Designing Single- and Multiple-Shell Sampling Schemes for Diffusion MRI Using Spherical Code

Jian Cheng, Dinggang Shen, and Pew-Thian Yap

Department of Radiology and BRIC, The University of North Carolina at Chapel Hill, USA
{jian_cheng,dgshen,ptyap}@med.unc.edu

Abstract. In diffusion MRI (dMRI), determining an appropriate sampling scheme is crucial for acquiring the maximal amount of information for data reconstruction and analysis using the minimal amount of time. For single-shell acquisition, uniform sampling without directional preference is usually favored. To achieve this, a commonly used approach is the Electrostatic Energy Minimization (EEM) method introduced in dMRI by Jones et al. However, the electrostatic energy formulation in EEM is not *directly* related to the goal of optimal sampling-scheme design, i.e., achieving large angular separation between sampling points. A mathematically more natural approach is to consider the Spherical Code (SC) formulation, which aims to achieve uniform sampling by maximizing the minimal angular difference between sampling points on the unit sphere. Although SC is well studied in the mathematical literature, its current formulation is limited to a single shell and is not applicable to multiple shells. Moreover, SC, or more precisely continuous SC (CSC), currently can only be applied on the *continuous* unit sphere and hence cannot be used in situations where one or several subsets of sampling points need to be determined from an existing sampling scheme. In this case, discrete SC (DSC) is required. In this paper, we propose novel DSC and CSC methods for designing uniform single-/multi-shell sampling schemes. The DSC and CSC formulations are solved respectively by Mixed Integer Linear Programming (MILP) and a gradient descent approach. A fast greedy incremental solution is also provided for both DSC and CSC. To our knowledge, this is the first work to use SC formulation for designing sampling schemes in dMRI. Experimental results indicate that our methods obtain larger angular separation and better rotational invariance than the generalized EEM (gEEM) method currently used in the Human Connectome Project (HCP).

1 Introduction

Diffusion MRI (dMRI) is a unique technique for exploring the underlying tissue properties of white matter in the human brain. A central problem in dMRI is to reconstruct the MR signal attenuation $E(\mathbf{q})$ from a limited number of measurements in the $\mathbf{q}$-space and to estimate some meaningful quantities such as the Ensemble Average Propagator (EAP) and the Orientation Distribution Function (ODF). An effective $\mathbf{q}$-space sampling scheme is critical for the acquisition of maximum information with minimum time cost. Since white matter fascicles traverse the brain in a wide range of directions, a uniform sampling scheme with no directional preference is often preferred [1].

In the last decade, two approaches have been widely used for designing sampling schemes for single-shell acquisition. The first approach involves tessellation of a unit sphere using basic shapes such as the icosahedron. Using such a spherical tessellation approach, however, one is unable to generate a sampling scheme with an arbitrary number of sample points. The second approach is the Electrostatic Energy Minimization (EEM) method, which was introduced in dMRI by Jones et al. [1]. Some best known solutions to the EEM problem have been collected in CAMINO [2]. EEM was also recently generalized for multi-shell sampling [3,4] with staggered samples in different shells and has been adopted in the Human Connectome Project (HCP) [5]. However, the electrostatic energy formulation in EEM is not directly related to the goal of sampling scheme design, which is to maximize the angular separation between sampling points, and it is still unknown why electrostatic energy matters in dMRI reconstruction.

A good sampling scheme should have large angular separation such that the reconstruction has large angular resolution and good rotational invariance. Thus a mathematically more natural way for sampling scheme design is to maximize the minimal angular difference between sampling points, i.e., *covering radius*, on a unit sphere. Determining such point configuration is essentially the Spherical Code (SC) problem[1], and there are a collection of best known solutions for the SC problem in $\mathbb{S}^2$ [6][2]. Although SC is well studied in the mathematical literature, its current formulation is limited to a single shell and is not applicable to multiple shells. Moreover, SC, or more precisely continuous SC (CSC), currently can only be applied on the *continuous* unit sphere and hence cannot be used in situations where several subsets of sampling points need to be determined from an existing sampling scheme. In this case, discrete SC (DSC) is required, where the solution space is discrete and determined by a set of predetermined sampling points.

