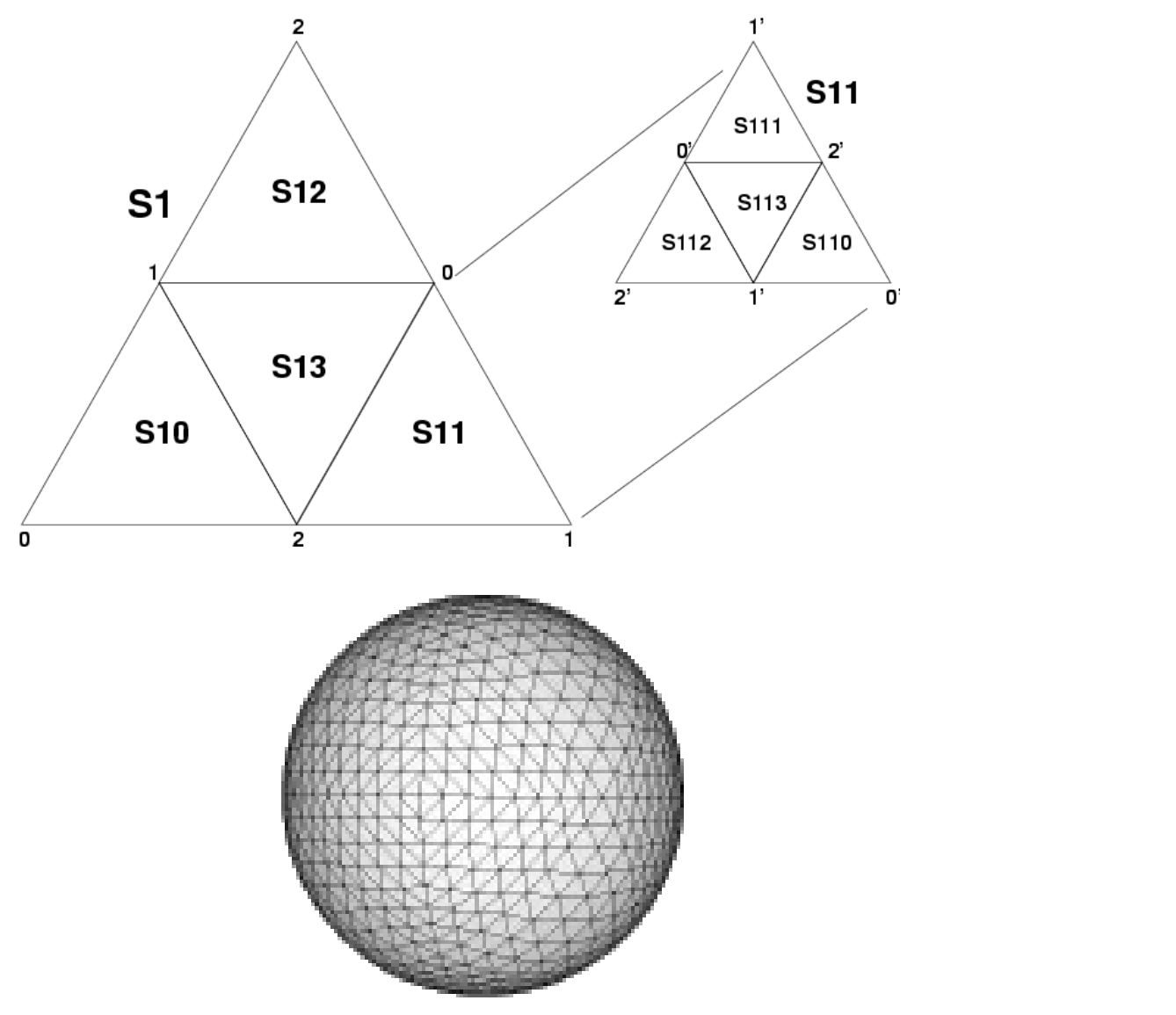
Indexing astronomical database tables using HTM and HEALPix

MCS, a flexible resource for astronomical projects

Luciano Nicastro, IASF-INAF, Bologna, Italy & Giorgio Calderone, Palermo Uni., Italy



HTM: Hierarchical Triangular Mesh sky pixelisation. Used for objects catalogues. Good to perform cross-matching between tables. "**www.sdss.jhu.edu/htm/**".

