MPA/ESO/MPE Joint Astronomy Conference

Mining The Sky 2000

Splitting the sky - HTM and HEALPix.

W. O'Mullane

Astrophysics Division, Space Science Department of ESA, ESTEC, 2200 AG Noordwijk, The Netherlands.

A. J. Banday

MPI für Astrophysik, P.O. Box 1523, D-85740 Garching, Germany.

K. Gorski

European Southern Observatory (ESO), Karl Schwarzschild Strasse 2, 85748 Garching, Germany

P. Kunszt, A. Szalay

The Johns Hopkins University, 3701 San Martin Drive, Baltimore, MD 21218, USA.

Abstract: Mission data sets are increasing in size and complexity. To enable efficient processing and storage of such data sets a tessellation scheme may be adopted. The Hierarchical Triangular Mesh (HTM) and the Hierarchical Equal Area isoLatitude Pixelisation (HEALPix) schemes are two popular schemes each with software tools available to exploit their particular properties. The initial drivers behind these two schemes are quite different but as their use continues the complimentary facilities of each package will become more relevant. Here we describe the motivation and facilities of HEALPix and HTM, we ask if it is possible to combine these schemes or make use of them in a combined manner and say why we may wish to do this.

1. Motivation

The data avalanche was mentioned a few times during this conference. Only an organised approach will enable us to deal with the large quantities of data being produced by new missions such as the Sloan Digitized Sky Survey, GAIA and Planck. Part of this organised approach is the adoption of a tessellation scheme to enable partitioning of the data. Several schemes now exist leaving many people in a quandary as to which they should use. An important factor is the original design drivers for the scheme. For HTM and Healpix these are quite different.

The motivation for HTM [Kunszt 2000] is to index the sphere and provide complex trigonometric queries in spherical space. This allows one to bin up data sources in to HTM triangles and perform complex spatial queries on them. As the name indicates it is Hierarchical meaning triangles fit inside each other allowing operations to be performed at different resolutions. The main application in mind was partitioning of a point source catalogue.

For HEALPix the story is quite different. The aim of HEALPix is to support fast numerical analysis on data over the sphere, not point sources. The notion is to support convolutions with local and global kernels, Fourier analysis with spherical harmonics, power spectrum estimation and topological analysis. Again it was designed to be hierarchical but also to have equal area pixels with an iso-lattitude distribution to facilitate numerical analysis.

Hence, though both schemes provide a tessellation, the facilities they provide are quite different and disjoint.

More detail on HEALPix will be given below or can be found on the HEALPix website http://www.tac.dk:80/~healpix/. A more detailed description of HTM can be found in [Kunszt 2000] or on the Hopkins website http://www.sdss.jhu.edu/

2. Practical uses of HTM and HEALpix

Both Planck and GAIA are targeted for the earth-sun Lagrange point L2. Both satellites intend to adopt a continuous scanning strategy about L2 to give full sky coverage. Both missions aim to provide complete surveys to the limit of the instruments' sensitivities hence providing large amounts of data and full sky coverage.

2.1 GAIA

GAIA is proposed for ESA's fifth cornerstone mission which has a prospective launch date of 2009. The objectives of GAIA are many-fold but the core objective is the discovery of the origin and formation of the Galaxy. To do this GAIA will combine information from astrometry, photometry, and radial velocity instruments using the proven principles of the Hipparcos mission.

The astrometry will be complete to V=20 magnitude with accuracies of 4 microarcsec at V=10, 10 microarcsec at V=15 and 0.2 milliarcsec V=20. The radial velocity measurements will have accuracies of 1 km/s at V~10 and 10 km/s at V~17. There will be 4 broadband photometric filters and 11 medium spectral filters in the GAIA photometric system.

GAIA is estimated to observe around 1.3 billion objects hundreds of times over its 5 year lifetime producing around 10Tb of raw data. The current estimate of processing needed for GAIA is 10^{19} flops.

The HTM is already in use on GAIA in the internals of the simulator where the simulation for a given triangle can be repeated at will to produce the same results. This allows simulation of small patches of sky with the need to do a full sky simulation. The GAIA simulator is the subject of the poster presentation [Luri 2000].

