The Aerospace Corporation Capstone

Adventures in Embedded GPGPU Processing

Moose Blazers: Jordan Pringle, Melissa Anewalt, Scott Walstead, Peter Gaede
Mentor: Ron Scrofano

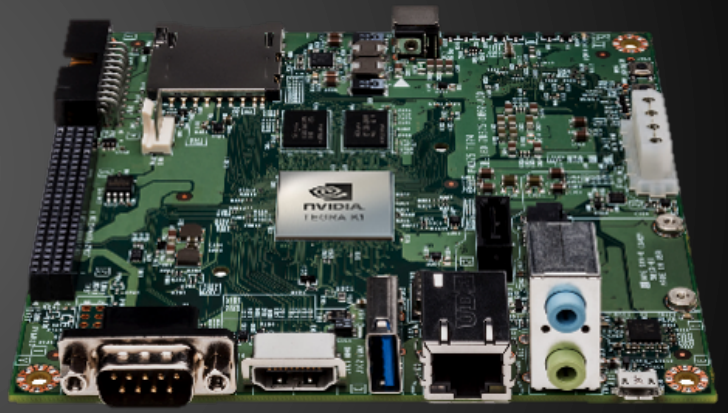
UC SANTA BARBARA engineering

Inspiration

- There is a strong demand for high performance processing in power constrained environments
- Current embedded systems meet the low power needs but do not have enough processing power with the CPU alone
- The Jetson TK1 has a powerful GPU but weak CPU performance

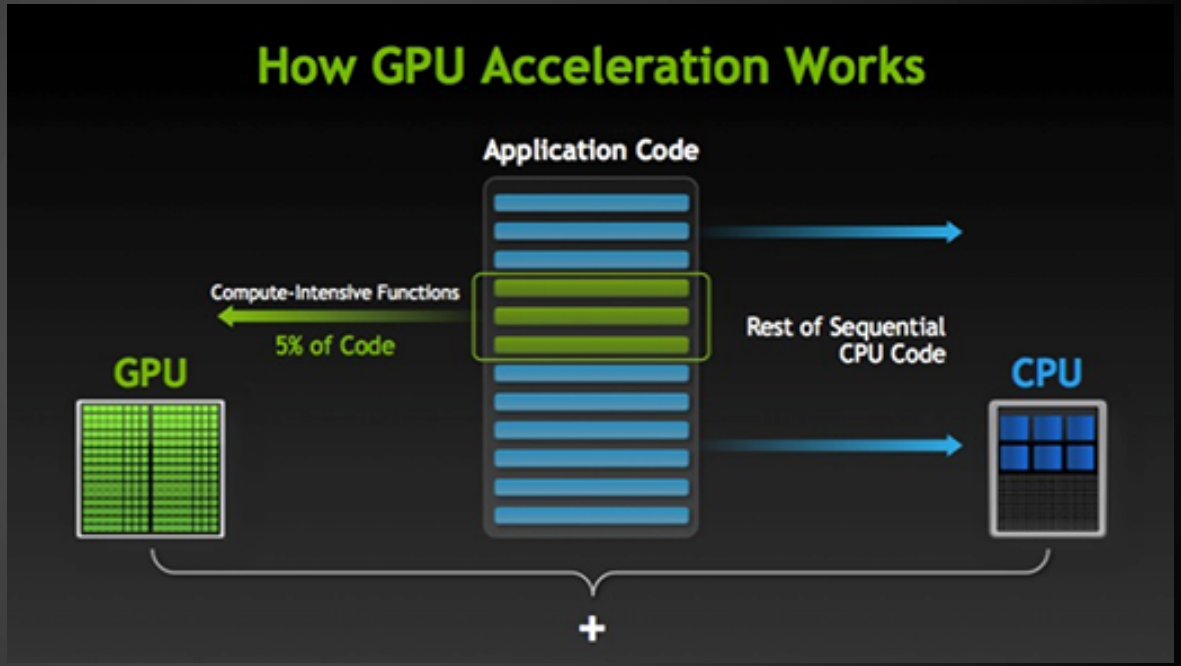
NVIDIA Jetson TK1

- GPU with 192 cores
- Quad-Core ARM Cortex A15 CPU
- CUDA support
- Uses ~5Watts under load



