_i$ are the

corresponding unit length eigenvectors, and n indicates the rank of the matrix. In the case of a simple neighborhood of a 2-D image, the orientation is represented by a rank 1 matrix.

However, the orientation matrix is not necessarily a rank 1 matrix for higher dimensional images. It is beyond the scope of this paper to discuss the interpretation of such orientations (see [4] for details). Depending on the application, we may interpret such an orientation as the rank 1 symmetric matrix closest to $\mathbf{T}$ in a least squares sense - thus dismissing the higher rank terms simply as unwanted artifacts - or we can use the rank to denote the complexity of the neighborhood.

For example in the 3-D case, if our local neighborhood consists of only parallel lines, then the fourier transform of this neighborhood will consist of a plane region in 3-D, denoting the orientation. In terms of our orientation matrix this will be represented by a rank 2 matrix with eigenvalues of approximately the same magnitude, and two eigenvectors spanning this plane region. These concepts can be extended to the 4-D case.

It can be seen that only in the case of a simple neighborhood are we guaranteed that the eigenvalues will all be positive (an indirect assumption we made above, in our interpretation of the orientation tensor). In practice our quadrature filter estimation techniques often lead to symmetric matrices with some negative eigenvalues. By convention, we typically set such eigenvalues to be zero [6], since we have no physical interpretation for them in our representation of orientation.

6. Adaptive Filtering

Adaptive filtering is used to remove image noise without blurring important image structures [4]. We can use the orientation information estimated using the previously described methods, to adaptively filter images. The general idea is in some ways similar to the concept of anisotropic diffusion for image enhancement [10].

Intuitively, if an image region has no, or very weak orientation - estimated using the eigenvalues of the previously defined symmetric matrix - we apply a lowpass filter H. If the region has strong orientation, we wish to preserve and enhance this structure, so we perform a combination of lowpass filtering and highpass filtering, by filtering the region with a linear combination of the lowpass filter H with a highpass filter $1-H$. In other words we filter the image region by $H + c(1-H)$, where c is a positive constant which depends on the region's orientation. Notice that the greater the values of c, the more the filter enhances high frequency structures such as lines and edges, while $c = 1$ gives an allpass filter. We need to define c more precisely and indicate how we can estimate it from the orientation information that we can derive with quadrature filters.

Consider the filter

$$H(\mathbf{u}) + (1 - H(\mathbf{u}))(\lambda_1(\hat{\mathbf{e}}_1^T\hat{\mathbf{u}})^2 + ... + \lambda_n(\hat{\mathbf{e}}_n^T\hat{\mathbf{u}})^2) \quad (9)$$

where $\hat{\mathbf{u}} = \frac{\mathbf{u}}{\|\mathbf{u}\|}$. Notice that the result of this filter on an image, is to enhance the image structure whose orientation matrix representation is similar to $\mathbf{T} = \lambda_1\hat{\mathbf{e}}_1\hat{\mathbf{e}}_1^T + ... + \lambda_n\hat{\mathbf{e}}_n\hat{\mathbf{e}}_n^T$ (by assigning a higher value to the c-value of the highpass filter $1-H$) and lowpass filter with a filter closer to $H(\mathbf{u})$, the rest of the image which has an orientation structure very different from $\mathbf{T}$. However, we want to use an efficient filter which adaptively filters each image region according to that region's orientation, so that it enhances the respective region's structure. Let $\hat{\mathbf{e}}_i(\mathbf{x})$ denoted the i^{th} eigenvector of the orientation matrix $\mathbf{T}(\mathbf{x})$ at a pixel/voxel/toxel $\mathbf{x}$. It can be shown that

$$H + (1 - H)(\lambda_1(\hat{\mathbf{e}}_1^T(\mathbf{x})\hat{\mathbf{u}})^2 + ... + \lambda_n(\hat{\mathbf{e}}_n^T(\mathbf{x})\hat{\mathbf{u}})^2) =$$

