Released: 15 September 2025

Nancy Grace Roman Space Telescope
Technical Report Roman-STScI-000708

The Roman tessellation of the celestial sphere

Fadda, D., Schlafly, E., Ferguson, H., Casertano, S., Desjardins, T.

Abstract

The Roman Telescope has adopted a tessellation scheme of the celestial sphere to obtain mosaics stored in the Roman archive. This tessellation partitions the sphere in 4058 sky tiles whose centers and limits in declination are defined by a Healpix double tessellation scheme with $N_{side} = 13$. The resulting sky tiles are approximately rectangular with similar area (≈ 10 sq. degrees). Since the sky tile centers are arranged in rings with the same declination, the limits in right ascension are defined as the mid values of the right ascensions of two consecutive centers lying on the same declination ring. Each projected sky tile is further divided in square patches of approximately $4\farcm6 \times 4\farcm6$ (i.e. $5K \times 5K$ pixels with a pixel size of $0\farcs055$), called sky cells, which include small overlaps between contiguous elements. Sky cells are obtained as gnomonic north-oriented projections centered at the center of the sky tile of the images obtained in the associated sky region. Such mosaics, conserved as individual files in the Roman archive, allow one to have a fast coadding of the data as well as an efficient source extraction and catalog production. Files belonging to the same sky tile can be seamlessly combined by users to obtain larger mosaics with the same projection center.

Contents

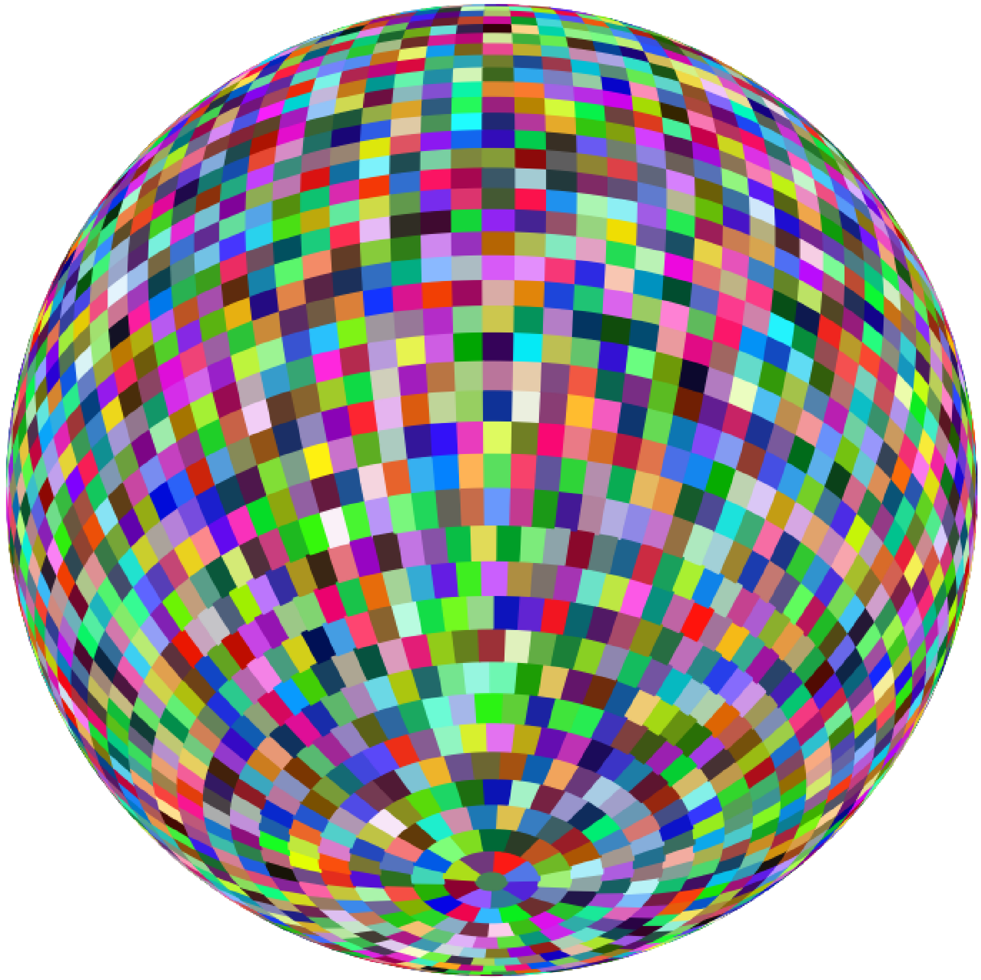

Figure 1: Tessellation of the sphere with 4058 sky tiles described in the report. For clarity, adjacent tiles are shown with different colors. Each sky tile is approximately rectangular, except for the circular tiles at the poles, and covers about 10 square degrees.

1 Introduction

The Roman telescope will perform wide-area surveys with its Wide-Field Instrument (WFI) as well as several general observations in any direction of the sky, including parallel observations when using the Coronagraph Instrument. In the course of its lifetime Roman will therefore cover extended parts of the celestial sphere. For archiving and ease of use of the WFI data, it is useful to develop a tessellation standardized for all surveys and spanning the whole sphere. Such tessellation allows building a continuous image of large areas by combining individual pointings and makes easier to stack images taken in different periods to increase the signal to noise ratio. Moreover, since source catalogs will be generated and

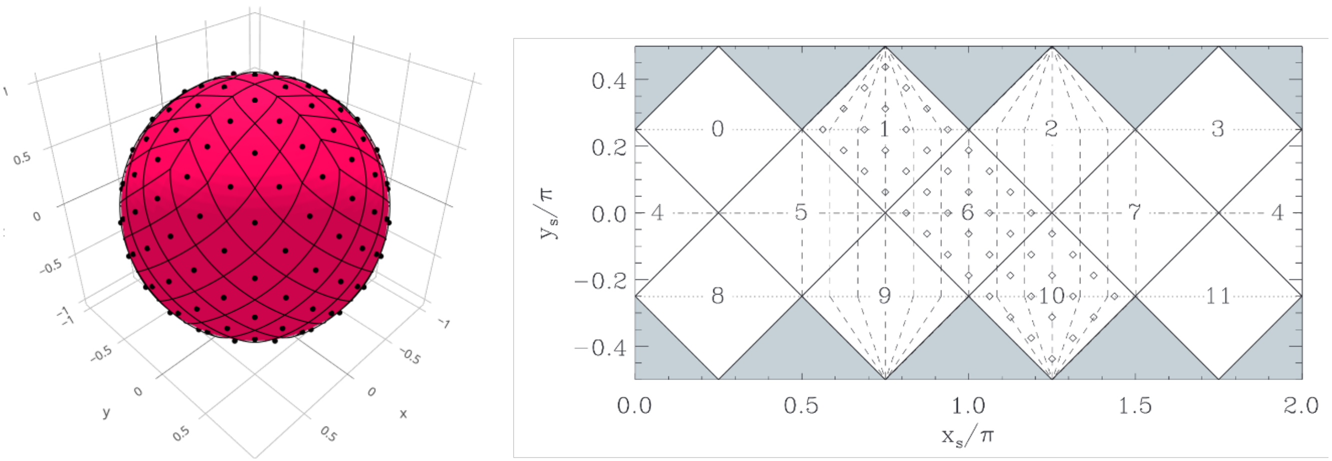

Figure 2: HEALPix tessellation of the sphere by dividing each of the 12 diamond sides into 4 parts ($N_{side} = 4$) and its pseudo-cylindrical mapping to the plane.

archived by the Roman mission, such a tessellation should provide a simple way to produce catalogs with unique entries.

