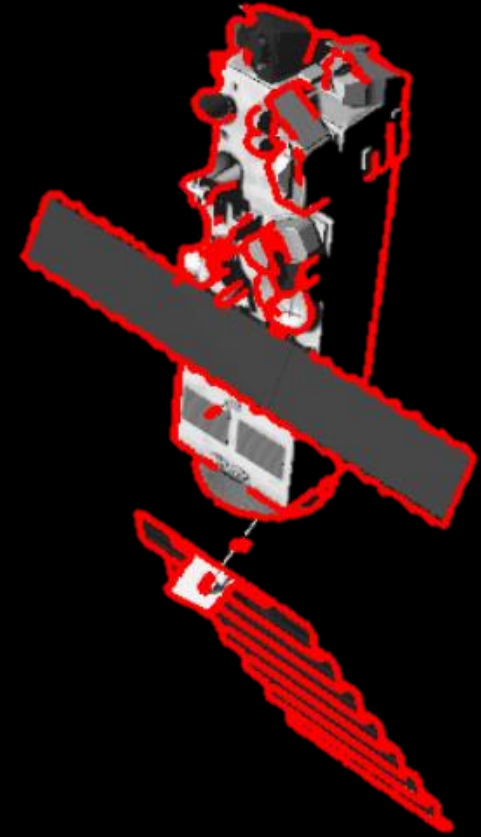


# High Performance Avionics Solution for Advanced and Complex GNC Systems for ADR **HIPNOS**

TEC-ED & TEC-SW Final Presentation Days  
ESA-ESTEC, December 12<sup>th</sup> 2017

David Gonzalez-Arjona (dgarjona@gmv.com)  
Gianluca Furano (gianluca.furano@esa.int)



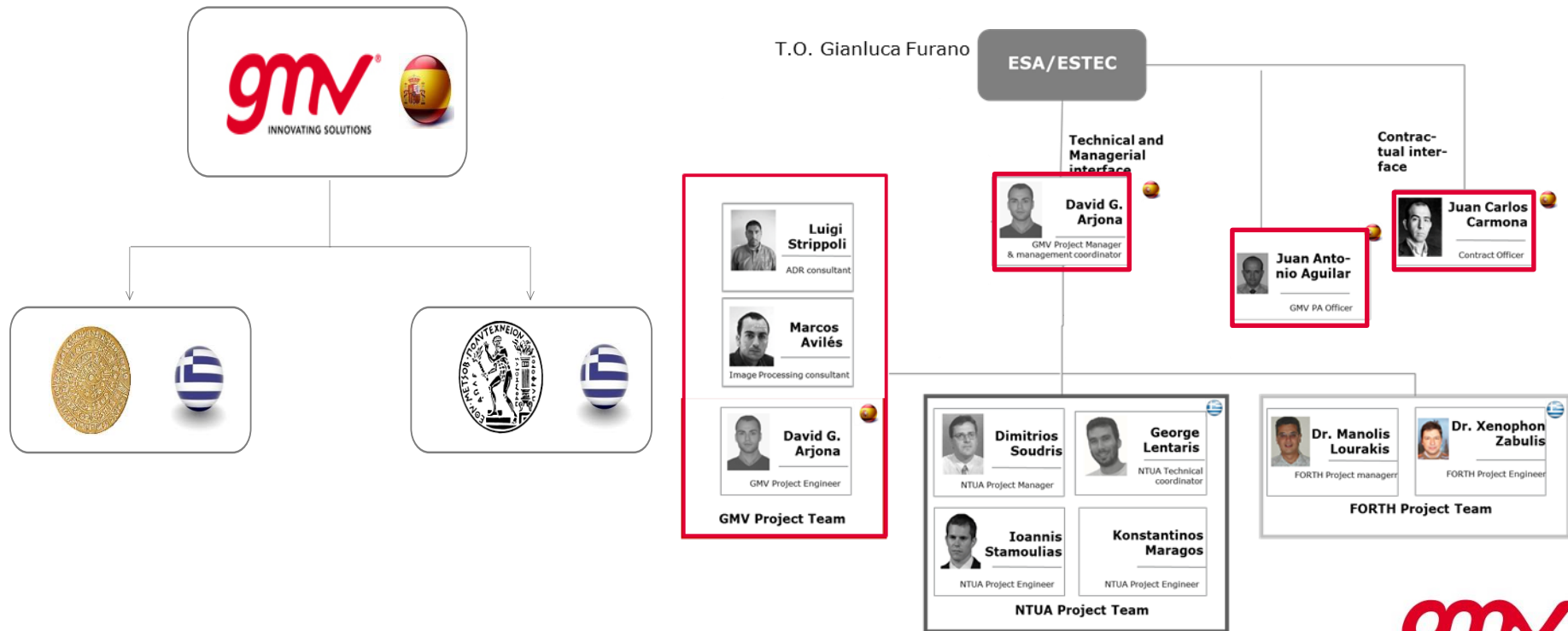
# HIPNOS PROJECT OVERVIEW

# HIPNOS Consortium

**GMV** expertise in ADR, Image Processing and Avionics Architecture for space finds the perfect complement in our collaborators:

**NTUA** who will provide the necessary expertise on FPGA and VHDL modelling, vectorial parallelization of computer-vision algorithms

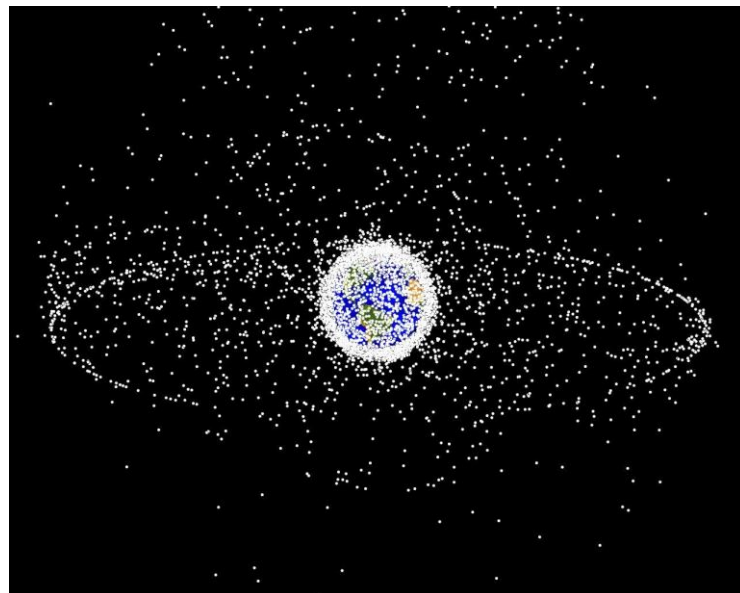
**FORH** who will provide the necessary and expressly required expertise on computer vision and pose estimation algorithms



