Towards a general framework for fast and feasible k-space trajectories for MRI based on projection methods

Sharma, Shubham; Coutino, Mario; Chepuri, Sundeep Prabhakar; Leus, Geert; Hari, K. V. S.

DOI
10.1016/j.mri.2020.06.016

Publication date
2020

Document Version
Final published version

Published in
Magnetic Resonance Imaging

Citation (APA)

Important note
To cite this publication, please use the final published version (if applicable). Please check the document version above.
Green Open Access added to TU Delft Institutional Repository

'You share, we take care!' - Taverne project

https://www.openaccess.nl/en/you-share-we-take-care

Otherwise as indicated in the copyright section: the publisher is the copyright holder of this work and the author uses the Dutch legislation to make this work public.
Towards a general framework for fast and feasible $k$-space trajectories for MRI based on projection methods

Shubham Sharma$^a,⁎$, Mario Coutino$^b$, Sundeep Prabhakar Chepuri$^a$, Geert Leus$^b$, K.V.S. Hari$^a$

$^a$ Department of Electrical Communication Engineering, Indian Institute of Science, Bangalore, India
$^b$ Department of Microelectronics, Delft University of Technology, Netherlands

Abstract

The design of feasible trajectories to traverse the $k$-space for sampling in magnetic resonance imaging (MRI) is important while considering ways to reduce the scan time. Over the recent years, non-Cartesian trajectories have been observed to result in benign artifacts and being less sensitive to motion. In this paper, we propose a generalized framework that encompasses projection-based methods to generate feasible non-Cartesian $k$-space trajectories. This framework allows to construct feasible trajectories from both random or structured initial trajectories, e.g., based on the traveling salesman problem (TSP). We evaluate the performance of the proposed methods by simulating the reconstruction of $128 \times 128$ and $256 \times 256$ phantom and brain MRI images in terms of structural similarity (SSIM) index and peak signal-to-noise ratio (PSNR) using compressed sensing techniques. It is observed that the TSP-based trajectories from the proposed projection method with constant acceleration parameterization (CAP) result in better reconstruction compared to the projection method with constant velocity parameterization (CVP) and this for a similar read-out time. Further, random-like trajectories are observed to be better than TSP-based trajectories as they reduce the read-out time while providing better reconstruction quality. A reduction in read-out time by up to 67% is achieved using the proposed projection with permutation (PP) method.

1. Introduction

Magnetic resonance imaging (MRI) is a non-invasive imaging modality that provides detailed soft tissue images without exposing the subject to any harmful radiation. The MR signal is frequency encoded and can be directly mapped to the spatial frequency domain called the $k$-space [1]. The image is constructed by taking the inverse Fourier transform of the $k$-space signal. The $k$-space is traversed along a trajectory, $s(t) : \mathbb{R} \rightarrow \mathbb{R}^2$, $s(t) = [s_x(t), s_y(t)]^T$, $t \in [0, T]$, (1) where the parameter $t$ corresponds to time. Here, we consider 2D scanning, i.e., we assume that a slice has been selected and the $k$-space is traversed in the $k_x$-$k_y$ plane. The trajectory has as initial and final points $s(0) = s_{\text{start}}$ and $s(T) = s_{\text{end}}$, respectively, and it is governed by the magnetic gradients $g(t) = [g_x(t), g_y(t)]^T$, $t \in [0, T]$ as

$s_x(t) = \int_0^t g_x(r) \text{d}r, \quad s_y(t) = \int_0^t g_y(r) \text{d}r$ .

(2)

where $\gamma$ is the gyromagnetic ratio, which is $\gamma = 42.58\text{MHz/T}$ for hydrogen. The most commonly used trajectory is the Cartesian trajectory in which the $k$-space is traversed line-by-line. Non-Cartesian trajectories such as the spiral [2,3] and the radial [4] trajectories have also been used extensively in practice. Non-Cartesian trajectories are advantageous over Cartesian trajectories as they can better utilize the hardware of an MRI scanner and provide incoherent artifacts. However, long acquisition times in MRI are a limitation and also pose a problem for claustrophobic patients. Therefore, to reduce the acquisition time in MRI, the design of faster pulse sequences [5–7], parallel imaging [8–11] and compressed sensing (CS) techniques [12,13] have been proposed. More recently, deep learning-based methods for MRI have been proposed in the literature as well [14–19]. In addition to providing an improvement in the reconstruction quality, the main advantage observed in these methods is the speed of reconstruction compared to iterative reconstruction methods.

CS theory allows undersampling the $k$-space data based on the fact that MR images are sparse in transform domains such as wavelet and finite differences. The sparse signal is then recovered by means of non-linear optimization methods [20–22]. However, there are two main considerations that need to be accounted for when designing sampling...
schemes for MRI. The first is related to variable density sampling (VDS) in which low frequency region of the k-space is more densely sampled as compared to the high frequency region. The use of VDS has become an integral part of the system owing to the theory of level sparsity as advocated in [23–25]. The second is the fact that the magnetic gradients are physically limited in their maximum magnitude and slew rate. Therefore, to achieve feasible trajectories, the design must satisfy these gradient constraints. Although there has been a lot of research to find optimal density functions [26,27], to design sparsifying transforms [28,29] and to develop CS image reconstruction methods [30,31] for a Cartesian trajectory, less attention has been given to the design of feasible non-Cartesian k-space trajectories satisfying the magnetic gradient constraints.

A recent approach to obtain a trajectory from randomly under-sampled points in the k-space using VDS is proposed in [32], in which the shortest path through these points is computed by solving a traveling salesman problem (TSP). However, TSP-based trajectories are not feasible for implementation in an MRI machine since the physical constraints as compared to the high frequency region. The use of VDS has become feasible if it satisfies the gradient constraints as discussed before are generally not satisfied. To formalize this, we introduce a notion of distance and velocity with respect to sampling in the k-space for MRI. Consider an arbitrary 2D trajectory \( s(t) = (s_x(t), s_y(t)) \). Assuming a fixed sampling interval \( t_h \), this trajectory can be described by an m-point discrete 2D trajectory \( s \in \mathbb{R}^{m} = [s_{x}^T \ s_{y}^T]^T \) where \( s_{x} \in \mathbb{R}^{m} \) and \( s_{y} \in \mathbb{R}^{m} \) denote the x- and y-coordinates, respectively. We denote the kth point along the trajectory as \( s_k = [(s_{x,k}) \ (s_{y,k})]^T, 1 \leq k \leq m \). Throughout this work, we assume that the sampling interval \( t_h \) is small compared to the overall scanning time. In practice, the minimum \( t_h \) is driven by the machine. Under these settings, (2) can be discretized (in 2D) as

\[ s_k = y \sum_{j=1}^{i} g_j t_h \quad 1 \leq i \leq m, \]

where \( g_i = [(g_{x,i}) \ (g_{y,i})]^T \), \( 1 \leq i \leq m \) is the magnetic gradient magnitude at the kth point.

Definition 1 (Read-out time) The read-out time is the amount of time required to collect the MR signal by traversing the k-space. It is directly proportional to the length of the trajectory. The discretization (4) assumes that two consecutive points in the trajectory are traversed in the sampling time, \( t_h \), hence, the kth point is reached at time \( t_k = mt_h \). The total read-out time required to traverse the trajectory will be \( T = mt_h \).

