Discrete Spherical Harmonic Oscillator Transforms on the Cartesian Grids using Transformation Coefficients

Soo-Chang Pei, Fellow, IEEE, and Chun-Lin Liu, Student Member, IEEE

Abstract—The analog harmonic oscillators are well-studied in quantum physics, including their energy states, wavefunctions, orthogonal properties, and eigenfunctions of the Fourier transform. In addition, the continuous solutions in different dimension and coordinate systems are known. Some discrete equivalents of the 1D wavefunctions were also studied. However, in the 3D spherical coordinate system, the discrete equivalents of the 3D wavefunctions are not established. In this paper, we focus on the spherical harmonic oscillator wavefunctions (SHOWs) the spherical harmonic oscillator transforms (SHOTs), and their discrete implementation. The SHOWs can be synthesized by linear combinations of the Hermite Gaussian functions with proper transformation coefficients. We find that computing the coefficients can be speeded up using the fast Fourier transforms or some recursive relations. These coefficients relate the Hermite transforms with the SHOTs. Some applications of the discrete SHOWs and the discrete SHOTs are introduced. First, the SHOWs are exactly the eigenfunctions of the 3D DFT. Also, the SHOTs can be used to derive the spherical harmonic oscillator descriptor (SHOTD), which is a rotational invariant descriptor. We find that the SHOD is not only compatible with the existing rotational descriptors for the spherically sampled data but also outperforms the existing rotational descriptors for 3D Cartesian sampled, bandlimited input data. Besides, the SHOTs can be used to decompose 3D signals into spherical components. Hence, 3D signal reconstruction is done using partially chosen spherical component, and 3D data compression for MRI data is demonstrated using SHOWs for medical applications.

Index Terms—Harmonic analysis, Multidimensional signal processing, Quantum harmonic oscillators, Fourier transforms, Fast Fourier transforms, Rotational invariant descriptors.

I. INTRODUCTION

The harmonic oscillator plays an important role in quantum mechanics because it is quite simple, elegant, and has close-form solutions. In the 1D case, with quadratic potential function, the harmonic oscillator wavefunctions are the Hermite Gaussian functions [1]. For a given potential function $V(x)$, one can use the Taylor series expansion to approximate $V(x)$ using the second-order polynomial. Therefore, lots of complicated problems are approximated by the harmonic oscillator. This method is quite successful in studying the behavior of nuclei and other problems [1], [2].

In signal processing, the harmonic oscillator wavefunctions are also widely used. In the 1D case, the Hermite Gaussian functions are used in signal analysis such as signal decomposition or signal representation [3], [4]. They are also the eigenfunctions of the Fourier transform [5], [6]. The fractional Fourier transform can be defined with the aid of the Hermite Gaussian functions [5].

In higher dimensions, the harmonic oscillator wavefunctions have different forms. For the Cartesian coordinate, the solution is the separable Hermite Gaussian functions. However, in the 2D polar coordinate, one possible solution is the Laguerre Gaussian functions, which are well-known in the study of optical beam modes or lasers [7], [8]. They possess the circular shape of disks or rings in the 2D plane. In the 3D spherical coordinate system, the corresponding harmonic oscillator wavefunctions are called 3D spherical harmonic oscillator wavefunctions (SHOWs) in this paper. The SHOWs are more complicated than the separable Hermite Gaussian functions. The SHOWs are composed of the radial function and the spherical harmonics in the $\theta$- and $\phi$-direction. In the field of signal processing, it is widely accepted that the spherical harmonics already have applications in 3D model retrieval [9]–[12], 3D SIFT in medical image analysis [13], [14], 3D orientation determination [15], inverse rendering [16], and 3D surround sound systems [17].

Finding the discrete equivalents of these functions is a great issue. The following two basic requirements should be met. Firstly, he discrete functions converge to the continuous when the number of discrete points approaches to infinity. In addition, the discrete functions should be orthogonal and complete as in the continuous case. The 1D discrete Hermite Gaussian functions have been studied extensively in the recent years [18]–[20]. The Kravchuk functions were found to be the discrete correspondents of the Hermite Gaussian functions [20]–[23]. Alternatively, in [19], [24], [25], the discrete functions were obtained numerically from the commuting matrix of the discrete Fourier transform (DFT). This method not only satisfies the two basic requirements but also yields perfect eigenvectors of the DFT. Hence, we adopt the latter method as the 1D discrete Hermite Gaussian functions.

For the 2D Laguerre Gaussian functions or the 3D SHOWs, we can synthesize them by the linear combination [26], as long as we have the coefficients. In optics, the concept of
mode conversion indicates that the separable Hermite Gaussian functions and the Laguerre Gaussian functions are associated with proper transformation coefficients [27]–[29]. As a result, we can construct the discrete separable Hermite Gaussian functions and use particular transformation coefficients to have the discrete Laguerre Gaussian functions [30], [31]. The obtained discrete Laguerre Gaussian functions are theoretically orthogonal. When it comes to the 3D case, there are lots of papers and books on the transformation coefficients of the SHOWs [2], [32]–[35]. With the aid of the transformation coefficients, we can implement the discrete SHOWs and define a new transform, called the spherical harmonic transform (SHOT), using the SHOWs as the transform kernel.

We have three main contributions to the discrete implementation of the SHOWs and the SHOTs. First of all, we proposed a novel method to compute the transformation coefficients efficiently using the fast Fourier transform algorithm. Secondly, the implementation of the discrete SHOWs and the discrete SHOTs is introduced. With the aid of the transformation coefficients, we can implement the discrete SHOWs and the discrete SHOTs on the Cartesian grids, where most 3D volume data are represented.

Our third contribution is to apply the discrete SHOWs and the discrete SHOTs to the field of signal processing and pattern recognition. Firstly, the SHOWs are the eigenfunctions of the 3D Fourier transform. In the discrete domain, the discrete SHOWs are also the eigenfunctions of the 3D DFT. Then, a rotational invariant feature can be derived from the SHOT, which is called the spherical harmonic oscillator descriptor (SHOD). Rotational invariant descriptors for 3D objects are quite useful in detecting the same object with different rotation angle, which is an important topic in 3D object retrieval [9]–[12]. Furthermore, signal expansion and reconstruction are realized by selecting dominant SHOTs and performing the inverse transform. The reconstructed signal enables us to analyze the signal information in the spherical coordinate. Finally, signal compression is also achieved with the concept of signal expansion and reconstruction. We only select dominant SHOTs and omit the other negligible SHOTs. The data size can be reduced and compressed.

This paper is organized as follows: Some preliminaries, mathematical notations, the concept of creation operators are reviewed in Section II. The properties of the transformation coefficients are discussed in Section III. Our contributions are the contents after Section IV, where the fast algorithm for the transformation coefficients is presented. With the transformation coefficients, discrete implementation for the SHOWs and the SHOTs is introduced in Section V. In Section VI, we show many applications for the discrete SHOWs and the discrete SHOTs. Several simulation results are shown in Section VII and Section VIII concludes this paper.

II. PRELIMINARIES

A. Bracket Notation and Signal Processing Notation

The 1D time-independent Schrödinger equation in the classical form is

$$-rac{\hbar^2}{2m} \frac{d^2}{dx^2} + V(x) \psi(x) = E \psi(x),$$  \hspace{1cm} (1)

where \( \hbar \) is the reduced Planck’s constant, \( m \) is the mass of the particle, \( x \) is the position, \( V(x) \) is the potential function of the system, \( E \) is the total energy, and \( \psi(x) \) is the wavefunction of the quantum state. The wavefunction \( \psi(x) \) describes the probability density of the particle. The bracket notation is also a convenient notation of the quantum states [1], [36]. This notation represents a state \( \psi \) as the ket notation |\( \psi \rangle \rangle \), which is equivalent to a column vector in signal processing. Dirac further defined the bra notation, \( \langle \psi | \)\), which is identical to \( \psi \) in the quantum state. The wavefunction in the 3D case is also written as

$$\langle \langle x | \psi \rangle \rangle = |\psi\rangle \langle \psi| \psi \rangle,$$

where |\( \psi \rangle \rangle \) can be viewed as the eigenvectors of the operator in the parentheses with eigenvalue \( E \). Although the operator on the left-hand side is defined with respect to \( x \) and |\( \psi \rangle \rangle \) is the states vector, which is not a function of \( x \), it is customary to write the Schrödinger equation into (2) when the brackets are used.

In the 3D case, the position operator becomes 3D and the resultant generalized eigenvector is indexed by the position vector \( r \). Note that the position operator can be defined in different coordinate systems. In the Cartesian coordinate system, \( r = (x, y, z) \) while in the spherical coordinate system, \( r = (r, \theta, \phi) \). The generalized eigenvectors are written as \( |r(x, y, z)\rangle \) or \( |r(r, \theta, \phi)\rangle \), depending on the coordinate system. As a result, in this paper, the wavefunction in the 3D case is written as \( \langle r(x, y, z)|\psi\rangle \) or \( \langle r(r, \theta, \phi)|\psi\rangle \) in the bracket notation, which is identical to \( \psi(x, y, z) \) or \( \psi(r, \theta, \phi) \) in the signal processing notation, respectively. The wavefunctions in both brackets and signal processing are summarized in Table I as a reference for the reader.

