

3D DT-MRI Data Visualisation using the Dual Tree Complex Wavelet Transform

Nick Kingsbury and Argyris Zymnis

Signal Processing Group, Cambridge University Engineering Department
Trumpington Street, Cambridge, CB2 1PZ, United Kingdom

Email: [ngk,az224]@eng.cam.ac.uk

1 Introduction

Diffusion Tensor Magnetic Resonance Imaging (DT-MRI) is a specific modality of MRI which is based on the calculation of the diffusion of water along different directions within the brain. It can be used efficiently to map white matter tracts within the human brain. In previous work [1], we have shown how the Dual Tree Complex Wavelet Transform [2], [3] can be used to visualise DT-MRI data in 2 dimensions. Our method produces images which contain more detail than images developed using standard techniques [4], [5] and are also more easy to physically interpret. In this paper we will present an extension of this method to 3 dimensions.

2 Method and Results

2.1 Preliminary Tensor Analysis

A DT-MRI dataset consists of a 3D array of voxels, for each of which a 3x3 diffusion tensor is calculated. By diagonalising each of these diffusion tensors we can obtain their sorted eigenvalues and corresponding eigenvectors. A computationally efficient way of calculating the eigenvectors and eigenvalues of a DT-MRI dataset is described in [6]. The eigenvectors show the main directions of diffusion and the relative size of the eigenvalues is a measure of preferential diffusion in a given direction [4], [5].

We decided to split diffusion into two components: a scalar isotropic component (X_{is}) and a vectorial anisotropic component ($\overrightarrow{X_{an}}$). The value of X_{is} is equal to the average of the two smaller eigenvalues and the magnitude of $\overrightarrow{X_{an}}$ is equal to the difference of the largest eigenvalue from X_{is} . The direction of $\overrightarrow{X_{an}}$ is along the eigenvector with largest eigenvalue.

2.2 2D Visualisation

In the method described in [1] that we developed, we are using the 2D Dual Tree Complex Wavelet Transform to visualise the data. The 2D DT-CWT is a multilevel transform that has 6 directional subbands per level. Each of these subbands picks out details of an image in different directions (i.e. at $\pm 15^\circ, 45^\circ, 75^\circ$) [3].

For each slice in a DT-MRI dataset, we can calculate the component of the anisotropic vector field along each CWT subband direction. We then feed these components into their corresponding subbands (using random values for the complex phase) and we also feed the isotropic component into the LoLo subband. By then applying the inverse CWT, we will get an image which will show isotropic diffusion as grey-scale pixel intensity and anisotropic diffusion as “stripiness”. We then apply the forward and inverse CWT iteratively for about five iterations, in order to achieve better phase alignment of the “stripes”. In the final iteration we apply 2 levels of inverse CWT in order to increase the resolution of the image by 2 in each dimension. This makes details more readily discernible.

We also need to include the anisotropic vertical component in the image. We do this in two ways. We firstly superimpose a red-scale image on our grey-scale image, where red pixel intensity represents the vertical component. Secondly, we superimpose an image with stripes which are perpendicular to the anisotropic component in order to synthesise an “end-on” view of stripes that are almost normal to the image plane. The amount of perpendicular stripes is equal to the amount of parallel ones if anisotropy is completely vertical and it is zero if anisotropy is completely horizontal.

Figure 1 shows the result of our 2D method applied to the 4th slice of a 128x128x8 DT-MRI dataset of a healthy person. It can be seen that our method can provide useful detail even in low anisotropy regions, such as close to the cortical surface (skull boundary).

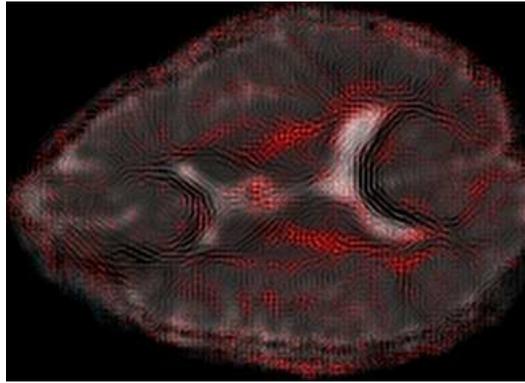


Figure 1: Representation of the fourth slice of the dataset in 2 dimensions

2.3 Interpolating between slices

Our initial dataset has 8 slices each with 128x128 voxels. The spacing between slices is 10mm and the intervoxel spacing is 1mm. Hence we decided to increase the number of slices from 8 to 64 and a number of different interpolation strategies was considered.

