

# GPU for Industrial Inspection Applications

Presented By Raveh Gonen at  
GPU & Parallel Computing Workshop

January 6th 2011

Algorithm Infrastructure Group  
[gonen.raveh@orbotech.com](mailto:gonen.raveh@orbotech.com)

***“Virtually **Every Electronic Device**  
in the World Is Produced  
Using **Orbotech Systems**“***  
*(From [www.orbotech.com](http://www.orbotech.com), 2.1.2011)*



# GPU Computing @ Orbotech - **Timeline**

## Algorithms

3D **Depth** Estimation

Image Processing: Linear & Non-Linear **Filters**, Contour **Comparison**, **Morphology** Op, **Boolean** Image Op, **Kuwahara** & **Bilateral** Filter, F.A.S.T **Corner** Detector

Computational Geometry: **Voronoi** Diagram, **Distance** Transform, 2D & 3D Image **Registration**

## Technologies

OpenCL

**NVIDIA CUDA**

OpenGL & GLSL

2002

2008

2009

2010

# The **Way We See** GPU Computing...

## We Started Here

- OpenGL/GLSL
- Shader
- Architecture
- E

## Meantime, we

- Develop in CUDA & OpenCL
- Test new **Tools** & APIs (Thrust)
- Cont' Reading GPU/Computer Games & Graphics Books, **Academic Papers**, and attend **Conferences**
- Use our Inter

## Concentrated on...

- Fighting OpenGL/GLSL API
- Comp Community

## And...

- Make the **Fastest** code that supports **1.6Gbyte/Sec** Camera Input Stream
- Test 200\$, 800\$ & 2500\$ GPUs
- Use 2-Way & **3-Way Pipeline** on FERMI Architecture (*Quadro* GPUs)
- Test **AMD** GPUs & Intel Multi Core

## And Struggle with...

- **Non-Monotonic** GPU Performance Penalties/Behavior
- **Hard-to-debug** Kernel Execution Errors (and **Crashes**)
- Hand-Coding optimizations (SMEM, SIMD, memories, Register Usage)
- **Occupancy** Calculations

# The Way We Do GPU Computing...

Understand Orbotech **Future Challenges**  
{ Algorithmic & Computational Aspects }

Define **Gaps**



Predict **Trajectory**

Research **Papers, Tools** and Massively  
Parallel **Techniques** (Berkeley Dwarfs)

Design **Algorithms**, Implement them using  
CUDA & OpenCL (CPU Version as a Reference)

Hand Tune Host/Kernel Code for Best  
Performance (Using Multiple GPU hw)

Write Technical **Reports**, Release  
**Source Code** and **Present Work**

Spread the new information to relevant  
People (**Managers**, Algorithm Developers)

Knowledge & Code **Transfer**

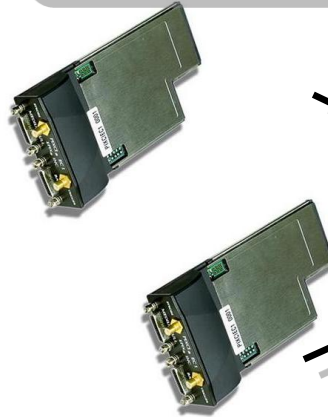
Best  
Practices  
Emerge

# The Computational Challenge...

Take ... 2 x 1.6 Giga Bytes/Sec Digital Camera



Add ... 2 x FPGA Camera Link Frame Grabbers



Process Both Streams on a Single GPU (3.2 Gb/s)



Package Them in A single Industrial PC



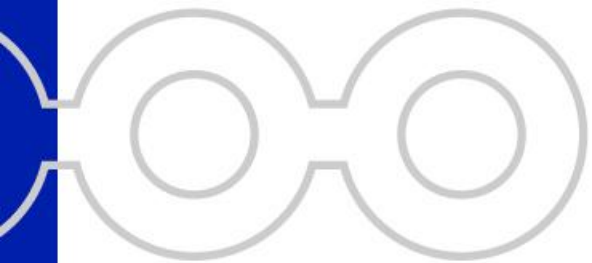
1 Move FPGA Code To GPU

2 Move Massively-Parallel-Friendly Code from CPU to GPU

3 Coarse Grain Parallelism

Stays on Multi-Core CPU

4 Shift Algorithm Developers to Massively-Parallel Thinking



Thank You!