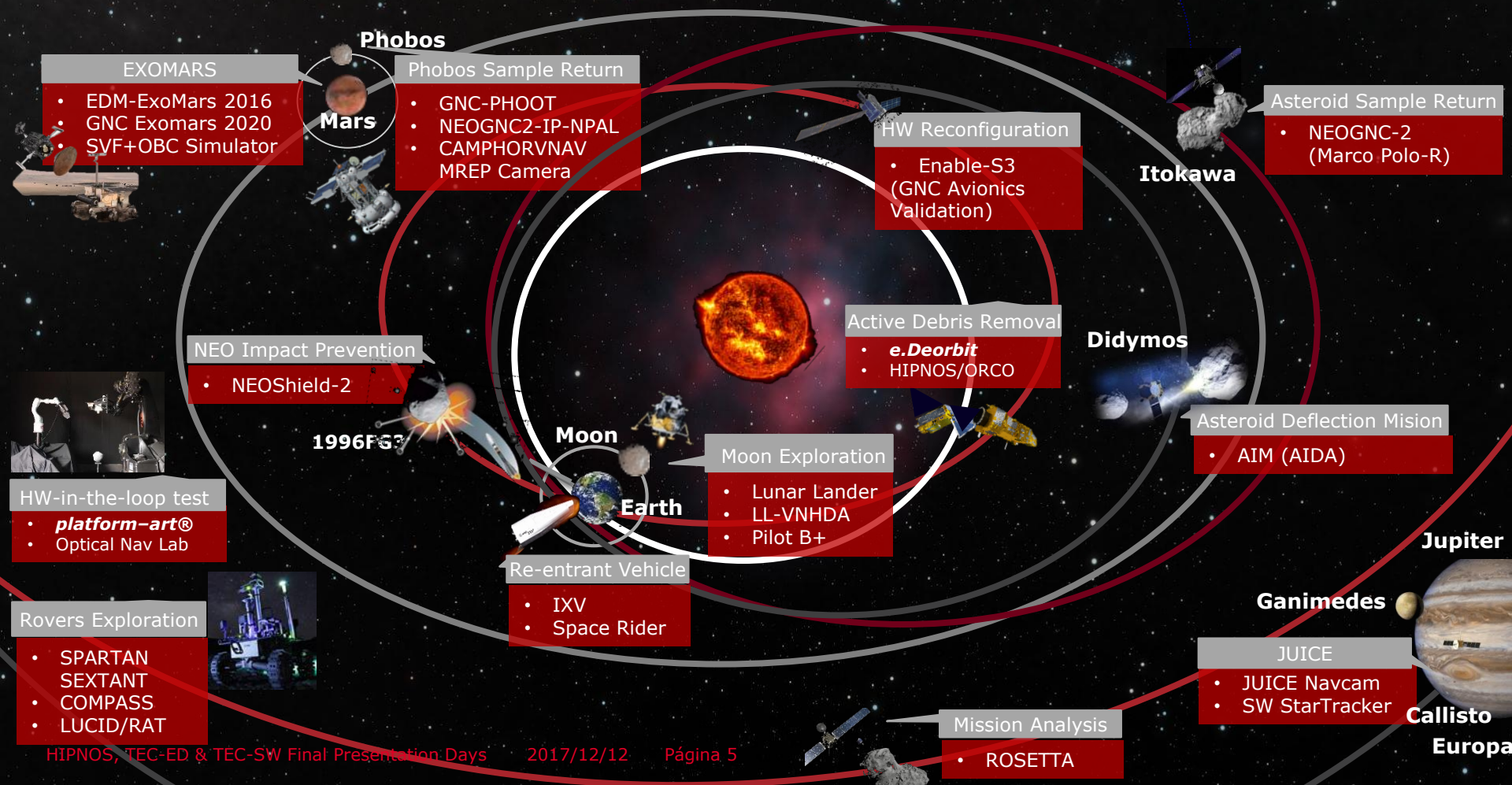
# SCOPE & OBJECTIVES

***Development of a representative HW/SW solution for a high-performance processing platform for Active Debris Removal missions. Implement COTS-based solution as Demonstration of the activity***

- study/define the high-level architecture of a high-performance computing system for space avionics for GNC in ADR missions
- design one high-accuracy & high-complexity chain of Computer Vision algorithms to support the ADR scenario of e.Deorbit
- select the most appropriate acceleration platform in terms of speed, power, rad-hardness, mass/size, flexibility, future trends
- accelerate the CV algorithms on FPGA or GPU, or DSP, or multi-core CPU to achieve the high-speed processing required in ADR
- develop and demonstrate a preliminary, proof-of-concept system (by using COTS components and high-definition videos) with a representative ADR use-case.
- present the feasibility of implementing demanding algorithms with real-time performance on future space-grade platforms



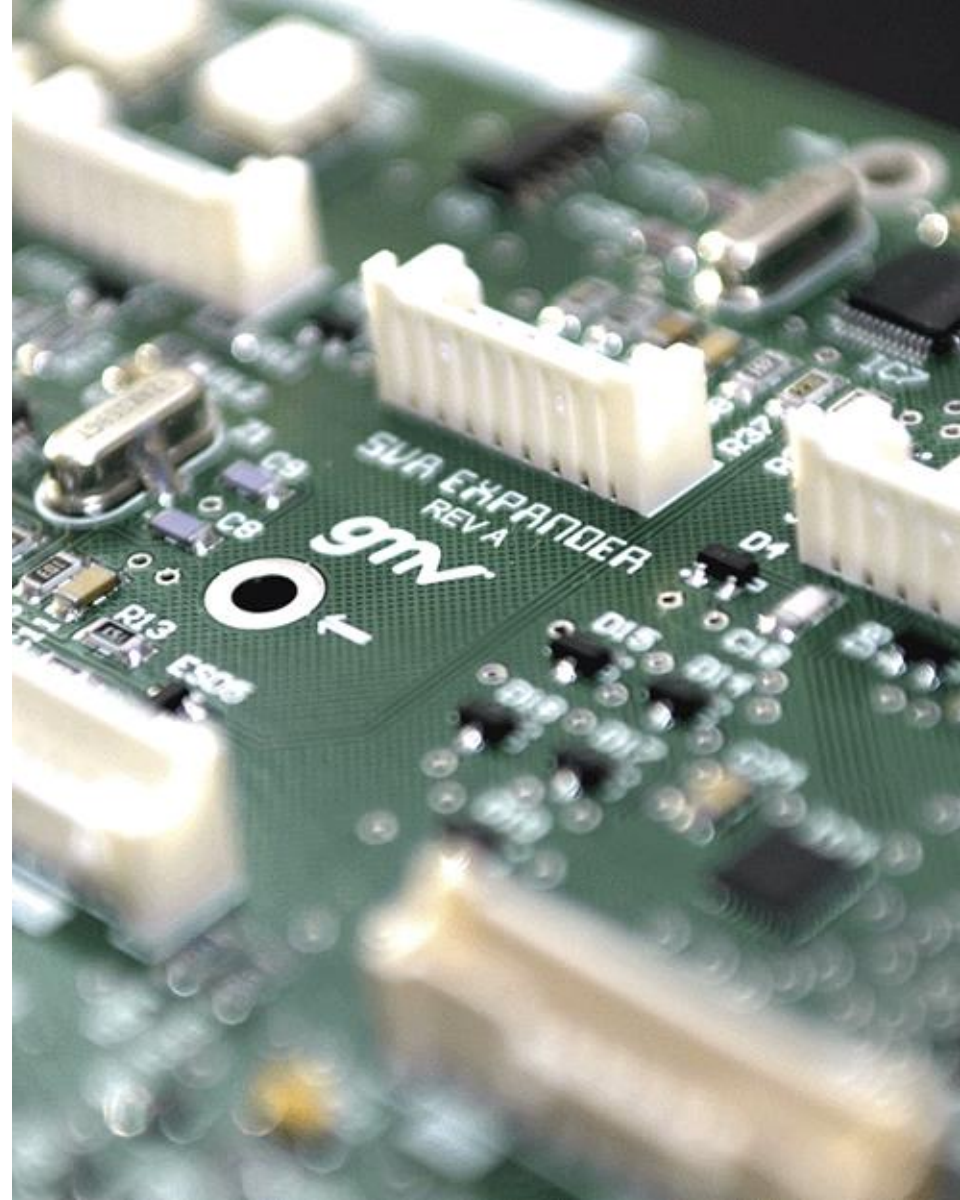
# GMV AUTONOMOUS NAVIGATION IN SPACE





# TECHNOLOGY ENVIRONMENT

- Hardened Devices protected versus:
  - Radiation
  - Solar Pressure
  - Electromagnetic Waves
  - Vibrations (launching environment)
  - Wide temperatura range
- Hard/Impossible to repair devices (up there):
  - Reliability is a must
- **Autonomy** is a key factor
  - Independent Systems
  - Huge delay/latency in Ground-Spacecraft communication**
- Limited power consumption on board
- Mass and volumen shall be minimized
- Design and implementation of **Fault-Tolerance** systems
- Critical, Precise and Deterministic systems in Hard Real-Time applications
- Extensive and intensive Validation and Verification
- Ad-hoc projects for each mission:
  - Nobody went there before → how to create representative environment, images, conditions?



# ADR SCENARIO REQUIREMENTS

## *High-demanding on-board space applications which cannot rely on common space-grade avionics*

- Large ESA owned dead satellite, uncooperative, non-passivated → e.Deorbit → ENVISAT
- RdV: assume a hold point at a distance from the target of 100m and another at 50m
  - Camera-based rendezvous until 100m
  - Forced motion approach or safe orbit approach to 50m
  - Spin synchronization
  - Approach in the target body frame to terminal hold point
  - Capture at 2m: "Capture Phase" shall assume a hold point at a distance of 2 m
- The chaser shall be able to perform relative navigation w.r.t. the target object during the full target orbit anytime of the year, fully autonomous without any ground intervention
- A camera sensor plus a relative trajectory + attitude propagation.  
LED based spotlights can provide illumination for the rest of the approach.
- Trade-offs (performance, power consumption, image resolution, arithmetic precision, mass/size budget, accuracy,...)
- Consider the most representative and computationally demanding computer vision and image processing algorithms (target 10fps 1024x1024 pixel images)



<b>Launch mass:</b>	<b>8,211 kg</b>
<b>Dimensions:</b>	<b>2.5 × 2.5 × 10 m</b>
<b>Orbit:</b>	<b>LEO 2°/s spin</b>

# AVIONICS PROCESSING REQUIREMENTS

Derived requirements for the processing board of HIPNOS based on e.Deorbit MSRD and relevant projects

