

# Equalizer

## Quickstart and Demonstration Guide



# Building Equalizer

---

- Get source tree from svn repository
- Linux, Mac OS X:
  - `cd src; make`
  - set library path as printed by make
  - see also src/README
- Windows:
  - Build src/VS2005/Equalizer.sln



# Running the Server

---

- Linux:  
`./server/eqServer.<arch> [configfile]`
- Mac OS X:  
`./server/eqServer [configfile]`
- Windows:
  - debug 'Equalizer Server'
  - OR: `build\VS2005\win32\debug\eqServer`



# Running the Example Application

---

- Linux:

```
cd src/examples/eqPly;./eqPly.<arch>
```

- Mac OS X:

- start X11

```
cd src/examples/eqPly;./eqPly
```

- Windows:

- debug 'eqPly Example'

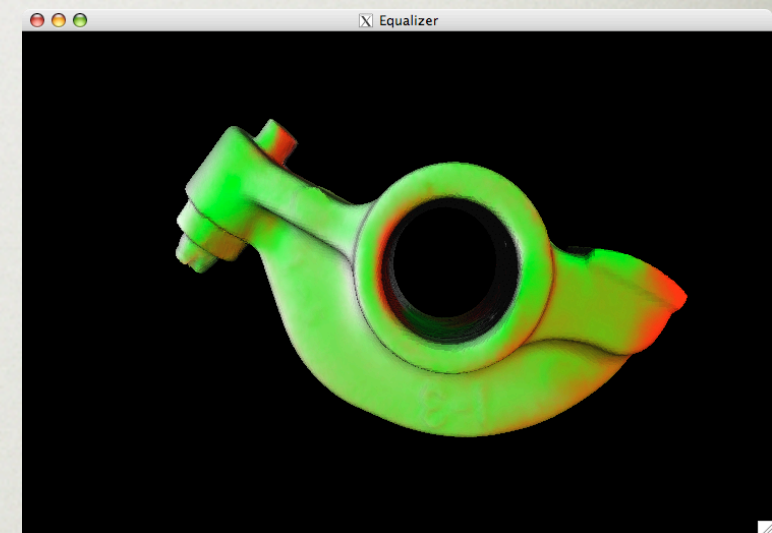
- OR: build\VS2005\win32\debug\eqPly Example



# Running the Example Application

---

- eqPly runs now with default config
  - one window, one pipe thread, one process
- Left mouse button rotates
- Middle mouse button zooms
- Right mouse button moves
- Exit by pressing <Esc>, all three mouse buttons or using window close button