Definition 2 (Distance) The distance traveled up to the kth point is given as

\[ l_i = \sum_{j=1}^{i} \| s_j - s_{j-1} \|, \quad 2 \leq i \leq m. \]

Definition 3 (Instantaneous velocity) The instantaneous velocity at the kth point on the trajectory can be obtained from (4) and (5) as

\[ v_i = \frac{l_i - l_{i-1}}{t_h} = \frac{\| s_i - s_{i-1} \|}{t_h} = \gamma \| g_i \|, \quad 2 \leq i \leq m. \]

The instantaneous velocity is proportional to the magnitude of the gradient magnitude at that time instant. Hence, the gradient constraints restrict the distance traversed per second. Note that the maximum velocity with which the trajectory can be traversed is \( v_{\text{max}} = \gamma g_{\text{max}} \).

Definition 4 (Instantaneous acceleration) The instantaneous acceleration is given as

\[ a_i = \frac{v_i - v_{i-1}}{t_h} = \gamma \frac{\| g_i \| - \| g_{i-1} \|}{t_h}, \quad 2 \leq i \leq m. \]

The maximum acceleration with which the trajectory can be traversed is \( a_{\text{max}} = \gamma g_{\text{max}} \).

Definition 5 (Feasible trajectories) The maximum magnitude and slew rate constraints on the magnetic gradients \( g \) are equivalent to velocity and acceleration constraints on the trajectory \( s \). A trajectory is feasible if it satisfies these constraints. To formalize this, we introduce the block diagonal matrix notation

\[ A^{(2)} = \text{blkdiag}(A) = \begin{bmatrix} A & 0 \\ 0 & A \end{bmatrix}. \]

The set of m-point feasible curves \( s \in \mathbb{R}^{2m} = [s_{x}^T \ s_{y}^T]^T \) can then be defined as
\( \mathcal{S}^m = \{ s \in \mathbb{R}^m : \| D_1^2 s \|_\infty \leq t_1 \gamma G_{\text{max}}, \| D_2^2 s \|_\infty \leq t_2^2 \gamma S_{\text{max}} \} \)

\begin{align*}
D_1^{(2)} &= \text{blkdiag}(D_1), \\
D_2^{(2)} &= \text{blkdiag}(D_2)
\end{align*}

where

\[
D_1 = \begin{bmatrix}
0 & 0 & 0 & \ldots & 0 \\
-1 & 1 & 0 & \ldots & 0 \\
0 & -1 & 1 & \ldots & 0 \\
0 & 0 & 0 & \ldots & 1
\end{bmatrix}_{m \times m}
\]

being the first-order difference matrix, and

\[
D_2 = -D_1^T D_1 \in \mathbb{R}^{m \times m}
\]

the second-order difference matrix. Here, the gradient constraints are taken to be rotation invariant, i.e., the gradient coil in each direction is assumed to work independent of the other coils and each of them can reach the maximum constraints.

### 2.2. Design of feasible k-space trajectories

A common heuristic method for designing feasible trajectories is to start from an arbitrarily parameterized curve. The curve is then reparameterized to obtain gradients which satisfy the magnetic gradient constraints using optimal control theory [33,34,41], or the trajectory is parameterized to obtain gradients which satisfy the magnetic gradient constraints as the maximum velocity at which the curve can be traversed is \( \gamma G_{\text{max}} \). In this work, we propose to describe the CVP operation using a linear operator \( A_0(\alpha, c) \in \mathbb{R}^{m \times n} \) that is applied to \( s \) such that \( c_{\text{par}} = A_0(\alpha, c) s \). This notation is to indicate that the linear CVP operator is dependent on \( \alpha \) and \( c \) and the subscript ‘C’ is to denote CVP. Hence, the problem in (9) becomes

\[
(P_{\text{proj-CVP}}): \min_{s \in \mathbb{R}^m} \frac{1}{2} \| s - A_0^{(2)}(\alpha, c) s \|_2^2
\]

(10)

where as before \( A_0^{(2)}(\alpha, c) = \text{blkdiag}(A_0(\alpha, c)) \). The linear CVP operator can be decomposed into two operators:

1. a linear interpolation matrix \( L \).
2. a selection matrix \( \Pi_{\alpha, c} \) such that the consecutive points are traversed with a constant velocity.

Hence, \( A_0(\alpha, c) = \Pi_{\alpha, c} L \). The details for the design method for the two matrices is provided in Appendix A.

#### 2.2.2. Other parameterizations

Although constant velocity parameterization (CVP) is used here as an initial parameterization, other parameterizations can be used. One of the simplest parameterizations of a curve is the arc-length parameterization. Following the convention, we can represent a trajectory \( s \) by samples in the arc-length parameterization as \( e(t) = \{ e_1(t), e_2(t), \ldots, e_m(t) \} \), \( 1 \leq i \leq m \), where \( t \) is as described before. When a trajectory is uniformly sampled in the arc-length parameterization, all consecutive sample points are equi-distant on the trajectory. Hence, traversing such a trajectory at a fixed sampling time interval \( t \), makes arc-length parameterization equivalent to CVP, which is used in the projection method [36]. This is the most common parameterization in the literature. Similar to CVP, we propose another possible parameterization: Constant acceleration parameterization (CAP). In this method, the trajectory is to be traversed at a constant acceleration \( a \) as a fraction of the maximum acceleration possible \( a_{\text{max}} = \gamma G_{\text{max}} \). The details are discussed in Section 2.4.1.

There are no strict or clear measures to find a good parameterization. Hence, in this paper, we generalize the projection method and propose further possible parameterizations that might result in a shorter read-out time and/or better reconstruction performance.

#### 2.3. Proposed generalized framework for projection-based trajectory design

The CVP of \( c \) as used in the projection method [36] is in fact a linear operation on \( c \) as shown in the previous section. Therefore, it is natural to ask if there are other such operations on \( c \) that could result in a feasible trajectory that provides a better reconstruction quality or reduces the read-out time with similar reconstruction performance. Here, we explore such possible linear operations on \( c \) under a generalized framework. This approach can be expressed as the following optimization problem

\[
(P_{\text{gen-proj}}): \min_{s \in \mathbb{R}^m} \frac{1}{2} \| s - A^{(2)}_1 c \|_2^2
\]

(11)

where \( A^{(2)}_1 c \) is a linear or non-linear function of the known vector \( c \) and \( \mathcal{A} \) is a constraint set (more details in Section 2.4) associated to the parameterization required to avoid a trivial solution to the problem (11). When this transformation is linear, i.e., \( A^{(2)}_1 = A^{(2)} \), with \( A^{(2)} \in \mathbb{R}^{2m \times 2m} \), the optimization problem could be defined as

\[
(P_{\text{gen-proj}}): \min_{s \in \mathbb{R}^m} \frac{1}{2} \| s - A^{(2)} c \|_2^2
\]

(12)

where \( A^{(2)} = \text{blkdiag}(A) \). Here, \( A^{(2)} \) may or may not depend on \( c \). We aim to obtain a non-trivial solution to the above problems. In the
previous section, we had provided a method to fit the original projection method into the proposed generalized framework. For \( A^{(2)} = A^{(2)}_v \) (the linear CVP operator), \((P_{\text{proj}})\) becomes the same as \((P_{\text{proj}})\).

In this case, \( A^{(2)} \) depends on \( c \). The advantage of the generalized framework is that it provides opportunities to explore different parameterizations using different structures on \( A^{(2)} \) which may lead to better projection-based feasible trajectories.

2.4. Proposed parameterizations

In the following subsections, we explore some possible variations of \( A^{(2)} \).

