An accelerated direct demodulation method for image reconstruction using spherical data from the hard X-ray modulation telescope *

Zhuo-Xi Huo and Jian-Feng Zhou

Department of Engineering Physics and Center for Astrophysics, Tsinghua University, Beijing 100084, China; zhoujf@tsinghua.edu.cn

Key Laboratory of Particle & Radiation Imaging (Tsinghua University), Ministry of Education, Beijing 100084, China

Key Laboratory of High Energy Radiation Imaging Fundamental Science for National Defense, Beijing 100084, China

Received 2012 October 18; accepted 2013 April 22

Abstract The hard X-ray modulation telescope (HXMT) mission is mainly devoted to performing an all-sky survey at $1 - 250$ keV with both high sensitivity and high spatial resolution. The observed data reduction as well as the image reconstruction for HXMT can be achieved by using the direct demodulation method (DDM). However the original DDM is too computationally expensive for multi-dimensional data with high resolution to be employed for HXMT data. We propose an accelerated direct demodulation method especially adapted for data from HXMT. Simulations are also presented to demonstrate this method.

Key words: methods: data analysis — methods: numerical — techniques: image processing — instrumentation: high angular resolution

1 INTRODUCTION

1.1 Hard X-ray Modulation Telescope

The hard X-ray modulation telescope (HXMT) is a space telescope in circular low Earth orbit with a $43^\circ$ inclination at the altitude of 550 km (Li 2007; Lu et al. 2010). There are three detectors onboard, including a high energy X-ray detector (HE), a medium energy X-ray detector (ME) and a low energy X-ray detector (LE). HE is the primary one, which consists of 18 modules and each module is composed of an HE collimator, a phoswich scintillation detector and readout electronics (Li 2007; Han et al. 2010).

The HE collimator is used to define the field of view (FOV) of the detector (Han et al. 2010). There are two types of HE collimators, and they differ in their FOVs: 15 of them have $5.7^\circ \times 1^\circ$ FOVs and three of them have $5.7^\circ \times 5.7^\circ$ FOVs. Their optical axes are parallel but the directions of their cross-sections are different.

* Supported by the National Natural Science Foundation of China.
There are two imaging observation modes designed for the scientific goal: the all-sky survey mode as well as the deep imaging observations of selected sky regions. The all-sky survey is performed while the satellite orbits as well as while the plane of its orbit precesses (Li 2007). The orbital period is about 95.5 min and the precession rate is $-5.45^\circ$ per day (Lu et al. 2010). The deep imaging observations will be performed by pointed observations with pointing directions distributed uniformly in the region and these pointed observations will be performed by progressive scanning.

1.2 Modulation

According to Li & Wu (1994), an observation process can be modeled as

$$\int p(\omega, x)f(x)dx = d(\omega),$$

where $f(x)$ is the intensity distribution of the object (i.e. the image) and $d(\omega)$ is the observed data modulated by an integral kernel (modulation function) $p(\omega, x)$. In practice, the observed data are recorded as discrete chunks (3-D), maps (2-D) or series (1-D). We have

$$d_k = \sum_{i=1}^{N} p_{k,i} f_i, \quad \forall k = 1, \cdots, M,$$

or

$$d = Pf,$$

where $d$ is an $M \times 1$ column vector (data vector) in data space $\{d\}$ representing all possible observed data records and $f$ is an $N \times 1$ column vector (object vector) in image space $\{f\}$ representing all possible images; thus $P$ is an $M \times N$ matrix, called the kernel matrix in the space $\{d\} \times \{f\}$.

1.3 Challenges in Image Reconstruction for HXMT

Generally speaking, the direct demodulation method (DDM) is designed to solve the demodulation problem, which is the process of finding the best object $f(x)$ to satisfy Equation (1) when the kernel $p(\omega, x)$ and the observed data $d(\omega)$ are known (Li & Wu 1993). Richardson–Lucy iteration (Richardson 1972; Lucy 1974) is used for DDM, hence the numerical evaluation of modulations is the most time-consuming part. The computational complexity increases rapidly as the size of the data increases in multi-dimensional modulation evaluation. As a result, it is not feasible to reconstruct an all-sky image from data observed by the HXMT using the original DDM. Hence, accelerations of the method are required.

According to Shen & Zhou (2007), modulations can be accelerated through fast Fourier transforms (FFTs). It is implied that the modulation is shift-invariant. In this way the modulation equation is reduced to a convolution equation. However because of the topology of a spherical surface, the point spread function (PSF) must satisfy certain a requirement so that a convolution can be defined there.

The FOVs of detectors in the HXMT are not circularly symmetric, i.e. contributions to a detector from the sources in its FOV not only depend upon the radial distances from the sources to the center of the FOV, but also the directions. In addition, during the all-sky survey, the FOV of each detector travels on a spherical surface instead of a plane. As a result, the FOVs of each detector around different positions on the celestial sphere are never parallel to each other. For example, the path of an FOV during the first phase of the all-sky survey is shown in Figure 1.

Therefore neither the Fourier transform nor spherical harmonics can be used to reduce the modulation of the kernel function of observations or data taken with the HXMT on the celestial sphere.
In the following sections, we present a pixelization and tessellation scheme for data defined on a spherical surface as well as a method to accelerate the numerical evaluation of modulation for such data, thus yielding the DDM for HXMT.

2 IMAGE RECONSTRUCTION FOR HXMT

2.1 Pixelization and Tessellation of Data on a Spherical Surface

As the first step in numerical analysis, pixelization of data on a spherical surface is always an attractive strategy (Tegmark 1996; Crittenden & Turok 1998; Doroshkevich et al. 2005; Górski et al. 2005). Research on this problem is driven by applications, e.g. analysis of Cosmic Microwave Background (CMB) data. Due to the topological nature of a spherical surface, a theoretically ideal pixelization scheme that would work for all cases does not exist for data defined on such a surface (Tegmark 1996; Górski et al. 2005).

