iPGA: Incremental Principal Geodesic Analysis with Applications to Movement Disorder Classification*

Hesamoddin Salehian[1], David Vaillancourt[2], and Baba C. Vemuri[1,**]

[1] Department of CISE, University of Florida, Gainesville, Florida, USA
[2] Department of Applied Physiology and Kinesiology,
University of Florida, Gainesville, Florida, USA
{salehian,vemuri}@cise.ufl.edu, vcourt@ufl.edu

Abstract. The nonlinear version of the well known PCA called the Principal Geodesic Analysis (PGA) was introduced in the past decade for statistical analysis of shapes as well as diffusion tensors. PGA of diffusion tensor fields or any other manifold-valued fields can be a computationally demanding task due to the dimensionality of the problem and thus establishing motivation for an incremental PGA (*iPGA*) algorithm. In this paper, we present a novel *iPGA* algorithm that incrementally updates the current Karcher mean and the principal sub-manifolds with any newly introduced data into the pool without having to recompute the PGA from scratch. We demonstrate substantial computational and memory savings of *iPGA* over the batch mode PGA for diffusion tensor fields via synthetic and real data examples. Further, we use the *iPGA* derived representation in an NN classifier to automatically discriminate between controls, Parkinson's Disease and Essential Tremor patients, given their HARDI brain scans.

1 Introduction

The nonlinear generalization of PCA called Principal Geodesic Analysis (PGA) pioneered by Fletcher et al. [4] was applied to achieve statistical analysis of manifold-valued data namely, neuro-anatomical structures which are represented as points on shape manifolds. PGA captures variability in the data by using the concept of principal geodesic subspaces which in this case are sub-manifolds of the Riemannian manifold on which the given data lie. In order to achieve this goal, it is required to know the Riemannian structure of the manifold, specifically, the geodesic distance, the Riemannian $\log$ and $\exp$ maps and the Karcher mean (see section 2 for definitions). PGA relies on use of the linear vector space structure of the tangent space at the Karcher mean by projecting all of the data points to this tangent space and then performing standard PCA in this tangent space followed by projection of the principal vectors back to the manifold using the Riemannian $\exp$ map yielding the principal geodesic subspaces. The representation of each manifold-valued data point in the principal geodesic subspace is achieved by finding the closest (in the sense of geodesic distance) point in the subspace to the given data point. A generalization of the PGA reported in [8,3] to symmetric positive definite (SPD) diffusion tensor fields was presented in [11]. Authors in [11] demonstrated that

* This research was funded in part by the NIH grant NS066340 to BCV.
** Corresponding author.

P. Golland et al. (Eds.): MICCAI 2014, Part II, LNCS 8674, pp. 765–772, 2014.

the Karcher mean of several given (registered) tensor fields computed using a voxel-wise Karcher mean over the field is equivalent to the Karcher mean computed using the Karcher mean in a product space representation of the tensor fields. However, for higher order statistics, such as variance, such an equivalence does not hold. This observation however holds for any manifold-valued fields, not just for the diffusion tensor fields.

PGA has been applied to many problems in the past decade, including statistical shape analysis [4] and tensor field classification [11] in medical image analysis. When dealing with large amounts of manifold-valued fields e.g., diffusion tensor fields, deformation tensor fields, ODF fields etc., performing PGA can be computationally quite expensive. That said, if we are provided the data incrementally, one tensor field at a time, rather than performing batch mode PGA it would be computationally more efficient to simply update the already computed PGA as new data are made available. To this end, we propose a novel incremental PGA or *iPGA* algorithm in which we incrementally update the Karcher mean and the principal sub-manifolds rather than performing PGA in a batch mode. This will lead to significant savings in computation time as well as space/memory.

