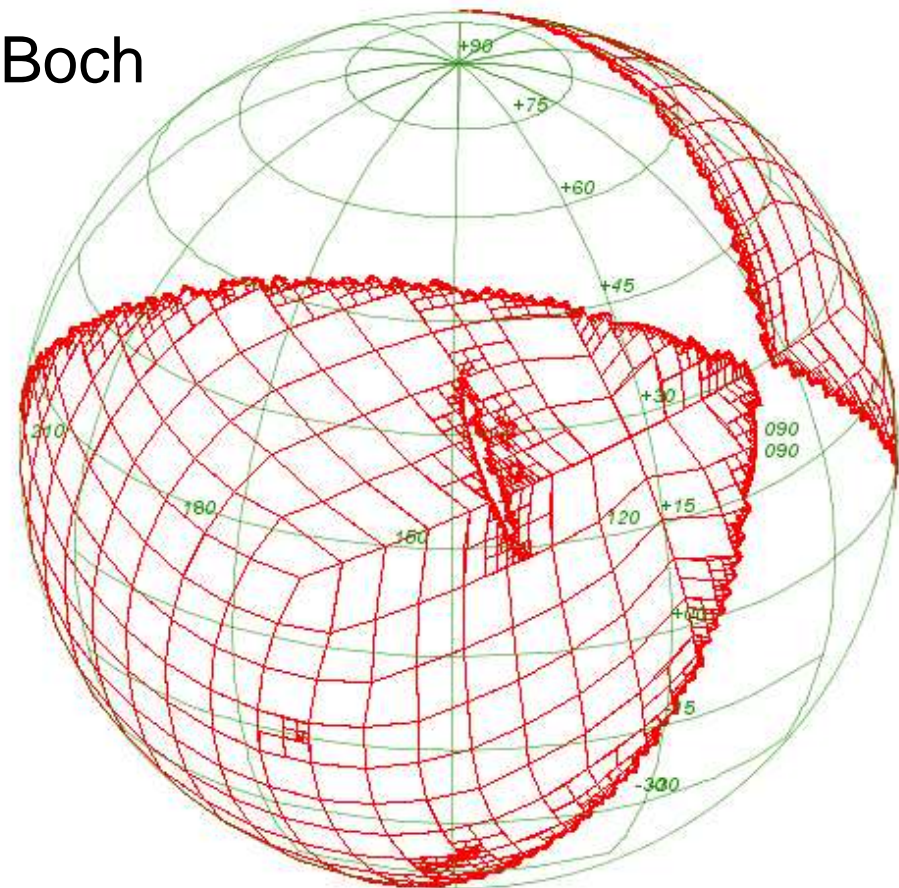


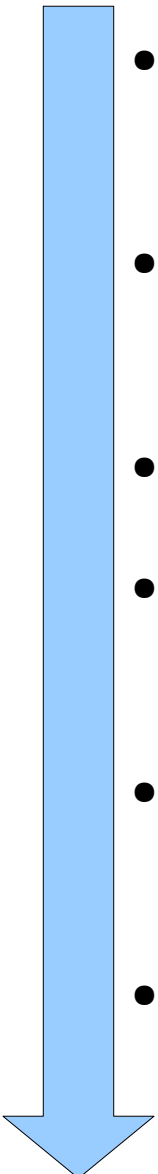
MOC

(HEALPix Multi-Order Coverage map)

Pierre Fernique – Thomas Boch

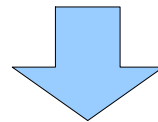


History

- 
- Jan 2011 • The idea + prototype implementation in Aladin
 - May 2011 • Napoli Interop => presentation during Application session
 - VizieR catalog MOCs generation
 - Nov 2011 • ADASS 2012 => poster + proceedings
 - Dec 2011 • TOPcat implementation (multicone-search)
 - Apr 2012 • IVOA note (Boch, Donaldson, Fernique, O'Mullane Reinecke, Taylor)

Brief reminder...