The total number of pixels in a map is governed by the *depth* parameter of the pixelisation:

`d` (depth) : [0 , 25]

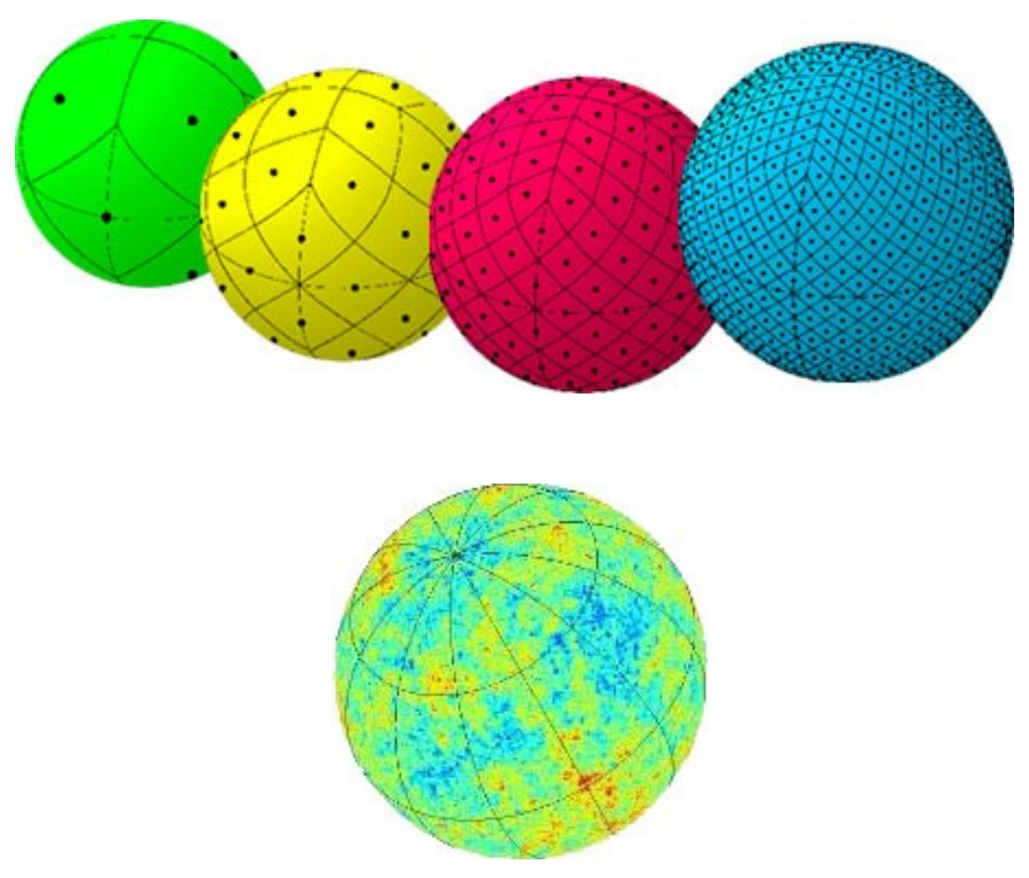
$N_{pix} = 8 \times 4^{d}$ $ID_{range} : [N_{pix} , 2 \times N_{pix} - 1]$

Background

Spatial indexes in relational databases has always been an important issue for multi dimensional data sets, in particular for those dealing with spherical coordinates, e.g. latitude/longitude for Earth locations, or RA/Dec for celestial objects. Some database servers offer built-in capabilities to create indexes on these (coordinate) columns which consequently speed up the execution of queries involving them. However 1. the use of these facilities could be not easy, 2. they typically use a syntax quite different from the astronomical one, 3. their performance is inadequate for the astronomical use.