## ▪ **Electrical Power**

- e.Deorbit MSRD: nothing specific (MIS-59 = mean and peak power TBD)
- similar projects: GMV-NEOGNC2-IP, GMV-CAM-PHOR-VBN
- relevant solutions with FPGA (NASA): 2-5 Watts for Zynq boards, and 5-20 Watts for multi-board SpaceCubes
- **requirement = 10 Watt (TBC)**

## ▪ **Mass/size**

- e.Deorbit MSRD: nothing specific (MIS-60 = mass TBD Kg)
- similar projects: GMV-NEOGNC2-IP, GMV-CAM-PHOR-VBN
- relevant solutions with FPGA (NASA): <100gr for Zynq boards, and 1.4-5.8Kg for multi-board SpaceCubes (+housing)
- relevant solutions with FPGA (NASA): 1U or 17x17x5cm for Zynq boards, 1U up to 13x18x23cm for V5QV (+housing)
- **requirement = 0.5 Kg (TBC)**
- **requirement = 20x20x10 cm<sup>3</sup> (TBC)**

## ▪ **Processing power (projected to CPU)**

- e.Deorbit MSRD: nothing specific
- similar projects: extrapolated IP results from [SEXTANT] for high-rate high-definition images show 100x more power
- **requirement = 15.000 MIPS (TBC)**

## ▪ **Interfaces**

- e.Deorbit MSRD: nothing specific
- **requirement = sensors-board : space-qualified link for 2Mpixel image at 10fps (TBC)**
- **requirement = OBC-board : TBD high-speed bus (data/image) + TBD low-rate bus (control)**

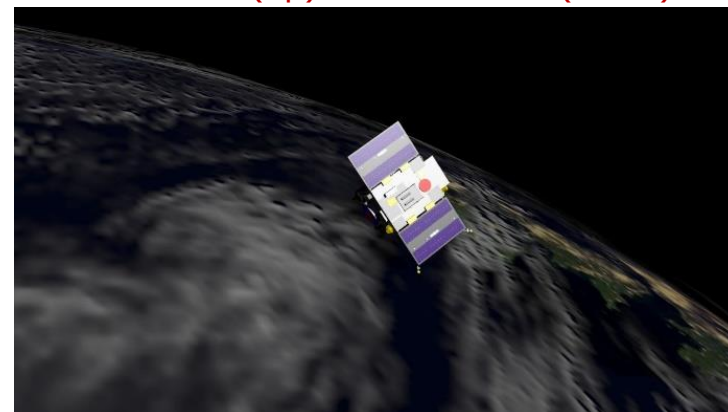


# ADR SCENARIO DEMONSTRATION

- ENVISAT sequences were generated using ASTOS camera simulator
- 2 trajectories for ENVISAT:
  - Observation phase: chaser stands about 50m from target
  - Approaching phase: From 30m down to 10m
- PROBA-2 sequences were generated using commercial rendering SW including the Earth in background
- 1 trajectory for PROBA-2
  - Spin Synchronization maneuvers: Chaser synchronizes rotation with target
  - Later approaches to the target



ENVISAT (up) and PROBA-2 (down)



**HIPNOS**  
**AVIONICS**  
**TRADE-OFF**

# TRADE-OFF AVIONICS

- STATE-OF-THE-ART

- SPACE-GRADE
- COTS

GR712RC Dual-Core SOC Leon3-FT	Space-Qualified	100 MHz	140-200 DMIPS	3-6 Watts
RAD750 PowerPC	Space-Qualified	200 MHz	400 DMIPS	15-20 Watts
Intel Core i5-2500K 4-core (2011)	COTS	3,300 MHz	83,000 DMIPS	73-100 Watts

# TRADE-OFF AVIONICS

- STATE-OF-THE-ART
  - SPACE-GRADE
  - COTS
  
- NEXT-GENERATION AVIONICS
  - SPACE-GRADE
  - COTS

GR712RC Dual-Core SOC Leon3-FT	Space-Qualified	100 MHz	140-200 DMIPS	3-6 Watts
RAD750 PowerPC	Space-Qualified	200 MHz	400 DMIPS	15-20 Watts
Intel Core i5-2500K 4-core (2011)	COTS	3,300 MHz	83,000 DMIPS	73-100 Watts

GR740 Quad-Core SOC Leon4-FT	Space-Qualified	250 MHz	425-1700 DMIPS	2-7 Watts
RAD5545 64-bit Quad-core PowerPC	Space-Qualified	800 MHz	5200 DMIPS	18-24 Watts
P4080 Octo-core board	Latch-up Immune Virtex-5 Voting System	1,500 MHz	27,600 DMIPS	>45 Watts



# TRADE-OFF AVIONICS

- STATE-OF-THE-ART

- SPACE-GRADE
- COTS

- NEXT-GENERATION AVIONICS

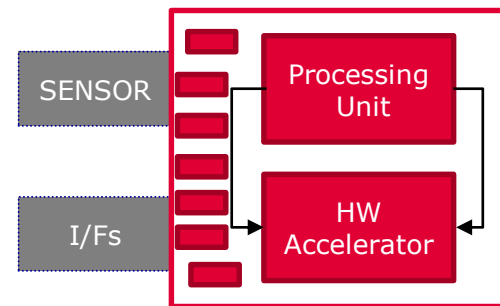
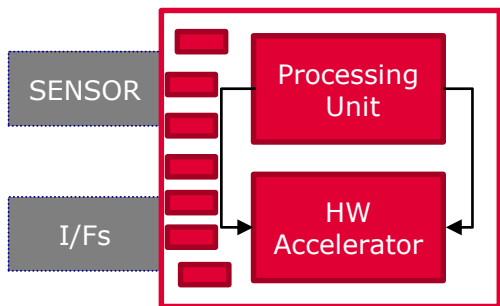
- SPACE-GRADE
- COTS

GR712RC Dual-Core SOC Leon3-FT	Space-Qualified	100 MHz	140-200 DMIPS	3-6 Watts
RAD750 PowerPC	Space-Qualified	200 MHz	400 DMIPS	15-20 Watts
Intel Core i5-2500K 4-core (2011)	COTS	3,300 MHz	83,000 DMIPS	73-100 Watts

- FIRST GENERAL ITERATION:

- CONVENTIONAL SPACE-GRADE PROCESSORS
- SPACE-GRADE SRAM-BASED FPGAs
- DSPS
- GPUS
- MULTICORE-MULTIPROCESSOR
- SYSTEM-ON-CHIP

GR740 Quad-Core SOC Leon4-FT	Space-Qualified	250 MHz	425-1700 DMIPS	2-7 Watts
RAD5545 64-bit Quad-core PowerPC	Space-Qualified	800 MHz	5200 DMIPS	18-24 Watts
P4080 Octo-core board	Latch-up Immune Virtex-5 Voting System	1,500 MHz	27,600 DMIPS	>45 Watts



# TRADE-OFF AVIONICS

- STATE-OF-THE-ART

- SPACE-GRADE
- COTS

- NEXT-GENERATION AVIONICS

- SPACE-GRADE
- COTS

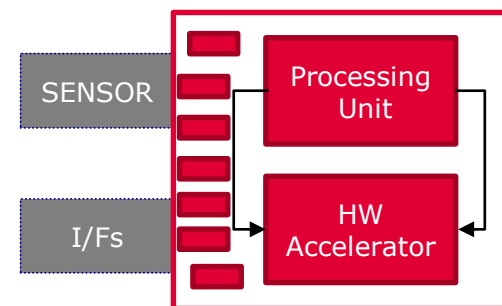
GR712RC Dual-Core SOC Leon3-FT	Space-Qualified	100 MHz	140-200 DMIPS	3-6 Watts
RAD750 PowerPC	Space-Qualified	200 MHz	400 DMIPS	15-20 Watts
Intel Core i5-2500K 4-core (2011)	COTS	3,300 MHz	83,000 DMIPS	73-100 Watts

- FIRST GENERAL ITERATION:

- CONVENTIONAL SPACE-GRADE PROCESSORS
- SPACE-GRADE SRAM-BASED FPGAs
- DSPs
- GPUS
- MULTICORE-MULTIPROCESSOR
- SYSTEM-ON-CHIP

GR740 Quad-Core SOC Leon4-FT	Space-Qualified	250 MHz	425-1700 DMIPS	2-7 Watts
RAD5545 64-bit Quad-core PowerPC	Space-Qualified	800 MHz	5200 DMIPS	18-24 Watts
P4080 Octo-core board	Latch-up Immune Virtex-5 Voting System	1,500 MHz	27,600 DMIPS	>45 Watts

