FPGA use in future space rover navigation

G. Lentaris, (glentaris@microlab.ntua.gr)
J. Stamoulias,
M. Lourakis,
X. Zabulis,
K. Maragos,
D. Diamantopoulos,
K. Siozios,
D. Soudris,
M. Aviles Rodrigalvarez

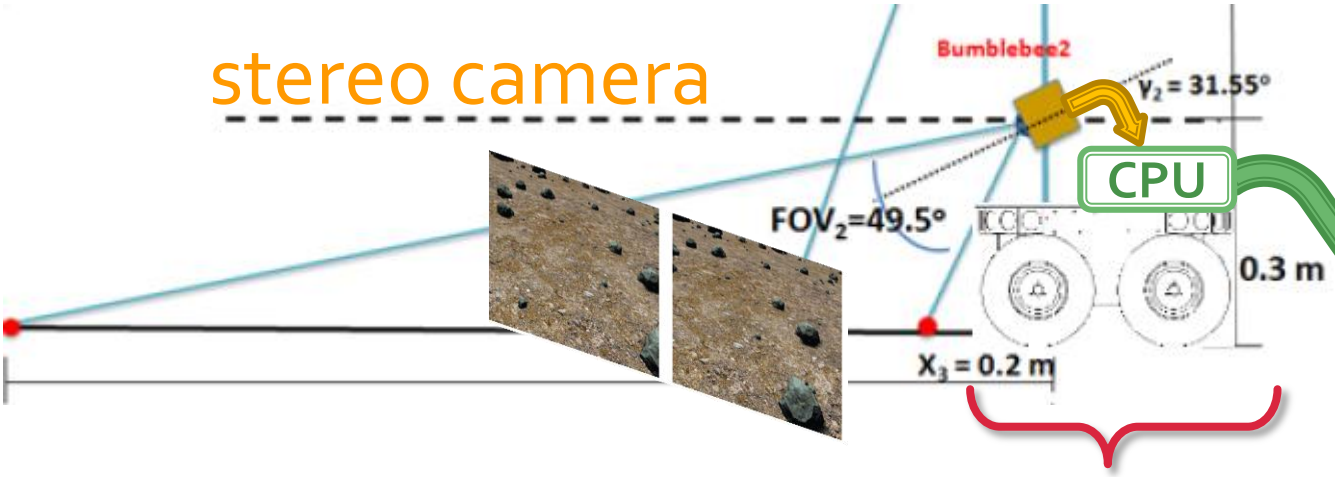
16-9-2014, ESTEC

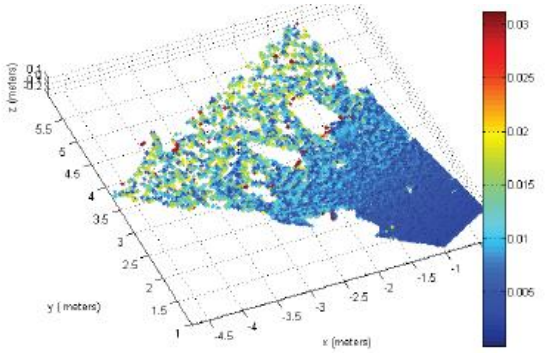
ECE, NTUA, GREECE

FORTH, CRETE, GREECE

GMV, MADRID, SPAIN

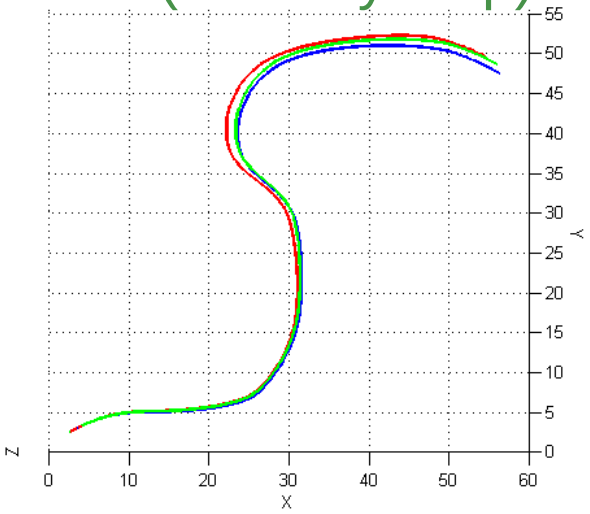
Autonomous Visual Navigation





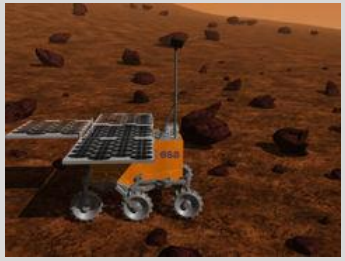
3D map

rover position
(at every step)