In this paper, we propose novel CSC and DSC methods for designing single-/multi-shell sampling schemes. We propose a greedy incremental estimation for rapid generation of solutions to the DSC and CSC problems, a Mixed Integer Linear Programming (MILP) method to solve the DSC problem, and a Riemannian gradient descent method to solve the CSC problem. Experimental results indicate that the proposed methods are capable of yielding larger covering radius and better rotational invariance than the state-of-the-art generalized EEM (gEEM) method currently used in the HCP [4,5].

2   Designing Sampling Scheme Using Spherical Code

2.1   Discrete Spherical Code (DSC) and Continuous Spherical Code (CSC)

For single-shell sampling, the SC problem is to determine a set of K points $\{\mathbf{u}_i\}_{i=1}^K$ such that the minimal distance between these points is maximized, i.e.,

$$\max_{\{\mathbf{u}_i \in \mathbb{D}\}_{i=1}^K} d(\{\mathbf{u}_i\}_{i=1}^K), \qquad d(\{\mathbf{u}_i\}_{i=1}^K) = \min_{i \neq j} \arccos |\mathbf{u}_i^T \mathbf{u}_j|, \qquad (1)$$

where $d(\{\mathbf{u}_i\}_{i=1}^K)$ is the minimal angular distance, or called *covering radius*, of point set $\{\mathbf{u}_i\}_{i=1}^K$, and $\mathbb{D} \subseteq \mathbb{S}^2$ is the solution domain. If $\mathbb{D} = \mathbb{S}^2$, Eq. (1) is a CSC problem

---

[1] http://mathworld.wolfram.com/SphericalCode.html
[2] http://neilsloane.com/grass/dim3/

for selecting K points from throughout the unit sphere $\mathbb{S}^2$. If $\mathbb{D} = \{\mathbf{u}_n\}_{n=1}^N$, a set of N predetermined points on $\mathbb{S}^2$, then Eq. (1) is a DSC problem for selecting K from N points. We use the absolute value of $\mathbf{u}_i^T \mathbf{u}_j$ in Eq. (1) because antipodal symmetric samples have the same role in diffusion MRI data reconstruction. Note that the original SC in mathematics only means CSC, while in this paper it is the first time that we propose both CSC and DSC and generalize them for multi-shell case for designing sampling scheme in dMRI.

For multi-shell sampling, the SC problem is to find a set of points $\{\mathbf{u}_{s,i}\}$ by solving

$$\max_{\{\mathbf{u}_{s,i} \in \mathbb{D}\}} wS^{-1} \sum_{s=1}^S d(\{\mathbf{u}_{s,i}\}_{i=1}^{K_s}) + (1-w)d(\{\mathbf{u}_{s,i}\}_{i=1,\ldots,K_s;s=1,\ldots,S}), \qquad (2)$$

where $\mathbf{u}_{s,i}$ is the i-th point on the s-th shell, S is the number of shells, K_s is the number of points on the s-th shell, and w is a weighting factor for balancing two terms. In Eq. (2), the first term is the mean covering radius of the S shells, and the second term is the covering radius for a combined shell containing all points from the S shells. Due to the second term, the estimated samples in different shell are staggered.

2.2   Greedy Incremental Solver

Similarly to EEM [7] and gEEM [3,4], we propose a greedy solver for incremental estimation of sampling schemes. Incremental estimation can be applied for both Eq. (1) and Eq. (2) when solving a DSC problem, i.e., when $\mathbb{D} = \{\mathbf{u}_i\}_{i=1}^N$. In step k, we estimate one point $\mathbf{u} \in \mathbb{D}$ that maximizes the cost function based on the $k-1$ points estimated in previous iterations. This incremental estimation technique can be applied to generate an approximate solution to a CSC problem, i.e., $\mathbb{D} = \mathbb{S}^2$, by approximating $\mathbb{S}^2$ using a large number of uniformly distributed points. In practice, 20481 points from a 7 order tessellation of the icosahedron are used. Incremental estimation can generate reasonable solutions in seconds.

