# Architecture, specification and reusability of aided Navigation sensors

ESA Workshop on Avionics Data, Control and Software Systems (ADCSS), round table AGASSE

Emmanuel HUARD // 03 November 2010

emmanuel.huard@astrium.eads.net

All the space you need

ASTRIUM
AN EADS COMPANY

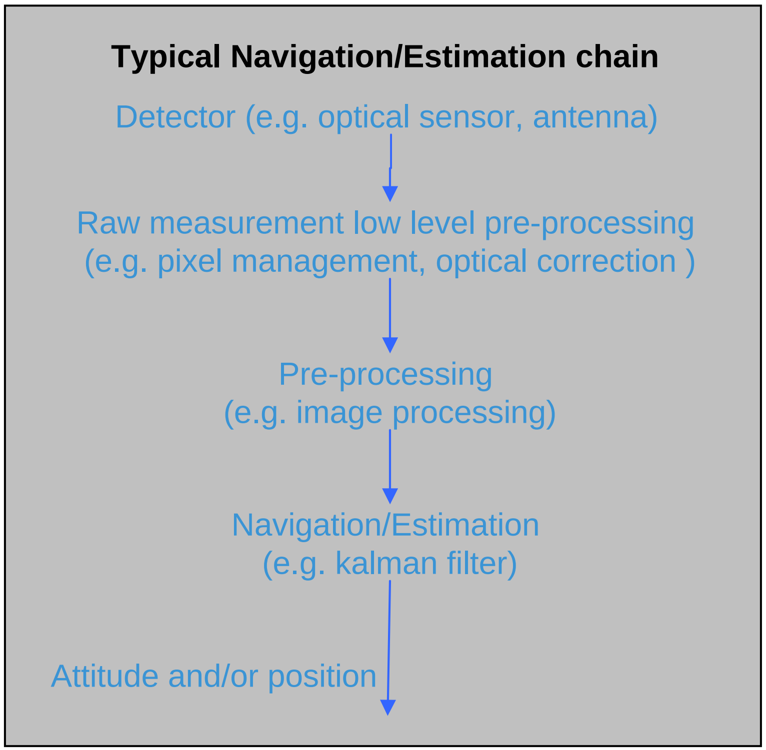
# Outline

- Introduction on Navigation/Estimation chains

- Aided navigation sensors architectures
  - The example of NPAL architecture
  - The example of VISNAV architecture
  - Conclusions

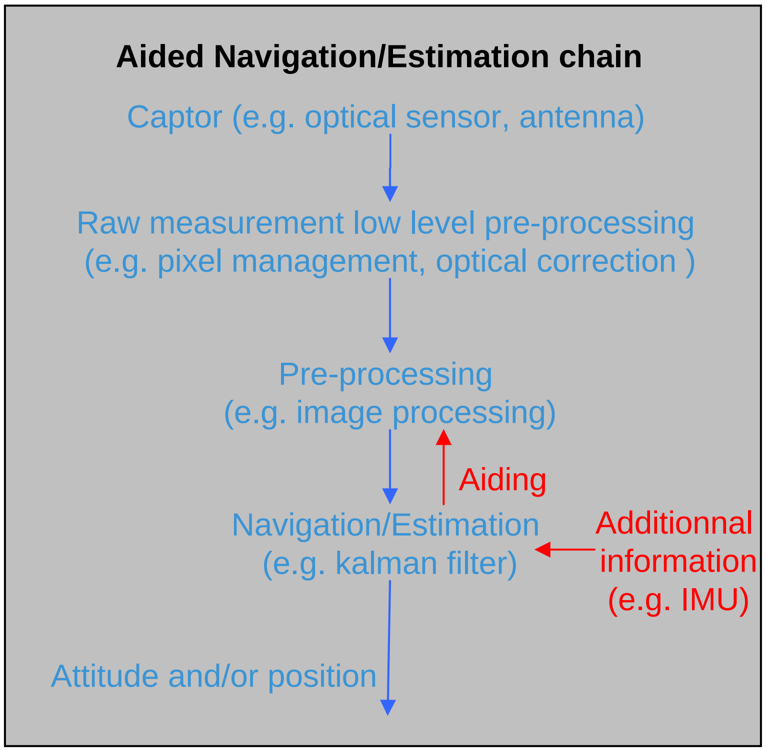
- Specifying algorithms of smart sensors
  - The example of the Image Processing (IP) for Vision Based Navigation (VBN)
  - Vision Based Navigation (VBN) Overall Validation Approach