# Exploring Equalizer

---

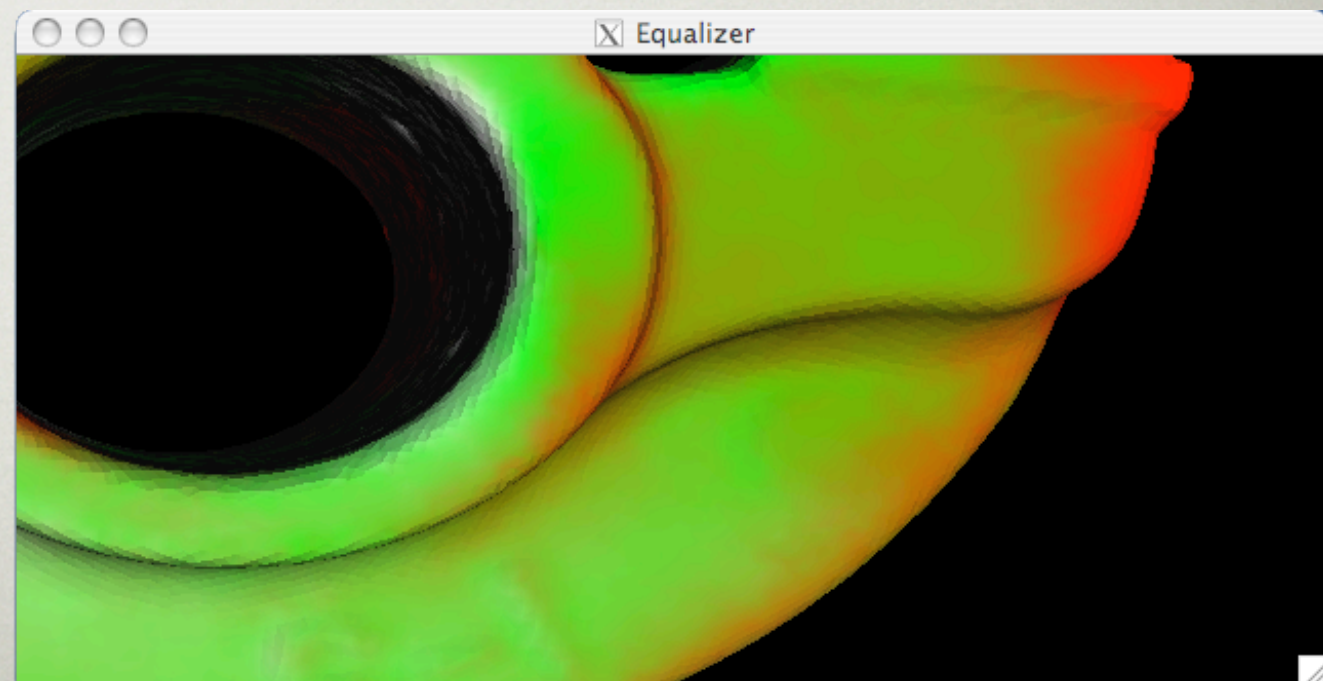
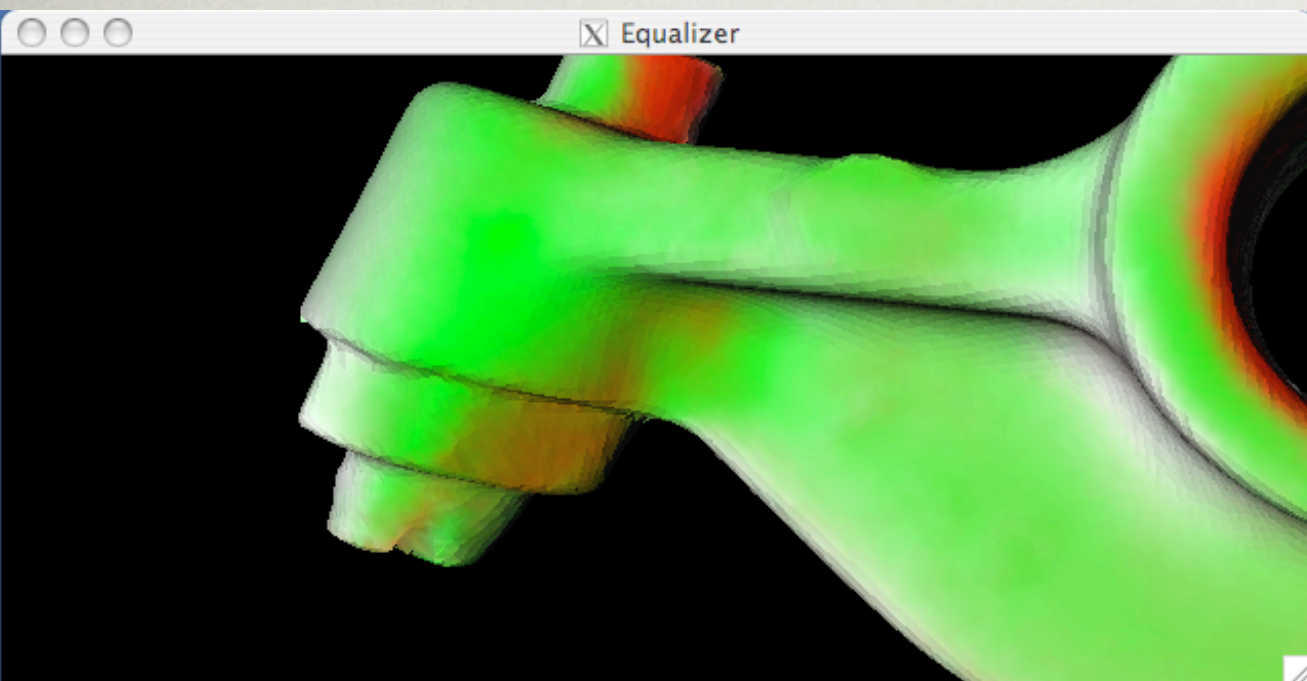
- To use a different config:
  - exit eqPly; stop server
  - start server with new config:  
`./server/eqServer examples/configs/2-window.eqc`
  - run eqPly again
- Load model with '--model <name>'
- Sample Models at [www.cyberware.com](http://www.cyberware.com)