The simplest strategy is averaging. This consists of averaging the eigenvalues of the top and bottom slice to compute the eigenvalues of the interpolated slice. The principal eigenvectors of the interpolated slice are then computed by just adding the principal eigenvectors of the top and bottom slices and normalising. The major disadvantage of this method is the fact that averaging causes blurring of the edges, thus reducing image quality.

The second strategy that was considered made use of a least squared error criterion. This consists of computing an interpolated voxel by looking at its neighbouring voxels in the top and bottom slices. For each interpolated voxel, the squared error (difference) of each eigenquantity is computed for pairs of voxels on the top and bottom slices, in a small square neighbourhood near the interpolated voxel. The pair of voxels that gives the minimum squared error is then used to compute the interpolated voxel's eigenquantities. This method, however seems to give worst results than averaging and was therefore not used. This was probably due to the fact that original slice spacing is much larger than intervoxel spacing. This method probably works better for smaller slice spacings.

The final method that was considered takes advantage of the anisotropic components of the 3D tensors in the dataset. This requires finding a pair of voxels in adjacent slices, whose principal eigenvector points to the same voxel in the interpolated slice. Having found this pair of voxels we then compute its eigenvalues and eigenvectors as some form of weighted mean of the eigenquantities of the top and bottom voxels. If no two vectors line up properly for a given interpolated voxel, we can just perform simple averaging. A threshold, defined by the dot product of each vector with the vector connecting the interpolated voxel to the original voxel, is used to decide if the vector criterion or simple averaging will be used. This technique only seems to work if this threshold is very large, i.e. the only vectors used are those that almost perfectly line up. Otherwise, it tends to flare up the areas with a large vertical component. It is also not efficient in calculating horizontal vectors.

We decided to initially use the simple averaging technique, despite its drawbacks. This was because the least squared error technique does not work in this case and the vector technique is still under development. However, an ideal interpolating technique should make use of the third vectorial method, as well as some sort of method of finding voxels whose eigenvectors are parallel.

2.4 3D Visualisation

By applying the previous method to a given DT-MRI dataset, we obtain a 3D set which contains useful information only parallel to the xy plane. The cross sections parallel to the xz or yz planes do not give us good images. The reason for this is the fact that our method is separately applied to each slice. In order to overcome this we must develop a transformation which is applied to the whole dataset simultaneously. Hence for 3D visualisation we make use of the 3D DT-CWT.

The 3D DT-CWT has 28 directional subbands per level, oriented approximately uniformly over the surface of a hemisphere (7 bands per quadrant). Each of these subbands has a planar-like impulse response.

For each anisotropic vector in the dataset, we want to generate a filament-like structure (similar in cross-section to the criss-cross pattern used previously for the vertical anisotropic component). To do this we excite two subbands, whose planes of impulse response both contain the given anisotropic vector and are approximately perpendicular to each other. One strategy is to compute the dot product of each anisotropic vector with each directional vector for the 28 subbands and then select the subband which gives us the smallest dot product value. Then we use a lookup table to find the subbands which are perpendicular to the one selected and out of those select the one which has the smallest

dot product with the given anisotropic vector. These two subbands are then excited with an amount proportional to the anisotropic magnitude.

As before, the isotropic component is fed into the LoLo subband and we iteratively apply the forward and inverse CWT to achieve better phase alignment of the filaments.

We found however, that the output images of this algorithm are not satisfactory. This is due to two main reasons. The first one is the fact that we are using all available highpass subbands, including the HiHiHi bands, which increase the amount of high frequency information within the image, thus causing distortion. The second one is the fact that the problem of finding two planes perpendicular to each other which contain a given vector does not have a single solution, as the normals of the two planes can rotate about the axis defined by the given vector. Thus small changes in the anisotropic direction can cause large changes in the output image, which is undesirable.

We thus need a way of interpolating between subbands which is unambiguous and which produces images that change smoothly with direction.

2.5 3D DT-CWT Subband Interpolation Strategy

In order to minimise ambiguities, the number of subbands used must be reduced. Hence it was decided to make use only of the LHH, HLH, HHL subbands (just 3 per quadrant of a hemisphere, making 12 in total). The normals of the impulse responses of these subbands are the vertices of the structure shown in figure 2. This structure is in essence a cube, with a tetrahedral volume removed from each of its corners and having six octagonal sides.