In Astronomy it is crucial to have coordinates' indexed tables. All sky optical and IR catalogues have up to 1 billion objects that will increase with forthcoming projects. Also partial sky surveys at various wavelengths can collect information (not just source lists) which can be saved in coordinate ordered tables. Selecting a sub-set of these entries or cross-matching them could be unfeasible if no indexing is performed. Sky tessellation with various mapping functions have been proposed. It is a matter of fact that the astronomical community is accepting the HTM and HEALPix schema as the default for object catalogues and for maps visualization and analysis, respectively.

Within the **`MCS`** library project, we have now made available as MySQL-callable functions various HTM and HEALPix facilities. This is made possible thanks to the capability offered by MySQL 5.1 to add external plug-ins. The **`DIF`** (Dynamic Indexing Facilities) package distributed within the **`MCS`** library, creates and manages a combination of Views, Triggers, DB-engine and plug-ins allowing the user to deal with database tables indexed using one or both these pixelisation schemas in a completely transparent way.



HEALPix: Hierarchical Equal Area isoLatitude Pixelization. Mainly used for analysis of large-scale spatial structures. Also suitable for source catalogues. "**healpix.jpl.nasa.gov**".

The total number of pixels in a map is governed by the *order* parameter of the pixelisation:

`k` (order) : [0 , 29]

$N_{side} = 2^{k}$ $N_{pix} = 12 \times N_{side}^{2}$

$ID_{range} : [0 , N_{pix} - 1]$ $\Omega_{pix} = \pi / (3 \times N_{side}^{2})$

Indexing on a sphere: HTM and HEALPix

In terms of database table indexing, mapping a sphere with a pixel schema means transforming a 2–d into a 1–d space, consequently a standard B-tree index can be created on the column with the pixel IDs. On a large astronomical table, depending on the "depth" of the pixelisation, this could lead to a gain of a 4–5 orders of magnitude in search efficiency. The HTM and HEALPix schemas are widely used in Astronomy and are now well mature to be considered as candidates for the **`MCS`** library. They are both open source and distributed as C++ libraries. HTM uses triangular pixels which can recursively be subdivided in four pixels. The base pixels are 8, 4 for each hemisphere. These "trixels" are not equal-area but the indexing algorithm is very efficient for selecting point sources in catalogues. HEALPix uses equal-area pseudo-square pixels, particularly suitable for the analysis of large-scale spatial structures. The base pixels are 12. For both HTM and HEALPix, using a 64 bit long integer to store the index IDs leads to a limit for the pixels size of about 0.4 milliarcsec on a side.

Being able to quickly retrieve the list of objects in a given sky region is crucial in several projects. For example hunting for transient sources like GRBs requires fast catalogues lookup so to quickly cross match known sources with the detected objects. The IR/optical robotic telescope REM uses HTM indexed catalogues to get the list of objects in 10′×10′ regions. Accessing catalogues like the 2MASS takes some 10 msec. Having a fully automatic HTM and HEALPix indexing would be crucial for the management of the DBs of future large missions like Planck and Gaia. Also the Virtual Observatory project would greatly benefit from adopting a common indexing schema for all the various types of archive it can manage.

The MCS DIF package

`MCS` is a set of C++ high level, easy to use, classes aimed at implementing an application server, that is an application that provides a service over the network. It provides classes to interact with, manage and extend a **MySQL** DB server. The **`MyRO`** package allows a per row management of DB tables whereas the **`DIF`** package allows the automatic management of sky pixelisation with the HTM and HEALPix schema.

To enable **`DIF`**, when installing MCS it is enough to give to the **`configure`** script the two options:

```
--enable-dif --with-mysql-source=PATH
```

where **`PATH`** is the path to the MySQL source directory. The HTM and HEALPix C++ libraries are included in the **`DIF`** package.