In the past few decades, the problem of incrementally updating the PCA has been well studied in literature e.g., [12]. However, these methods require the data samples to live in a Euclidean space, and hence are not directly applicable to the PGA problem. On the other hand, Cheng et al. [1] and Ho et al. [6] have reported incremental algorithms for computing the Karcher expectation of a given set of SPD matrices. Our *iPGA* algorithm is a novel combination of the incremental PCA idea and the incremental Karcher expectation algorithm in [1,6]. This is derived for SPD tensor fields. To this end, we reformulate the SPD tensor-field PGA algorithm introduced in [11] to achieve *iPGA*. Then, we apply our *iPGA* to a group of SPD tensor fields derived from high angular resolution diffusion magnetic resonance images (HARDI), for classification of patients with movement disorders. We present synthetic experiments depicting the effectiveness and accuracy of *iPGA*, compared to the batch-mode PGA. Further, in the real data experiments, given 67 human brain HARDI data, our *iPGA* based NN classifier aims to distinguish between controls, Parkinson's Disease (PD) and Essential Tremor (ET) patients. Our results demonstrate the effectiveness of *iPGA*, compared to the batch mode scheme.

The rest of the paper is organized as follows. Section 2 contains background material on differential geometry of the space of SPD tensor fields. Next, in section 3 the proposed *iPGA* technique is described in detail. Sections 4 and 5 contain synthetic and real data experiments respectively, comparing PGA and *iPGA* with respect to computation time and accuracy. We draw conclusions in Section 6.

2 Riemannian Geometry of SPD Tensor Fields

We now briefly introduce the basic relevant concepts of Riemannian geometry of the space of SPD tensor fields denoted by $P(n)^m$ following the notation from [11]. For details on the Riemannian geometry of $P(n)$ we refer the reader to [3]. $P(n)$ is the space of $n \times n$ symmetric positive definite (SPD) matrices, which is a Riemannian manifold with $GL(n)$, the general linear group as the symmetry group. This can be easily generalized to$P(n)^m$, the product space of $P(n)$ using the product Riemannian structure. In

particular, expressions for the Riemannian geodesic distance, log and exponential maps can be easily derived. Specifically, the group $GL(n)^m$ acts transitively on $P(n)^m$ with the group action specified by $\phi_{\mathbf{G}}(\mathbf{X}) = (G_1 X_1 G_1^T, \ldots, G_m X_m G_m^T)$, where each $G_i \in GL(n)$ is a $n \times n$ invertible matrix and X_i is an $n \times n$ positive-definite matrix. The tangent space of $P(n)^m$ at any point can be identified with $Sym(n)^m$ because the tangent space of a product manifold is the product of tangent spaces. Let $\mathbf{Y}, \mathbf{Z} \in T_{\mathbf{M}} P(n)^m$ be two tangent vectors at $\mathbf{M} \in P(n)^m$. The inner product between two vectors using the product Riemannian metric is given by, $\langle \mathbf{Y}, \mathbf{Z} \rangle_{\mathbf{M}} = \sum_{i=1}^{m} \mathrm{tr}(Y_i M_i^{-1} Z_i M_i^{-1})$. The Riemannian exponential map at $\mathbf{M}$ maps $\mathbf{Y}$ the tangent vector, to a point in $P(n)^m$ and is given by, $\mathrm{Exp}_{\mathbf{M}}(\mathbf{Y}) = \big(G_1 \exp(G_1^{-1} Y_1 G_1^{-T}) G_1^T, \ldots, G_m \exp(G_m^{-1} Y_m G_m^{-T}) G_m^T\big)$, where $G_i \in GL(n)$ such that $\mathbf{M} = \big(G_1 G_1^T, \ldots, G_m G_m^T\big)$.

Given $\mathbf{X} \in P(n)^m$, and the log map at $\mathbf{M}$ is given by,

$\mathrm{Log}_{\mathbf{M}}(\mathbf{X}) = \big(G_1 \log(G_1^{-1} X_1 G_1^{-T}) G_1^T, \ldots, G_m \log(G_m^{-1} X_m G_m^{-T}) G_m^T\big)$.

Using this definition of the log map in $P(n)^m$, the geodesic distance between $\mathbf{M}$ and $\mathbf{X}$ is computed as $d(\mathbf{M}, \mathbf{X}) = \|\mathrm{Log}_{\mathbf{M}}(\mathbf{X})\| = \sqrt{\sum_{i=1}^{m} \mathrm{tr}\left(\log^2(G_i^{-1} X_i G_i^{-T})\right)}$.