TCLS ARM4SPACE	DAHLIA quad-core ARM cortex-R52	MPSoc Multicore ARM + FPGA logic Zynq Ultrascale+	ARM Cortex M0+ COTS ARM Cortex-A Snapdragon 820	LEON-5
HPDP manycore	MPPB/SSDP multi-dsp board	RC64 CEVA DSP manycore 500x faster GR712	TI SMV320C6727B C67x + VLIW DSP	ARM (4-core) + TMS320C66x (8-core C66x DSP)
Virtex5-QV	RTG4	RTAX, ProAsic3, CTOS SmartFusion	BRAVE: NG-MEDIUM NG-LARGE (+ARM) NG-ULTRA (+ARM)	Zynq7000 SoC Zynq UltraScale+
Myriad Movidius	LPGPU rad-tolerant	NVIDIA GPU	SYSTEM ON CHIP	





# BENCHMARKING

- done extensive testing/evaluation
- with both in-house & literature work
- for all processor categories
- for computer vision tasks
- focused mainly (but not only) on **performance** and **Watt**
- **in total, more than 30 platforms and 10 benchmarks**

	In-house development & testing	Literature survey	
platforms	FPGA	Xilinx Virtex6 VLX240T-2, Zynq7000 (Z7020, Z7045)	Altera Stratix III E260, Xilinx Virtex 5QV, Virtex4 VLX100, Virtex6 VSX475T, Zynq7000
	CPU	desktop Intel i5-4590, laptop i3-4010U embedded: ARM CortexA9, Intel X1000 space: LEON3, OCE E698PM (LEON4)	desktop: Intel i7-3820, AMD FX-8120 embedded ARM CortexA15 space-grade BAE RAD5545
	GPU	Nvidia GeForce GTX 670, GTX 680, GTX 960	Nvidia GTX 980Ti / 650Ti / 295, Tesla C2050, mobile: Nvidia Tegra K1/X1, ARM Mali-T760
	DSP	space-grade Xentium MPPB embedded multi-core Myriad2	TI multi-core TMS320C6678 and 66AK2H14, 1-core C674x, 64-core RH RC64 (MACSPACE)
benchmarks	2D convolutions (5x5 to 11x11), Harris-corner & Canny-edge detectors, Stereo Matching, Hyperspectral search, Pose Estimation (incl. feature detection, description, matching), Super-Resolution	2D convolutions and SAD (up to 25x25), Harris-corner & Canny-edge detectors, Stereo Matching, Image Denoising and Block Matching, Hyperspectral imaging, <i>etc.</i> (plus nominal DMIPS and MFLOPS figures)	

- clouds of results, vary per platform & benchmark (peculiarities of computational model, chip node/size, etc.)
- challenge tackled methodically, comparison converged in big consistent picture

# COMPARISONS

## 1<sup>st</sup> ITERATION (BIG PICTURE)

- CPUs worst performance/Watt
- FPGAs best perf/Watt (10x)
- FPGA vs desktop-GPU, not far w.r.t. speed, but w.r.t. power..
- mobile-GPU vs desk-GPU, trade 10x performance for Watt
- mob-GPU vs many-DSP, similar performance and power

	rad-hard CPU (1-core)	embedded CPU (1-core)	desktop CPU (1-core)	mobile GPU	high-end DSP	FPGA	desktop GPU
Throughput	0.2–1.7	0.5–2	20–100	50–150	50–240	300–1460	200–2000
Power (W)	1–18	1–2	20–90	6–10	1–10	2–10	70–195
Perf/Watt	0.1–0.6	0.25–2	0.5–1	8–15	12–50	60–250	5–25

# COMPARISONS

## 1<sup>st</sup> ITERATION (BIG PICTURE)

- CPUs worst performance/Watt
- FPGAs best perf/Watt (10x)
- FPGA vs desktop-GPU, not far w.r.t. speed, but w.r.t. power..
- mobile-GPU vs desk-GPU, trade 10x performance for Watt
- mob-GPU vs many-DSP, similar performance and power

	rad-hard CPU (1-core)	embedded CPU (1-core)	desktop CPU (1-core)	mobile GPU	high-end DSP	FPGA	desktop GPU
Throughput	0.2–1.7	0.5–2	20–100	50–150	50–240	300–1460	200–2000
Power (W)	1–18	1–2	20–90	6–10	1–10	2–10	70–195
Perf/Watt	0.1–0.6	0.25–2	0.5–1	8–15	12–50	60–250	5–25

## 2<sup>nd</sup> ITERATION (BEST 28nm COTS)

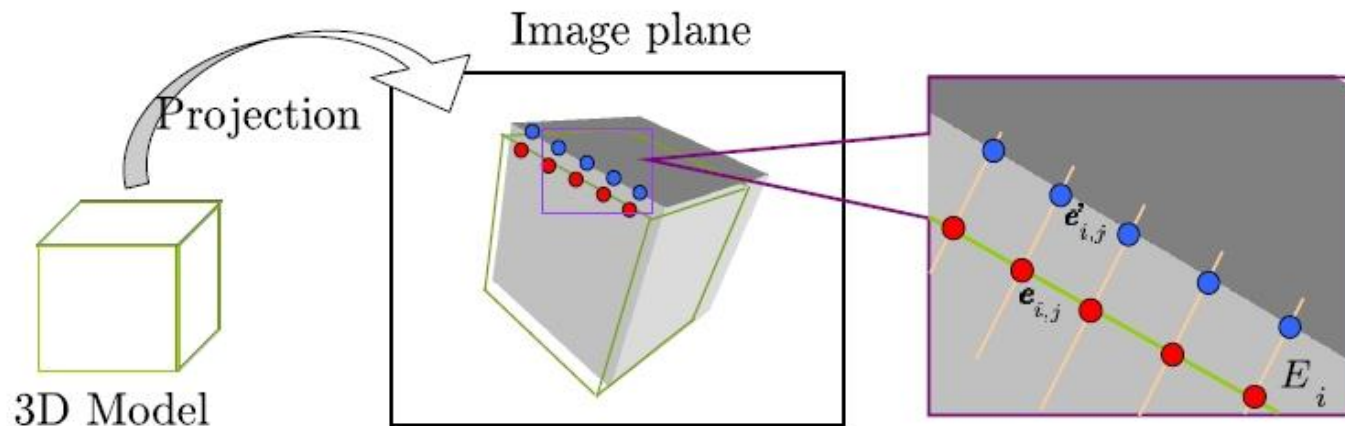
- SoC are most useful for acceleration
- DSPs are closing in (vs FPGA), but
  - Zynq 10x perf vs Myriad2 for HD
  - Zynq 10x perf/W vs TI-C66x
- Myriad2 lowest power (1W)
- Zynq highest speed (with slow clk)

	4-core LEON4 (E698PM) 600 MHz	12-core VLIW (Myriad2) 600 MHz	8-core DSP (66AK2H14) 1200 MHz	FPGA (Zynq7045) 200–300 MHz	desktop GPU (GTX 670/680/960) ≥ 1 GHz
Performance	< 20	50	70–240	430–1460	600–1800
Power (W)	≈ 2.8	1	≈ 10	4–6	> 70
Perf/Watt	< 7	50	7 – 24	110–240	< 25

➤ **selected**: Zynq7000 on MMP board (6x10cm<sup>2</sup> 65gr), also due to connectivity & rad mitigation opportunities

# HIPNOS DEVELOPMENT

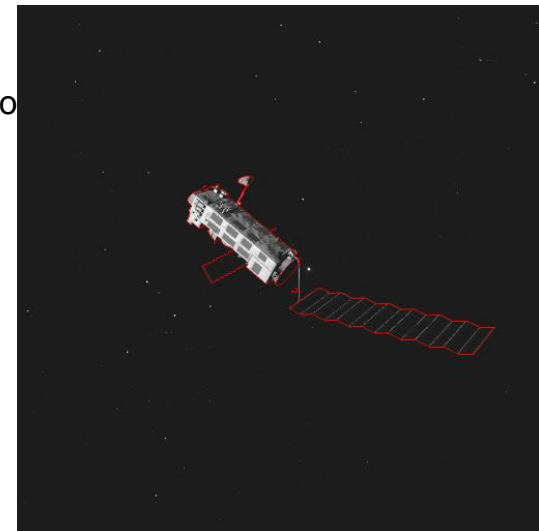
# POSE ESTIMATION PRINCIPLE



- Edges are suitable for tracking weakly textured objects as they
  - are robust to noise and illumination/viewpoint changes
  - can be accurately and rapidly localized in images
- RAPID (Real-time Attitude and Position Determination) is the archetypal edge-based tracker
- A RAPID-like tracking algorithm was developed in HIPNOS
- The developed algorithm imposes no constraints on the employed 3D model