2.3   DSC via Mixed Integer Linear Programming (MILP)

Instead of incrementally estimating samples one by one, Mixed Integer Linear Programming (MILP) can be used to estimate samples simultaneously. For $\mathbb{D} = \{\mathbf{u}_n\}_{n=1}^N$, Eq. (1) can be solved using MILP in Eq. (3a) as follows:

$$\max_{y, \{h_i\}_{i=1}^N} y \qquad (3a)$$

$$\text{s.t. } \arccos(|\mathbf{u}_i^T \mathbf{u}_j|) \geq y - (2 - h_i - h_j)M, \ \forall i > j \qquad (3b)$$

$$d_{\text{LB}} \leq y \leq d_{\text{UB}}(2K) \qquad (3c)$$

$$\sum_{i=1}^N h_i = K; \qquad h_i = 0, 1, \forall i \qquad (3d)$$

where $h_i = 1$ indicates that $\mathbf{u}_i$ is selected as one of the K points, d_{LB} is the lower bound of the covering radius, which can be set to 0 or the covering radius from an

existing sampling scheme, $d_{\text{UB}}(2K) = \arccos \sqrt{4 - \csc^2\left(\frac{\pi K}{6(K-1)}\right)}$ is the theoretical upper bound of the covering radius for $2K$ points on $\mathbb{S}^2$ [8], and M is the difference between the maximal and minimal distances of any two points $\mathbf{u}_i, \mathbf{u}_j \in \mathbb{D}$, $i \neq j$. Note that 1) after solving MILP, the solution of y, denoted as y^*, is the covering radius of the selected K samples; 2) the constraint in Eq. (3b) only takes effect when $h_i = h_j = 1$, and is automatically satisfied when $h_i = 0$ or $h_j = 0$, because the chosen M is large enough such that $\arccos(|\mathbf{u}_i^T \mathbf{u}_j|) \geq 0 \geq y^* - (2 - h_i - h_j)M$ when $h_i + h_j \leq 1$. Similarly, Eq. (2) can be solved using MILP in Eq. (4a) as follows:

$$\max_{\{y_s\},\{h_{s,i}\}} wS^{-1} \sum_{s=1}^{S} y_s + (1 - w)y_0 \tag{4a}$$

$$\text{s.t. } \arccos(|\mathbf{u}_i^T \mathbf{u}_j|) \geq y_s - (2 - h_{s,i} - h_{s,j})M, \ \ \forall s, i > j \tag{4b}$$

$$\arccos(|\mathbf{u}_i^T \mathbf{u}_j|) \geq y_0 - (2 - h_{s,i} - h_{s',j})M, \ \ \forall s, s', i > j \tag{4c}$$

$$d_{\text{LB},s} \leq y_s \leq d_{\text{UB}}(2K_s), \ \forall s; \quad d_{\text{LB},0} \leq y_0 \leq d_{\text{UB}}\left(2\sum_{i=1}^{S} K_s\right) \tag{4d}$$

$$\sum_{i=1}^{N} h_{s,i} = K_s, \ \forall s; \quad \sum_{s=1}^{S} h_{s,i} \leq 1, \ \forall i; \quad h_{s,i} = 0, 1, \ \forall i, s \tag{4e}$$

where $h_{s,i} = 1$ indicates that $\mathbf{u}_i$ is selected as one of the K_s points on the s-th shell, $d_{\text{LB},s}$ and $d_{\text{UB}}(2K_s)$ are the lower and upper bounds of the covering radius on the s-th shell, and $d_{\text{LB},0}$ and $d_{\text{UB}}(2\sum_{i=1}^{S} K_s)$ are the lower and upper bounds for the combined shell with all points. Constraints in Eq. (4b) and Eq. (4c) are respectively for the first and second terms in Eq. (4a). $\sum_{s=1}^{S} h_{s,i} \leq 1$ makes $\mathbf{u}_i$ to be selected at most one shell such that the estimated samples are staggered in different shells.

MILP problem can be solved using branch and bound method which iteratively solves the relaxed LP program. In our implementation, we solve Eq. (3a) and Eq. (4a) using GUROBI [9], which can obtain the global solution or at least a reasonable solution within minutes for DSC. In practice, we progressively increase the lower bound d_{LB} based on the solutions estimated in previous iterations to find a better feasible solution within 10 minutes which is good enough in experiments.

