

# VIRTIS-VEX Data Manual

R. Politi, G. Piccioni, F. Henry, S.Erard, S.Jacquiod, P. Drossart

April 1, 2014



# Contents

<b>Contents</b>	<b>iii</b>
<b>1 VIRTIS data general presentation</b>	<b>1</b>
1.1 The data repository . . . . .	1
1.2 The data format . . . . .	2
1.3 The files . . . . .	3
1.4 Header parameters . . . . .	5
1.5 Housekeeping values . . . . .	5
<b>2 Archive / data set status</b>	<b>7</b>
2.1 Archive status: 2014 . . . . .	9
2.2 Archive documentation . . . . .	11
<b>3 Pipeline description</b>	<b>13</b>
3.1 A few data about VIRTIS on Venus Express . . . . .	15
3.2 From the spacecraft to co-Is . . . . .	16
3.3 Telemetry to raw data . . . . .	17
3.4 Database ingestion . . . . .	18
3.5 Geometry computation . . . . .	19
3.6 Calibration . . . . .	20
3.7 Automated computations . . . . .	21
3.8 Data archive delivery to PSA . . . . .	22
<b>4 Virtis-M calibration</b>	<b>23</b>
<b>5 Virtis-H calibration</b>	<b>25</b>
5.1 Basic scheme . . . . .	27
5.2 Computing transfer function . . . . .	28
5.3 Applying transfer function . . . . .	30
5.4 Results . . . . .	31
<b>6 Computation of the Geometry</b>	<b>33</b>
6.1 Access to geometry files . . . . .	35
6.2 Main points . . . . .	36
6.3 Tools used to compute geometry . . . . .	39
6.4 GEOVIRTIS inputs . . . . .	43
6.5 GEOVIRTIS outputs . . . . .	44

6.6	SCET and Ephemeris Time . . . . .	45
6.7	Instrument Field of View . . . . .	46
6.8	Instrument footprint . . . . .	47
6.9	Geometric cubes size . . . . .	48
6.10	Order of the computed parameters . . . . .	49
6.11	Units & Multiplying factors . . . . .	51
6.12	Detailed description of each parameters . . . . .	52
6.13	Limb observations and restituted distances . . . . .	66
6.14	Missing values . . . . .	70
6.15	Handling time . . . . .	71

# Chapter 1

## VIRTIS data general presentation

### 1.1 The data repository

The VIRTIS data could be downloaded from the ESA site

*<ftp://psa.esac.esa.int/pub/mirror/VENUS-EXPRESS/VIRTIS/>*.

You will find a directory for each mission phase:

Phase	Start	MTPs	Orbits
Nominal mission	4 June 2006	1 - 19	23 - 521
Extension 1	3 October 2007	20 - 40	563 - 1132
Extension 2	31 May 2009	41 - 56	1139 - 1575
Extension 3	22 August 2010	57 - 82	1587 - 2289

The folders structure is:

```
/
├── VEX-V-VIRTIS-2-3-EXT1-V2.0
├── VEX-V-VIRTIS-2-3-EXT2-V2.0
├── VEX-V-VIRTIS-2-3-EXT3-V2.0
├── VEX-V-VIRTIS-2-3-V3.0
│   ├── AAREADME.TXT
│   ├── BROWSE
│   ├── CALIB
│   ├── CATALOG
│   ├── DATA
│   ├── DOCUMENT
│   ├── ERRATA.TXT
│   ├── EXTRAS
│   ├── GEOMETRY
│   ├── INDEX
│   └── VOLDESC.CAT
```

<b>AARAEDME.TXT</b>	Readme file containing the description of the archive.
<b>BROWSE</b>	Reduced resolution versions of data products.
<b>CALIB</b>	Calibration files used in the original processing of the data, or needed to use the data.
<b>CATALOG</b>	PDS catalog files.
<b>DATA</b>	Contains the subdirectories of data products. The DATA subdirectory is used to unclutter the root directory of a volume by providing a single entry point to multiple data subdirectories.
<b>DOCUMENT</b>	Documentation, supplementary and ancillary information to assist in understanding and using the data products.
<b>ERRATA.TXT</b>	Comments on the delivery.
<b>EXTRA</b>	Value added elements included by the data preparer, but outside the scope of the PDS archive requirements.
<b>GEOMETRY</b>	Files describing the observational geometry (SPICE kernels).
<b>INDEX</b>	Indices to assist in locating data of interest.
<b>VOLDESC.CAT</b>	Readme file containing the description of the dataset.

## 1.2 The data format

The VIRTIS data are stored in a structure named data cube (or Qube following the Planetary Data System Standard notation). The qube is a 3-dimension matrix containing the the digital numbers for the raw data and the radiance for the calibrated data, stored in physical units:

$$R = \frac{W}{m^2 \mu m sr} \quad (1.1)$$

The raw data are stored using 2 bytes unsigned integer while the calibrated data is saved in floating precision, using 4 bytes per element.

The axes of the qube are called **band** (**b**), spectral band defined by the wavelength also called *spectral dimension*, **sample** (**s**), spatial direction along the slit also called *spatial dimension*, and **line** (**l**), acquisitions in successive time steps also called *temporal dimension*.

The three types of 2-dimension structures in a QUBE are called:

- frame** (fixed *line*), provided by the detector at a given time;
- image** (fixed *band*), a spatial information reconstituted through time at a given wavelength;
- slice** (fixed *sample*), spectra acquired through time at the same position along the slit.

The three types of 1-dimension structures in a QUBE are called:

- row** (fixed *band* and *line*), spatial information acquired in a single time step at a given wavelength;
- spectrum** (fixed *line* and *sample*), relative to a given footprint on the target;
- scan** (fixed *band* and *sample*), spatial information reconstructed through time.

The atomic structure of the QUBE is called either **pixel** or **spectel**, depending whether it is considered a part of an image or a spectrum. The storage order of the QUBE is

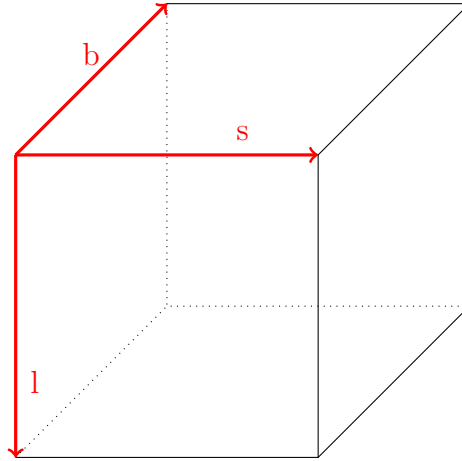


Figure 1.1: Data QUBE structure.

always (Band, Sample, Line) (i.e., band interleaved by pixel, or BIP) to respect the order of the telemetry data flow and to minimize data handling operations.

## 1.3 The files

The main VIRTIS-VEX file have the extensions .QUB, .CAL and .GEO.

- .QUB contain the row data ;
- .CAL contain the calibrated data;
- .GEO contain the geometric information of each pixel.

The directory structure of the directory tree is the following:

```
DATA
├── MTPXXX.....MTP (Medium Term Planning) contain 28 orbit, (4 week of obser-
    │               vations).
    └── VIRXXXX.....orbit directory. Each orbit is 24h of observations
        ├── RAW ..... containing the .QUB files
        ├── CALIBRATED.....containing the .CAL files
        └── GEOMETRY.....containing the .GEO files
```

The official software suite for VIRTIS-VEX is developed in ITT IDL and is called LecturePDS. Please check on the server if you are using the last version (2.8.1). The last version of the software is on the server in the directory SOFTWARE.

The routine for the lecture of the data is virtispds, using the syntax:

```
res = virtispds(<filename>,[/Silent])
```

where res is the return structure, filename is the name of the file you want read (with the complete path) and Silent is an optional keyword that disables the output on screen.

In the return structure, if you read a calibrated cube, there are the following fields:

<b>LABEL:</b>	containing the file header, is a string array, one element for each line.;
<b>TABLE:</b>	is float array (band x sample x 3) containing the wavelength, bandwidth (FWHM) and radiance uncertainty (1-sigma) ;
<b>QUBE_NAME:</b>	is a strings array with two elements providing the cube name and unit ;
<b>QUBE_DIM:</b>	an array with 3 elements providing the cube dimensions ;
<b>QUBE:</b>	3D data core of the qube (float). Size=(band x sample x line) ;
<b>SUF_NAME:</b>	list of suffix parameters (SCET) ;
<b>SUF_DIM:</b>	an array with 2 elements providing the suffix dimensions ;
<b>SUFFIX:</b>	SCET for each frame. Size=(3, # of frames) .

The detail are reported in the file VIRTISPDS.pdf available in the software package. You can visualize basically two kind of data: a spectrum or an image. For the spectrum you can use the usual plot procedure:

```
plot, res.table[:,0,0], res.qube[:,100,100], min_value=0
```

in order to plot the spectrum of the pixel 100,100 of the cube with the right wavelength. The keyword min\_value =0 is used in order to exclude the band with some problems, like dead pixels, saturated pixels etc., that are set to a value less equal than -1000 (see calibration document for details). The image visualization is more complex and is more easy to understand using an example. We want to visualize the image at 1.97 micron (band 100). The pipeline for the visualization is

```
1 res=virtispds(Dialog_PickFile(),/Silent) ;read the file
img=res.qube[100,*,*] ;put the image in a variable
3 img=Reform(img) ;transform the image in a 2D matrix
img[Where(img LT 0)!=!values.f_nan ;change the negative pixels in NaN
5 sz=Size(img,/Dimension) ;Get the size of the image
Window,0,xsize=sz[0],ysize=sz[1] ;open a direct draw device
7 Tvscl, img, /NaN ;visualize the image
```

The geometric parameters of the cube are in the .GEO file, as wrote above. In order to read this file you must use the virtispds procedure too, with the previous syntax:

```
geo = virtispds(<filename>, [/Silent])
```

In this case the structure geo has the following fields:

<b>LABEL:</b>	containing the file header, is a string array, one element for each line.;
<b>QUBE_NAME:</b>	Containing the names of the parameters;
<b>QUBE_COEFF:</b>	containing coefficient conversion coefficients;
<b>QUBE_DIM:</b>	an 3 elements array providing the cube dimensions ;
<b>QUBE:</b>	3D data core of the qube (float). Size=(# of coef, # of sample, # of line) .



For details see *VIRTIS\_Geometry.pdf*.

If you want know the latitude, longitude and local time of the spectrum in the pixel 100, 100 of the first example you can use the following commands:

```
1 ; Print the latitude (center of the pixel above the clouds)
  Print, geo.qube[25,100,100]*geo.qube_coeff[25]
3 ; Print the longitude (center of the pixel above the clouds)
  Print, geo.qube[24,100,100]*geo.qube_coeff[24]
5 ; Print the local time
  Print, geo.qube[15,100,100]*geo.qube_coeff[15]
```

## 1.4 Header parameters

Many of the operative parameters of the instrument are stored in the cube header<sup>1</sup> and can be retrieved automatically. For instance, here is how to get the exposure time:

```
fp = v_pdspar(res.label, 'frame_parameter_desc')
2 ip = Where(v_listpds(fp) EQ 'EXPOSURE_DURATION')
  ; Extract the value
4 Tint = (v_listpds(v_pdspar(res.label, 'frame_parameter')))[ip]
  Tint = Reform(Tint) ; turned to a scalar
6 ; Extract the unit
  sUnit = (v_listpds(v_pdspar(res.label, 'frame_parameter_unit')))[ip]
8 Print, STRTrim(Tint,2)+' '+sUnit
```

## 1.5 Housekeeping values

The housekeeping parameters are stored in the sideplane of the raw PDS qubes in a rather compact format to save disk space and memory. After a call to `virtispds`, they are accessible in the suffix tag of the result structure in a more suitable way. Successive HK structures are separated (instead of being packed as they are in the qube sideplane), but the values are still encoded according to the TM scheme. These quantities provide all the information required for data processing purposes, including calibration. For detailed HK monitoring, the library provides a way to convert HK parameters into physical values. Multiple HK are first split into independent quantities, then these quantities are converted using a mission-specific transfer function for Rosetta and Venus-Express. The conversion is performed as follows:

```
oParTab = v_pdshk(result)
```

In the return structure there are the following fields:

---

<sup>1</sup>The header of the cube represent the PDS label, infact the VIRTIS-VEX data are stored in *attached label* format.