Now let's assume one has a DB **`MyDBname`** with a table **`MyCatname`** containing the two coordinates column **`RAcs`** and **`DECcs`** representing the ***centi-arcsec*** converted J2000 equatorial coordinates (this requires 4 bytes instead of the 8 necessary for a double value). To make the table manageable using both HTM and HEALPix pixelisation schemas it is enough to give the command:

```
dif --index-both MyDBname MyCatname 6 0 8 "RAcs/3.6E5" "DECcs/3.6E5"
```

where 6 is the HTM depth and 8 the HEALPix order; 0 (1) will select the RING (NESTED) scheme. If the coordinates where already degrees, then it would have been enough to give their names, e.g. **`dif`** … **`RA DEC`**. The MySQL root password is needed.

Adding a function to perform simple catalogues cross matching is relatively easy. We'll release it soon.

Having an HTM indexed catalogue, the query string to obtain the list of objects in a circular region centred on α=60° and δ=30° with radius 40′ will be:

```
SELECT * FROM MyCatname_htm WHERE DIF_HTMCircle(60,30,40)
```

note the **`_htm`**! For a rectangle with the same centre and sides 50′ along the α axis and 20′ along the δ axis:

```
SELECT * FROM MyCatname_htm WHERE DIF_HTMRect(60,30,50,20)
```

giving only three parameters would imply a square selection.

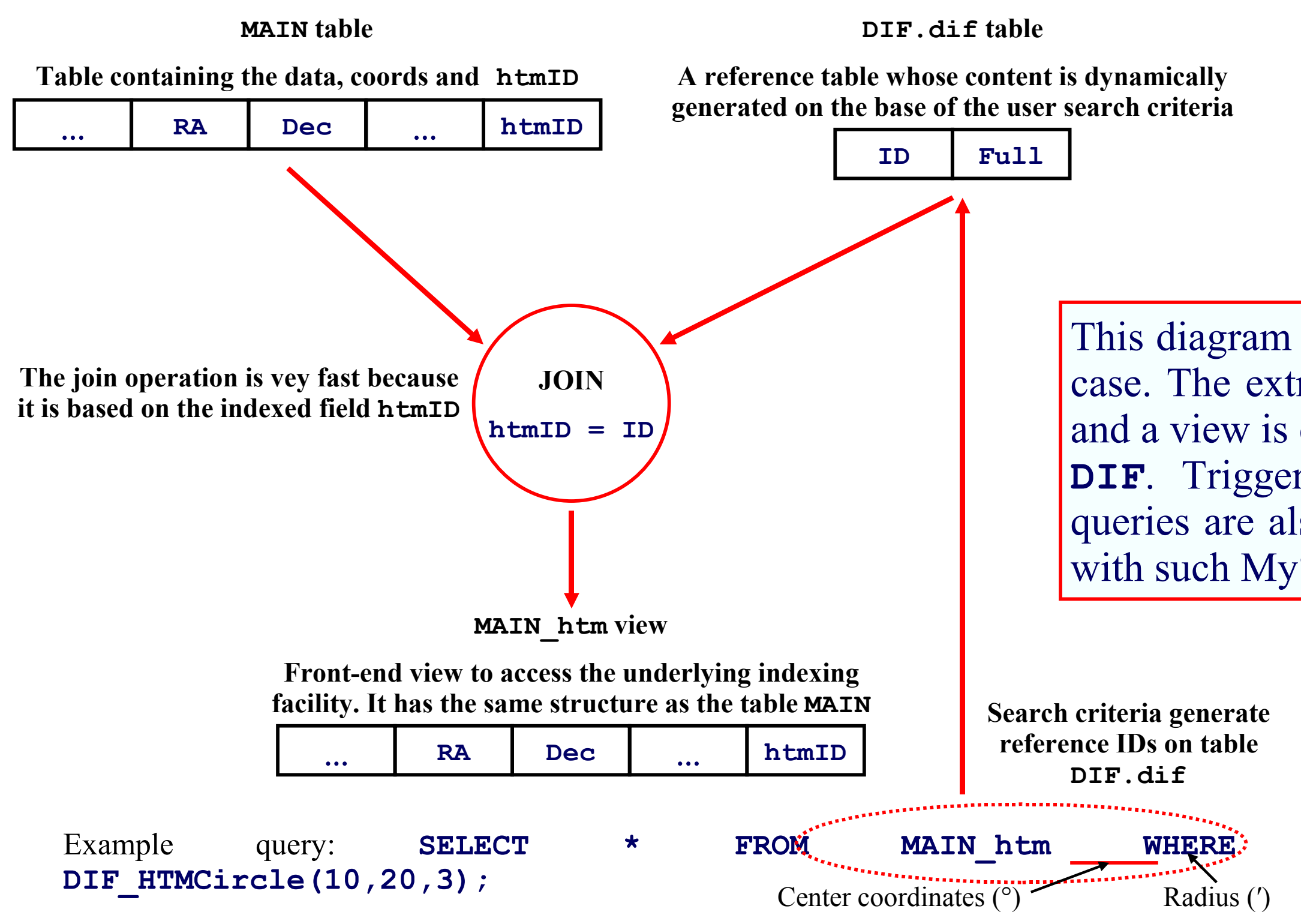
Having chosen to use both HTM and HEALPix indexing, one could request all the HEALPix IDs of the objects in a 50′ square by using an HTM function:

```
SELECT healpID FROM MyCatname_htm WHERE DIF_HTMRect(60,30,50)
```