B. The Harmonic Oscillator System in Different Dimensions

The 1D harmonic oscillator wavefunctions are the solution to the time-independent Schrödinger equation, (1), with the quadratic potential \( mw^2x^2/2 \). For simplicity, the mass of the particle \( m \), the oscillator frequency \( \omega \) and the reduced Planck constant \( \hbar \) are normalized. Therefore, we obtain

$$\left[ \frac{d^2}{dx^2} - x^2 \right] |n_x\rangle = -(2n_x + 1) |n_x\rangle,$$  \hspace{1cm} (3)

This paper is organized as follows: Some preliminaries, mathematical notations, the concept of creation operators are reviewed in Section II. The properties of the transformation coefficients are discussed in Section III. Our contributions are the contents after Section IV, where the fast algorithm for the transformation coefficients is presented. With the transformation coefficients, discrete implementation for the SHOWs and the SHOTs is introduced in Section V. In Section VI, we show many applications for the discrete SHOWs and the discrete SHOTs. Several simulation results are shown in Section VII and Section VIII concludes this paper.
TABLE 1
THE BRACKET NOTATION AND THE CORRESPONDENTS IN SIGNAL PROCESSING

<table>
<thead>
<tr>
<th>Notation</th>
<th>Brackets</th>
<th>Signal processing</th>
</tr>
</thead>
<tbody>
<tr>
<td>State vector, ket</td>
<td>$</td>
<td>\psi\rangle$</td>
</tr>
<tr>
<td>State vector, bra</td>
<td>$\langle \psi</td>
<td>$</td>
</tr>
<tr>
<td>Inner product</td>
<td>$\langle \psi</td>
<td>\psi \rangle$</td>
</tr>
<tr>
<td>Outer product</td>
<td>$</td>
<td>\psi\rangle \langle \psi'</td>
</tr>
<tr>
<td>Operator $O$ on $</td>
<td>\psi\rangle$</td>
<td>$O</td>
</tr>
<tr>
<td>1D wavefunctions</td>
<td>$\langle x</td>
<td>\psi\rangle$</td>
</tr>
<tr>
<td>Hermite Gaussian functions</td>
<td>$</td>
<td>\psi\rangle$</td>
</tr>
<tr>
<td>3D wavefunctions in Cartesian coordinates</td>
<td>$\langle r(x, y, z)</td>
<td>\psi\rangle$</td>
</tr>
<tr>
<td>3D wavefunctions in spherical coordinates</td>
<td>$\langle r(r, \theta, \phi)</td>
<td>\psi\rangle$</td>
</tr>
<tr>
<td>3D Hermite Gaussian functions</td>
<td>$\langle \tau(x, y, z)</td>
<td>n_x n_y n_z\rangle</td>
</tr>
</tbody>
</table>
| 3D spherical harmonic oscillator wavefunctions | $\langle \tau(r, \theta, \phi) |m\rangle |\psi\rangle$ | \(1\)

where the states are characterized to the non-negative integer $n_x$. The corresponding state vector is expressed as $|n_x\rangle$.

**Definition 1.** The wavefunction of (3) is the Hermite Gaussian functions, which are

$$
\langle x|n_x\rangle = h_{n_x}(x) = N_{n_x} H_{n_x}(x) e^{-x^2/2},
$$

(4)

where $N_{n_x} = (2^{n_x} \pi^{1/4})^{-1/2}$ is the normalization constant and $H_{n_x}(x) = e^{x^2}(-d/dx)^{n_x} e^{-x^2}$ are the Hermite polynomials. The index $n_x$ is a non-negative integer.

The wavefunction is then $\langle x|n_x\rangle$, which is the same as $h_{n_x}(x)$ in signal processing. The Hermite Gaussian functions form a complete and orthonormal basis for $L^2(\mathbb{R})$. With the brackets, the completeness of the basis can be written as $\sum_{x=0}^\infty \langle x|n_x\rangle \langle n_x|x\rangle = 1$, where $1$ is the unit operator. The orthogonality is also denoted by $\langle n'_x|n_x\rangle = \delta_{n'_x,n_x}$.

**Definition 2.** For a 1D signal $f(x) \in L^2(\mathbb{R})$, the Hermitian transform of $f(x)$ is

$$
\mathcal{H} f = a_{n_x} = \langle x|f\rangle = \int_\mathbb{R} \langle x|n_x\rangle f(x) \, dx = \int_\mathbb{R} h_{n_x}(x) f(x) \, dx,
$$

(5)

where $n_x \in \{0, 1, 2, \ldots\} = \mathbb{N}_0$ and $\mathcal{H}$ is the Hermitian transform operator.

Due to the orthonormal relation, the inverse Hermite transforms are easily obtained from the linear combination of $\langle x|n_x\rangle = h_{n_x}(x)$. As the result, the inverse Hermite transforms are

$$
f(x) = \sum_{n_x=0}^\infty \langle x|n_x\rangle \langle n_x|f\rangle = \sum_{n_x=0}^\infty \langle x|n_x\rangle a_{n_x}.
$$

(6)

The Fourier transform is another important transform in signal processing.

$$
1^L(\mathbb{R}) = \{ f(x) \mid \int_{-\infty}^\infty |f(x)|^2 \, dx < \infty \}
$$

means the set of all functions with finite energy.

**Definition 3.** The Fourier transform maps a $L^2(\mathbb{R})$ signal $f(x)$ to $(\mathcal{F} f)(u)$, where $u \in \mathbb{R}$, according to

$$
(\mathcal{F} f)(u) = \int_\mathbb{R} f(x) e^{-jux} (2\pi)^{1/2} \, dx,
$$

(7)

where $\mathcal{F}$ denotes the Fourier transform operator and $j = \sqrt{-1}$ is the imaginary unit.

**Property 1.** Hermite Gaussian functions $h_{n_x}(x) = \langle x|n_x\rangle$ are the eigenfunctions of the Fourier transforms with eigenvalues $(-j)^{n_x}$. That is, $\mathcal{F} |n_x\rangle = (-j)^{n_x} |n_x\rangle$.

**Property 2.** For the input signal $f(x)$, its Fourier transform can be eigen-decomposed to be

$$
(\mathcal{F} f)(u) = \sum_{n_x=0}^\infty (-j)^{n_x} |u|n_x\rangle a_{n_x},
$$

(8)

where $a_{n_x} = \langle n_x|f\rangle$ is the Hermite transform of $f(x)$ and $|u|n_x\rangle = h_{n_x}(u)$ is the Hermite Gaussian function. An operator form is that $\mathcal{F} = \mathcal{H}^{-1} D_{n_x} \mathcal{H}$ with $D_{n_x} |n_x\rangle = e^{-jn_x\alpha} |n_x\rangle$. $D_{n_x}$ is a diagonal matrix.

The fractional Fourier transform is derived from changing the eigenvalues $(-j)^{n_x}$ into $e^{-j\alpha n_x}$ with respect to a fractional angle $\alpha$ [5]. We can define the fractional Fourier transform by defining a new operator $\mathcal{F}_\alpha$.

**Definition 4.** Define the fractional Fourier transform operator $\mathcal{F}_\alpha$ by $\mathcal{F}_\alpha = \mathcal{H}^{-1} D_\alpha \mathcal{H}$.

The operator $\mathcal{F}_\alpha$ reduces to $\mathcal{F}$ when $\alpha = \pi/2$. More properties of the fractional Fourier transform were studied in [5], [6].

For the higher dimensional case, the ordinary differential equation (3) becomes the partial differential equation and we can solve it in different coordinates. In the Cartesian coordinate system $(x, y, z)$, the solution is the 3D separable Hermite Gaussian functions.

**Definition 5.** The 3D separable Hermite Gaussian functions $\langle \tau(x, y, z)|n_x n_y n_z\rangle$ are $\langle \tau(x, y, z)|x_n y_n z_n\rangle = \langle x|n_x\rangle \langle y|n_y\rangle \langle z|n_z\rangle$. 


\( \langle x, y, z \rangle |n_x n_y n_z \rangle \) are a complete and orthonormal basis for \( L^2(\mathbb{R}^3) \) because it is a direct extension of the 1D case.

In this paper, we focus on the 3D spherical coordinate system, where the position is specified by \((r, \theta, \phi)\). The conversion between \((x, y, z)\) and \((r, \theta, \phi)\) is
\[
x = r \sin \theta \cos \phi, \quad y = r \sin \theta \sin \phi, \quad z = r \cos \theta, \quad (9)
\]
where \( r \in \mathbb{R}^+ \), \( \theta \in [0, \pi] \), and \( \phi \in [0, 2\pi] \).

The Schrödinger equation in terms of state vectors becomes
\[
(\nabla^2 - r^2) |nlm\rangle = -[2(2n + l + 3)] |nlm\rangle \quad (10)
\]
where \( \nabla^2 \) is the Laplacian and \( |nlm\rangle \) is the state vector. The wavefunction in this case is written as \( \langle r, \theta, \phi |nlm\rangle \). Here the ket “\( |\rangle \) is modified into the round bracket “\( \rangle \),” as used in quantum mechanics, to avoid confusion when the numerical values are put in the brackets. (10) can be decomposed into three equations in the \( r-, \theta-, \) and \( \phi\)-direction, respectively. The details can be referred to [2], [37].

**Definition 6.** The wavefunction \( \langle r, \theta, \phi |nlm\rangle \), called the spherical harmonic oscillator wavefunctions (SHOWs) or the 3D isotropic harmonic oscillator wavefunctions, are
\[
\langle r, \theta, \phi |nlm\rangle = N_{nl} r^{l+1/2} (2^r e^{-r^2/2} Y_{lm}(\theta, \phi)), \quad (11)
\]
where \( N_{nl} \) is the normalization factor, \( L_r^{l+1/2} (\cdot) \) is the generalized Laguerre polynomial, and \( Y_{lm}(\theta, \phi) \) is the spherical harmonics, defined by
\[
Y_{lm}(\theta, \phi) = \sqrt{\frac{2l + 1}{4\pi} \frac{(l-m)!}{(l+m)!}} P_{l}^{m}(\cos \theta) e^{im\phi}. \quad (12)
\]

\( P_{l}^{m}(\cdot) \) is the associated Legendre polynomial. \( n, l \) are non-negative integers while \( m \), magnetic quantum number, is still a integer ranged from \(-l\) to \( l \).