# 2-window

---

- Two windows, one pipe thread
- Compound wall description to produce side-by side image

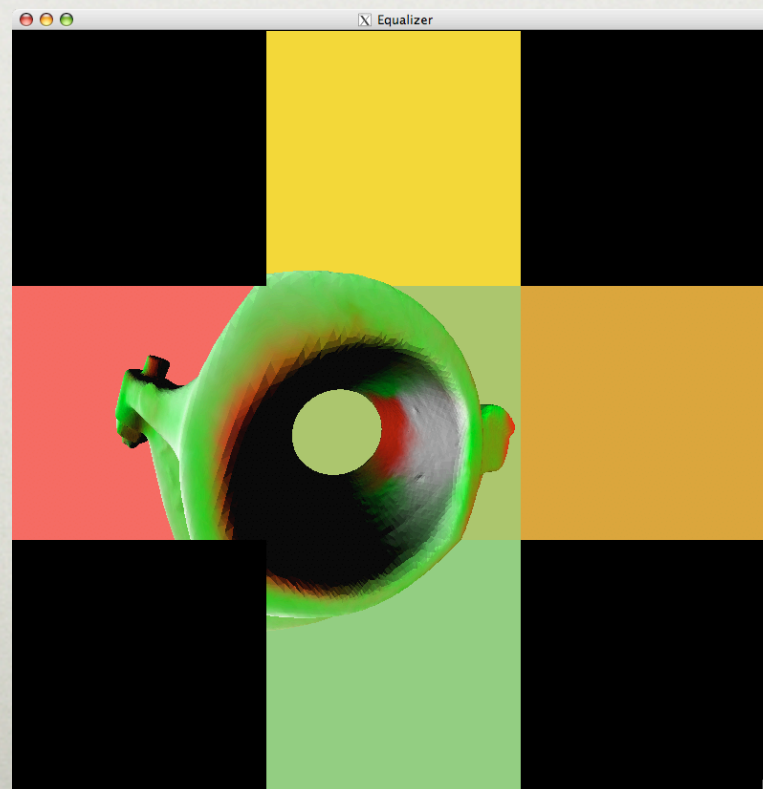




# 2-window

---

- Set EQ\_TAINT\_CHANNELS to get channel background colors
- One window, five channels
- Simulate a CAVE™ on a single PC

