

# Advanced Multi-Frame Rate Rendering Techniques

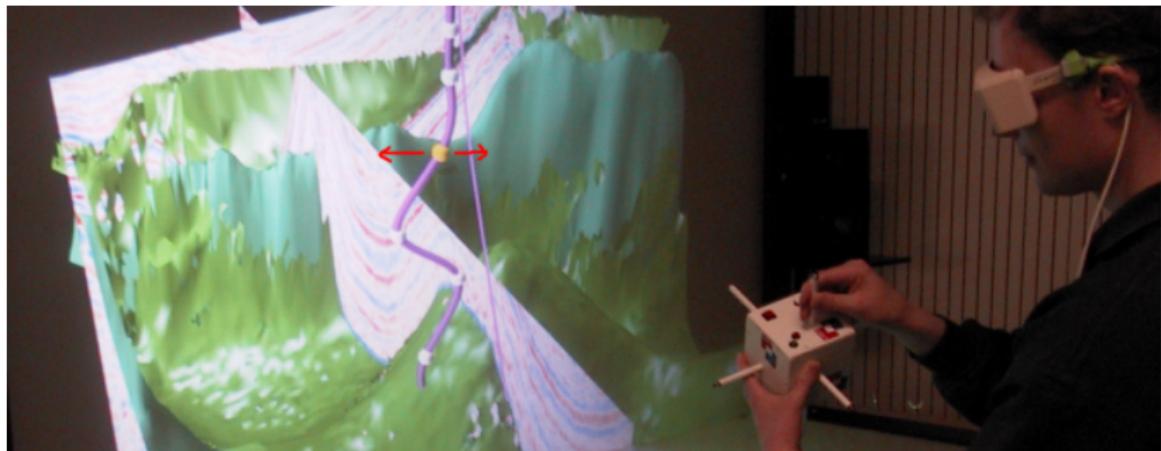
Jan P. Springer<sup>1</sup> Christopher Lux<sup>1</sup>

Dirk Reiners<sup>2</sup> Bernd Fröhlich<sup>1</sup>

<sup>1</sup> Bauhaus-Universität Weimar

<sup>2</sup> University of Louisiana at Lafayette

# Observations for Complex Applications



## High Frame Rates:

- ▶ Object manipulation
- ▶ System control

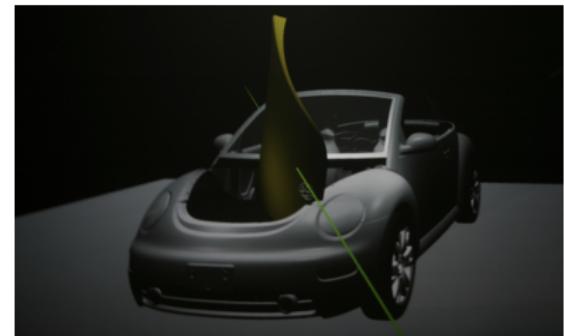
## Low(er) Frame Rates:

