DIFFUSION TENSOR IMAGES EDGE-DIRECTED INTERPOLATION

Zhenhua Mai¹, Wolfgang Jacquet¹, Marleen Verhoye², Jan Sijbers¹

[1]: IBBT-Visionlab, Physics Department, Universiteit Antwerpen, Belgium.
[2]: Bio-Imaging Lab, Biomedical Department, Universiteit Antwerpen, Belgium.

ABSTRACT

It has been demonstrated that, for scalar images, edge-directed interpolation techniques are able to produce better results, both visually and quantitatively, than non-adaptive traditional interpolation methods. We have extended the edge-directed concept to the interpolation of multi-valued diffusion tensor images. The interpolation framework is based on the improved edge-directed scalar image interpolation, which does not require estimation of local edges, and which interpolates the tensors in their mathematically native Riemannian space in order to respect tensor geometry of positive semi-definiteness. The framework has been tested on phantom scalar images and real diffusion tensor images, and is shown to be able to achieve better results than the Euclidean non-adaptive methods.

Index Terms— Edge-directed interpolation, Diffusion Tensor Images, Riemannian space

1. INTRODUCTION

Image interpolation, in the sense of upscaling the image to a higher resolution, is generally an ill-posed problem, since the number of unknown high-resolution pixels usually exceeds that of the known low-resolution pixels. Certain models concerning the relation between high-resolution pixels and low-resolution ones will have to be used in order to interpolate the high-resolution pixels from the low-resolution ones. The most widely used conventional interpolation methods, such as bi-linear interpolation and bicubic interpolation, readily employ space-invariant models that fail to respect the statistics around edges in the image. Consequently they produce artifacts such as jagged edges, blurring and edge halos.

In order to improve the interpolation quality, numerous methods that involve models of more sophistication have been proposed[1, 2, 3, 4, 5, 6]. Adaptive interpolation techniques [1] exploit the spatial relation of intensities to adapt the interpolation coefficients to better capture local features around the edges. Iterative methods such as PDE-based methods [4] impose constraints according to edge information and converge toward solution through iterations. Edge-directed interpolation (EDI) techniques [2, 3, 5] employ models that extract edge information in various ways in order to guide the interpolation to better reflect the image structures. It should also be mentioned that the isophote-based method [4] can also be regarded as part of the edge-directed interpolation group. The edge-directed interpolation techniques are shown to be quite successful in reducing those undesirable artifacts aforementioned. However, the pitfall of those techniques are that they have to explicitly extract the edge information, and/or discretize the interpolation direction into a finite set. The successful extraction of edge information still presents a challenge up to date, especially for certain low-contrast medical images as well as for multi-valued images, while a finite set of interpolation directions can prove to be unstable. In order to avoid the problems associated with explicitly estimating edges, Li et al proposed to exploit the “geometric duality” between the covariance of a low-resolution image and a high-resolution image, with the help of linear prediction theory, and guide the interpolation with the implicitly retained edge information in covariance [6]. Harnessing the mathematical elegance, their method (NEDI) is able to produce results both visually pleasing and quantitatively competitive. However, the original algorithm still leaves room for improvements, which will form the base for our diffusion tensor images (DTI) edge-directed interpolation method.

DTI processing heavily involves interpolations in both pre- and post-processing tasks such as spatial normalization, registration, and tractography. Due to the computational demand as well as implementation ease, the most commonly used interpolation techniques are bi(tri)-linear interpolation and bicubic interpolation that directly operates on the tensor matrix coefficients, also denoted as Euclidean interpolation. However, this practice does not respect the positive semi-definiteness of the diffusion tensors, and may also result in certain “swelling” artifacts, as shown by Pennec et al in [7]. They also proposed a Riemannian framework to accommodate tensor processing in their mathematically native Riemannian space. For example, a linear tensor interpolation between two neighbor tensors can be viewed as a walk on the Riemannian geodesic with end nodes defined as both neighbors. Furthermore, interpolation that involves bigger neighborhood can be done via the Riemannian intrinsic weighted sum.

Our framework is inspired by the Riemannian interpolation scheme that preserves the tensor geometry, and our aim
is also to preserve the edge information that is inherent in the tensor image. However, we did not adopt the Riemannian edge detection scheme, mainly because of the aforementioned drawbacks caused by an explicit edge-estimation scheme. Instead, we supply a scalar map, e.g., Fractional Anisotropy (FA), that contains sufficient information on tensor edges, into our improved EDI to get the interpolation weights. Finally, in conjunction with a Riemannian interpolation scheme, we obtain the edge-directed tensor interpolation.

This paper will be organized as follows: Section 2 will cover our methodology: Firstly, section 2.1 will briefly introduce NEDI; secondly, section 2.2 will detail our improvements upon it; and then section 2.3 will delve into the detail of our Adaptive Riemannian Tensor Interpolation method. Afterward, section 3 will give the results we have obtained from both phantom and real image tests, and finally section 4 will give some conclusion as well some prospect for future work.