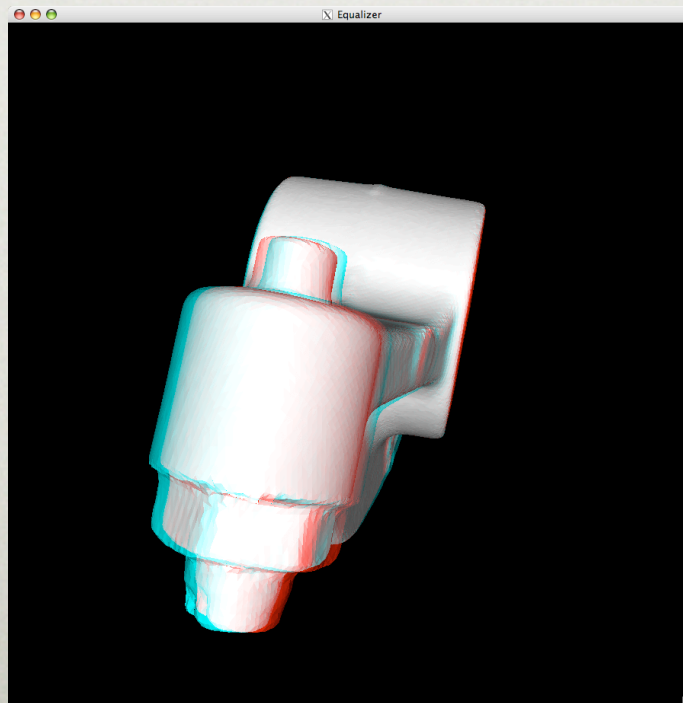




# 1-pipe.stereo.anaglyph

---

- Start eqPly with option -b
- Use anaglyphic (colored) glasses
- Two sequential eye passes
- Support for active (quad-buffer) stereo

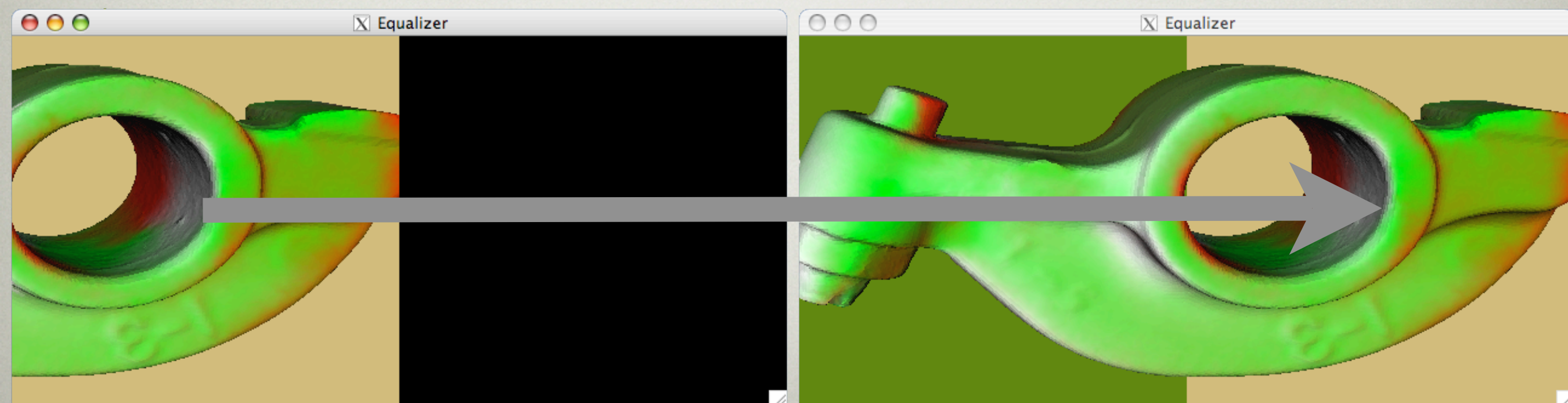




# 2-window.2D

---

- Left window renders half of the viewport for right window
- For deployment, windows are on separate pipes (GPUs) for scalability