- ▶ Head tracking
- ▶ Navigation

# Multi-Frame Rate Basics



Slow client



Fast client

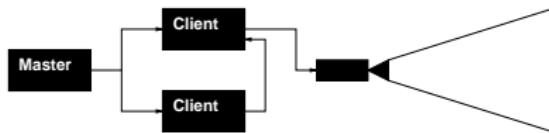
# Multi-Frame Rate Basics



Slow client



Fast client



Digital composition setup

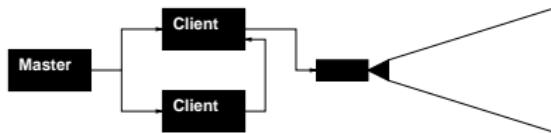
# Multi-Frame Rate Basics



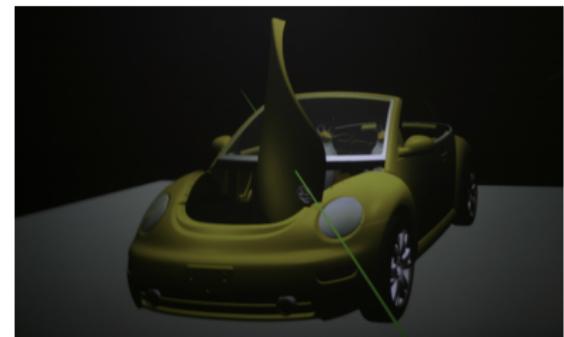
Slow client



Fast client

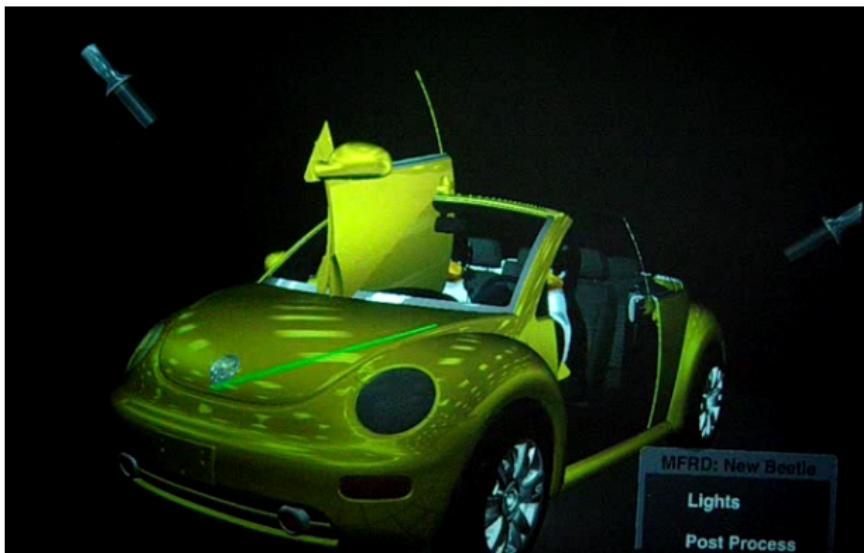


Digital composition setup



Digitally composed image

# Multi-Frame Rate Basics



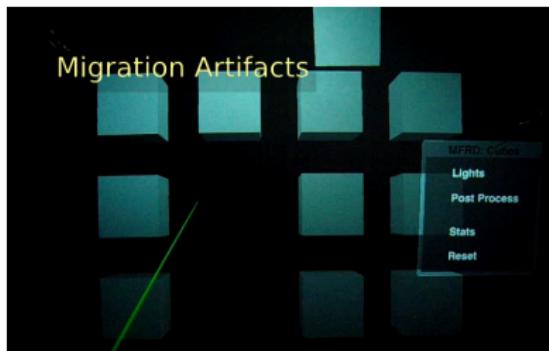
# Migration Artifacts for Object Manipulation

- ▶ Selection Artifact
- ▶ Release Artifact
  
- ▶ How to avoid these artifacts **without** introducing synchronization?

# Migration Artifacts for Object Manipulation

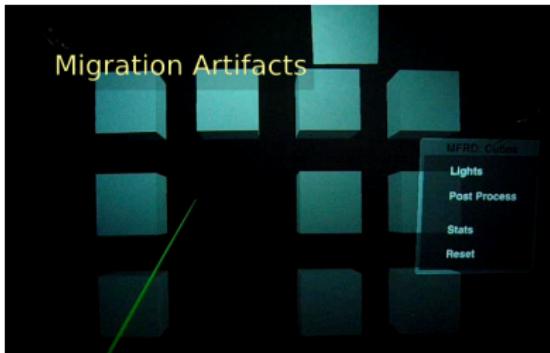
- ▶ Selection Artifact
- ▶ Release Artifact
  
- ▶ How to avoid these artifacts **without** introducing synchronization?