$$H + \sum_{k=1}^{\alpha} < \tilde{\mathbf{N}}_k, \mathbf{T}(\mathbf{x}) > H_k \quad (10)$$

where $H_k(\mathbf{u}) = (1 - H(\mathbf{u}))(\hat{\mathbf{n}}_k^{\mathbf{T}}\hat{\mathbf{u}})^2$ and $\alpha \geq n(n+1)/2$ and n is the dimension of the image we are filtering. By choosing $\alpha > n(n+1)/2$ we are using a frame that is not an orthonormal basis, which can give us numerically more stable results. For the case $n = 4$ in this paper, $\alpha = 12$ and we use a dual pair of frames (given in appendix B), since this tends to give better results in practice [4]. But notice, however, that in (10), the only location where we use the orientation matrix, is in the norm, which does not depend on the variable $\mathbf{u}$.

So we can perform adaptive filtering by filtering the image with the filters $\{H, H_1, ..., H_\alpha\}$ and then, for every pixel/voxel/toxel $\mathbf{x}$ of the n-dimensional image estimate the value of $< \tilde{\mathbf{N}}_k, \mathbf{T}(\mathbf{x}) >$ and use (10) to get the adaptive filtered image.

By modifying the eigenvalues of the matrices $\mathbf{T}$ in the image we can enhance certain image structures such as lines and edges, as is shown in [11]. It is also recommended that we use a gaussian filter to smooth the matrix representations $\mathbf{T}(\mathbf{x})$ (by Gaussian filtering each matrix element of $\mathbf{T}(\mathbf{x})$), since in practice this has been shown to significantly reduce the effects of noise on the orientation matrix and results in better enhanced images [4].

7. Filter Shifts

Given a continuous filter $f(\mathbf{x})$ and its shifted and scaled version $g(\mathbf{x}) = f(\mathbf{A}(\mathbf{x} + \mathbf{b}))$, its fourier transform is $G(\mathbf{u}) = \frac{1}{det|A|}e^{i\mathbf{u}^T\mathbf{b}}F(|\mathbf{A}^T|^{-1}\mathbf{u})$. This has two implications. By setting $\mathbf{b} \in [0,1]^{n\times 1}$ and $\mathbf{A} = \mathbf{I}$ (the identity matrix) we have a filter with a subpixel/subvoxel/subtoxel shift

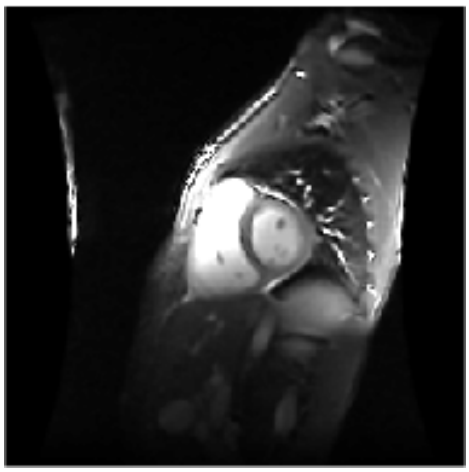

Figure 2. An x-y axis view of a resampled image, at a particular height (z-value) and particular time instance. 256x256 pixels.

Figure 3. A y-z axis view of a resampled image, at a particular x-axis value and a particular time instance. 84x256 pixels (showing the region containing the heart).

- for the 2-D/3-D/4-D case respectively -. The resulting discrete filter allows us to perform subpixel/subvoxel/subtoxel interpolation and thus get a finer sampling grid. In the case of 3-D resampling, this has been shown to be at least qualitatively superior than more common interpolation methods used in medical imaging, such as linear interpolation, spline based interpolation and sinc interpolation [11].