**VALUES** : is an array of instrument parameters converted to physical units;  
**NAMES** : is a string array conatining the names of these instrument parameters.

## Chapter 2

Archive / data set status

# VIRTIS-VEx

---

## Archive / data set status

Stéphane Erard, P. Drossart, Florence Henry, Sophie Jacquinod

VEx workshop, 5/2014



# Archive status: 2014

- Heavily iterated between team & PSA
  - No known PSA/PDS compliancy issue ; includes complete doc
- Includes raw + calibrated data, all three channels: H, M-Vis, M-IR
  - All data files / observations
  - Geometrical information on a pixel basis
  - Comprehensive calibration information
  - Software and data handling instructions
  - Browse images (used in PSA interface)
  - Catalogues
- Current version delivered May/June 2013
  - Complete data set (=> August 2012)
  - Completed/updated on a regular basis
- Big!
  - 95 359 files — 908 Gbytes (at least)

# Archive status: 2014

- Pending
  - M-vis not yet fully calibrated
- Scheduled updates:
  - Improvement of M-IR calibration
  - First tentative calibration of M-vis channel
  - Refinement of signal-to-noise ratio, FWHM, etc... for H
- Data (cubes) selection
  - See PSA interface functions
  - Search in PDS index (on selected parameters)
  - In project: Virtual Observatory access from LESIA

# Archive documentation

- EAICD (archive documentation) **VVX-LES-IC-2269**  
Version 1.6
- Geometry documentation **VVX-LES-SW-2268**  
Version 1.2
- Software/data handling documentation **VVX-LES-SW-2264**  
Version 3.1 available  
<http://voparis-europlanet.obspm.fr/othertool.shtml>
- Calibration docs: **H (7/2008) & M (5/2008)**  
+ Instrument docs/manual



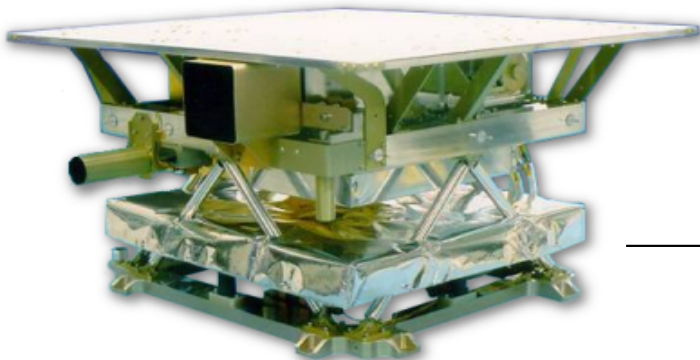


## Chapter 3

### Pipeline description

# VIRTIS data pipeline

Florence HENRY  
*[florence.henry@obspm.fr](mailto:florence.henry@obspm.fr)*

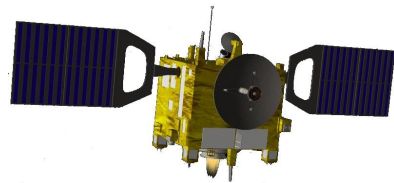


## A few data about VIRTIS on Venus Express

- On orbit around Venus since April 2006
- Up to 1 GB of raw data everyday
- Geometry and calibration computed within a few days
- 26000 raw, 21500 calibrated, 21500 geometry files
- Archive total size : 2.7 TB



# From the spacecraft to co-Is



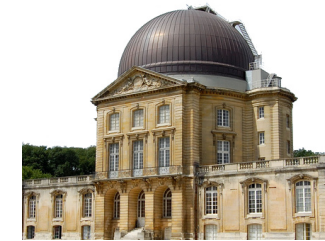
Venus Express  
Observations



ESOC



IAPS Rome  
Telemetry



LESIA - Meudon

EGSE

Raw data

Geometry

Updated raw data

H calibrated data

M calibrated data

Raw data

Geometry

Updated raw data

H calibrated data

M calibrated data

Archive delivery to PSA

ftp

co-Is distribution

web

## Telemetry to raw data

- TM is processed through the EGSE, operated in IAPS
  - Order the TM packets
  - Slice TM into sessions
  - Write PDS-formatted files, with a label, the data and a selection of HK
  - Timing information has poor accuracy



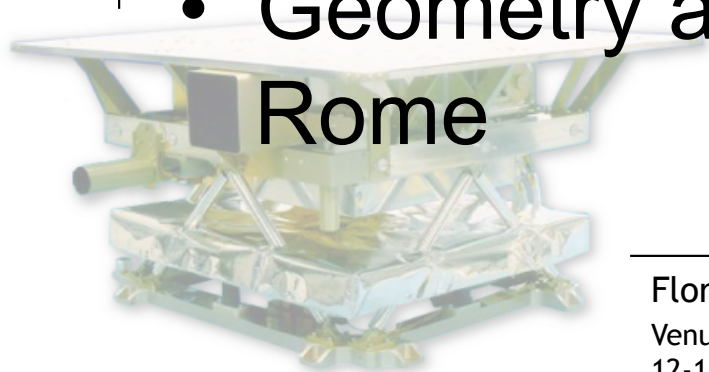
# Database ingestion

- Mirroring from Rome to Meudon
  - PDS raw data
  - Kernels for Virtis-M mirror
- Data label and HK are ingested into a SQL database
- Through the ESA CVS repository, ITLs and PTRs are also ingested
- Spice kernels are mirrored from the ESA ftp server



# Geometry computation

- Triggered upon reception of raw data and/or new spice kernels
- One geometric file is written for each raw PDS file (see Sophie's presentation)
- Raw data files are updated
  - geometry and catalogue keywords in the label
  - Fine UTC timing
- Geometry and updated raw are mirrored to Rome



# Calibration

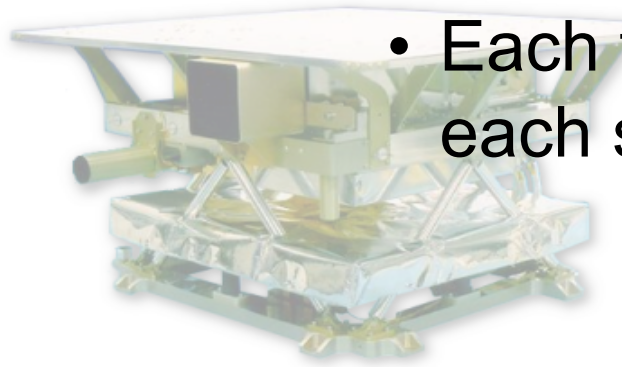
- From the updated raw files :
  - Virtis-M calibration is made in Rome
  - Virtis-H calibration is made in Meudon
  - See specific presentations
- Both calibrated channels are mirrored at both sites





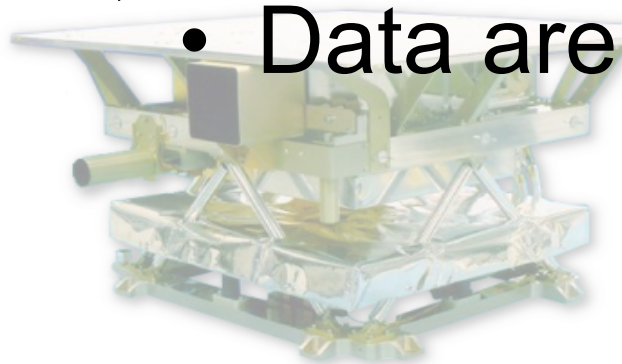
# Automated computations

- All the pipeline is automated
  - cron tasks
    - Mirroring (rsync/svn/cvs...)
    - Check new/updated files
    - Insertion/update into database
    - Geometry computation
    - Calibration
  - Workflow is handled by a database todo-list
    - Each file has a status TODO/DONE/ERROR for each step of the pipeline



## Data archive delivery to PSA

- All the datasets are reviewed internally before delivery to PSA
  - Fix minor problems
  - Check consistency of the labels
  - Check files formatting
  - Create catalogues and indexes
  - PVV tool is used
- Data are recomputed if necessary



## Chapter 4

### Virtis-M calibration



# Chapter 5

## Virtis-H calibration

The full document describing Virtis-H calibration is VVX-LES-RP-2333 (Version 1.4), named VIRTIS\_H\_CALIBRATION.PDF in the PSA archive.

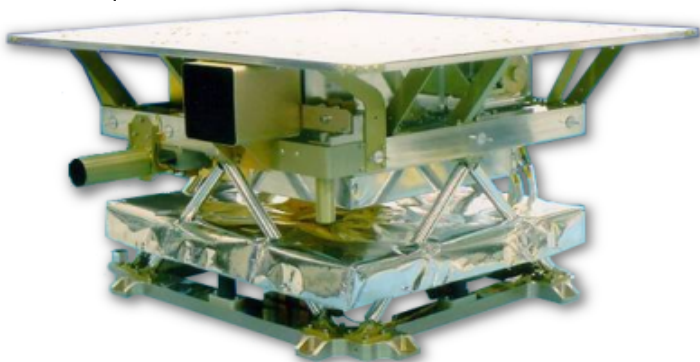
# Virtis-H calibration

Florence HENRY

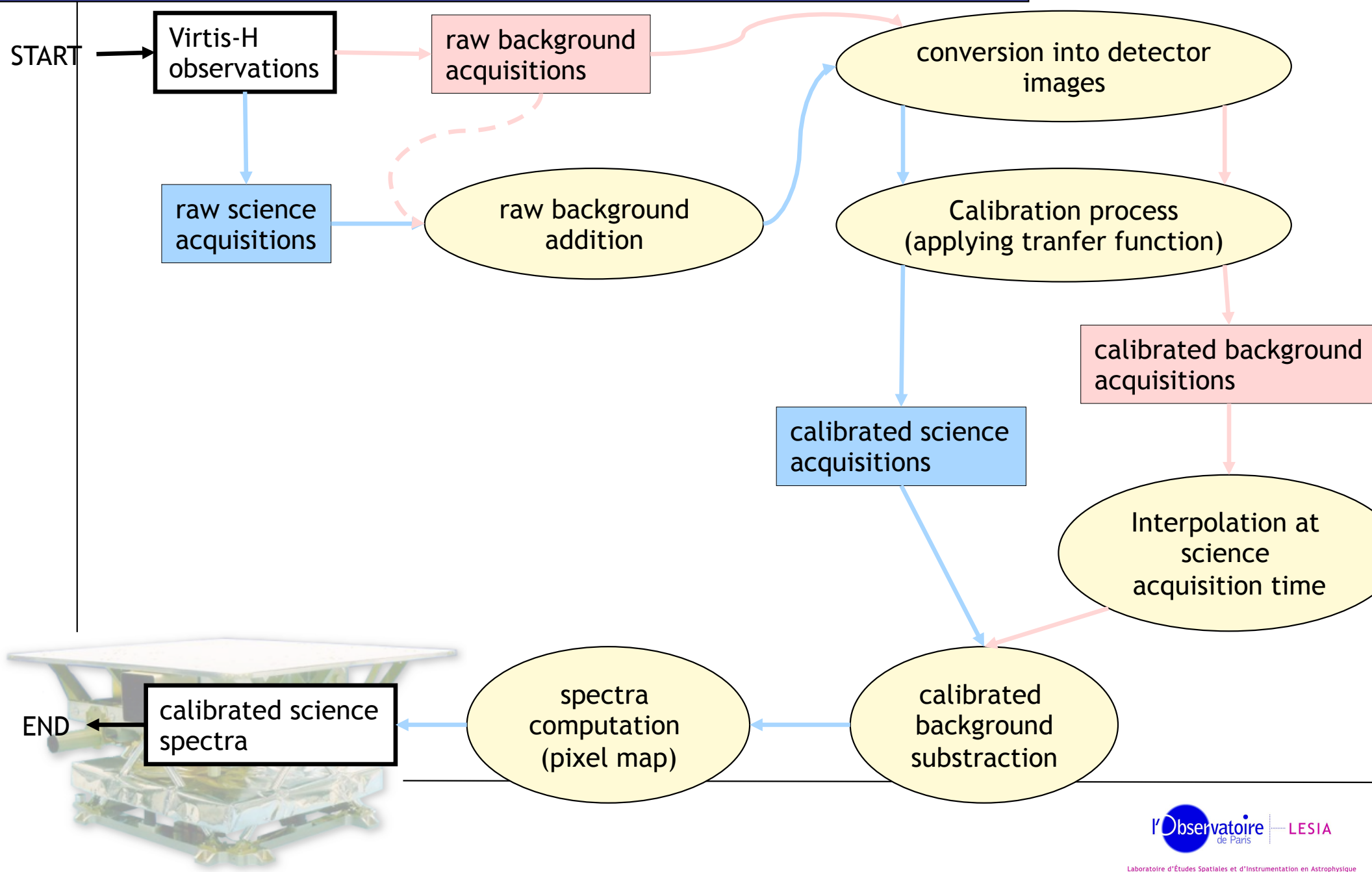
[florence.henry@obspm.fr](mailto:florence.henry@obspm.fr)