# Example: Selection Artifact

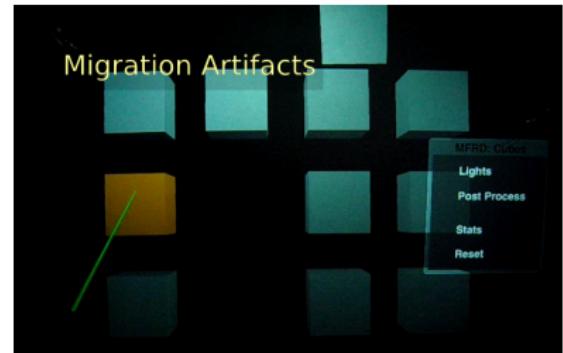


Frame: N-3

# Example: Selection Artifact

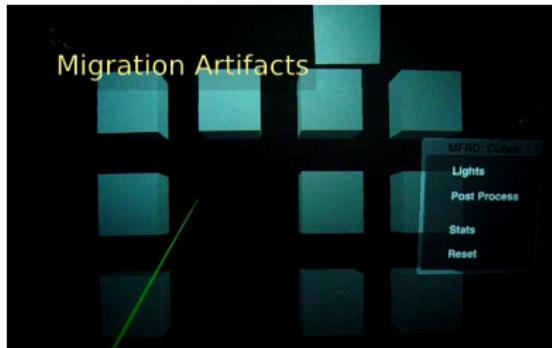


Frame: N-3

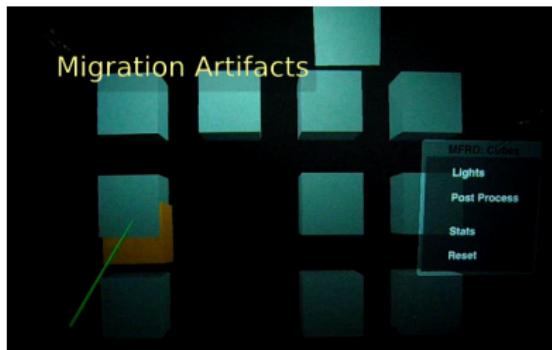


Frame: N

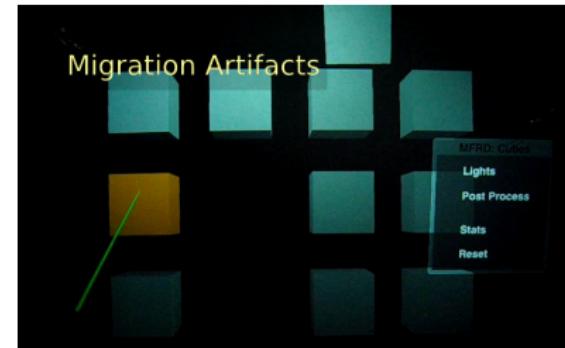
# Example: Selection Artifact



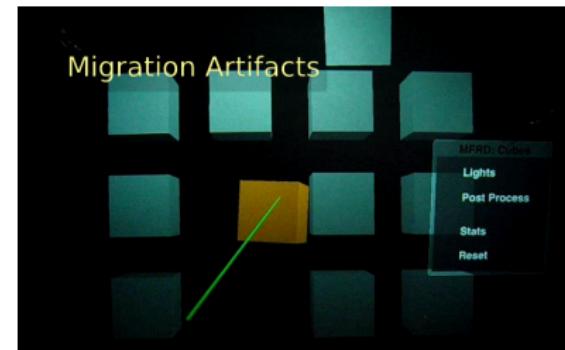
Frame: N-3



Frame: N+3

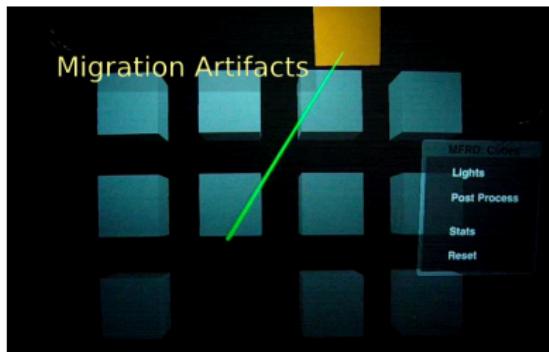


