


Eagle Eye – Feature Extraction from Satellite Images

Razvan Dobre
University Politehnica of Bucharest

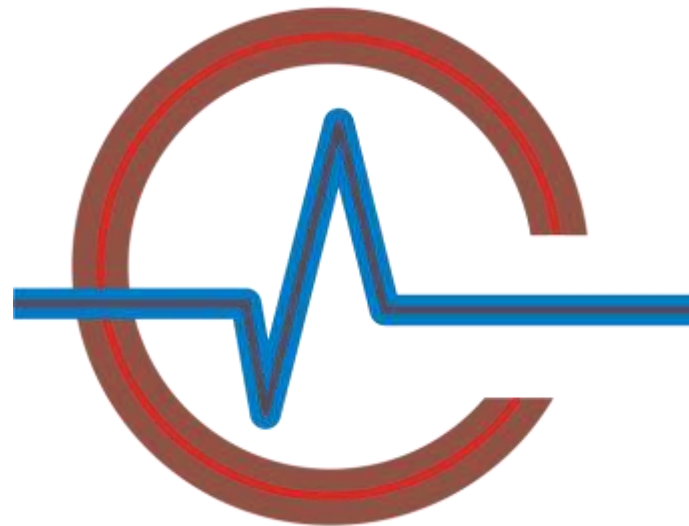


Topics

- ▶ Who am I?
 - ▶ Eagle eye
 - Introduction
 - Architecture
 - Implementation
 - Results
 - Conclusions
- 

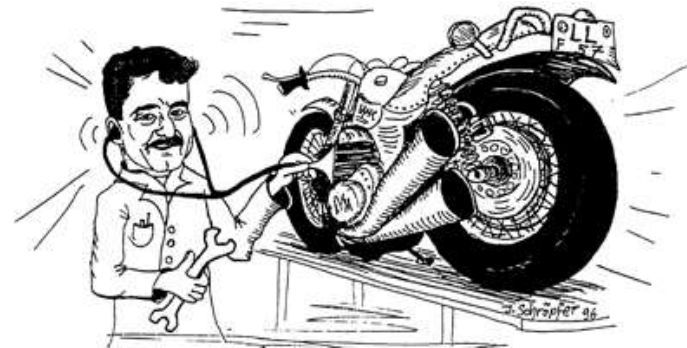
Who am I

- ▶ PhD Student at the Faculty of Automatic Control and Computers, UPB



Who am I

- ▶ My domains of interests are:
 - Networking and cluster management
 - Building a HPC cluster from ground up
 - Virtualization technologies
 - Cloud computing over normal batch-system
 - Tuning
 - “under construction”



Eagle Eye – Introduction

- ▶ Our application partially funded by the HP-SEE FP7 project
- ▶ Goals
 - Extract information from satellite images
 - Roads, forests, infields, etc
 - Reconstruct a 3D map out of 2D photo
 - Done for Romania

Eagle Eye – Introduction

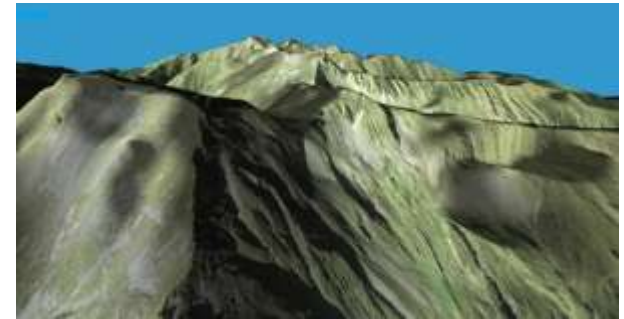
▶ Input

- High resolution material
 - 1px per square meter
 - Military maps, old ones
 - Satellite images



▶ Output

- 3D map
- Road map
- Corn fields, forests distribution

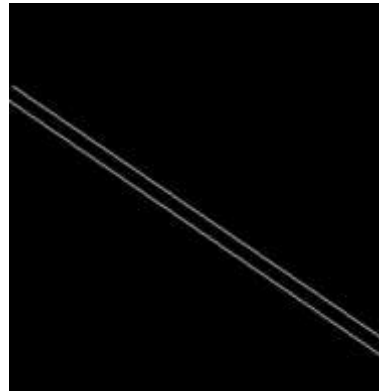


Eagle Eye – Algorithms

- ▶ Hough Transform
 - a technique that locates shapes in images
 - Lines, circles, ellipses



