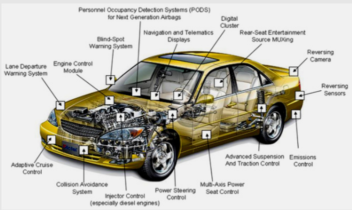
Driver Assistance:
Speed-Limit Sign Recognition on the GPU

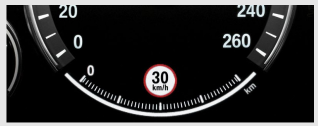
Vladimir Glavtchev, Pınar Muyan-Özçelik*, John D. Owens*, Jeff Ota
BMW Technology Office, *University of California, Davis

GPU for Automotive



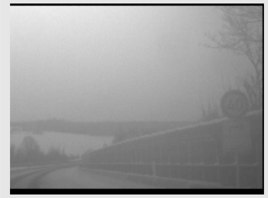
We investigate the use of different GPU-based implementations for performing **real-time** speed limit sign recognition on a **resource-constrained** embedded system.

We compare these alternative approaches at:
- ✓ success rate
- ✓ run-time
- ✓ how well each maps to the GPU



Consolidate and replace many of these with a GPU
- ✓simplifies design
- ✓is cheap
- ✓is programmable

Goals and Challenges



Foggy, snowy, rural road

Preprocessing using CLAHE
(Contrast-Limited Adaptive Histogram Equalization)

Digital sign

Tilted sign

Challenges

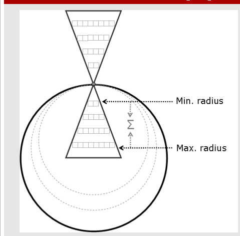
A driver-assistance system must operate reliably under all driving conditions:
- ✓ Day / night
- ✓ Rain / snow
- ✓ Fog / blinding sun

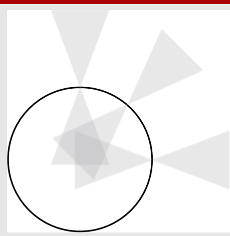
And different environments:
- ✓ Urban / rural roads
- ✓ Highways and construction zones
- ✓ Tunnels

Goals

The solution must:
- ✓ Execute in real-time on embedded-level hardware
- ✓ Achieve rates over 85% recognition, with no false positives and no misclassifications

Approach I



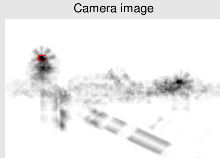


- Edge pixel voting
  - Votes speculate the location of the polygon centroid of which the pixel might be a part of
- All pixels vote in parallel
- Votes are accumulated
  - Darker areas indicate potential polygon centroid

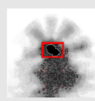
Radial Symmetry Detector

Camera image

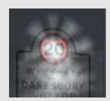
Sobel filter (CUDA)



Radial Symmetry Detector
(CUDA-OpenGL interop)



Maximum reduction
(CUDA)



Detected target

Results

Processor	Cores	Runtime	Rate (fps)	Speedup
Dual-Core Intel Atom 230 @ 1.6GHz	**2**	**235 ms**	**4.25**	**--**
Intel Core2Duo 6300 @ 1.86GHz	2	130 ms	7.7	1.8X
NVIDIA GeForce 9200M GS	**1**	**65 ms**	**15.4**	**3.6X**
NVIDIA GeForce 8600 GTS	4	23 ms	43.5	10.2X

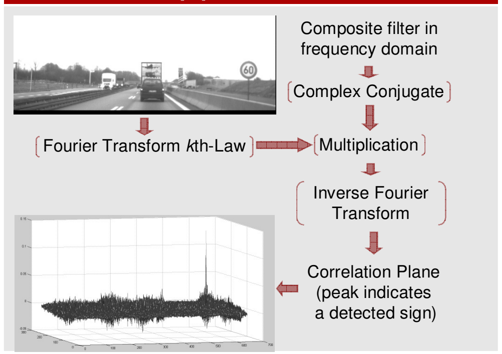
- Embedded-space aware processing
  - Single memory transfer to GPU (input image to gmem.)
  - Memory transfers avoided using PBOs, FBOs, and VBOs

Summary

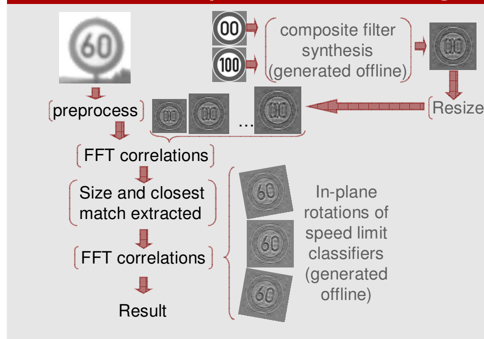
Conclusion
- Real-time sign detection only possible because of:
  - GPU-acceleration using native rendering hardware
  - Texture caching used in Sobel and Reduction steps
- Hardware Acceleration
  - Sobel and Reduction are per-pixel CUDA kernels
  - Triangular voting patterns rendered and blended using OpenGL
- First known **real-time** implementation of the Radial Symmetry Detector on **embedded hardware**

Future Work
- Detecting other traffic signs (stop signs, yield, etc)
- Improving accuracy

Approach II

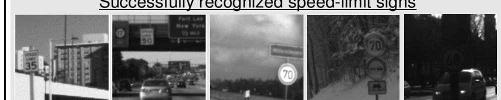


FFT Template Matching



Results

Successfully recognized speed-limit signs



Tilted | Overhead | Bright, tilted | Occluded | Dark

Region	Video Clips	Signs	Success Rate	MC[1]	FP[2]
EU	90	120	90%	0	0
United States	54	41	88%	1	0

GPU	SMs	Rate	PFU[3]	Success Rate	MC[1]	FP[2]
9600M GT	32	18.5fps	100%	90%	0	0
8400M GS	16	0.3x	-40%	-6.67	+1	+1
8800 GTS	128	2.3x	--	90%	0	0

[1]Missclassified, [2]False positive, [3]Percentage of input frames used

Summary

Conclusion
- Proved concept:
  - Near real-time execution
  - High success rates (~90%) on both European Union and United States speed-limit signs
  - Extendable to other traffic signs

GPU Acceleration
- Hardware Acceleration
  - All stages are performed in CUDA
  - FFT operations accelerated using CUFFT
  - Global and shared video memory used as a cache