Frame: N



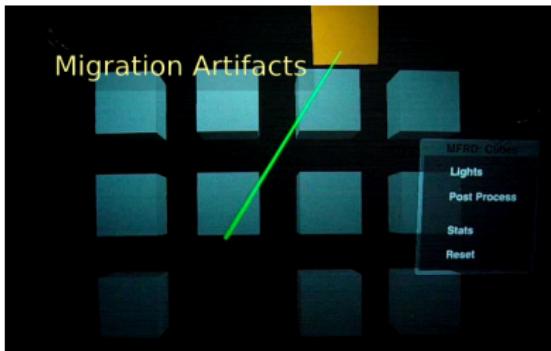
Frame: N+15

# Example: Release Artifact

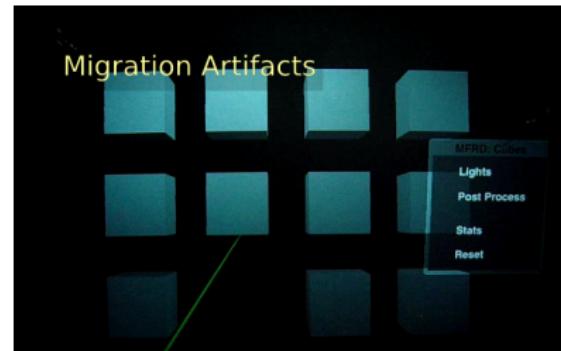


Frame: N

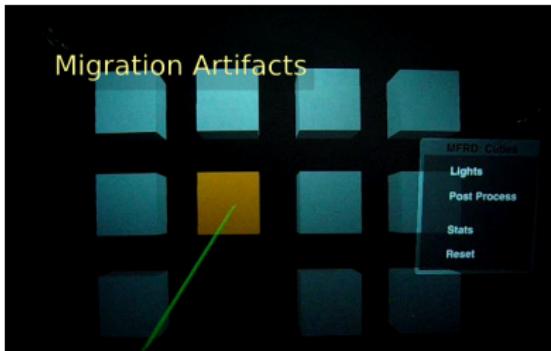
# Example: Release Artifact



Frame: N

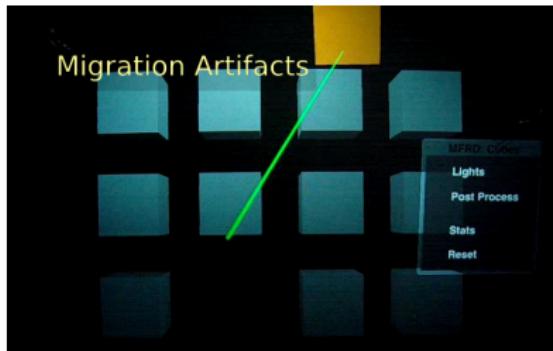


Frame: N+9

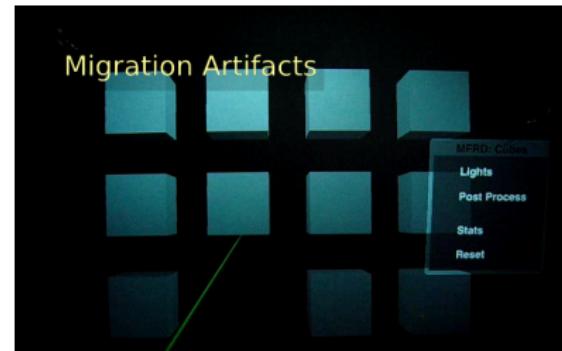


Frame: N+16

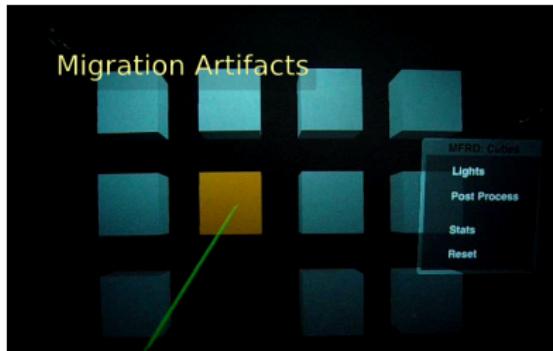
# Example: Release Artifact



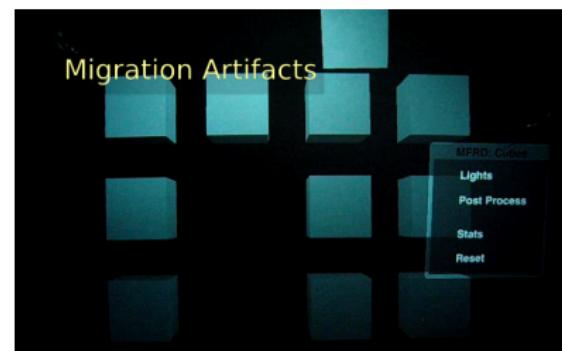
