

## How Fast Can PAS Go?

K. E. Sakaie<sup>1</sup>

<sup>1</sup>Radiology, The Cleveland Clinic, Cleveland, OH, United States

**Introduction:** Diffusion tensor imaging can be used to map the orientation of white matter fibers in the brain, but is limited in its ability to resolve the orientation of crossing fibers. High angular resolution diffusion imaging [1], combined with advanced postprocessing methods, can resolve crossing fibers. Of the large number of postprocessing methods, Persistent Angular Structure (PAS) has desirable properties in terms of angular resolution, accuracy, and robustness against noise [2]. Unfortunately, the publicly available software (Camino) [3] is extraordinarily slow-- requiring weeks to months to analyze a typical brain dataset on a single processor. In this contribution, we show that coding the PAS calculation using widely available nonlinear optimization code can speed up the computation considerably, making this powerful method an option for more widespread use.

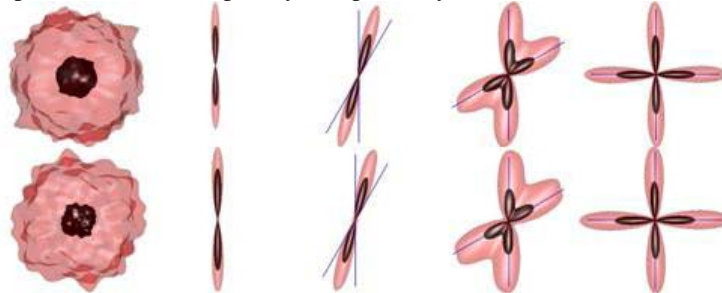
**Theory:** The PAS is a function whose peaks reflect the orientation of the white matter fibers in a manner akin to the orientation functions derived by other methods, such as spherical deconvolution [4], Q-Ball [5], and the Diffusion Orientation Transform [6]. Jansons and Alexander derived a maximum-entropy form of the PAS [7]:

$$p(x) = \exp\left(\lambda_0 + \sum_{i=1}^N \lambda_i \cos(Rq \cdot x)\right)$$

where  $\mathbf{R}$  is an empirically determined parameter, which controls the tradeoff between angular resolution and noise,  $\lambda_i$  are Lagrange multipliers, which reflect constraints from normalization and from the measurements, and the sum is over the diffusion gradient directions. The goal is to find the set of Lagrange multipliers that optimizes the agreement with the acquired signal [7]. Camino [3] is implemented in Java using a Levenberg-Marquardt (LM) [8, 9] optimization algorithm from Numerical Recipes [10]. The Numerical Recipes version of LM is notoriously poor in terms of efficiency and reliability [11]. We compare the performance of the PAS calculation using Camino and using a MATLAB [Mathworks, Natick, MA] implementation of an optimized LM algorithm [12] on simulated data.

**Methods:** Data mimicking isotropic, single, and crossing fibers were generated using the diffusion tensor model [13], a Rician noise distribution (SNR=20) with 1000 noise realizations for each configuration, 71 diffusion-weighting directions chosen by an electrostatic repulsion algorithm [14], and 8  $b=0$  acquisitions. Eigenvalues of  $(1.4, 0.35, 0.35) \times 10^{-3} \text{mm}^2/\text{sec}$ , for individual fibers in the single and crossing-fiber cases, and eigenvalues of  $(0.7, 0.7, 0.7) \times 10^{-3} \text{mm}^2/\text{sec}$  were used for the isotropic case. The crossing fibers had partial volume fractions of 0.5 each with crossing angles of  $30^\circ$ ,  $60^\circ$ , and  $90^\circ$ . Camino was implemented by parallelization with the Sun Grid Engine on a Linux cluster with 10 dual-core 2.33GHz Intel Xeon 5140 cpus. MATLAB calculations were performed on a single cpu. Computation times are quoted in cpu-hours for the sake of comparison.

**Results:** The figure shows mean and mean+standard deviation images of the PAS calculated using PAS (top) and MATLAB (bottom) for (left to right) the isotropic, single-fiber, and crossing fiber cases of  $30^\circ$ ,  $60^\circ$ , and  $90^\circ$ . Blue lines indicate the orientation of the individual fibers for the crossing fiber cases. Computation time was approximately 22 cpu-hours for Camino and 2.5 cpu-hours for MATLAB, which translates to 3 cpu-months and 10 cpu-days, respectively, for a 100,000 voxel whole-brain dataset.



**Discussion:** As can be seen from the above figure, the MATLAB implementation of the LM algorithm yields PAS functions which are nearly the same as those generated by Camino. Additional gains in speed can be expected by implementing a compiled version of the LM algorithm in C [15]. These preliminary results indicate that orders of magnitude of calculation speed may be gained by further optimization of the code behind the PAS calculation.

**References:** [1] Tuch, D.S., et al., *Magn Reson Med*, 2002. **48**(4):577-82. [2] Alexander, D.C., *Ann N Y Acad Sci*, 2005. **1064**:113-33. [3] Cook, P.A., et al., *14th ISMRM*. 2006. Seattle: 2759. [4] Tournier, J.D., et al., *Neuroimage*, 2004. **23**(3):1176-85. [5] Tuch, D.S., *Magn Reson Med*, 2004. **52**(6):1358-72. [6] Ozarslan, E., et al., *Neuroimage*, 2006. **31**(3):1086-103. [7] Jansons, K.M. and D.C. Alexander, *Inverse Problems*, 2003. **19**:1031-1046. [8] Levenberg, K., *Quart Appl Math*, 1944. **2**:164-168. [9] Marquardt, D., *SIAM J Appl Math*, 1963. **11**:431-441. [10] Press, W.H., et al., *Numerical Recipes in C*. [11] Hanson, R.J., *SIAM News*, 1990. **28**(3):18. [12] More, J.J., in *Numerical Analysis 630*, 1977, Springer Verlag: 105-116. [13] Bassler, P.J., et al, *Biophys J*, 1994. **66**(1):259-67. [14] Jones, D.K., et al., *Magn Reson Med*, 1999. **42**(3):515-25. [15] Galassi, M., et al., *GNU Scientific Library Reference Manual*. 2 ed. 2006, Bristol: Network Theory Ltd.