



Real Time GPU Stereo Visual Simultaneous Localization and Mapping

Brent Tweddle May 13, 2009





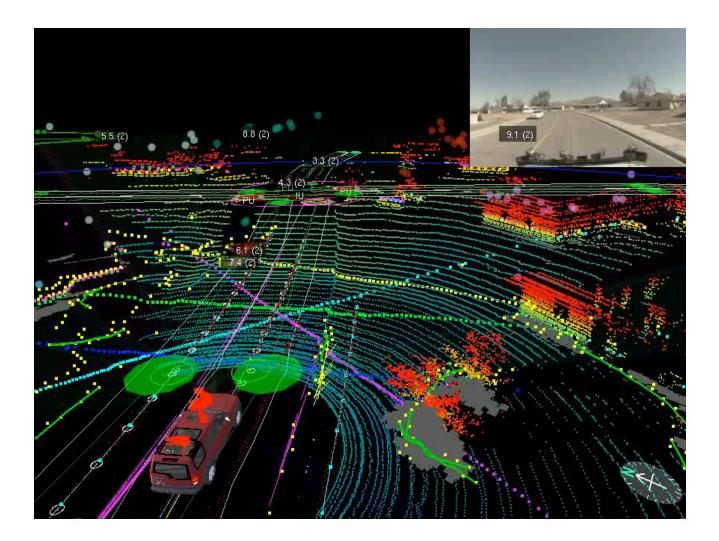
- Darpa Grand Challenge vehicles represent the current state of the art in autonomous mobile robotics
- 3 Steps performed Online
 - Navigation (Localization and Mapping)
 - Path Planning
 - Control





Perception Video





Plii

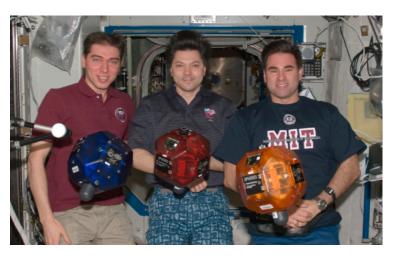
DGC Computing



- Sensors:
 - 15 radars
 - 12 Single axis lidars
 - 1 rotating lidar
 - 5 cameras
 - GPS
 - Inertial measurement system
- Computing
 - 10 Blade Cluster
 - Each Computer is Quad-Core 2.3 GHz Xeon
 - Total Power consumption: 4000W

This power consumption is impractical for a large number of applications, especially aerospace robotics.





GPU's for Real Time Robotics



Processor	Theoretical Peak GFLOPS	Watts	Watts per GFLOPS
Quad "Bloomfield" Xeon 3.2 GHz	25.6 GFLOPS	130 W	5.078
Core 2 Duo "Penryn" 2.53 GHz	20.2 GFLOPS	25 W	0.810
Cell Processor	152 GFLOPS	80 W	0.526
NVIDIA Tesla C870	518 GFLOPS	170 W	0.328
NVIDIA GeForce 9800 GT	504 GFLOPS	105 W	0.208
NVIDIA GeForce 8800M GTS	240 GFLOPS	35 W	0.145

- Assumptions:
 - Xeon issues 2 flops per cycle per core
 - Core2Duo issues 4 flops per cycle per core

http://icl.cs.utk.edu/hpcc/hpcc_desc.cgi?field=Theoretical%20peak





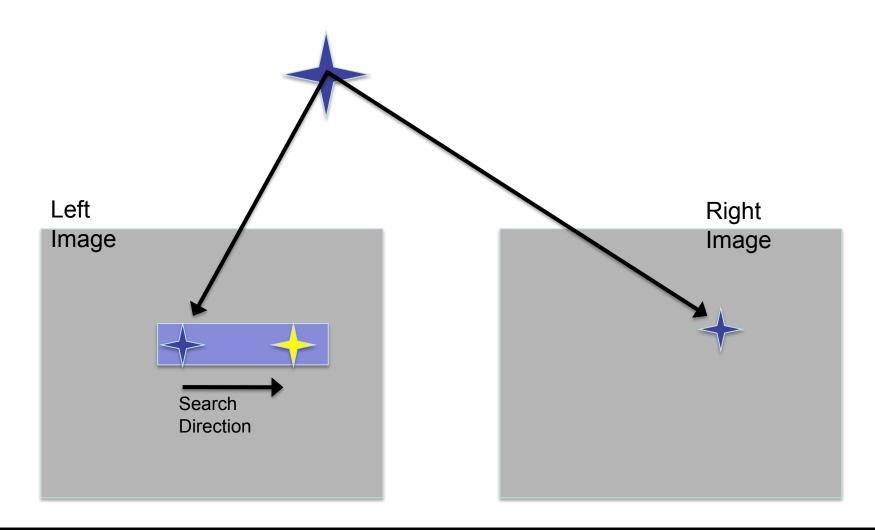
Inverse Depth Parametrization for Monocular SLAM: Loop Closing Sequence

AUTHORS: Javier Civera, jcivera@unizar.es Andrew J. Davison, ajd@doc.ic.ac.uk J. M. M. Montiel, josemari@unizar.es |'|;_



- Stereo Visual SLAM
 - Using stereo cameras create a map of your environment and locate yourself within it
- Grid Map
- Algorithm Flow:
 - Dense Stereo Correspondence (18.337)
 - Scan Matching (Thesis)
 - Particle Filter Grid Map (Thesis)