Frame: N



Frame: N+9

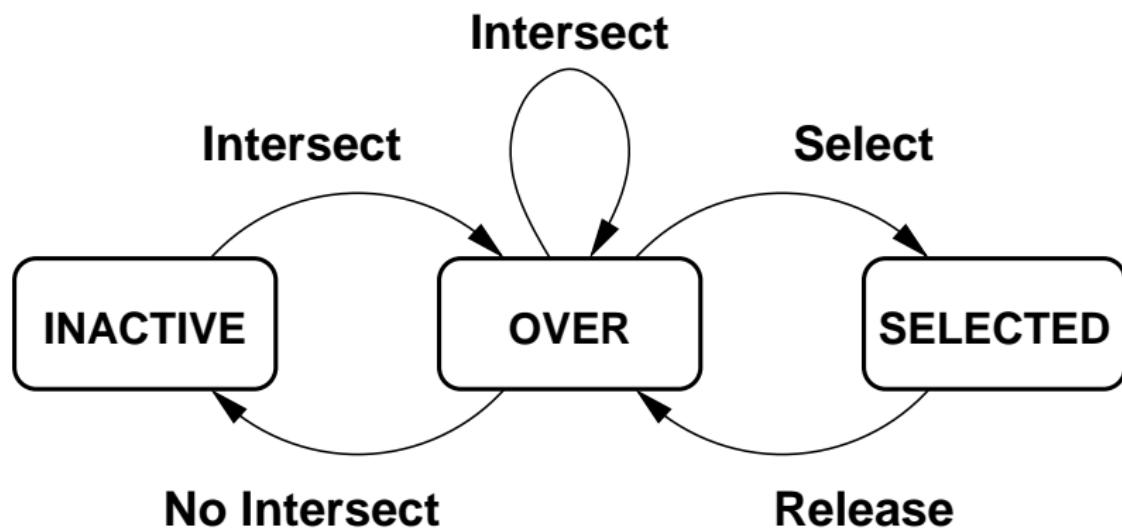


Frame: N+16

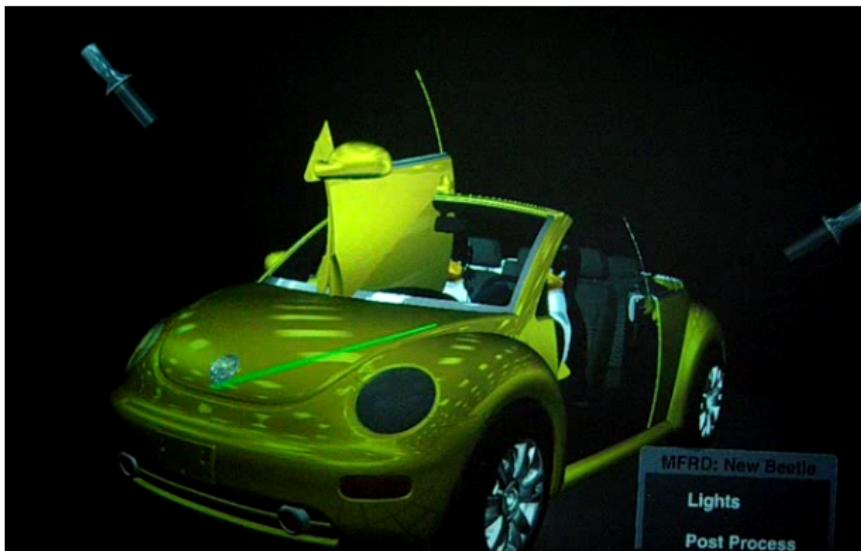


Frame: N+30

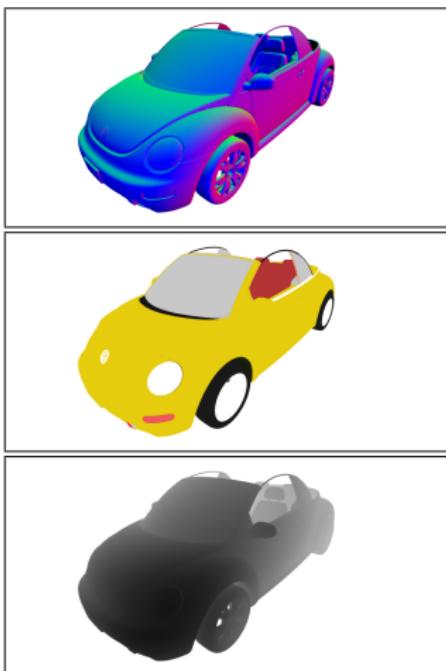
# State Transition Diagram



# No Migration Artifacts for Object Manipulation

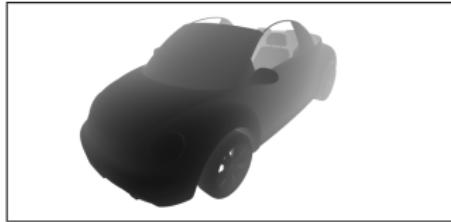
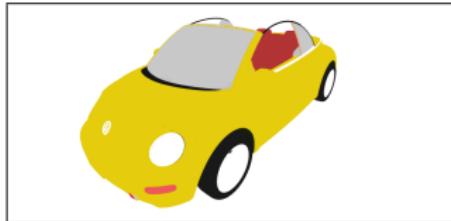
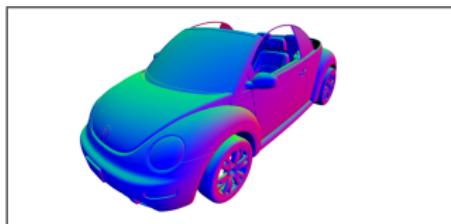