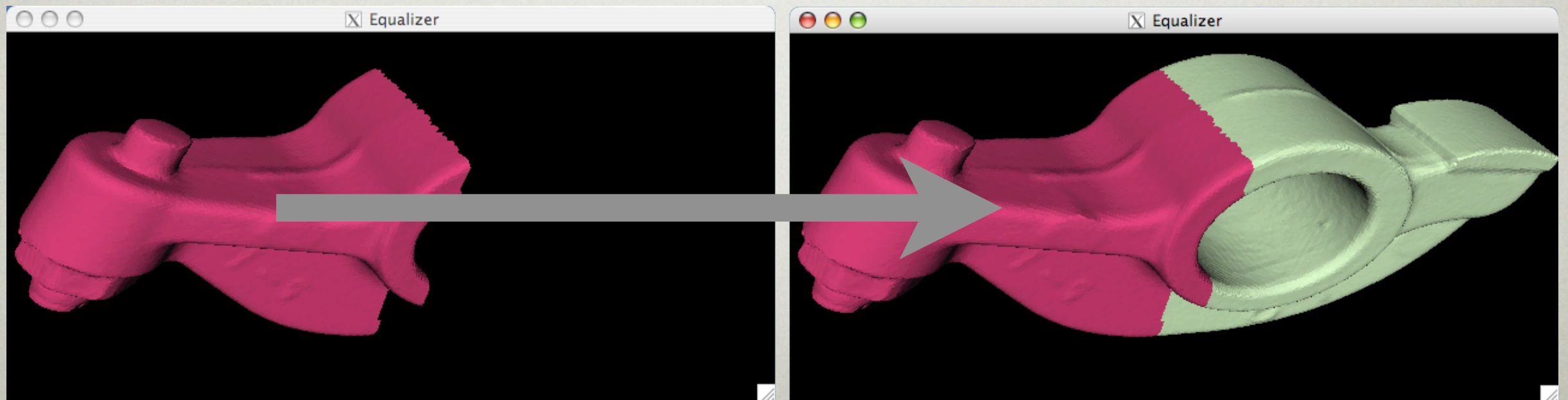




# 2-window.DB

---

- Left window renders part of the database for the right window
- Coloring is implemented in eqPly
- Data is combined using Z-Buffer information

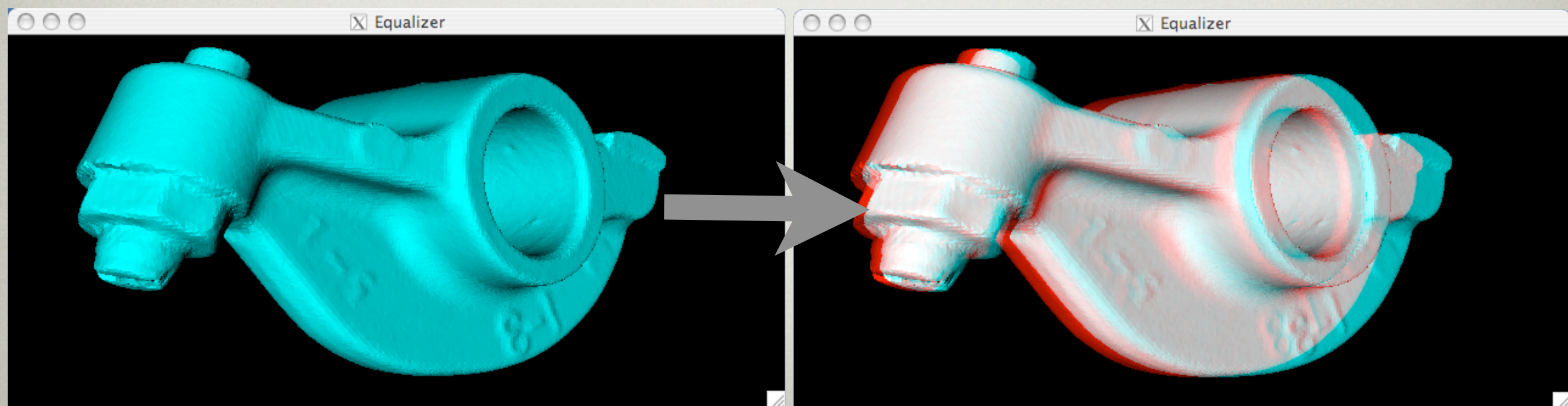




# 2-window.EYE.anaglyph

---

- Left window renders right eye
- Right window renders left eye
- Very good scalability on two pipes
- Also works for active stereo