The typical approach of optical and infrared surveys is to store mosaics of sky regions obtained with a gnomonic projection, a tangential projection with origin in the center of the sphere. Such regions are usually large enough to contain extended galaxies or other objects with small predefined distortions, have minimal overlaps, and can be stored in smaller patches which can be then retrieved and seamlessly combined into larger mosaics without requiring any resampling.

Several approaches have been used in previous optical and infrared surveys. A few surveys such as PanSTARRS and the upcoming LSST used the so-called RING (V3) method [1]. However, this method is not a well defined tessellation since it has been shown to leave gaps around the polar regions. Alternatively, surveys done with the Euclid space telescope opted for tessellations based on HEALpix (Euclid Collaboration, 2022). Since HEALpix naturally defines rhombic regions on a sphere (see Figure 2), each projected HEALpix tile is divided in square sub-regions that follow the diagonal contours in a non-optimal way in order to obtain rectangular north-oriented regions.

For Roman we chose a method that combines the advantage of having rectangular tiles organized in rings with the elegance of the HEALpix tessellation. To reach this objective we defined a tessellation described in section 2.2 that is based on the so-called "double pixelization", an extension of HEALpix proposed by Calabretta & Roukema (2007).

[1] `https://outerspace.stsci.edu/display/PANSTARRS/PS1+Sky+tessellation+patterns`

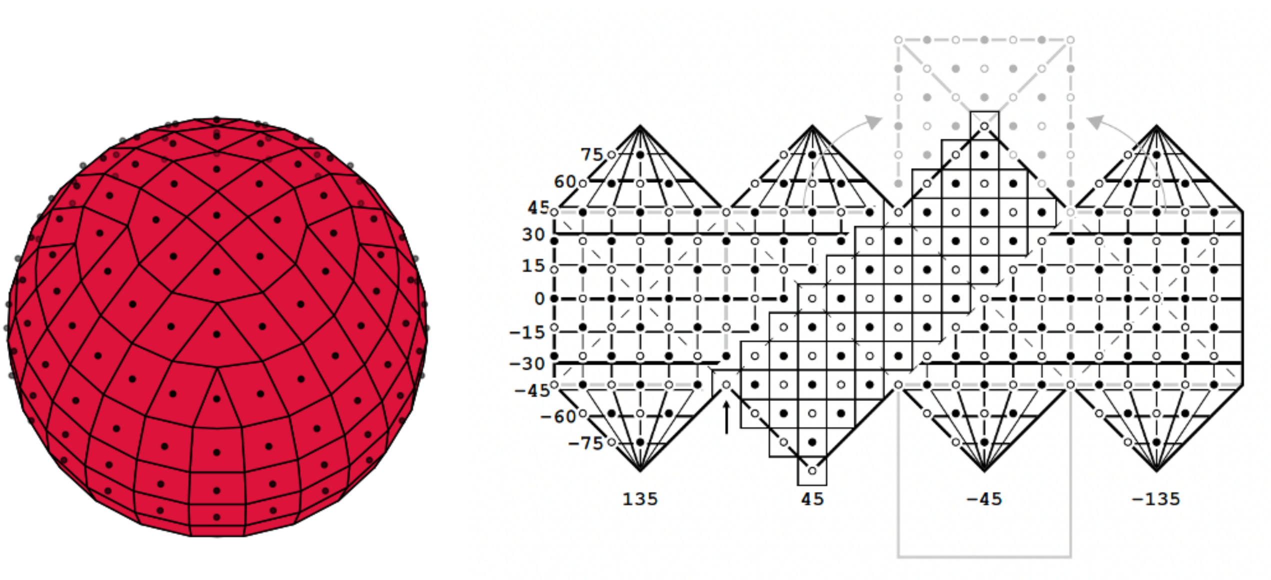

Figure 3: Double pixelization tiling of the sphere with $N_{side} = 3$. The right panel shows how the tiles are projected into the plane on squares oriented along parallels of the sphere.

2 Method

To explain the method chosen for the Roman tessellation, it is useful to introduce some basic concepts about HEALpix (Gorski et al., 2005) and the derivative method proposed by Calabretta & Roukema (2007).

2.1 HEALpix and double pixelization

An elegant way to achieve an equal-area isolatitude tessellation of the sphere is to use the HEALpix projection (Gorski et al., 2005). A sphere can be partitioned with a regular rhombic grid consisting of N_θ rings containing the centers of the rhombi subdivided in N_ϕ equatorial cuts. There will be $N_\theta N_\phi$ tiles each with area equal to $4\pi/(N_\theta N_\phi)$ steradians. Each side of a rhombus can be subdivided in N_{side} parts to generate a hierarchical tessellation. The optimal choice in cosmic microwave background surveys is to adopt $N_\phi = 4$ and $N_\theta = 3$ which subdivides the sphere in 12 diamond-shaped spherical polygons and leads to a pixelization with minimal distortion. Each rhombus can be subdivided in N^2_{side} identical rhombi by dividing each side into N_{side} parts to reach the desired resolution (see Fig. 2). The area of each tile is therefore $4\pi/(N_\theta N_\phi N^2_{side}) = \pi/(3N^2_{side})$ steradians. Therefore, to have tiles with a given angular size α, one has to subdivide each rhombus $N_{side} = \sqrt{\pi/3}/\alpha$ times.