# Interactive Light Manipulation

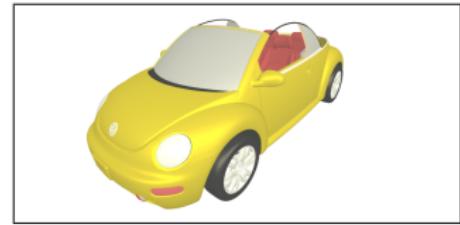


Render Pass #1

# Interactive Light Manipulation

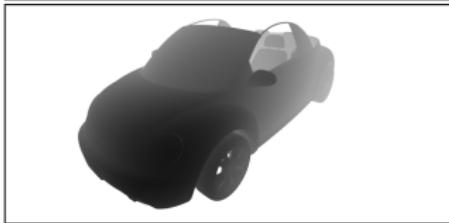
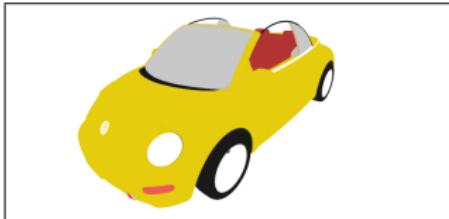
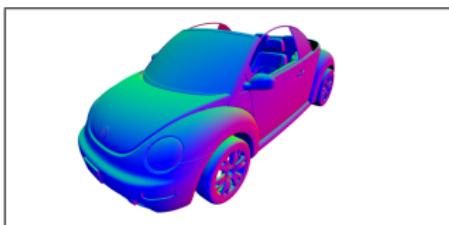


Render Pass #1



Render Pass #2

# Interactive Light Manipulation



**Slow Client**



**Fast Client**

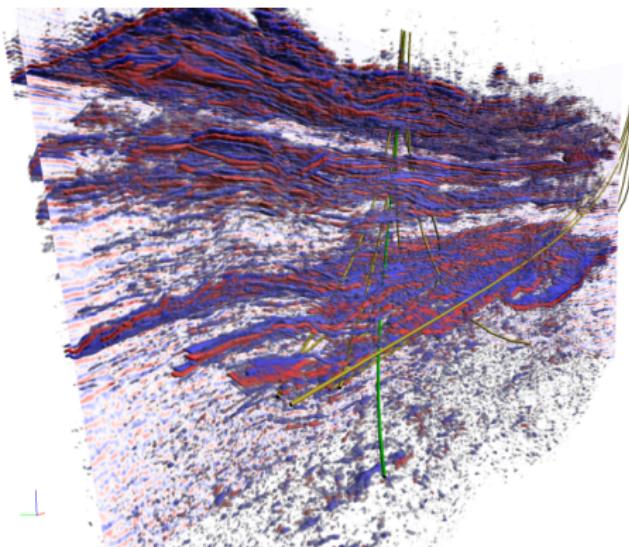
# Interactive Light Manipulation



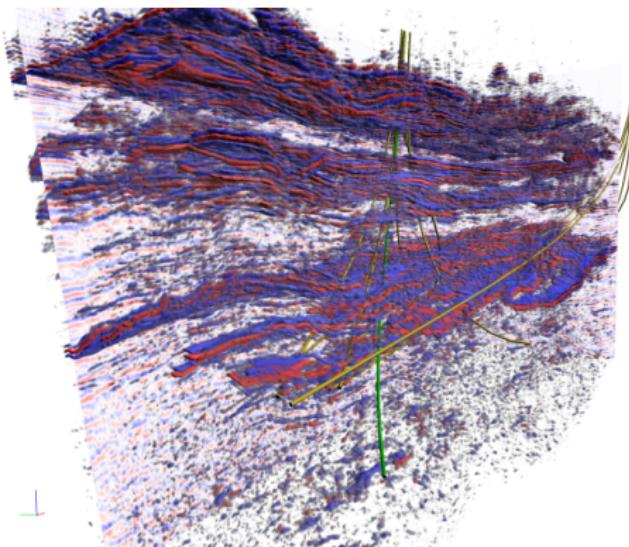
# Properties and Issues

- + Interactive manipulation of lights and light properties
- + Extendable to other scene-global effects (e.g. lighting properties)
  