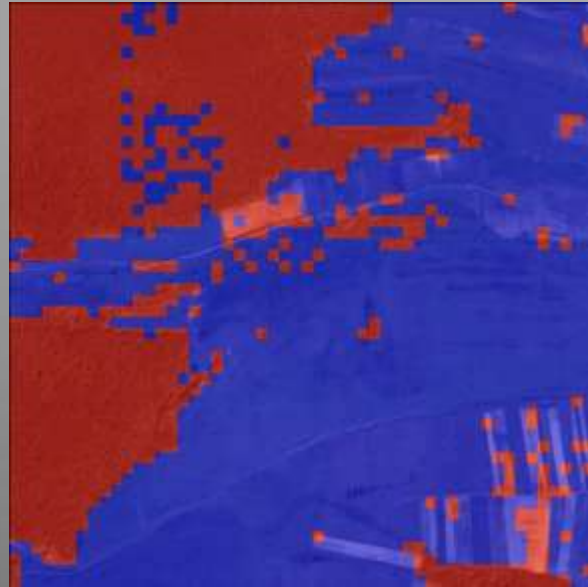
Original



Drum final

Eagle Eye – Algorithms

- ▶ Edge detection – Sobel
 - determine the sudden changes in brightness, texture etc. that the human eye identifies as a border of a given area



Eagle Eye – Infrastructure

- ▶ Hybrid implementation
 - X86 nodes
 - 2 * 4 core Intel Xeon E5405 2.00GHz
 - 16GB ram
 - 1 Gigabit interconnect
 - Cell BE i8 nodes
 - 2 * power + 2 * 8 SPE (SIMD)
 - 8GB ram
 - 1 Gigabit interconnect

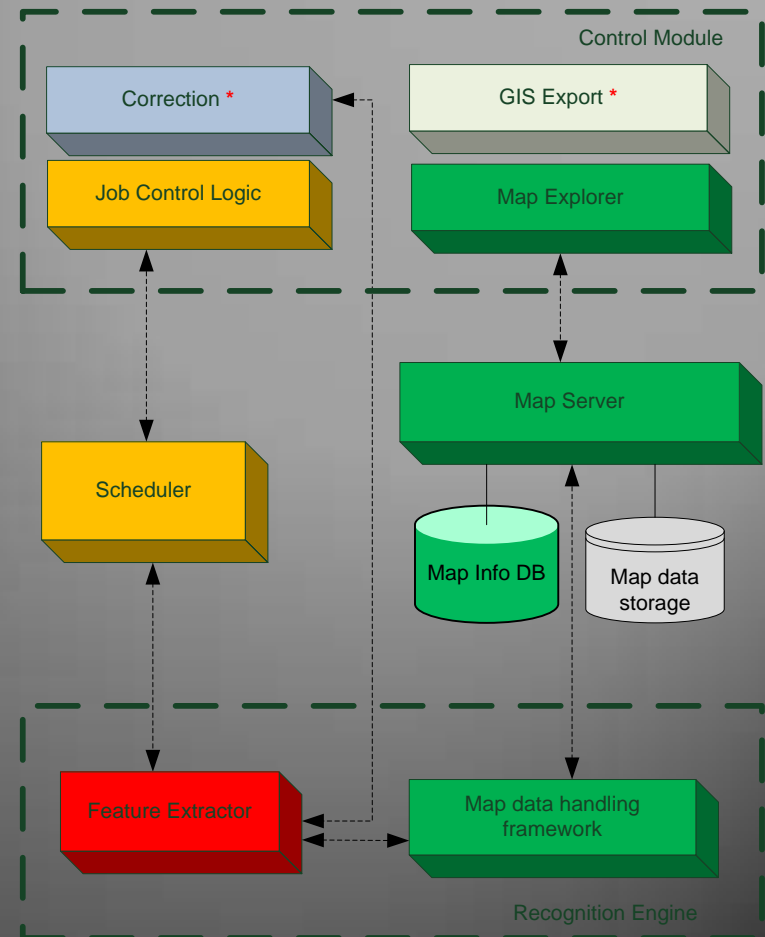


EagleEye – Infrastructure

- ▶ Why hybrid?
 - Specialized processing units for a specific set of applications
 - Image processing – applying same operations on a set of pixel
 - Accelerators (Cell Be SPE core) – good for math
 - X86 and power cores – good for branch prediction
 - TCP/IP communication between nodes (dedicated protocol)