GPGPU Processing

- Accelerated computing
- CPU with GPU
- GPU used for independent intensive calculations



Goals

- Speed up processing using GPGPU
- Gain experience in CUDA and OpenCV C++
- Compare processing performance (serial vs. parallel)
- Create an interesting image processing application

Ping Pong

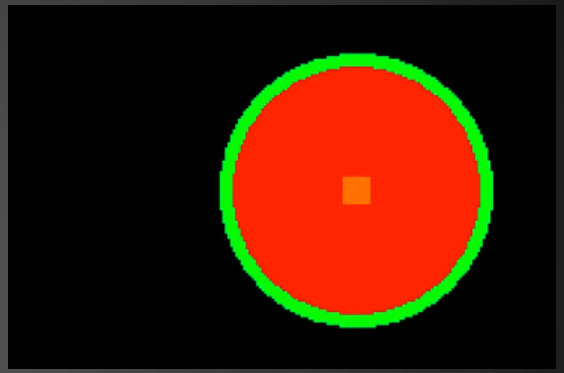
- Track the ball
- Velocity
- Predict the trajectory
- Will the ball go over the net?

Motion Tracking

- Filter by color thresholds
- Find the center
- Detect changes in position
- Compute velocity in pixels per second

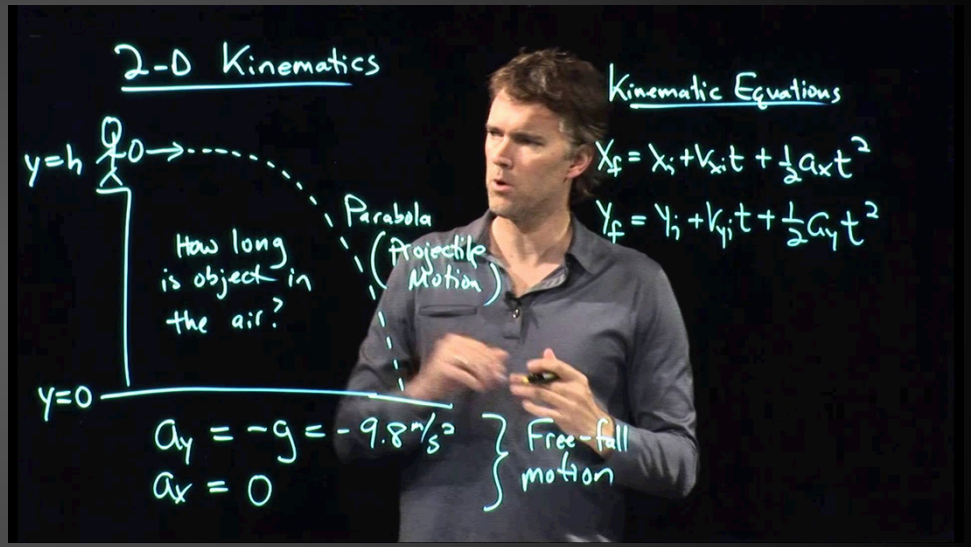
Edge Detection

- Alternative to using color
- Hoffman Circles
- Detect the center
- Problems
  - Low Frames Per Second (fps)
  - False positives
  - False negatives



Kinematic Equations

- 2D velocity and position
- Use gravity approximation
- Predict the trajectory
- Will it go over the net?