MER Rovers (2003)

Curiosity (2012)



ESA ExoMars (2018)

Autonomous Visual Navigation

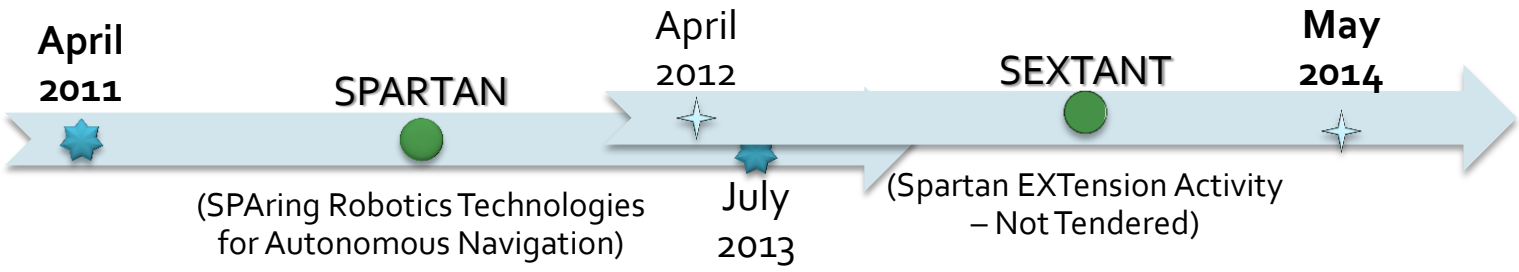
- highly complex Computer Vision algorithms
- low processing power CPUs (space-grade)
- huge execution time, not very practical to use
  - MER rover: speed only 10 m/h with VO (124 without!)
  - use only for dangerous maneuver, e.g., edge of crater

- future: faster + more accurate (more complex!)
  - 1 hour for 3D map on 150 MIPS CPU (budget = 20sec)
  - 1 minute for 1 step on 150 MIPS CPU (budget = 1sec)
  - looking for speed-up factors 10x to 1000x

Solution: Space-Grade FPGA

SPARTAN/SEXTANT projects (ESA, completed)

- HW/SW co-design of rover navigation algorithms
- commercial FPGA/CPU, emulate Martian scenarios
  - project time to 150 MIPS CPU and limit the FPGA resources
  - synthetic datasets of Mars, real images of Atacama, Devon
  - ✓ achieved "localization" in 1sec with 512x384 images
  - ✓ achieved "mapping" in 20sec with 1120x1120 images

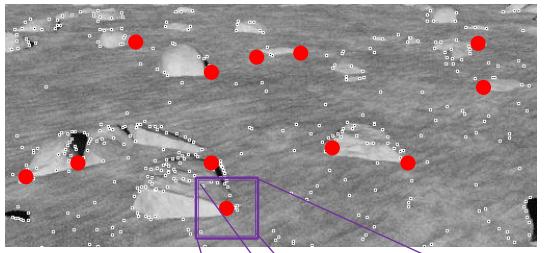


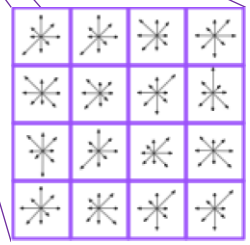
SEXTANT: Localization Algorithm

- *input*: 1 stereo image, per second (2x 512x384 pixels)
- *output*: pose of rover, per second (6D vector)

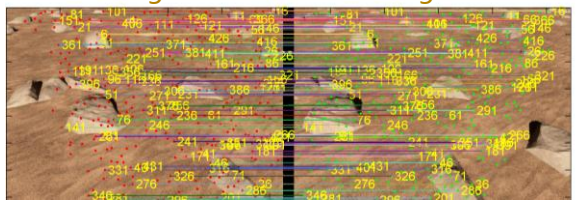
Iterations, per second:

1. Feature Detection: Harris (x2)
2. Feature Description: SIFT (x2)
3. Matching: x^2-distance (x2)
4. Filter outlier matches
5. Motion Estimation