Image reconstructions for both all-sky survey and observations of selected sky regions can be conquered locally region by region since given a single point of the observed data, only objects within the current FOV are relevant, but given a single point on the unknown image, only locally observed data are relevant. Therefore all-sky pixelization is not necessary. We only need a local pixelization scheme adequate for a sky region larger than the FOV of HXMT, which is $5.7^\circ \times 5.7^\circ$.

Virtues of the pixelization scheme in this work include:

- All pixels have equal size, which speeds up and simplifies the evaluation of numerical integration, the elementary operation of modulations.
- The geodesic between two points $p = (i_x, i_y)$ ($i_x$ and $i_y$ are indices) and $q = (i'_x, i'_y)$ and the azimuth of one point with respect to the other are both shift-invariant. That is,
  \[ d[(i_x, i_y), (i'_x, i'_y)] = d[(i_x + j_x, i_y + j_y), (i'_x + j_x, i'_y + j_y)], \quad \forall i_x, i_y, i'_x, i'_y, j_x, j_y, \]  

  \[ \alpha[(i_x, i_y), (i'_x, i'_y)] = \alpha[(i_x + j_x, i_y + j_y), (i'_x + j_x, i'_y + j_y)], \quad \forall i_x, i_y, i'_x, i'_y, j_x, j_y, \]

where $d(p, q)$ is the geodesic distance between $p$ and $q$, which is the length of the minor arc on the great circle from $p$ to $q$, and $\alpha(p, q)$ is the angle where the geodesic arc from $p$ to $q$ crosses the meridian containing $p$. Therefore all pixels should have the same shape. If the PSF of the telescope is circularly

\[ \text{Fig. 1 Scanning path of the FOV for a detector on HXMT during the first phase of its all-sky survey.} \]

We exaggerate the precession rate of the satellite by 20 times to separate adjacent scanning circles in this diagram.
symmetric, i.e., the PSF can be expressed as a function of distance from the center of the FOV, the modulation then degenerates to convolution.

- 2-D orthogonal pixel indexing. Pixels indexed with two indices \(i_x\) and \(i_y\) in Equations (4) and (5) suggest that the pixel indexing is defined by a 2-D Cartesian coordinate system. If \(I_{i_x,i_y}\) represents \(I(i_x s_x, i_y s_y)\), where \(s_x\) and \(s_y\) are the sampling intervals along the \(x\)-axis and \(y\)-axis respectively, then \(I_{i_x,i_y}\) is the sampled image value on \((i_x, i_y)\).

The equidistant cylindrical projection (ECP) method is commonly used in geophysics and climate modeling. Equidistant pixels on the surface of a cylinder \(c\) are projected to the surface of an inscribed sphere \(s\) of the cylinder \(c\) towards their symmetry axis. Adjacent pixels on the spherical surface either have the same right ascension (R.A.) or the same declination (Dec.). Pixels on the same parallel (or meridian) are uniformly spaced on adjacent meridians (or parallels). Pixels with higher latitudes (closer to poles) have smaller sizes and greater distortions than pixels at lower latitudes (closer to the equator).

HEALPix by Górski et al. (2005) has recently become a standard structure for spherical data analysis, especially for CMB experiments. Although equal areas of pixels in HEALPix are vital for spherical harmonic transforms, the shapes of pixels are different and thus the shift-invariance is not exactly correct. Therefore HEALPix is not adapted for HXMT data in this article. Neither ECP nor the HEALPix method is adequate for HXMT.

2.1.1 Pixelization based on quadrilateral projection

A pixelization scheme is designed for HXMT data. We use the scheme introduced here to define a grid of pixels for a specific region of the sky. A pixel on a sphere is an elementary area on the sphere inside its boundary and around its center, the boundary of which is on a sphere and is defined by four vertices of the pixel and four geodesics of the sphere between each adjacent pair. The position of the center of a pixel on a sphere is fixed to indicate the area inside the boundary of the pixel.

We start by generating pixels with equal areas and the same shape in a square on a tangent plane \(p\) of the unit sphere.

This area is divided into \(N_p = N \times N\) square pixels. Then the centers of all the pixels are projected from the plane \(p\) to the surface of the unit sphere \(s\) (as shown in Fig. 2). The projection can be either radial or parallel. In the radial mode, the center of each pixel is projected towards the center of the sphere \(s\), but in parallel mode it is projected perpendicularly to the plane \(p\).

Once the centers of all pixels are fixed to the sphere (the first step in Fig. 3), find the geodesics between each center and its four nearest neighbors (the second step in Fig. 3). Draw an arc, representing part of a great circle, perpendicularly across the midpoint of each geodesic (the third step in Fig. 3). The four points where each perpendicular arc intersects another two arcs are the vertices and the four perpendicular arcs form the boundary. With the center indicating the area inside of the boundary, a pixel is defined on the sphere around each center (the last step in Fig. 3).

Pixels on the spherical surface projected farther from the center of the square have greater distortions than those projected from the central area of the square. Obviously, the amount of distortion in the pixel is independent of the position of the pixel on the sphere. It only depends on its original position in the plane, more specifically the distance from the center of the square on the plane; therefore we can always use a sufficiently small square to make the distortion negligible.

We use the term tessella to refer to a set of pixels on the spherical surface, which are projected from all pixels within a square on a plane. Hence it suggests that we divide the problem of pixelization of a whole spherical surface into two problems, pixelization of any small region of the spherical surface and tessellation of the whole spherical surface using tessellae of pixels. Generally speaking, there should not be overlaps or gaps between adjacent tessellae, however, since we will not perform numerical evaluations on different tessellae at the same time, overlaps only make evaluations of pixels in the overlapping regions redundant.
**Fig. 2** Plane projection around the null position of the surface of a unit sphere. The unit sphere $s$ is centered on $(0, 0, 0)$, the origin of the $xyz$ 3-D Cartesian coordinate system. The null position is at $(1, 0, 0)$. The plane $p$ is parallel to the plane $yOz$. There are four pixels on the plane. They are projected in either the radial or parallel directions to $s$.