*Jean-Michel REESS*

*jean-michel.reess@obspm.fr*



# Virtis-H calibration : basic scheme



## Computing transfer function (1/2)

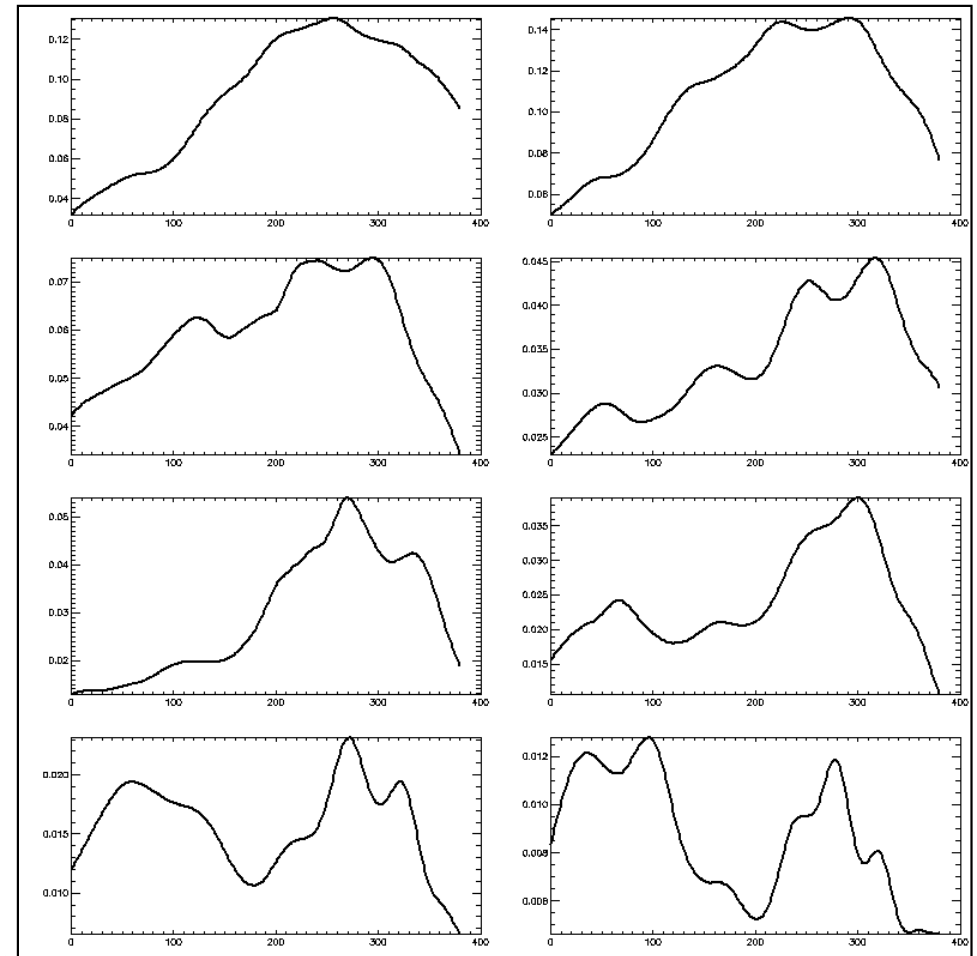
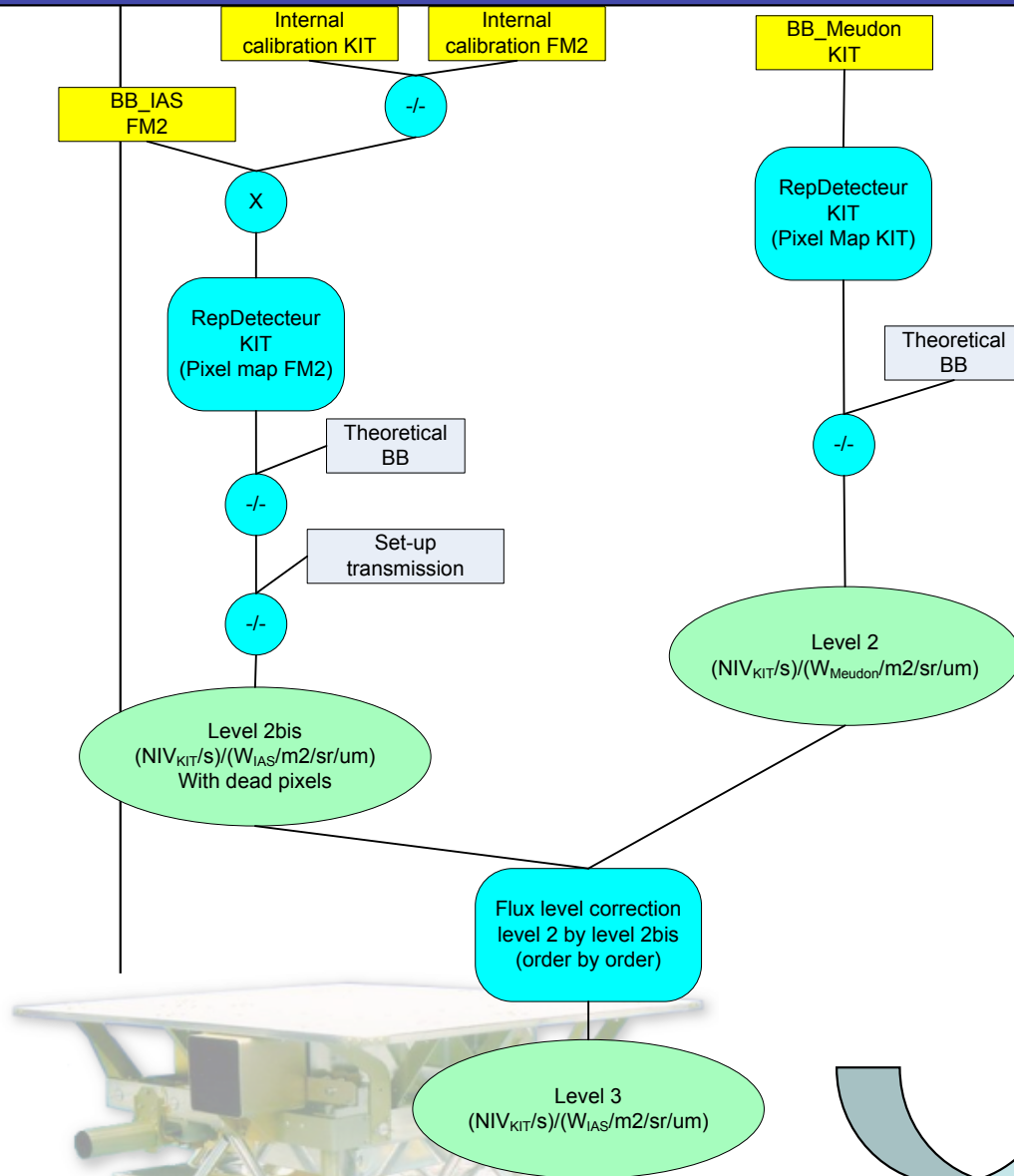
- **Difficulties to get the transfer function:**
- Detector has been changed after ground calibration
- Partial calibration performed with the new detector (full with the old one)
- Need to use different measurements with the old and the new detector to get the correct transfer function



Florence HENRY - Virtis-H calibration  
Venus Express VIRTIS and VMC Data Workshop  
12-15 May 2014, ESAC, Madrid, Spain



# Computing transfer function (2/2)



# Applying transfer function

This function handles the detector response dependencies with:

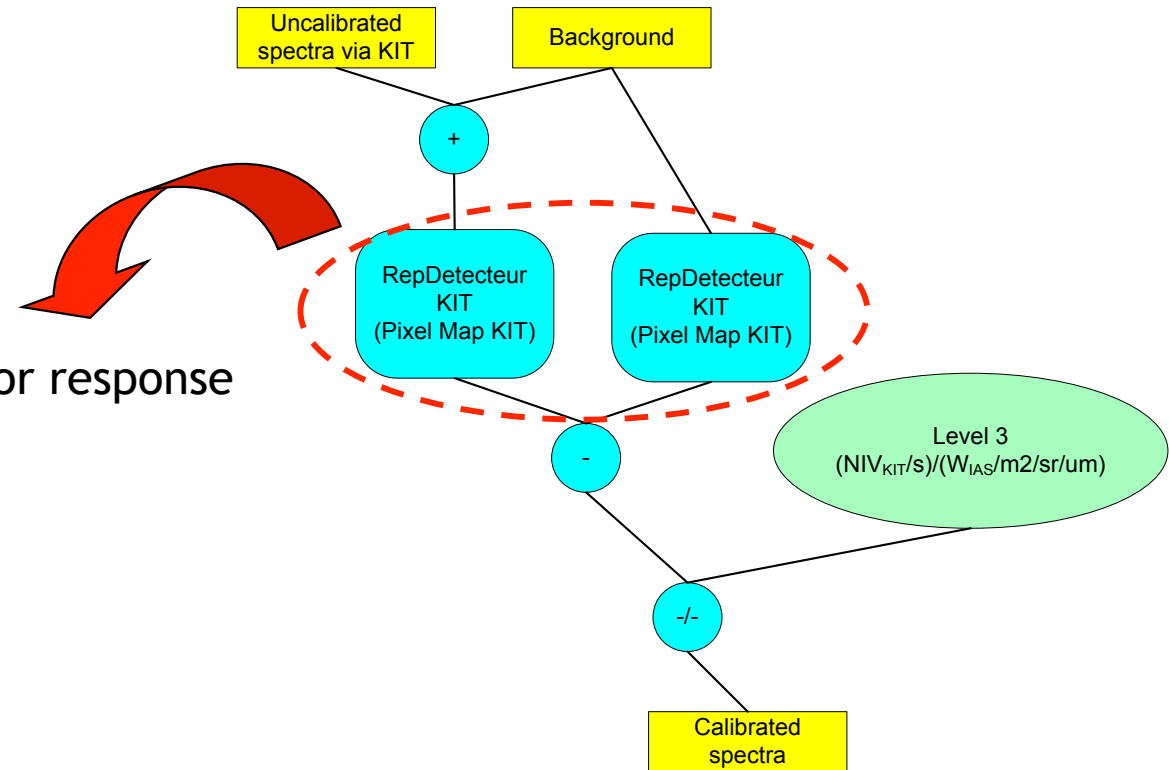
- the integration time:

$$\text{ADU} = f(t), f \text{ constant}$$

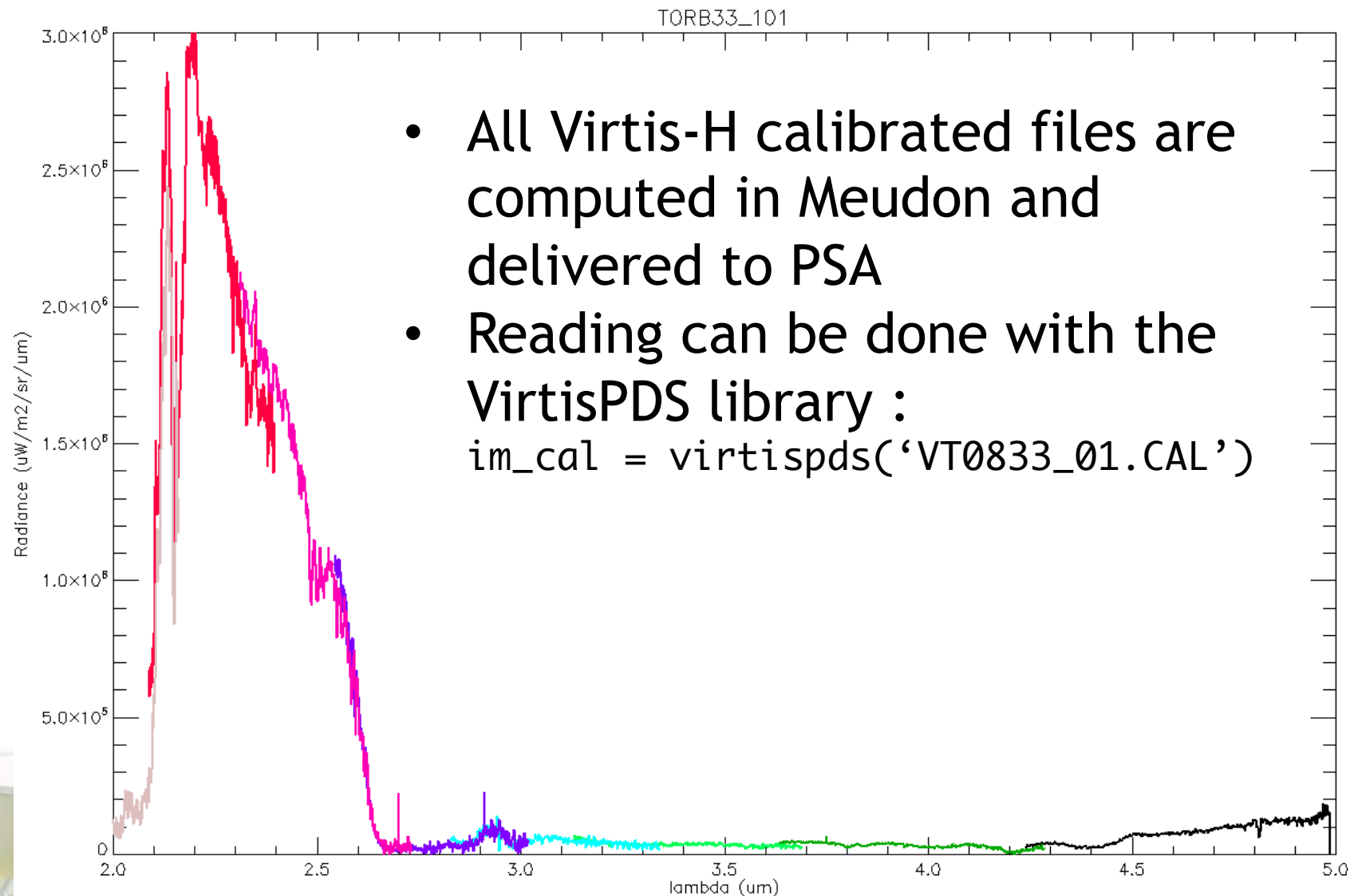
- the flux level

$$\text{ADU}/s = g(f), \text{ADU constant}$$

*Use of the detector calibration @ subsystem level*



# Transfer function : results



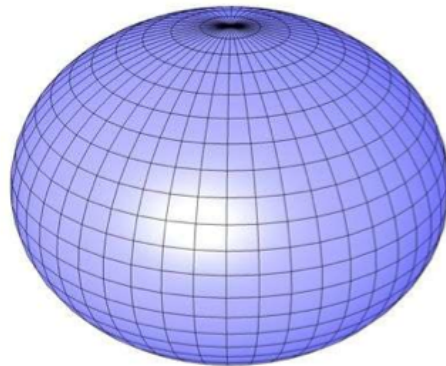


## Chapter 6

# Computation of the Geometry

# Computation of the GEOMETRY

## General description



Sophie JACQUINOD [sophie.jacquinod@obspm.fr](mailto:sophie.jacquinod@obspm.fr)  
Stéphane ERARD [stephane.erard@obspm.fr](mailto:stephane.erard@obspm.fr)

# Access to geometry files

- Geometry documentation :
  - Geometry file documentation : **VVX-LES-SW-2268**  
(Issue 1.2, July 2008)
  - EAICD : **VVX-LES-IC-2269**  
(Issue 1.6, April 2012)
  - The Virtis PDS/IDL Software manual : **VVX-LES-SW-2264**  
(Issue 2.8.1, October 2010)
- File names : **<data file root>.GEO**
- Reading software : **v\_pds library** (version 2.8)
  - Example to read a cube :  
image = **virtispds**('data\_file\_root.CAL')  
geom = **virtispds**('geom\_file.GEO')

# Main points

One software for Virtis-M & H, called GEOVIRTIS:

- same computing method for M & H
- specific IDL library developed and maintained in LESIA
- rely on the Spice toolkit (NAIF/JPL) for IDL (ICY, version N0062)

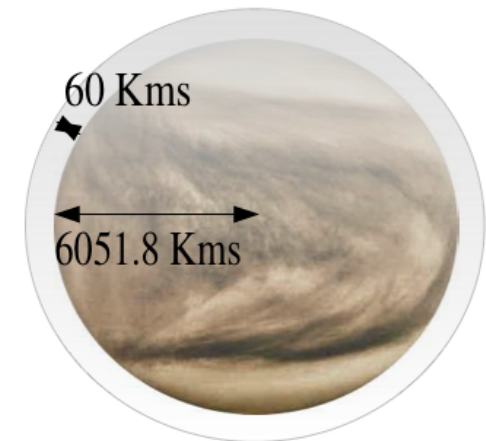


# Main points

- Geometric files have to be generated several times, as navigation Spice kernels are updated by ESA
  - 1<sup>st</sup> time : when the raw data file is generated by the EGSE, using predicted orbital parameters
  - 2<sup>nd</sup> time : about three weeks later, when the final (reconstructed) orbital parameters and time correlation files become available.
  - The first generation of geometry files have extension “PRE”, whereas the final version have extension “GEO”. Only the .GEO files, which are the final versions, are included in the archive.
- One geometric file associated to each data file
  - geometric files are relevant to one focal plane only (separated files for H, M-vis, M-IR)
  - when processing a data file, only the corresponding geometry file needs to be loaded

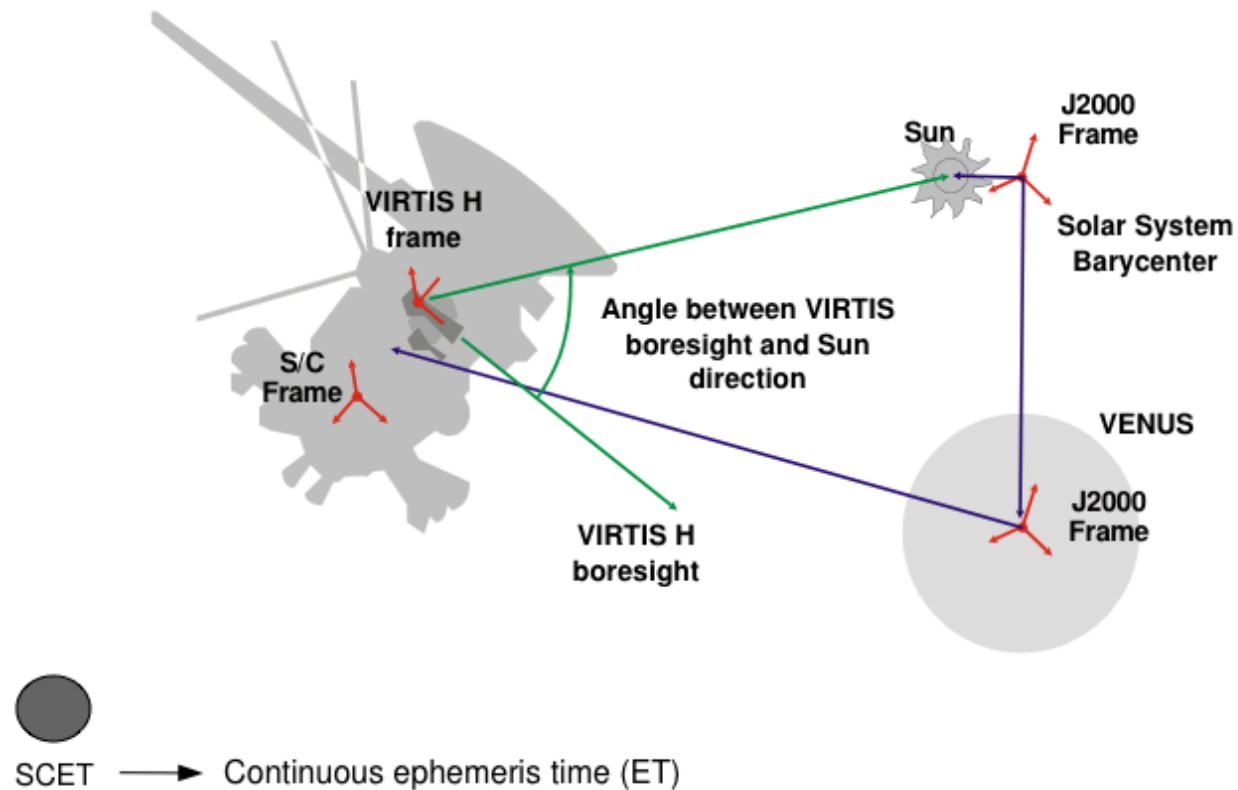
# Main points

- Two projection surfaces are used to allow for simple interpolation of the footprint at various altitudes in the atmosphere :
  - 1<sup>st</sup> projection surface : **a sphere with 6051.8 km radius** : Venus reference surface used in SPICE kernels, (IAU 2000 standard)
  - 2<sup>nd</sup> projection surface : **a cloud layer at 60km** above the reference ellipsoid (This particular altitude is selected to match approximately the upper cloud layer on the day side).



# Tools used to compute geometry

- SPICE IDL routines (ICY Library)  
→ SPICE deals with observation geometry and time



# Tools used to compute geometry

- SPICE kernels given by ESA : 7 types

**SPK**

- Space vehicle ephemeris (trajectory)
- Planet, satellite, comet and asteroid ephemerides
- Position of something relative to something else

**PCK**

- Planet, satellite, comet and asteroid orientations, sizes, shapes
- Possibly other similar “constants” such as parameters for gravitational model or atmospheric model

**IK**

- Instrument information such as:
  - Field-of-View specifications
  - Internal timing

**CK**

- Spacecraft attitude
- Orientation of something relative to a specified reference frame

# Tools used to compute geometry

**FK**

- **Frames Definitions**
  - Definitions and specifications of relationships between reference frames (coordinate systems)

**LSK**

- **Leapseconds Tabulation**
  - Used for UTC <--> ET time conversions

**SCLK**

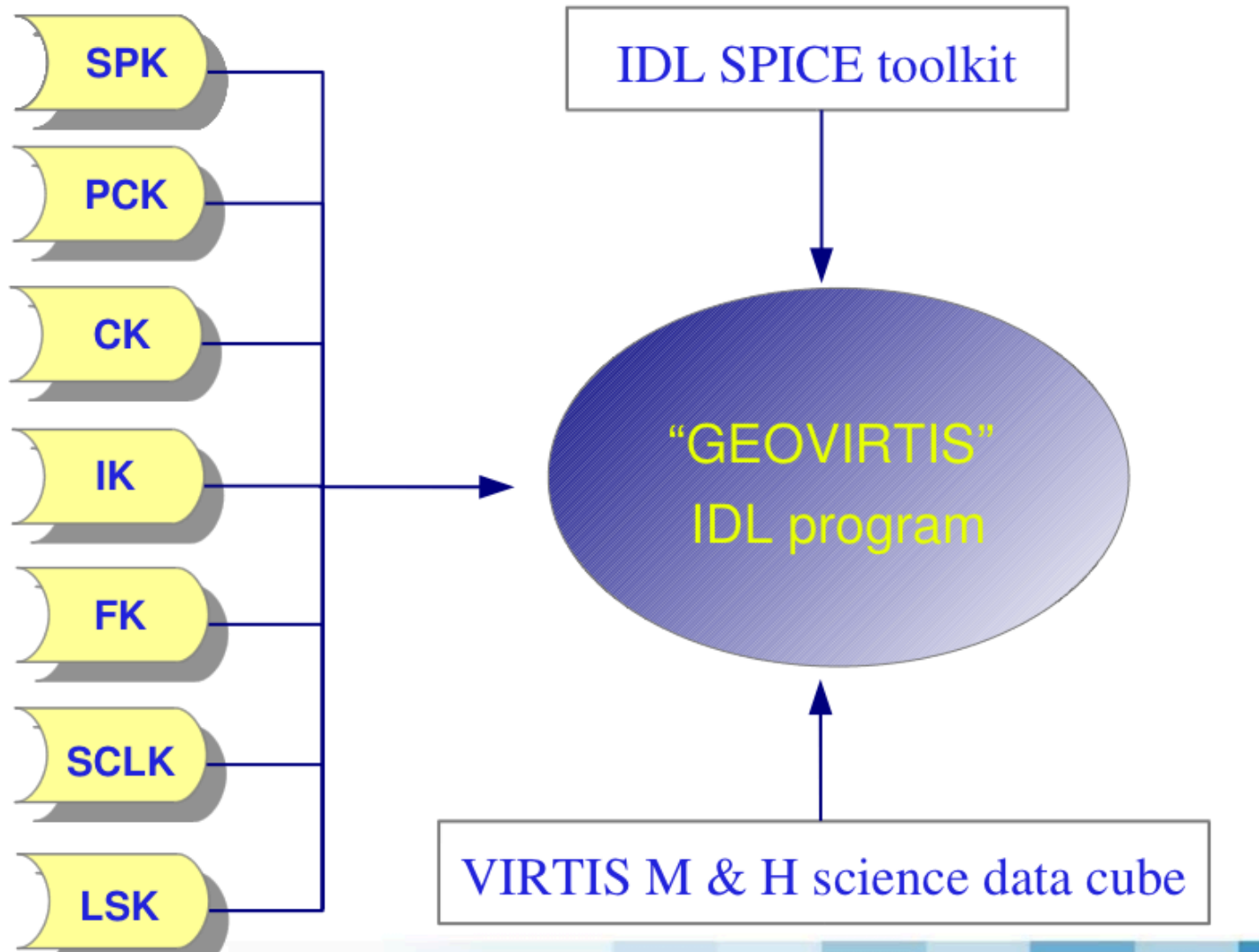
- **Spacecraft Clock Coefficients**
  - Used for SCET <--> ET time conversions