2.4   CSC via Riemannian Gradient Descent

When $\mathbb{D} = \mathbb{S}^2$, Eq. (1) can be solved using a Riemannian gradient descent method. For each iteration, we detect the pairs of points $\{(\mathbf{u}_{p_0}, \mathbf{u}_{p_1})\}$ whose angular differences are equal to the minimal angular difference computed from all point pairs. Noting that the Euclidean gradient of function $d(\{\mathbf{u}_j\}_{j=1}^K)$ is

$$\frac{\partial d(\{\mathbf{u}_j\}_{j=1}^K)}{\partial \mathbf{u}_i} = \begin{cases} -\sum_{p_1} \frac{1}{\sqrt{1-(\mathbf{u}_{p_0}^T \mathbf{u}_{p_1})^2}} \text{sign}(\mathbf{u}_{p_0}^T \mathbf{u}_{p_1})\mathbf{u}_{p_1} & \text{if } i = p_0 \\ -\sum_{p_0} \frac{1}{\sqrt{1-(\mathbf{u}_{p_0}^T \mathbf{u}_{p_1})^2}} \text{sign}(\mathbf{u}_{p_0}^T \mathbf{u}_{p_1})\mathbf{u}_{p_0} & \text{if } i = p_1 \\ 0 & \text{if } i \notin \{p_0, p_1\} \end{cases} \tag{5}$$

the Riemannian gradient for $\mathbf{u}_i$ can be computed as [10]

$$\nabla_{\mathbf{u}_i} d(\{\mathbf{u}_j\}) = \frac{\partial d(\{\mathbf{u}_j\})}{\partial \mathbf{u}_i} - \left(\mathbf{u}_i^T \frac{\partial d(\{\mathbf{u}_j\})}{\partial \mathbf{u}_i}\right) \mathbf{u}_i. \tag{6}$$

Algorithm 1. CSC via Riemannian Gradient Descent

Input: Initialization $\{\mathbf{u}_{s,i}\}_{i=1}^{K_s}$, $s = 1, ..., S$.
Output: Refined $\{\mathbf{u}_{s,i}\}_{i=1}^{K_s}$, $s = 1, ..., S$.
$k = 0$, $t_0 = 0.1$, determine the set of point pairs $P^{(k)} = \{(\mathbf{u}_{p_0^s}, \mathbf{u}_{p_1^s})\}_{s=0}^{S}$ that have minimal distances in $S + 1$ shells;
repeat

> Record: $P_0^{(k)} \leftarrow P^{(k)}$;
> **repeat**
>
> > Calculate Riemannian gradient $\boldsymbol{v}_{s,i}$, $\forall s, i$, using point-pair set $P_0^{(k)}$ and Eq. (6) ;
> > **if** $\forall s, i$, $\|\boldsymbol{v}_{s,i}\| \leq \epsilon$ **then** break;
> > Choose step size $t \in (0, t_0]$ via inexact line search;
> > $\mathbf{u}_{s,i}^{(k+1)} = \mathrm{Exp}_{\mathbf{u}_{s,i}^{(k)}}\left(t\frac{\boldsymbol{v}_{s,i}}{\max_{s,i}\{\|\boldsymbol{v}_{s,i}\|\}}\right)$, $\forall s, i$;
> > Detect updated set $P^{(k+1)}$ based on $\{\mathbf{u}_{s,i}^{(k+1)}\}$;
> > **if** $P^{(k+1)} \not\subseteq P_0^{(k)}$ **then** $P_0^{(k)} \leftarrow P_0^{(k)} \cup P^{(k+1)}$;
>
> **until** $P^{(k+1)} \subseteq P_0^{(k)}$;
> $k \leftarrow k + 1$;

until *Cost function does not change*;