By appropriately changing the matrix $\mathbf{A}$, we can control the filter's sensitivity along differing axes. For example, in the 2-D case, if the sampling distance along the x-axis was twice the sampling distance along the y-axis, we need to account for this pixel anisotropy if we want our filter to show the same sensitivity for structures along both axes. This can be done by setting

$$\mathbf{A} = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}.$$

By appropriately changing the matrix $\mathbf{A}$, we can also make filters designed for signals acquired with a non-evenly spaced sampling grid or simply filters which show different sensitivity to structures along different axes.

In the experiments below, we demonstrate the method's applicability for resampling 4-D data acquired from cardiac MRI to a finer grid. This method can help us in cases where we need to resample time varying data, as described in the introduction.

8. Results

We tested our algorithm on a set of 3-D time series short axis cardiac MRI. The data was made up of 10 time samples of the cardiac cycle. Each time sample consisted of 84 slices of the heart along the z-axis, and each one of these slices was a 256×256 pixel image. The original data had a z-axis dimension of 14 slices only, so we initially used Westin's method [11] to resample each 3-D volume to a z-axis resolution of 84 slices, so that we minimized the voxel anisotropy along the z axis (see figure 3).

The implementation and execution of our test cases was done on an Intel Xeon 3.06 Ghz with 3GB RAM using Matlab 7.01. The image sequence took approximately 400MB to store in memory. We resampled the data along the time dimension, resulting in a four fold resolution increase along the time dimension. We resampled the data by using filters shifted by $0, 0.25, 0.5$ and 0.75 toxels along the time dimension. The resulting data had a dimension of $256 \times 256 \times 84 \times 40$ toxels. The parameters we used for the lognormal filters which evaluated the orientation matrices were $\rho_i = 0.7$, $B = 3$ and the scaling matrix from section 7 was

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 2 \end{pmatrix}.$$

We used a scaling of 1 for the x, y and z dimensions since the sampling grid along those dimensions was isotropic. Deciding the scaling for the time dimension was more difficult, since time is measured in different units than the x,y,z coordinates. We chose a scaling of 2, since this scale gave slightly better resampling results than some other values we experimented with.

In figures 2-6 we present our results. Figures 2 and 3 show a particular time instance's volume acquired from a filter that was shifted 0.5 toxels in the time dimension. Figures 4-6 show the results using various resampling methods. The displayed data are slices from the solution acquired by holding the x and z coordinates constant. The respective z

values are indicated by the figures.

Figure 4 shows the data with no interpolation method. Figure 5 uses linear interpolation and figure 6 uses our adaptive filtering based method. An inspection of figure 5 shows that linear interpolation creates staircase artifacts, especially in regions of high curvature. As it can be seen in figure 6, much of those staircase artifacts are removed when we use the adaptive filtering method, while simultaneously denoising and enhancing regions with strong orientation. Moreover, it is seen in figure 6 that the adaptive filtering resampling method makes it much easier to discern the moment when the left ventricle has contracted the most, thus making it easier to detect end systole. The higher the resampling that we perform along the time axis, the more obvious do the advantages of the adaptive filtering based method become, compared to other methodologies. Admittedly, it is difficult to quantitatively evaluate such a method, but our preliminary results are quite promising we believe.

9. Conclusion

We have presented a high quality resampling algorithm for 3D time series data. Since the most common source of such data is from MRI, we presented our results from resampling a sequence of 3-D time series short axis cardiac MRI. This algorithm can be used to make a better estimate of the time when end systole and end diastole occur, can be used as a method for aligning a set of images to a common sampling grid while simultaneously denoising the image without blurring important structures, or in any situation in general where we need high quality resampling. More work is needed to understand how the algorithm parameters affect the quality of the results. A bad parameter choice can significantly degrade the resampling quality, and quite likely, a more fine tuned choice of parameters would lead to even better results.

Acknowledgements

We thank Dr. Paul Babyn and Dr. Shi-Joon Yoo of the Department of Diagnostic Imaging at the Hospital for Sick Children in Toronto for providing us with the MRI data. JKT holds a Canada Research Chair in Computational Vision and acknowledges its financial support. AA holds an NSERC PGS-M and acknowledges its financial support.