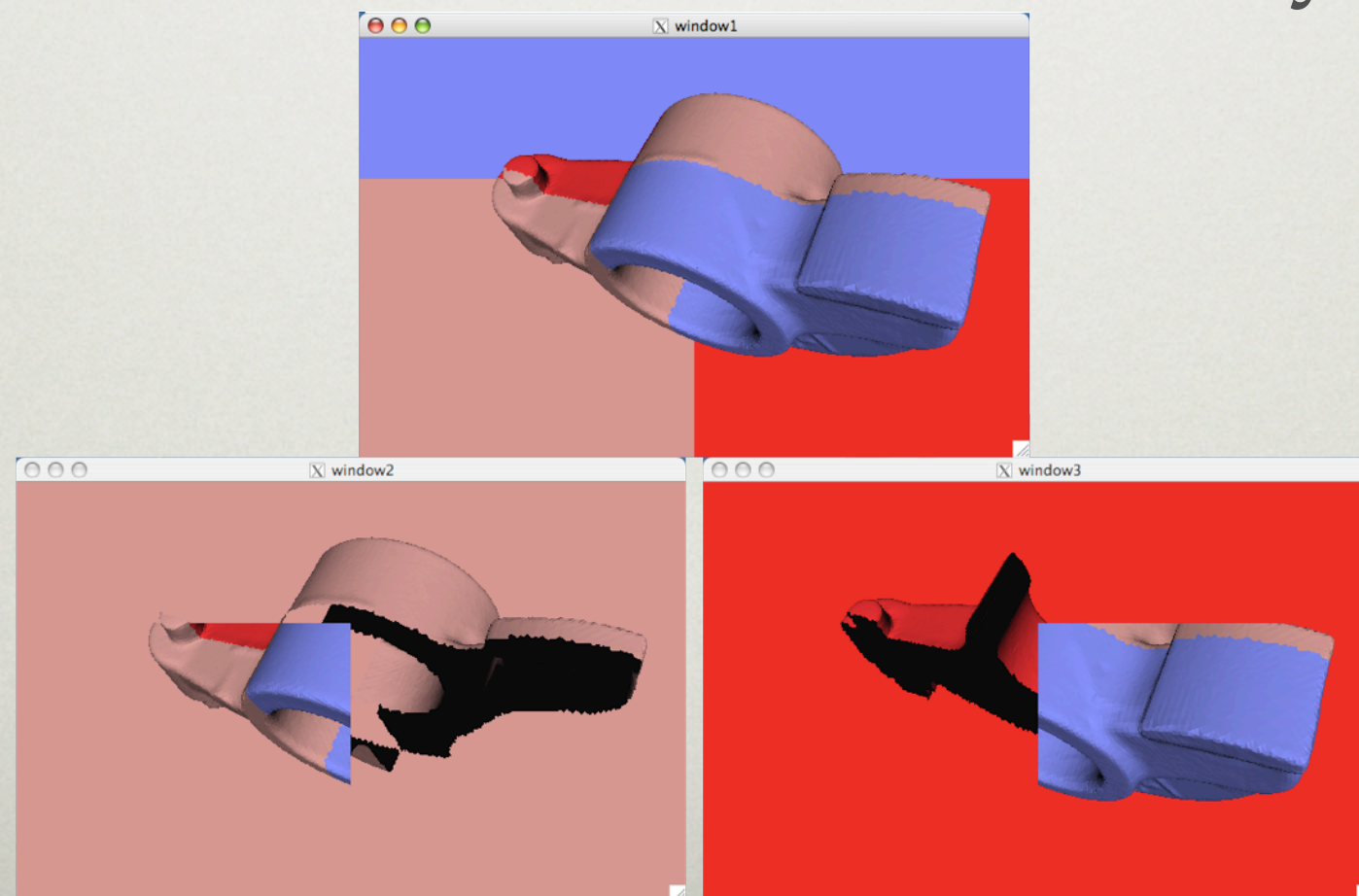




# 3-window.DB.ds

---

- Parallel compositing (direct send)
- Each channel renders and composites
- Run 4-window.DB.bs for binary swap