Using the expression for the geodesic distance given above, we can define the (intrinsic) mean of N tensor fields as that tensor field which minimizes the following sum of squared geodesic distances expression: $\mathbf{M} = \arg\min_{M \in P(n)^m} \frac{1}{N} \sum_{i=1}^{N} d(M, \mathbf{X}_i)^2$. The sum of squares on the RHS above can be re-written as a sum over all points in Ω. This means that the value of $\mathbf{M}(p)$ of $\mathbf{M}$ at a point $p \in \Omega$ is the usual Karcher mean in $P(n)$ of $\mathbf{X}_1(p), \cdots, \mathbf{X}_N(p)$. In particular, since the Karcher mean is unique on $P(n)$ [3], this shows that $\mathbf{M}$ will be unique as well, and it can be computed using an iterative algorithm similar to the one in [3]. After obtaining the intrinsic mean $\mathbf{M}$ of the input tensor fields $\mathbf{X}_1, \ldots, \mathbf{X}_N$, we compute the modes of variation using the PGA algorithm for tensor fields described in [11].

3 *iPGA*: Incremental Principal Geodesic Analysis

In order to develop the incremental Principal Geodesic Analysis on the space of SPD tensor fields, we will break down the problem into two key components involving the development of, (i) an incremental Karcher mean update technique applicable to tensor fields and (ii) an incremental updating method for principal submanifolds. We will address these two sub-problems in the following paragraphs.

3.1 Incremental Karcher Expectation Estimator

As described earlier, the Karcher mean of the SPD tensor fields is defined as the minimizer of the sum of squared geodesic distances. Unfortunately, this minimization problem does not have a closed form solution for a population of size greater than two. In [6], authors presented a recursive Karcher expectation estimator, *RKEE*, for SPD matrices (not SPD tensor fields). Given the estimated Karcher mean of the first k SPD tensors, denoted by M_k,and the new sample X_{k+1}, *RKEE* locates the new mean, M_{k+1}, on the geodesic curve between M_k and X_{k+1} using the Euclidean weight. More formally, $M_{k+1} = Exp_{M_k}(tLog_{M_k}(X_{k+1}))$, where $t = \frac{1}{k+1}$.

We now generalize the above incremental Karcher mean formula to the case where the data samples are SPD tensor fields (not just SPD matrices), using exp and log maps defined earlier on the product manifold of SPD tensor fields. Let $\mathbf{M_k} = (M_{k,1}, ..., M_{k,m})$ denote the estimated Karcher mean of the first k samples, and $\mathbf{X}_{k+1} = (X_{k+1,1}, ..., X_{k+1,m})$ be the new given tensor field. Based on the *RKEE* algorithm and the product space representation chosen here, it is straightforward to generalize the *RKEE* to the product space of tensor fields $P(n)^m$. Thus, the new mean then is obtained by:

$$\mathbf{M_{k+1}} = (Exp_{M_{k,1}}(\frac{1}{k+1} Log_{M_{k,1}}(X_{k+1,1})), ..., Exp_{M_{k,m}}(\frac{1}{k+1} Log_{M_{k,m}}(X_{k+1,m}))) \tag{1}$$

3.2 *iPCA*: Incremental Principal Component Analysis

Principal component analysis of an input data matrix is tightly related to its Singular Value Decomposition (SVD) [5]. Let $A \in \mathbb{R}^{d,n}$ be the data matrix, where its n columns correspond to d dimensional observations. The SVD of A is given by USV^T, where $S \in \mathbb{R}^{d,n}$, and $U \in \mathbb{R}^{d,d}$ and $V \in \mathbb{R}^{n,n}$ are orthonormal matrices. Given that the diagonal elements of S are sorted in descending order, the first r principal components of data matrix A correspond to the first r columns of matrix U, denoted by $U_r \in \mathbb{R}^{d,r}$. Therefore, incremental update of the PCA of a set of observations, can be reduced to the incremental update of SVD of the corresponding data matrix.