More ambitiously it may be possible to partition mission data coming from GAIA using a scheme such as HTM to allow the type of temporal/spatial access needed to it for data processing. Either HTM or HEALPix schemes allow spatial splitting of the sky. A second split may then be performed in the temporal domain.

A consideration for a mission like GAIA in adoption of a scheme like HTM is also interaction with other catalogues. GSCII [McLean 2000] uses HTM as do Sloan [Zsalay 2000] for example so there may be advantages on basing at least the GAIA catalogue on a similar scheme. Some later global statistical products may be easier using something like HEALpix with its equal area property.

For further information on GAIA see http://astro.estec.esa.nl/GAIA.

2.2 Planck

Planck is an accepted medium sized mission and is due for launch in 2007. Planck will provide a major source of information relevant to several cosmological and astrophysical issues, such as testing theories of the early universe and the origin of cosmic structure.

The angular resolution of Planck will be 10 arcmin. Two instruments will be on board to give frequency coverage of 30-850 Ghz. The temperature sensitivity of Planck will be DT/T ~ $2x10^{-6}$ in the channels where the CMB is the dominant signal, and as close to this value as technically possible in all other channels.

Planck will produce nine complete maps of the sky in different frequency ranges. The raw data produced by the instrument will amount to around 0.5 - 1 Tb.

HEALPix is in use already on Planck for simulations and has been adopted by the Science Team as the standard for Planck data maps. See for example [Giardino 2000].

For further information on Planck see http://astro.estec.esa.nl/Planck.

3. HEALPix

In this section a brief overview of the construction principles of the **H**ierarchical **E**qual **A**rea iso**L**atitude **Pix**elisation (HEALPix) is given.

HEALPix may be seen as a division of the sphere into 12 base pixels (see Figure 1 (left)) which are then subdivided to give the desired resolution. The amount of subdivision is given be the NSIDE number where NSIDE is always a power of 2 hence 1,2,4,8,16,32 etc.

The base pixels (or faces) are number 0 - 11 where 0-3 are the North faces, 4-7 the equatorial and, 8-11 the South faces.

The power to which 2 is raised may be equated to the notion of depth in the HTM e.g. the HTM depth 2 means each of the 8 base pixel are subdivided in to 4 sub pixels in HEALPix NSIDE 2 has the same meaning - each of the 12 base pixels are subdivided into 4 parts. This HEALPix hierarchy is described below.

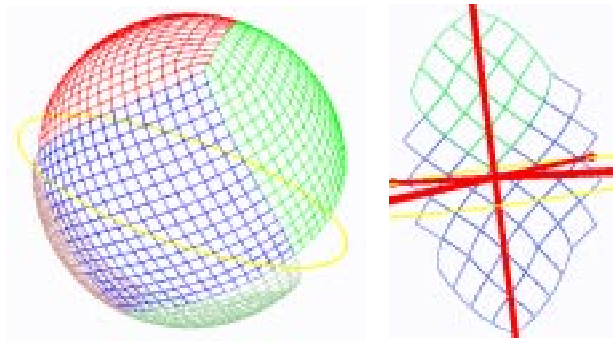

Figure 1. 5 of the 12 Healpix base pixels NSIDE 16 (left) Zone 0 at NSIDE 4 (right).

3.1 zones

The HEALPix construction is based on an octahedron and the current implementation is based about 4 orange segment like zones Pi/2 wide, these zones cut across the equatorial base pixels. All calculations are performed in 0 to Pi/2 space and then simply rotated depending on the zone (see Figure 1(right)). The zones are numbered [0,3] hence a rotation involves adding zone*(Pi/2) to any Phi value. Within the zone, theta (north, south) is divided into (4*NSIDE -1) rings. Each ring in turn is divided into equal units allowing integer addressing of each pixel using ring number and offset. The number of pixels on a ring in a given zone is NSIDE in the equatorial region (hence 4*NSIDE for the sphere), while the pixel number decreases by one pixel per ring per zone as one moves north and south of the equatorial region. The polar regions comprise NSIDE rings each while the equatorial region holds the remaining (2*NSIDE -1) rings. On the north and south tip of any zone there is only one pixel giving 4 pixels on the polar tips of the sphere. It should be noted in the equatorial region that pixels are offset by half the width of a pixel (Pi/2 /NSIDE) in each alternate ring (as may be seen in Figure 1 (right)).