2.4.1. Constant acceleration parameterization (CAP)

As already mentioned in Section 2.2.2, we can consider a parameterization method using constant acceleration. Just as the CVP curve has successive points that have constant velocity (i.e., equidistant points for fixed sampling time \( t_s \)), CAP will have successive points that have constant acceleration (i.e., the distance between points increases with the progression of the trajectory). Acceleration naturally occurs in curves like the spiral where the trajectory starts from the center of the \( k \)-space and moves towards the boundary. The CAP operator can also be described by a linear operator which will be denoted by \( A^{(2)}_s(\beta, c) \in \mathbb{R}^{n \times n} \) where \( \beta \) is the fraction of \( c_{\text{max}} \) used for the parameterization. The trajectory design problem now becomes:

\[
(P_{\text{CAP}}): \min_{s \in \mathbb{R}^m} \frac{1}{2} \| s - A^{(2)}(\beta, c) s \|^2
\]

(13)

where \( A^{(2)}_s(\beta, c) = \text{blkdiag}(A^{(2)}_s(\beta, c)) \). The linear CAP operator can be computed similar to the constant velocity matrix by first interpolating \( c \) using the linear interpolation matrix \( I \) and then picking points with the selection matrix \( I_{\beta} \) such that the resultant points have constant acceleration. These two operations can be combined as \( A^{(2)}_s(\beta, c) = I_{\beta} I \). The design method for these two matrices is provided in Appendix B.

One should also note that when a curve is parameterized with constant velocity, the acceleration constraints \( \|D_c(2)s\|_\infty \leq \beta \gamma_{\text{max}} \) are generally not taken care of. This results in a distortion of the original trajectory after projection onto the feasible curves, which in certain cases, leads to a poor reconstruction performance. For example, in case of a spiral curve, the CVP curve might not satisfy the acceleration/slew rate constraint near the center of the curve as the curvature is high in that region. Similarly, if the spiral is constant acceleration parameterized, the velocity/gradient magnitude constraint \( \|D_c(2)s\|_\infty \leq \beta \gamma_{\text{max}} \) might not be satisfied near the boundary as the curvature is low in that region. This will result in distortion of the trajectory after projection (near the center for CVP and near the boundary for CAP). Hence, proper care needs to be taken for either parameterization. One should properly parameterize the curve by adjusting the velocity and acceleration such that the spiral is not distorted after projection. However, this may result in a significantly higher read-out time [42].

2.4.2. A as a general banded matrix

The structure of \( A^{(2)}_v(\alpha, c) \) and \( A^{(2)}_s(\beta, c) \) is similar to a banded matrix. The structure is so because these matrices work on a few consecutive points in \( c \). Note that the number of columns of the matrix remains the same for all values of \( \alpha \) and \( \beta \). However, with increasing \( \alpha \) (or \( \beta \)), the number of rows increases as there are fewer points to be chosen with higher velocity (or acceleration).

We now adapt the structure of the CVP/CAP matrices by making \( A \) a square matrix with a banded structure, denoted as \( A_\omega \). \( A_\omega \) is made a square matrix instead of a tall matrix because we do not want to specify a certain parameterization. Remember that the number of rows of \( A \) in CVP/CAP depends on the parameterization. Further, the varying bandwidth of the CVP/CAP operators is changed to have a constant bandwidth, i.e., a diagonal band of fixed bandwidth is used. The design problem then becomes:

\[
(P_{\text{BP}}): \min_{s \in \mathbb{R}^m} \frac{1}{2} \| s - A^{(2)}_s(\beta, c) s \|^2
\]

(14)

where \( A^{(2)}_s(\beta, c) = \text{blkdiag}(A^{(2)}_s(\beta, c)) \), \( A_\omega \in \mathcal{S}_b \) with \( \mathcal{S}_b \) being the set of banded matrices with \( l \) and \( u \) as the lower and upper bandwidths, respectively. This method will be called general banded projection (GBP) method henceforth. We consider two specific cases:

- GBP (Type-1): when \( A_\omega \) is taken as the identity matrix.
- GBP (Type-2): when \( A_\omega \) is taken as a banded matrix with normal random entries and rows normalized to add up to 1.

In the GBP (Type-1) method, \( c \) is directly projected onto the set of feasible trajectories. In the GBP (Type-2) method, a random linear combination of a few consecutive points in \( c \) is projected onto \( \mathcal{S} \). The effect of the two cases is discussed in Section 3.

2.4.3. Other variations with \( A = I \)

It is assumed that each sample is taken in the sampling time \( t_s \). Hence, to reduce the read-out time we need to limit the number of sample points in the trajectory. However, fewer sample points might not be enough for a good image reconstruction [25,43]. Hence, there is a trade-off between the read-out time and image quality. We aim to reduce the read-out time such that the resultant trajectory provides a minimal compromise on the image quality. The main idea in the following two methods is that we project \( c \) as it is without doing any reparameterization. Then to obtain feasible trajectories with more sample points such that the reconstruction quality is improved, we use interpolation methods.

Here, we discuss two variations of this idea.

(a) Constrained Length Trajectory (COLT)

In this variation, \( c \) is not parameterized as was done in the projection method, i.e., \( A = I \). In addition to just the projection function in (12), we here also add a cost on the length of the segments of \( s \). This method to find a feasible trajectory is hereby called the constrained length trajectory (COLT) method [44]. It can be formulated as the following constrained convex optimization problem

\[
(P_{\text{COLT}}): \min_{s \in \mathbb{R}^m} \frac{1}{2} \| s - c \|^2 + \frac{\lambda}{2} \| D_c(2)s \|^2
\]

(15)

where \( \lambda \in \mathbb{R}^+ \) is a weighting parameter, \( \| D_c(2)s \|^2 \) is the sum of squares of the Euclidean distances between consecutive points of \( s \). This imposes a cost on the segments of the trajectory \( s \) which in turn decides the overall length of the trajectory. Hence, we project the given curve \( c \) with arbitrary parameterization (i.e., it contains the original randomly sampled points in any order, for instance the TSP order to construct the TSP curve) onto the set of feasible curves and include a cost on the length of the segments of the trajectory. The resultant trajectory \( s_{\text{COLT}} \in \mathbb{R}^{2m} \) will have the same number of sample points as that of \( c \). As the final step, to reduce the variation of the velocity in consecutive points, \( s_{\text{COLT}} \) is parameterized using a constant velocity, denoted \( s_{\text{COLT}}' \in \mathbb{R}^{2m} \). A variable number of sample points, and hence a variable read-out time is achieved by this parameterization. This is because the higher the value of the weighting parameter, the smaller the number of sample points after CVP, and thus the shorter the read-out time. So, in effect, the second term in the cost function provides control over the read-out time which provides the motivation to include this additional term in the cost function. The gradient constraints in this case will still be satisfied as the velocity chosen for parameterization is feasible and the curve is smooth without sharp edges. Fig. 1 shows the effect of varying \( \lambda \) for a 4098 point TSP trajectory with initial points taken from the density \( \pi \) as before. For larger values of \( \lambda \), the consecutive points come closer to each other, resulting in a reduction of the overall length.
of the trajectory. Subsequently, the additional sample points introduced by CVP will also be less for larger \( \lambda \). Also note that with increasing \( \lambda \), the trajectories tend to become smoother.