**Fig. 3** Steps involved in defining a spherical pixel around its center after being projected from a plane.
Since all tessellae have the same area and shape, we can generate a set of pixels (i.e. a tessella) around the null position of the spherical surface and rotate this initial tessella to a series of positions to cover the whole sphere. In this way we can pixelize any small region of a spherical surface by the initial tessella and its rotation is expressed by a quaternion (Appendix A).

As shown in Figure 2, we define the null position of the surface of a unit sphere \( s \) as \( (1, 0, 0) \) and the plane \( p \) as \( x = 1 \). The center of this square is projected perpendicularly from \( (1, 0, 0) \) to \( p \).

As shown in Figure 4 the pixel distortions are negligible in a small region close to the equator, for all the three projection-based pixelization schemes. The distortions become significant when the region expands. Although in a region with low latitude the distortion of the ECP scheme is suppressed better than those based on plane projection, we soon find that in regions with high latitude the ECP scheme suffers from severe distortion, even in a small region. Obviously, distortions based on plane projection schemes are independent of latitude. Statistics on the shapes and areas of pixels on the sphere can be seen in Figure 5.

The distribution of relative pixel area is a measure of the uniformity of the pixelization scheme, which is critical for numerical integration implemented by unweighted summation. The accuracy of such implementation can be derived from this distribution. Summation weighted by the relative pixel area can be used to improve the accuracy if necessary. For a grid of \( N \times N \) pixels there are \((N - 1) \times (N - 1)\) pairs of adjacent pixels. For each pair we calculate the geodesic distance of the centers of two pixels. The distribution of relative pixel distance is a measure of the uniformity of the sampling intervals. For each pixel we calculate the angle between the geodesics of its right and
Fig. 5 Pixel statistics. 1024 × 1024 pixels are projected onto a 11.25° × 11.25° spherical surface in different ways. The first row shows ECP pixels centered at 0° latitude; the second row shows ECP pixels centered at 60° latitude; the third row shows pixels with a parallel plane projection; the last row shows pixels with a radial plane projection. The first column (from left to right) shows relative areas (proportion of the average of areas of all the pixels); the second column shows relative distances (proportion of the average of distances between adjacent pixels); the last column shows angles between adjacent sides of the boundary of each pixel.
down adjacent pixels. The distribution of angles is a measure of the orthogonality in the pixelization scheme.

2.1.2 Tessellation scheme

The size of the tessella is \( \lambda \times \lambda \), where

\[
\lambda = \frac{\pi}{2M}, \quad M = 1, 2, \ldots .
\]  

For the radial projection the size of the square on the tangent plane is \( 2 \tan^{-1} \frac{\lambda}{2} \times 2 \tan^{-1} \frac{\lambda}{2} \). The tessellation of the unit sphere is implemented in three steps.

The first step is tessellation of the prime meridian of the sphere. Rotate the initial tessella from the null position towards the north pole as well as the south pole at intervals of \( \lambda \), as shown in Figure 6. Then the prime meridian is covered by \( 2M + 1 \) tessellae. We call each of those tessella the meridianal tessella (Fig. 6).

The second step is tessellation of the equatorial area of the sphere. Rotate the initial tessella around the \( z \)-axis, also at intervals of \( \lambda \), so that the equator of the sphere is covered by \( 4M \) equatorial tessellae, as shown in Figure 6.
The last step is tessellation of the remaining area of the sphere. Rotate each meridional tessella except the initial one and the two covering the polar caps around the z-axis at intervals of \( \frac{\lambda}{\cos(|\theta_w| - \pi/2)} \), where \( \theta_w \) is the latitude of the center of each tessella,

\[
\theta_w = \pm \frac{m\pi}{2M}, \quad m = 1, \ldots, M - 1.
\]  

The tessellation along a given latitude is shown in Figure 6.

Finally the number of tessellae covering the whole sphere is

\[
N_t = 4M + 2 + 2 \sum_{m=1}^{M-1} \left\lceil \frac{\pi}{\arctan \left( \frac{\tan \frac{m\pi}{2M}}{\cos \frac{m\pi}{2M} + \sin \frac{m\pi}{2M} \tan \frac{\pi}{4M}} \right)} \right\rceil.
\]  

where \( \left\lceil \cdots \right\rceil \) is the ceiling operator denoting the smallest integer no less than the number inside it. The tessellation of the celestial sphere is illustrated in Figure 7.

The quaternion representing the rotation applied to each tessella is

\[
q_{\phi_w, \theta_w} = \cos \frac{\theta_w}{2} \cos \frac{\phi_w}{2} + \sin \frac{\theta_w}{2} \sin \frac{\phi_w}{2} i - \sin \frac{\theta_w}{2} \cos \frac{\phi_w}{2} j + \cos \frac{\theta_w}{2} \sin \frac{\phi_w}{2} k.
\]
where $\phi_w$ and $\theta_w$ represent the position of the center of the tessella on the sphere, in longitude and latitude respectively. The corresponding rotation matrix is

$$
R_{\phi_w, \theta_w} = \begin{pmatrix}
\cos \theta_w \cos \phi_w & -\sin \phi_w & -\sin \theta_w \cos \phi_w \\
\cos \theta_w \sin \phi_w & \cos \phi_w & -\sin \theta_w \sin \phi_w \\
\sin \theta_w & 0 & \cos \theta_w
\end{pmatrix}.
$$

(10)

Given a pixel $r_{i,j}$ on the initial tessella, where $i$ and $j$ are the indices of each pixel, the corresponding pixel on the tessella around $(\phi_w, \theta_w)$ is then $R_{\phi_w, \theta_w} r_{i,j}$.