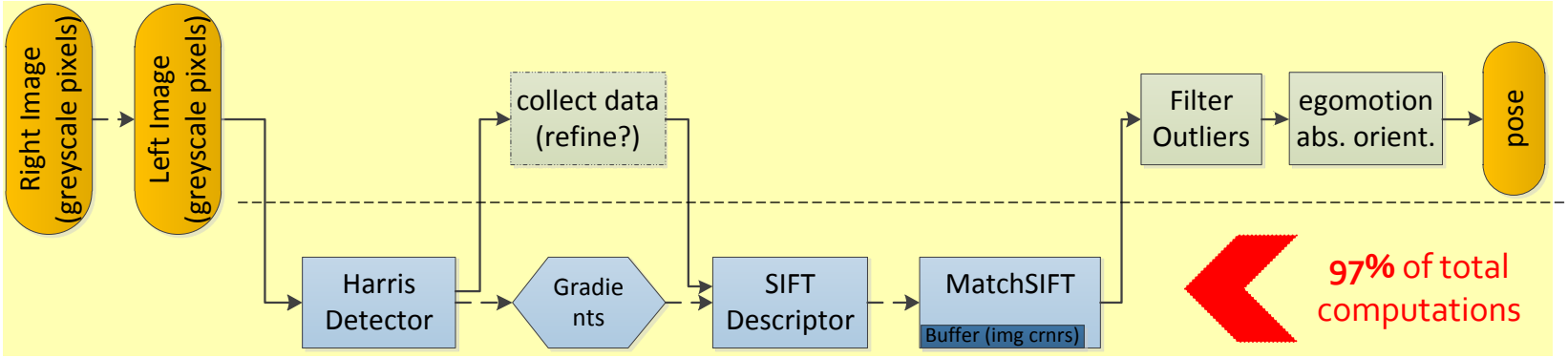
histogram of gradients



matching of histograms

HW/SW co-design methodology

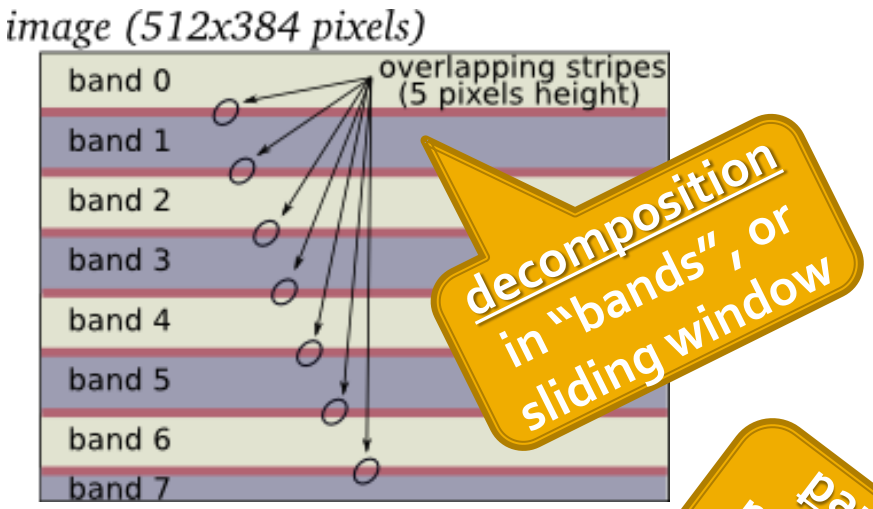
- Phase 1: SW coding, algorithm analysis/profiling
  - complexities: time, mem., comm., parallelizable, etc
- Phase 2: partitioning to FPGA and CPU
- Phase 3: design HW architecture, develop VHDL
- Phase 4: integration, tuning, verification