**Property 3.** The SHOWs are a complete and orthonormal basis for \( L^2(\mathbb{R}^3) \).

**Property 4.** \( \langle r, \theta, \phi |nl0\rangle \) is real.

**Property 5.** \( \langle r, \theta, \phi |nl - m \rangle = (-1)^m \langle r, \theta, \phi |nlm\rangle^* \), where \( * \) denotes complex conjugates.

The above properties are trivial under Definition 6.

Similar to the Hermite transforms, we can define the spherical harmonic oscillator transform (SHOT) on a 3D signal.

**Definition 7.** For a 3D signal \( f(r) \in L^2(\mathbb{R}^3) \), the spherical harmonic oscillator transform (SHOT) is
\[
|nlm\rangle = \int f(r) \langle r, \theta, \phi |nlm\rangle^* d^3 r, \quad (13)
\]
where \( d^3 r = r^2 \sin \theta \, dr \, d\theta \, d\phi \) in the spherical coordinate.

By Definition 7 and Property 3, the inverse spherical harmonic oscillator transforms become
\[
f(r) = \sum_{n,l,m} \sum_{s=0}^l N^{(1)}_{n,l,m} (\eta_x^2 + \eta_y^2 + \eta_z^2)^s (\eta_x^s \eta_y^s \eta_z^s) |000\rangle \quad (14)
\]
where \( \langle r, \theta, \phi |nlm\rangle \) is the SHOW and \( |nlm\rangle \) is the SHOT of \( f(r) \).

The 3D Fourier transform is also related to the SHOWs as its eigenfunctions. Here we still define the 3D Fourier transform \( \mathcal{F} \), whose transform kernel is the product of the Fourier transform kernels in each dimension. \( \langle r, \theta, \phi |n_x n_y n_z\rangle \) are the eigenfunctions of \( \mathcal{F} \) because its integral kernel is separable. If the coordinate system is changed to the spherical case, the SHOWs are also the eigenfunctions. We have the eigen-equation for the SHOWs
\[
\mathcal{F} |nlm\rangle = (-j)^{n+l} |nlm\rangle. \quad (15)
\]

The creation operator is another method in quantum mechanics rather than solving the differential equation directly. The concept is that any state is created from the proper operator working on the ground state. Here we follow the notation and derivation in [34]. For 1D harmonic oscillator wavefunctions, it is
\[
|n_x\rangle = (n_x l)^{-1/2} \eta_x^{n_x} |0\rangle, \quad (16)
\]
where \( n_x \), or a more popular notation \( a_x^+ \), is the raising operator and the operator \( (n_x l)^{-1/2} \eta_x^{n_x} \) is often called the creation operator because it creates the state \( |n_x\rangle \) from the ground state \( |0\rangle \). From (16), it is observed that the state \( |n_x\rangle \) is raised to \( |n_x + 1\rangle \) by the raising operator \( n_x \) due to
\[
n_x |n_x\rangle = \sqrt{n_x + 1} |n_x + 1\rangle. \quad (17)
\]

The creation operator for the 3D separable Hermite Gaussian functions is trivial. It is
\[
|n_x n_y n_z\rangle = (n_x l n_y n_z l)^{-1/2} \eta_x^{n_x} \eta_y^{n_y} \eta_z^{n_z} |000\rangle, \quad (18)
\]
where \( n_y \) and \( n_z \) are creation operators in the \( y\) and \( z\)-axis, respectively. The creation operator for the SHOWs is not as simple as the 3D separable Hermite Gaussian functions. The creation operator, (3.2) in [34], for the SHOWs when \( m > 0 \) is
\[
|nlm\rangle = \sum_{s=0}^l N^{(2)}_{s,l,m} (\eta_x^s + \eta_y^s + \eta_z^s)^s \eta_x^{l-2s} \eta_y^{m-2s} \eta_z^{n-2s-m} |000\rangle, \quad (19)
\]
\[
N^{(1)}_{n,l,m} = (-1)^{n+m} \left[ \frac{(2l + 1)(n + l)!((l-m)!)^{1/2}}{2^n l!(2n + 2l + 1)!(l+m)!} \right]^{1/2}, \quad (20)
\]
\[
N^{(2)}_{s,l,m} = \frac{(-1)^s (2l - 2s)!}{s! (l-s)! (l-2s-m)!}. \quad (21)
\]

Expanding (19) and using (18) yield the following theorem:

**Theorem II.1.** There exists a unique set of transformation coefficient \( C^{n,l,m}_{n_x n_y n_z} \) such that
\[
|nlm\rangle = \sum_{n_x, n_y, n_z} C^{n,l,m}_{n_x n_y n_z} |n_x n_y n_z\rangle, \quad (22)
\]
where the summation has index variables, \( n_x, n_y, n_z \in \{0, 1, 2, \ldots \} \) with the constraint that \( N = 2n + l = n_x + n_y + n_z \). \( N \) is also called the order of the SHOWs.

In the following discussion, we adopt \( C^{n,l,m}_{n_x n_y n_z} \) as our notation for the transformation coefficients, instead of the bracket notation. Note that Theorem II.1 is consistent with our previous work on the general form of 2D Fourier transform.
eigenfunctions [26]. The complicated close-form transformation brackets are first derived in [33]. The result is, however, very lengthy and has double summations. In addition, in the close-form solution, each transformation coefficient needs a lot of computation.

III. PROPERTIES OF THE TRANSFORMATION COEFFICIENTS AND THE SHOWS

Now we relate the SHOWs to the 3D separable Hermite Gaussian functions with a large number of coefficients. In this section, we seek for a more elegant arrangement for the coefficients, instead of six-parameter notation $C_{n_x,n_y,n_z}^{l,m}$. Let’s start from (22) and set $N = 2n + l = n_x + n_y + n_z = 1$ for all $n, l, m$ as an example. Enumerating all possible SHOWs and 3D separable Hermite Gaussian functions gives us $(N + 1)(N + 2)/2$ distinct SHOWs and 3D separable Hermite Gaussian functions. We rearrange these functions into a vector and then (22) is rewritten as

$$
\begin{bmatrix}
|011\rangle \\
|010\rangle \\
|01−1\rangle \\
|nlm\rangle
\end{bmatrix} =
\begin{bmatrix}
0.0, 1.0 & 0.0, 1.0 & 0.0, 1.0 & 0.0, 1.0 \\
0.0, 1.0 & 0.0, 1.0 & 0.0, 1.0 & 0.0, 1.0 \\
0.0, 1.0 & 0.0, 1.0 & 0.0, 1.0 & 0.0, 1.0 \\
0.0, 1.0 & 0.0, 1.0 & 0.0, 1.0 & 0.0, 1.0
\end{bmatrix}
\begin{bmatrix}
|001\rangle \\
|010\rangle \\
|100\rangle \\
|n_x,n_y,n_z\rangle
\end{bmatrix},
$$

which is much more elegant than (22). The 3-by-3 squared matrix is denoted by $T_1$. The order of the vectors formed by $|n_x,n_y,n_z\rangle$ and $|nlm\rangle$ can be arbitrarily chosen as long as the corresponding entries are correct. For instance, we can write the vector on the right-hand side to be $[[100], [010], [001]]^T$ and permute the elements in $T_1$ as long as the matrix form satisfying (22). To avoid this ambiguity, the vectors formed by $|n_x,n_y,n_z\rangle$ and $|nlm\rangle$ should be clearly defined to make $T_1$ unique. Because (22) involves the 3D Hermite Gaussian functions and the SHOWs with the same order, we must separate the coefficients with different order $N$ to different matrices. Therefore, we add a subscript $N$ to indicate the order. In this paper, the definition of $T_N$ is

**Definition 8.** The transformation matrix $T_N$ relates $|nlm\rangle$ with $|n_x,n_y,n_z\rangle$ for $N = 2n + l = n_x + n_y + n_z$ and $T_N$ is an $\frac{1}{2}(N + 1)(N + 2)$-dimensional squared matrix. The entries, $C_{n_x,n_y,n_z}^{l,m}$, are sorted by the rule:

1) For all of the 3D Hermite Gaussian functions with $N$, sort the columns of $T_N$ by $n_x$ in the ascending order. For the same $n_x$, sort the 3D Hermite Gaussian functions by $n_y$ in the ascending order.

2) For all of the SHOWs with $N$, sort the rows of $T_N$ by $n$ in the ascending order. For the same $n$, sort the SHOWs by $m$ in the descending order.

**Property 6.** $T_N$ is an unitary matrix.

**Property 7.** $C_{n_x,n_y,n_z}^{l,m} = (-1)^m C_{n_x,n_y,n_z}^{l,m}$ for $m \geq 0$.

The above two properties are already known [38].

In this paper, we proposed a useful property for the SHOT expansion and reconstruction.

**Theorem III.1.** For the same order $N$, the result for the inverse SHOT in the spherical coordinate and the 3D inverse Hermite transform in the Cartesian coordinate is identical. That is,

$$
\sum |nlm\rangle \langle nlm|f = \sum |n_x,n_y,n_z\rangle \langle n_x,n_y,n_z|f,
$$

where the left-hand side summation involves $n, l, m$ with $2n + l = N$ while the right-hand side summation is indexed by $n_x, n_y, n_z$ with $n_x + n_y + n_z = N$.

**Proof:** Starting from the left-hand side of (24) and using (22) for $|nlm\rangle$ and $|nlm\rangle$ yields

$$
\sum (n_x',n_y',n_z') \sum (C_{n_x',n_y',n_z'}^{l,m})^* C_{n_x,n_y,n_z}^{l,m} |n_x,n_y,n_z\rangle
= \sum (n_x',n_y',n_z') \delta_{n_x'-n_x,n_y'-n_y,n_z'-n_z} |n_x,n_y,n_z\rangle.
$$

The above equality in (25) is due to Property 6. Then, the contribution of $n_x', n_y', n_z'$ is then cancelled and (25) becomes the right-hand side of (24).