### 2.1.3 Pixelization of observed data

The original observed data from each detector module are recorded as a time of arrival (TOA) sequence of X-ray photons. The following processes require converting the original observed data from time series into a pixel-wise format, i.e., pixelization of the observed data. Along with the scientific data, engineering data are also being recorded such as the attitude of the spacecraft. By computing interpolations, the attitude of the spacecraft can be determined at any given point during the observation time, i.e. for each detected photon the attitude of the spacecraft can be assigned. From the attitude of the spacecraft we can calculate the position on the celestial sphere where each collimator is pointing. For example, in the equatorial coordinate system, given a specific detector module, each detected photon from this module comes with not only its TOA but also the R.A. and Dec. of a position on the celestial sphere the collimator is pointing, which is the projection on the celestial sphere along the optical axis of the collimator. The projection on the celestial sphere is then called the position of the detected photon.

For each photon, first we find the tessella that cover its position on the celestial sphere. Then for each tessella that covers its position, we locate the pixel at this position. Because the boundary of each tessella, as well as for each pixel, is defined, the problem of finding whether a specific point on the celestial sphere lies inside or outside a tessella or a pixel is classified as a point-in-square-on-sphere problem. See Appendix B for our approach to solving this problem.

Once the position of each detected photon is mapped to pixels and the tessella are defined on the celestial sphere, the total number of detected photons can be calculated for each pixel. Taking the time interval between the TOA of a photon and the next one on the same detector as the exposure time of the former photon, the combined exposure time of all the photons mapped to each pixel yields the total exposure time on this pixel. Divide the total number of photons by the total exposure time on each pixel, and we have the counting rate for each pixel, representing the observed data in pixel-wise format.

### 2.2 Acceleration of Modulation

#### 2.2.1 Status parameters of the collimated detector

Evaluating numerical modulation is ubiquitous in both simulation of observed data and image reconstruction with a direct demodulation method. In observations made by HXMT, the observed data are a function of the given collimator status, which includes the collimator identifier $c$ and its orientation.

To describe the orientation of a given collimator, we use a unit vector along its optical axis as well as another unit vector along a fixed axis perpendicular to its optical axis (e.g. the long edge of one of its slices). We call the unit vector along its optical axis and the later one pointing vector and position vector (acting as the position angle parameter of a telescope) respectively. Then we define a null status of a given collimator where its pointing vector points at the null position of the current coordinate system (along the $x$-axis) and its position vector points at $(0, 0, 1)$ (along the $z$-axis). We use three status parameters $\phi$, $\theta$ and $\psi$ to represent the current status of the given collimator. First
Accelerated DDM for HXMT

2.1 PSF and numerical evaluation of the modulation kernel function

In the equatorial coordinate system, an image is a function of position on the celestial sphere denoted by R.A. and Dec.

Given the observed data \( d(\phi, \theta, \psi, c) \) and the image \( f(\phi, \theta) \), the modulation in Equation (1) yields

\[
d(\phi, \theta, \psi, c) = \int_{\Omega} p(\phi, \theta, \psi, c, \phi', \theta') f(\phi', \theta') d\Omega, \tag{18}
\]
where \( p(\phi, \theta, \psi, c, \phi', \theta') \) is the modulation kernel function representing the response of collimator \( c \) with status \( \left( \begin{array}{c} \phi \\ \theta \\ \psi \end{array} \right) \) to a unit point object at position \( \left( \begin{array}{c} \cos \theta' \cos \phi' \\ \cos \theta' \sin \phi' \\ \sin \theta' \end{array} \right) \), and \( \Omega \) represents the solid angle.

The PSF of collimator \( c \) is defined through its modulation kernel function as

\[
P(\phi, \theta, c) = p(\phi, \theta, 0, c, 0),
\]

which is the response of collimator \( c \) to a unit point source located at the null position of the celestial sphere when the collimator is pointing at \( (\phi, \theta) \) and its position angle parameter remains zero. In practice, the PSF is measured during the calibration of the detector (Han et al. 2010), estimated from observations, or predicted theoretically. The response to the unit object is only determined by the position of the unit object relative to the collimator rather than their absolute positions in any coordinate system, so the modulation kernel function in a specific coordinate system can be evaluated through the PSF \( P(\phi, \theta, c) \) by coordinate transformations. For example, the PSF of collimators as well as the HE detector are shown in Figure 8.

To evaluate the modulation kernel function \( p(\phi, \theta, \psi, c, \phi', \theta') \) we find a coordinate system where the coordinate of the unit object is \( \left( \begin{array}{c} 1 \\ 0 \\ 0 \end{array} \right) \), i.e., the null position of the new coordinate system and the position angle parameter of the collimator yield zero. Once we find the pointing parameters \( \Phi \) and \( \Theta \) of the collimator in the new coordinate system, the modulation kernel function can be evaluated immediately.

This is implemented by the following steps.

1. Rotate the current coordinate system \( S_0 \) to the first auxiliary coordinate system \( S_1 \) where the unit object lies on \( \left( \begin{array}{c} 1 \\ 0 \\ 0 \end{array} \right) \). For example, we can first rotate the original coordinate \( S_0 \) around its \( z \)-axis by \( \phi' \) then around its \( y \)-axis by \( -\theta' \).
2. Find the status parameters \( \phi_1, \theta_1 \) and \( \psi_1 \) in \( S_1 \). The coordinate of a vector in the rotated coordinate system \( S_1 \) is equivalent to rotating the vector inversely, that is rotating it around the \( z \)-axis of \( S_0 \) by \( -\phi' \) then rotating around the \( y \)-axis of \( S_0 \) by \( \theta' \). We formulate the rotation by quaternions as:

\[
q_1 = \cos \frac{\phi'}{2} - \sin \frac{\phi'}{2} k,
q_2 = \cos \frac{\theta'}{2} + \sin \frac{\theta'}{2} j,
q = q_2 q_1,
\]

where \( q_1 \) and \( q_2 \) are auxiliary quaternions. The corresponding rotation matrix is

\[
R_{\theta', \phi'} = \begin{pmatrix}
\cos \theta' \cos \phi' & \cos \theta' \sin \phi' & \sin \theta' \\
-\sin \phi' & \cos \phi' & 0 \\
-\sin \theta' \cos \phi' & -\sin \theta' \sin \phi' & \cos \theta'
\end{pmatrix}.
\]