# POSE ESTIMATION ALGORITHM

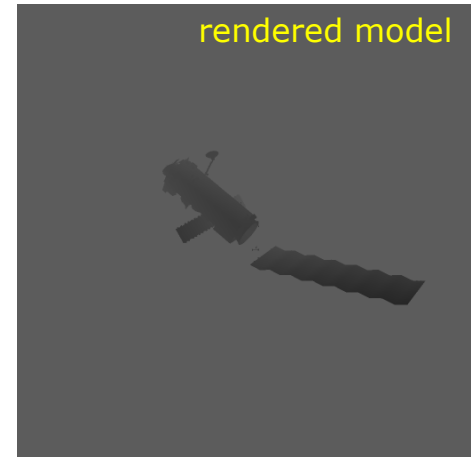
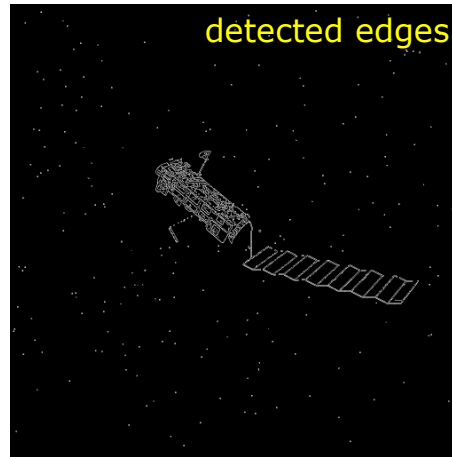
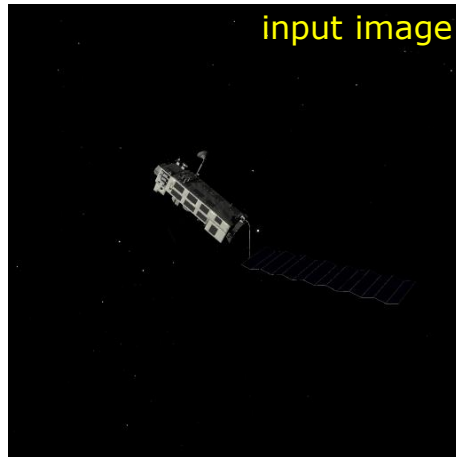
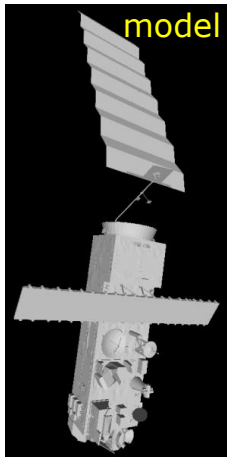
- A model-based, monocular 3D tracking algorithm based on edges
- **Input:** sequence of images + approximate initial 3D pose and object mesh model
- **Algorithm:**
  1. Intensity edges are detected in an input image
  2. The model is rendered (i.e. projected) with the approximate 3D pose; visible depth edges are detected in the projection
  3. Depth edges are matched with intensity edges, searching in directions parallel to the depth gradients
  4. Perpendicular matches are used for estimating the incremental (i.e., frame to frame) pose; outliers are filtered out in a two-level robust regression framework (LMedS + M-estimation)
  5. Incremental pose is integrated with the approximate pose to yield a new pose estimate
  6. The process repeats with a new frame and the pose estimate as the approximate pose
- “frozen” version: Depth edges found with Canny, latter work has led to a more robust scheme



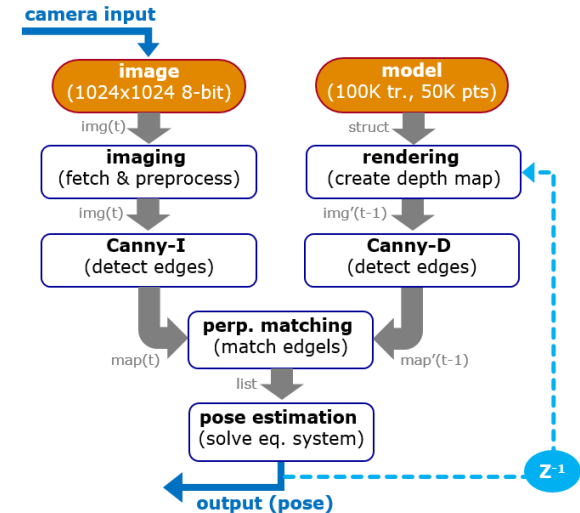
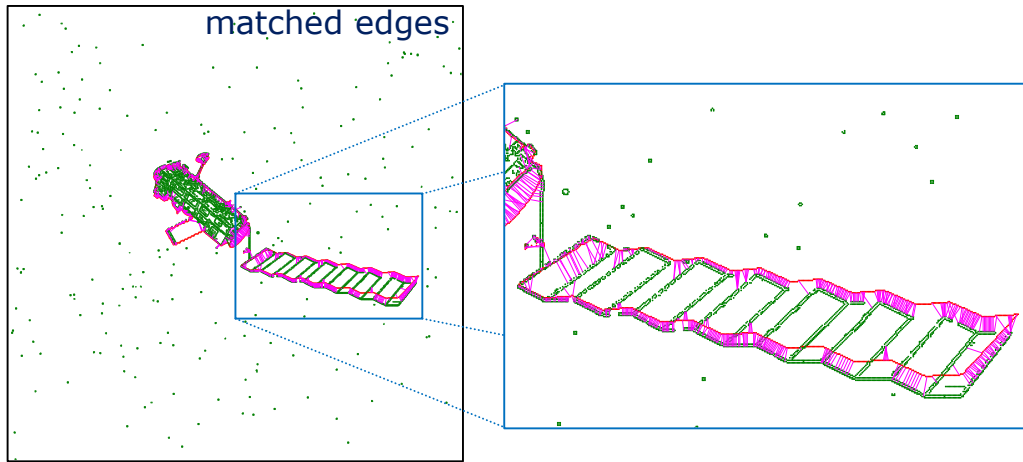


# POSE ESTIMATION DEVELOPMENT 1/2

- **Developed** a model-based, monocular 3D tracking algorithm based on edges (edges are robust to noise and illumination changes; can be accurately and rapidly localized in images)
- Inputs object mesh model (+ approximate initial 3D pose) and the sequence of images
- **Algorithm:**
  1. Intensity **edges are detected** on the input image (with Canny)
  2. The **model is rendered** (i.e. projected) with the approximate 3D pose (of the previous state); visible depth **edges are also detected** in the projection (rendered model)



# POSE ESTIMATION DEVELOPMENT 2/2



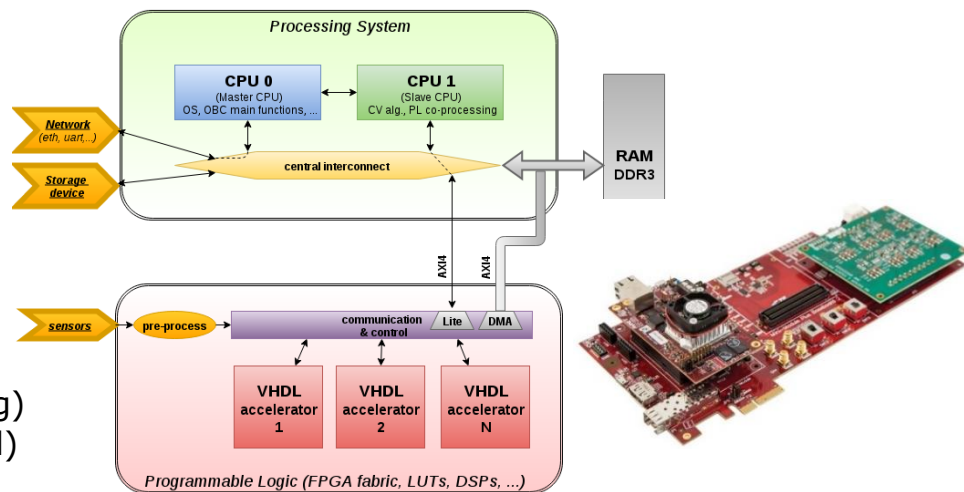
## • Algorithm (cont'd):

3. Depth edges are matched with intensity edges, searching in directions perpendicular to the depth edges
  4. Perpendicular matches are used for estimating the incremental (i.e., frame to frame) pose; outliers are filtered out in a two-level robust regression framework (LMedS + M-estimation)
- ❖ Incremental pose is integrated with the approximate pose (previous state) to yield a new pose estimate (current state) and the process repeats with a new frame

# DESIGN OVERVIEW

## PLATFORM

- System-on-Chip Zynq (MMP module + baseboard)
  - 2x CPU : one dedicated to the CV task
  - FPGA : to accelerate selected CV functions
  - comm. : over AXI4, with DMA for ~3Gbps
- deploy Ubuntu OS with Xillybus (for fast prototyping)
- *sdcard* for pre-stored images (imaging not analyzed)