(b) Spline Interpolation-based Projection (SIP) method

This variation uses spline interpolation instead of CVP to increase the number of sample points in the trajectory. First, we solve (12) with \( A = I \) (\( n = m \)).

\[
(P_{\text{SIP}}): \min_{s \in \mathbb{R}^m} \frac{1}{2} ||s - c||_2^2. \tag{16}
\]

The main difference between this variation and the projection method is that in (9), instead of \( c \), a parameterized \( c \) is used. In this method, we introduce new points in the trajectory or resample the trajectory by interpolating the trajectory obtained instead of interpolating the trajectory before projection. This reduces the computational complexity and allows us to choose the interpolation factor by doing multiple simulations without having to project the trajectory for each trial. Hence, interpolating the trajectory after projection reduces the complexity overall. Note that although the trajectory \( s \) obtained from (16) is feasible and can be traversed, the number of sample points is not enough to result in a good reconstruction. Although linear interpolation is generally used for reparameterization, here we use splines for interpolation as they provide piece-wise polynomial interpolation [45]. Although any spline could be used, we have used cubic splines as they are smooth (\( C^2 \) continuous). We rewrite the 2D trajectory \( s \in \mathbb{R}^{2m} \) as a complex trajectory, denoted as \( s_c \in \mathbb{C}^m \). The trajectory \( s_c \) can be assumed to be sampled from a trajectory \( s_i(p), p \in [0, m - 1] \) such that \( \{s_i\} = s_c(i), i = 0, 1, 2, ..., m - 1 \) where \( \{s_i\} \) are the original points from (16) and \( s_c(i) \) are uniformly sampled points on \( s_i(p) \). We want to approximate \( s_i(p) \) by piece-wise cubic polynomial curves, i.e., between every two consecutive points \( i, i + 1 \), we have \( s_i(p) = P_i^3(p), p \in [i, i + 1] \) where \( P_i^3(p) = c_{i, 0} + c_{i, 1} p + c_{i, 2} p^2 + c_{i, 3} p^3 \) with \( c_{i, k} \in \mathbb{C} \), \( k \in \{0, 1, 2, 3\} \). Interpolation is done by evaluating the trajectory locations using the obtained piece-wise polynomials at points finer than the given \( i \)'s depending on the oversampling factor (OSF). The interpolated trajectory is rewritten as a \( 2m \cdot \text{OSF} \) length vector by concatenating the real and imaginary parts, denoted as \( s_{4m} \in \mathbb{R}^{2m \cdot \text{OSF}} \). Such interpolation using cubic splines provides enough sample points to ensure good image recovery. This also provides another advantage of being able to control the read-out time by varying the interpolation factor. The formulation of the SIP method in (16) is mathematically similar to the one of the projection method in the discrete domain.

In case we use the SIP method with a random-like input, certain permutations of the points might result in a better performance than others. Hence, the method can be modified to permute the sequence of points until a permutation with a desired reconstruction quality is obtained. Note that this does not have to be the permutation leading to the TSP sequence in order to perform well. This also holds for all the methods described previously.

2.4.4. A designed as a permutation matrix

When \( c \) is a TSP solution (denoted as \( c_{\text{TSP}} \)) to the randomly sampled points, we can consider it as a special permutation \( (\Pi_{\text{TSP}}) \) of the sampled points such that the path covers the shortest distance. This may or may not be a feasible trajectory. Thus, we propose to find a different permutation matrix \( \Pi \in \mathbb{R}^{m \times m} \) that is not necessarily a TSP solution but leads to a feasible trajectory. The idea is to obtain a random-like trajectory by alternatively optimizing over \( I \) and \( s \). This is done by first initializing \( c \) by randomly sampling points from the density function \( \pi \propto 1/|k|^2 \) [27,46]. Then we initialize \( s \) similar to \( c \) by randomly sampling points from the same density function \( \pi \) and then find a permutation matrix \( \Pi \) such that points in \( c \) that are close to points in \( s \) are paired together. By doing this, a feasible \( s \) is obtained using the resultant \( \Pi \). This is repeated until updates of \( s \) converge (\( e = ||s_i - s_{i-1}||^2 < \epsilon \), where it is the current iteration and \( \epsilon \) is a small positive number). The related problem can be written as

\[
s_{\in \mathbb{R}^m} \min_{\Pi^{(2)} \in \mathbb{R}^{m \times m}} \prod_{i} \prod_{l=1}^{2m} \prod_{j=l}^{l+1} \left( \frac{1}{2} ||s - \Pi^{(2)} c||_2^2 \right)
\]

subject to \( \Pi^{(2)} = 1 \) \( \Pi^{(2)} \) \( \Pi^{(2)} \)

Note that (17) is convex in \( s \) for fixed \( \Pi^{(2)} \) as it is the projection of \( \Pi^{(2)} c \) onto the convex set \( \mathbb{R}^m \). On the other hand, although for fixed \( s \) the problem is not convex in \( \Pi^{(2)} \) due to the Boolean constraints in the entries of \( \Pi^{(2)} \), the solution for \( \Pi^{(2)} \) can be obtained efficiently by solving a bipartite matching problem [47]. This method will be called the projection with permutation (PP) method henceforth.
2.5. Experiments

We discuss the methods used to test and compare the performance of the various methods discussed previously in the following subsections.

2.5.1. k-space data estimation

The k-space data for an image is generated by taking the non-uniform Fourier transform of the image at the trajectory points. The non-uniform fast Fourier transform (NUFFT) is used because the k-space points do not necessarily lie on the Cartesian grid. The NUFFT is implemented using Fessler’s Michigan image reconstruction toolbox (MIRT) [48]. It is assumed that the data thus obtained is the undersampled data from an MRI machine and is then used to reconstruct the image as described below.

2.5.2. Reconstruction of image

An MRI image is sparse in various transform domains such as the wavelet domain, frequency domain or discrete cosine transform (DCT) domain [12]. Let $X$ be the desired image to be estimated. Then, by CS theory, the image can be non-linearly reconstructed from a sparsely and incoherently sampled k-space using, for example, [12]:

$$\hat{X} = \arg\min_X \|\text{NUFFT}(X) - Y\|_2^2 + \lambda_1 \|\mathcal{W}(X)\|_1 + \lambda_2 \|X\|_1$$

(18)

where $Y$ is the observed k-space data, $\mathcal{W}(\cdot)$ is the wavelet transform and $\|\cdot\|_1$ is the total variation (TV) norm:

$$\|X\|_1 = \sum_j \sum_i \sqrt{|X_{ij+1} - X_{ij}|^2 + |X_{ij+1} - X_{ij}|^2}$$

(19)

Numerous sparsity penalties have been used in MRI such as wavelets and finite differences. Note that (18) is a convex optimization problem which can be solved using a non-linear conjugate gradient method with a fast and cheap backtracking line-search [12,49].

2.5.3. Performance metrics

To compare the reconstruction performances of different methods, we use structural similarity index (SSIM) and peak signal-to-noise ratio (PSNR) as measures. SSIM provides a comparison in perception of the two images using local statistics over windows $\hat{X}$ and $X$ of $\hat{X}$ and $X$, respectively as

$$\text{SSIM}(X, \hat{X}) = \frac{(2\mu_X\mu_{\hat{X}} + c_1)(2\sigma_{X\hat{X}} + c_2)}{\mu_X^2 + \mu_{\hat{X}}^2 + c_1(\sigma_X^2 + \sigma_{\hat{X}}^2 + c_2)}$$

(20)

where $\mu_X, \mu_{\hat{X}}, \sigma_X, \sigma_{\hat{X}}, \sigma_{X\hat{X}}$ are the local means, standard deviations, and cross-covariance. $c_1 = 10^{-4}$ and $c_2 = 9 \times 10^{-4}$ are constants. The SSIM of the whole image is calculated by taking the mean of SSIMs over all windows. PSNR is a measure of the error in intensity values and is given by

$$\text{PSNR} = 10\log_{10} \frac{(\max X(i,j))^2}{\frac{1}{N} \sum_{i,j} (X(i,j) - \hat{X}(i,j))^2}$$

(21)

where, $\text{MSE} = \frac{1}{N} \sum_{i,j} (X(i,j) - \hat{X}(i,j))^2$. Mean SSIM and mean PSNR are calculated from 100 different trials for each method for better comparison.