Given that the pointing vector and position vector of the collimator are \( p_{\phi, \theta, \psi} \) and \( a_{\phi, \theta, \psi} \) in \( S_0 \) as stated in Equations (16) and (17) respectively, the two vectors of the collimator are

\[
p_1 = R_{\theta', \phi'} p_{\phi, \theta, \psi},
\]

and

\[
a_1 = R_{\theta', \phi'} a_{\phi, \theta, \psi}.
\]
(3) Rotate $S_1$ around its $x$-axis by $-\alpha$ so that the position angle parameter of the given collimator in the rotated coordinate system $S_2$ is zero. Given the pointing vector $p_2$ and the position vector $a_2$ of the collimator in $S_2$, we have $(a_2 \times p_2) \cdot k = 0$, i.e. the cross-product of the position vector and the pointing vector lies on the $xOy$ plane of $S_2$. Therefore we have

$$\alpha = \text{Arg}(R_{\theta', \phi'}(a_{\phi, \theta, \psi} \times p_{\phi, \theta, \psi}) \cdot k, R_{\theta', \phi'}(a_{\phi, \theta, \psi} \times p_{\phi, \theta, \psi}) \cdot j),$$

(26)

where $\text{Arg}(x, y)$ is the principal argument of a complex number with $x + yi$. 

---

**Fig. 8** PSFs of the main detector on HXMT. (a) PSF of the collimator with a $5.7^\circ \times 1^\circ$ FOV; (b) PSF of the collimator with a $5.7^\circ \times 5.7^\circ$ FOV; (c) PSF of the high energy detector, overlaid.
Find the pointing parameters $\phi_2$ and $\theta_2$ of the collimator in $S_2$ and use them to evaluate the PSF, which is equivalent to the modulation kernel function in $S_2$. Given the rotation matrix

$$R_\alpha = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{pmatrix},$$

where $\alpha$ is solved in Equation (26), the pointing vector $p_2$ of the collimator is

$$p_2 = R_\alpha^{-1} R_{\phi_2, \theta_2} p_{\phi_2, \theta_2}.$$  

Hence we can find the pointing parameters $\phi_2$ and $\theta_2$ from $p_2$ as

$$\phi_2 = \text{Arg}(p_2 \cdot \mathbf{j}, p_2 \cdot \mathbf{i}),$$

$$\theta_2 = \arcsin(p_2 \cdot \mathbf{k}).$$

Finally the modulation kernel function is evaluated as

$$p(\phi, \theta, \psi, c, \phi', \theta') = P(\phi_2, \theta_2, c).$$

### 2.2.3 Numerical evaluation of modulation

Since the response of an HXMT detector to an object only depends upon the position of the object relative to the detector, we can always rotate both the detector and the object to the initial tessella of the celestial sphere to calculate the modulation kernel function, if the tessella is larger than the FOV of the detector,

$$p(p_{\phi, \theta, \psi}, a_{\phi, \theta, \psi}, c, p_{\phi', \theta'}) = p(R_{\phi, \theta, \psi}^{-1} p_{\phi, \theta, \psi}, R_{\phi, \theta, \psi}^{-1} a_{\phi, \theta, \psi}, c, R_{\phi, \theta, \psi}^{-1} p_{\phi', \theta'}).$$

where $p_{\phi, \theta, \psi}$ and $a_{\phi, \theta, \psi}$ represent the pointing vector and the position vector of the collimator which are related to its status $(\phi, \theta, \psi)$; $p_{\phi', \theta'}$ represents the point $(\phi', \theta')$, so $p(p_{\phi, \theta, \psi}, a_{\phi, \theta, \psi}, c, p_{\phi', \theta'})$ is equivalent to $p(\phi, \theta, \psi, c, \phi', \theta')$. Rotation matrix $R_{\phi, \theta, \psi}$ rotates the initial tessella to $(\phi, \theta, \psi)$.

We use the first-order terms of its Taylor series to approximate the modulation kernel function around $\phi = 0$, $\theta = 0$, $\phi' = 0$ and $\theta' = 0$ (i.e. on the initial tessella), as

$$p(\phi, \theta, \psi, c, \phi', \theta') = (R_\psi P)(\phi - \phi', \theta - \theta', c) + O(\phi^2) + O(\theta^2) + O(\phi'^2) + O(\theta'^2),$$

where we put a rotation matrix $R_\psi$ to the left of the PSF $P$ to define a new PSF $(R_\psi P)$ which is rotated from the original one. $O(\phi^2)$, $O(\theta^2)$, $O(\phi'^2)$ and $O(\theta'^2)$ are the remainders.

Taking the pixelization of observed data into account, the corresponding discrete modulation equation yields

$$d_{i,j,c} = \sum_{i', j'} (R_{\psi_{i,j}} P)_{i-i', j-j', c} f_{i', j'},$$

where $\psi_{i,j}$ is the position angle parameter of the collimator $c$ on $(i,j)$ and $P_{i,j,c}$ is the value of the PSF of the collimator on $(i,j)$. 
2.2.4 Position angle cluster analysis and approximation

Because of the topology of the spherical surface, the position angle parameter of each collimator varies during the all-sky survey and the variance depends upon the position on the celestial sphere. Given a specific scanning program for the all-sky survey, e.g., the satellite orbits along the path shown in Figure 1 and its roll angle is fixed to a constant value such as 0°, −30° or 30°, the position angle parameter of a collimator is approximately either $\psi_c - 43^\circ$ or $\psi_c + 43^\circ$ ($\psi_c$ is the angle between the position vector of the collimator and the instantaneous velocity of the satellite, i.e., the tangent vector of the scanning trajectory, and the inclination of the satellite orbit is 43°, Li 2007) in the initial tessella, but varies significantly over a wide range in some other tesselae, as shown in Figure 9.

