Océ

Océ PRISMAsync Controller

Using GPU's processing power in high-end printers.

océ Printing for Professionals

Maurice Luttmer
Océ Technologies B.V.

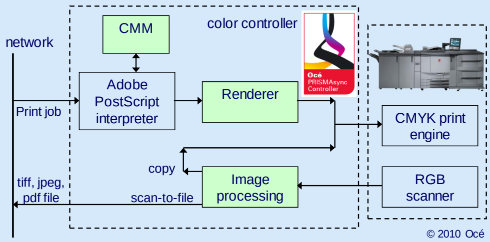
Summary of this presentation.

- What is a color controller?
  - Some context.
- Specific functions:
  - Rendering PostScript page images.
  - Color Management.
  - Copier/Scan-to-file image processing.
- Conclusions.

Océ PRISMAsync Controller

What is a color controller?

- A dedicated PC driving a color scan/print engine.
  - Functionality: PostScript printing, copying, scan-to-file.
  - Outputs CMYK images (32bits/pixel) @600 dpi to engine.
    - About 140MB per A4 at about 60 A4/minute.

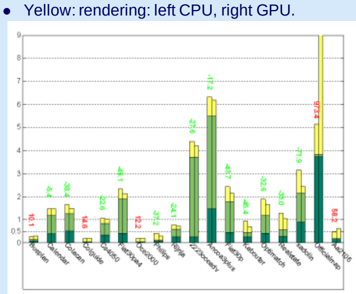


PostScript renderer.

- Paints the pixels in the output image.
  - PostScript interpreter converts page description to simple drawing primitives:
    - trapezoids.
    - sampled image (resample).
    - bitblts (e.g. cached character masks).
    - etc.
- OpenGL implementation.
  - 600 dpi.
  - Output image divided into tiles.
  - CPU: Intel Core Duo @2.66GHz, 2GB RAM(800 MHz).
  - GPU: Geforce 8500GT.

Some results: time/page for several documents.

- Dark green: PostScript language interpretation.
- Light green: color management.
- Yellow: rendering: left CPU, right GPU.



GPU renderer results.

- GPU render performance was disappointing.
  - Some documents render faster (upto factor 2).
  - But some documents render (considerably) slower.
- Render time @600 dpi not dominant.
  - But @1200 dpi, it would increase by a factor 4.
  - Color management often the main bottleneck.
- Better solution to improve PostScript render speed
  - PostScript rendering is generally memory bound.
  - E.g. painting all pixels white at start of each page.
  - Get speedup by removing memory bottleneck.
    - Render in compressed output image (on CPU).

Color management.

- Convert device color values and get the “same” color.
  - RGB for a certain monitor to CMYK for a certain printer.
  - CMYK for a certain printer to CMYK for another printer.
- To make it fast:
  - Construct a 3D (RGB input) or 4D (CMYK input) LUT.
    - e.g. 33x33x33 or 17x17x17x17.
  - Each LUT entry contains a CMYK output value.
  - Interpolation within the LUT (multi-linear or tetrahedral).
- Using LUT still takes considerable time on CPU.
  - See diagram on slide 5.

Color management on a GPU.

- Resulting color only depends on input color.
- And there are very many input colors in a photograph.
  - So well suited for massively parallel solution.
- Speedup of factor 7 to 10 compared to CPU.
  - Depending on CPU and GPU.
  - Transfer over PCI is large part of color management time.
  - Considerable improvement on total time (see slide 5).

Copy/scan-to-file image processing.

- Several image processing steps needed, a.o.:
  - User color adjustments (contrast, brightness, color balance)
  - Moire reduction (for halftoned originals).
  - Edge enhancement (for text).
  - Color management (RGB-> RGB, RGB->CMYK).
  - Scaling.
  - Rotation.
  - Etc.
- CPU implementation infeasible at scan speed (40 ppm).
  - So no complete prototype was developed on CPU.
  - 2 prototypes on GPU: OpenGL and CUDA.
  - Geforce 9800GT.

Results.

- CUDA: 250ms/A4, OpenGL: 500 ms/A4.
  - More than fast enough.
- Flexible development:
  - CUDA prototype used to optimize print quality.
    - A bunch of kernels performing useful processing steps.
    - Usable by print quality experts with no GPU knowledge.
  - Much more flexible than (traditional) FPGA solution.
- Optimizing performance of CUDA version is complex.
  - OpenGL has few possibilities for optimisation.

Conclusions.

- Adding GPU to color controller helps:
  - Good performance improvements at low price.
  - New implementation technology for copy/scan path.
    - Much more flexible than FPGA’s.
- CUDA faster than OpenGL.
  - but more effort and learning required to optimize.
- Beware of platform instability:
  - Memory leak in driver (OpenGL).
  - Inconsistent performance behavior on Vista (OpenGL).
  - Random CUDA “unspecified lauch faillure” errors.