18.337: Project

Dense Stereo Correspondence





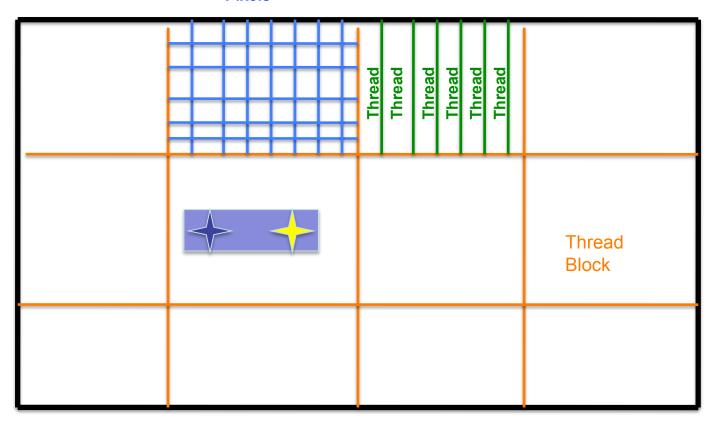
Fig. 6. Results with simulated stereo images. (a) Virtual left stereo image. (b) By S&S DP without noise. (c) By WTA MW5 Ir with noise. (d) By S&S DP with noise.

- Large body of work exists on dense stereo:
 - Scharstein, Szeliski "A Taxonomy and Evaluation of Dense Tow-Frame Stereo Correspondence Algorithms", IJCV 2002
 - Brown, Burschka, "Advances in Computational Stereo", IEEE PAMI, 2003
- Optimized algorithms for CPU SIMD hardware (512x512: <0.1s)
 - Van der Mark, Gavrila, "Real-Time Dense Stereo for Intelligent Vehicles", IEEE Trans. ITS, 2006
- Cuda Implementation by NVIDIA's Joe Stam
 - Crude and no published timings





Pixels



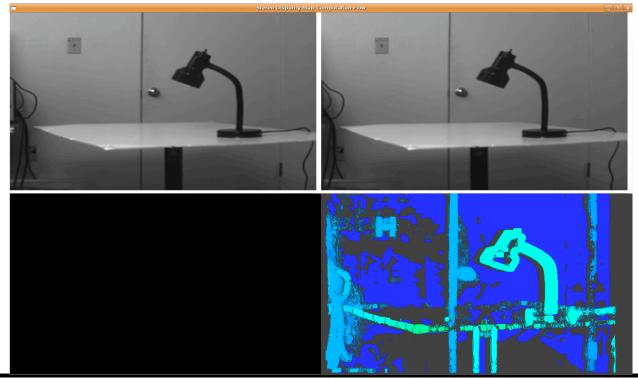


- Left-Right Consistency Check
 - Perform correspondence on both sides and check that results match
 - Naïve implementation doubles FLOPS
 - Implemented on the GPU by storing calculations in two 3D grids
 - Same number of FLOPS, but more memory is needed
 - Had to add additional kernel to avoid race conditions
- Threshold for minimization to avoid disparity noise in textureless regions
 - New < Best-250</p>

Plii



- Visually appears much more accurate
- Runs in 25ms still less than
 - More than 16ms, but still less than most CPU implementations



Performance Limitations



- Algorithm is not memory bandwidth limited
- However it is limited by:
 - Memory Latency
 - Multiprocessor Warp Occupancy
 - Compute 1.1, 20 registers, 80 bytes shared mem, 20 bytes constant mem
 - **Register Limited** Θ 🔿 🔿 CUDA_Occupancy_calculator.xls 📍 🔯 🔚 🗮 🖏 🗈 🌔 🗳 🌀 • 🗠 - 🗕 + 🍌 🥻 🛅 👸 100% • 🧿 New Open Save Print Import Copy Paste Format Undo AutoSum Sort A-Z Sort Z-A Gallery Toolbox Zoom Help Sheets Charts SmartArt Graphics WordArt G M н K N 0 0 For more information on NVIDIA CUDA, visit http://developer.nvidia.com/cud Your chosen resource usage is indicated by the red triangle on the graphs. Just follow steps 1, 2, and 3 below! (or click here for help) The other data points represent the range of possible block sizes, register counts, and shared memory allocation (Help) Varying Block Size Varying Register Count 8 2.) Enter your rese (Help) 32 Registers Per Threa Max Occur Max Occur (Don't edit anything below this line) 2 3.) GPU Occupancy Data is displayed here and in the grap Active Threads per Multiprocessor Active Warps per Multiprocessor Active Thread Blocks per Multiprocessor Multiprocessor Warp Occupancy Multiprocess Warp Occupar 384 12 (Help) 16 My Register Count 20 Occupancy of each Multiprocessor 50% Physical Limits for GPU: Threads / Warp 1.1 32 24 768 23 24 25 26 27 Warps / Multiprocessor Threads / Multiprocessor Thread Blocks / Multiprocessor 144 208 272 336 400 464 16 12 28 0 4 8 16 20 24 32 Total # of 32-bit registers / Multiprocesso Register allocation unit size Shared Memory / Multiprocessor (bytes) 8192 256 16384 Threads Per Block rs Per Thread Regis 28 Warp allocation granularity (for register allocation Varving Shared Memory Usage 32 Allocation Per Thread Block 32 2560 Reaister red Me hese data are used in computing the occupancy data in blue Maximum Thread Blocks Per Multiprocesso 39 mited by Registers / Multiprocessor Multipe Varp O 41 ited by Shared Memory / Multin Thread Block Limit Per Multiprocessor highlighted 43 CUDA Occupancy Calculator 45 145080 14008 11228 11288 11288 11089 9216 1089 9216 1089 9216 1089 9204 Shared Memory Per Three

18.337: Project





- GPU's are a valid method to use for robotic navigation
- Showed implementations of first step of navigation algorithm (accurate stereo vision)
- Analyzed performance limitations of implementation and suggested future recommendations