Table 1. Incremental SVD Algorithm

1: **Input** U_r, S_r, V_r, and k new observations $X \in \mathbb{R}^{d,k}$
2: Compute QR decomp. of matrix $(I - U_r U_r^T)X = QR$
3: Compute the rank-r SVD of matrix $\begin{pmatrix} S_r & U_r^T X \\ 0 & R \end{pmatrix} = \hat{U}\hat{S}\hat{V}$
4: **Output** $(A, X) = ([U_r, Q]\hat{U})\hat{S}(\begin{pmatrix} V_r & 0 \\ o & I \end{pmatrix}\hat{V})^T$

In our implementation, we applied the incremental SVD algorithm in [12] which is summarized in Table 1. In this algorithm, $A_r \in \mathbb{R}^{d,n}$ is the best rank-r approximation of data matrix A for n given observations. A_r is defined by: $A_r = U_r S_r V_r^T$, where U_r is defined earlier, $S_r \in \mathbb{R}^{r,r}$ denotes the first r rows and r columns of S, and $V_r \in \mathbb{R}^{n,r}$ is the first r columns of V. Note that in the case of a single new observation, i.e., $X \in \mathbb{R}^d$, the QR decomposition outputs, $Q(1,j) = X/||X||$ (other rows of Q are mutually orthogonal), and $R(1,1) = ||X||$ (other elements of R are zero).

3.3 Proposed Algorithm

In this section we will develop the incremental version of the PGA algorithm in [11]. Very briefly, in [11], the PGA computation problem on the space of SPD tensor fields is approximated by applying PCA in the tangent plane anchored at the Karcher mean, in the following manner. First, the Karcher mean, $\mathbf{M}$, of the set of tensor fields is computed. Next, each tensor field is projected to the tangent space at the mean (i.e.,

$T_{\mathbf{M}}P(n)^m$), using $\log$ map, then transformed to the tangent space at the identity. This tangent space is a standard Euclidean space denoted by $T_{\mathbf{I}}P(n)^m$, where $\mathbf{I}$ is the tensor field consisting of m identity matrices. Therefore, the ordinary PCA algorithm is performed at $T_{\mathbf{I}}P(n)^m$, and the obtained principal components are transformed back to $T_{\mathbf{M}}P(n)^m$. *Note that this operation of transforming to the identity is crucial, since, the inner product defined for $P(n)^m$ corresponds to the inner product in the Euclidean space only at the identity* $\mathbf{I}$.

Equipped with the two algorithmic tools presented thus far (*RKEE* and *iPCA*), we are ready to reformulate this algorithm in an incremental form. In a similar fashion, each SPD tensor field is projected using the $\log$ map and transformed (by applying the group action) to $T_{\mathbf{I}}P(n)^m$. More formally, let $\mathbf{X_i}$ denote the i^{th} tensor field, and $\mathbf{M_k}$ be the Karcher mean of the k given samples. Define $\mathbf{Y_i} = Log_{\mathbf{M_k}}(\mathbf{X_i}) \in T_{\mathbf{M_k}}P(n)^m$. Each $\mathbf{Y_i}$ is then transformed to $T_{\mathbf{M}}P(n)^m$, to obtain $\mathbf{Z_i}$. Accordingly, the data matrix at $T_{\mathbf{I}}P(n)^m$, denoted by $\mathbb{A}_k$, can be constructed where its i^{th} column corresponds to $\mathbf{Z_i}$ in a vectorized form.

Table 2. Incremental PGA Algorithm

1: **Input** the SVD of $\mathbb{A}_k$ for k samples: U_k, S_k, V_k the new tensor field $\mathbf{X_{k+1}}$, and the old mean $\mathbf{M_k}$
2: Compute $\mathbf{M_{k+1}}$ from $\mathbf{X_{k+1}}$ and $\mathbf{M_k}$, using Eq. 1
3: $\mathbf{Y_{k+1}} = Log_{\mathbf{M_{k+1}}}(\mathbf{X_{k+1}})$
4: $\mathbf{Z_{k+1}} = \Phi_{\mathbf{G}^{-1}}(\mathbf{Y_{k+1}})$, defined in Eq. 2
5: Given $\mathbf{Z_{k+1}}$, update (U_k, S_k, V_k) to (U_{k+1}, S_{k+1}, V_{k+1}) using algorithm in Table 1.
6: Translate j^{th} principal component, $\mathbf{P_j}$, back to $T_{\mathbf{M_{k+1}}}P(n)^m$, via $\mathbf{Q_j} = \Phi_{\mathbf{G}}(\mathbf{P_j})$