- **Home made kernels :**
  - mirror positions CK (for V-M) by G. Piccioni
  - New cloud body PCK & SPK

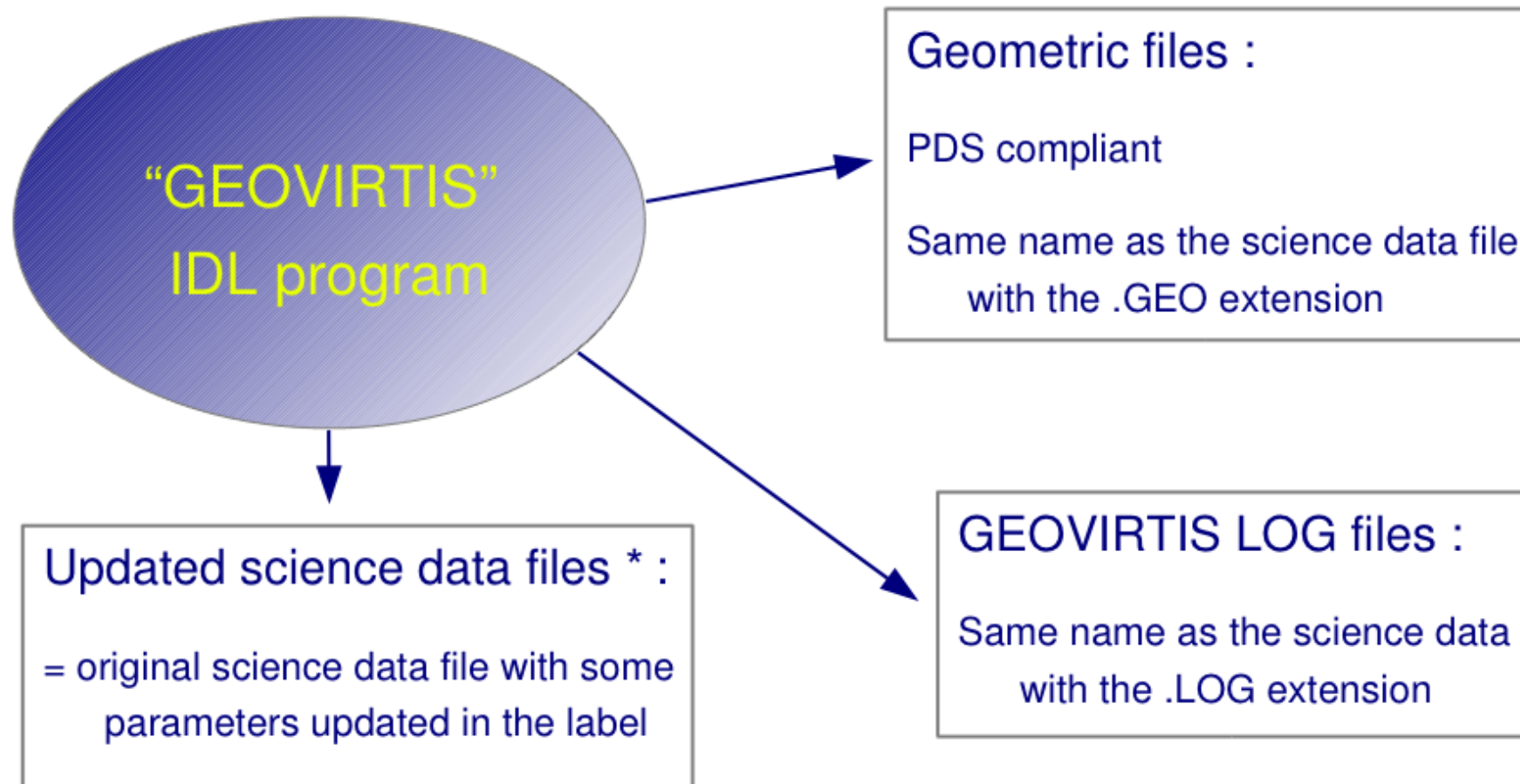
# Tools used to compute geometry

- All this tools allow to :
  - change between different types of time :  
**SCET ↔ UTC ↔ ET**
  - compute coordinates in different frames :  
**VIRTIS frame ↔ VEX frame ↔ Venus frame**
  - make geometry computations such as:  
computing distances and intersections between lines,  
planes and ellipsoids in 3D, normal vector to the  
surface ...

# GEOVIRTIS inputs



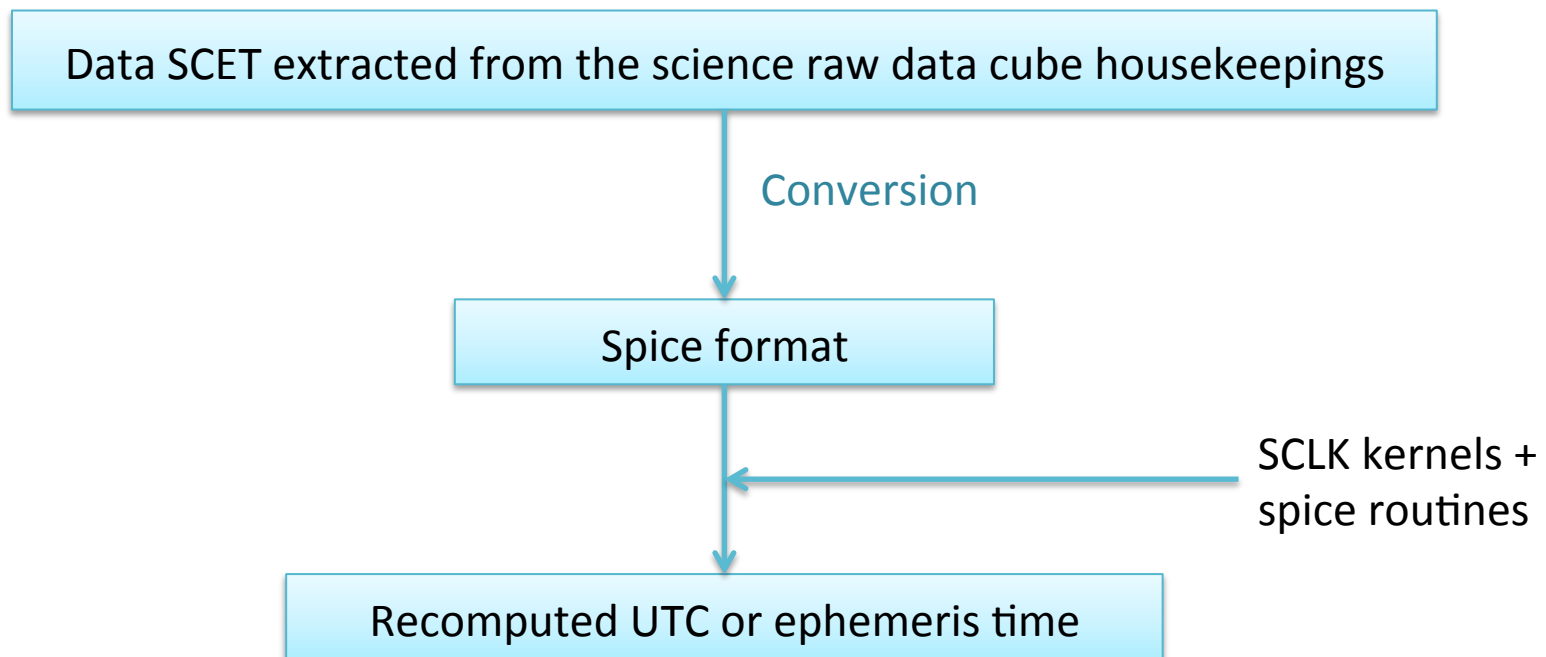
# GEOVIRTIS outputs





# SCET and Ephemeris Time

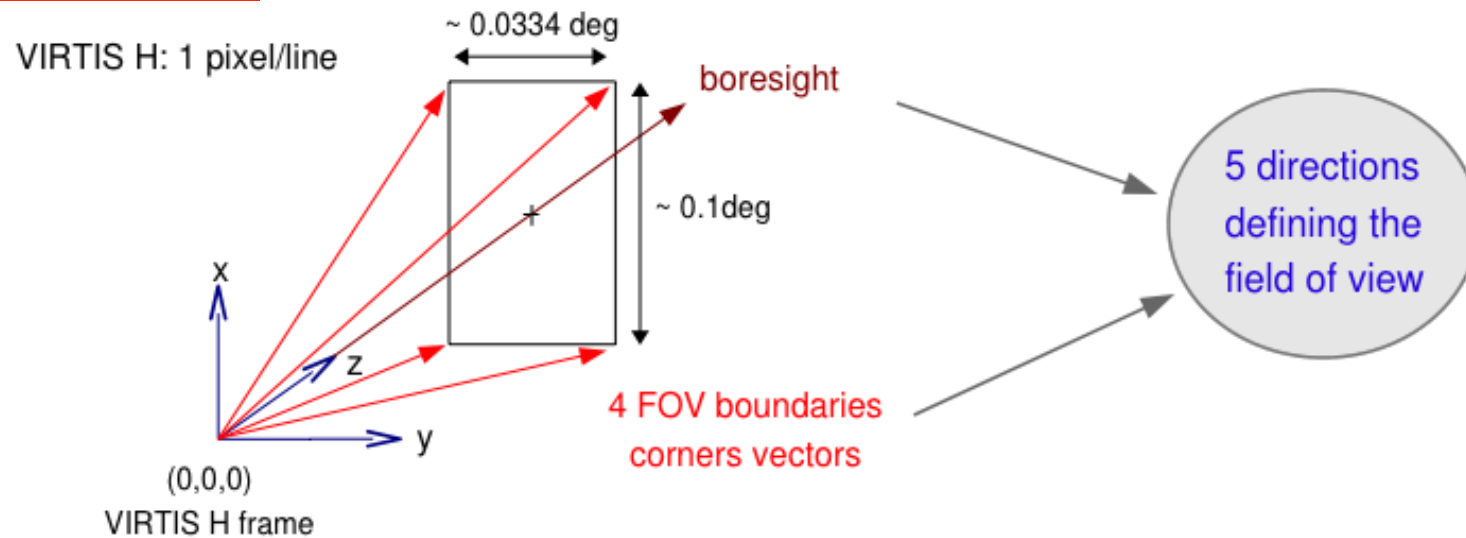
- Observation data time is recomputed from SCET & SCLK kernel