On the pixel grid of a small sky region, we can rotate the PSF around itself and shift it along two orthogonal directions to approximately evaluate the modulation kernel function, as shown in

Fig. 9 Position angle variance and distribution. (a) Position angle variance along scanning path of an all-sky survey in the equatorial area; (b) Position angle distribution in equatorial area; (c) Position angle variance along the scanning path of an all-sky survey in the high latitude area; (d) Position angle distribution in the high latitude area.
Equation (34). Because of the variance from the position angles as well as the lack of circular symmetry in the PSF, the modulation kernel function is shift-variant. We can approximate the modulation kernel function with the sum of a finite series of functions so that the position angle parameter of the collimator status is fixed for each function, as

\[ p(\phi, \theta, \psi, c, \phi', \theta') = \int_{-\pi}^{\pi} p(\phi, \theta, \psi', c, \phi', \theta') \delta(\psi - \psi') d\psi' \]

\[ \approx \sum_{k=1}^{K} p \left( \phi, \theta, \frac{2\pi k}{K} - \pi, c, \phi', \theta' \right) \delta_{k, \frac{\psi + \pi}{2\pi K}}, \]

where \( \frac{\psi + \pi}{2\pi K} \) is rounded to the nearest integer to evaluate the Kronecker delta \( \delta_{k, \frac{\psi + \pi}{2\pi K}} \), which is equivalent to the Dirac delta function \( \delta(\psi - \psi') \).

We use cluster analysis to assign the position angle parameters of all the observed data in a given sky region into groups so that the difference between each position angle parameter of a group and the cluster center of the group is less than a pre-determined upper limit \( \epsilon \) according to the required precision. Therefore the modulation kernel function is approximately shift-invariant if all of its possible position angle parameters are in the same group. Hence, we can rewrite Equation (34) by expanding the right-hand side:

\[ d_{i,j,c} = \sum_{i',j'} (R_{\psi, P})_{i-v, j-j', c} f_{i', j'} \]

\[ = \sum_{k=1}^{K} \sum_{i',j'} (R_{\psi, P})_{k-v, j-j', c} f_{i', j'} \delta(|\psi_k - \psi_{i,j}| < \epsilon), \]

where

\[ \delta(|\psi_k - \psi_{i,j}| < \epsilon) = \begin{cases} 1 & \text{if } |\psi_k - \psi_{i,j}| < \epsilon, \\ 0 & \text{otherwise}. \end{cases} \]

Finally we approximate the modulation by the sum of a finite series of convolutions as

\[ d_{i,j,c} = \sum_{k=1}^{K} [(R_{\psi, P})_{c} * f]_{i,j} \delta(|\psi_k - \psi_{i,j}| < \epsilon), \]

where \( [(R_{\psi, P})_{c} * f] \) represents the convolution between the rotated PSF of collimator \( c \) and the image \( f \). Then we can employ FFT algorithms to accelerate the modulation.

3 SIMULATED DATA AND RECONSTRUCTED IMAGES

The 7-year INTEGRAL all-sky survey catalog (Krivonos et al. 2007) is used as an input to simulate HXMT data, as shown in Figure 10.

Two small regions are selected for simulated observations as well as reconstructions. One of them is in an equatorial area (the model image is shown on the top left of Figure 11) while the other is in a high latitude area (as shown on the top right of Fig. 11). A background of 0.01 mCrab at 20 keV is simulated. The PSF of HE is shown in Figure 8. The total effective area of HE detectors corresponding to each position is 1500 cm². The exposure towards a 11.25° × 11.25° region in the equatorial area lasts 1.6 × 10⁴ s and it lasts 3.0 × 10⁴ s in the high latitude area. Hence the background level of observed data is 100 cts and 200 cts in the equatorial area and in high latitude area respectively. Poisson noise is added, so the standard deviation of the noise is about 10 cts and
Fig. 10 Sources in the 7-year INTEGRAL all-sky survey catalog. Upper box: a tessella in the equatorial area. The R.A. and Dec. of its center are $-80^\circ$ and $0^\circ$ respectively. Lower box: a tessella in the high latitude area. The R.A. and Dec. of its center are $-101^\circ$ and $-37^\circ$ respectively.

Table 1 Computational Costs of the Accelerated DD Method and the Original DD Iterations

|                | Time (s) | Memory (byte) | Iterations | Size (pixel) | Hardware performance (GFlop) |
|----------------|----------|---------------|------------|--------------|-------------------------------|
| Equatorial area on CPU | 3        | $1 \times 10^7$ | 100        | $512 \times 512$ | 96                           |
| High-latitude area on CPU | 9        | $2 \times 10^7$ | 100        | $512 \times 512$ | 96                           |
| Equatorial area on GPU | 0.2      | $1 \times 10^7$ | 100        | $512 \times 512$ | 1000                         |
| High-latitude area on GPU | 0.6      | $2 \times 10^7$ | 100        | $512 \times 512$ | 1000                         |
| Original DD      | 1440     | $2.8 \times 10^9$ | 100        | $121 \times 121$ | 19.2                         |

The acceleration of the algorithm is independent of hardware performance or the size of the problem. It is calculated as $(T_0/T_1) \cdot (P_0/P_1)$, where $T_0$ and $T_1$ are time costs of solving a problem with the same size by original DD and by the accelerated method respectively; $P_0$ and $P_1$ are the performances of the computers where original DD and the accelerated method are implemented respectively. Since the complexity of the original DD is $O(n^2)$ ($n$ is the number of pixels), the time cost of 100 iterations on $512 \times 512$ pixels is about $1440 \times (512/121)^4 = 461637$ s. For example, the acceleration of the algorithm for the equatorial area on a CPU is $(461637/3) \times (19.2/96) = 30776$.

14 cts in the equatorial area and in the high latitude area respectively. The position angle variance in the high latitude area is more significant, thus the simulated observation as well as the reconstruction consume more computation time.