Theorem III.1 is meaningful and important in signal processing. The left-hand side summation for the spherical coordinate represents the inverse SHOT for the certain order $N$. On the right-hand side, the summation for the Cartesian coordinate is related to the inverse Hermite transform for the same order $N$. If the input $f(r)$ is represented in the Cartesian coordinate, the right-hand side is more efficient for implementation because the transform kernel is separable. As a result, for the same order $N$, instead of evaluating the left-hand side directly, we can compute the right-hand side to speed up the whole process. This theorem provides an efficient approach to compression of 3D volume data.

In the study of nuclear physics, the transformation matrices are tabulated for low-order cases, as in [33], [35], [38]. However, it lacks efficient computation methods for the transformation matrices.

IV. EFFICIENT CONSTRUCTION OF THE SHOW TRANSFORMATION COEFFICIENTS

A. For one SHOW

In this section, we propose an algorithm to speed up the coefficient computation. The previous works are actually to expand (19) with brute-force. Nowadays, we all agree that the FFT can accelerate the polynomial expansion.

In order to calculate the transformation coefficients, the creation operator $(\eta_x, \eta_y, \eta_z)$ can be viewed as usual variables $(X, Y, Z)$, respectively. The overall creation operator for the SHOWs is now considered a polynomial of variables $(X, Y, Z)$, say $p(X, Y, Z)$. According to (18) and the right-hand side of (22), $p(X, Y, Z)$ is written as

$$
\sum (n_x,n_y,n_z)!^{-1/2} C_{n_x, n_y, n_z}^{l,m} X^{n_x} Y^{n_y} Z^{n_z}.
$$

Let $X, Y, Z$ to be the twiddle factors

$$
X = e^{-j 2\pi x_k / N}, \quad Y = e^{-j 2\pi y_k / N}, \quad Z = 1,
$$

where $k_x, k_y = 0, 1, \ldots, N$. Substituting (27) into (26) yields

$$
\sum_{n_x, n_y \in N_0} p_1 C_{n_x, n_y}^{l,m} X^{n_x - n_y} Y^{n_y} e^{-j2(\frac{2\pi n_x + n_y}{N+1})}.
$$
where \( p_1 = (n_xn_y(N - n_x - n_y))^ {1/2} \). Obviously, (28) is the 2D \((N + 1)\)-by-\((N + 1)\) discrete Fourier transform of the 2D discrete signal

\[
c_{n_x,n_y} = \left\{ \begin{array}{ll}
p_1C_{n_x,n_y,N-n_x-n_y} & n_x + n_y \leq N, \\
0 & n_x + n_y > N,
\end{array} \right.
\]

where \( n_x, n_y = 0, 1, \ldots, N \). The mapping relation between \( c_{n_x,n_y} \) and \( C_{n_x,n_y,m,n} \) is visualized in Fig. 1. \( C_{n_x,n_y,m,n} \) is a 3D array, arranged into a cube. However, if the homogeneous condition \( (n_x + n_y + n_z = N) \) is added, \( C_{n_x,n_y,m,n} \) is confined in a plane, as the boxes with thick lines in Fig. 1. (29) indicated that \( c_{n_x,n_y} \) is obtained from the projection of \( C_{n_x,n_y,m,n} \) onto the \( n_xn_y \) plane, as marked in Fig. 1.

On the other hand, from (19) and (27), \( p(X,Y,Z) \) is

\[
p(X,Y,Z) = N_{n,l,m}^{(1)}D_{k_x,k_y}^{m} \sum_{s=0}^{[(l-m)/2]} N_{s,l,m}^{(2)}E_{k_x,k_y}^{n+s}
\]

\[D_{k_x,k_y} = e^{-j \frac{2\pi k_x}{N}} + je^{-j \frac{2\pi k_y}{N}},\]

\[E_{k_x,k_y} = 1 + e^{-j \frac{2\pi k_x}{N}} + e^{-j \frac{2\pi k_y}{N}}.
\]

From the signal processing point of view, (30) can be regarded as two multiplications in the frequency domain, indexed by \((k_x,k_y)\). Therefore, in the time domain, \( c_{n_x,n_y} \) is regarded as the convolution, associated with the two different signals \( d_{n_x,n_y} \) and \( e_{n_x,n_y} \), defined by

\[
d_{n_x,n_y} = \delta_{n_x-1,n_y} + j\delta_{n_x,n_y-1},\]

\[
e_{n_x,n_y} = \delta_{n_x,n_y} + \delta_{n_x-2,n_y} + \delta_{n_x,n_y-2},\]

where \( \delta_{n_x,n_y} \) is the 2D delta function. Equalling (28) and (30) and taking the inverse 2D DFT on both sides yield

\[
c_{n_x,n_y} = N_{n,l,m}^{(1)}C_{n_x,n_y,m,n} \sum_{s=0}^{[(l-m)/2]} N_{s,l,m}^{(2)}e^{(n+s)}.
\]

The above discussion is the process to find the coefficient in a systematic way. To sum up, the steps are

1) Establish \( d_{n_x,n_y} \) and \( e_{n_x,n_y} \) by (33) and (34).

2) Apply \( N + 1 \)-point 2D FFT on \( d_{n_x,n_y} \) and \( e_{n_x,n_y} \) to have \( D_{k_x,k_y} \) and \( E_{k_x,k_y} \).

3) For a given \((n,l,m)\), compute \( p(X,Y,Z) \) by (30).

4) Take the inverse 2D FFT on \( p(X,Y,Z) \) to have \( c_{n_x,n_y} \).

5) Relate \( c_{n_x,n_y} \) and \( C_{n_x,n_y,m,n} \) by (29).

**B. For the complete SHOWs**

The method, proposed in Section IV-A, computes the transformation coefficients for a single \((n,l,m)\). However, it is not suitable for the transformation matrices \( T_N \) for all \( N \) because we have to apply FFT for each SHOW and it is not economical. We want to find some recursive formulae to speed up the process, which is useful for the SHOTs.

First, we want to observe the symmetry of \( T_N \). As the sorting rule in Definition 8 states, the first few SHOWs are

\[
\begin{align*}
& n_l = 0, n_m = 0, n_m = 0, \quad N_{n_l,m}^{(1)} = 1, \\
& n_l = 0, n_m = 0, n_m = 0, \quad N_{n_l,m}^{(2)} = 1, \\
& n_l = 0, n_m = 0, n_m = 0, \quad N_{n_l,m}^{(3)} = 1, \\
& n_l = 0, n_m = 0, n_m = 0, \quad N_{n_l,m}^{(4)} = 1.
\end{align*}
\]

However, in this case, (35) is more computationally efficient because only \( e_{n_x,n_y} \) is needed and \( e_{n_x,n_y} \) is a three-point 2D discrete signal, as shown in (34). Besides, \( e_{n_x,n_y}^{(s)} \), \( N_{n,l,m}^{(1)} \), and \( N_{n,l,m}^{(2)} \) for all \( s \) are stored in memory for the recursive purpose.

2) When \( n = 0 \), \( l = N \), \( m = 0 \), we want to derive some recursive relation. From (20) and (21), we have

\[
N_{n,l,m}^{(1)} = -((l + m)(l - m + 1))^{1/2} N_{n,l,m-1}^{(1)}
\]

\[
N_{n,l,m}^{(2)} = (l - 2s - m + 1) N_{s,l,m-1}^{(2)}.
\]

Note that \( e_{n_x,n_y}^{(s)} \) is computed in the \([0N0]\) step. By (35), the coefficient for \([01]\) can be computed from the previous knowledge. Recursively using (36) and (37) gives us the coefficients for \( m > 0 \).

3) For \( n = 0 \), \( l = N \), \( m < 0 \), the coefficients are directly computed from Property 7.

4) For the \( n > 0 \) case, the recursive relation is also simple. Considering \([n - 1l]m \) and \([nlm] \) and using (19), we have

\[
|nlm| = (N_{n-1,l,m}^{(1)}/N_{n-1,l,m}^{(1)})(n_x^2 + n_y^2 + n_z^2) |n - 1l|.
\]

Using (22), we have the recursive relation

\[
C_{n,l,m}^{n,l,m} = (N_{n,l,m}^{(1)}/N_{n-1,l,m}^{(1)})C_{n_x,n_y} \ast C_{n_x-1,n_y}^{n-1,l,m}.
\]

The recursive relation (39) is very useful to synthesis SHOWs from low-order SHOWs. We can compute \( T_N \) from \( T_{N-2} \) with the aid of (39). Besides, (40) is the close-form normalization factor between \([n - 1l]\) and \([nl] \). However, in practice, (40) is not very important because we can compute \( e_{n_x,n_y} \ast C_{n_x,n_y,n_z}^{n,l,m} \) and then normalize it, due to Property 6.

Fig. 2 serves as a summary for the process to generate all matrices \( T_N \) for all \( N \). Based on \( n,l,m \), we found different
V. THE DISCRETE SHOWS AND SHOTS

In this section, the discrete implementation of the SHOWs and the SHOTs is discussed. Our method depends greatly on the transform coefficients \( C_{n_l,n_m,n_z}^{n_l,m,m} \), as in Theorem II.1, which is related to a finite summation with respect to \( n_x, n_y, \) and \( n_z \). However, later, we observe that for the high-order SHOTs, not all summation terms are available. Before the discussion, some terminologies for the discrete SHOWs and the discrete SHOTs are first defined.

Definition 9. The error for the discrete SHOWs |nlm) or the discrete SHOTs \((nlm|f)\) is defined as

\[
E(n_l,m,m) = \sum_{n_x, n_y, n_z} \left| C_{n_x,n_y,n_z}^{n_l,m,m} \right|^2,
\]

where \( N = 2n_l + m \) is the summation index and the summation runs over all unused \( n_x, n_y, n_z \) in implementing the discrete SHOWs or the discrete SHOTs.