If one has just the HEALPix IDs then can select entries on a circular region:

```
SELECT * FROM MyCatname_healp WHERE DIF_HEALPCircle(0,60,30,40)
```

note the **`_healp`**! Rectangular selections for only-HEALPix indexed tables will be available soon.



`MAIN` table

Table containing the data, coords and `htmID`

...	RA	Dec	...	htmID

`DIF.dif` table

A reference table whose content is dynamically generated on the base of the user search criteria

ID	Full

The join operation is vey fast because it is based on the indexed field `htmID`

JOIN

`htmID = ID`

`MAIN_htm` view

Front-end view to access the underlying indexing facility. It has the same structure as the table `MAIN`

...	RA	Dec	...	htmID

Search criteria generate reference IDs on table `DIF.dif`

Example query: `SELECT * FROM MAIN_htm WHERE DIF_HTMCircle(10,20,3);`

Center coordinates (°)

Radius (′)

This diagram summarises the way **`DIF`** works for the HTM indexing case. The extra column `htmID` is added to the original table **`MAIN`** and a view is created which performs a join with the dynamical table **`DIF`**. Triggers to automatically manage **`INSERT`** and **`UPDATE`** queries are also created. We recall that **`MCS`** allows users to interact with such MySQL server using programs written in any language.

`http://ross.iasfbo.inaf.it/MCS/`

(nicastro@iasfbo.inaf.it | gcalderone@ifc.inaf.it)

`MCS` is open source and it is downloadable from the Web or requested to us. Documentation is included in the distribution and it is available as doxygen HTML pages. A descriptive document (TeX+PDF) is also included. The distributed tarball provides the typical and simple `configure/make/make` install sequence. Only open source external libraries are required. In particular MySQL (the **`DIF`** package needs v.5.1). A test suite is included together with demo programs in the various supported languages. If you wish IDL/PHP/Python/Perl/Java compatibility then these packages need to be pre-installed too. The *row oriented* DB tables management is accomplished by **`MyRO`**, a Perl+SQL package we developed and distribute with **`MCS`**.

Please note that the library is in continuous improvement so we ask the interested people to get in contact with us. We are seeking collaborations and we plan to have a distribution mailing list.

Using DIF

*When using the **`DIF`** facilities the user must understand that in the **`WHERE`** clause the functions **`DIF_HTMxyz`** must be used on the view **`MyCatname_htm`** and the **`DIF_HEALPxyz`** on the view **`MyCatname_healp`** in order to make use of the corresponding indexing. The functions **`DIF_HTMLookup`** and **`DIF_HEALPLookup`** instead can be used anywhere, being simple coords → ID converters. For example:*

```
SELECT DIF_HEALPLookup(0,12,10,30);
```

*would return **50323911**, the order 12 RING pixel ID at α=10° and δ=30° . Similarly assuming one has a table which is indexed only with the HTM schema, to have e.g. the order 10 HEALPix IDs of the entries in a circular region centred at α=22°, δ=33° and radius 20′ a query like this can be used (assuming coordinates columns are in degrees):*