The model images, observed data as well as the reconstructed images are shown in Figure 11. Figure 11 shows that with the accelerated DD we can obtain expected images from observed data, thus the proposed method is valid.

To evaluate the performance of this method, the time and memory consumptions for different setups are shown in Table 1.
Fig. 11 Top left: model image in the equatorial area. Top right: model image in the high latitude area. Middle left: observed image in the equatorial area, with Poisson noise. Middle right: observed image in the high latitude area, with Poisson noise. Bottom left: reconstructed image using the accelerated DD method, with 100 000 iterations (about 3 min). Bottom right: reconstructed image using the accelerated DD method, with 100 000 iterations (about 10 min).

The time costs of accelerated DD on a CPU is measured on an Intel Core i7-2720QM with a single process. But considering that its AVX feature (advanced vector extension) as well as the FFT is implemented using an FFTW library with pthread (POSIX Threads) enabled, multiple CPU cores are involved in a single process. Its theoretical multi-core performance is 96 GFlops. The time costs on a GPU is measured on an Nvidia Quadro 1000M GPU and projected to a more realistic desktop GPU with about 1 TFLOPs/s of processing power. The computational costs of the original DD iterations are also included in Table 1 for comparison (Shen & Zhou 2007). The original DD
Fig. 12 Upper four images: observed data of point sources with 20σ, 5σ, 3σ and 1σ significance. The SNRs are 11.1, 5.1, 2.88 and −2.07 dB respectively. Lower four contour diagrams: reconstructed images of the point sources.
was tested in 2007 on an Intel Core 2 Duo E6600 CPU. Its theoretical multi-core performance is 19.2 GFlops. The accelerated and original DD have been tested on different systems and the mainstream computing power has been improved, therefore we take into account not only the time costs but also the hardware performances. The converted algorithm acceleration is also shown in Table 1, which is hardware independent.

To test the resolving ability and reliability of the DD method for HXMT all-sky survey data, we set up a point source as well as a uniform background so that the significance of the source is $k\sigma$, where $\sigma$ is the standard deviation of the fluctuation of the background. We simulated characteristics of the observed data and the reconstructions for all the set-ups, as shown in Figure 12.

As we can see in Figure 12, bright point sources with significance more than 5$\sigma$ can be accurately reconstructed. However, reconstruction of faint point sources with significance less than 3$\sigma$ is affected by the noise. Resolution better than 5' and positioning accuracy better than 2' can be achieved with our method.

4 CONCLUSIONS

In this article we present an approximation of the kernel function for image reconstruction from HXMT data and the pixelization scheme optimized for this accelerated DD method. The pixels are nearly squares with the same size that make up a grid suitable for convolution. However due to the spherical topology, it is impossible to tessellate the spherical surface with perfect squares of the same size without gaps or overlaps. The error of the sizes of the pixels is propagated through the results of numerical integrations. The distortions of the pixels cause an error in the distance between any two pixels, which eventually propagates to the numerical convolutions. These errors will not be accumulated or magnified through iterations because each iteration of the DD method is directly based on the modulation equation (Li & Wu 1993). The efficiency of the spherical tessellation is not optimal, since two adjacent tessellae at the same latitude overlap each other. This is a trade-off for better regularity of the pixelization scheme (with better uniformity and better orthogonality). Extra computational costs are brought by the overlaps, as shown in Table 2.

| Tessella size  | Relative extra computational costs |
|---------------|-----------------------------------|
| $22.5^\circ \times 22.5^\circ$ | 20.3% |
| $15^\circ \times 15^\circ$   | 14.5% |
| $11.25^\circ \times 11.25^\circ$ | 11.1% |
| $9^\circ \times 9^\circ$    | 10.0% |
| $5.625^\circ \times 5.625^\circ$ | 6.0% |

With simulated data we demonstrate that our proposed method works for both equatorial data and data observed in a high latitude area. We can obtain a reconstructed image from the given data in several minutes on ordinary desktop PCs.

Since we are focused on the acceleration of the DD method, some other topics are not discussed, such as noise suppression, as well as treatment of faint and extended sources, which are also necessary for image reconstruction with HXMT data. For example, in Figure 11 we adjust the SNR of the observed images so that images using the model images can be calculated, although a number of the sources in the INTEGRAL 7-year catalog are actually too faint for a single phase of the HXMT all-sky survey. In addition we use known backgrounds as background constraints in the DD method, which is usually too idealized to achieve in practice.

The acceleration technique we propose in this article is necessary and sufficient for our future work. It serves as a tunable underlying engine of the flexible DD method.
Acknowledgements This work was supported by the National Natural Science Foundation of China (Grant Nos. 11173038 and 11103022), and also by the Tsinghua University Initiative Scientific Research Program (Grant No. 20111081102).

Appendix A: QUATERNION AND RELATED OPERATIONS

In a 3-D Cartesian coordinate system, a quaternion is defined as

$$q = a + bi + cj + dk,$$  \hspace{1cm} (A.1)

where $a$ is its scalar part and $bi + cj + dk$ is its vector part.

A quaternion $q = a + bi + cj + dk$ is a unit quaternion if and only if $a^2 + b^2 + c^2 + d^2 = 1$. A unit quaternion is used to formulate a rotation performed on a rigid body in 3-D space. Given a unit quaternion

$$q = a + bi + cj + dk = \cos \frac{\alpha}{2} + \sin \frac{\alpha}{2} (\cos \theta \cos \phi i + \cos \theta \sin \phi j + \sin \theta k),$$  \hspace{1cm} (A.2)

$q$ then indicates the right-handed rotation around vector $\begin{pmatrix} \cos \theta \cos \phi \\ \cos \theta \sin \phi \\ \sin \theta \end{pmatrix} = \cos \theta \cos \phi i + \cos \theta \sin \phi j + \sin \theta k$ by angle $\alpha$.

The conjugate of quaternion $q = a + bi + cj + dk$ is defined as

$$\bar{q} = a - bi - cj - dk.$$  \hspace{1cm} (A.3)

If the quaternion $q$ is a unit quaternion then its conjugate $\bar{q}$ is equivalent to its inverse $q^{-1}$.