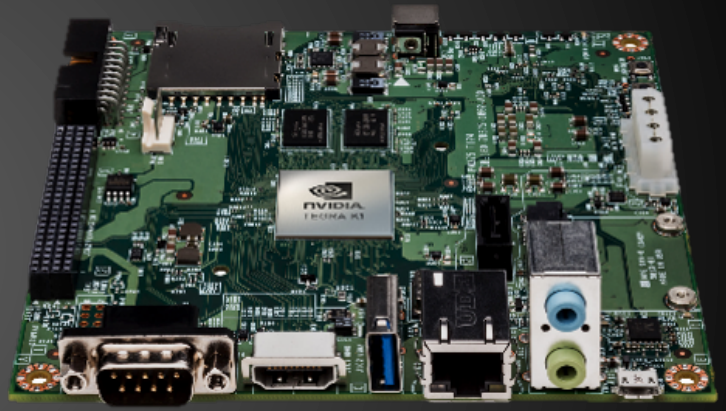


Demo

Prediction Analysis

Jetson TK1

- Set up the board
- Board was recently released
- Not compatible with certain libraries
- Random logouts

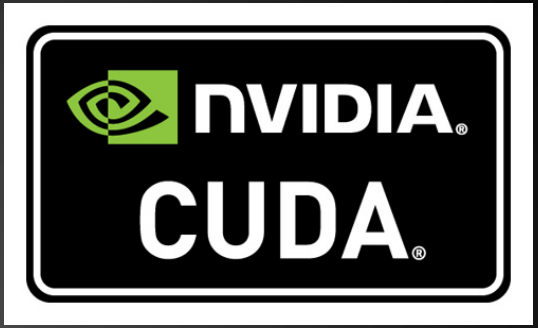


OpenCV

- Edge detection - Hoffman circles
- Motion detection using color thresholding
- Library function calls

CUDA

- Edge detection in CUDA
- Motion detection in CUDA
- Parallel processing is faster
- More difficult than serial



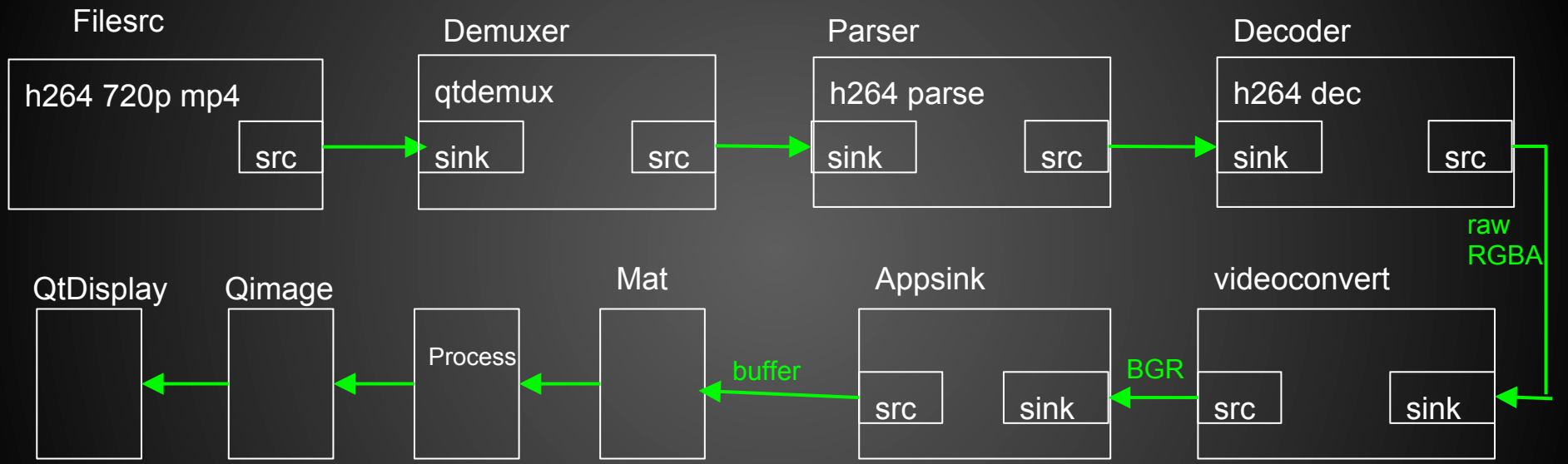
Video Input

- We first used the OpenCV method VideoCapture to bring in individual video frames for processing
- We discovered that VideoCapture was far too slow for our application
- Our solution: use Gstreamer to hardware accelerate the h.264 video decoding