# Introduction on Navigation/Estimation chains (1/2)




- Every Navigation/Estimation chain requires computation capacity which can be shared between:
  - The electronic of the sensor
  - The OBC
- The choice is performed by the system responsible according to the needs/constraints
  - E.g. the star tracker: Originally in the OBC, the attitude computation function has been moved to the sensor to simplify the interfaces, the validation and leading to products development.

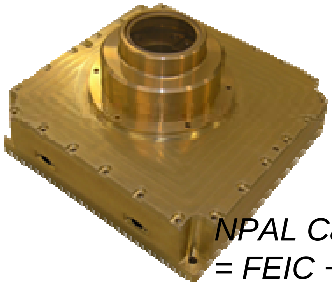
# Introduction on Navigation/Estimation chains (2/2)




- Aided navigation requires additional sensing information, typically IMU measurements, to improve the performance and robustness.
- This creates a tight coupling between the navigation and the measurements. This coupling is managed by the Navigation.
- It increases the data flow and the interfaces, thus requiring an optimisation of both the data exchange and the algorithms implementation. Both shall be optimised to master the overall performance and robustness.

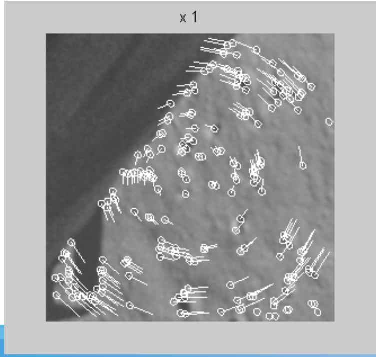
# NPAL - An optimized navigation chain for landing

- ESA-Science Critical Technology Program (2001-2005)
- Objectives of the project:
  - To provide for a system level demonstration to serve as a reference for planetary landers.
  - Proof of concept with the development of two key technologies: the FEIC (Feature Extraction Integrated Circuit) & Vision Based Navigation Camera (VBNC)
- Activities:
  - Design of an aided navigation system and camera for autonomous Landers
  - Development of a generic validation test bench, virtual scenes
  - Real Time validation of the navigation functional chain