Then the gradient descent update is performed using

$$\mathbf{u}_i^{(k+1)} = \mathrm{Exp}_{\mathbf{u}_i^{(k)}}\left(t\frac{\nabla_{\mathbf{u}_i} d(\{\mathbf{u}_j\})}{\max_i\{\|\nabla_{\mathbf{u}_i} d(\{\mathbf{u}_j\})\|\}}\right), \quad \mathrm{Exp}_{\mathbf{u}}(\boldsymbol{v}) = \mathbf{u}\cos\|\boldsymbol{v}\| + \frac{\boldsymbol{v}}{\|\boldsymbol{v}\|}\sin\|\boldsymbol{v}\|, \quad (7)$$

where the largest norm of gradient vectors is used for normalization of all gradient vectors, $\mathrm{Exp}_{\mathbf{u}}(\boldsymbol{v})$ is the exponential map [10] that maps the gradient vector $\boldsymbol{v}$ from the tangent space of the unit sphere at $\mathbf{u}$ to the unit sphere itself. Note that Riemannian gradient descent is performed on all $\{\mathbf{u}_i\}_{i=1}^{K}$ simultaneously. For the multi-shell case, point pairs $\{(\mathbf{u}_{p_0^s}, \mathbf{u}_{p_1^s})\}$ with minimal distances are detected from all S shells and the combined shell with all points. Then similar gradient descent is performed, taking the gradient of Eq. (4a) as a summation of gradients from $S + 1$ shells.

One important issue of the proposed Riemannian gradient descent method is that, after each gradient descent, the set of point pairs $\{(\mathbf{u}_{p_0^s}, \mathbf{u}_{p_1^s})\}$ with minimal distances may change, and the cost function and its gradient, which depends on these pairs, may also change. To solve this issue, in each iteration, we compare the updated set of point pairs with the previous pair set before the one step gradient descent, and if the updated pair set has some new pairs which are not in the previous pair set, then we re-perform the gradient descent using the joint set of these two sets. See Algorithm 1 for implementation details. The algorithm is efficient and converges in seconds.

Since the optimization problem is highly non-convex, a good initialization is important for a good solution. Initialization can be set as a set of random points, the solution given by incremental estimation, or the solution given by MILP. Note that when using MILP for the initialization, solving the CSC becomes slow when a large number of points are involved. In practice, we apply MILP to 321 points obtained from spherical tessellation and use the MILP solution as the initialization of the gradient descent method for CSC.

Table 1. Covering radii of multi-shell sampling schemes with number of samples 28×3 and 90×3 generated by various methods. Note that gEEM results for 90×3 are not available in [4], and EEM results from CAMINO are individually for each single shell (the scheme with 270 samples is not available).

	Shell 1 (28)	Shell 2 (28)	Shell 3 (28)	Combined (28 × 3)
gEEM [4]	22.2°	22.2°	22.0°	13.2°
Incr. gEEM [4]	19.2°	19.7°	19.3°	4.7°
Incr. CSC ($N = 20481$)	21.3°	19.3°	21.1°	10.5°
MILP ($N = 321$)	23.8°	23.8°	24.3°	13.3°
Incr. CSC + Grad. Desc.	24.3°	23.5°	24.2°	10.9°
MILP + Grad. Desc.	**25.7°**	**25.7°**	**25.4°**	**13.6°**
EEM (CAMINO) [1,2]	25.7°	25.7°	25.7°	15.6°
	Shell 1 (90)	Shell 2 (90)	Shell 3 (90)	Combined (90 × 3)
Incr. gEEM [4]	10.8°	10.3°	10.5°	2.4°
Incr. CSC ($N = 20481$)	10.4°	9.7°	10.4°	4.6°
MILP ($N = 321$)	13.3°	13.5°	13.3°	**7.9°**
Incr. CSC + Grad. Desc.	13.0°	13.5°	12.8°	5.2°
MILP + Grad. Desc.	**14.6°**	**15.0°**	**14.8°**	7.5°
EEM (CAMINO) [1,2]	15.1°	15.1°	15.1°	-

3   Experiments

Separation of Sampling Schemes. We evaluated the effectiveness of the proposed multi-shell MILP-based DSC method in Eq. (4a) with $w = 1$, by gauging whether it can separate points on a set of samples into several subsets, keeping the points in each subset as uniform as possible. We used the `subsetpoints` program in CAMINO [1,2] for comparison, which performs the same task by using simulated annealing. For this evaluation, we randomly mixed two sets of uniform points, one set consisting of 81 points generated by spherical tessellation and the other set consisting of 60 points from CAMINO generated by EEM. Separation of these 141 points into subsets respectively with 81 and 60 samples should ideally give results that match the original uniform point sets. Our method, which uses MILP, gave results that exactly match the original point sets within 5 seconds. `subsetpoints` in CAMINO saves the result every hour when it runs. It gave two incorrect points in each subset after running for 2 hours, 7 incorrect points after 8 hours, and the correct result after 9 hours. Although the correct result had been obtained, the program continued to run for hours until the simulated annealing temperature was finally small enough.