Given the equation of basis elements $i$, $j$ and $k$ of quaternions as well as 3-D vectors

$$i^2 = j^2 = k^2 = ijk = -1,$$  \hspace{1cm} (A.4)

the multiplication of quaternions and 3-D vectors can be calculated distributively.

Let $q$ and $v$ be a unit quaternion and an arbitrary 3-D vector respectively, then the multiplication $qvq^{-1}$ yields the vector we obtain on rotating $v$ as described by $q$.

Appendix B: POINT-IN-SQUARE-ON SPHERE PROBLEM AND SPHERICAL-RAY-CASTING ALGORITHM

The point-in-square-on-sphere (PISOS) problem asks whether a given point on the spherical surface lies inside, outside or on the boundary of a spherical square. A spherical square is a regular spherical quadrilateral which has four equal sides and four equal angles. Each side is the arc of a great circle passing two adjacent vertices, and each angle is formed by the tangents of two adjacent sides.

The PISOS problem arises when we try to decide which observed data or pixel lies in which sky region. This problem is a special case of a PIP (Point-in-polygon) problem in computational geometry, which can be tackled by the ray-casting algorithm. According to the algorithm, one can find whether a given point is inside or outside a polygon by testing how many times a ray, starting from any known point inside the polygon and going towards the given point, intersects the edges of the polygon. If the number of intersections is odd the point is outside, but if the number is even it is inside.

We designed a more specific method, the spherical-ray-casting algorithm, to solve the PISOS problem. Assuming the spherical square is smaller than a semispherical surface, the PISOS problem reduces to the following problems:
(1) Find intersections of two given great circles on a unit sphere. Either great circle is defined by two points on the unit spherical surface.

This problem is reduced to finding the normal vector of a plane defined by two points on a unit spherical surface as well as the origin. For example, let \( p_1, p_2, p_3 \) and \( p_4 \) be points on the unit sphere \( x^2 + y^2 + z^2 = 1 \), where \( n_1 \) is the normal vector of the plane \( p_1, p_2, p_3 \) and \( n_2 = \frac{p_2 \times p_4}{|p_2 \times p_4|} \). If \( q \) is the intersection of plane \( p_1, p_2, p_3 \) and \( p_2, p_4 \), vector \( q \) is perpendicular to both \( n_1 \) and \( n_2 \) and thus any vector on the plane \( n_1, n_2 \); therefore \( q \) is the normal vector of this plane, defined by \( q = \frac{n_1 \times n_2}{|n_1 \times n_2|} \).

Let \( \theta_1, \phi_1 \) and \( \theta_2, \phi_2 \) be the altitudes and azimuthal angles of two points on the unit sphere, and \( \theta_n \) and \( \phi_n \) be those of their normal vector. We have

\[
\begin{align*}
\cos \theta_n \cos \phi_n \cos \phi_1 + \cos \theta_n \sin \phi_n \cos \theta_1 \sin \phi_1 + \sin \theta_n \sin \theta_1 &= 0, \\
\cos \theta_n \cos \phi_n \cos \phi_2 + \cos \theta_n \sin \phi_n \cos \theta_2 \sin \phi_2 + \sin \theta_n \sin \theta_2 &= 0,
\end{align*}
\]

hence,

\[
\begin{align*}
D &= \sqrt{\cos^2 \theta_1 \sin^2 \theta_2 + \sin^2 \theta_1 \cos^2 \theta_2 - 2 \cos \theta_1 \cos \theta_2 \sin \theta_1 \sin \theta_2 \cos(\phi_1 - \phi_2)}, \\
\tan \theta_n &= \frac{\cos \theta_1 \cos \theta_2 \sin(\phi_2 - \phi_1)}{D}, \\
\sin \phi_n &= \frac{\sin \theta_1 \cos \theta_2 \cos \phi_2 - \cos \theta_1 \sin \theta_2 \cos \phi_1}{D}, \\
\cos \phi_n &= \frac{\cos \theta_1 \sin \theta_2 \sin \phi_1 - \sin \theta_1 \cos \theta_2 \sin \phi_2}{D},
\end{align*}
\]

(2) Find whether a given minor arc of a great circle contains a point on the same great circle. Let \( p_0 \) and \( p_1 \) be the endpoints of a minor arc on the great circle and let \( q \) be a point on the great circle. The point \( q \) is on the given arc if and only if the sum of geodesic distances from \( q \) to \( p_0 \) and to \( p_1 \) is equal to the length of the arc.

References

Crittenden, R. G., & Turok, N. G. 1998, arXiv:astro-ph/9806374
Doroshkevich, A. G., Naselsky, P. D., Verkhodanov, O. V., et al. 2005, International Journal of Modern Physics D, 14, 275
Górski, K. M., Hivon, E., Banday, A. J., et al. 2005, ApJ, 622, 759
Han, D., Jin, Y., Wu, M., & Zhang, Z. 2010, Science in China G: Physics and Astronomy, 53, 36
Krivonos, R., Revnivtsev, M., Lutovinov, A., et al. 2007, A&A, 475, 775
Li, T.-P. 2007, Nuclear Physics B Proceedings Supplements (proceedings of the Third International Conference on Particle and Fundamental Physics in Space), 166, 131
Li, T.-P., & Wu, M. 1993, Ap&SS, 206, 91
Li, T.-P., & Wu, M. 1994, Ap&SS, 215, 213
Lu, Y., Zhang, W., Qu, J., et al. 2010, Science in China G: Physics and Astronomy, 53, 31
Lucy, L. B. 1974, AJ, 79, 745
Richardson, W. H. 1972, Journal of the Optical Society of America (1917-1983), 62, 55
Shen, Z. J., & Zhou, J. F. 2007, High Energy Physics and Nuclear Physics, 31, 1016
Tegmark, M. 1996, ApJ, 470, L81