If the summation index is the same as Theorem II.1, \( E(n_l,m,m) = 0 \) because \( C_{n_x,n_y,n_z}^{n_l,m,m} \) are used. Those discrete SHOWs or the discrete SHOTs are called “exact” because \( E(n_l,m,m) = 0 \).

In the discrete implementation process, some summation terms are getting removed, which will be discussed later in detail. In this case, \( 0 < E(n_l,m,m) \leq 1 \) and these discrete SHOWs and discrete SHOTs are “inexact” because they are not completely implemented by Theorem II.1. \( E(n_l,m,m) \) serves as a measure to show how much the discrete version is close to the exact form. As smaller \( E(n_l,m,m) \) is closer to 0, the discrete SHOWs or the discrete SHOTs are said to be a good approximation. Larger \( E(n_l,m,m) \) indicates that the discrete SHOWs or the discrete SHOTs deviate the exact result more.

A. The Discrete SHOWs

The discrete SHOW is simple. First, in [19], [24], [25], we can solve the discrete 1D Hermite Gaussian functions and then the 3D separable Hermite Gaussian functions. From Theorem II.1, we can further compute \( C_{n_x,n_y,n_z}^{n_l,m,m} \) for given \( n_l, m, m \) by the method presented in Section IV-A. The combined results are the discrete SHOWs, which are perfectly orthonormal. It is because the discrete Hermite Gaussian functions and the combination coefficients are all orthonormal.

Note that for high-order discrete SHOWs, Theorem II.1 cannot be applied directly and some terms should be eliminated. The details will be discussed and illustrated in the next part.

B. The Discrete SHOTs

The most natural way to implement the SHOTs is to generate the SHOWs for all \((n_l,m,m)\), as defined in (13), and then take the inner-product with \( f(r) \). This method is simple but not very efficient. Different SHOWs have to be computed separately and stored in memory separately because the SHOWs are not separable in the Cartesian coordinates. Assume that the number of discrete point in one-dimension is \( N_p \). To store a SHOW, \( O(N_p^3) \) memory space is required. Besides, there are \( N_p^3 \) SHOWs in total. The overall space needed is \( O(N_p^6) \), which is very large and impractical.

Converting into spherical coordinates and then performing the SHOT in each direction separately seem a possible approach, but interpolation is inevitable because usual discrete signals are sampled over Cartesian grids. However, interpolation harms the signal integrity and makes perfect reconstruction impossible.

We find that the discrete SHOTs can be first decomposed into the 3D Hermite transforms and do the linear combination using the transformation coefficients. To derive this property, we can combine the definition of the SHOT, (13), and the linear combination, (22), and we obtain

\[
(n_l,m,m|f) = \sum \left( C_{n_x,n_y,n_z}^{n_l,m,m} \right)^* \langle n_x,n_y,n_z|f \rangle,
\]

where the right-hand summation is for \( n_x, n_y, n_z \) with \( n_x + n_y + n_z = N \). (42) can be explained as follows. \( \langle n_x,n_y,n_z|f \rangle \) is the 3D Hermite transform of \( f(r) \) and \( (n_l,m,m|f) \) is the SHOT of \( f(r) \). (42) means that the SHOT is obtained by the linear combination of the 3D Hermite transform with the same order \( N \). The combination coefficients are \( \left( C_{n_x,n_y,n_z}^{n_l,m,m} \right)^* \). (42) allows us to implement the SHOT in two steps: First compute the 3D Hermite transform and then combine the Hermite transform with the appropriate coefficients. This is much faster than generating the complete SHOWs. When it comes to the discrete case, all we have to do is to replace the 3D Hermite transform with its discrete version. Note that the 3D discrete Hermite transform is defined on the Cartesian grids so the discrete SHOT is also on the same grids. Hence, if the input discrete signal is defined on the Cartesian grids, the discrete SHOT is exactly on the same grids.

Similar to the example in Section III, we can also rewrite (42) into the matrix form. Take \( N = 2n_l + m \) for an example. For the six SHOTs \((nlm|f)\) with \( N = 2 \), we arrange them into a vector so do the six 3D Hermite transforms \( \langle n_x,n_y,n_z|f \rangle \) with \( N = 2 \). The result is shown in (43). Here the order of the vectors is stated in Definition 8. The matrix connecting the 3D Hermite transform and the SHOT is \( T_2 \), which is the complex conjugate of the transformation matrix in Definition 8. Here we list the entries of \( T_2 \), which can be either computed by the algorithms in Section IV, or looked up in [38].

However, in the discrete case, one major problem is that the high-order coefficients are not available. Here we assume...
that the number of discrete points in the discrete HGFs \( N_{pt} \) is 2, for illustration. There are only two discrete Hermite Gaussian functions, |0\rangle, and |1\rangle, in one dimension. Then, the 3D discrete separable Hermite Gaussian functions are only 8 functions, which correspond to the states |000\rangle, |001\rangle, |010\rangle, |100\rangle, |011\rangle, |101\rangle, |110\rangle, and |111\rangle. The higher-order states, such as |002\rangle, cannot be represented by the discrete Hermite Gaussian functions. Hence we have to modify the implementation steps for the high-order coefficients.

We continue the example for \( N_{pt} = 2 \). We then consider the SHOT and the Hermite transform for \( N = 2 \), which are also listed in (43). In the continuous case, (43) works fine but in the discrete case, we do not have \(|002\rangle\), \(|021\rangle\), and \(|200\rangle\).

It means that the first, the third, and the sixth element of the \(|n,n,n|f\rangle\) vector should be eliminated, or equivalently set to zero. Then, (43) is reduced to

\[
\begin{bmatrix}
|022\rangle_f^\text{inexact} \\
|021\rangle_f^\text{exact} \\
|020\rangle_f^\text{inexact} \\
|02 - 1\rangle_f^\text{inexact} \\
|02 - 2\rangle_f^\text{inexact} \\
|100\rangle_f^\text{inexact}
\end{bmatrix}
= \begin{bmatrix}
0 & 0 & j/\sqrt{2} \\
0 & -j/\sqrt{2} & -1/\sqrt{3} \\
\sqrt{2/3} & 0 & -1/\sqrt{6} \\
0 & -j/\sqrt{2} & 0 \\
0 & 0 & 0 \\
-1/\sqrt{3} & 0 & 0
\end{bmatrix}^* \begin{bmatrix}
|002\rangle_f \\
|011\rangle_f \\
|020\rangle_f \\
|101\rangle_f \\
|110\rangle_f \\
|200\rangle_f
\end{bmatrix},
\]

(43)

After this modification, the transformation matrix is reduced to a 6-by-3 matrix. It is obvious that the lack of high-order Hermite transform leads to inexact discrete SHOTs. Note that the certain SHOTs, such as \(|020\rangle_f\) and \(|100\rangle_f\) in (44), are inexact and always zero and some SHOTs, \(|022\rangle_f\) and \(|02 - 2\rangle_f\) in (44), are inexact but non-zero. Some inexact SHOTs should be removed from the discrete SHOTs.

Our problem then goes to how many number of discrete SHOTs should be kept and which one out of the inexact SHOTs is the good approximation. Because we only keep three discrete Hermite transforms, we should also keep three discrete SHOTs to make the degree of freedom equal. The error for these discrete SHOTs is computed as

\[
E(0, 2, 1) = E(0, 2, -1) = 0,
\]

(45)

\[
E(0, 2, 2) = E(0, 2, -2) = 1/\sqrt{2},
\]

(46)

\[
E(0, 2, 0) = E(1, 0, 0) = 1.
\]

(47)

According to these numerical values, the SHOTs \(|021\rangle_f\) and \(|02 - 1\rangle_f\) should be first kept because they are exact. For those inexact SHOTs, we choose one by one according to their \(E(n, l, m)\) in the ascend order until the degree of freedom constraint is met. Then \(|022\rangle_f\) is chosen for illustration. Here is the example, we choose \(|022\rangle_f\), \(|021\rangle_f\), \(|02 - 1\rangle_f\) and then (44) is modified into

\[
\begin{bmatrix}
|022\rangle_f^\text{inexact} \\
|021\rangle_f^\text{exact} \\
|02 - 1\rangle_f^\text{exact} \\
|02 - 2\rangle_f^\text{inexact} \\
|100\rangle_f^\text{inexact}
\end{bmatrix}
= \begin{bmatrix}
0 & 0 & j/\sqrt{2} \\
0 & -j/\sqrt{2} & -1/\sqrt{3} \\
\sqrt{2/3} & 0 & -1/\sqrt{6} \\
0 & -j/\sqrt{2} & 0 \\
0 & 0 & 0 \\
-1/\sqrt{3} & 0 & 0
\end{bmatrix}^* \begin{bmatrix}
|011\rangle_f \\
|101\rangle_f \\
|110\rangle_f \\
|200\rangle_f
\end{bmatrix}.
\]

(48)

Here we obtain a transformation relationship between the 3D discrete Hermite transforms and the discrete SHOTs, when the order \( N = 2n + l = n_x + n_y + n_z \) exceeds \( N_{pt} - 1 \), where \( N_{pt} \) is the number of discrete points in one dimension. Note that the modified transformation matrix in (48) is no longer unitary.

There are about \( N_{pt}^3/6 \) exact coefficients in our discrete SHOTs. By Definition 8, there are \((N + 1)(N + 2)/2\) SHOWs with order \( N \). We also notice that the exact SHOWs are ranged from \( N = 0 \) to \( N = N_{pt} - 1 \). Then, we have the number of the exact SHOWs,

\[
\sum_{N=0}^{N_{pt}-1} (N + 1)(N + 2) = \frac{N_{pt}(N_{pt} + 1)(N_{pt} + 2)}{6},
\]

(49)

which is approximately \( N_{pt}^3/6 \). In other words, we have \( N_{pt}^3 \) SHOWs in total. and the 1/6 of them are orthogonal and exact. The other coefficients, which contain 5/6 of the total coefficients approximately, are inexact. For instance, when \( N = N_{pt} \), the summation in (42) requires the 3D Hermite transform, \(\langle 00N_{pt}|f\rangle\), which is unavailable due to \( 0 \leq n_x, n_y, n_z \leq N_{pt} - 1 \) and the obtained SHOW is inexact.