2.5.4. $T_2$ decay

$T_2$ decay is the decay of the transverse magnetization ($M_x$) due to interactions among the close by spins. This is an exponential decay with time-constant $T_2$ which depends on the tissue. Long read-out times result in blurring and loss of SNR due to $T_2$ decay [50]. Since most of the trajectories discussed here have long read-out times, we study the effect of $T_2$ decay by comparing the amplitude loss (AL) of the peak of the point spread function (PSF) [50]. To obtain the PSF, the NUFFT $S(k_x, k_y)$ of a 256 $\times$ 256 point image $f(x,y)$ is calculated first using trajectories by different methods. The PSF is then obtained by

$$\text{PSF} = \text{NUFFT}^{-1}(S(k_x, k_y)),$$

where NUFFT$^{-1}$ is the inverse NUFFT operator. To include the effect of $T_2$ decay, PSF$_{\text{decay}}$ is calculated by weighting the k-space signal $S(k_x, k_y)$ by the exponential function $e^{-t/T_2}$, $0 \leq t \leq T$ and then taking the inverse NUFFT.

$$\text{PSF}_{\text{decay}} = \text{NUFFT}^{-1}(S(k_x, k_y)e^{-t/T_2}), \quad 0 \leq t \leq T,$$

where $T$ is the total read-out time. The percentage amplitude loss is then calculated as

$$\text{AL} = \frac{\max(\text{PSF}) - \max(\text{PSF}_{\text{decay}})}{\max(\text{PSF})} \times 100\%.$$

2.5.5. Algorithm to solve the general projection problem

A proximal gradient descent based iterative algorithm is used to solve the general projection problem (11) and is described in Appendix C. The algorithm

2.5.6. Simulation framework

In this section, we test the performance of the methods discussed in the previous sections for the reconstruction of a $128 \times 128$ Shepp-Logan phantom, a $256 \times 256$ realistic analytical head phantom [51] and a T1-weighted sagittal brain MRI image (obtained using Cartesian imaging) as shown in Fig. 2. All the simulations are performed in

![Fig. 2. Reference images (256 $\times$ 256) for simulation results: (a) realistic analytical phantom, (b) an MRI image.](image-url)
MATLAB 2018b in a 64 bit UBUNTU system using an Intel Core i5-4460 CPU, with 16 GB RAM. The test images are made complex by adding complex Gaussian random noise. The gradient constraints are taken as $G_{\text{max}} = 40 \text{mT/m}$ and $S_{\text{max}} = 150 \text{mT/m/s}$ to be consistent with the TOC and projection methods. The sampling frequency is taken to be 250 kHz. The variable density function $\pi \propto 1/|k|^2$ is used for all trajectories. The k-space data for the test images is then obtained by using NUFFT [48] along the feasible trajectories. The results are computed over multiple trials (each trial consists of a new selection of reference curve $c$). We define the sampling factor in the same way as in [36] as $N^2/m \times 100\%$. Here, $N$ corresponds to the image size and $m$ is the number of sample points in the k-space trajectory. The initial reference trajectory $c$ in the TOC and projection methods are taken such that the sampling factor is about 50%.

3. Results and discussion

In this section, we compare and discuss the performance of the proposed methods for different test images.

3.1. Simulation results

3.1.1. 128 × 128 Shepp-Logan phantom

Simulation results of all methods are summarized in Table 1. For the TOC method, $c$ is considered a 400 point TSP curve to get a sampling factor of about 50%. For the projection-CVP method, $c$ is a TSP curve with $128 \times 128/4 = 4096$ points. The same initial trajectory $c$ is used for the projection-CAP, COLT-TSP and SIP-TSP methods. A velocity of $0.5v_{\text{max}}$ and an acceleration of $0.3a_{\text{max}}$ are used for parameterizing $c$ to get a sampling factor of about 50% for the projection-CVP and projection-CAP methods, respectively. The projection-CAP provides a better reconstruction quality compared to its CVP counterpart. An improvement of about 0.1951 in SSIM is observed on average. In case of the COLT-TSP method, with increasing velocity $v$ of the parameterization and $\lambda$, the mean read-out time reduces and so does the reconstruction performance as shown in Fig. 4(a). A decline in performance is expected since the number of sample points in the trajectory reduces. As can be observed from the figure, the read-out time reduces drastically with $\alpha$ but the decline in performance is not as drastic. Hence, this method can be used as a method to reduce read-out time with $\lambda = 1$. The same is observed for the COLT-random method as well, however, the COLT-random method provides a better reconstruction compared to COLT-TSP. A mean improvement of about 0.2757 in SSIM is observed with a 50% reduction in read-out time over the COLT-TSP method. Similarly, the SIP-random provides a much better reconstruction performance than the SIP-TSP with an improvement of 0.2545 in SSIM. The GBP methods (Type-1 and Type-2) are used to obtain TSP-based and random-like trajectories with $8192 = 128 \times 128/2$ points. In both types of methods, random-like trajectories perform better than TSP-based trajectories. For random-like trajectories, the GBP (Type-2) method performs better than the GBP (Type-1) method. For TSP-based trajectories, the two types of GBP methods show very similar performance. The PP method with 5461 points outperforms all other methods with just 21.84 ms read-out time. In Fig. 3, the simulation results for one trial of all methods are shown. The feasible TSP-based trajectories for these methods are shown in Fig. 3(a), (b), (c), (d), (e), (h) and (j). Trajectories obtained from the COLT, SIP, GBP (Type-1) and GBP (Type-2) methods when random points are used as $c$ are shown in Fig. 3(f), (g), (i) and (k), respectively. The trajectory from the PP method is shown in Fig. 3(l).

3.1.2. 256 × 256 analytical phantom and brain MRI images

For higher resolution images, we compare the methods for single-shot and multi-shot (2-shot and 4-shot) trajectories. Multi-shot trajectories are constructed only for TSP-based trajectories. To obtain the multi-shot trajectories, we divide the points sampled from the density function $\pi \propto 1/|k|^2$ into two halves and four quadrants for 2-shot and 4-shot trajectories, respectively with some overlap between the regions. A TSP-based trajectory is obtained for each region and a corresponding feasible TSP-based trajectory is to be traversed in one RF excitation. The performance of all methods is summarized in Table 2 (also see Fig. 6). For the TOC method, 2500 points are sampled on the k-space such that the feasible single-shot trajectory takes a read-out time of about 133 ms. For projection-CVP, projection-CAP, COLT-TSP and SIP-TSP methods, 16384 (256 × 256/4) points are sampled to get $c$. The velocity and acceleration used for the parameterization are taken as $0.5v_{\text{max}}$, $0.25v_{\text{max}}$ and $0.8a_{\text{max}}$ for the projection-CVP, COLT-TSP and projection-CAP methods, respectively. The projection-CAP method performs better than the projection-CVP method and comparable with the TOC method. Similar to the lower resolution phantom image, random-like trajectories using the COLT and SIP methods perform better than the TSP-based trajectories obtained by these methods. The GBP method is used to create 21845 (256 × 256/3) point TSP-based and random-like trajectories. GBP (Type-1) random-like trajectories perform comparable to the GBP (Type-2) TSP-based trajectories on average with a read-out time of nearly 87 ms. The PP method provides a good reconstruction with an SSIM of 0.8191 and a significantly lower read-out time of nearly 43 ms in a single shot.