Appendix A

Theorem. *Quadrature filters are phase invariant for simple neighborhoods $f(\xi) = g(\xi^T \hat{\mathbf{x}})$, where g is a sinusoidal signal with a single frequency.*

Proof. Without loss of generality assume $g(x) = cos(\omega x + \phi)$. We want to show that for a simple neighborhood $f(\xi) = g(\xi^T \hat{\mathbf{x}})$, and for a quadrature filter q, the magnitude of $f * q(\mathbf{0})$ is phase invariant (does not depend on the value of ϕ).

$$\begin{aligned}
(f * q)(\mathbf{0}) &= \int_{-\infty}^{+\infty} f(-\mathbf{x}) q(\mathbf{x}) d^n\mathbf{x} \\
&= \int_{-\infty}^{+\infty} f^{**}(-\mathbf{x}) q(\mathbf{x}) d^n\mathbf{x} \\
&= \int_{-\infty}^{+\infty} (f^*(-\mathbf{x}))^* q(\mathbf{x}) d^n\mathbf{x} \\
&= \frac{1}{(2\pi)^n} \int_{-\infty}^{+\infty} (F^*(\mathbf{u}))^* Q(\mathbf{u}) d^n\mathbf{u} \\
&\quad \text{(By the Power theorem and the lemma)} \\
&= \frac{1}{(2\pi)^n} \int_{-\infty}^{+\infty} F(\mathbf{u}) Q(\mathbf{u}) d^n\mathbf{u} \; (*)
\end{aligned}$$

where f^* denotes the complex conjugate of a function f. But since F has an impulse line it follows from (*) that [6]:

$$\begin{aligned}
(f * q)(\mathbf{0}) &= \frac{1}{2\pi} \int_{-\infty}^{+\infty} G(u) Q(u\hat{\mathbf{x}}) du \\
&= \frac{1}{2\pi} \int_{0}^{+\infty} G(u) Q(u\hat{\mathbf{x}}) du + \\
&\quad \frac{1}{2\pi} \int_{0}^{+\infty} G(-u) Q(-u\hat{\mathbf{x}}) du \\
&= \frac{1}{2} \int_{0}^{+\infty} [\delta(u+\omega) e^{-i\phi}] Q(u\hat{\mathbf{x}}) du + \\
&\quad \frac{1}{2} \int_{0}^{+\infty} [\delta(u-\omega) e^{i\phi}] Q(u\hat{\mathbf{x}}) du + \\
&\quad \frac{1}{2} \int_{0}^{+\infty} [\delta(u-\omega) e^{-i\phi}] Q(-u\hat{\mathbf{x}}) du + \\
&\quad \frac{1}{2} \int_{0}^{+\infty} [\delta(u+\omega) e^{i\phi}] Q(-u\hat{\mathbf{x}}) du \\
&= \frac{1}{2} [Q(\omega\hat{\mathbf{x}}) e^{i\phi} + Q(-\omega\hat{\mathbf{x}}) e^{-i\phi}]
\end{aligned}$$

where $\hat{\mathbf{x}}$ denotes the unit eigenvector of the symmetric matrix representing the orientation. But since one of the two Q terms is zero for each ω, the magnitude of the above equation is either $|\frac{1}{2} Q(\omega\hat{\mathbf{x}})|$ or $|\frac{1}{2} Q(-\omega\hat{\mathbf{x}})|$ which does not depend on ϕ, as wanted. In the case of spherically separable filters it is seen that the above magnitude is equal to