The general formulae for the pixel boundaries are $\cos\Theta = a+b/\Phi^2$ in the North cap $\cos\Theta = a \pm b \cdot \Phi$, ιn Equatorial and $\cos\Theta = a+b/(\Pi/2-\Phi)^2$ in the South cap. This gives non-geodesic pixels of equal area. The shape of the pixels varies slightly in the polar regions.

3.2 Hierarchy - nested numbering scheme.

HEALPix has a hierarchical design. For any given NSIDE the base pixels are subdivided into exactly NSIDE^2 equal area pixels. As the NSIDE is increased each pixel becomes sub divided while maintaining the original pixel boundaries e.g. the 4 new pixels fit exactly inside the original one pixel.

A hierarchical binary numbering scheme may also be adopted where each two bits represent a pixel number at a given depth i.e. by shifting off the last two bits one can find the parent pixel number of a pixel. The step from NSIDE 2 to NSIDE 4 is shown in Figure 4 here we can see the numbering scheme clearly in the left pixel - the decimal numbers are also given for clarity. For each subsequent subdivision the resulting 4 pixels are numbered in the same way but the resulting bits are added to the parent pixel number as shown in the right hand pixel (NSIDE 4).

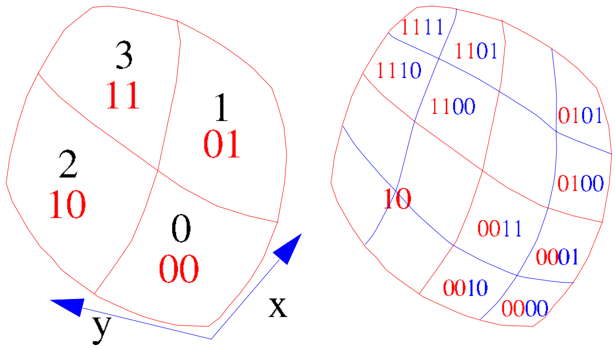

Figure 2. A healpix base pixel at NSIDE2 and NSIDE 4 showing hierarchical numbering.

Within the nested scheme there are also routines to find neighbours easily using X,Y offsets. The bottom pixel is considered 0,0. The axis directions are shown on the left side of Figure 2.

The pixels numbers from each face are prepended with the integer face number* (NSIDE^2) giving unique numbers for all pixels in the map.

3.3 Rings

The iso latitude nature of HEALpix means that pixels align on rings (as described in the zone above) which is particularly useful for numerical operations such as FFTs. Effectively HEALPix lays a grid on the sphere so any pixel may be referred to by its by its ring number and offset or i_theta and i_phi as it is sometimes referred to. Figure 3 shows 4 of the 31 rings for NSIDE 8 with the pixels of face 0 also shown numbering starts from i_phi =0 in each ring - these are the numbers shown extending down the sphere - subsequent pixels are numbered in increments of one as i_phi increases.

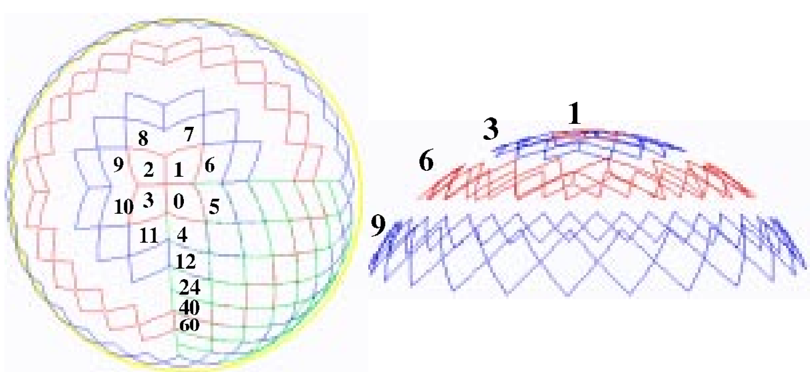

Figure 3. Some pixel numbers (left top down view) and ring numbers for NSIDE 8.