The eliminating process can be extended to very large \( N \). We can follow the steps in the example and then extract the submatrix of the original \( T_N \) as the new transformation matrix. First, the columns corresponding the unavailable 3D discrete Hermite transforms are removed, as illustrated from (43) to (44). Then, the row with largest \( E(n, l, m) \) is eliminated until the modified transformation becomes squared, as in (48).

Finally, the error for the high-order inexact discrete SHOWs is discussed. The error for the high-order inexact discrete SHOWs is not our main concern because it is highly dependent on the input signal.

**Theorem V.1.** The error between the high-order exact and inexact SHOWs is

\[
\int |\langle \mathbf{r}|nlm\rangle_{\text{inexact}} - \langle \mathbf{r}|nlm\rangle_{\text{exact}}|^2 d^3\mathbf{r} = E(n, l, m)^2,
\]

(50)

where \(\langle \mathbf{r}|nlm\rangle_{\text{exact}}\) is the continuous exact SHOW using all summation terms in (22), and \(\langle \mathbf{r}|nlm\rangle_{\text{inexact}}\) is the continuous inexact SHOW evaluating the summation terms in the discrete implementation.
Proof: Starting from the error of the exact and the inexact SHOW, we have

\[
\int \left| \langle \mathbf{r} | nlm \rangle_{\text{inexact}} - \langle \mathbf{r} | nlm \rangle_{\text{exact}} \right|^2 \, d^3 \mathbf{r} = \int \left| \sum_{n_z,n_y,n_z, \text{used}} \langle \mathbf{r} | n_x n_y n_z \rangle \langle n_x n_y n_z | nlm \rangle \right|^2 \, d^3 \mathbf{r}
\]

\[
= \sum_{n_x,n_y,n_z, \text{used}} \left| \langle n_x n_y n_z | nlm \rangle \right|^2,
\]

where the summations run through all eliminated \( n_x, n_y, n_z \). According to (41), the error can be written as

\[
\int \left| \langle \mathbf{r} | nlm \rangle_{\text{inexact}} - \langle \mathbf{r} | nlm \rangle_{\text{exact}} \right|^2 \, d^3 \mathbf{r} = E(n,l,m)^2.
\]

Hence, the error of the high-order SHOWs are directly related to \( E(n,l,m) \). The higher \( E(n,l,m) \) is, the larger the basis error is. Our eliminating process first computes \( E(n,l,m) \) for the entire discrete basis and eliminates those with larger \( E(n,l,m) \). This can be explained as those with higher basis error to be removed first.

VI. APPLICATIONS

A. Eigenfunctions of the 3D DFT

From (15), the continuous SHOWs are the eigenfunctions of the 3D Fourier transform. In the discrete point of view, the discrete SHOWs are the eigenfunctions of the 3D DFT, in the spherical coordinates. Because the discrete Hermite Gaussian functions are exactly the eigenfunctions of the 1D DFT, the 3D discrete Hermite Gaussian functions are undoubtedly the eigenfunctions of the 3D DFT, with eigenvalue \((-j)^N\).

With the SHOWs and the SHOTs, one can define the 3D fractional Fourier transform in the spherical coordinates, rather than simply combine the conventional fractional Fourier transform in the Cartesian coordinates separately. The kind of the fractional Fourier transform is more suitable for analyzing the 3D data with spherical symmetry. With the discrete SHOT, the samples are always on the Cartesian grid, which ensures the perfect reconstruction of the original signal.

B. Rotational invariant feature

Finding the rotational invariant feature for the 3D signals is an important topic in pattern recognition. A popular tool is the spherical harmonic transform (SHT), which decompose the surface data \( f(\theta, \phi) \) into the linear combination of the spherical harmonics \( Y_{l,m}(\theta, \phi) \) [9], [39]. In essence, the SHT is a two-dimensional transform because the kernel is merely 2D. To use the SHT for the 3D volume data, one should select a fixed radius and computes the data on the sphere, as suggested in [9]. The corresponding rotational invariant feature is called the spherical harmonic descriptor (SHD) in this paper.

In [40], [41], Wang et al. use the eigenfunctions of the Laplacian operator as the integration kernel and define the spherical Fourier transform (SFT) in 3D. The rotational invariant descriptor, derived from the spherical Fourier transform, is called the spherical Fourier descriptors (SFD).

Although the SHOTs and the SFT have different transform kernels, the only difference between them is the radial components. The radial component for the spherical Fourier transform is the spherical Bessel function while the radial component for the SHOT is shown in (11). The radial function does not affect the rotational invariance [40]. As a result,

\[
\sqrt{\sum_{m=-l}^{l} |\langle nlm | f \rangle|^2}
\]

is also rotational invariant. Note that the SFD is similar to (51). To avoid confusion, (51) is called the spherical harmonic oscillator descriptor (SHOD).

When it is compared with the SHD or the SFD, the main advantage of the SHOD is its discrete implementation. There is no coordinate conversion in the SHOT while the other two transforms involve Cartesian-to-spherical coordinate conversion, which requires interpolation inevitably. Inaccurate interpolations degrade the performance.

C. 3D signal expansion and reconstruction

The SHOT can be used for 3D signal reconstruction in the spherical coordinate system. In quantum mechanics, the three quantum numbers \( n, l, \) and \( m \) have their own meanings so does the total quantum number \( N = 2n + l \).

\( N \) is associated with the total energy of the SHOWs, which is closely connected with the energy compaction. The smaller \( N \) is, the more compact the SHOW is. \( m \) is the spin of the SHOW, which is connected with the variation in the \( \phi \)-direction. More interpretation of the SHOWs is found in [1].

Because the SHOWs have their strong physical meaning, the SHOT is also meaningful. Reconstruction along \( N = 2n + l \) approximates the original signal radially, which can be observed from the parameter \( n \) and \( l \) of the SHOWs. From (11), the radial function of the SHOWs are \( N_n, l r^l L_n^{l+1/2} (r^2) e^{-r^2/2} \). For small \( n \) and \( l \), the radial function is highly compact at the origin. As \( n \) and \( l \) increases, implying larger \( N \), the support of the radial function increases. This means that the SHOTs for small \( n \) and \( l \) contain the information at the origin. If we reconstruct the signal along \( N \), the information is first revealed from the origin and then the support for the reconstructed signal increases gradually.

On the other hand, increasing \( m \) reveals the fine details around the \( \phi \)-direction. This phenomenon can be also interpreted from (11). The function in the \( \phi \)-direction is \( e^{jm\phi} \). For example, \( m = 2 \) stands for the SHOW has two periods in the \( \phi \)-direction. The larger \( m \) we used, the more periods, or the more details are exhibited in the \( \phi \)-direction.

With these physical interpretations, the 3D signal reconstruction along different indices is achieved up to our needs.

D. 3D volume data compression

The signal expansion and reconstruction are applied to not only signal analysis but also data compression, which is important for the huge number of 3D volume data. The SHOT converts the input 3D volume data into a huge set of coefficients, \( \langle nlm | f \rangle \). Selecting partial \( \langle nlm | f \rangle \) and taking the inverse SHOT yield the reconstructed signal, as the previous section does. In terms of compression, the SHOTs \( \langle nlm | f \rangle \) are going to be stored or transmitted rather than the raw data.
VII. SIMULATION RESULTS AND DISCUSSIONS

A. Visualization of the SHOWs

In this simulation, we want to visualize the SHOWs according to the method proposed in this paper. The discrete SHOWs are shown in Table II. Here we show six SHOWs with $N = 2$ and $N_{pt} = 101$. In Table II, the first three rows are the real parts, the imaginary parts, and the magnitudes of the SHOWs. These figures are the isosurface plots, which are the constant value surfaces in space. These constant values are called the isovales. The red surfaces are computed by the isovalue $T$, which is 20% of the maximum value. The green surface is obtained from the isovalue $-T$. The magnitude of the SHOWs is invariant in the $\phi$-direction, which can be explained from the magnitude of (11).

From Table II, we can verify some properties of the SHOWs. It is described in Property 4 that $|nl0|$ is real. Property 5 states the relationship between $|nlm|$ and $|nl - m|$. These properties are all verified in Table II.

B. Eigenfunctions of the 3D DFT

It is mentioned in Section VI-A that the SHOWs are the eigenfunction of the 3D DFT with eigenvalue $(-j)^{3}$. The number of discrete points is set to be $N_{pt} = 101$. For the SHOW, $|nlm| = |154|$, its eigenvalue is $(-j)^{2n+l} = (-j)^{7} = j$, implying that $f_3(r)$ is indeed the eigenfunction of the 3D DFT with its eigenvalue $j$.