The projection is equivalent to a cylindrical equal area projection in the central part. In the polar region the projection changes. To compute the angle where the projection changes one can notice that the area of the polar cup at this angle corresponds to 1/6 of the area of the sphere. We assume in the following that the latitude angle θ is in the interval $[-\pi/2, \pi/2]$. So, if θ_x is the transition latitude angle $(-\pi/2 \leq \theta \leq \pi/2)$, this means that the area of the polar cup is $2\pi(1 - sin\theta_x) = 4\pi/6$ and therefore:

$$\theta_x = \sin^{-1}(2/3) \approx 41^\circ.81 \tag{1}$$

Points on the sphere with spherical coordinates (ϕ, θ) can be mapped to points in the plane with coordinates (x, y) using the following pseudo-cylindrical transformation.
A simple cylindrical transformation is used in the equatorial region, $|\sin(\theta)| \leq 2/3$:

$$x = \frac{\phi}{\pi} \tag{2}$$

$$y = \frac{3}{8}\sin\theta \tag{3}$$

While, in the polar regions, $|\sin\theta| > 2/3$:

$$x = \frac{\phi}{\pi} - (|2 - \sqrt{3(1-\sin\theta)}| - 1)\left(\left(\frac{\phi}{\pi} \mod \frac{1}{2}\right) - \frac{1}{4}\right) \tag{4}$$

$$y = \frac{2 - \sqrt{3(1-\sin\theta)}}{4} \tag{5}$$

We use, in this case, the (x,y) coordinates normalized by π as shown in Figure 2.

Conversely, coordinates from the plane can be converted to spherical coordinates with the following equations.
In the equatorial region, $|y| \leq \frac{1}{4}$:

$$\phi = \pi x \tag{6}$$

$$\sin\theta = \frac{8}{3}y \tag{7}$$

In the polar regions, $\frac{1}{4} < |y| < \frac{1}{2}$:

$$\phi = \pi x - \pi\frac{|y| - \frac{1}{4}}{|y| - \frac{1}{2}}\left(\left(x \mod \frac{1}{2}\right) - \frac{1}{4}\right) \tag{8}$$

$$\sin\theta = \left(1 - \frac{1}{3}(2 - 4|y|)^2\right)\frac{|y|}{y} \tag{9}$$

with x restricted to:

$$\left|\left(x \mod \frac{1}{2}\right) - \frac{1}{4}\right| < \frac{1}{2} - |y| \tag{10}$$

The main drawback of this tessellation for storing imaging data is the unusual orientation of the tiles (diamonds oriented at 45°). A way to obtain tiles oriented along the parallel and meridians of the sphere has been proposed by Calabretta & Roukema (2007) with the so-called "double pixelization." This is achieved by adding a tile between each pair of tiles at the same latitude and two extra tiles at the two poles. The number of tiles increases from $12N_{side}^2$ to $24N_{side}^2 + 2$. Each tile has an area of $\pi/(6N_{side}^2)$ (half of the HEALpix tile) with the exception of the 8 pixels inside the corners which have an area 3/4 of the normal tile. The missing area is contained in the two extra tiles at the poles (see Fig. 3). It is possible to use the same indexing as in the HEALpix case by doubling the HEALpix indices and incrementing them by a unity so that they run from zero to $24N_{side}^2 + 1$ with the first and last at the poles.

2.2 Roman tessellation

The direct definition of the HEALpix double pixelization only makes sense when reprojecting the sphere using the HEALpix projection since this allows one to map the spherical diamonds into squares. In our case we use a gnomonic projection to generate projected tile mosaics. So, using the tiles defined with HEALpix will result in rhombic rather than rectangular shapes especially towards the poles (see Fig. 3). To define the Roman tessellation we therefore make use of the centers and limits in declination defined by the HEALpix double pixelization. We further define limits in right ascension as the middle point between two consecutive centers at the same latitude. In this way we obtain a very simple tiling of the sphere with most of the tiles approximately rectangular and of similar size (see Fig. 4). Distortions from the rectangular shape are more noticeable towards the poles so that the rectangular mosaic containing the projected tile is only partially covered. The resulting tiles are also completely symmetric with respect to the meridians of the sphere.

3 Technical details

3.1 Definitions

We will see in the next sections that a large tile is convenient to study galaxies and galaxy systems. However, it is more efficient to store smaller mosaics that allow one to retrieve data from the archive more easily and also facilitate the production of source catalogs. For this reason, each tile is subdivided in smaller square patches. In this section we define the terms used in the tessellation and the creation of mosaics. A graphical illustration is given in Fig. 5. We define:

- **Sky tile** as the region of the sky defined by the tessellation. Each sky tile is delimited in declination and right ascension and its center is used as the reference center for

Figure 4: The Roman tessellation with $N_{side} = 3$.

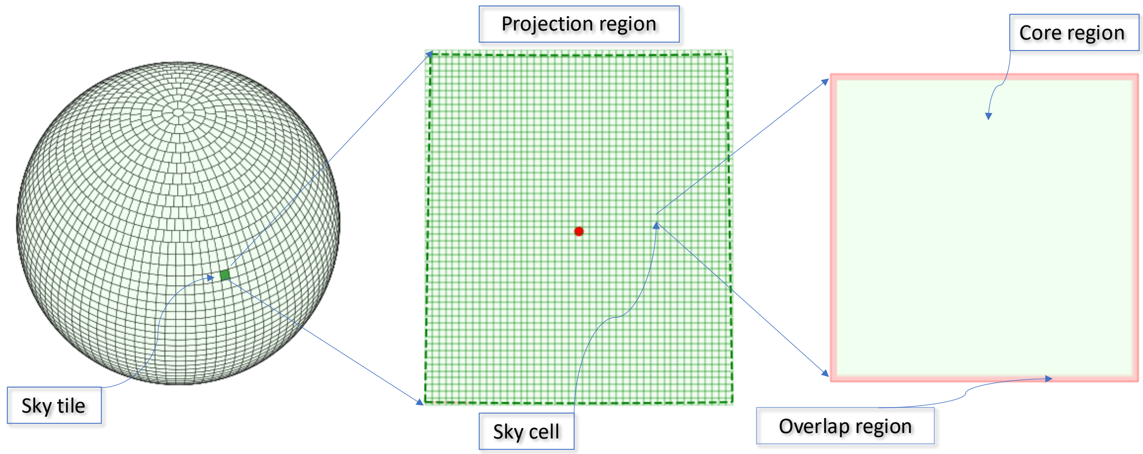

Figure 5: A sky tile (#1000 on the left) is projected with a gnomonic projection into a projection region (middle). The projection region is then subdivided in several square mosaics (sky cells, right) which consist of a core and an overlap region which is shared by contiguous cells. The overlap region allows the detection of sources on the borders of the cell and it is not considered when combining adjacent sky cells to obtain larger mosaics.

the gnomonic projection. Each sky tile is indexed according to the Healpix double pixelization scheme. In Fig. 5 we selected for illustration the tile # 1000 which is marked with dark green on the sphere.