- **A MOC** : just a simple and powerful method to specify sky regions.
- **Goals** : provide very fast comparisons and data access methods.
- **Principle** : based on HEALPix sky tessellation :

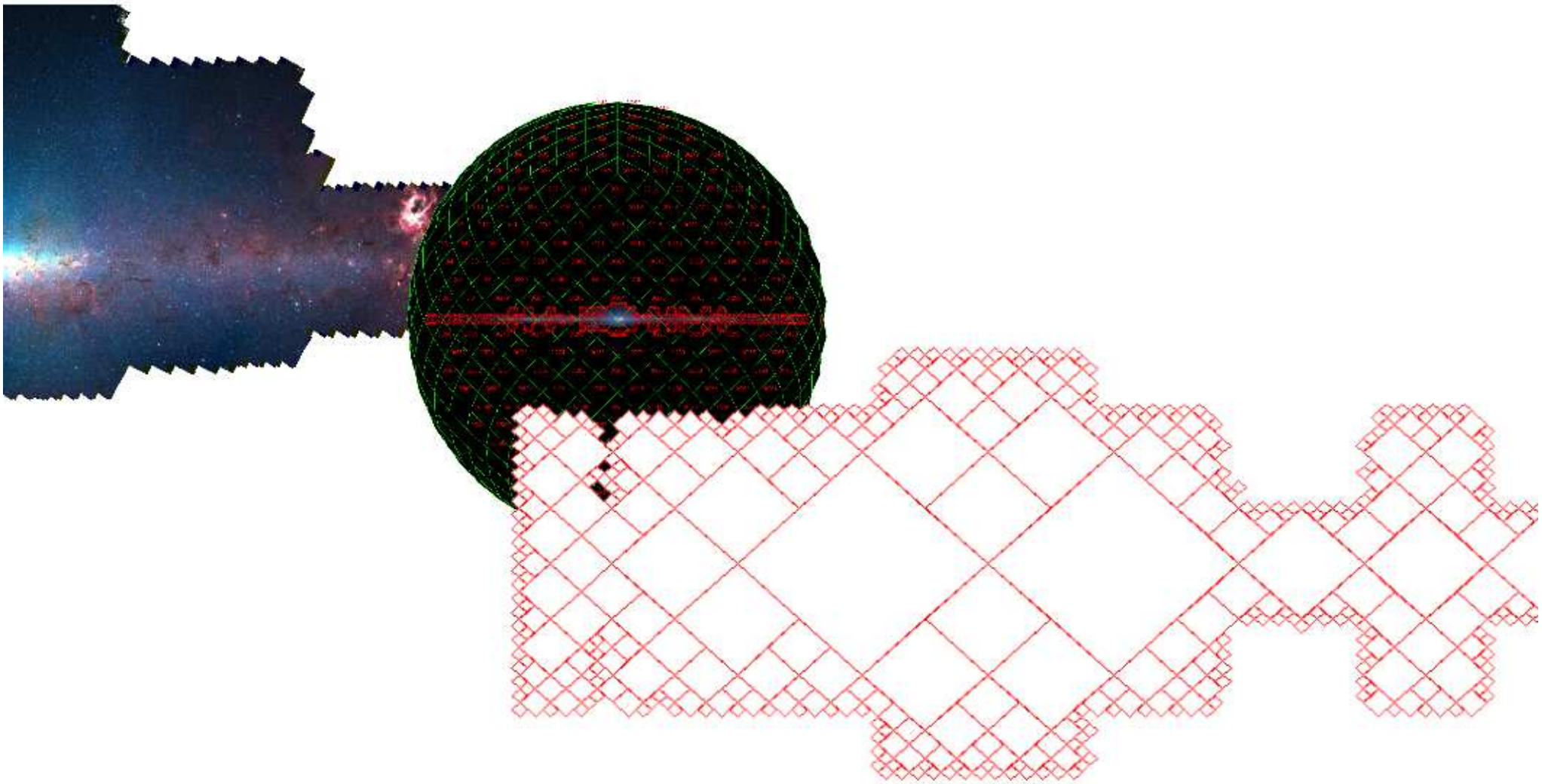


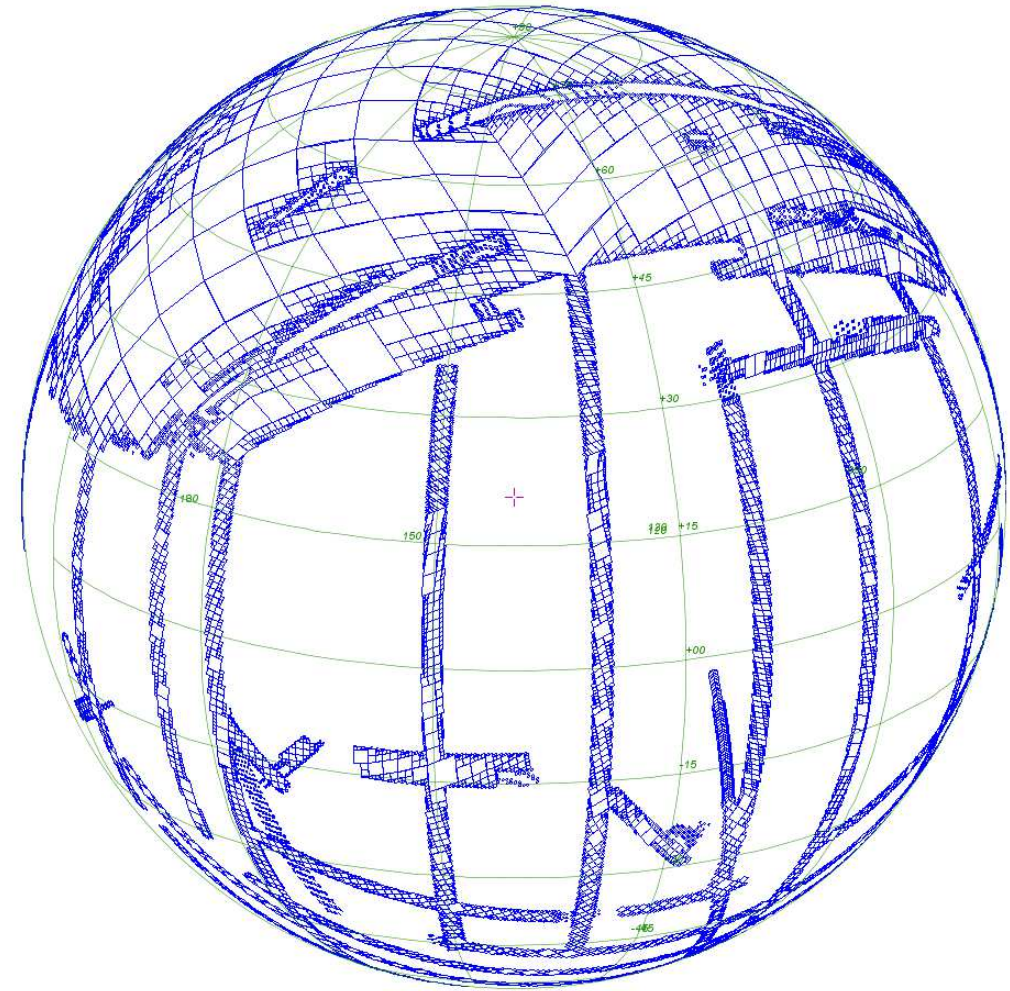
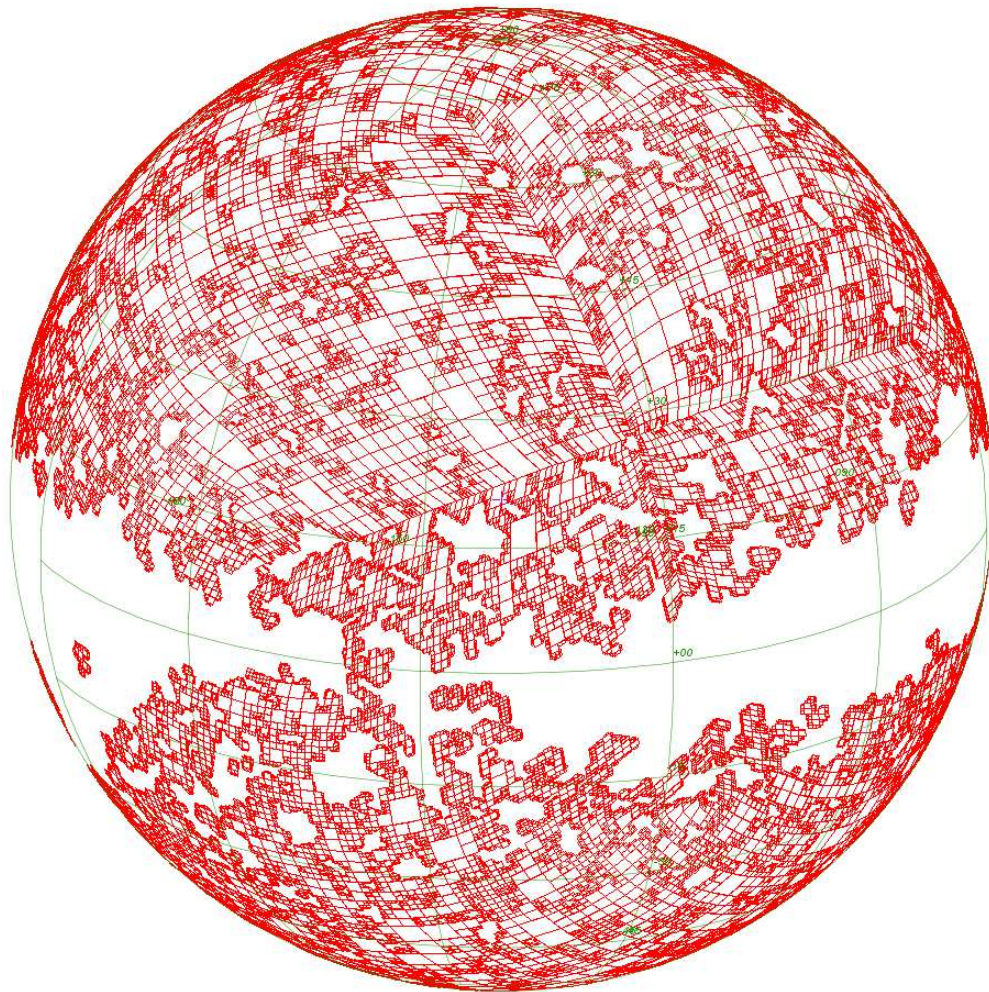
*a list of HEALPix cells,
grouped hierarchically*