# Instrument Field of View

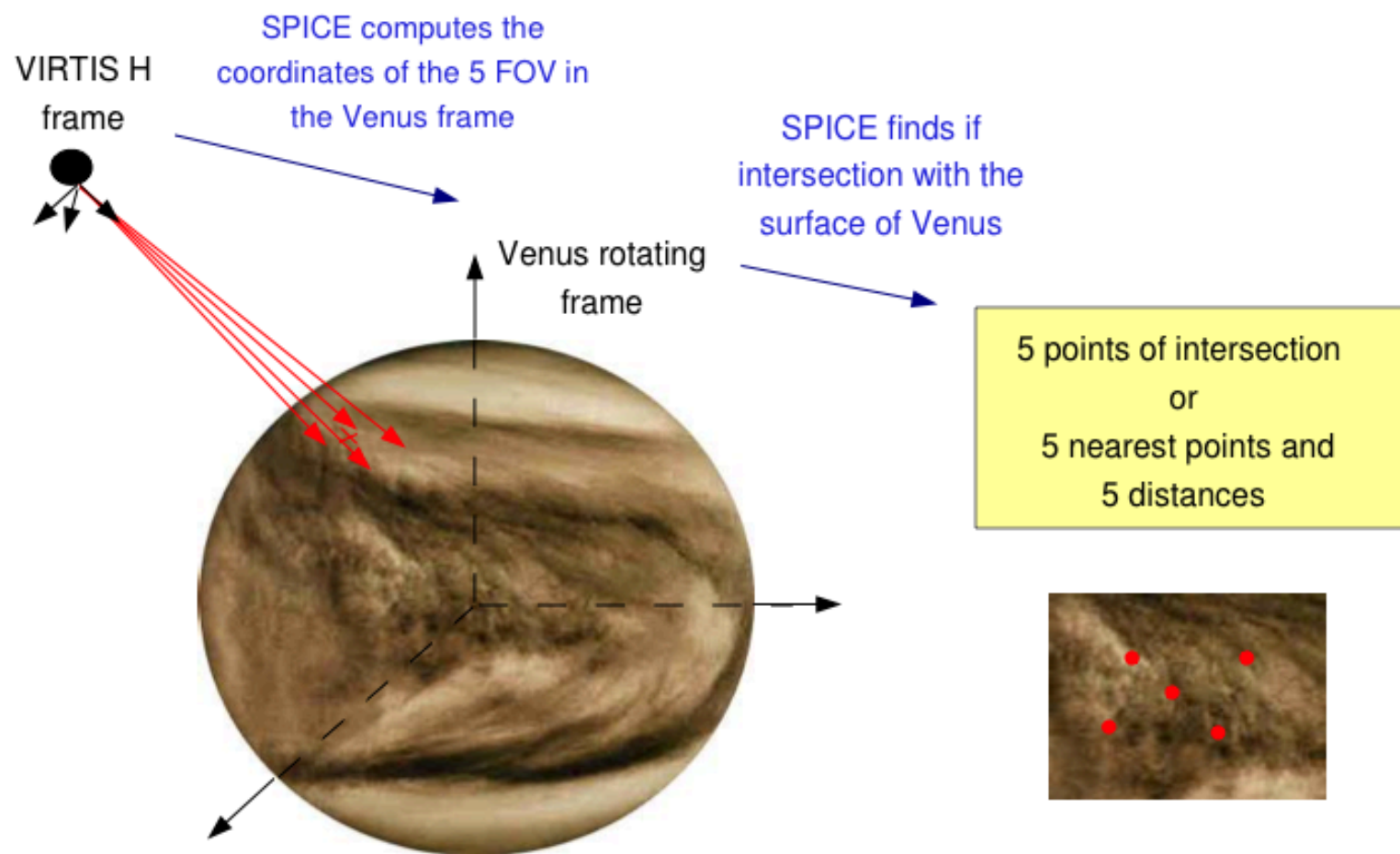


## Example for V-H :



# Instrument footprint

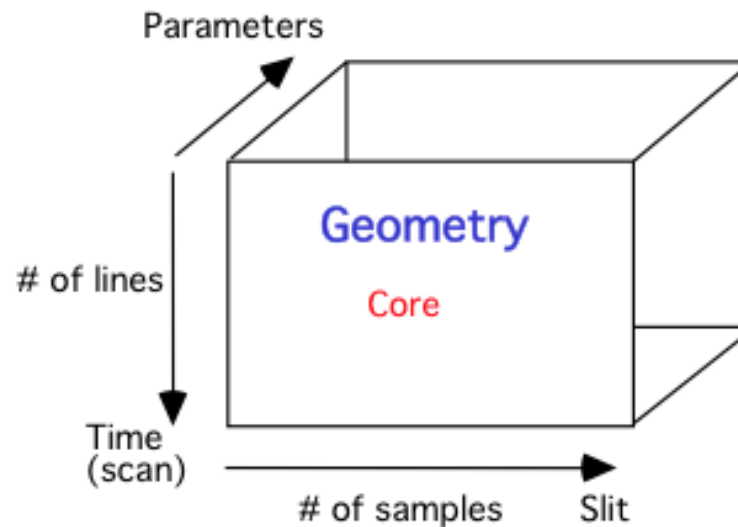
At time ET :



# Geometric cubes size

- One Geometric cube per science data cube
- Same X/Y spectrum reference as science data cube

33 parameters for M  
41 parameters for H } computed at mid-exposure



# Order of the computed parameters

Plane #	Parameter description
1-4	Longitudes of 4 pixel footprint corner points
5-8	Latitudes of 4 pixel footprint corner points
9-10	Longitude & latitude of pixel footprint center on surface ellipsoid
11-13	Incidence, emergence & phase at footprint center, relative to Venus center direction
14	Surface elevation (footprint corners average)
15	Slant distance (line of sight from spacecraft to surface ellipsoid at pixel center)
16	Local time at footprint center
17-20	Longitudes of 4 corner points on cloud layer
21-24	Latitudes of 4 corner points on cloud layer
25-26	Longitude & latitude of pixel center on cloud layer
27-29	Incidence, emergence & phase, relative to local normal of cloud layer
30	Surface elevation at the vertical of cloud layer intercept
31-32	Right ascension and declination of pointing direction.

Geometrical projection  
on surface ellipsoid

Geometrical projection on  
cloud layer (60 km)

# Order of the computed parameters

For M:	
33	One frame-common plane
	1-2 Original data SCET from TM
	3-4 UTC
	5-6 Sub-spacecraft coordinates (longitude/latitude)
	7-8 Sine and cosine of M mirror angle
	Sun direction: 9: angle between Sun direction and Virtis Z axis; 10: azimuth of Sun direction in instrument XY plane (counted from 0° at X axis).
For H:	9 supplementary planes
33-34	Original data SCET from TM
35-36	UTC
37-38	Sub-spacecraft coordinates (longitude/latitude)
39	Slit orientation
40-41	Sun direction: 40: angle between Sun direction and Virtis Z axis; 41: azimuth of Sun direction in instrument XY plane (counted from 0° at X axis).

M frame-common information

H reconstructed information / pixel

# Units & Multiplying factors

- All parameters stored as long integers
- Angles :  $\text{deg} \times 10\,000$
- Distances : meters
  - no intersection on surface :  $\text{offset} = + 20\,000$
  - Surface elevation missing value =  $- 20\,000$

# Detailed description of each parameters

- Geometric information that is considered to be constant within the data cube sub-session is stored in the label of the geometry files :
  - **Longitude and latitude of sub-solar point** : nearest point on Venus to Sun. Stored in degrees.
  - **Solar distance**: distance from Venus to Sun. Stored in Kilometers.
  - **Solar longitude** : The [planetocentric](#) longitude of the sun, as seen from Venus. Stored in degrees.
  - **Spacecraft orientation**: The spacecraft orientation corresponds to the coordinates of the normalized spacecraft speed vector with respect to Venus in the spacecraft reference frame.

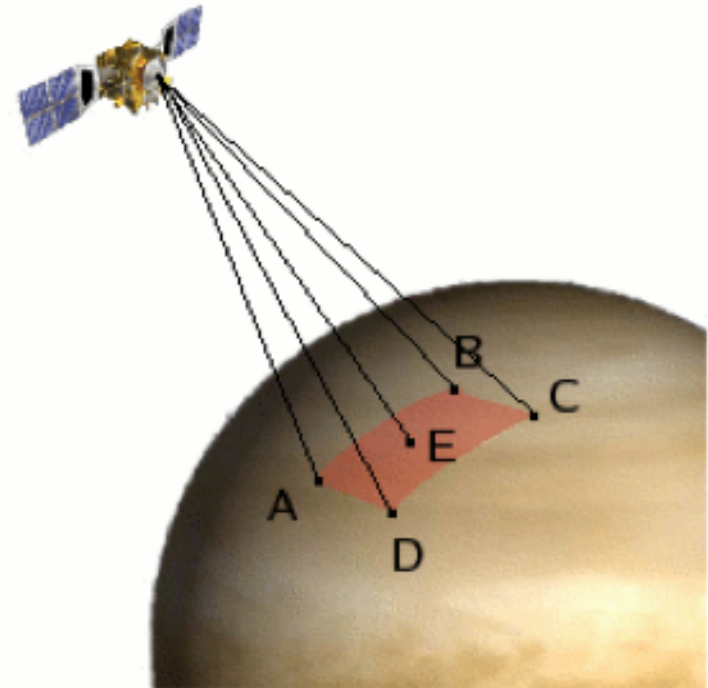


# Detailed description of each parameters

- **Spacecraft pointing mode:** can be **NADIR**, **ACROSSTRACK**, **ALONGTRACK** or **NADIR\_POW** for a nadir pointing, **INERT** for an inertial pointing or **MOSAIC** when pointing direction is moving from Venus direction within the sub-session.
- **Right ascension and declination** of pointing direction: set to non-null values for an INERT pointing mode.
- **Maximum and minimum latitudes / Easternmost and westernmost longitudes.** Stored in degrees.
- **Slant distance:** mean slant distance for all the cube sub-session. Stored in Kilometers.

# Detailed description of each parameters

- Geometric parameters computed on a pixel basis are stored in the different planes of the cube:
  - **Parameters 1 to 10: Longitudes and latitudes of the four pixel footprint corner points A, B, C & D and of the pixel footprint center E on the surface ellipsoid**
    - E corresponds to the boresight extension on the ellipsoid.
    - Longitudes and latitudes are stored in degrees multiplied by 10 000.
    - Longitudes are increasing eastward.

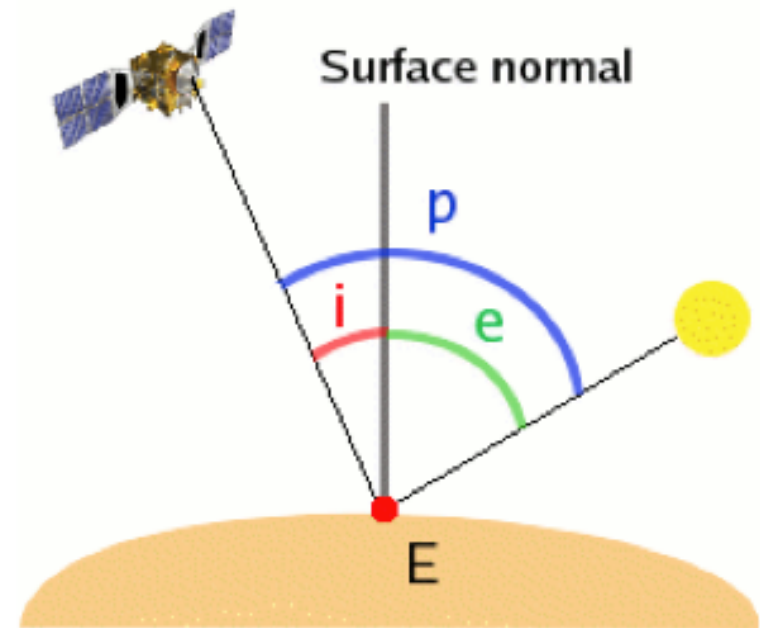


# Detailed description of each parameters

- **Parameters 11 to 13:** Lighting angles (**incidence, emergence and phase**) on the Venus ellipsoid reference surface

→ These angles are computed at point E with respect to the surface normal (= Venus center direction as Venus is spherical)

→ Angles given in degrees x 10 000



*i: incidence angle, e: emergence angle, p: phase angle*

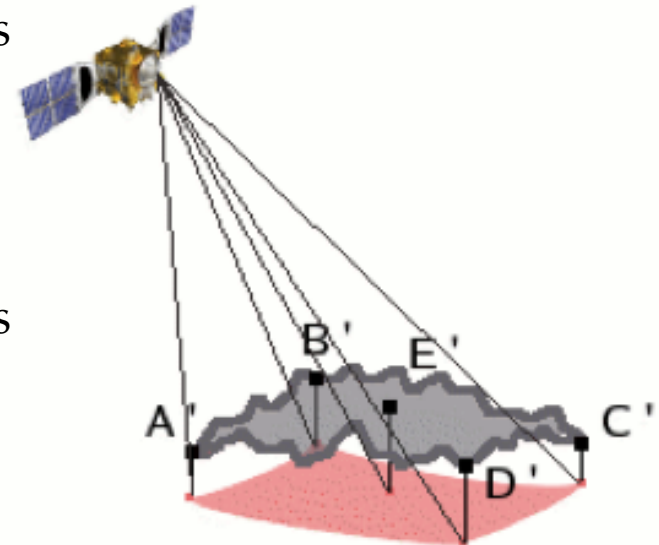
# Detailed description of each parameters