In the our algorithm, we keep track of the best rank-r SVD decomposition of the data matrix, $\mathbb{A}_k$, at $T_{\mathbf{I}}P(n)^m$. Formally, $\mathbb{A}_k = U_k S_k V_k^T$. Note that, the columns of U_k correspond to the principal components in the tangent space. Let $\mathbf{X_{k+1}}$ and $\mathbf{M_k}$ denote the new SPD tensor field, and the Karcher mean over all previous k tensor fields, respectively. Then, to update the principal components we need to augment the data matrix with an appropriate vector which represents $\mathbf{X_{k+1}}$, in $T_{\mathbf{I}}P(n)^m$.

In order to find this vector, we first locate the new Karcher mean $\mathbf{M_{k+1}}$, using Eq. 1 , then project $\mathbf{X_{k+1}}$ to the tangent space at $\mathbf{M_{k+1}}$, i.e., $\mathbf{Y_{k+1}} = Log_{\mathbf{M_{k+1}}}(\mathbf{X_{k+1}})$. This tangent vector is moved to $T_{\mathbf{I}}P(n)^m$ using the group action on $P(n)^m$ as shown below, where, $\mathbf{G} = (G_1, ..., G_m)$, and $\mathbf{G}$ is such that $\forall i, M_{k+1,i} = G_i G_i^T$.

$$\mathbf{Z_{k+1}} = \Phi_{\mathbf{G}^{-1}}(\mathbf{Y_{k+1}}) = (G_1^{-1} Y_{k+1,1} G_1^{-T}, ..., G_m^{-1} Y_{k+1,m} G_m^{-T}) \tag{2}$$

Now, the best rank-r SVD decomposition of $\mathbb{A}_k$ and the vector Z_{k+1} are both in $T_{\mathbf{I}}P(n)^m$ which is the standard Euclidean space. Hence, we can readily apply the incremental SVD in section 3.2, and update the rank-r SVD to estimate U_{k+1}, V_{k+1} and S_{k+1}, accordingly. At the end, the new principal components which correspond to the columns of U_{k+1} are transformed back to $T_{\mathbf{M_{k+1}}}P(n)^m$, using the transformation $\Phi_{\mathbf{G}}$, where Φ and $\mathbf{G}$ are the same as in Eq. 2. This method is summarized in Table 2.

4 Synthetic Experiments

Data Description. We generated a group of 25, 16×16 SPD tensor fields, synthetically. The 3×3 SPD matrices in all tensor fields are ellipsoidal. There are two types of SPD matrices in each tensor field, whose principal eigenvectors differ by 90 degree. In generated tensor fields, the angles of principal eigenvectors of the first and the second matrices are uniformly chosen in $[0, \pi]$ and $[\frac{\pi}{2}, \frac{3\pi}{2}]$, respectively.

Computation Time/Space Consumption. Given a pool of tensor fields, they are incrementally input (in random order) to both *iPGA* and PGA algorithms and the CPU time and the memory consumed (on an Intel-7 $2.76GHz$ CPU with $8GB$ RAM) by each method to compute the principal components is recorded. We repeat this experiment 10 times on the data pool of 25 tensor fields and plot the average time/accuracy for each method. The plot in the middle of Fig. 1 demonstrates that CPU time consumption for *iPGA* is significantly less compared to that of PGA, especially for a large number of input data samples. Besides, the plot on the left in Fig. 1 depicts that iPGA requires roughly a constant space for any number of input tensor fields, while PGA's space consumption grows linearly, as expected.