- **Projection region**, middle panel of Fig. 5, as the sky tile projected on a plane tangential to the celestial sphere centered on the center of the tile. We consider here a gnomonic projection, a tangential projection with origin in the center of the sphere, in equatorial coordinates. The limits of the tile are marked as a dashed green line.

- **Sky cell** as the square mosaic (level 3 product) conserved in the Roman archive, part of the projection region (left side of Fig. 5). Each projected sky tile is divided in sky cells with a size of approximately $4\rlap{.}'6 \times 4\rlap{.}'6$ which roughly corresponds to a single Roman/WFI detector. Such cells contain a number of pixels small enough to efficiently extract sources and retrieve archived data using a native pixel or a small oversampling factor. The central sky cell is centered on the projection center.

- **Core and overlap region of a cell.** The core region of a sky cell is the set of pixels unique to the sky cell (which comes from the direct partition of the projected sky tile). A small overlap region with adjacent cells (width of 5 arcsec) allows the detection of slightly extended sources which fall on the border of the cell. When combining contiguous cells to obtain larger mosaics this part is not considered.

3.2 Choice of pixel size and sky tile size

3.2.1 Pixel size

Once a tessellation scheme is adopted, one has to decide the optimal size of the sky tile and the size of the pixel of the projections. The size of the pixel is dictated by the pixel size of the detector, which for Roman is 0.11 arcsec. In order to exploit the dithering to reconstruct the point-spread function of point sources, it is advisable to have at least a pixel half the size of the original detector. Moreover, even with shallow coverage, it is convenient to reproject the original images on a pixel with size smaller than the native one. In fact, the orientation of the camera pixels can be different from the one of the projection and the reprojection can result in a loss of resolution. For this reason we choose a default pixel size of 0.055 arcsec. However, in cases where a minimal loss of resolution is not an issue, since the number of pixels for each cell is even it is easy to revert to a native pixel size while maintaining the same size of tiles and cells.

3.2.2 Distortion

The size of the sky tile depends on considerations linked to the size of the detector array (i.e. the size of a pointed observation) and the size of extended objects, as well as the amount of the distortion introduced by the projection.

We consider here a gnomonic projection, a typical choice in astronomy, that consists in projecting the sphere on a tangential plane with the centre of the sphere as reference point for the projection. The point where the tangent plane touches the sphere is called projection center. The gnomonic projection is neither conformal nor equal-area. Shape, area, and distance distortions increase with distance from the center. It is therefore important to limit the size of the tile to have negligible distortion.

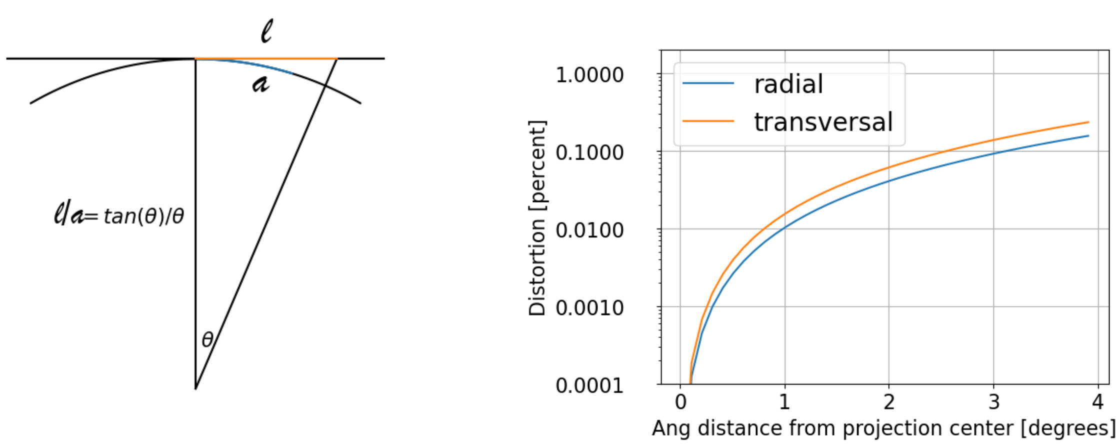

Figure 6: Distortion of the distance from the projection center as a function of the angular separation from the projection center for a gnomonic projection (tangential plane). The sketch in the left panel explains the formula used to compute the radial distortion. For angles less than 2°, i.e. tiles with sides shorter than 4°, the distortion is less than 0.1%.

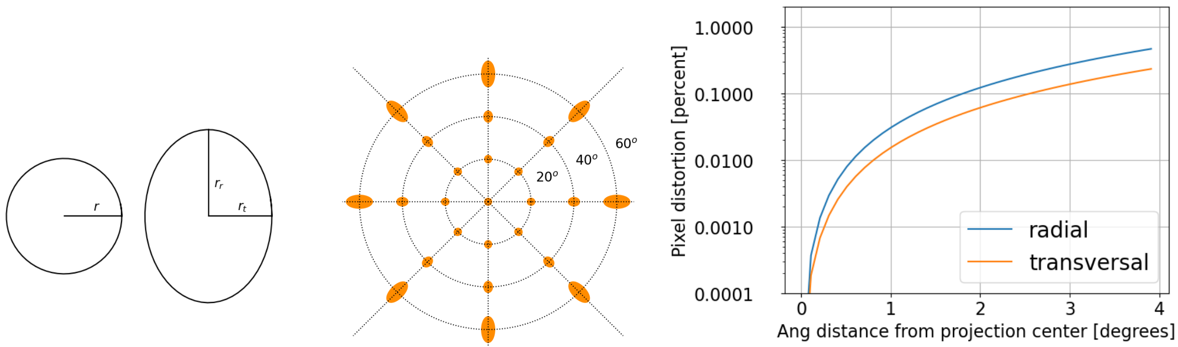

Figure 7: Circles on the sphere are distorted into ellipses when projected on the tangential plane. The middle panel shows the Tissot's indicatrix for a gnomonic projection. For angles lower than 2°, i.e. tiles with sides shorter than 4°, the distortion in shape is less than 0.1%.

The projection stretches each radial arc of the sphere into a radial segment. As shown in Fig. 6, the ratio between a radial arc and its projection can be computed with simple trigonometry and it is equal to:

$$\frac{l_r}{a_r} = \frac{\tan(\theta)}{\theta}, \tag{11}$$

where θ is the angular distance from the projection center, a_r is the length of the arc on a circle passing through the projection center, and l_r the length of its projection. The length of a circle centered on the projection center is also stretched while projected. In this case, since the radius of the circle on the unit sphere is equal to $\sin(\theta)$ and the radius of the circle on the tangential plane is equal to $\tan(\theta)$, the ratio of the length of a projected arc l_t to that of the corresponding arc on the sphere a_t is:

$$\frac{l_t}{a_t} = \frac{1}{\cos(\theta)} \tag{12}$$

Consequently, distances across the sphere in the radial and transverse directions become bigger at larger angular separations. For angles lower than 2° the distortion in distance is less than 0.1%.