$$\frac{1}{2} R(\omega) [D_k(-\hat{\mathbf{x}}) + D_k(\hat{\mathbf{x}})]$$

where D_k and R were defined in section 3, and are obviously independent of ϕ. □

Lemma. *The Fourier Transform of $f^*(-\mathbf{x})$ is $F^*(\mathbf{u})$.*

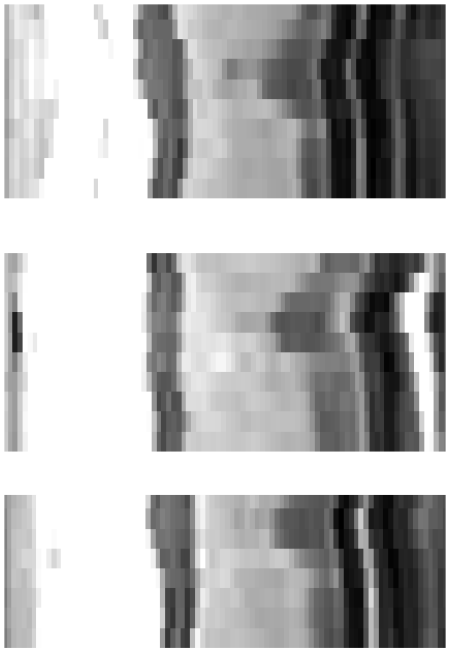

Figure 4. y-axis vs time-axis views of the original data, taken for 3 different z-values (z=41,30,45).

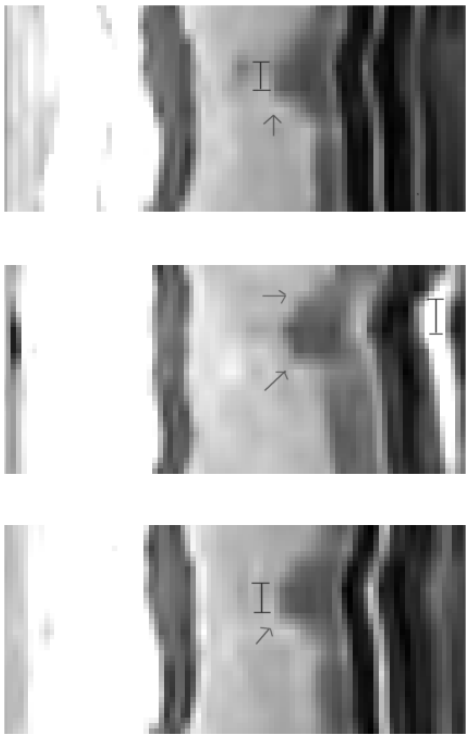

Figure 5. y-axis vs time-axis views of the original data resampled using linear interpolation for z=41,30,45. Arrows point to staircase artifacts, and interval markers indicate blurred peaks.

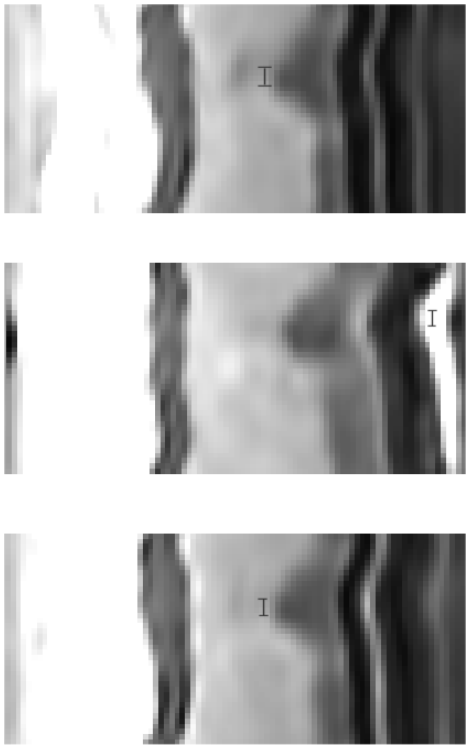

Figure 6. y-axis vs time-axis views of the original data resampled using the adaptive filtering method, taken for 3 different z-values (z=41,30,45). Notice the reduction in staircase artifacts as compared to figures 4 and 5, and that it is easier to discern the exact moment when the left ventricle has contracted the most. Notice our interval markers in the image, showing how much easier it becomes to detect the peaks in areas of high curvature as compared to figures 4 and 5. The reduction of the interval length in the top, middle and bottom images respectively is approximately 40%, 60% and 50%. See original images in electronic form for a more crisp view.