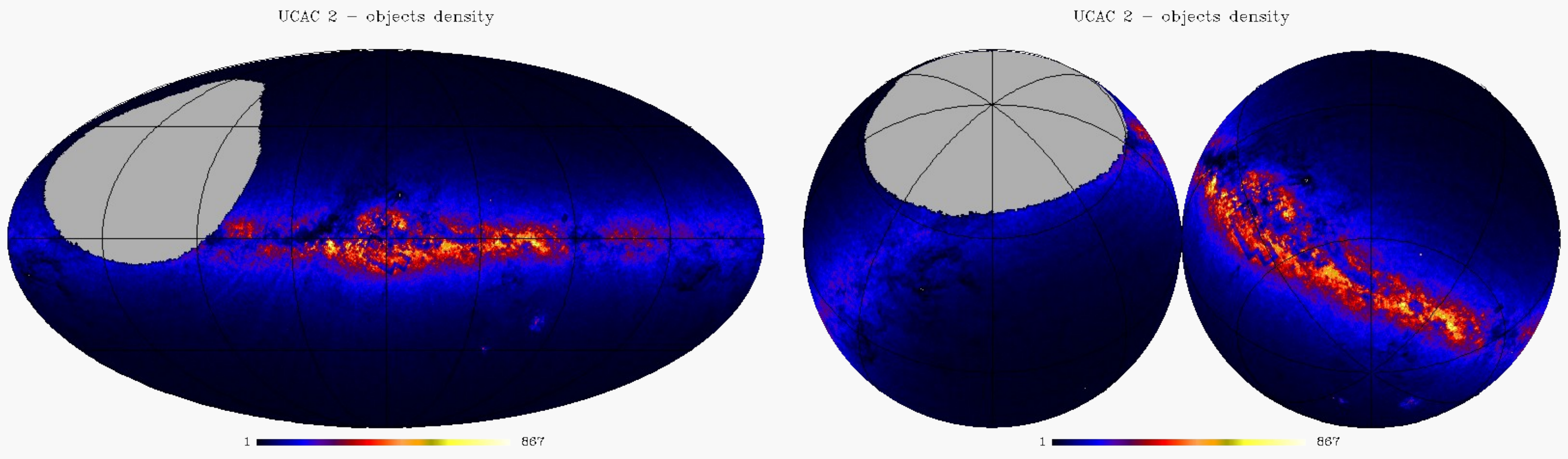
```
SELECT  DIF_HEALPLookup(0,10,RAdeg,DECdeg) FROM MyCatname_htm
        WHERE DIF_HTMCircle(22,33,20);
```

*The HTM depth must be in the range **[0 , 25]**, whereas the HEALPix order in the range **[0 , 29].***