The ring numbering scheme is not hierarchical in nature, fast routines exist however to switch between a ring scheme number and a nested scheme number. Working with i_theta and i_phi at a given resolution means neighbour searches in the ring scheme are also achieveable.

4. Special considerations

Any sphere tessellation will have some anomalies which must be taken into account by developers using the scheme in their algorithms. In the case of HTM the pixel shape and orientation vary and pixels are not iso-lattitude making global numerical analysis difficult. The main author has spent more time developing extensions to HEALPix and hence knows more about the intricacies of this scheme which are discussed below. It should be noted that these are only issues for those wishing to develop on HEALPix, but any large project will inevitably end up extending any library (HTM or HEALPix).

4.1 Pixel location on sphere

As with HTM the pixel shape varies over the sphere with healpix but this is generally not a problem is a large enough resolution is used. The shape varies in the polar regions, a zone (Figure 1) is effectively split in three parts: North, Equatorial and South. Slightly different formulae are needed for each part hence when doing pixel operations one must check where the pixel is.

4.2 Seven Neighbours

Most HEALPix pixels have eight neighbours but the north and south pixel of each equatorial face have only seven neighbours (eight pixels). This has an obvious impact on any kind of box extraction and neighbour searching (Figure 4).

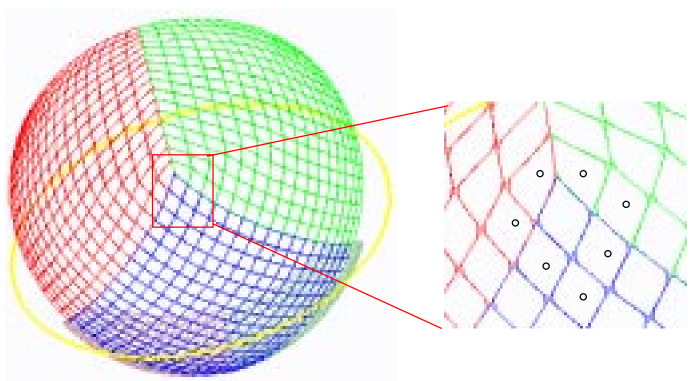

Figure 4. Eight pixels have seven neighbours - north pixel of face 4 example.

4.3 Orientation of base pixels

As stated before in the nested scheme it is useful to use the x,y offset within a base pixel to find neighbours etc. One must beware however when crossing the boundary between base pixels since the neighbours orientation may be different.

Figure 5 shows that when traversing to the North West neighbour of face 10 to face 5 the x,y orientation stays the same also going from 5 to 1 all is okay. But, the NW neighbour of face 1 is 0 and it is rotated 90 degrees so the x,y orientation is different. Face 0 is also the northern neighbour of face 3 where the orientation differs by 180 degrees.

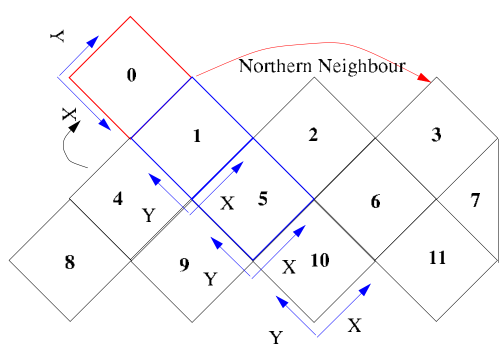

Figure 5. Base pixel orientation. Faces shown flat.

5. Resolution of the tessellation.

There is no theoretical limit to the depth of either tessellation scheme. Computers however are not theoretical so a given implementation is limited in the numbers it can use. A 32 bit integer is long enough to address pixels to NSIDE 8192 but no more - this is the current HEALPix implementation limit. Similarly in HTM the java version uses 32 bit integers and can only go to HTM depth 14 - the C++ version uses 64 bit integers and can be used to depth 25 where the double precision maths make the vertices indistinguishable form one another.

As stated above the HTM depth and the power of 2 which gives the NSIDE in HEALpix are similar though yielding different pixel areas. Figure 6 lists HTM depth and nominal pixel area(arcsec^2) follow by NSIDE for HEALPix and the actual pixel area. The last two columns give the actual number of pixels/leaves in the complete set.