Error Measurement. In order to measure the accuracy of each method, we computed the *residual sum* defined in [9] for estimated principal components. For N input tensor fields, the residual sum is defined by $\frac{1}{N}\sum_{j=1}^{N} d^2(\mathbf{X_j}, \hat{\pi}_{S_U}(\mathbf{X_j}))$, where d is the geodesic distance on $P(n)^m$, and $\hat{\pi}_S(\mathbf{X_j})$ is the estimated *projection* of $\mathbf{X_j}$ to the geodesic subspace spanned by the principal components, denoted by S_U. The projection, π_{S_U}, is estimated in the tangent space (see Eq.6 in [9] for details). The bar chart on the right in Fig. 1 depicts the error comparison between PGA and *iPGA* at each iteration. It can be seen that *iPGA*'s residual error is very close to PGA's. Thus, from an accuracy viewpoint, *iPGA* is on an equal footing with PGA but from a computational efficiency viewpoint, it is significantly better.

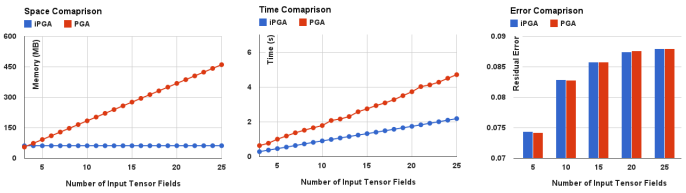

Fig. 1. Time & space consumption, and residual error comparison between iPGA and PGA

5 Real Data Experiments: Classification of PD vs. ET vs. Controls

In this section we present an application of *iPGA* to real data sets. Our real data consists of HARDI acquisitions from patients with Parkinson's disease (PD), essential tremor (ET) and controls. The goal here is to be able to automatically discriminate between

these groups using features derived from the HARDI data. Earlier work in this context in the field of movement disorders involved use of DTI based ROI analysis specifically using scalar valued measures such as fractional anisotropy [10]. They showed that DTI had high potential of being a non-invasive early trait biomarker. All our HARDI data were acquired using a 3T Phillips MR scanner with $TR = 7748ms, TE = 86ms$, $b-$values: $1000\frac{s}{mm^2}$, 64 gradient directions and voxel size = $2 \times 2 \times 2mm^3$.

For the features, we use the ensemble average propagator (EAP) at each voxel estimated using the technique in [7]. We extract the Cauchy deformation tensor field which is computed from a non-rigid registration of the given EAP fields to the control atlas EAP field (constructed using the approach in [2]). The Cauchy deformation tensor is defined as $\sqrt{JJ^t}$, where J is the Jacobian of the deformation at each voxel. The Cauchy deformation tensor is an SPD matrix of size $(3, 3)$ in this case. This gives us an SPD field as a derived feature corresponding to each given EAP field. We use the *iPGA* described earlier and use the geodesic distance-based NN to classify the probe data set. Note that the geodesic distance in this case is the distance between the probe data set and the geodesic submanifold representation of each class namely, PD, ET and Controls. The probe is assigned the label of that class with smallest geodesic distance. Classification is performed on 26 PD, 16 ET and 25 control subjects using the PGA of the Cauchy deformation tensor fields described above, where 10 subjects from PD and control, as well as 6 subjects from ET were picked as test group, and the rest of the subjects we used for training. Table 3 summarizes the accuracy for each method. For comparison, we also used the standard PCA method, as well as SVM with Radial Basis Function (RBF) kernel, which are applied to a vectorized version of the tensor fields. The size of the tensor fields was restricted to the ROIs instead of the whole image. Thus, the dimensionality was $600 * 6 = 3600$ and we used just the first two principal components in all competing methods to achieve the classification reported in the table. From the table, it is evident that iPGA and PGA provide very similar accuracies in all three classifications, while iPGA's computation time is significantly less compared to PGA's. Further, iPGA is considerably more accurate than PCA and SVM, because in the two later cases, the non-linearity of $P(n)^m$ is not taken into account.