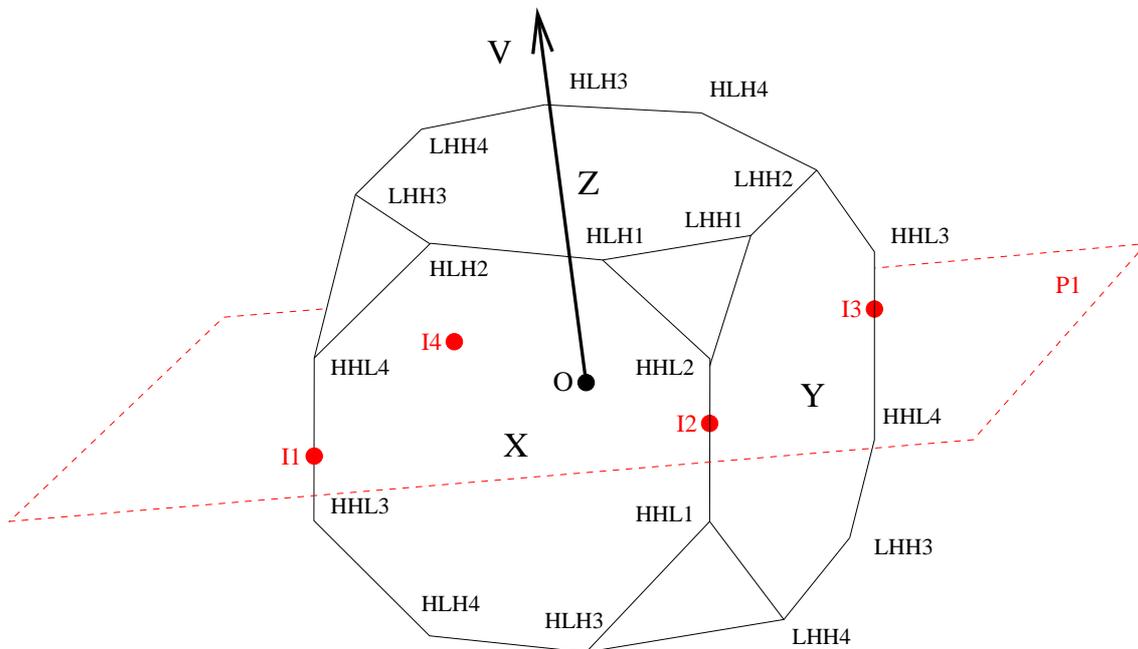


Figure 2: Structure depicting used subband normals

Suppose we want to visualise an anisotropic vector that has a direction parallel to vector \mathbf{V} in figure 2. In order to do that, we draw the plane whose normal is \mathbf{V} and which passes through the origin O . In the above case this plane is $P1$. The points of intersection of this plane are then computed ($I1$, $I2$, $I3$ and $I4$ in figure 2). Note that due to the spherical symmetry of the 3D DT-CWT subband structure, points $I1$ and $I3$ are identical and so are $I2$ and $I4$. These intersection points show which subbands have to be used to visualised a specific anisotropic vector. For this particular example, we can use subbands $HHL1$, $HHL2$, $HHL3$ and $HHL4$ to synthesize \mathbf{V} .

The number of edge intersections indicates what proportion of the total anisotropy must be allocated to each pair of subbands in each edge. In this case there are essentially two edge intersections (due to symmetry), thus each edge gets half the total anisotropy. This edge components is then divided between the two subbands using as a weighting factor the exact position of the intersection point. Hence for \mathbf{V} $HHL2$ will get a higher weighting than $HHL1$ and similarly $HHL3$ will get a higher weighting than $HHL4$. If for example \mathbf{V} was parallel to the z-axis (i.e. vertical), the four subbands would each get a weighting of a quarter.

The method described above works well for vectors which lie close to the x, y or z axes. However, vectors which are not close to a particular axis are not very well represented. This is especially true for vectors which define a line that cuts through any of the four triangular sides of the structure (i.e. which have almost equal amounts of x, y and z components). Some steps can be taken to avoid this ambiguity, for example we can ignore edges that are almost parallel to the plane defined by a specific vector. This technique is currently under development.

We then apply the above interpolation method to each anisotropic vector within the DT-MRI dataset, weighing each subband magnitude by the specific anisotropic vector's magnitude. We can then apply the iterative technique described for 2D to get an output image of the DT-MRI dataset.