C. Rotational invariant feature analysis

As suggested in Section VI-B, the SHD, the SFD, and the SHOD all have rotational invariant properties. However, in discrete implementation, the three different methods are different. Different sampling schemes lead to different performance. In the simulation, we choose six 3D signals

\begin{align*}
    f_1(r) &= \text{sinc}(x)\text{sinc}(y/2)\text{sinc}(2z), \\
    f_2(r) &= \text{sinc}(x')\text{sinc}(y'/2)\text{sinc}(2z'), \\
    f_3(r) &= e^{-(x^2+y^2/4+4z'^2)/\gamma}, \\
    f_4(r) &= \text{sinc}(\|((x - 1)/2, (y - 0.5)/3, (z + 1)/4\|_\infty), \\
    f_5(r) &= \text{sinc}(\|((x' - 1)/2, (y' - 0.5)/3, (z' + 1)/4\|_\infty), \\
    f_6(r) &= e^{-(x'-1)/2, (y'-0.5)/3, z'+1/4\|_\infty},
\end{align*}

where the new coordinate $(x', y', z')$ is obtained from the 3D rotation matrix $R_{y}(\gamma)R_{z}(-\alpha)R_{x}(-\beta)$, $R_{x}$, $R_{y}$, and $R_{z}$ denote rotation matrices about the $x$, $y$, or $z$ axis and $\|x, y, z\|_\infty = \max \{ |x|, |y|, |z| \}$ is the infinity norm. $f_1(r)$ and $f_2(r)$ are bandlimited because their Fourier transforms are composed of three rectangular finite-support functions in each dimension. $f_3(r)$ is in the form of 3D Gaussian function with different variances in each dimension so its Fourier transform is also a Gaussian function. Strictly, Gaussian functions are not bandlimited. However, based on their time-frequency
representations, the Gaussian functions are highly localized and own minimal uncertainty. Hence, the aliasing error for Gaussian functions is negligible. \( f_4(r) \), \( f_5(r) \), and \( f_6(r) \) are non-bandlimited due to the infinity norm. It is also obvious that \( f_2(r) \) is the rotated version of \( f_1(r) \) and \( f_3(r) \) is also the rotated version of \( f_4(r) \). These six 3D volume signals serve as reference signals and the discrete samples are taken directly from the continuous signals. Note that the discrete samples can be sampled either in the Cartesian coordinate or in the spherical coordinate. Here we consider both methods.

1) For the Cartesian coordinate, we select \( N_{pt} = 101 \) in each dimension. The sampling interval \( \Delta x = \Delta y = \Delta z = \sqrt{2\pi}/N_{pt} \). There are approximately \( 10^6 \) samples in total.

2) For the spherical coordinate, the data are sampled uniformly with \( \Delta r = \sqrt{2\pi}/101 \) for \( r \in \mathbb{R}^+ \), \( \Delta \theta = \pi/102 \) for \( \theta \in [0, \pi] \), and \( \Delta \phi = \pi/51 \) for \( \phi \in [0, 2\pi] \), as the sampling scheme proposed in [42]. We have 51 samples in the \( r \)-direction and 102 samples in the \( \theta \)- and \( \phi \)-direction, respectively. The total number of discrete samples is also about \( 5 \times 10^5 \).

The first sampling scheme, in the Cartesian coordinate, extracts the signal information in the cube, centered at the origin while the second sampling method, in the spherical coordinate, takes discrete data from the sphere which inscribes the cube in the first sampling scheme. The number of samples is chosen to match the volume ratio of the cube and the sphere. Therefore, the total number of sphere sampling is approximately half of the Cartesian sampling.

We sweep \( \alpha \in [0, 2\pi] \) and set \( \beta = -\pi/5 \), and \( \gamma = \pi/3 \). For each \( \alpha \), we select 12 descriptors to form the feature vector. For the SHD feature vector, we choose \( r = 1, 2, 3 \) and \( l = 0, 1, 2, 3 \) for the SFD feature vector, descriptors with \( n = 1, 2, 3, 4 \) and \( l = 0, 1, 2 \) are selected; the SHOD whose \( N \leq 5 \) are computed. Assume that the feature vectors for \( f_1(r) \) to \( f_6(r) \) are written as \( f_1 \) to \( f_6 \), respectively.

Among the six signals, we divide them into two groups: bandlimited signals and non-bandlimited signals. In each group, we compute the normalized Euclidean distances between these signals. For the bandlimited signals, \( ||f_i - f_1||/||f_1|| \), for \( i = 2, 3 \), is computed for each angle \( \alpha \). For the non-bandlimited signals, \( ||f_i - f_1||/||f_1|| \), for \( i = 5, 6 \), is then evaluated.

The simulation aims to compare the performance of Cartesian/spherical sampling scheme for bandlimited/non-bandlimited signals, as shown in Fig. 4. We discuss the results in the following three parts.

1) **Errors of the descriptors in the Cartesian coordinates:**
   First consider the input data defined on the Cartesian grids, as shown in Fig. 4(a) and Fig. 4(c). In theory, matched signals have identical feature vectors so the distance between \( f_1 \) and \( f_2 \), and the distance between \( f_3 \) and \( f_4 \) are zero. From Fig. 4(a)(c), all descriptors for the matched signals are relatively small. In the bandlimited, Cartesian coordinate case, the SHOD has the error of \( 10^{-13} \), which is much smaller than the SHD and the SFD, whose error are about \( 10^{-5} \). On the other hand, the distance of mismatched signals should be as far apart as possible. In this case, three descriptors perform well because there is no ambiguity between the matched signals and the mismatched signals.

2) **Rotational invariance of the descriptors:**
   In addition, the descriptor should be invariant over all rotation angles, as shown in \( f_2 \) in Fig. 4(a)(c). For the matched signals, \( f_1/f_2 \) or \( f_3/f_5 \), the SHOD exhibits good invariance while the SHD and the SFD fluctuate more, such as SHD \( f_2 \) and SFD \( f_2 \) in Fig. 4(a) as well as SHD \( f_3 \) in Fig. 4(c). For the distance between mismatched signals, \( f_1/f_3 \) or \( f_2/f_6 \), all descriptors remain constant. As a result, three descriptors all have the invariance property but the SHOD is the most stable one.

3) **Errors of the descriptors in the spherical coordinates:**
   The same experiment is repeated when the input data are sampled from the spherical coordinate. The SHD and the SFD are easy to implement from the spherical samples. However, for the spherical samples, the SHOT cannot be implemented using the transformation coefficients. Instead, the SHOWs are sampled spherically from (11). The error for the descriptors are shown in Fig. 4(b)(d), compared side-by-side with the Cartesian case. It is observed that the SHOD is compatible with the SHD and the SFD in different cases. For the bandlimited signal \( f_2 \) in Fig. 4(b), the error floor of the SHOD is between that of the SHD and that of the SFD. However, for the non-bandlimited signal \( f_5 \), the error floor of the SHOD is less than that of the SFD but is greater than that of the SHD.

4) **The interpolation error in coordinate conversion:**
   Next, the error introduced by interpolation in the SHD and the SFD can be obtained from the results of Cartesian/spherical samples. It is observed that for the bandlimited inputs, the error for SHD and the SFD in the spherical coordinates (Fig. 4(b)) is about 10 times smaller than that in the Cartesian coordinates (Fig. 4(a)), implying that data interpolation really increases the error floor. However, for the non-bandlimited signal \( f_5 \),
the error floor for the SHD in the spherical coordinate, without interpolation is about 10 times smaller than that in the Cartesian samples with interpolation. For the same signal \( f_5 \), the result for the SFD is similar in both sampling schemes.

The following is a short summary for the above rotational invariance experiments.

1) For the Cartesian sampled, bandlimited signal \( f_2 \), SHOD(best) < SFD < SHD, in terms of error.
2) For the Cartesian sampled signals, SHOD(most stable) > SFD > SHD in terms of stability, which means that the descriptors are invariant and do not change too much with the rotation angles.
3) If the input data are sampled from spherical coordinates, the SHOD is compatible with the existing methods. For \( f_2 \), SFD(best) < SHOD < SHD in terms of error. For \( f_5 \), SHD(best) < SHOD < SFD in terms of error.
4) For the spherical samples, the SFD and the SHD without interpolation achieve less error (about 10 times smaller) than the Cartesian samples with interpolation.

5) For the most practical Cartesian-sampled volume data, the SHD is the best choice for the bandlimited case.

D. 3D signal expansion and reconstruction

The SHOT is a complete and orthogonal transform for \( L^2(\mathbb{R}^3) \) so it can be used for signal expansion and reconstruction. We can reconstruct the signal by the order \( N \), index \( n \), index \( l \), or index \( m \). Each index represents different features of the original signal.

In this simulation, the number of discrete points in each dimension is set to be \( N_{pt} = 30 \) and the input signal \( f(r) \) is symmetric and centered at the origin with the vertices \((1, 1, 1)\), \((1, 4, 1)\), and \((4, 1, 1)\) in the first octant. The function value inside this region is one while the outside is zero. This function shapes like a cross. Then, performing the discrete SHOT on \( f(r) \) yields \( \langle nlm | f \rangle \).

Our simulation aims to verify the geometric meaning for the indices. The simulation results are in Table III. The parameter with subscript “max” means that \( \langle nlm | f \rangle \) is selected up to
then becomes a pie. The components represent the signal variation in the $n$th order. As $n$ increases, the corners become sharper and finally it converges to the original signal. The $l$-components represent the signal variation in the $\theta$-direction. The reconstruction signal begins with a sphere ($l_{\text{max}} = 0$), then becomes a pie ($l_{\text{max}} = 2$), wider than the $N_{\text{max}} = 2$ case, and finally approximates the cross. The $m$ index represents the variation in the $\phi$-direction. The first approximate ($m_{\text{max}} = 0$) shapes like a pie and the energy is limited to $[-1,1]$ in the $z$-direction. As $m_{\text{max}}$ increases, the pie is cut into four arms, which approximates the cross further.

An interesting observation is that the top view of the solid cross appears differently for different reconstruction schemes. For $N_{\text{max}}$ and $l_{\text{max}}$, the intermediates look like circles, for $n_{\text{max}}$, the sides are always squared, while for $m_{\text{max}}$, the sides resemble rectangles. Although different intermediates, they converge to the squares when more coefficients are used.

From these results, the SHOT is suitable for analyzing spherically-symmetric objects. $n$, $l$, and $m$ represent the signal component in the $r$-, $\theta$-, and $\phi$-direction roughly, while the order $N$ is related with the energy compaction.