Multi-shell Angular Separation. We evaluated the effectiveness of the proposed multi-shell DSC (MILP) and CSC (incremental estimation) methods in generating a three shell sampling scheme, each shell consisting of K sampling points. We tested two cases: $K = 28$ and $K = 90$. These two cases were used such that we could compare our results with those given by gEEM and incremental gEEM [4]; results for 28 points per shell were reported in [4] and results for 90 points per shell were utilized in the HCP. MILP was used to select $K \times 3$ points from 321 points given by spherical tessellation. In incremental CSC, the selection was carried out using 20481 uniformly distributed points. Gradient descent was then used to refine these results. In Table 1, the
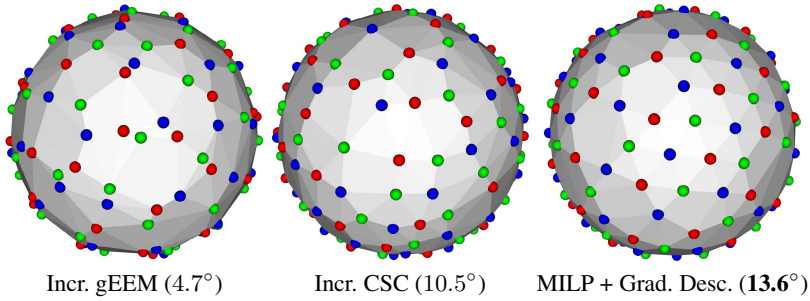
Incr. gEEM $(4.7°)$ Incr. CSC $(10.5°)$ MILP + Grad. Desc. $(\mathbf{13.6°})$

Fig. 1. Multi-shell sampling schemes with 28×3 samples generated by three methods and their combined covering radii showed in Table 1. The colors differentiate the sampling points from the three shells.

covering radii, i.e., the minimal angular differences, of these results were compared with those given by gEEM and incremental gEEM [4]. The multi-shell results of gEEM with $K = 28$ were extracted from [4]. The results of incremental gEEM with $K = 28, 90$, which have been used in HCP, were obtained from the website[3] created for [4]. The covering radii for single-shell results given by EEM in CAMINO are shown for reference. Fig. 1 visualizes the results with $K = 28$ generated by incremental gEEM, incremental CSC, and gradient descent with MILP initialization. Table 1 and Fig. 1 demonstrate clearly that the proposed MILP method and incremental CSC estimation yield larger covering radii than gEEM and incremental gEEM in all three shells and the combined shell containing all points. The proposed gradient descent method with MILP initialization yields the best angular separation in the multi-shell case, and its results are comparable with the single-shell results given by EEM in CAMINO, although the optimization was done with respect to all shells.

Rotational Invariance in Reconstruction. We tested the multi-shell sampling schemes with 28×3 samples in Table 1 on whether they give consistent reconstruction results for the synthetic signals generated by rotated models. A mixture of tensor model was used: $E(q\mathbf{u}) = 0.5 \exp(-q^2 \mathbf{u}^T \mathbf{D}_1 \mathbf{u}) + 0.5 \exp(-q^2 \mathbf{u}^T \mathbf{D}_2 \mathbf{u})$, where $b = q^2 = 1000, 2000, 3000 \, \text{s/mm}^2$, and the two tensors $\mathbf{D}_1$, $\mathbf{D}_2$ have the same eigenvalues $[1.7, 0.2, 0.2] \times 10^{-3} \, \text{mm}^2/\text{s}$ and have a crossing angle of $60°$. This signal generation is repeated 20481 times by rotating the model according to 20481 uniformly distributed directions generated by spherical tessellation. We performed Spherical Polar Fourier Imaging (SPFI) with spherical order 6 and radial order 2 [11] to estimate the EAP profiles with radius of 15μm, detected the peaks of the EAP profiles, and compared the detected peaks with the ground-truth fiber directions in these 20481 tests. The means and standard deviations are shown in Table 2. Note that we have omitted the results for gEEM because the algorithm is not publicly available. It is clear from the table that the proposed methods yield significantly lower mean angular differences (paired t-test, $p < 0.001$) with lower standard deviations than incremental gEEM. Similar to Table 1, gradient descent with MILP initialization gives the best result.