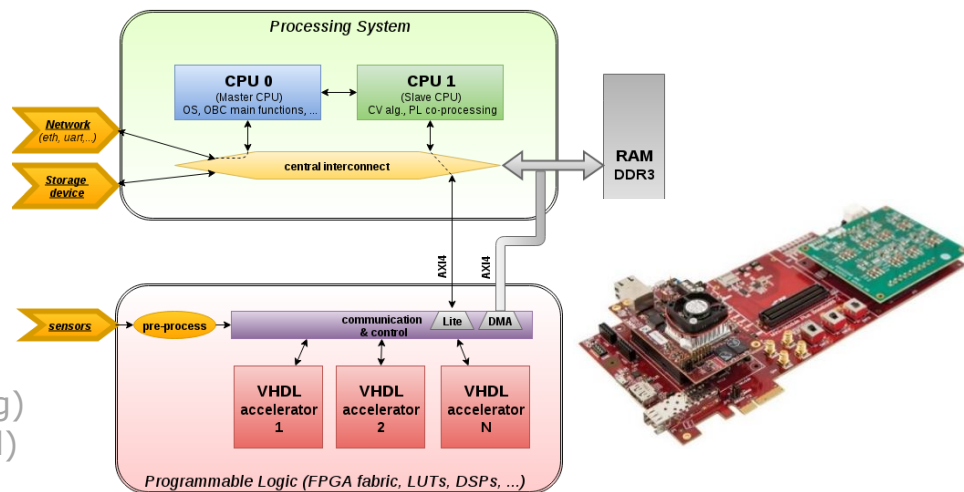
# DESIGN OVERVIEW

## PLATFORM

- System-on-Chip Zynq (MMP module + baseboard)
  - 2x CPU : one dedicated to the CV task
  - FPGA : to accelerate selected CV functions
  - comm. : over AXI4, with DMA for ~3Gbps
- deploy Ubuntu OS with Xillybus (for fast prototyping)
- *sdcard* for pre-stored images (imaging not analyzed)

## PROFILING

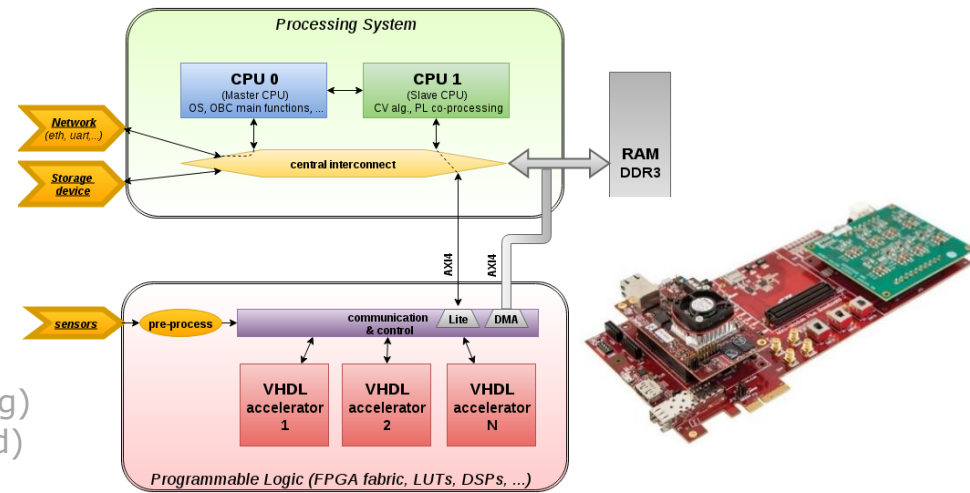
- on ARM Cortex-A9 @ 667MHZ (C/C++, single-threaded)
  - time/frame = 1–1.8 sec (~ 1/2 FPS, depends on distance)
  - 90% for pixel-based processing (Rendering, Canny)
- for few functions, achieved ~3x via NEON SIMD, but goal=10–50x



# DESIGN OVERVIEW

## PLATFORM

- System-on-Chip Zynq (MMP module + baseboard)
  - 2x CPU : one dedicated to the CV task
  - FPGA : to accelerate selected CV functions
  - comm. : over AXI4, with DMA for ~3Gbps
- deploy Ubuntu OS with Xillybus (for fast prototyping)
- *sdcard* for pre-stored images (imaging not analyzed)

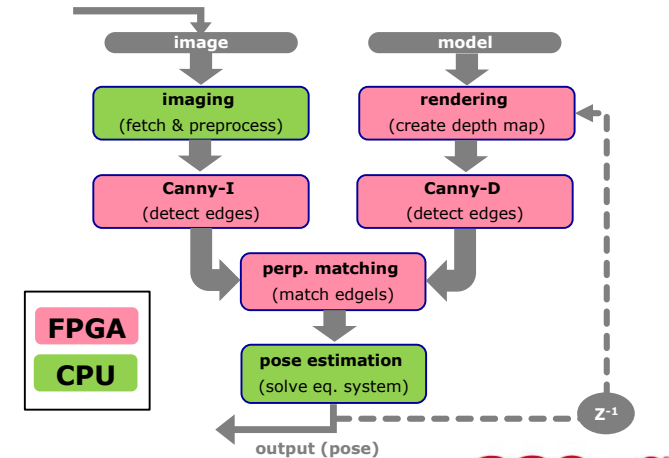


## PROFILING

- on ARM Cortex-A9 @ 667MHZ (C/C++, single-threaded)
  - time/frame = 1–1.8 sec ( $\sim 1/2$  FPS, depends on distance)
  - 90% for pixel-based processing (Rendering, Canny)
- for few functions, achieved  $\sim 3x$  via NEON SIMD, but goal=10–50x

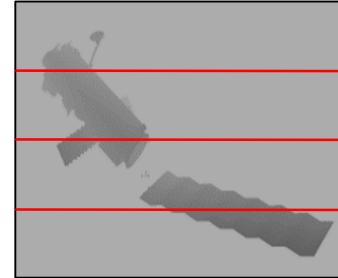
## HW/SW PARTITIONING

- base on methodology, consider multiple requirements per function (time, memory, arithmetic, SW complexity, communication,...)
- roughly: pixel/edge processing on FPGA, algebra equations on CPU
- 94–97% of computation accelerated on FPGA



# HW ARCHITECTURE, VHDL

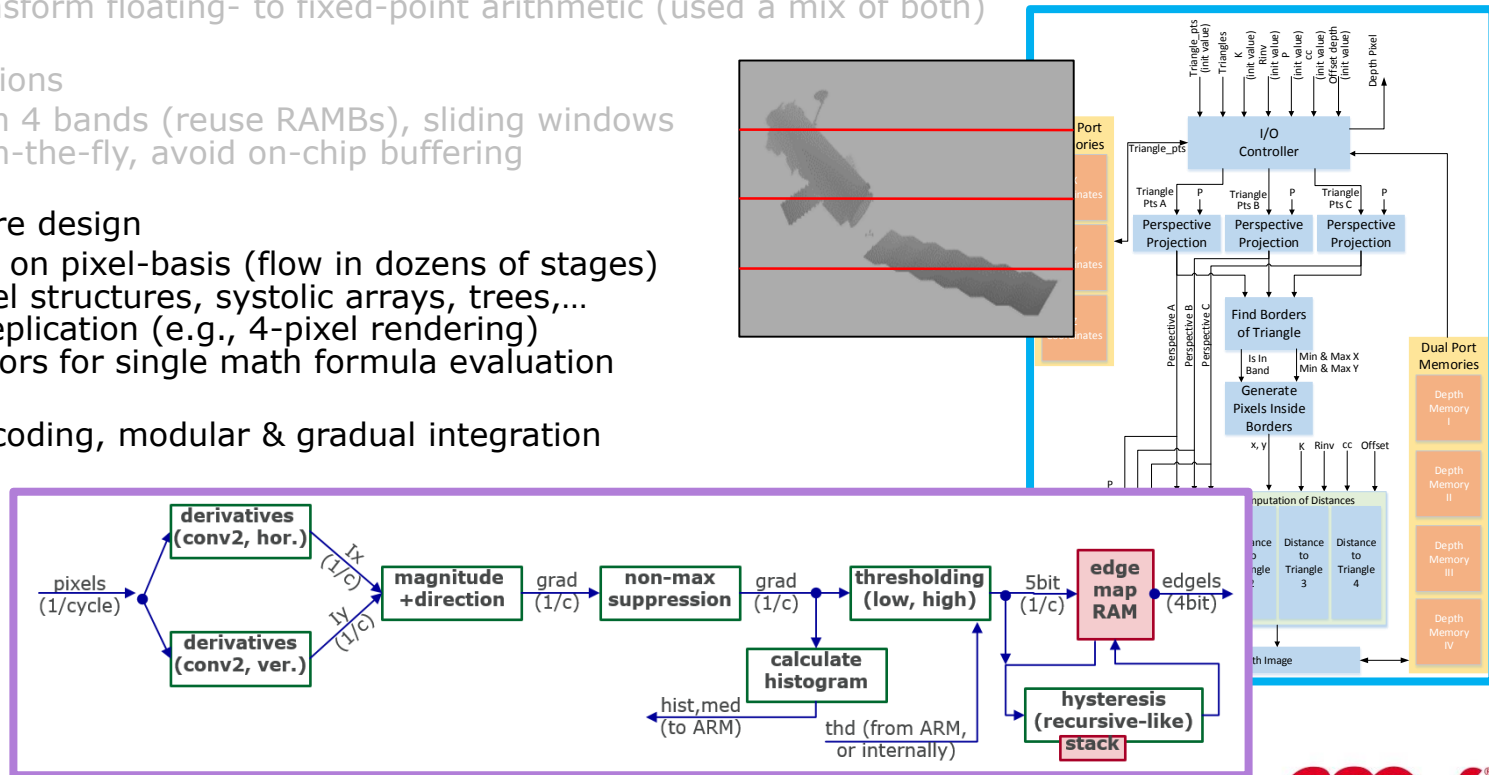
- parameter tuning and customization to given problem
  - tests to fix edge thresholds, number of control points, word-lengths, etc...
  - analysis to transform floating- to fixed-point arithmetic (used a mix of both)
- memory optimizations
  - render image in 4 bands (reuse RAMBs), sliding windows
  - process data on-the-fly, avoid on-chip buffering