### Table III

**Signal Reconstruction Along $N$, $n$, $l$, and $m$.**

<table>
<thead>
<tr>
<th>$N_{\text{max}} = 0$,</th>
<th>2,</th>
<th>4,</th>
<th>6,</th>
<th>8,</th>
<th>10,</th>
<th>20,</th>
<th>29.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n_{\text{max}} = 0$,</td>
<td>1,</td>
<td>2,</td>
<td>3,</td>
<td>4,</td>
<td>5,</td>
<td>6,</td>
<td>7.</td>
</tr>
<tr>
<td>$l_{\text{max}} = 0$,</td>
<td>2,</td>
<td>4,</td>
<td>6,</td>
<td>8,</td>
<td>10,</td>
<td>12,</td>
<td>14.</td>
</tr>
<tr>
<td>$m_{\text{max}} = 0$,</td>
<td>2,</td>
<td>4,</td>
<td>6,</td>
<td>8,</td>
<td>10,</td>
<td>12,</td>
<td>14.</td>
</tr>
</tbody>
</table>

E. 3D MRI data compression

The signal expansion and reconstruction can be utilized to actual application, such as the compression of the medical MRI data. We use the MRI volume data provided in MATLAB as a practical application of the SHOTs. The dataset comprises 27 grayscaled section images with size $128 \times 128$. The data is first subsampled by 2, making the size to be $64 \times 64 \times 14$. Then these data are placed in the center of the $128 \times 128 \times 128$ cube, which is the input signal of the SHOT.

Note that we have four reconstruction scheme given the SHOTs. Here the reconstruction along the order $N$ is chosen. The notation $N_{\text{max}}$ still stands for the cut-off threshold for the parameter $N$. By setting the coefficients beyond the cut-off to be zero, we reduce the data size and achieve compression.

However, this type of compression is lossy so the error between the actual signal and the compressed signal is a concern. In Table IV, the isosurfaces and the slices are compared along with the actual MRI signal. The isosurface for $N_{\text{max}}$ resembles that for the original data but the ripple differences in the slices are still visible. Besides, when $N_{\text{max}} = 20$, the nose and the ears in slice image 60 are blurred. These errors becomes indistinguishable for $N_{\text{max}} = 80$. The Gibbs-like ripples reduce greatly and the nose and the ears of the patient is apparent. In addition, the image details of the brain are preserved at this level of reconstruction.

The above simulation only visualizes the errors, which should be compared numerically, and besides, the compression ratio is not shown in Table IV. To evaluate this, the root-mean-squared-error (RMSE) and the compression ratio (CR) are defined as follows

$$\text{RMSE} = \sqrt{\frac{1}{N^3} \sum_r |f(r) - \hat{f}(r)|^2}, \quad (52)$$

$$\text{CR} = \frac{\text{Original data size}}{\text{Compressed data size}}, \quad (53)$$

Note that we have four reconstruction scheme given the SHOTs. Here the reconstruction along the order $N$ is chosen. The notation $N_{\text{max}}$ still stands for the cut-off threshold for the parameter $N$. By setting the coefficients beyond the cut-off to be zero, we reduce the data size and achieve compression.

However, this type of compression is lossy so the error between the actual signal and the compressed signal is a concern. In Table IV, the isosurfaces and the slices are compared along with the actual MRI signal. The isosurface for $N_{\text{max}}$ resembles that for the original data but the ripple differences in the slices are still visible. Besides, when $N_{\text{max}} = 20$, the nose and the ears in slice image 60 are blurred. These errors becomes indistinguishable for $N_{\text{max}} = 80$. The Gibbs-like ripples reduce greatly and the nose and the ears of the patient is apparent. In addition, the image details of the brain are preserved at this level of reconstruction.

The above simulation only visualizes the errors, which should be compared numerically, and besides, the compression ratio is not shown in Table IV. To evaluate this, the root-mean-squared-error (RMSE) and the compression ratio (CR) are defined as follows

$$\text{RMSE} = \sqrt{\frac{1}{N^3} \sum_r |f(r) - \hat{f}(r)|^2}, \quad (52)$$

$$\text{CR} = \frac{\text{Original data size}}{\text{Compressed data size}}, \quad (53)$$
TABLE IV
THE ORIGINAL MRI DATA AND THE RECONSTRUCTED MRI DATA AFTER COMPRESSION

<table>
<thead>
<tr>
<th>Isosurface</th>
<th>Slices 60</th>
<th>Slices 63</th>
<th>Slices 66</th>
<th>Slices 69</th>
<th>Slices 72</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td><img src="image1.png" alt="Image" /></td>
<td><img src="image2.png" alt="Image" /></td>
<td><img src="image3.png" alt="Image" /></td>
<td><img src="image4.png" alt="Image" /></td>
<td><img src="image5.png" alt="Image" /></td>
</tr>
<tr>
<td>Reconstructed</td>
<td><img src="image6.png" alt="Image" /></td>
<td><img src="image7.png" alt="Image" /></td>
<td><img src="image8.png" alt="Image" /></td>
<td><img src="image9.png" alt="Image" /></td>
<td><img src="image10.png" alt="Image" /></td>
</tr>
<tr>
<td>$N_{\text{max}} = 5$</td>
<td><img src="image11.png" alt="Image" /></td>
<td><img src="image12.png" alt="Image" /></td>
<td><img src="image13.png" alt="Image" /></td>
<td><img src="image14.png" alt="Image" /></td>
<td><img src="image15.png" alt="Image" /></td>
</tr>
<tr>
<td>$N_{\text{max}} = 10$</td>
<td><img src="image16.png" alt="Image" /></td>
<td><img src="image17.png" alt="Image" /></td>
<td><img src="image18.png" alt="Image" /></td>
<td><img src="image19.png" alt="Image" /></td>
<td><img src="image20.png" alt="Image" /></td>
</tr>
<tr>
<td>$N_{\text{max}} = 20$</td>
<td><img src="image21.png" alt="Image" /></td>
<td><img src="image22.png" alt="Image" /></td>
<td><img src="image23.png" alt="Image" /></td>
<td><img src="image24.png" alt="Image" /></td>
<td><img src="image25.png" alt="Image" /></td>
</tr>
<tr>
<td>$N_{\text{max}} = 40$</td>
<td><img src="image26.png" alt="Image" /></td>
<td><img src="image27.png" alt="Image" /></td>
<td><img src="image28.png" alt="Image" /></td>
<td><img src="image29.png" alt="Image" /></td>
<td><img src="image30.png" alt="Image" /></td>
</tr>
<tr>
<td>$N_{\text{max}} = 80$</td>
<td><img src="image31.png" alt="Image" /></td>
<td><img src="image32.png" alt="Image" /></td>
<td><img src="image33.png" alt="Image" /></td>
<td><img src="image34.png" alt="Image" /></td>
<td><img src="image35.png" alt="Image" /></td>
</tr>
</tbody>
</table>

where $N_{\text{pt}}$ is the number of discrete points in each dimension, $f(r)$ is the samples of the original signal, and $\hat{f}(r)$ is the reconstructed signal. The compressed data size is the number of SHOT coefficients selected. To achieve a good compression, RMSE should be small while CR should be large.

The RMSE and the CR are shown in Fig. 5. These curves decrease monotonically as $N_{\text{max}}$ increases. Based on the RMSE and the CR curve, the SHOT is a possible candidate for the transform coding of the 3D volume data compression/decompression system. Take $N_{\text{max}} = 20$ and $N_{\text{max}} = 80$ for an example. From Fig. 5, the RMSEs for $N_{\text{max}} = 20$ and $N_{\text{max}} = 80$ are $10^{-2}$ and $2.1 \times 10^{-3}$, respectively and the CRs for $N_{\text{max}} = 20$ and $N_{\text{max}} = 80$ are 1000 and 20, respectively. However, in Table IV, although $N_{\text{max}} = 20$ achieves higher compression ratio, the undesired ripples are visible to human eyes. On the other hand, for $N_{\text{max}} = 80$, the reconstructed data is almost indistinguishable to human eyes. The error resembles the Gibbs’ phenomenon due to direct truncation of 3D SHOT coefficients.

VIII. CONCLUSION AND FUTURE WORK

In this paper, we proposed 1) the fast computation algorithm of the SHOW transformation coefficients using the fast Fourier transform, 2) the implementation of the discrete SHOW and the discrete SHOT, and 3) the possible applications. First, from the creation operators, the SHOW was the linear combination of the 3D Hermite Gaussian functions with proper transformation coefficients, which could be computed by the FFT instead of the close-form solution. To implement the transformation coefficients for the complete SHOWs, there were some recursive formulae to speed up the whole computation. Secondly, in terms of discrete implementation, The SHOT was implemented by two steps: The 3D Hermite transform and then a linear coefficient transformation, associated with the transformation coefficients. However, for the high-order discrete SHOTs, some Hermite transforms were unavailable so an eliminating processing was proposed.

Many applications were pointed out and verified through simulations. The discrete SHOWs were the eigenfunctions of the 3D DFT. The rotational invariance feature SHOD outper-
formed the SHD and the SFD for the bandlimited/Cartesian samples. For the signal expansion and reconstruction purpose, different reconstruction parameters indicated different reconstruction behavior. Compression was realized by partial information on the SHOTs.

There are still some problems to be tackled in the future. For instance, the transformation matrix in the high-order case is not unitary, as illustrated in (48). It is possible to find a unitary transformation matrix while keeping the accuracy to some extent. We can implement the fractional 3D DFT by the SHOT. The SHOT can be directly computed in the Cartesian coordinates without coordinate conversion. We can achieve perfectly orthogonality and additivity for the obtained 3D DFT and the 3D discrete fractional Fourier transform. The signal expansion and reconstruction as well as compression are useful in the actual MRI data. The human brain, shaping like a sphere, seems suitable for the SHOTs. This is a very promising tool.

Fig. 5. (a) The root-mean-squared-error and (b) the compression ratio between the original MRI data and the compressed one.

References