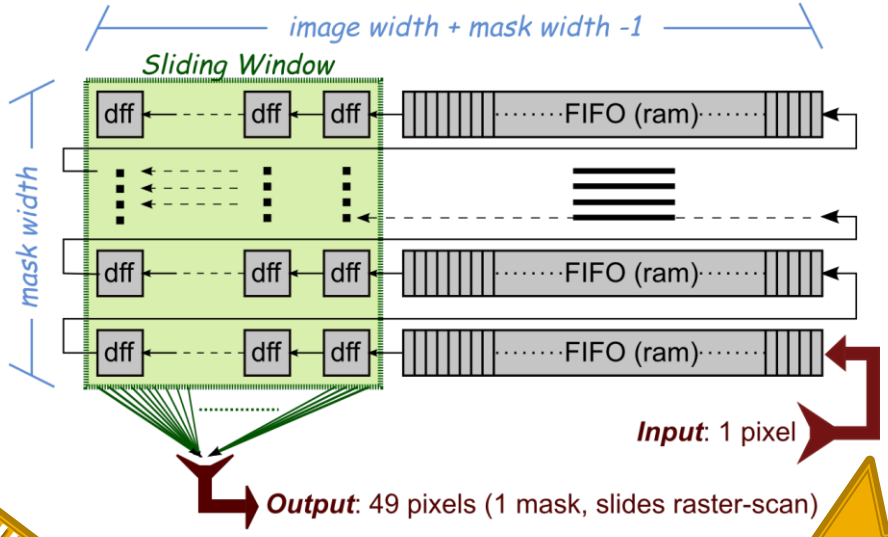
HW architecture: overview

- **Target low-cost implementations**
  - especially w.r.t. memory: bottleneck for CV on FPGA
    - resource reuse: decompose input data, process successively
- **Target sufficient speed-up** (for ESA specs)
  - pipelining on pixel-basis
    - burst read of image, transform on-the-fly (1 datum/cycle)
  - parallel memories & parallel processing elements
    - parallel calculation of arithmetic formulas
- **Target configurability** (tuning, adaptation)
  - parametric VHDL: data size, accuracy, parallelization,

HW architecture: techniques



decomposition in "bands", or sliding window



parallel memories, circular buffers

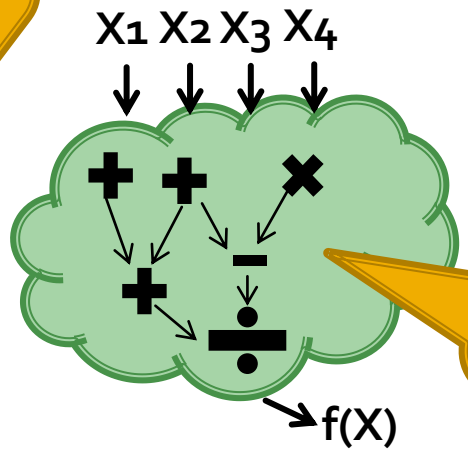
pipeline on pixel basis

map

image

4 parallel banks

1	2	1	2	...
3	4	3	4	...
1	2	1	2	...
3	4	3	4	...
...	...	...	...	...



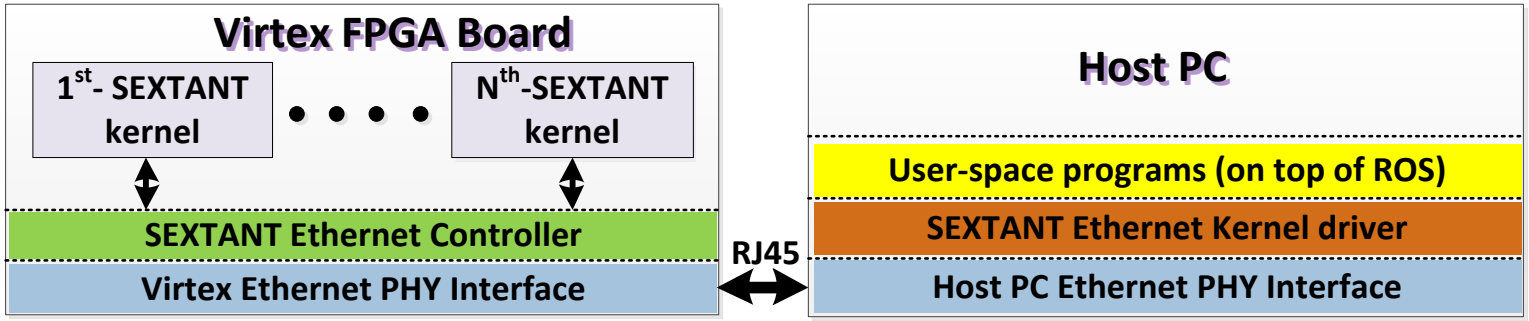
parallel calculation (+pipelined) (+parametric)