- Increased memory overhead and bandwidth requirements
- Shifting of computation load ( $SC \rightarrow FC$ )
- Transparent objects (in general) on  $FC$

# Volume Rendering Support



# Volume Rendering Support

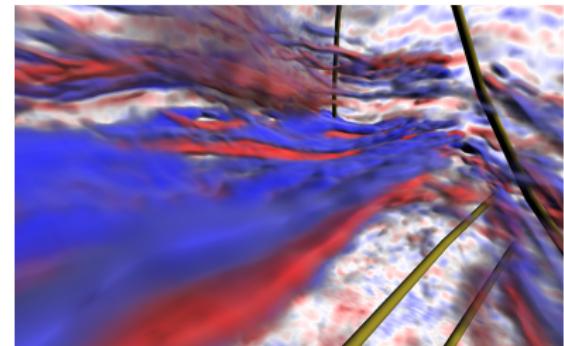


**Goal:** Interactive manipulation of geometric data within a volume

# Participating Buffers



SC: depth (– vol. ray casting)

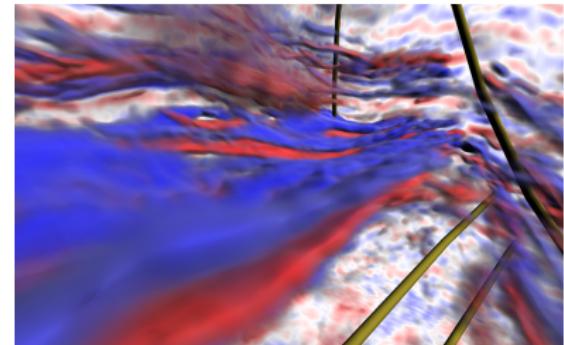


SC: color (+ vol. ray casting)

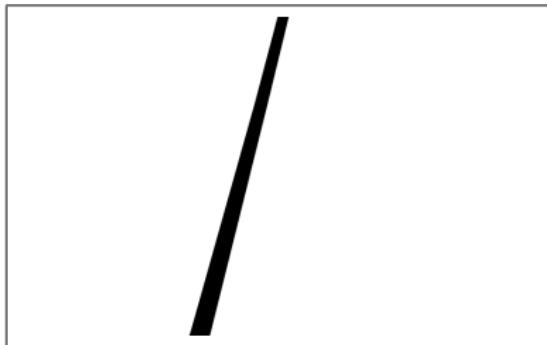
# Participating Buffers



SC: depth (– vol. ray casting)



SC: color (+ vol. ray casting)



FC: stencil

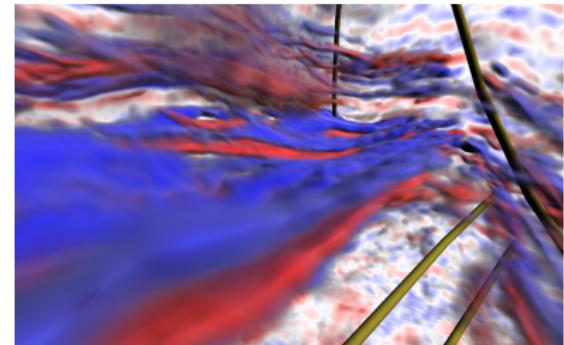
# Participating Buffers



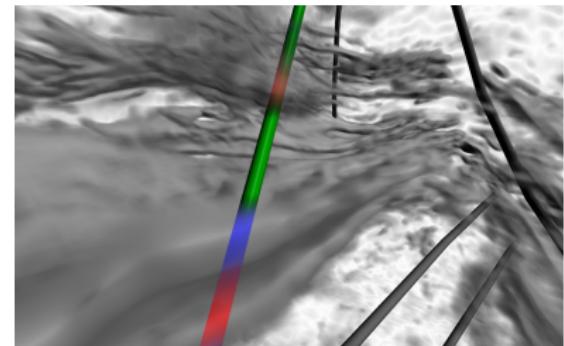
SC: depth (– vol. ray casting)