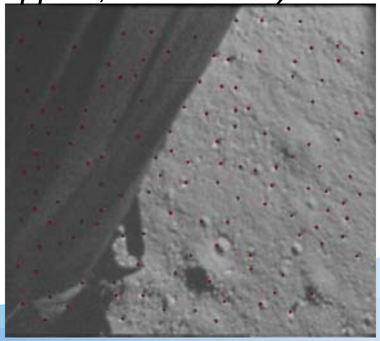


*NPAL Camera = FEIC +VBNC*

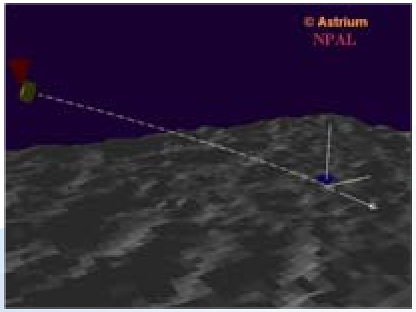
*Robustness Testing…*



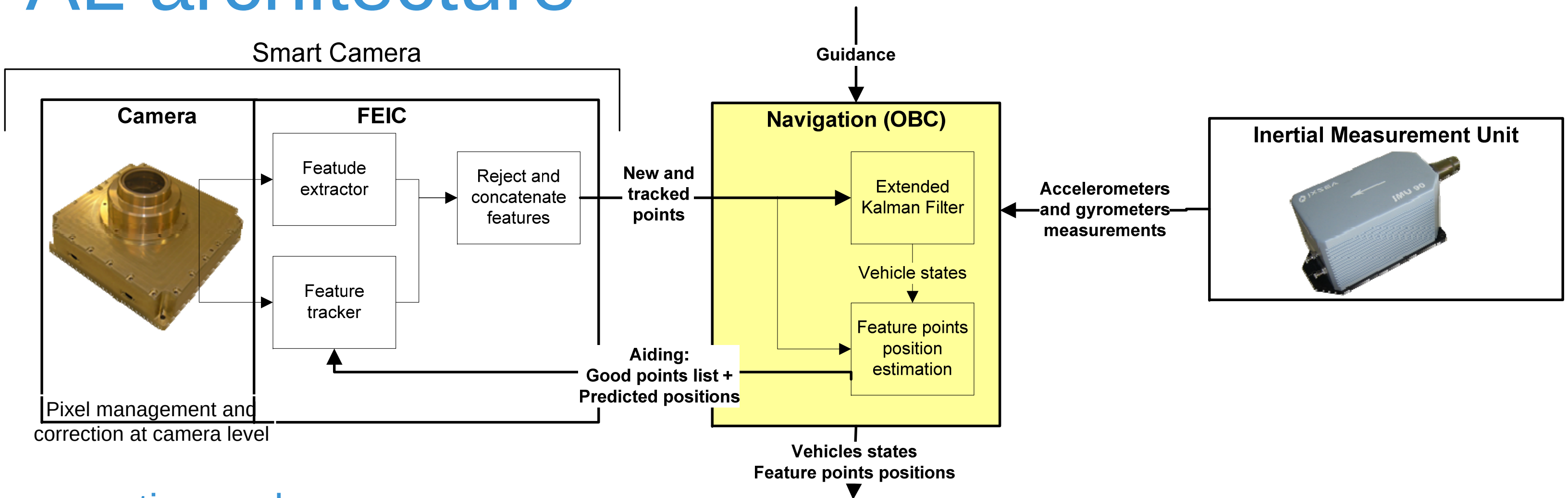

*…On virtual & real scenes (Appolo, Clementine)…*



*…a complete demonstration of the Navigation concept (Pangu virtual scenes)*



# NPAL architecture




- 2 computing nodes:
  - FEIC for heavy algorithms with a high amout of data.

    Communication Camera->FEIC: RAM
  - OBC for navigation algorithms.

    Communication FEIC<-> OBC: Spacewire because of the high frequency aiding loop (10Hz)
- Tight coupling between navigation filter/IMU and the NPAL camera (including IP)
- Advantages
  - Lower mass and consumption since it lowers the communication needs on the bus and the electronics is integrated in the detector
- Drawbacks
  - Specifically designed for landing applications → Adaptation to Rendez Vous / Rover is not straightforward.

# NPAL Industrial Development

- Industrial organisation
  - Navigation filter design and development: Astrium
  - Image Processing (IP) designed by Astrium, implemented by U. of Dundee in the FEIC
  - Camera design and development: Selex Gallileo

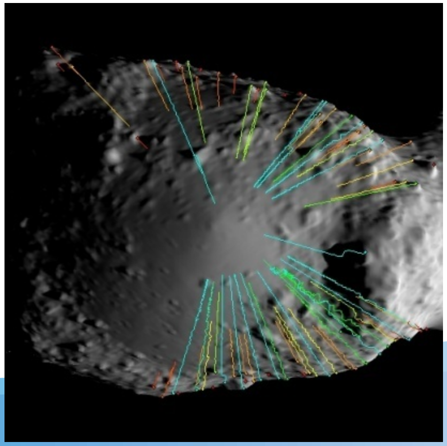
➔ Validation of the performances of the Navigation + Image Processing by Astrium (first with the FEIC algorithms in Matlab then with the FEIC algorithms on a breadboard FPGA, finally on the smart camera).