Proof.

$$\int_{-\infty}^{+\infty} f^*(-\mathbf{x})e^{-i\mathbf{u}^T\mathbf{x}}d^n\mathbf{x} = [\int_{-\infty}^{+\infty} f(-\mathbf{x})e^{i\mathbf{u}^T\mathbf{x}}d^n\mathbf{x}]^*$$
$$= [\int_{-\infty}^{+\infty} f(\mathbf{x})e^{-i\mathbf{u}^T\mathbf{x}}d^n\mathbf{x}]^* = F^*(\mathbf{u})$$

□

Appendix B

Let $c = \frac{1}{\sqrt{2}}$, $d = \frac{1}{2}$, $\bar{d} = -\frac{1}{2}$, $e = \frac{1}{6}$, $\bar{e} = -\frac{1}{6}$, $f = \frac{2}{6}$, $\bar{f} = -\frac{2}{6}$, $g = \frac{3}{6}$, $\bar{g} = -\frac{3}{6}$. The twelve vectors below are the filter directions along which we filtered our set of twelve quadrature filters and twelve highpass filters.

$$\hat{\mathbf{n}}_\mathbf{1} = c\begin{pmatrix} 1,1,0,0 \end{pmatrix}^T, \hat{\mathbf{n}}_\mathbf{2} = c\begin{pmatrix} 1,-1,0,0 \end{pmatrix}^T,$$
$$\hat{\mathbf{n}}_\mathbf{3} = c\begin{pmatrix} 1,0,1,0 \end{pmatrix}^T, \hat{\mathbf{n}}_\mathbf{4} = c\begin{pmatrix} 1,0,-1,0 \end{pmatrix}^T,$$
$$\hat{\mathbf{n}}_\mathbf{5} = c\begin{pmatrix} 1,0,0,1 \end{pmatrix}^T, \hat{\mathbf{n}}_\mathbf{6} = c\begin{pmatrix} 1,0,0,-1 \end{pmatrix}^T,$$
$$\hat{\mathbf{n}}_\mathbf{7} = c\begin{pmatrix} 0,1,1,0 \end{pmatrix}^T, \hat{\mathbf{n}}_\mathbf{8} = c\begin{pmatrix} 0,1,-1,0 \end{pmatrix}^T,$$
$$\hat{\mathbf{n}}_\mathbf{9} = c\begin{pmatrix} 0,1,0,1 \end{pmatrix}^T, \hat{\mathbf{n}}_\mathbf{10} = c\begin{pmatrix} 0,1,0,-1 \end{pmatrix}^T,$$
$$\hat{\mathbf{n}}_\mathbf{11} = c\begin{pmatrix} 0,0,1,1 \end{pmatrix}^T, \hat{\mathbf{n}}_\mathbf{12} = c\begin{pmatrix} 0,0,1,-1 \end{pmatrix}^T.$$

The following two sets of matrices $\{\hat{\mathbf{N}}_\mathbf{k}\}_{k=1}^{12}$ and $\{\tilde{\mathbf{N}}_\mathbf{k}\}_{k=1}^{12}$ form a frame of the space of 4-D symmetric matrices. Notice that $\hat{\mathbf{N}}_\mathbf{k} = \hat{\mathbf{n}}_\mathbf{k}\hat{\mathbf{n}}_\mathbf{k}^\mathbf{T}$.