The projection also distorts shapes in different positions of the map. A good way to visualize the shape distortion is to compute the projection of an infinitesimal circle on the sphere in different locations, a technique called Tissot's indicatrix (see middle panel of Fig. 7). Because of the symmetry of the gnomonic projection, a circle is projected into an ellipse with the major axis aligned on the direction of the projection center. The distortion can be expressed as the deviations from the radius of the circle for the two axes of the ellipses. The distortion is maximal in the radial direction and minimal in the transverse direction.

The transverse component will be stretched with the same ratio as the total circumference, so:

$$\frac{r_t - r}{r} = \frac{1}{\cos(\theta)} - 1 \tag{13}$$

The stretch of the radial component can be computed by simply differentiating the length of radial ratio $dl/da = d\tan(\theta)/d\theta$, obtaining:

$$\frac{r_r - r}{r} = \tan^2(\theta) \tag{14}$$

As shown in Fig. 7, the maximal distortion of a pixel, the smallest element of a mosaic, is less than 0.1% for angular distances lower than 2°.

3.2.3 Galaxies and continuous field of view

Because of the wide field of view of the WFI camera ($0^{\circ}.4 \times 0^{\circ}.8$), it makes sense to generate mosaics that are at least as large. It is advantageous to have mosaics with the same projection center for many studies of galaxies, such as clusters of galaxies or even surveys. In the case of extended sources, having a source broken into several pieces can be a serious hurdle since one has to reproject many adjacent fields to have a complete image of a source. The right panel of Figure 8 shows the probability of imaging a galaxy the size of the Andromeda galaxy (∼50 kpc) in one sky tile when the galaxy is at different redshifts. Since a face-on galaxy with angular diameter D can be entirely imaged if its center lies at least half diameter inside the FOV, the probability to cover such galaxy with a FOV of side s can be simply computed as the ratio between the area of the square containing the centers (of side $s - D$) and that of the total FOV (side s), i.e.:

$$P = (s - D)^2/s^2. \tag{15}$$

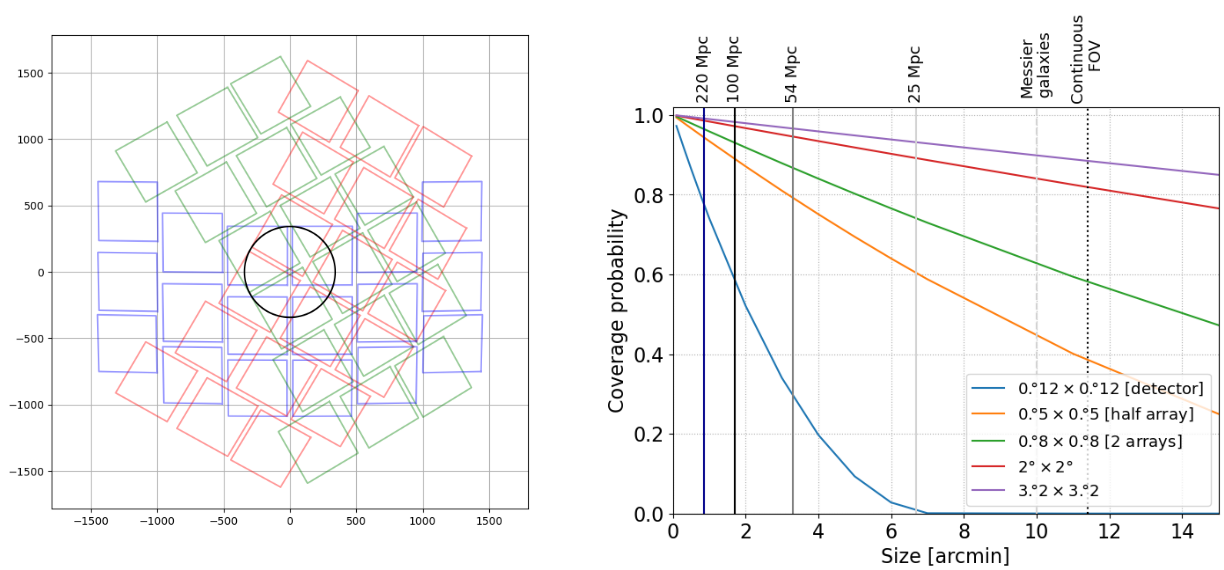

Figure 8: *Left:* FOV of the WFI rotated at three angles corresponding to three different observation times. Units are arcseconds. The minimal dithering to cover the gaps is shown in one case (blue contours). The continuous WFI field of view is traced with a black circle. *Right:* Probability that Andromeda–size galaxies (∼50 kpc) at different distances, Messier-type galaxies, or the continuous FOV of WFI lie entirely inside sky tiles with different sizes.

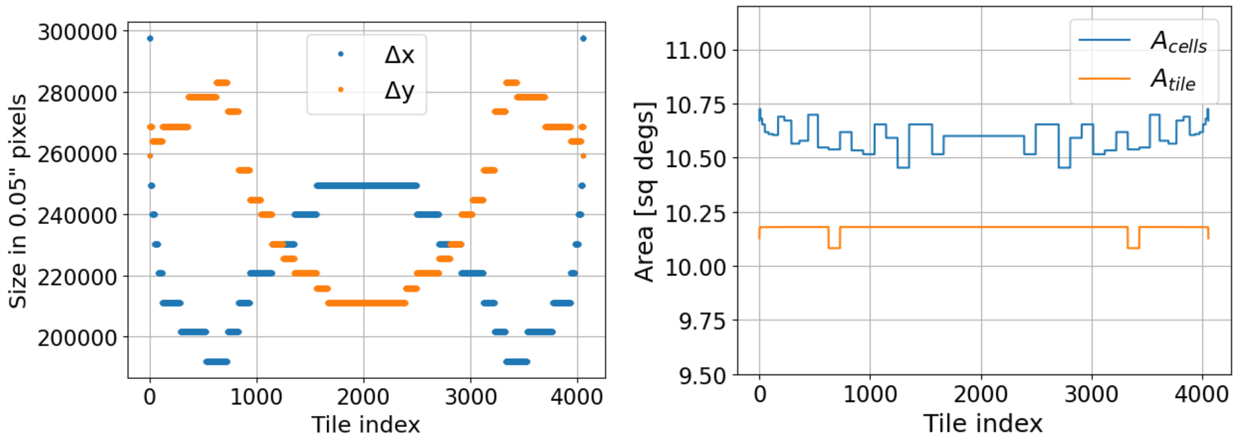

Figure 9: *Left*: Sides in pixel units of the smallest rectangular mosaic containing the gnomonic projection of the sky tile. *Right*: Area of the projected sky tiles (orange line) and of the sky cells needed to cover the tile (blue line). Since the projected sky tiles are not perfect rectangles, the sky cells cover approximately 4% more than the tile area.