Fig. 4(b) compares the performance of the projection-CVP and COLT-TSP methods for the 256 × 256 Shepp-Logan phantom image reconstruction with the same TSP-based trajectories as discussed above. The mean read-out times, PSNRs and SSIMs are plotted as functions of the fraction $\alpha$ of $v_{\text{max}}$ used for CVP in both methods. The variation with respect to $\lambda$ in the COLT-TSP method is also depicted. For lower values of $\alpha$ ($< 0.5$), the projection method performs really well with SSIMs greater than 0.9 and high PSNRs. However, these values are achieved at very high read-out times. The effect of $\alpha$ is not observed to be as prominent for the COLT-TSP method. For a similar mean PSNR of 29dB, the read-out time for the COLT-TSP method is 37.60 ms as compared to 137.13 ms for the projection-CVP method.

3.1.3. 256 × 256 phantom MRI images

Fully sampled k-space data using Cartesian trajectory was acquired by scanning a spherical (10496625) and a cylindrical (06824186) Siemens phantom placed side-by-side on a 3T Siemens Prisma machine. The scan was done by placing the arrangement inside a 20 channel head coil. A single image is obtained by combining the reconstructed images from each channel by root-sum-of-squares (RSS) method. In the RSS method, firstly, inverse FT is applied to k-space data of each coil and then the individual images are combined as

| Table 1 | Mean read-out time and SSIM over 100 trials for the 128 × 128 phantom image reconstruction using different methods under a single-shot scheme to obtain feasible trajectories. |
|---------|-----------------|-----------------|-----------------|
| Method  | Read-out time (ms) | Sampling factor | SSIM | PSNR (dB) |
| TOC     | 33.11            | 50.50%          | 0.7607 | 25.46 |
| Projection-CVP | 33.95          | 51.80%          | 0.6613 | 28.25 |
| Projection-CAP | (c = 0.5v_{\text{max}}) |               | 0.8764 | 32.20 |
| COLT-TSP (v = 0.2v_{\text{max}}) | 16.76          | 25.57%          | 0.5620 | 22.70 |
| COLT-random | 15.46          | 23.57%          | 0.8377 | 27.43 |
| SIP-TSP (OSF = 2) | 32.77          | 50%             | 0.6726 | 25.81 |
| SIP-random (OSF = 2) | 32.77          | 50%             | 0.9271 | 32.22 |
| GBP (Type-1) = random | 32.77          | 50%             | 0.9113 | 29.25 |
| GBP (Type-2) = TSP | 32.77          | 50%             | 0.8845 | 34.08 |
| GBP (Type-2) = random | 32.77          | 50%             | 0.9062 | 28.96 |
| PP method | 21.84          | 33.33%          | 0.9942 | 42.54 |
where $X_{\text{rss}}$ is the image reconstructed using the $i^{th}$ coil and $n_c$ is the total number of coils (here, $n_c = 20$). We obtain $256 \times 256$ images of resolution $1 \times 1 \times 4$ mm using TSE sequence with TR/TE of 600/6.4 ms. We use the single reconstructed image $X_{\text{rss}}$ to get the $k$-space data at the proposed non-Cartesian trajectories and reconstruct the images back using (18). The two images considered here are shown in Fig. 5. Fig. 5: (a) is a sagittal image of the arrangement of the two phantoms and (b) is a coronal image. The trajectories used here are the same as the ones used for $256 \times 256$ images previously for the analytical phantom and brain MRI images. The results are summarized in Table 3.

### 3.2. Robustness of methods

To test the robustness of the methods discussed in this paper, we compare the histograms of SSIMs of the reconstructed $256 \times 256$ analytical realistic brain phantom image over 100 trials for different reference curves $c$ as shown in Fig. 7. It is observed that the TSP-based trajectories obtained using the TOC and GBP (Type-2) methods and random-like trajectories obtained using COLT, SIP and PP methods provide consistent performances over trials. SSIMs from other methods vary a lot where some trials result in a really poor performance with an SSIM less than 0.5 and some have a really good performance with SSIM greater than 0.95. Similar to most methods that require an initialization, the proposed methods are also susceptible to such initializations, such as the sample points on the $k$-space, choice of $c$ from the sampled points (in random-like trajectory case). Since selection of a trajectory is done offline, it is advised to do multiple simulations and finally choose a trajectory that leads to the best reconstruction performance on the test images: phantom and MRI of different anatomical structures.

### 3.3. Effect of $T_2$ decay

Table 4 shows the percentage AL for $T_2 = 90$ ms and 400 ms (values typical for gray matter and cerebrospinal fluid, respectively at 1.5T field strength [1]) for trajectories obtained by different methods. Methods resulting in shorter trajectories suffer less from $T_2$ decay. Hence, proposed methods like COLT, SIP, GBP and PP with shorter read-out times are expected to be robust to the effects of $T_2$ decay, especially in single-shot trajectories.

### 3.4. Discussion

The problem of finding short yet feasible trajectories for MRI sampling has been addressed here. The results indicate that random-like trajectories are better than TSP-based trajectories under the proposed projection-based method. For $128 \times 128$ images, the PP method is observed to be better than other existing and proposed methods in terms of reconstruction performance and read-out time. This method is suggested to be used in lower resolution scans. For $256 \times 256$ images, the PP method provides the shortest trajectories that can be implemented in practice. Although COLT-random and SIP-random provide better reconstruction than other proposed methods, these cannot be implemented in the multi-shot scenario. The single-shot scheme of COLT-random and SIP-random may not be practical in all cases because of longer read-out times. The same is true for the GBP methods. Hence, for high resolution single-shot imaging, the PP method and COLT-random methods are recommended to be used if the read-out time can

---

**Fig. 3.** Comparison of performance of TOC, projection-CVP, projection-CAP, COLT, SIP, GBP and PP method for $128 \times 128$ phantom image. $\lambda = 1$ for COLT method. OSF = 1.5 for SIP method.

**Fig. 6.** Trajectories and reconstructed $256 \times 256$ analytical realistic brain phantom and brain MRI images for the TOC, projection-CVP, projection-CAP, COLT, SIP, GBP and PP methods.
be close to 40 ms and 60 ms, respectively. In the multi-shot scheme, the 4-shot TSP trajectory with TOC method performs the best.

A drawback of the TSP and random-like trajectories from projection-based methods is that the sampled points do not follow the density function. This is solved using the PP method which provides a trajectory with points close to the original density. The better performance of the PP method compared to other existing and proposed methods proves that the TSP-based trajectory is not necessarily the best trajectory to follow the density.

The proposed trajectories have not yet been tested on an MRI scanner. Since the trajectories are designed to satisfy the physical constraints and are feasible, they should be implementable in a scanner without any difficulty. Since the trajectories are smooth, there should not be very loud acoustic noise. Some possible sources of image artifacts might occur due to eddy currents and deviation from the desired trajectory which are common in non-Cartesian trajectories. There could also be motion-related errors in case of trajectories with longer read-out times. Various methods exist in literature to address these potential issues [38,52].