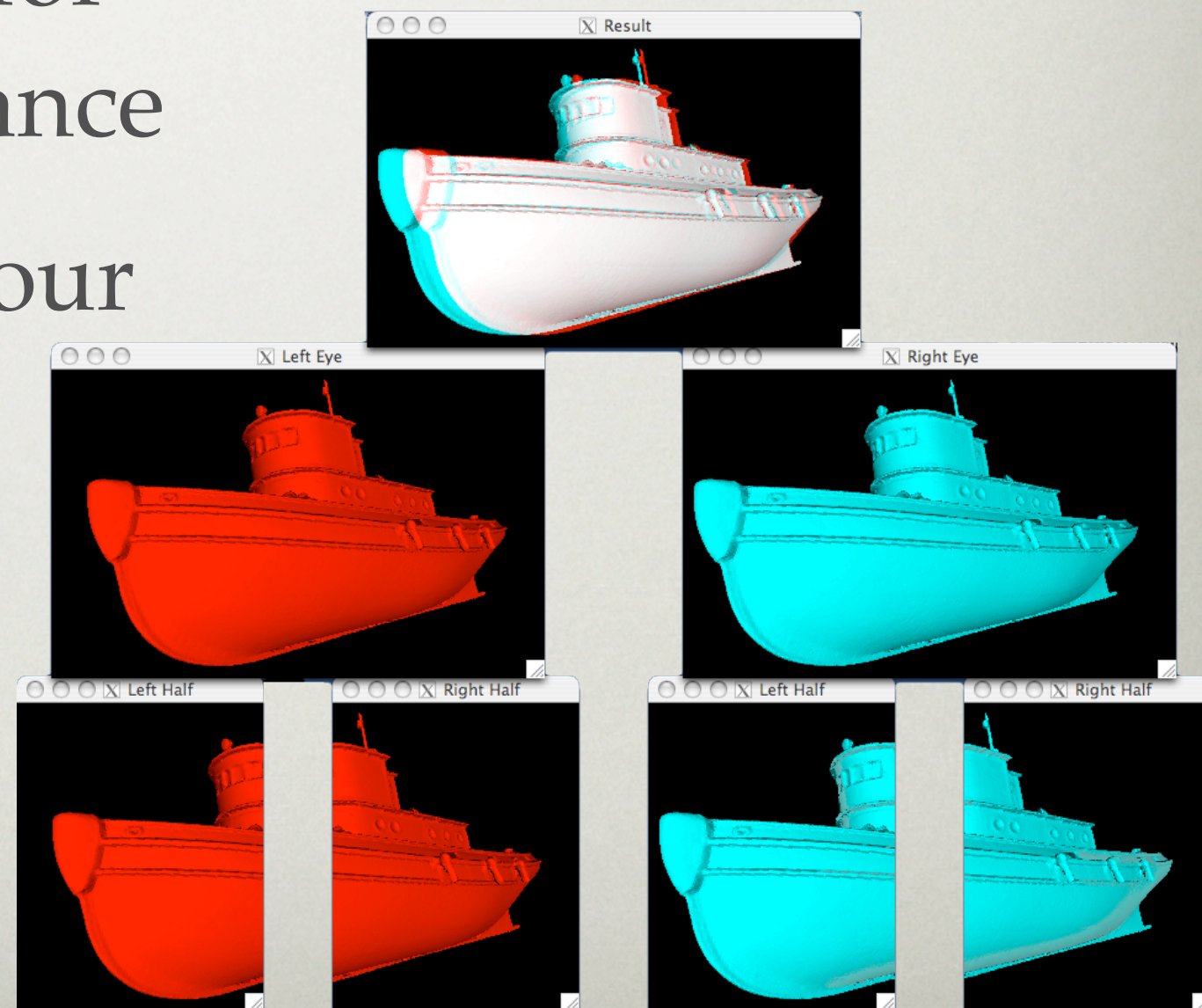




# 7-window.EYE.2D

---

- Multilevel configuration
- Combine modes for optimal performance
- Deployment on four pipes





# Next Steps

---

- Cluster configurations need password-less ssh setup to launch processes
- Change hostnames to reflect your setup
- Active stereo requires stereo visuals, i.e., high-end graphic cards
- Configuration file specification is available online