– **Parameters 14: Surface elevation.** The surface elevation corresponds to an average on points A', B', C', D' and E', [on Venus topography](#)

→ If one of the corner value is missing, the elevation is taken at central point E'. When this value is missing, a -20 000 default value is given

→ Elevation is stored in meters

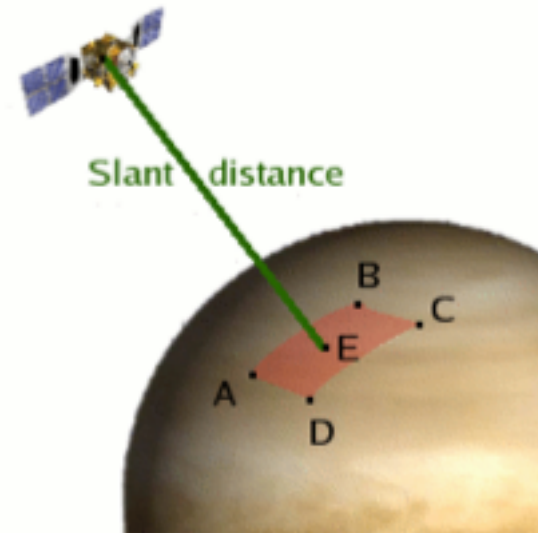
→ Elevation = Magellan topography – Venus mean radius (6051.8 km)



# Detailed description of each parameters

- **Parameters 15: Slant distance.** Distance between spacecraft and pixel footprint center E

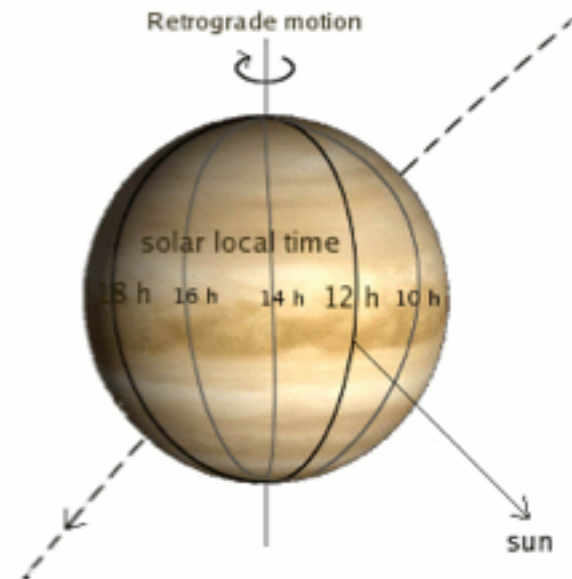
→ Slant distance is stored in meters



# Detailed description of each parameters

## – Parameters 16: Solar local time

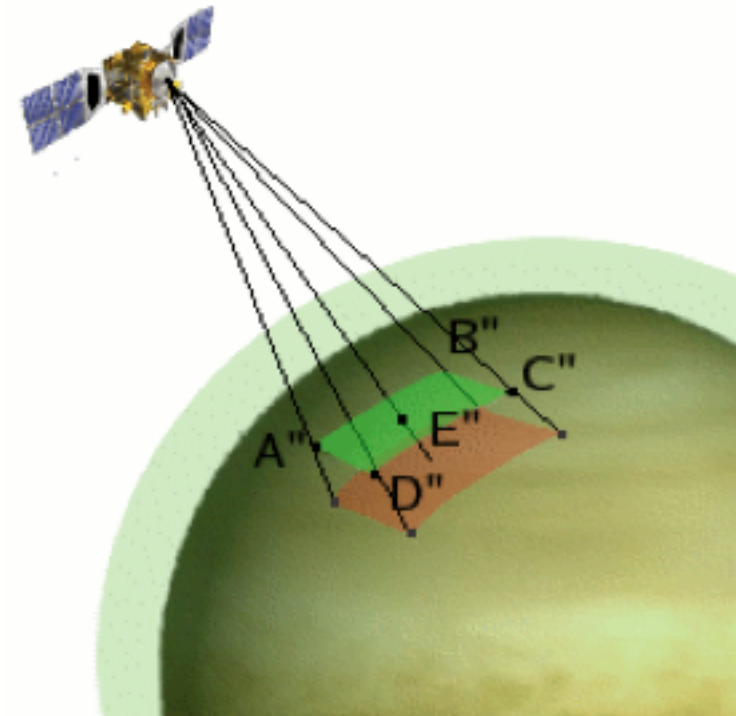
- Given at point E.
- Increasing with time.
- Stored in units of Venus period/24 multiplied by 100 000.



# Detailed description of each parameters

- **Parameters 17 to 26: Longitudes and latitudes of the four pixel footprint corner points  $A''$ ,  $B''$ ,  $C''$  &  $D''$  and of the pixel footprint center  $E''$  on the cloud surface**

- $E''$  corresponds to the boresight extension on the cloud surface.
- Longitudes and latitudes are stored in degrees multiplied by 10 000.

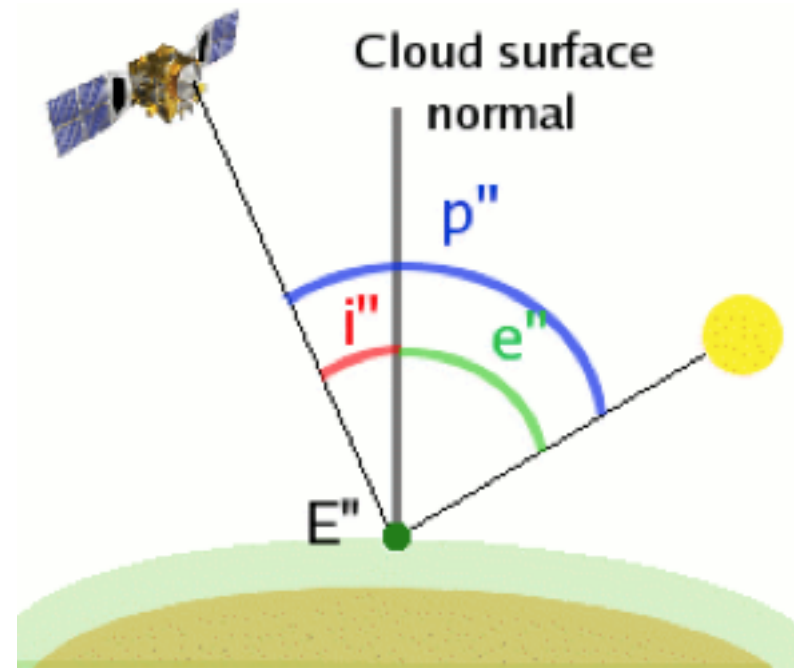


# Detailed description of each parameters

- **Parameters 27 to 29:** Lighting angles (**incidence, emergence and phase**) on the cloud surface

→ These angles are computed on  $E''$  with respect to cloud surface normal (= Venus center direction as Venus cloud surface is spherical)

→ Angles given in degrees  $\times 10\,000$



$i$ : incidence angle,  $e$ : emergence angle,  $p$ : phase angle

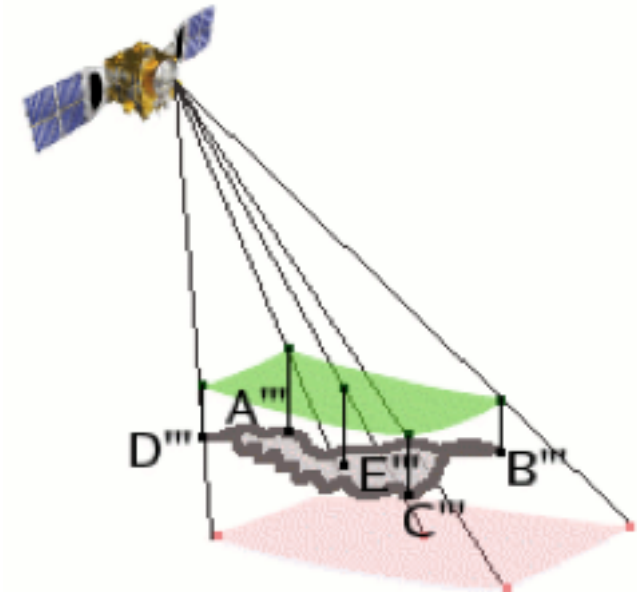


# Detailed description of each parameters

- **Parameters 30: Surface elevation** at the cloud footprint vertical. The surface elevation corresponds to an average on points  $A'''$ ,  $B'''$ ,  $C'''$ ,  $D'''$  and  $E'''$ , at the vertical of points  $A''$ ,  $B''$ ,  $C''$ ,  $D''$  and  $E''$  on Venus topography.

→ If one of the corner value is missing, the elevation is taken at central point  $E'''$ . When this value is missing, a -20 000 default value is given

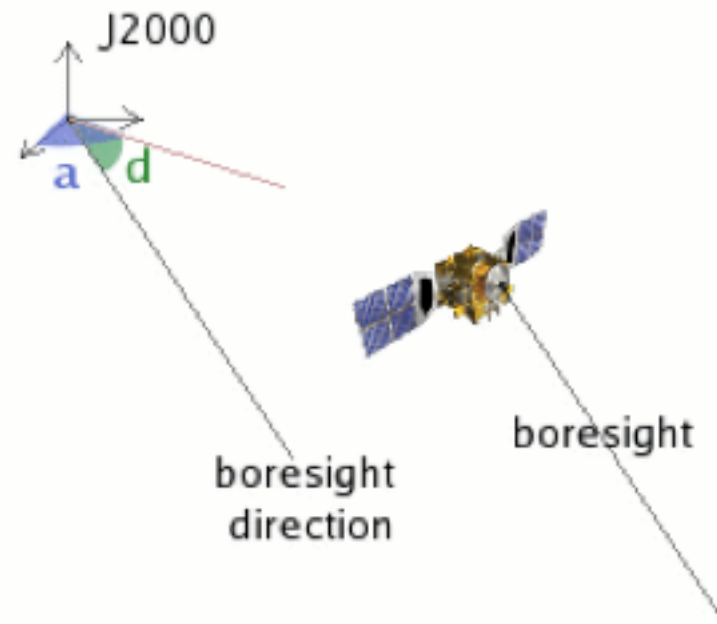
→ Elevation is stored in meters



# Detailed description of each parameters

- **Parameters 31 and 32: Right ascension  $a$  and declination  $d$  of the line of sight in the J2000 frame**

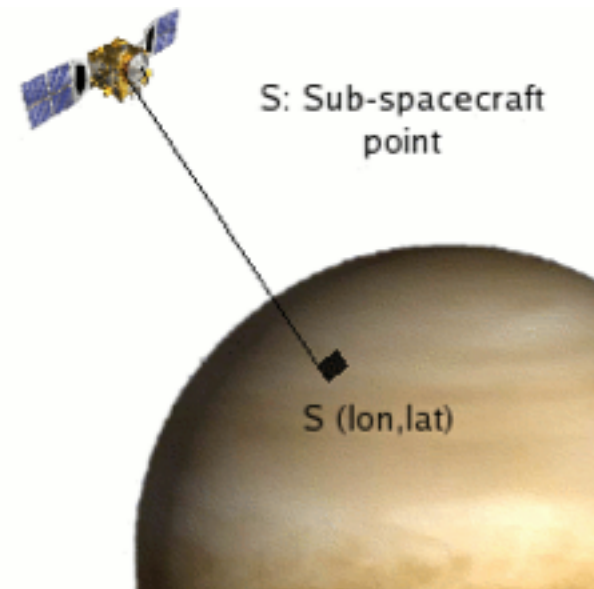
→ Angles given in degrees  $\times 10\,000$



# Detailed description of each parameters

- Special parameters stored in supplementary planes :
  - **Sub-spacecraft coordinates** (parameter 33/5-6 for V-M, 35 and 36 for V-H) : **Longitude and latitude of the sub-spacecraft point.**