=

a sky region

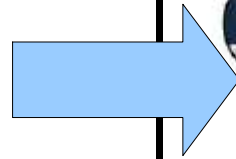
From data to the MOC...





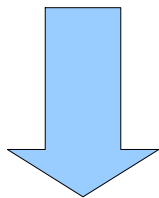
*GALEX AIS MOC (order 8 – 70,000 cells)
SDSS (oder 10 – 225,000 cells)*

All details described here



- Why HEALPix ?
- Examples of use cases
- Scope and performances
- MOC encoding
- Libraries
- ...

Java library here



<http://cds.ustrasbg.fr/resources/lib/exe/fetch.php?media=mocsrc.jar>



International
Virtual
Observatory
Alliance

MOC – HEALPix Multi-Order Coverage map

Version 1.0

IVOA Note 12th April 2012

Previous version(s):
None

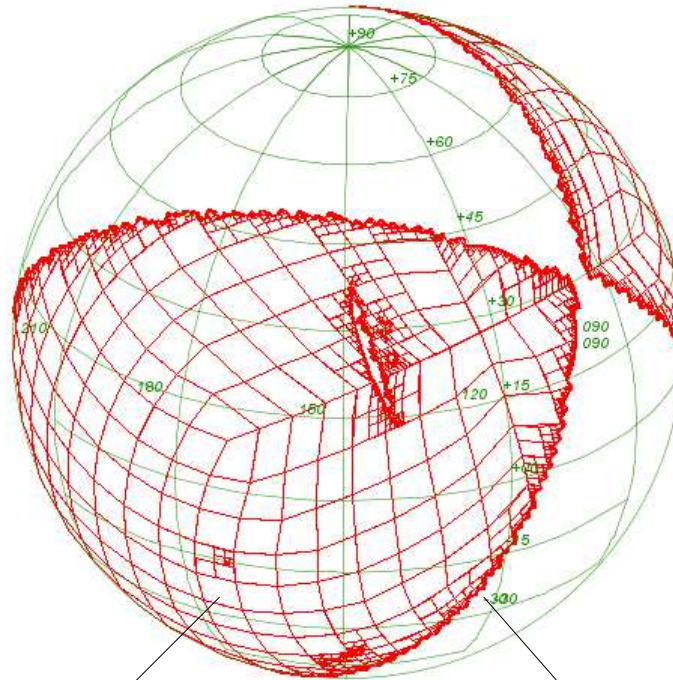
Authors:
Thomas Boch
Tom Donaldson
Pierre Fernique
Wil O'Mullane
Martin Reinecke
Mark Taylor

Editor:
Pierre Fernique

Abstract

This note describes a simple and powerful method to specify sky regions. The goal is to have a way for providing very fast comparisons and data access methods. The principle is based on HEALPix sky tessellation. It boils down to defining a list of sky cells, grouped hierarchically. This method has been tested and validated in two tools: Aladin and Topcat, and in a front-end tool for Vizier catalogs.

Two encoding formats



versatility

performance

JSON MOC => easy to write, to read and to parse

FITS MOC => designed to be processed as quick as possible (no parsing, loading or writing in one fast step)

```
{ "1": [1, 2, 4], "2": [12, 13, 14, 21, 23, 25], ... }
```

FITS binary table

Few figures...

2.5s	for loading 7,000 Vizier MOCs (level 8)
7ms	for HEALPix query (amongst 7,000 Vizier tables, which tables are concerned by this cell?)
90ms	for MOC query (amongst 7,000 Vizier tables, which tables intersect this MOC (4000 cells)?)
30ms	for generating MOC intersection (SDSS level 10 & GALEX level 8)

=> See Thomas' demo