4. Conclusion

We focus on a recent method for obtaining feasible trajectories which uses the projection of a CVP curve onto a convex set and we propose a generalization of this method to obtain faster trajectories. We have discussed some special instances of this generalized method starting from some TSP-based and random-like trajectories. We have proposed a new parameterization scheme based on constant acceleration which is observed to perform better than CVP. Using the COLT, SIP, GBP and PP methods, we have shown the effectiveness of random-like trajectories in reducing the read-out time while maintaining a good reconstruction performance. The general framework opens many more possibilities for trajectory design. The considered methods consist of solving a constrained convex optimization problem for which an iterative algorithm in the dual space has been provided. They also give the designer the freedom to choose from various design parameters (the weighting parameter, OSF) in order to tune the trade-off between read-out time and reconstruction performance. The choice of method in practice would be determined according to the application at hand. The trajectories designed here are feasible under the gradient constraints and hence can be implemented in a real scanner. However, the results discussed here are preliminary and may vary in practice due to imprecisions in trajectories because of eddy currents and other scanner related errors. Faster acquisition using the proposed methods would be useful in applications such as dynamic imaging, cardiac imaging, etc.

Acknowledgment

The authors would like to thank the Ministry of Electronics and Information Technology (MeitY), India for the financial support to conduct this work. We are also grateful to the IISc-TU Delft collaboration agreement.

Appendix A. Design of the CVP operator

As discussed in Section 2.2.2, the CVP matrix is the product of an interpolation matrix \(L\) and a selection matrix \(\Pi_m\). The design method for the two matrices is described below.

**Interpolation matrix \(L\):**

Let \(x \in \mathbb{R}^N\) be a vector of \(n\) points \(x_1, x_2, \ldots, x_N\). To interpolate this by a factor of \(N\), i.e., to have \(N\) points between all two consecutive points, we design the interpolated points

\[
x_{i,j} = \frac{j}{N} (x_{i+1} - x_i) + x_i
\]

where \(x_{i,j}\) denotes the \((j+1)^{th}\) interpolated point between \(x_i\) and \(x_{i+1}\), \(1 \leq i \leq N - 1, 0 \leq j \leq N - 1\). This operation on all points of \(x\) can be written in matrix form as

\[
x_{ij} = \frac{j}{N} (x_{i+1} - x_i) + x_i
\]
Fig. 4. Effect of velocity and $\lambda$ in COLT-TSP method for (a) 128 × 128 and (b) 256 × 256 Shepp-Logan phantom and its comparison with projection-CVP method.

Fig. 5. (a) Phantom MRI image-1: Sagittal image and (b) Phantom MRI image-2: Coronal image.
where $D \in \mathbb{R}^{N \times N} = I \otimes \text{diag}(0,1,2,\ldots,N-1)$, and $I_E \in \mathbb{R}^{nN \times n} = I \otimes 1_N$. We denote $c_L = L_2 c \in \mathbb{R}^{2nN}$, where $L_2 = \text{blkdiag}_2(L)$. Selection matrix, $\prod_{\alpha} c$: As discussed in Section 2.1, consecutive points in $c$ are assumed to be traversed in time $t_s$. Since the distance between points is not the same, this results in varying velocity through the trajectory. To achieve CVP, we need the distance between consecutive points to be the same. To do this, let us introduce $t_i$ as the time taken to traverse two consecutive points in $c_L$ in terms of instantaneous velocity, which is given by $t_i = \frac{l_i - l_j}{v_{\text{max}}}$, $1 \leq i \leq (n-1)N - 1$ (A.3)

where $l_i = \sum_{j=1}^{i} ||c_{L,i+1} - c_{L,j}||_2$ as defined in (5) with $c_{L,j} = [(c_{L,j})_1 (c_{L,j})_2]^T$. Points from $c_L$ are then chosen using a selection matrix $\Pi_{\alpha} c$ such that the total time between consecutive selected points is $t_s$. For this, we define the set $\mathcal{S} = \{j: t_{j+1} - t_j \approx t_s, 1 \leq j \leq n - 1\}$ and the selection matrix $\Pi_{\alpha} c = [e_1 \ 0 \ e_2 \ 0 \ e_3 \ 0 \ \ldots \ e_m]$ with $e_i$'s at the column positions given by the set $\mathcal{S}$. Here, $\{e_i\}_{i=1}^m$ are the columns of an $m \times m$ identity matrix.

The procedure to obtain $\Pi_{\alpha} c$ is summarized in Algorithm 1.

Algorithm 1: Algorithm to construct CVP matrix $A_\alpha(c,a)$
Input: $c$, $a$, $N$
Result: $A_\alpha(a,c) = \Pi_{\alpha} c$

1. Construct $L = \frac{1}{N} (D_0 + I_E = I \otimes \text{diag}(0,1,2,\ldots,N-1)$.
2. Construct $c_L = Lc$.
3. Construct $t_i = \frac{l_i - l_j}{v_{\text{max}}}$, where $l_i = \sum_{j=1}^{i} ||c_{L,i+1} - c_{L,j}||_2$, $1 \leq i \leq (n-1)N$.
4. Obtain set $\mathcal{S} = \{j: t_{j+1} - t_j \approx t_s, 1 \leq j \leq n - 1\}$.
5. Construct $\Pi_{\alpha} c$ as a selection matrix to select indices in $\mathcal{S}$.

---

Table 3
Mean read-out time and SSIM over 100 trials for the $256 \times 256$ phantom MRI image reconstruction using different methods under single-shot scheme to obtain feasible trajectories.

<table>
<thead>
<tr>
<th>Method</th>
<th>Read-out time (ms)</th>
<th>MRI image-1 SSIM</th>
<th>MRI image-1 PSNR</th>
<th>MRI image-2 SSIM</th>
<th>MRI image-2 PSNR</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOC</td>
<td>133.35</td>
<td>0.9589</td>
<td>37.54</td>
<td>0.9611</td>
<td>39.13</td>
</tr>
<tr>
<td>Projection-CVP $(v = 0.5v_{\text{max}})$</td>
<td>137.13</td>
<td>0.6826</td>
<td>26.35</td>
<td>0.6608</td>
<td>24.49</td>
</tr>
<tr>
<td>Projection-CAP $(a = 0.4a_{\text{max}})$</td>
<td>136.05</td>
<td>0.8871</td>
<td>34.99</td>
<td>0.8884</td>
<td>35.32</td>
</tr>
<tr>
<td>COLT-TSP $(v = 0.25v_{\text{max}})$</td>
<td>75.21</td>
<td>0.7720</td>
<td>30.10</td>
<td>0.7601</td>
<td>28.60</td>
</tr>
<tr>
<td>SIP-TSP $(OSF = 1.5)$</td>
<td>98.30</td>
<td>0.7813</td>
<td>30.21</td>
<td>0.7680</td>
<td>28.78</td>
</tr>
<tr>
<td>COLT-random $(v = v_{\text{max}})$</td>
<td>61.59</td>
<td>0.9022</td>
<td>33.50</td>
<td>0.8994</td>
<td>33.87</td>
</tr>
<tr>
<td>SIP-random $(OSF = 1.5)$</td>
<td>98.30</td>
<td>0.9510</td>
<td>35.11</td>
<td>0.9493</td>
<td>35.61</td>
</tr>
<tr>
<td>GBP (Type-1) $(c = TSP)$</td>
<td>87.38</td>
<td>0.8240</td>
<td>32.05</td>
<td>0.7986</td>
<td>30.31</td>
</tr>
<tr>
<td>GBP (Type-1) $(c = random)$</td>
<td>87.38</td>
<td>0.8482</td>
<td>32.11</td>
<td>0.8508</td>
<td>32.22</td>
</tr>
<tr>
<td>GBP (Type-2) $(c = TSP)$</td>
<td>87.38</td>
<td>0.9118</td>
<td>34.77</td>
<td>0.9093</td>
<td>34.71</td>
</tr>
<tr>
<td>GBP (Type-2) $(c = random)$</td>
<td>87.38</td>
<td>0.7333</td>
<td>28.49</td>
<td>0.7204</td>
<td>27.31</td>
</tr>
<tr>
<td>PP</td>
<td>43.69</td>
<td>0.8932</td>
<td>33.57</td>
<td>0.9056</td>
<td>34.12</td>
</tr>
</tbody>
</table>