D	PixArea	NSIDE	PixArea	NPIX	HTM_leaves
10	1.77E1	1,024	1.18E1	12,582,912	8,388,608
11	4.43E0	2,048	2.95E0	50,331,648	33,554,432
12	1.11E0	4,096	7.38E-1	201,326,592	134,217,728
13	2.77E-1	8,192	1.84E-1	805,306,368	536,870,912
14	6.92E-2	16,384	4.61E-2	3,221,225,472	2,147,483,648
15	1.73E-2	32,768	1.15E-2	12,884,901,888	8,589,934,592
16	4.32E-3	65,536	2.88E-3	51,539,607,552	34,359,738,368
17	1.08E-3	131,072	7.20E-4	206,158,430,208	137,438,953,472
18	2.70E-4	262,144	1.80E-4	824,633,720,832	549,755,813,888
19	6.75E-5	524,288	4.50E-5	3,298,534,883,328	2,199,023,255,552
20	1.69E-5	1,048,576	1.13E-5	13,194,139,533,312	8,796,093,022,208
21	4.22E-6	2,097,152	2.81E-6	52,776,558,133,248	35,184,372,088,832
22	1.06E-6	4,194,304	7.03E-7	211,106,232,532,992	140,737,488,355,328
23	2.64E-7	8,388,608	1.76E-7	844,424,930,131,968	562,949,953,421,312
24	6.60E-8	16,777,216	4.40E-8	3,377,699,720,527,872	2,251,799,813,685,248
25	1.65E-8	33,554,432	1.10E-8	13,510,798,882,111,488	9,007,199,254,740,992

32 bit integer limit

64 bit double limit

Figure 6. Table of HTM and HEALpix areas and pixel counts showing implementation limitations.

For Planck the typical NSIDE required will be 1024. Assuming a double for each pixel this is a 300MB map - this does not include error values. Should Planck require the next level of resolution the maps will be over a Giga Byte in size.

6. Wish list

6.1 Standardisation

It would be good to have an agreed interface for index/pixelisation schemes. This would facilitate interchange as well as interoperation of different schemes. There will never be one scheme used by everyone, some specific needs will always require certain missions to adopt their own tessellation scheme. Furthermore improvements may be made to a given scheme - will this nullify work done with the older version. The interoperation of archives is now being postulated via schemes such as the National Virtual Observatory (NVO) - so should we not try to standardise the interface to the scheme rather than the scheme itself?

Practically what would this mean? The most basic realisation would be to have a similar set of routines which operate with the same parameters. For example most operations in HTM are done using unit vectors (x,y,z) while in HEALPix everything is done in terms of spherical coordinates (theata,phi). Conversion between the two is easy but really both libraries should support both types of interaction. In this example HTM is easy to modify since all that has to be changed is the SpatialVector to allow a constructor, set and get methods for spherical coordinates. HEALPix is generally in FORTRAN so addition of unit vectors would require some more effort in that scheme. A second example is that of

NSIDE versus depth as discussed above. Here depth seems more intuitive than NSIDE. Again this may require major re-coding in HEALPix since it is written in a procedural language.

A standard way for describing features on the sphere would allow complex geometric queries to be run on different systems. Here again the underlying scheme should not be the driver rather the way of interacting with it. It seems the convex method used in HTM is flexible and may suit well - this will of course suffer from “not invented here” syndrome for other missions/users. These type of geometric extraction features will become important for missions like Planck at some point in time. Already masking out the galactic plane is done in some Planck processing - but it is done in a specific rather than generic way i.e. to mask a different region means changing code rather than expressing a different set of convexes as in HTM.

On the numerical analysis side it would be great to have a common set of analyses tools which could be used on any scheme - here of course HEALPix is ahead since this was its primary objective. Eventually projects like Sloan who are concerned now mainly with point sources will need fast numerical methods for their indexing scheme too.

6.2 Benchmarks

If some standardisation is bought to spatial indexing then it would also be possible to define bench marks for different schemes. Benchmarks are of course notoriously untrustworthy but in this case they would facilitate computer scientists in achieving specific goals set by astronomers.