Using a $3.^\circ2 \times 3.^\circ2$ tile, a galaxy of less than 12 arcmin size is available about 90% of the time in a single sky tile. Messier objects are typically smaller than 10 arcmin, so they lie in a unique $3.^\circ2 \times 3.^\circ2$ sky tile more than 90% of the time. Smaller size tiles will break extended objects much more frequently.

The left panel of Fig. 8 shows the field covered by a pointed observation with a random orientation of the field (i.e. a random date of observation). Taking into account all the possible roll angles, a pointed observation will always cover at an homogeneous depth a circular field of $11'.4$ diameter. Such a field will be entirely lying inside a $3.^\circ2 \times 3.^\circ2$ sky tile approximately 90% of the time (see dotted line in the right panel of Fig. 8).

From these results it is clear that a sky tile size between 3 and 4° is sufficiently big for various science cases and small enough to have negligible distortion effects on the most peripheral pixels. Hence we adopt a size of $3.^\circ2$, i.e. a tile with an area of approximately 10 square degrees, which covers the continuous viewing zone 90% of the time.

3.3 Mosaics

In this report we refer to mosaics as images obtained by combining several observations, each one consisting of an image from the WFI detectors. Roman images are in the equatorial coordinate system.

For a single sky tile we obtain a mosaic by combining all the WFI images overlapping the tile. Although the areas of the sky tiles are very similar by construction, their shape varies going from the equator to the poles (see Fig. 9). The projected sky tiles have a maximum x-size of 297409 pixels, while the maximum y-size is 282252. So, by using square patches with sides of 4800 pixel as sky cells, it is possible to cover them with a maximum of 62 cells in the x-direction and 59 in the y-direction. To allow the extraction of point- and slightly extended sources, it is useful to have overlap between adjacent sky cells of approximately 10 arcsec, which means adding an overlap region around each cell 5 arcsec wide (100 pixels).

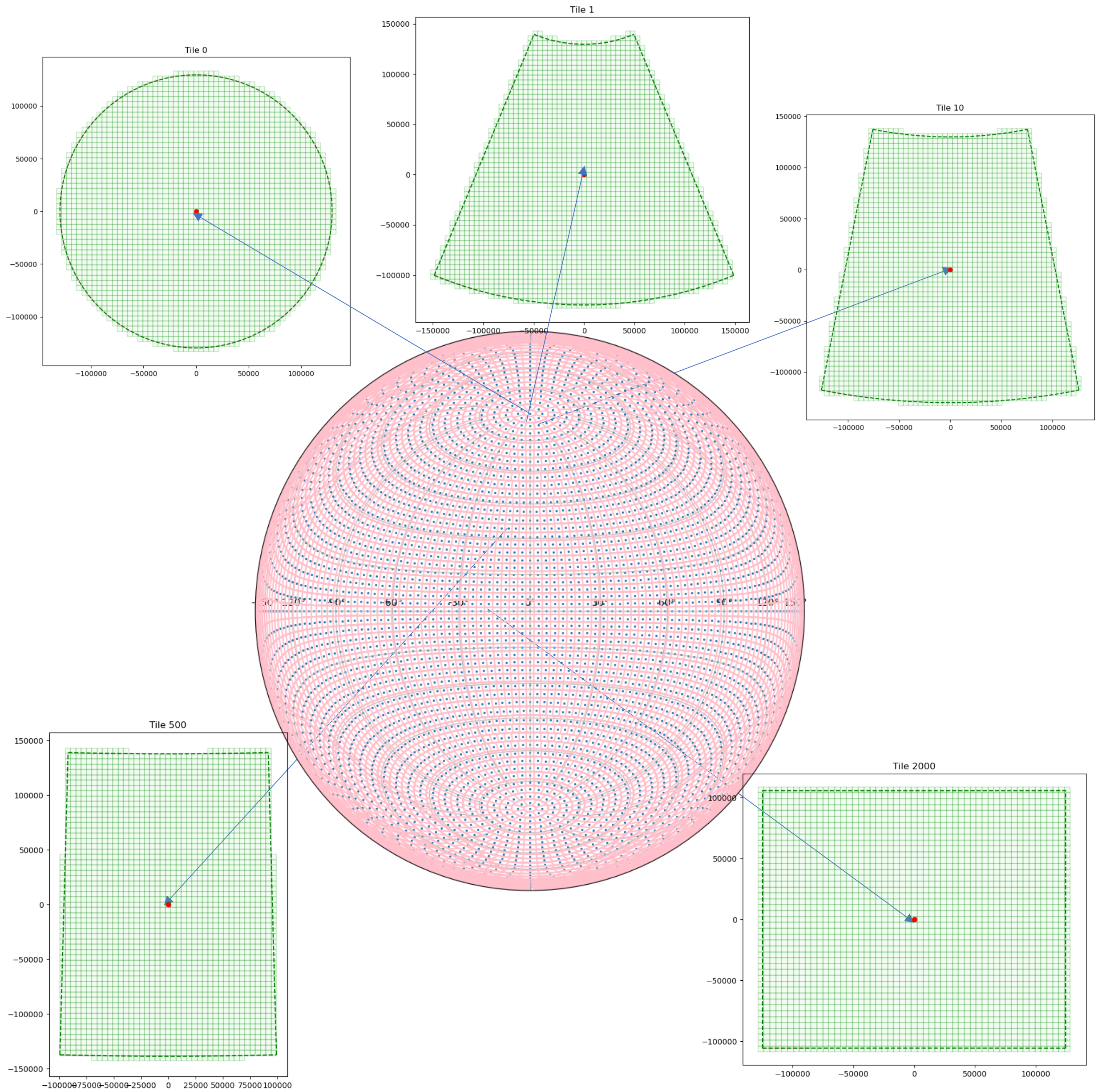

Figure 10: Lambert projection of the tessellation of the celestial sphere with projection centers shown as blue dots. A few projection regions obtained as gnomonic projections of sky tiles at different latitudes are shown. The shape of a projected tile is generally close to a rectangle except for high latitude tiles which have a more trapezoidal shape and the polar ones which are circular. The limit of each sky tile is marked with a dashed green line. Each mosaic is further divided into smaller square mosaics (sky cells) to facilitate the archival process and source extraction. The coordinates have the origin in the projection center (red dots in the projections) and pixel size units ($0\farcs055$).