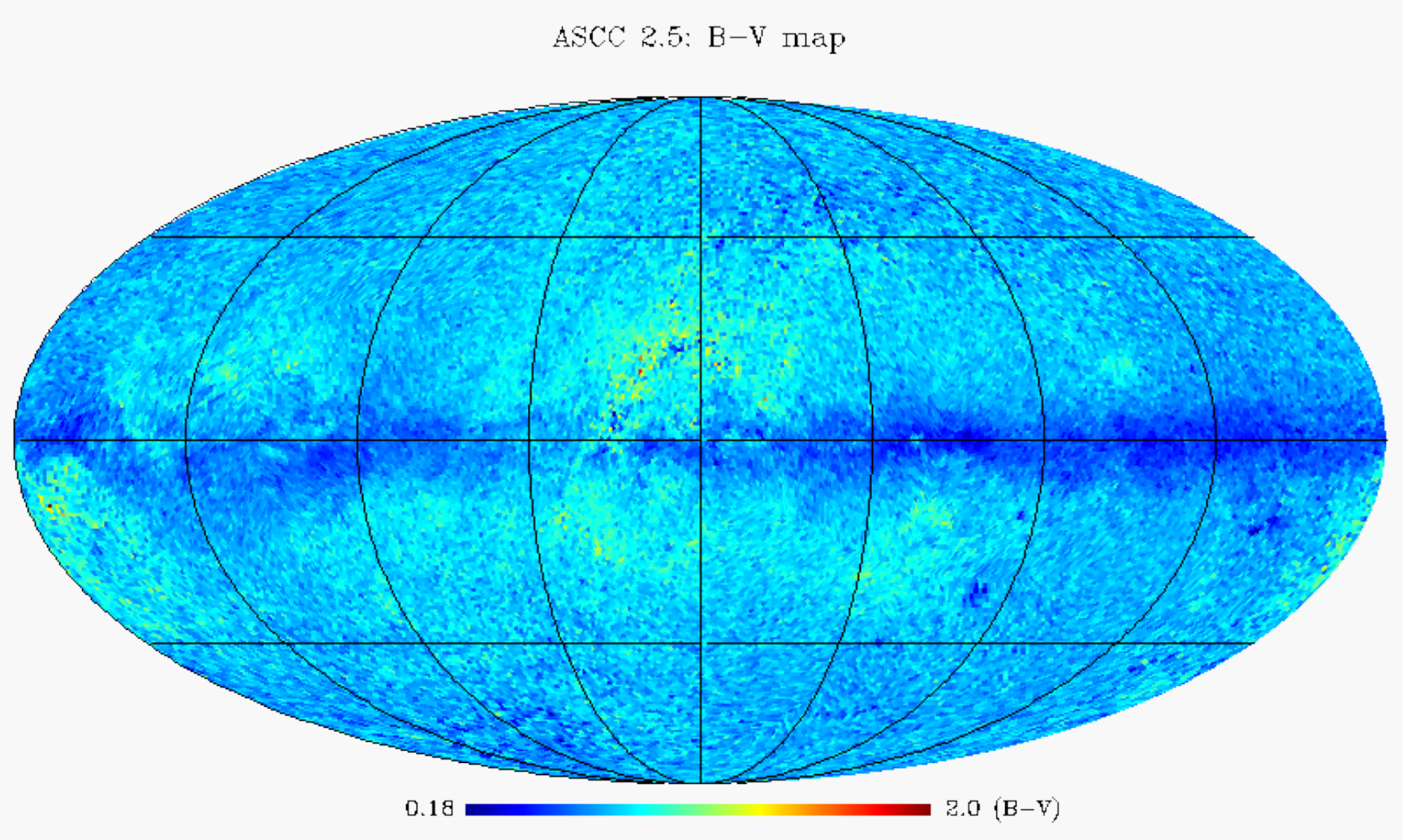
***`MCS`** provides a number of interfaces to different programming languages: C/C++, Fortran, IDL, PHP, Python and soon Perl and Java. We have started writing high-level contributed library which make the use of **`MCS`** even easier. In particular the IDL library includes some specific functions aimed at building and saving HEALPix maps which can then be plotted using existing ad-hoc tools (`mollview` etc.). The core of a typical program would look like this (note that all relevant parameters are passed through dedicated common blocks):*

```
; DB info initialization
host='localhost'    & c_user='dbusr_name' & c_password='dbusr_pass'
c_dbname='MyDBname' & tabname='MyCatname'
htmtab=tabname+'_htm' & hptab=tabname+'_healp'
; The query to execute; a dedicated function could build it
c_query='SELECT healpID,Rmag FROM '+htmtab+' WHERE DIF_HTMRect(60,30,500,300)'
; DB connection and query execution
err = dbexec_qry(Host=host)
; If OK then retrieve the selected healpID and Rmag columns into 'map' arrays
if (not err) then err = healp_mapfill(/norm) ; Ask to normalise Rmag map
dbfree  ; Close connection and free DB resources
; Save the two maps (N_entries/pix and Average_Rmag/pix) into FITS files
save_map(Outf='map_nocc.fits') & save_map(/sum,Outf='map_aveR.fits')
; Plot in gnomic projection the two maps
gnomview, hp_map_nocc,/online,coord=['c','c'],rot=[60,30],units='Objs/pix'
gnomview, hp_map_sum,/online,coord=['c','c'],rot=[60,30],units='Average Rmag'
```