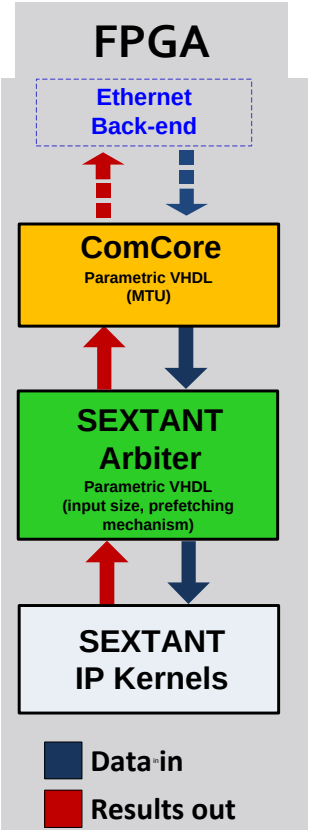
SEXTANT: Mapping Algorithm

- *input*: 3 stereo images (3x 2x 1120x1120 pixels)
- *output*: depth map (1037x3111 values)

- basic kernel: "space-sweep"
  - bruteforce search: loop over depths, loop over pixels
    - in brief: assume 200 depths in front of rover. Hypothesize pixels were recorded at each depth, back-project them to other camera based on geometry. Find correct projection
  - very intensive, entire algorithm on FPGA (99.9%)
    - double decomposition, two pixel-based pipelines, 8-bank memory, parallel aggregation of pixels (convolution-like)

CPU-FPGA communication

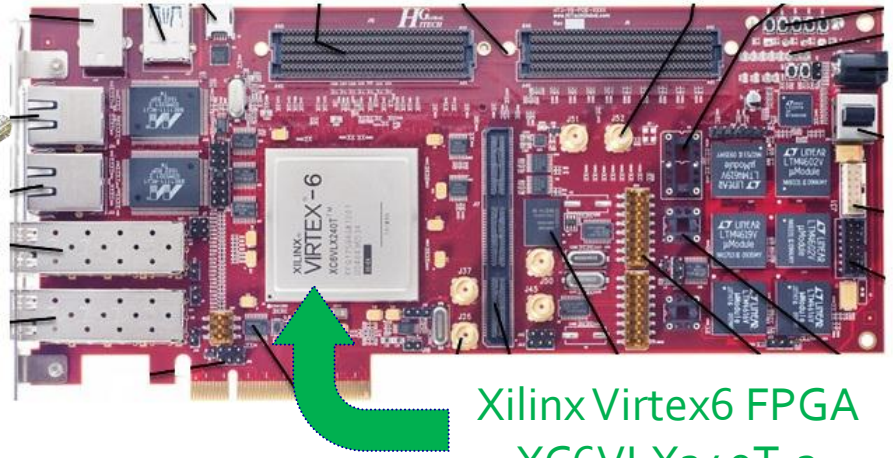




- custom scheme with **raw Ethernet**
  - on CPU: developed kernel driver
    - LKM, Rx-Tx SysCalls at Network layer, C++ API
  - on FPGA: developed data-flow controller
    - low-level functions of Link-layer by "Eth. MAC IP" from OpenCores (CSMA/CD LAN IEEE 802.3)
    - custom: packets, handshake, backoff, arbitration

System Integration, Tuning, Tests

- Intel Core 2 Duo E8400
- Running Ubuntu OS
- Executing C algorithms
  (time scaled to 150 MIPS)
- Calling FPGA accelerators



Xilinx Virtex6 FPGA
XC6VLX240T-2
(150K-LUT6, 768-DSP, 416-RAMB36)

- Modular integration: algorithms @ compile-time
  - many combinations were implemented and tested
    - SURF, SIFT, Harris, BRIEF, FAST, Matching, 2D3, Horn, …
  - tuning/exploration to meet all requirements (time, accuracy, cost): importance of parametric VHDL

Conclusion

- Successfully designed and implemented both Localization (VO) and Mapping (3Drec.) on FPGA
  - meet all ESA specifications: time, accuracy, HW cost
- Space applications can greatly benefit by FPGA
  - speed-up factors 10x to 1000x
  - robotics (not only) become more practical/useful
- Future work:
  - COMPASS: optimization, multi-FPGA (2014-2015)
  - PELORUS: put on MUSE FPGA (2M €, start in 2015?)

Thank You!
Questions?

George Lentaris, ECE, NTUA, Greece (glentaris@microlab.ntua.gr)

* this work was partially supported by "TEAChER: TEach AdvanCEd Reconfigurable architectures and tools: project funded by DAAD (2014)