Therefore, each sky cell will have a file size of 5000×5000 pixels.

Fig. 10 shows how the shape of the projection regions changes with the latitude. The polar sky tiles are circular. The sky tiles close to the poles are trapezoidal. As one goes farther from the poles, sky tiles are approximately rectangular. Each projected region of a sky tile is divided in square sky cells used to archive the data.

3.4 Image selection and source detection

For each sky tile, we will select all the level 2 images that have at least one corner inside the limits in R.A. and Declination of the sky tile. After computing background offsets for all these files to match their background, subsets of them will be selected to obtain mosaics for each sky cell covering the sky tile. Sky cells are defined as square images with pixels of 0.055 arcsec sides which partition the projected sky tile. The central sky cell is centered on the projection center of the sky tile. Each sky cell consists of a core region of 4800×4800 pixels which is a unique part of the projected sky tile. To allow detection of sources on the border of each sky cell, the cell is extended by 5 arcsec to overlap contiguous cells. With a pixel of 0.05 arcsec, this corresponds to a border of 100 pixels and results in a square sky cell with a side of 5000 pixels. Such a border is called *overlap region* and it is drawn in red in Fig. 5.

The metadata of each cell will contain the limits in R.A. and Declination of the relative tile. Taking into account this information it will be possible to avoid duplications in the source catalog by retaining only the detected sources whose center falls inside the tile. To be precise, in order to completely avoid duplications one has to consider semi-open intervals, i.e. a source with coordinates α, δ can be associated to a sky tile with limits in R.A. α_1, α_2 and declination δ_1, δ_2 only if:

$$\alpha_1 \leq \alpha < \alpha_2 \tag{16}$$

$$\delta_1 \leq \delta < \delta_2. \tag{17}$$

When combining sky cells in the same tile to obtain extended sky maps, one will only consider the central 4800×4800 pixels of each sky cell.

3.5 Names of sky cell files

Each sky cell corresponds to a file in the Roman archive. The file name will contain a string which will allow the user to rapidly identify the sky tile to which it belongs and its position with respect to the center of projection.

The sky tile will be identified by the center of projection in R.A. and Declination expressed in degrees (rounded to the degree). Finally, to identify the position of the sky cell, two integers will identify its x and y position relative to the center of projection in units of sky cells.

The string containing this information will have the following format:

$$\alpha\alpha\alpha\mathbf{s}\delta\delta\mathbf{x}\xi\xi\mathbf{y}\upsilon\upsilon,$$

where:

- $\alpha\alpha\alpha$ is a three digit integer (0-360) corresponding to the right ascension of the sky tile center expressed in degrees;

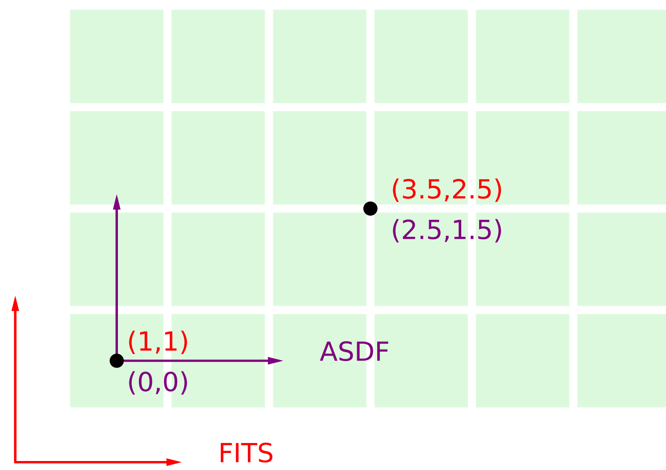

Figure 11: Conventions for pixel coordinates are different in standard FITS files and in the ASDF skycell files. The center of the pixel in the lower left corner has coordinates (1,1) in the case of FITS files (red axes), while in the case of the ASDF files used for the Roman skycells (purple axes) it has coordinates (0,0). In the example with an array of 6×4 pixels, the center of the array has coordinates (3.5,2.5) in the case of FITS files while it has coordinates (2.5,1.5) for ASDF skycell files.

- $\delta\delta$ is a two digit integer (0-90) reserved to the declination of the sky tile center expressed in degrees;
- **s** is the sign, that could be p or m, for positive or negative, respectively;
- $\xi\xi$ and $\upsilon\upsilon$, are sky cell coordinates in units of cell size whose value at the center of projection is (50,50);
- **x, y** are letters indicating x and y coordinates, respectively.

As an example, a sky cell that lies two cells to the left and three cells up from the center of the tile and belongs to a sky tile with coordinates $\alpha = 100°$ and $\delta = -20°$ will be labeled as:

100m20x48y53.

3.6 Metadata of sky cell files and pixel convention

To define the astrometry in the skycells metadata we adopted the convention that identifies the origin of the pixel coordinates with the center of the lower left pixel of an array of pixels. As shown in Fig. 11, this convention differs from the FITS convention which assigns to the bottom left pixel center the coordinates (1,1).

For each skycell we computed the R.A. and Declination of the corners of the cell. To be precise, the metadata contain the coordinates of the external corners of the corner pixels, not those of their centers.

The metadata file is available through CRDS at the URL: `https://roman-crds.stsci.edu/browse/roman_wfi_skycells_0004.asdf`.