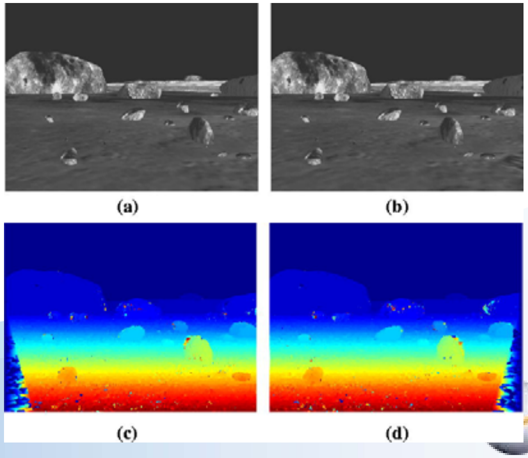
- Requires that the navigation responsible specifies algorithms in the sensor

# VisNav - Towards Industrialisation

- Follow-on of the "Navigation for Planetary Approach and Landing" (NPAL) contract
- Bridging activity enlarging the application fields (not only landing) towards industrialisation for a first mission.
- Scope :
  - VisNav system encompass the complete navigation chain, from Kalman navigation filter to sensors.
- Main goal :
  - Define and Consolidate a Preliminary Architecture in preparation of industrialisation

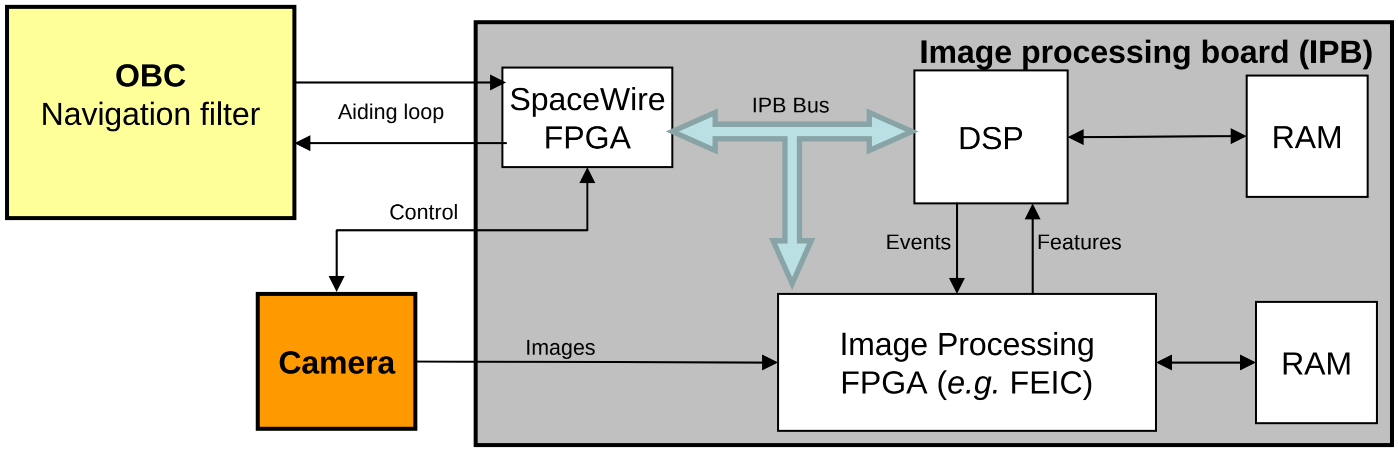
*Landing*
*Built on NPAL heritage*






*Rover*
*Relative navigation*
*Hazard Detection*

# VisNav Architecture




- A generic Camera design dedicated to image acquisition
  - Only simple IP functions are performed by the camera (flux data regulation, simple lossless compression algorithms). No microprocessor is required in the camera and memory is reduced.
  - Communicates with Image Processing Board is made through a high rate SpaceWire link.
- One versatile computing node : the image processing board
  - IP functions are shared between a FPGA (possibly an ASIC for FM) and/or a DSP. Main low level functions are defined. A preliminary IP FPGA design has been performed.
  - Generic IPB architecture enables final sharing and implementation to depend on the actual mission.
- Navigation algorithms to be implemented on the OBC