6 Discussion and Conclusion

In this paper we introduced a novel *iPGA* technique for statistical analysis of SPD tensor fields. From the synthetic experiments it is evident that the time consumption of iPGA is significantly less than the batch mode PGA. This time gain is achieved via a

Table 3. Classification results of iPGA, PGA, PCA and SVM

	Control vs. PD				Control vs. ET				PD vs. ET			
	iPGA	PGA	PCA	SVM	iPGA	PGA	PCA	SVM	iPGA	PGA	PCA	SVM
Accuracy	**0.90**	0.95	0.70	0.75	**0.93**	0.93	0.75	0.81	**0.87**	0.87	0.68	0.81
Sensitivity	**1.00**	1.00	0.90	0.90	**0.83**	0.83	0.66	0.83	**1.00**	0.83	0.83	0.83
Specificity	**0.80**	0.90	0.50	0.60	**1.00**	1.00	0.80	0.80	**0.80**	0.90	0.60	0.80
Time (s)	**15.76**	59.96			**10.60**	48.40			**10.61**	43.69		

novel combination of the incremental update of the mean tensor field as well as the incremental updates of the SVD of the data matrix. Further, it can be observed from accuracy comparisons that *iPGA* yields almost the same accuracy as the batch mode PGA. This makes our method an appealing choice for principal geodesic analysis, especially when the dimensionality of the data or the population size is a significant issue. Finally, using a simple geodesic nearest neighbor classifier, we were able to achieve high rate of classification of movement disorders from HARDI scans.

References

1. Cheng, G., Salehian, H., Vemuri, B.C.: Efficient recursive algorithms for computing the mean diffusion tensor and applications to DTI segmentation. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part VII. LNCS, vol. 7578, pp. 390–401. Springer, Heidelberg (2012)
2. Cheng, G., Vemuri, B.C., Hwang, M.-S., Howland, D., Forder, J.R.: Atlas construction from high angular resolution diffusion imaging data represented by gaussian mixture fields. In: ISBI 2011, pp. 549–552. IEEE (2011)
3. Fletcher, P.T., Joshi, S.: Riemannian geometry for the statistical analysis of diffusion tensor data. Signal Processing 87(2), 250–262 (2007)
4. Fletcher, P.T., Lu, C., Pizer, S.M., Joshi, S.: Principal geodesic analysis for the study of nonlinear statistics of shape. TMI 23(8), 995–1005 (2004)
5. Golub, G.H., Reinsch, C.: Singular value decomposition and least squares solutions. Numerische Mathematik 14(5), 403–420 (1970)
6. Ho, J., Cheng, G., Salehian, H., Vemuri, B.: Recursive karcher expectation estimators and geometric law of large numbers. In: AISTATS 2013, pp. 325–332 (2013)
7. Jian, B., Vemuri, B.C.: A unified computational framework for deconvolution to reconstruct multiple fibers from diffusion weighted mri. TMI 26(11), 1464–1471 (2007)
8. Pennec, X.: Intrinsic statistics on riemannian manifolds: Basic tools for geometric measurements. J. of Math. Imaging and Vision 25(1), 127–154 (2006)
9. Sommer, S., Lauze, F., Hauberg, S., Nielsen, M.: Manifold valued statistics, exact principal geodesic analysis and the effect of linear approximations. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part VI. LNCS, vol. 6316, pp. 43–56. Springer, Heidelberg (2010)
10. Vaillancourt, D.E., Spraker, M.B., Prodoehl, J., Abraham, I., Corcos, D.M., Zhou, X.J., Comella, C.L., Little, D.M.: High-resolution diffusion tensor imaging in the substantia nigra of de novo parkinson disease. Neurology 72(16), 1378–1384 (2009)
11. Xie, Y., Vemuri, B.C., Ho, J.: Statistical analysis of tensor fields. In: Jiang, T., Navab, N., Pluim, J.P.W., Viergever, M.A. (eds.) MICCAI 2010, Part I. LNCS, vol. 6361, pp. 682–689. Springer, Heidelberg (2010)
12. Zhao, H., Yuen, P.C., Kwok, J.T.: A novel incremental principal component analysis and its application for face recognition. IEEE Trans. on SMC 36(4), 873–886 (2006)