$$\hat{\mathbf{N}}_\mathbf{1} = \begin{pmatrix} d & d & 0 & 0 \\ d & d & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \hat{\mathbf{N}}_\mathbf{2} = \begin{pmatrix} d & \bar{d} & 0 & 0 \\ \bar{d} & d & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$
$$\hat{\mathbf{N}}_\mathbf{3} = \begin{pmatrix} d & 0 & d & 0 \\ 0 & 0 & 0 & 0 \\ d & 0 & d & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \hat{\mathbf{N}}_\mathbf{4} = \begin{pmatrix} d & 0 & \bar{d} & 0 \\ 0 & 0 & 0 & 0 \\ \bar{d} & 0 & d & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$
$$\hat{\mathbf{N}}_\mathbf{5} = \begin{pmatrix} d & 0 & 0 & d \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ d & 0 & 0 & d \end{pmatrix}, \hat{\mathbf{N}}_\mathbf{6} = \begin{pmatrix} d & 0 & 0 & \bar{d} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \bar{d} & 0 & 0 & d \end{pmatrix}$$
$$\hat{\mathbf{N}}_\mathbf{7} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & d & d & 0 \\ 0 & d & d & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \hat{\mathbf{N}}_\mathbf{8} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & d & \bar{d} & 0 \\ 0 & \bar{d} & d & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$
$$\hat{\mathbf{N}}_\mathbf{9} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & d & 0 & d \\ 0 & 0 & 0 & 0 \\ 0 & d & 0 & d \end{pmatrix}, \hat{\mathbf{N}}_\mathbf{10} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & d & 0 & \bar{d} \\ 0 & 0 & 0 & 0 \\ 0 & \bar{d} & 0 & d \end{pmatrix}$$
$$\hat{\mathbf{N}}_\mathbf{11} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & d & d \\ 0 & 0 & d & d \end{pmatrix}, \hat{\mathbf{N}}_\mathbf{12} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & d & \bar{d} \\ 0 & 0 & \bar{d} & d \end{pmatrix}$$

$$\tilde{\mathbf{N}}_\mathbf{1} = \begin{pmatrix} f & g & 0 & 0 \\ g & f & 0 & 0 \\ 0 & 0 & \bar{e} & 0 \\ 0 & 0 & 0 & \bar{e} \end{pmatrix}, \tilde{\mathbf{N}}_\mathbf{2} = \begin{pmatrix} f & \bar{g} & 0 & 0 \\ \bar{g} & f & 0 & 0 \\ 0 & 0 & \bar{e} & 0 \\ 0 & 0 & 0 & \bar{e} \end{pmatrix}$$
$$\tilde{\mathbf{N}}_\mathbf{3} = \begin{pmatrix} f & 0 & g & 0 \\ 0 & \bar{e} & 0 & 0 \\ g & 0 & f & 0 \\ 0 & 0 & 0 & \bar{e} \end{pmatrix}, \tilde{\mathbf{N}}_\mathbf{4} = \begin{pmatrix} f & 0 & \bar{g} & 0 \\ 0 & \bar{e} & 0 & 0 \\ \bar{g} & 0 & f & 0 \\ 0 & 0 & 0 & \bar{e} \end{pmatrix}$$
$$\tilde{\mathbf{N}}_\mathbf{5} = \begin{pmatrix} f & 0 & 0 & g \\ 0 & \bar{e} & 0 & 0 \\ 0 & 0 & \bar{e} & 0 \\ g & 0 & 0 & f \end{pmatrix}, \tilde{\mathbf{N}}_\mathbf{6} = \begin{pmatrix} f & 0 & 0 & \bar{g} \\ 0 & \bar{e} & 0 & 0 \\ 0 & 0 & \bar{e} & 0 \\ \bar{g} & 0 & 0 & f \end{pmatrix}$$
$$\tilde{\mathbf{N}}_\mathbf{7} = \begin{pmatrix} \bar{e} & 0 & 0 & 0 \\ 0 & f & g & 0 \\ 0 & g & f & 0 \\ 0 & 0 & 0 & \bar{e} \end{pmatrix}, \tilde{\mathbf{N}}_\mathbf{8} = \begin{pmatrix} \bar{e} & 0 & 0 & 0 \\ 0 & f & \bar{g} & 0 \\ 0 & \bar{g} & f & 0 \\ 0 & 0 & 0 & \bar{e} \end{pmatrix}$$
$$\tilde{\mathbf{N}}_\mathbf{9} = \begin{pmatrix} \bar{e} & 0 & 0 & 0 \\ 0 & f & 0 & g \\ 0 & 0 & \bar{e} & 0 \\ 0 & g & 0 & f \end{pmatrix}, \tilde{\mathbf{N}}_\mathbf{10} = \begin{pmatrix} \bar{e} & 0 & 0 & 0 \\ 0 & f & 0 & \bar{g} \\ 0 & 0 & \bar{e} & 0 \\ 0 & \bar{g} & 0 & f \end{pmatrix}$$
$$\tilde{\mathbf{N}}_\mathbf{11} = \begin{pmatrix} \bar{e} & 0 & 0 & 0 \\ 0 & \bar{e} & 0 & 0 \\ 0 & 0 & f & g \\ 0 & 0 & g & f \end{pmatrix}, \tilde{\mathbf{N}}_\mathbf{12} = \begin{pmatrix} \bar{e} & 0 & 0 & 0 \\ 0 & \bar{e} & 0 & 0 \\ 0 & 0 & f & \bar{g} \\ 0 & 0 & \bar{g} & f \end{pmatrix}$$