# VisNav Development

- Industrial organisation
  - Navigation filter design and development: Astrium
  - Image processing and IPB designed by Astrium
  - Camera design : Selex Gallileo
- Advantages
  - “Physical” allocation of the responsibilities (Navigation / IPB / Camera)
  - Flexible architecture compatible with very different applications
  - Better thermal control and environment isolation (separated IPB)
- High amount of data on the specific image data link. Power consumption …. (> 3W for the spacewire)
- The IPB is not integrated in the detector → Extra mass, volume and power

# Conclusions on the architecture

- 2 exemples of architectures that varies according to the location of the computation capacity
- In both case, the IP algorithms are specified and validated by the navigation responsible
  - The IP are specified/defined at the navigation level because the performances of the IP are function of the navigation algorithms and vice-versa (tight coupling)

- Focusing on the VNB, the choice of the architecture should also take into account following discussion that concerns future projects that are going to enter in B,C/D phase
  - How to specify the IP algorithms ? How to validate their performances ? How to validate the navigation overall performances ?

⇨ *Next slides*

# IP Specification. Solution 1: Reference cases definition

- Solution 1: Definition of functional interfaces and reference cases
- Partial solution. More an assessment than a validation. Difficulties to define the worst case scenario because of non-linearities, of coupling and of the high number of cases (images are more complex than scalar parameters in a range !)
- However, interesting to benchmark the solutions, for an early validation and for non regression tests.
- Required standardisation to ease benchmarking. This standard would also define the data format for the future missions to be able to benchmark off-line the algorithms. Reference case definition could be performed at ESA level.
    - Lessons learnt to validate the VBN : it's difficult to have the required data with the correct format and especially a sufficient time synchronisation (or at least time information with the same reference).

# IP Specification. Solution 2: Error models

- Solution 2: Definition of functional interfaces and of error behaviour models (=performances models)

- Requires maturing the models as for the star trackers in the last two decades (FoV error, pixel error, …)

- Remark : the star tracker are now mainly a black box for the S/C thanks to the work performed on the error models that are required to have an overall optimised and validated system whereas the STR algorithms are not mastered by the S/C manufacturer

➔huge work! Feasibility TBC because of non-linearities and of the various parameters (sun elevation, image features varying with the different environments, etc.)

➔Implies certainly to have only simple algorithms inside the smart sensors to have models that are physically understandable.

# VBN Overall Validation Approach

- Reference case for benchmarking, early validation and non regression tests

- Open loop validation
  - Monte Carlo runs for performance and robustness assessment.
  - Image generation off line
  - Photorealistic images are required
    - still some improvements to be done

  *Not required if reliable/understandable error models are available*

- Close loop validation :
  - Requires on-line image generation
  - May be real time with (part of) the sensor in the loop (e.g. validation of star tracker)
  - ➔ no available tool with real time capacity but it seems feasible (see virtual reality industry)

- Representative visual environment is still required to:
  - Validate images models that will be used for open/close loop validation
  - End to end validation on dynamic test bench for phase C/D programs

# Conclusions

- Smart sensors are an optimised solution for aided navigation. They offer high performances solutions with limited consumption/volume/mass ➔ to be privileged for navigation.
- Their architecture will result from technical drivers but also from the industrial organisation/cooperation because the pre-processing (e.g. image processing for VBN) algorithms are specified/validated by the navigation responsible.
- Needs of reference cases for benchmark, non regression tests and early performance assessment. Standardized data would also benefit to the whole community.
- Errors models specification could be developed to ease the smart sensor specifications but the effort is huge.
- Still some efforts to be performed on the image generation, in particular for real time.
- Building blocks are possible at sensor level on basic algorithms and HW. Their justification will mainly depend of the number and consistency of the mission (level of re-use)

# Questions ?