# HW ARCHITECTURE, VHDL

- parameter tuning and customization to given problem
  - tests to fix edge thresholds, number of control points, word-lengths, etc...
  - analysis to transform floating- to fixed-point arithmetic (used a mix of both)
- memory optimizations
  - render image in 4 bands (reuse RAMBs), sliding windows
  - process data on-the-fly, avoid on-chip buffering
- parallel architecture design
  - deep pipelining on pixel-basis (flow in dozens of stages)
  - serial-to-parallel structures, systolic arrays, trees,...
  - function/unit replication (e.g., 4-pixel rendering)
  - multiple operators for single math formula evaluation
- parametric VHDL coding, modular & gradual integration



# RESULTS

## RESOURCES

- tested on biggest Zynq7000 FPGA (xc7z100-2 of MMP)
  - 36% LUTs, 48% DSPs, 77% RAMBs,  $F_{max} > 200\text{MHz}$ 
    - most demanding is Renderer (94% logic of design)
  - power  $\approx 4.5\text{W}$  (peak 9W) (CPU@667MHz, PS@200MHz)
- rough estimations for other FPGA devices
  - xc7z045/xc7z030 (smaller): maybe feasible, requires much optimization, tolerable penalty in time/accuracy
  - zu19eg (big upcoming RT): easy fit, utilization  $< 30\%$
  - ng-large (EUR):  $\sim 20\%$  more challenging than xc7z030

	LUT*	DFF	DSP	RAMB36
<i>Canny**</i>	2948	3174	4	346,5
<i>Matching</i>	298	389	-	5,5
<i>Renderer**</i>	93383	148071	966	224
<i>Xillybus+Misc</i>	2895	3777	-	6
<b>TOTAL</b>	<b>99524</b> (36%)	<b>155411</b> (28%)	<b>970</b> (48%)	<b>582</b> (77%)

\*Zynq xc7z100-2L (277K LUTs, 554K DFFs, 2020 DSPs, 755 RAMB36)

\*\*Image 1024x1024 16-bit (Canny: 2 maps) (Renderer: 4 stripes)

# IMPLEMENTATION RESULTS

## RESOURCES

- tested on biggest Zynq7000 FPGA (xc7z100-2 of MMP)
  - 36% LUTs, 48% DSPs, 77% RAMBs,  $F_{max} > 200\text{MHz}$ 
    - most demanding is Renderer (94% logic of design)
  - power  $\approx 4.5\text{W}$  (peak 9W) (CPU@667MHz, PS@200MHz)
- rough estimations for other FPGA devices
  - xc7z045/xc7z030 (smaller): maybe feasible, requires much optimization, tolerable penalty in time/accuracy
  - zu19eg (big upcoming RT): easy fit, utilization  $< 30\%$
  - ng-large (EUR):  $\sim 20\%$  more challenging than xc7z030

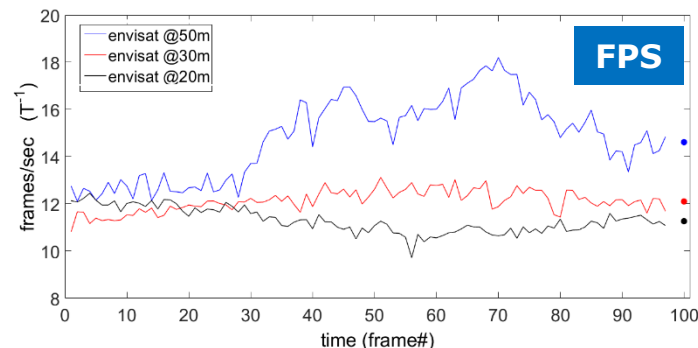
## SPEED

- Time per HW kernel = 5–11ms (plus 55ms for SW function)
- Acceleration (vs ARM) = 19x (system, up to 62x for kernels)
- FPS = 12 on avg. (10–16 depending on ENVISAT distance)

	LUT*	DFF	DSP	RAMB36
Canny**	2948	3174	4	346,5
Matching	298	389	-	5,5
Renderer**	93383	148071	966	224
Xillybus+Misc	2895	3777	-	6
<b>TOTAL</b>	<b>99524</b> (36%)	<b>155411</b> (28%)	<b>970</b> (48%)	<b>582</b> (77%)

\*Zynq xc7z100-2L (277K LUTs, 554K DFFs, 2020 DSPs, 755 RAMB36)

\*\*Image 1024x1024 16-bit (Canny: 2 maps) (Renderer: 4 stripes)



# RESULTS

## RESOURCES

- tested on biggest Zynq7000 FPGA (xc7z100-2 of MMP)
  - 36% LUTs, 48% DSPs, 77% RAMBs,  $F_{max} > 200\text{MHz}$ 
    - most demanding is Renderer (94% logic of design)
  - power  $\approx 4.5\text{W}$  (peak 9W) (CPU@667MHz, PS@200MHz)
- rough estimations for other FPGA devices
  - xc7z045/xc7z030 (smaller): maybe feasible, requires much optimization, tolerable penalty in time/accuracy
  - zu19eg (big upcoming RT): easy fit, utilization  $< 30\%$
  - ng-large (EUR):  $\sim 20\%$  more challenging than xc7z030

## SPEED

- Time per HW kernel = 5–11ms (plus 55ms for SW function)
- Acceleration (vs ARM) = 19x (system, up to 62x for kernels)
- FPS = 12 on avg. (10–16 depending on ENVISAT distance)

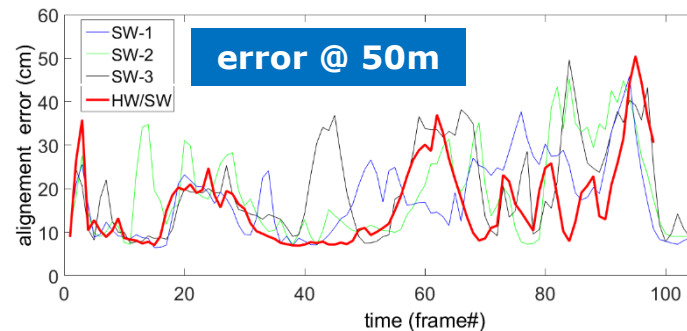
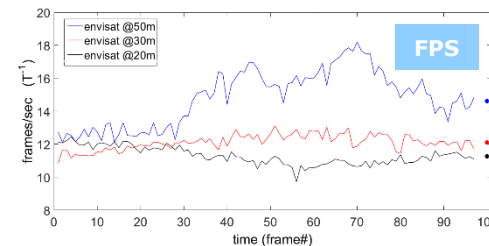
## ACCURACY

- analyzed for 100's frames at 50m, 30m, and 20m
- in general, alignment error  $< 1\%$  (as good as in SW)
- few bad cases (7%, lost track), corrected in new alg.