---

[3] http://www.emmanuelcaruyer.com/q-space-sampling.php

Table 2. Angular differences between estimated and ground-truth fiber directions for the sampling schemes generated by different methods

	Incr. CSC	Incr. CSC + Grad. Desc.	MILP	MILP + Grad. Desc.	Incr. gEEM
Angular Difference	$1.44° \pm 0.69°$	$1.40° \pm 0.69°$	$1.30° \pm 0.68°$	$\mathbf{1.29° \pm 0.68°}$	$1.72° \pm 0.79°$

4   Conclusion

To our knowledge, this is the first work on designing single-/multi-shell sampling schemes using continuous spherical code (CSC) and discrete spherical code (DSC) formulations. We propose an incremental estimation method for both CSC and DSC, a mixed-integer linear programming (MILP) method for DSC, and a Riemannian gradient descent method for CSC. The experimental results showed that, compared with the gEEM method and its incremental variant that has been used in the HCP, the sampling schemes by the proposed gradient descent with MILP initialization and incremental CSC yield larger covering radius and better rotation invariance.

References

1. Jones, D.K., Horsfield, M.A., Simmons, A.: Optimal strategies for measuring diffusion in anisotropic systems by magnetic resonance imaging. Magnetic Resonance in Medicine 42, 515–525 (1999)
2. Cook, P., Bai, Y., Nedjati-Gilani, S., Seunarine, K., Hall, M., Parker, G., Alexander, D.: Camino: Open-source diffusion-MRI reconstruction and processing. In: ISMRM (2006)
3. Caruyer, E., Cheng, J., Lenglet, C., Sapiro, G., Jiang, T., Deriche, R.: Optimal Design of Multiple Q-shells experiments for Diffusion MRI. In: CDMRI Workshop, MICCAI 2011 (2011)
4. Caruyer, E., Lenglet, C., Sapiro, G., Deriche, R.: Design of multishell sampling schemes with uniform coverage in diffusion MRI. Magnetic Resonance in Medicine (2013)
5. Sotiropoulos, S.N., Jbabdi, S., Xu, J., Andersson, J.L., Moeller, S., Auerbach, E.J., Glasser, M.F., Hernandez, M., Sapiro, G., Jenkinson, M., et al.: Advances in diffusion MRI acquisition and processing in the Human Connectome Project. NeuroImage 80, 125–143 (2013)
6. Conway, J.H., Hardin, R.H., Sloane, N.J.: Packing lines, planes, etc.: Packings in Grassmannian spaces. Experimental Mathematics 5(2), 139–159 (1996)
7. Deriche, R., Calder, J., Descoteaux, M.: Optimal real-time q-ball imaging using regularized kalman filtering with incremental orientation sets. Medical Image Analysis 13(4), 564–579 (2009)
8. Tóth, L.F.: On the densest packing of spherical caps. The American Mathematical Monthly 56(5), 330–331 (1949)
9. Gurobi Optimization, Inc.: Gurobi optimizer reference manual (2014)
10. Cheng, J., Ghosh, A., Jiang, T., Deriche, R.: A Riemannian Framework for Orientation Distribution Function Computing. In: Yang, G.-Z., Hawkes, D., Rueckert, D., Noble, A., Taylor, C. (eds.) MICCAI 2009, Part I. LNCS, vol. 5761, pp. 911–918. Springer, Heidelberg (2009)
11. Cheng, J., Ghosh, A., Jiang, T., Deriche, R.: Model-free and Analytical EAP Reconstruction via Spherical Polar Fourier Diffusion MRI. In: Jiang, T., Navab, N., Pluim, J.P.W., Viergever, M.A. (eds.) MICCAI 2010, Part I. LNCS, vol. 6361, pp. 590–597. Springer, Heidelberg (2010)