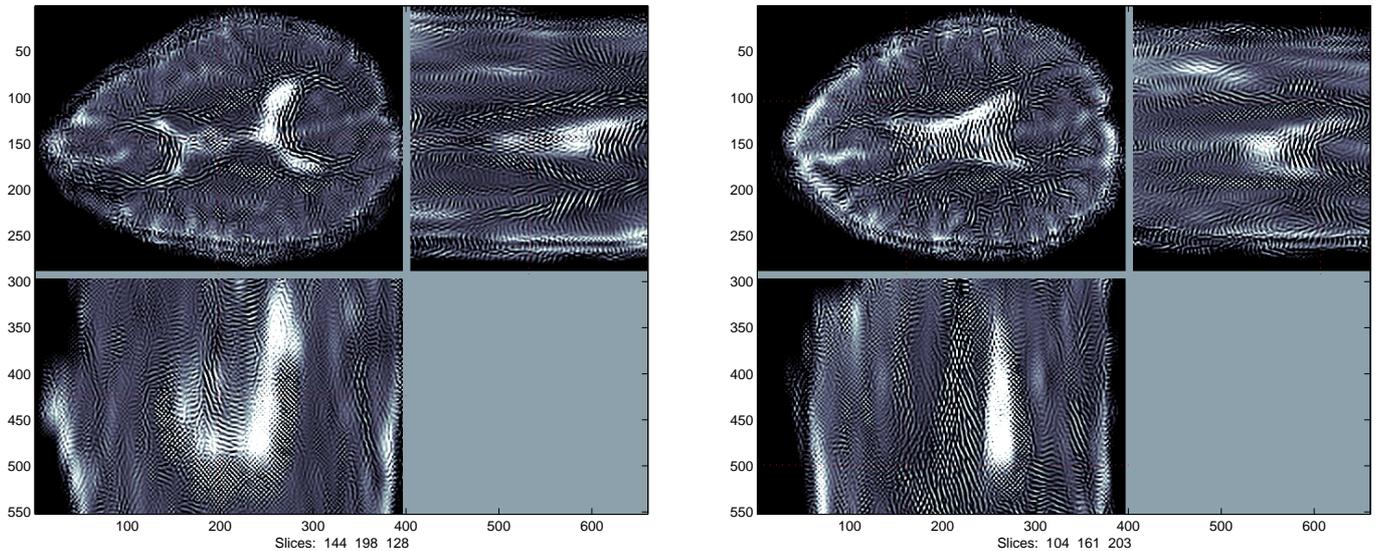


Figure 3: Output images for 3D technique

Figure 3 shows two sets of slices viewed along the x (top right), y (bottom left) and z (top left) axes for a $72 \times 99 \times 64$ cuboid within our original dataset. The algorithm interpolates the dataset up to a cuboid size of $288 \times 396 \times 256$. Note how the top left output image for the set of slices on the left is very similar to figure 1. This shows that our 3D technique is a natural evolution of the 2D technique.

By looking at the images of figure 3, it is evident that our method produces criss-cross patterns for groups of vectors viewed end on. This can be clearly seen in the bottom left subimage of the left side set of slices. It is also evident that if these same vectors are viewed along a direction perpendicular to their direction, they create “stripy” patterns. These two properties suggest that our method creates filament-like structures for anisotropy vectors, as desired. Even though stripe continuity is not always guaranteed, image quality is adequate for viewing purposes.

3 Conclusions

We have presented an iterative algorithm which can be used to visualise Diffusion Tensor Magnetic Resonance Imaging data in three dimensions. The distinct advantage of our method compared to standard techniques is that it contains both the isotropic and anisotropic components of diffusion on the same image. Using this method a high level of detail can be achieved and the images are also easy to physically interpret.

References

- [1] N. G. Kingsbury, A. Zymnis, and A. Pena, “DT-MRI data visualisation using the dual tree complex wavelet transform,” in *Proc. (IEEE) International Symposium of Biomedical Imaging (ISBI' 04)*, Arlington, USA, Apr. 2004, p. currently under submission.
- [2] N. G. Kingsbury, “Image processing with complex wavelets,” in *Phil. Transactions of London Royal Society*, 1999.
- [3] N. G. Kingsbury, “Complex wavelets for shift invariant analysis and filtering of signals,” *Journal of Applied and Computational Harmonic Analysis*, vol. 10, 2001.
- [4] Y. Masutani, S. Aoki, O. Abe, N. Hayashi, and K. Otomo, “MR diffusion tensor imaging: recent advance and new techniques for diffusion tensor visualisation,” *European Journal of Radiology*, vol. 46, pp. 53–66, 2003.
- [5] C. F. Westin, S. Maier, H. Mamata, A. Nabavi, F. A. Jolesz, and R. Kikinis, “Processing and visualisation for diffusion tensor MRI,” *Medical Image Analysis*, vol. 6, pp. 93–108, 2002.
- [6] K. M. Hasan, P. J. Basser, D. L. Parker, and A. L. Alexander, “Analytical computation of the eigenvalues and the eigenvectors in DT-MRI,” *Journal of Magnetic Resonance*, vol. 152, pp. 41–47, 2001.