Benchmarks also help to demonstrate speed difference which can be seen by inspection. For example one may expect HTM to computes a triangle to a given depth on the sphere using a recursive method while HEALPix does it in a fixed number of calculations to which NSIDE is a parameter. We would expect HEALPix to be much faster and indeed a test with 20,000 pointings (in java for both systems) shows HTM to be considerably (order of magnitude) slower for a depth 12/NSIDE 1024. One can also see that HEALPix times do not change with NSIDE whereas HTM time increase with depth. On the face of it this would seem to be damming for HTM however for an application like object matching the algorithm is I/O limited - the HTM is of adequate speed.

Hence benchmarks are useful but direct comparison of simple features can be misleading.

6.3 3D/4D support

Both HEALpix and HTM are spherical and deal in a curved space. However in both cases data is effectively projected out to the surface of a unit sphere. For catalogues the likes of which GAIA will produce the distances to stars will be very accurate it would be useful to have the proper third dimension in the scheme. GAIA will also measure proper motions very accurately so adding not just a third dimension but at least a fourth dimension to the data - this is a very complex search space when time is taken into consideration. This is also a challenge faced by the NVO where they have multi epoch data and many observations of individual sources over time. A kind of onion skin layering may take care of the 3 dimensions but a more clever scheme will be needed to address the time issue.

6.4 Balanced/multi resolution tree support.

To support a proper tree structure for database purposes it must be possible to have different depths in the tree i.e. all leaves need not necessarily be at the same level. The reason for this is to have what is called a balanced tree, where we are guaranteed that a similar amount of data exist in each leaf node thus being able to guarantee a worst case retrieval time. Neither HTM nor HEALPix support this.

In the case of Sloan this does not matter as the data seems pretty uniform. However for a mission like GAIA there will be far more data in the Galactic plane hence we may wish to have a deeper mesh here to partition the data evenly.

For integration with a database the index must be split from the containers, which will be database dependant. Some design work needs to be done to achieve this.

The equivalent way of looking at this in HEALpix terms is having a multi resolution map. The advantages of this for maps is a sort of h-tree compression when a group of pixels have the same value they can be collapsed into their parent pixel. This would not be as efficient as a h-trees since the grid is laid down in advance. This would take care of sparse maps which occur in Boomerang.

Sparse maps have been dealt with in other ways and for the Cosmic Microwave background (CMB) the probability of any neighbouring pixels having the same value is very low so this is not a big issue for Planck.

Any changes like these would mean quite some effort in the implementation. It is easy to deal with trees of uniform depth.

6.5 Bit list support in HEALpix

A nice feature of HTM is the use of bitlists to mask the sphere e.g. one bit is used to represent the inclusion of a triangle in the set. This has the advantage of allowing quick merging of leaf sets using logical operators e.g. anding two complex queries.

It would be advantages to have this sort of bit list masking in HEALPix but again processing software would need to take account of it so it would require some effort.

6.6 Interoperation

It is desirable, and in principle possible, to have geometric queries on HEALPix similar to HTM. The HTM queries are based on convexes and a routine exists in HEALPix to extract all pixels in a disc (again there is a syntax difference in the two schemes). Furthermore it is possible to extract all pixels in a given triangle so a HTM query could be done returning triangles which could then be converted to HEALPix numbers. This will require some further work.

7. Conclusion

With large data volume cleaver index/pixelisation schemes will be necessary. The effort and problems encountered in developing a system should not be underestimated - it would be advantageous to have schemes available for use. Such schemes exist however it is difficult to compare them and to know if they meet the need of a particular project. Some form of standardisation would make the initial selection of a scheme easier and would facilitate possible migration or integration of other schemes later.

In the era of the NVO interoperation will be the key. Index developers should work together with this goal in mind.

8. References

Giardino G. et al. 2000, Analysis of CMB foregrounds using a database for Planck, These Proceedings.

Kunszt P. et al. 2000, Mapping the Sky using the Hierarchical Triangular Mesh, These Proceedings.

Luri X. et al. 2000, Simulation of the GAIA mission data using Java and UML, These Proceedings.

McLean B. et al. 200, An Overview of the Multi-Terabyte GSC-II Database and some Preliminary Scientific Results, These Proceedings.

Szalay, A.S. 2000, Large Databases in Astronomy, These proceedings.