From this file it is possible to generate the world-coordinate system (WCS) of a skycell and the projected tile with the following commands:

```
from astropy.modeling import models
from astropy import coordinates as coord
from astropy import units as u
from gwcs import wcs
from gwcs import coordinate_frames as cf
import roman_datamodels as rd

nskytile, nskycell = 1, 1443
with rd.open("roman_wfi_skycells_0004.asdf") as f:
    tile = f.projection_regions[nskytile]
    start, end = tile['skycell_start'], tile['skycell_end']
    cell = f.skycells[start:end][nskycell]
    meta = f.meta

ra0, dec0 = tile['ra_tangent'], tile['dec_tangent']
pix = meta['pixel_scale']/3600
nxy = meta['nxy_skycell']

# Astrometry of the skycell 1443 on tile 1
x0, y0 = cell['x_tangent'], cell['y_tangent']
pixshift = models.Shift(-x0) & models.Shift(-y0)
pixscale = models.Scale(pix) & models.Scale(pix)
tangent_projection = models.Pix2Sky_TAN()
celestial_rotation = models.RotateNative2Celestial(ra0, dec0, 180.)
det2sky = pixshift | pixscale | tangent_projection | celestial_rotation
detector_frame = cf.Frame2D(name="detector", axes_names=("x", "y"),unit=(u
                                    .pix, u.pix))
sky_frame = cf.CelestialFrame(reference_frame=coord.ICRS(), name='icrs',
                                    unit=(u.deg, u.deg))
wcs_cell = wcs.WCS([(detector_frame, det2sky),(sky_frame, None)])
icrs2xy_cell = wcs_cell.get_transform('icrs','detector')

# Astrometry of the projected tile 1
x0, y0 = tile['x_tangent'], tile['y_tangent']
pixelshift = models.Shift(-x0) & models.Shift(-y0)
pixelscale = models.Scale(pix) & models.Scale(pix) # 0.1 arcsec/pixel
tangent_projection = models.Pix2Sky_TAN()
celestial_rotation = models.RotateNative2Celestial(ra0, dec0, 180.)
det2sky = pixshift | pixscale | tangent_projection | celestial_rotation
detector_frame = cf.Frame2D(name="detector", axes_names=("x", "y"),unit=(u
                                    .pix, u.pix))
sky_frame = cf.CelestialFrame(reference_frame=coord.ICRS(), name='icrs',
                                    unit=(u.deg, u.deg))
wcs_tile = wcs.WCS([(detector_frame, det2sky),(sky_frame, None)])
icrs2xy_tile = wcs_tile.get_transform('icrs','detector')
```

Then, to transform an equatorial coordinate into pixel coordinate, or vice versa, one has to use the convention with initial pixel coordinates set to zero. For instance, the coordinates of the corners of a skycell and the position of a direction in the sky can be computed as follows:

```
# From x,y positions to ICRS coordinates
x, y = [ -0.5, 4999.5, 4999.5, -0.5], [ -0.5, -0.5, 4999.5, 4999.5]
a, d = wcs_cell(x, y)
# From RA, Dec [in degs] direction to x, y position in pixels
ra, dec = 10.5, 20.2
x, y = icrs2xy_cell(ra, dec)
```

Such coordinates should correspond to the coordinates of the corners precomputed in the metadata. Table 1 describes the fields in the metadata of skycell files.

3.7 From coordinates to skycell

In this section we show how to find the skycell covering a given direction of the sky.

```
from astropy.coordinates import SkyCoord
from astropy import units as u
import roman_datamodels as rd
import numpy as np

# Direction coordinates [Example]
c = SkyCoord('00h42m30s', '+41d12m00s', frame='icrs')

# Reading the file
with rd.open("roman_wfi_skycells_0004.asdf") as f:
    tiles = f.projection_regions
    skycells = f.skycells

    # Find the skytile
    ramin, ramax = tiles['ra_min'], tiles['ra_max']
    decmin, decmax = tiles['dec_min'], tiles['dec_max']
    ra, dec = c.ra.deg, c.dec.deg
    idtile, = np.where((ra >= ramin) & (ra < ramax) &
                        (dec <= decmax) & (dec > decmin))
    tile = tiles[idtile[0]]

    # Find the skycell
    start, end = tile['skycell_start'], tile['skycell_end']
    cells =  skycells[start:end]

centers = SkyCoord(ra=cells['ra_center']*u.degree,
                    dec=cells['dec_center']*u.degree, frame='icrs')
sep = c.separation(centers)
idmin = np.argmin(sep.arcsec)

# Index and name of the cell
print('Index: ', idmin, ' Cell filename :', cells['name'][idmin])
```

The code will print the index of the skycell and the name of the respective file.

Field	**Format**	**Content**	**FITS analogue**
meta			
nxy_skycell	i4	Number of pixels of a cell side	NAXIS1,2
skycell_border_pixels	i4	Number of pixels of a cell border	
pixel_scale	f4	Pixel size in arcsec	CDELT1,2
projection_regions			
index	i4	Skytile index	
ra_tangent	f8	Skytile center R.A.	CRVAL1
dec_tangent	f8	Skytile center declination	CRVAL2
ra_min	f8	$RA \geq RA_{min}$	
ra_max	f8	$RA < RA_{max}$	
dec_min	f8	$Dec \geq Dec_{min}$	
dec_max	f8	$Dec < Dec_{max}$	
x_tangent	f8	Center of the projection region	
y_tangent	f8	Center of the projection region	
orientat	f4	Orientation of the projection	CROTA2
nx	i4	X-size of projection region in pixels	
ny	i4	Y-size of projection region in pixels	
skycell_start	i4	Starting line of skycells for a tile	
skycell_end	i4	Ending line of skycells for a tile	
skycells			
name	u16	File name (see previous section)	
ra_center	f8	Cell center R.A.	
dec_center	f8	Cell center declination	
orientat	f4	Local orientation of the cell	
x_tangent	f8	Cell x coord from the tile center	CRPIX1
y_tangent	f8	Cell y coord from the tile center	CRPIX2
ra_corn1	f8	Cell first corner R.A.	
dec_corn1	f8	Cell first corner dec.	
ra_corn2	f8	Cell second corner R.A.	
dec_corn2	f8	Cell second corner dec.	
ra_corn3	f8	Cell third corner R.A.	
dec_corn3	f8	Cell third corner dec.	
ra_corn4	f8	Cell fourth corner R.A.	
dec_corn4	f8	Cell fourth corner dec.	

Table 1: Metadata tables

4 Software

4.1 Github repository

The software developed to obtain the Roman tessellation described in this report as well as the routines used to obtain the figures can be found at the github repository of the Space Telescope: `https://github.com/spacetelescope/skymap`.

To install the package, it is recommended to create an environment and clone it locally. This can be achieved with the commands in following block:

```
git clone git@github.com:spacetelescope/skymap.git
cd skymap
conda env create -f environment.yml
conda activate skymap
conda install asdf -c astropy
pip install -e .
```

Notebooks containing the commands to generate the figures of this report, to read the "asdf" file containing the metadata, and to visualize tiles and cells are conserved in the directory notebook of the repository.

4.2 Generating the metadata

To obtain the file containing the metadata of all the skycell files described in Table 1 execute the following commands:

```
from skymap.skymap import romantessellation, tiles2asdf

NSIDE = 13
theta, phi, ramin, ramax, \
decmin, decmax, vertices = romantessellation(NSIDE)
tiles2asdf(theta, phi, ramin, ramax,
           decmin, decmax, outfile='roman_wfi_skycells.asdf')
```

The output file *roman_wfi_skycells.asdf* is written in ASDF format. It has two extensions, one containing the general keywords for each skytile and the second one containing the specific keywords for each one of the skycells.

5 Bibliography

Calabretta & Roukema, 2007, MNRAS, 381, 872

Gorski et al., 2005, ApJ, 622, 759

Euclid Collaboration, 2022, A&A, 662A.112E