2. METHODS

2.1. NEDI

Without loss of generality, we assume the low-resolution image $X_{i,j}$, to be defined on a domain $\Omega$ with $\Omega = \{(i, j)|1 \leq i \leq W, 1 \leq j \leq H\}$, to be a downsampled version of the high resolution image $Y_{2i−1,2j−1}$ of dimension $(2W−1) \times (2H−1)$, i.e., $Y_{2i−1,2j−1} = X_{i,j}$, as shown in Fig. 1. The goal of interpolation is to compute pixels $Y_{2i,2j}$, $Y_{2i−1,2j}$ and $Y_{2i,2j−1}$. Let’s start from the reconstruction of $Y_{2i,2j}$. It is assumed that this value can be reconstructed from its immediate four neighbors:

$$\hat{Y}_{2i,2j} = \sum_{k=0}^{1} \sum_{l=0}^{1} \omega_{2k+l} Y_{2(i+k−1),2(j+l−1)}$$  \hspace{1cm} (1)

According to Wiener filtering theory[8], the optimal interpolation coefficients are given by:

$$\vec{\alpha} = R^{-1} \vec{r}$$  \hspace{1cm} (2)

where $R$ and $\vec{r}$ are the local covariance for image signal, the superscript ‘h’ stands for “high resolution”. Due to the “geometric duality” [6] that couples the high-resolution covariance and the low-resolution one, $R^h$ and $r^h$ can be estimated from the low-resolution counterpart:

$$R = \frac{1}{4} C^T C, \vec{r} = \frac{1}{4} C^T \vec{y}$$  \hspace{1cm} (3)

Now that the pixels $Y_{2i,2j}$ (the gray pixels in Fig. 1) are obtained, one may proceed to compute the pixels $Y_{2i−1,2j}$ and $Y_{2i,2j−1}$ (the white pixel in Fig. 1). In fact, it can be readily observed that the lozenge-shape of the white pixel’s neighborhood is isomorphic to a square one up to a scaling factor of $2^{\frac{1}{2}}$ and a rotation operation of $\pi/4$. Therefore, with a neighborhood thus selected, one can repeat the previous steps for the gray pixels on the white pixels and compute their values.

Here we would like to summarize the main shortcomings of the original NEDI algorithm. First, in order to reduce the computational complexity, NEDI differentiates between edge pixels and non-edge pixels by a variance threshold of pixel values in the $4 \times 4$-neighborhood. It is only applied on edge pixels, while for non-edge pixels only bi-linear interpolation is used. This in practice could undermine the competitiveness of NEDI against higher order non-adaptive interpolation methods such as bicubic interpolation when faced with images of less defined edges. Second, the training matrix $C$ largely determines the outcomes of interpolation, and its formation is in turn determined through the neighborhood selection. The square neighborhood used in NEDI is not the best way to maximize the local information for covariance estimation, and the use of fixed neighborhood size for all cases does help, either. Thirdly, the validity of the “geometric duality” is ultimately not space-invariant, but depends on the scale of the image feature. This, in conjunction with the second issue, could result in artifacts when faced with thin edge patterns [5, 6].

2.2. Improved Adaptive Covariance-based interpolation

As for our discussion regarding the shortcomings of NEDI, we propose two improvements to the original NEDI algorithm:

1. Sinc interpolation for non-edge pixels: Windowed sinc interpolation has been widely used for image processing due to its relatively low computation load and higher accuracy, especially so for one of its variants,
Lanczos [9]. Therefore we chose to adopt Lanczos interpolation for non-edge pixels in our Improved Adaptive Covariance-based interpolation (IACI).

2. Adaptive neighborhood window for optimal covariance estimation: We first designed the neighborhood window to be circular instead of square in order to maximize the local information that could be contained in the window, and also to reduce the preference of vertical, horizontal and diagonal edge directions caused by the square-shaped window. Furthermore, the window size is no longer fixed, but able to increase from a minimum size to a maximum size, while keeping track of the approximate mean square error (AMSE) in the covariance estimation process. The AMSE can be calculated as [10]:

\[ AMSE = \| \tilde{y} - C \times \alpha \|^2 \]  \hspace{1cm} (5)

The convergence of AMSE toward a preset threshold is the stop criterion for window size increase.