Table 4
Effect of $T_2$ decay.

<table>
<thead>
<tr>
<th>Method</th>
<th>$T_2 = 90$ ms</th>
<th>$T_2 = 400$ ms</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOC</td>
<td>47.85 %</td>
<td>14.96 %</td>
</tr>
<tr>
<td>Projection-CVP</td>
<td>48.67 %</td>
<td>15.34 %</td>
</tr>
<tr>
<td>Projection-CAP</td>
<td>48.44 %</td>
<td>15.23 %</td>
</tr>
<tr>
<td>COLT-TSP</td>
<td>32.22 %</td>
<td>8.84 %</td>
</tr>
<tr>
<td>COLT-random</td>
<td>27.58 %</td>
<td>7.32 %</td>
</tr>
<tr>
<td>SIP</td>
<td>39.16 %</td>
<td>11.34 %</td>
</tr>
<tr>
<td>GBP</td>
<td>36.01 %</td>
<td>10.17 %</td>
</tr>
<tr>
<td>PP</td>
<td>30.22 %</td>
<td>5.27 %</td>
</tr>
</tbody>
</table>

Fig. 7. Histograms with bin size of 0.05 of SSIMs of $256 \times 256$ analytical realistic brain phantom image using different methods.
Appendix B. Design of the CAP operator

The first step for the construction of the linear CAP operator is the same as that for the CVP operator, i.e., interpolation of $c$ using $L$ as given in (A.2). The difference is in the second step, i.e., in picking the points from the interpolated $c$ using the selection matrix $\Pi_\beta$. To obtain $\Pi_\beta$, we do the following. First, find the time taken between consecutive points in $c_\ell = Lc$ assuming acceleration $a = \beta a_{\text{max}}$, where $\beta$ is the fraction of the maximum acceleration $a_{\text{max}} = \gamma S_{\text{max}}$ needed for the parameterization. This results in

$$t_i = \frac{\|s_{i+1} - s_i\|}{\beta a_{\text{max}}}, \quad 1 \leq i \leq (n-1)N - 1$$

(B.1)

where $s_i = \frac{s(\ell + 1) - s(\ell)}{t_i}$ as defined in (6) with $c_\ell = \{c(\ell), [c(\ell)]\}$T. Then, a set $\mathcal{F}$ is defined as before. Finally, define $\Pi_\beta, e$ as the selection matrix $\Pi_\beta$. $e = \{e_1, 0, e_2, 0, e_3, 0 \ldots e_m\}$ with $e_i$s at the column positions given by the set $\mathcal{F}$. This procedure is summarized in Algorithm 2.

Algorithm 2: Algorithm to construct CAP matrix $A_\beta(c, e)$

Input: $c, e, N$
Result: $A_\beta(c, e) = \Pi_\beta c$

1. Construct $L = \frac{1}{n} D L D_1 + L_0$.
2. Construct $c_\ell = Lc$.
3. Construct $t_i = \frac{\|s_{i+1} - s_i\|}{\beta a_{\text{max}}}, \quad 1 \leq i \leq (n-1)N - 1$.
4. Obtain set $\mathcal{F} = \{t_j, t_j - t_i \leq j \leq i \leq n - 1\}$.
5. Construct $\Pi_\beta, e$ as a selection matrix to select indices in $\mathcal{F}$.

Appendix C. Algorithm to solve the general projection problem

An iterative algorithm based on proximal gradient descent on the dual is used to solve the following general projection problem.

$$\min_{s \in \mathbb{R}^m} \frac{1}{2} \|s - A^Tc\|^2$$

subject to $\|D(s)\|_\infty \leq \gamma G_{\text{max}}$

$$\|D(s)\|_\infty \leq \gamma S_{\text{max}}$$

(C.1)

where $G_{\text{max}}$, $S_{\text{max}}$, and $c$ are defined. $\Pi_\beta, e$ is as given in (A.2).

To evaluate $F(q_1, q_2)$, we need to find

$$s^\ast(q_1, q_2) = \arg \min_{s \in \mathbb{R}^m} \frac{1}{2} \|s - A_{\beta}c\|^2 + \|D(s)q_1\| + \|D(s)q_2\|$$

(C.4)

which has the closed form solution

$$s^\ast(q_1, q_2) = A_{\beta}c - D_{\beta}^Tq_1 - D_{\beta}^Tq_2$$

(C.5)

With this, the proximal gradient descent algorithm to solve (C.3) is summarized below in Algorithm 3. Here, $L = \|D_1^T s_\gamma D_1 + D_2^T D_2\|$ is the Lipschitz constant of $VF(q_1, q_2)$. The algorithm converges at a rate $O(1/t^2)$ for a fixed step size $t \in (0, 1/L)$ [53, 54].

Algorithm 3:

Input: $c \in \mathbb{R}^m, \lambda > 0, \epsilon > 0, L, \alpha, \beta$
1. Initialization: $q_1^{(0)} = [q_1^{(0)}, q_2^{(0)}]^T = 0, y^{(0)} = q^{(0)}$, $t = 1/L$, $e = c$, $k = 0$
2. while $\|e\|^2 < \epsilon$

1. Obtain set $\mathcal{F} = \{t_j, t_j - t_i \leq j \leq i \leq n - 1\}$.
2. Construct $\Pi_\beta, e$ as a selection matrix to select indices in $\mathcal{F}$.

1. Set $e' = \Pi_\beta e$.
2. Set $e'' = I - e'$.
3. Set $c' = \Pi_\beta c$.
4. Set $c'' = I - c'$.
5. Set $\gamma = \gamma c''$.
6. Find $s^\ast(q_1, q_2)$.
7. Set $q_1 = q_1 + \alpha (s^\ast(q_1, q_2) - q_1)$.
8. Set $q_2 = q_2 + \alpha (s^\ast(q_1, q_2) - q_2)$.
3. \( k = k + 1 \)

4. \( z^{(k)} = z^{(k-1)} + \gamma V \mathbf{F}^T y^{(k-1)} \)

5. \( q^{(k)} = \arg \min \frac{1}{2} ||z^{(k)} - q||^2 + Q(q) \)

6. \( y^{(k)} = q^{(k)} - \frac{1}{\tau}(q^{(k)} - q^{(k-1)}) \)

7. \( e = q^{(k)} - q^{(k-2)} \)

8. end

Output: \( s = s^{(3)}(k) \)

References