References

[1] American Heart Association, "International Cardiovascular Disease Statistics," [Online]. Available: http://www.americanheart.org, 2004.

[2] J. G. Bosch, S. C. Mitchell, B. P. F. Lelieveldt, F. Nijland, O. Kamp, M. Sonka "Automatic Segmentation of Echocardiographic Sequences by Active Appearance Motion Models" *IEEE Transactions on Medical Imaging*, Vol. 21, No. 11, pp.1374-1383, 2002.

[3] A.F. Frangi, W.J. Niessen, and M.A. Viergever, "Three-Dimensional Modeling for Functional Analysis of Cardiac Images: A Review," *IEEE Transactions on Medical Imaging*, Vol.20, No. 1, pp. 2-25, 2001.

[4] G.H. Granlund and H. Knutsson, “Signal Processing for Computer Vision ,” *Kluwer Academic Publishers*, 1995.

[5] D. H. Hubel and T. N. Wiesel, “Brain Mechanisms of Vision,” *Scientific American*, September 1978.

[6] H. Knutsson “Edupack: Orientation ,” Technical Report [Online] April 2000.

[7] H. Knutsson “Representing Local Structure Using Tensors” *In Proceedings of the 6th Scandinavian Conference on Image Analysis*, pp. 244-251, June 1989.

[8] R. M. Lapp, M. Lorenzo-Valdes and D. Rueckert ”3D/4D Cardiac Segmentation Using Active Appearance Models, Non-rigid Registration and the Insight Toolkit” *In Proceedings of the 7th International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI)*, pp.419-426, 2004

[9] S. C. Mitchell, J. G. Bosch, B. P. F. Lelieveldt, R. J. van der Geest, J. H. C. Reiber, and M. Sonka, “3-D Active Appearance Models: Segmentation of Cardiac MR and Ultrasound Images,” *IEEE Transaction on Medical Imaging*, Vol.21, No. 9, pp. 1167-1178, 2002.

[10] P. Perona and J. Malik “Scale-Space and Edge Detection Using Anisotropic Diffusion,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.12, No. 7, pp. 629-639, 1990.

[11] C.F. Westin, J. Richolt, V. Moharir and R. Kikinis, “Affine Adaptive Filtering of CT Data,” *Medical Image Analysis* Vol. 4, No. 2, pp. 161-177, 2000.