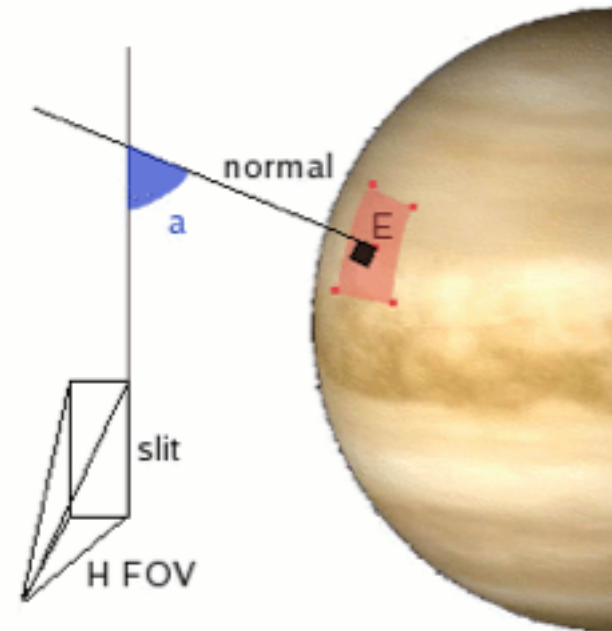
→ Angles given in degrees  $\times 10\,000$



# Detailed description of each parameters

- **Slit orientation for Virtis-H** (parameter 39): **Angle between the long side of H slit and the surface normal** at point E

→ Angles given in degrees  $\times 10\,000$



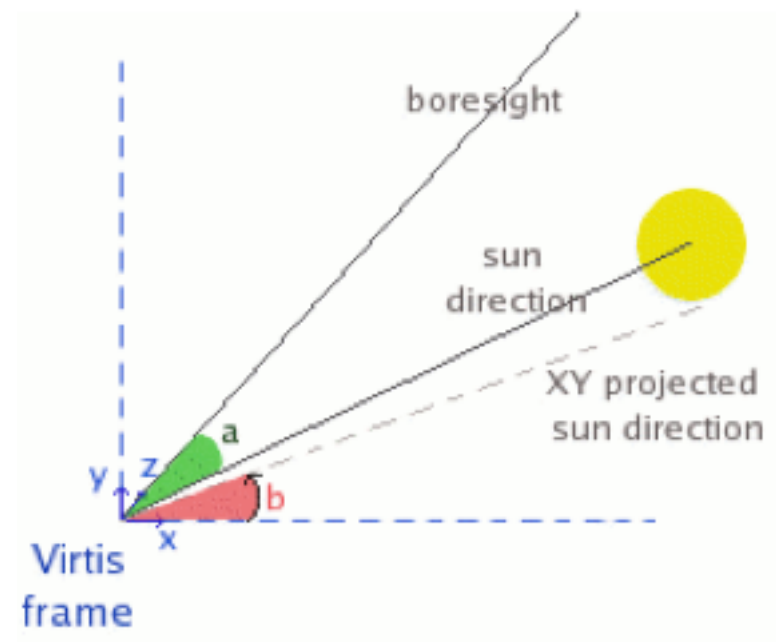
# Detailed description of each parameters

- **Sun direction : 2 parameters** (parameter 33/9-10 for V-M, 40 and 41 for V-H):

→  **$a$  : Angle between the sun direction and boresight** (Virtis Z axis)

→  **$b$  : Azimuth of the sun direction in the Virtis XY plane relative to the Virtis X axis** (0 on S/C axis)

→ Angles given in degrees x 10 000



# Detailed description of each parameters

- **Original data SCET from TM : 2 parameters** (parameter 33/1-2 for V-M, 33 and 34 for V-H)
  - The **first value** stores the **integer part** (the SCET first two words),
  - The **second one** stores the **fractional part** (third word).
- **UTC : 2 parameters** (parameter 33/3-4 for V-M, 35 and 36 for V-H)
  - Encoded UTC recomputed through the spice system.
  - The **first value** contains the **number of days since January, 1st, 2000**
  - The **second one** contains the **number of seconds since 0h**, multiplied by 10 000.
- **Sine and cosine of M mirror angle :** (parameter 33/7-8)
  - Values taken from the HK, multiplied by 1 000.

# Limb observations and restituted distances

- When observations intercept the surface

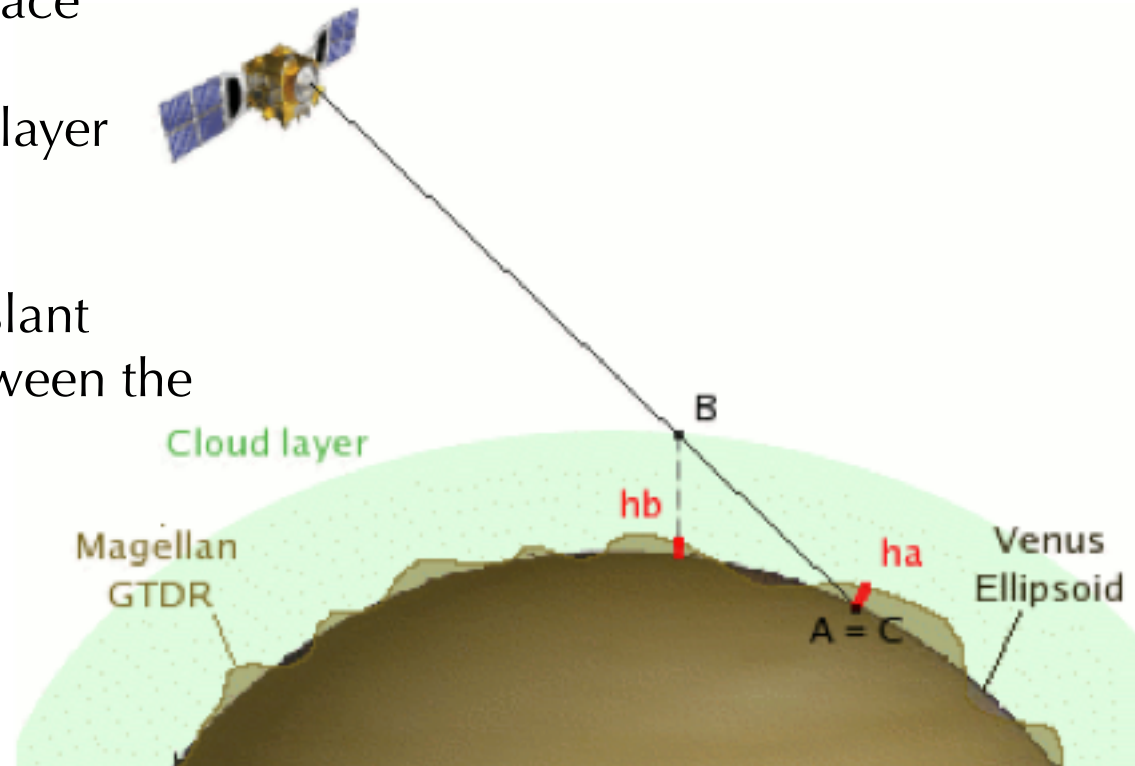
**A** : footprint on Venus surface (reference ellipsoid)

**ha** : elevation on Venus surface

**B** : footprint on cloud layer

**hb** : elevation below cloud layer

In this case, the computed slant distance is the distance between the spacecraft and point A.



# Limb observations and restituted distances

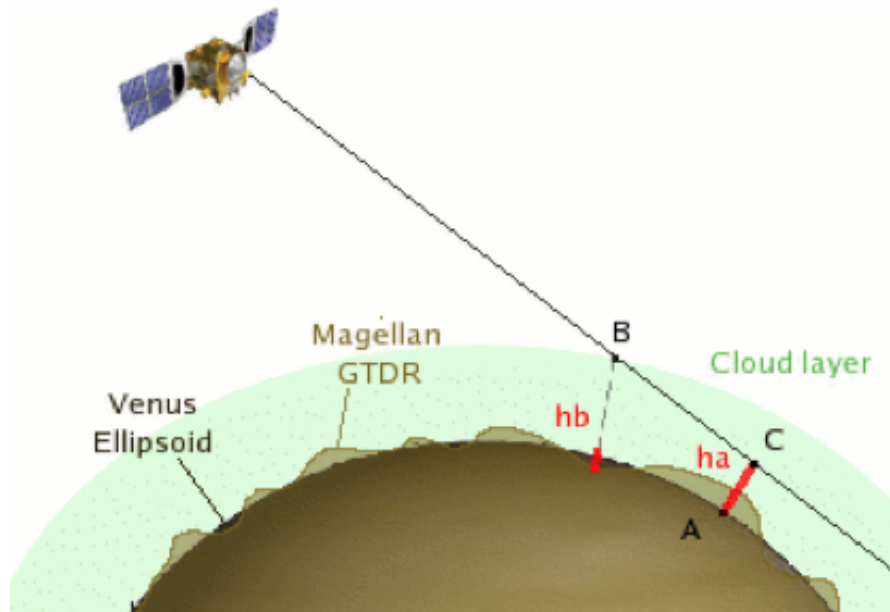
- When the line of sight doesn't intercept the surface :  
 → 2 possible cases  
 1. intersection with cloud layer. No intersection with Venus surface

**A** : nearest point on Venus surface to the line of sight

**ha** : distance from Venus to line of sight = distance AC (+ 100 000 m offset)

**B** : footprint on cloud layer

**hb** : elevation below cloud layer



Surface elevation at the ellipsoid intercept (**plane 14**) is substituted by the **tangent altitude** (impact parameter above the surface) with the addition of a large offset (100 000 m). This offset is intended to select or filter limb observations easily inside a file. **The 100 000 m offset must be subtracted from plane 14 to retrieve the tangent altitude.**

Surface elevation at the cloud layer intercept (**plane 30**) is maintained whenever possible



# Limb observations and restituted distances

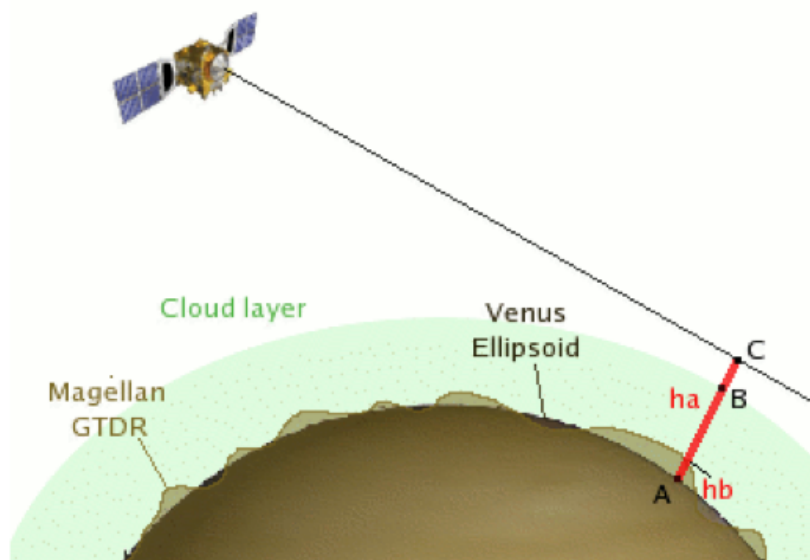
## 2. No intersections

**A** : nearest point on Venus surface to the line of sight

**ha** : distance from Venus to line of sight = distance AC (+ 100 000 m offset)

**B** : nearest point on cloud layer to the line of sight

**hb** : elevation below cloud layer



Surface elevation at the cloud layer intercept (**plane 30**) is substituted by surface elevation at the vertical of the tangent point. A surface elevation is therefore always available in the geometry cubes, although not necessarily below the tangent point.

Angles, local time, and slant distance are computed at the intersection with the local vertical (tangent point).

The H slit orientation cannot be retrieved from surface coordinates, but is available in **plane 39** (computed at the tangent point).

# Missing values

- All missing values from Housekeepings are replaced by the **-2147483648** value which correspond to 80000000 in hexadecimal.
- When no CK-kernel information is available for one frame (that happened during the safe mode in Nov. 2006), all the geometric information for that particular frame is set to **- 2147483648.**

# Handling time

- All computations are done at the **middle of exposure**.
- When the integration time is not constant in one sub-session, no computation is done.
- START\_TIME and STOP\_TIME from label are recomputed through the Spice routines **from the first and last SCET values** (**NOT mid-exposure**).