	LUT*	DFF	DSP	RAMB36
Canny**	2948	3174	4	346,5
Matching	298	389	-	5,5
Renderer**	93383	148071	966	224
Xillybus+Misc	2895	3777	-	6
<b>TOTAL</b>	<b>99524</b> (36%)	<b>155411</b> (28%)	<b>970</b> (48%)	<b>582</b> (77%)

\*Zynq xc7z100-2L (277K LUTs, 554K DFFs, 2020 DSPs, 755 RAMB36)

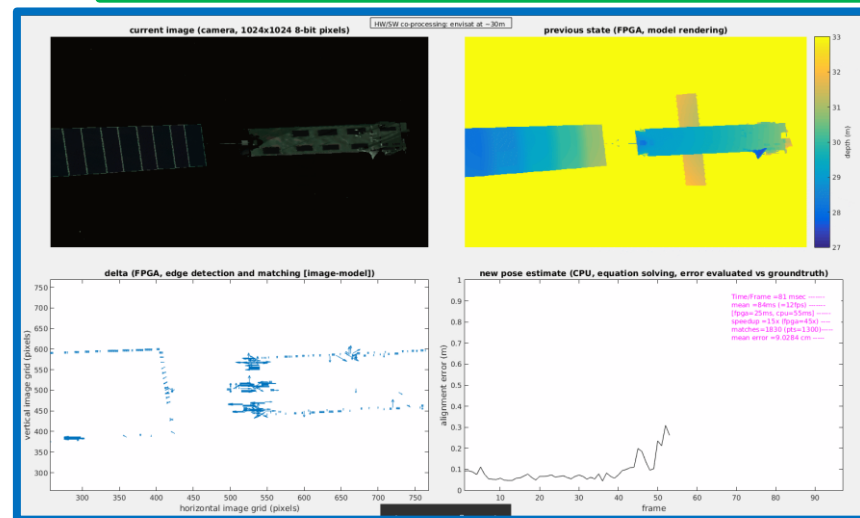
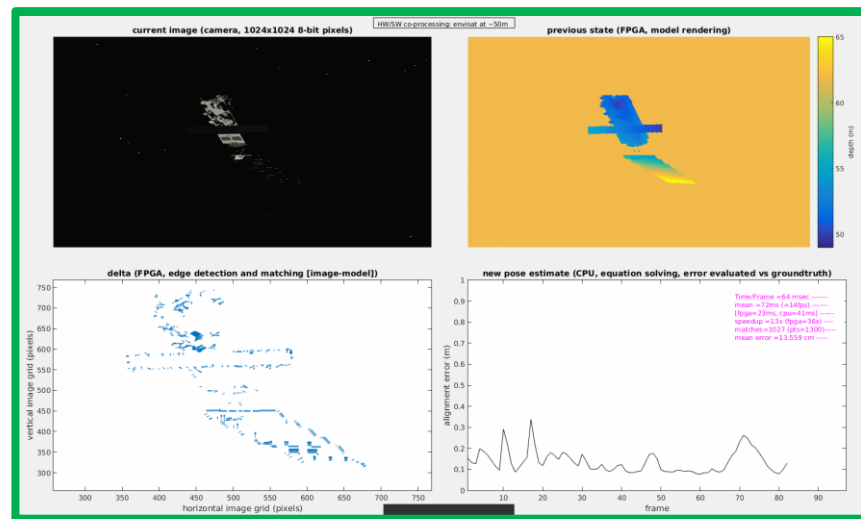
\*\*Image 1024x1024 16-bit (Canny: 2 maps) (Renderer: 4 stripes)



# HIPNOS DEVELOPMENT CONCLUSIONS

# DEMO (RECORDED)

- **video**, with off-line processing (huge debug info, ~1GB)
  - info recorded in files, analyzed/shown in MATLAB
  - not done real-time (much slower than 12 FPS claim)
- 3 datasets (@50m, @30m, @20m), 100 frames each
- **shown**: 4 windows + running statistics (per frame)
  - 1) captured image (before contrast enhancement)
  - 2) rendered model (depth values from FPGA)
  - 3) matches between images (returned from FPGA)
    - ~1300 vectors: start-end to detected edgels
    - some outliers, most agree with object motion
  - 4) error plot (drawn gradually)
    - statistics: time analysis (recorded timestamps)
      - use *gettime*, add FPGA and PoseEst. time
      - omit imaging, error calculation, debug info, and OS spikes (e.g., 3x 1-sec, due to sdcard)
      - compare to all-SW *gettime* results (stored)
        - "speedup" refers to entire HW/SW system
        - "fpga" speedup is for VHDL kernels, alone
    - statistics: matches + error
      - "matches" refers to FPGA results (varies)
      - "pts" refers to limit (actually used for PE)





# DEMO (LIVE)

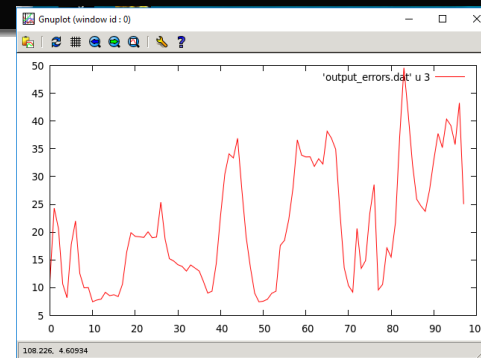
- **console**, with results on-the-fly (minimal information to user)
    - connect from PC to MMP via *ssh* (point-to-point Ethernet)
    - run script "pedemo", get 6-10 FPS (due to imaging overhead)
  - 3 datasets (@50m, @30m, @20m), ~1500 frames total
  - **shown**: console with running results (per frame)
    - start/end of main functions
    - 6D pose estimates
    - error (calculated vs groundtruth)
    - time (cumulative, not filtered/analyzed)
  - **shown optionally**: summary of results (upon end)
    - call *gnuplot* to show "error vs time"
    - open files of recorded timestamps
  - **not shown**: graphics at run-time (as in MRR)
    - cannot plot edges (now, hidden in FPGA)
    - could plot stuff (e.g., depth map), but very slowly (~1 FPS)
- tip: first run all-SW (w/ and w/o graphics), feel the acceleration

```

File Edit View Search Terminal Help
Estimated motion: 0.404926 -0.8
@@@ 0.438518 0.162921 -0.33075
GT motion: -0.751855 1.495478 3
Frame 1015: positional error 0.
Elapsed time: 0.17 seconds, 165

...simpler matching on SW...
estimating motion with 1300 points
Estimated motion: 0.408969 -0.821995 -2.026649 -0.097275 -0.159226 20.942781
@@@ 0.442460 0.164842 -0.331319 -0.816875 -0.097275 -0.159226 20.942781
GT motion: -0.764571 1.502200 3.780547 -0.034805 -0.085423 20.958580
Frame 1016: positional error 0.097974 (m), angular error 0.637351 (deg), alignment error 11.623241 (cm) [total change 1.175696 51.382568]
Elapsed time: 0.17 seconds, 172.6847 msec

```



# CONCLUSIONS

## FROM TRADE-OFF STUDY

- latest space-grade CPUs 10x faster than predecessors, still slow for high-performance VBN (e.g., 0.1x)
- by offering best perf/Watt vs all platforms, FPGAs can bridge the gap with reasonable power budget (<10W)

## FROM ALGORITHM DESIGN

- edges are good/sufficient as features for ENVISAT
- rendering allows any 3D model to be used (without preprocessing/assumptions)

## FROM SYSTEM DEVELOPEMNT

- Xilinx Zynq on MMP achieves (specifically for pose estimation in ADR with passive sensors/cameras)
  - 10+ FPS for 1024x1024 images (or 5+ for 2048x2048)
  - power around 5W (peak 9W)
  - error around 1%, most often less than that

# HIPNOS ROADMAP

# FINAL ROADMAP

## TRADE-OFF STUDY WILL NEVER END

- Technology is alive, continuous improvement and new developments

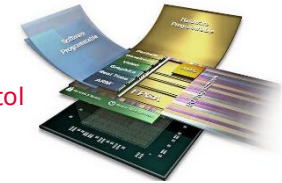
## INCREASE TRL

- Real-Time Operating System
- FDIR, EDAC, Scrubbing memory (Zynq ARM processors)
- Redundancy Mitigation: TMR, Dual-core Lock-Step (margins)
- ¿HW Shielding, SOI process, lead, current limiters, power cycle?
- Fault-tolerant additional SW design
- In-flight Reconfiguration and Supervisor
- Validation and Verification →
  - MIL → SIL → PIL → HIL
  - Tests in Representative Environments
  - Fault-Injection Tests
  - Radiation Characterization
  - Radiation Tests Campaigns
- Road to Flight Model, HW including camera detector Embedded/Clean Solution



Intel to Introduce new  
CPU-FPGA Hybrid Chip  
Supported by Acceleration Stack

Xilinx Zynq SoC road to  
Zynq Ultrascale MPSoC rad-tol



BRAVE NanoXplore Rad-hard  
SoC: FPGA + ARM processor





**THANK YOU  
questions?**



[www.facebook.com/infoGMV](http://www.facebook.com/infoGMV)

[@infoGMV](https://twitter.com/infoGMV)



DAVID GONZALEZ-ARJONA:

# Sorry, could not attend presentation due to storm