2.3. Adaptive Riemannian Tensor Interpolation

In this section we will briefly summarize the necessary mathematical tools so as to use them in our Adaptive Riemannian Tensor interpolation (ARTI). A symmetric positive definite 2-tensor is a \( n \times n \) matrix that belongs to the space of all symmetric positive-definite matrices denoted by \( P(n) \). For any two 3-dimensional tensors \( p_1, p_2 \in P(3), (p_1 \neq p_2) \), there exists one geodesic linking from \( p_1 \) to \( p_2 \), parameterized by a real number \( t(\in [0, 1]) \):

\[ \gamma(t) = p_1^{\frac{1}{2}} \exp(tp_1^{-\frac{1}{2}} T p_1^{-\frac{1}{2}}) p_1^{\frac{1}{2}} \]  \hspace{1cm} (6)

where \( T \) is the tangent vector between \( p_1 \) and \( p_2 \) [7]. The geodesic can be viewed as a 1-dimensional linear interpolation between two tensors \( p_1 \) and \( p_2 \). Yet, moving on to 2-dimensional interpolation, a Riemannian counterpart of bi-linear interpolation is not valid due to the tensor incommutability. However, one can calculate the Riemannian intrinsic weighted sum (RIWS) as the interpolation inside the neighborhood weighted by certain coefficients:

\[ p^w = \arg\min_{p^w} \left( \sum_{k=1}^{N} \alpha_k \text{dist}^2(p^w, p_k) / \sum_{k=1}^{N} \alpha_k \right) \]  \hspace{1cm} (7)

where \( p^w \) is the weighted sum, \( \alpha_k \) is the weight associated with the \( k \)th neighbor, \( p_k \), and \( \text{dist}(p_1, p_j) \) is the Riemannian distance between two tensors \( p_1 \) and \( p_j \) [7]. This “summation” is actually an iterative optimization that minimizes the weighted sum of the square distance between the sum tensor and its neighbors.

Now that we have all the tools we need, the assembling is quite natural: For each tensor \( p^j \) to be interpolated, we first used a tensor scalar map that is able to reflect the tensor edge information (e.g., FA map) in IACI to calculate the weights \( \{ \alpha_k \} \) that reflects edge orientation, after which we supply the weights and the selected neighbors \( \{ p_k \} \) into Eq. 7 to obtain the final \( p^w \).

3. RESULTS AND DISCUSSION

In this section we present our results for both IACI (compared against bicubic method and NEDI) and ARTI (compared against bicubic Euclidean interpolation). For the former, phantom scalar images were used, while for the latter, real medical diffusion tensor images were shown. For numerical evaluation, we simply used PSNR for the phantom scalar image. For tensor images, we used the Overlapping of Eigenvector/Eigenvector pairs (OVL) [11] as well as the UIQI (Universal Image Quality Index [12]) for the FA map to evaluate how well the interpolation is in agreement with the ground truth. For scalar images, we only show phantom results here. For the tensor images, we show here the results from real rat brain data, which has a dimension of \( 265 \times 256 \times 21 \), with 6 gradient directions. As shown in Fig. 2, both edge-directed methods have been able to better capture the edges where the bicubic method renders as jaggies. However, due to the limitations of NEDI discussed in section 2.1, the NEDI result features artifacts (“bad spots”) around the thin edges. The IACI result not only corrects for that, and its PSNR is also better than the others’. The IACI-based ARTI is able to better capture the tensor shape and orientation information, as shown in Table 1. A visual demonstration in Fig. 3 with respect to the angular difference of first eigenvector and FA difference between the ground truth and the interpolations also confirms that incorporating shape information improves the interpolation accuracy.

<table>
<thead>
<tr>
<th></th>
<th>Real DTI</th>
<th>Euc. BC</th>
<th>ARTI</th>
</tr>
</thead>
<tbody>
<tr>
<td>OVL</td>
<td>0.831</td>
<td>0.873</td>
<td></td>
</tr>
<tr>
<td>UIQI for FA</td>
<td>0.9772</td>
<td>0.9826</td>
<td></td>
</tr>
</tbody>
</table>

Table 1. Comparative table for real data results of Euclidean Bicubic and ARTI.

4. CONCLUSION

In this work, we have improved upon NEDI based on our appreciation for its shortcomings, and, as an application, extended the edge-directed concept onto the DTI interpolation, combined with the benefits of a Riemannian framework. The scalar results improve upon the NEDI algorithm, while the tensor results confirm that DTI interpolation, with tensor geometry preserved, indeed benefits from the inclusion of shape information. This edge-directed concept for DTI interpolation can be further investigated with other scalar map as ref-
Fig. 2. The comparison on a phantom image of dense thin edge patterns. Figs. 2(a)-2(d) are the original high-resolution phantom, bicubic interpolation, NEDI result and IACI result respectively. The numbers in the subtitles of Fig. 2(b)-2(d) are their respective PSNR.

Fig. 3. The comparison on the real image. Figs. 3(a)-3(b) are the first eigenvector angular difference (scaled between 0° (black) and 90° (white)) of the ground truth and Euclidean BC interpolation, and that of the ARTI. Figs. 3(c)-3(d) are the FA difference of the ground truth and Euclidean BC interpolation, and that of the ARTI (Both equally rescaled for better visibility).

5. ACKNOWLEDGEMENT

The work presented in this paper is supported by the SBO Quantiviam project, and the authors would also wish to thank Dirk Poot and Yifan Zhang for their helpful discussions.

6. REFERENCES


