Power Efficiency of Volume Raycasting on Mobile Devices

M. Heinemann, V. Bruder, S. Frey, and T. Ertl

Motivation and Goal

- Smartphones and mobile devices are ubiquitous
- Mobile volume rendering useful in various scenarios
- Rendering is computational intensive
  - High energy consumption
  - Limited battery capacity on mobile devices
- Investigate impact of parameters and API on energy usage to reduce power consumption

Methods

- Volume raycasting implementation as Android app
- Front-to-back compositing with early ray termination
- Supports OpenCL and OpenGL API modes
- Variable parameters + record/play of interaction sequences
- Measure power consumption with Intrinsyc Open-Q 820 Development Kit and ARM Energy Probe
  - Android 6.0
  - Qualcomm Snapdragon 820 SoC (including Adreno 530 GPU)

Results

OpenCL implementation

- Power consumption correlates linearly with utilization
- FPS, integration step size, resolution, interaction: only indirect impact (by changing utilization)
- Power consumption varies for different datasets
- Possibly cache coherency effects influenced by volume resolution and early ray termination
- Inaccurate measurements of frame execution times for 60 FPS
- Could be caused by v-sync / context sharing with using OpenGL (to display the result)

OpenCL vs. OpenGL

- Comparison based on same parameter settings
- OpenGL version: ~25% less energy on average, possibly caused by
  - Overhead using both API’s (draw to screen)
  - Better Android SDK integration of OpenGL

Measurement

- Automated measurement series
  - 450 test cases
  - Varied parameters: viewport resolution, frame rate, raycasting step size, dataset, recorded interaction/constant view
  - Measurement period of 60 sec.
  - Record total power consumption of the system’s core components
  - Record render time
    - utilization: rendertime / measure period time (time when GPU is active)

Conclusion

- Various parameters influence energy consumption only through computation time
- Different datasets show distinctive linear trend in power consumption
- Reducing computational time / limiting FPS seems to reduce energy consumption consistently
- OpenGL shows higher power efficiency than OpenCL