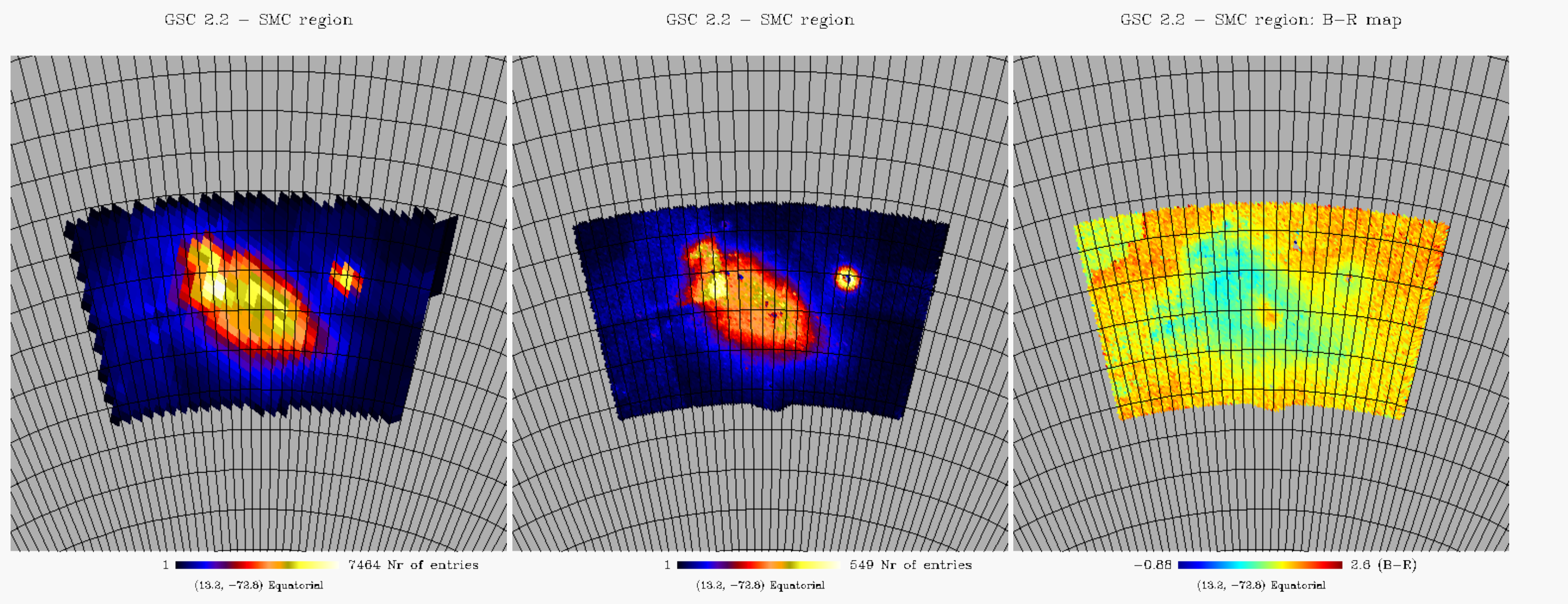
Some examples for full-sky and sky patch are shown in the figures below.



UCAC 2 (48.3 M_{objs}): Galactic and equatorial view of the objects density. The uncovered region around the North pole is visible. "**ad.usno.navy.mil/ucac/**"



ASCC 2.5 (2.5 M_{objs}): Galactic view of the average objects colour. The uncovered region around the North pole is visible. *Kharchenko et al. 2001*. "**vizier.u-strasburg.fr**"



GSC 2.2 (455 M_{objs}): Gnomic view of the object in a region 500′×300′ centred on SMC.

Left: objects density with pixel scale ~14′. *Centre:* objects density with pixel scale ~3.4′. It is evident the not uniform coverage of the catalogue. *Right:* Average objects B-R colour per pixel. "**www.gsss.stsci.edu**"

The catalogues used for these demos are browseable on ross.iasfbo.inaf.it/