Displaying the Output

- OpenCV provides imshow function
- Really laggy due to slow CPU
- OpenGL version of imshow not compatible with Jetson TK1 (OpenGL uses GPU)
- Our solution: Use Qt to display output frames

Full Implementation

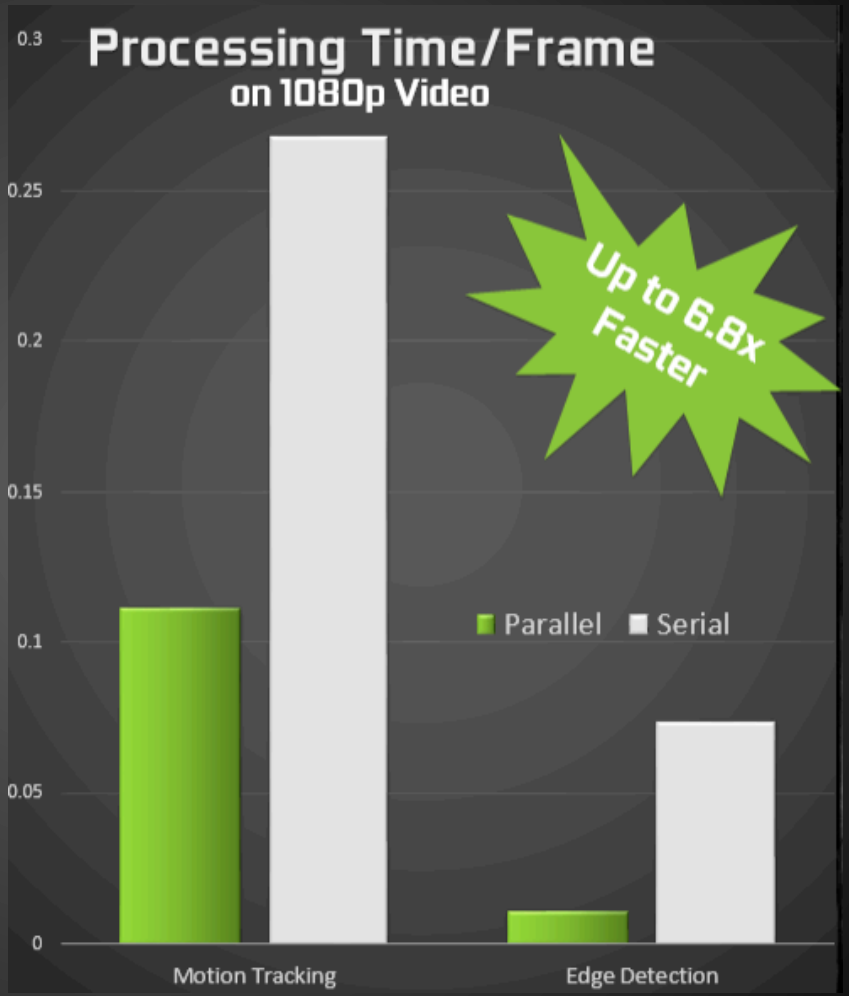


Testing

- Tried different threshold values for edge detection and color detection
- Filmed at many different angles, lighting conditions, distances, FPS, and resolutions
- Used 720p at 60 fps

Performance

- CPU vs. GPU



Other Applications - Basketball, Tennis

Conclusion

- Embedded processors and spaceborne processors operate in power constrained environments
- Computationally intensive applications, such as computer vision, are a challenge for low-power embedded systems
- GPU acceleration is viable, but presents challenges

Thank you!