EagleEye – Architecture

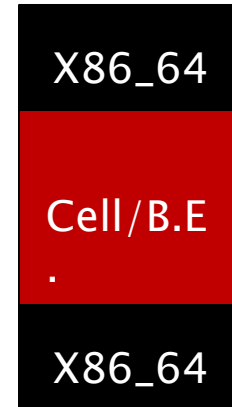
- ▶ Eagle Eye is a framework that consists of:
 - Recognition engine
 - Map server
 - Control Module
 - Scheduler



* future work

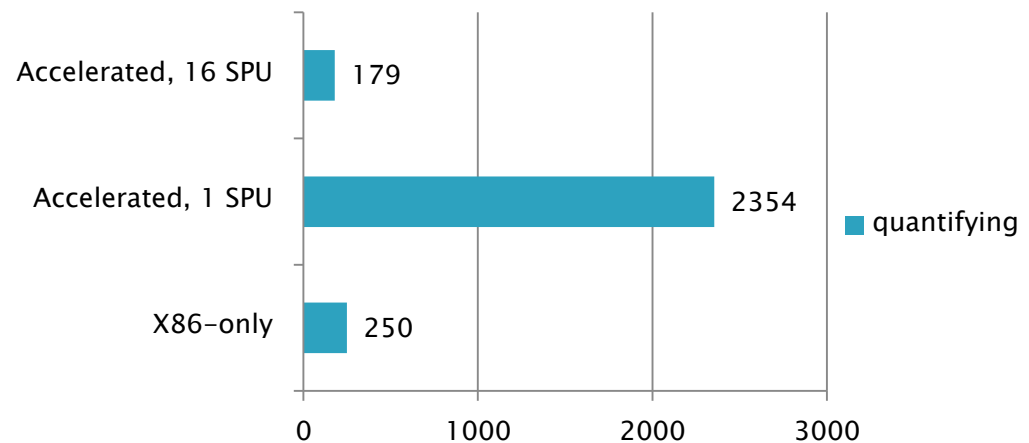
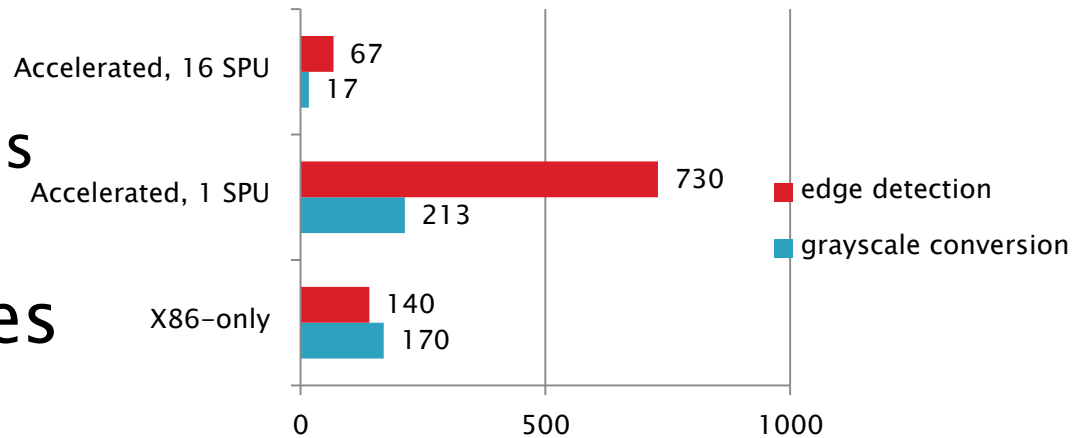
Eagle Eye – Architecture

- ▶ Processing steps
 - JPEG decompression
 - Grayscale conversion
 - Roads extraction
 - Edge detection (Sobel)
 - Hough transformation
 - Finding roads
 - Infields area and forests
 - Computing co-occurrence matrix and quantifiers
 - Classifications



EagleEye – Results

- ▶ X86-only uses Openmp 8 threads
- ▶ Accelerated uses 16 SPUs



EagleEye – Conclusions

▶ Pro

- Satisfying performance of the application
- Useful output

▶ Cons

- TCP/IP communication is a bottleneck in both latency and bandwidth
 - Use infiniband (rdma)

Thank you