FC: stencil

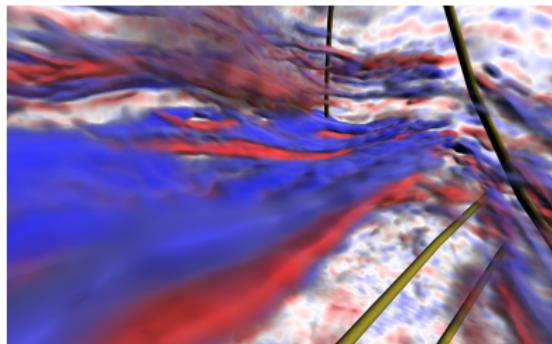


SC: color (+ vol. ray casting)

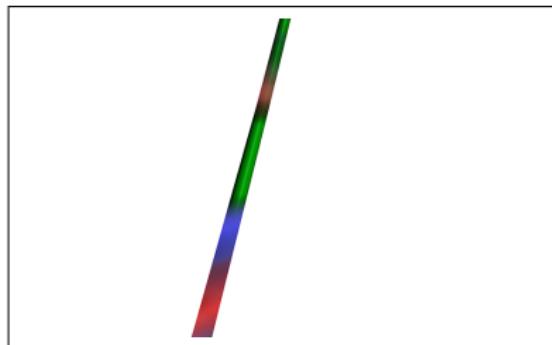


FC: color (+ vol. ray casting)

# Final Composition

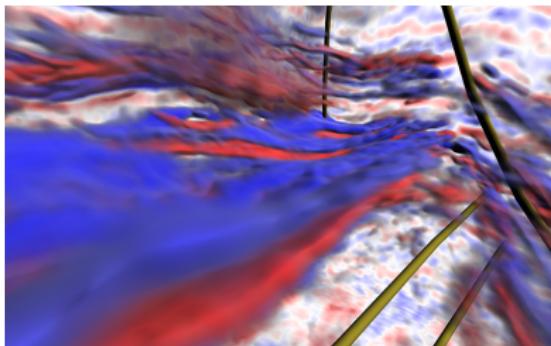


Slow Client

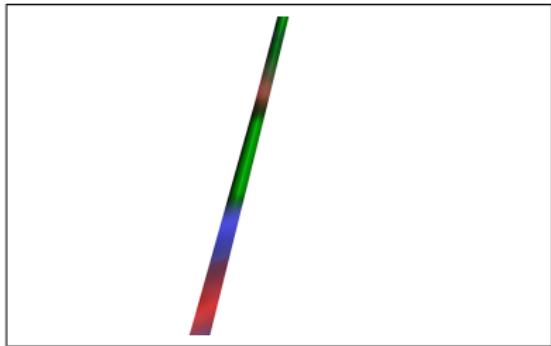


Fast Client

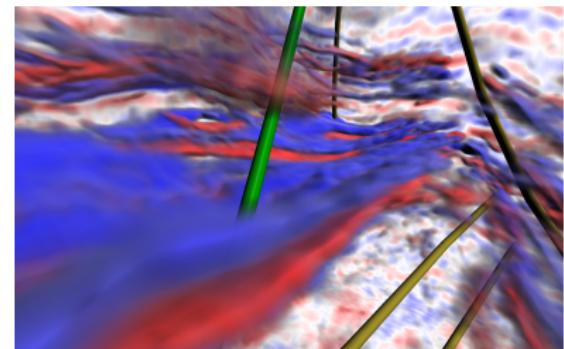
# Final Composition



Slow Client



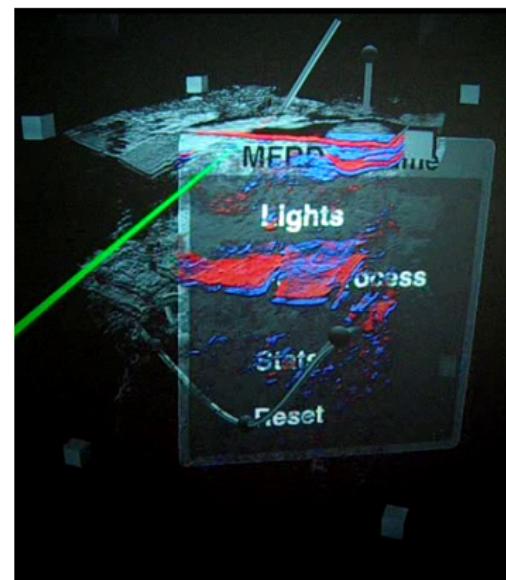
Fast Client



Final Image

# Properties and Issues

- + Interactive manipulation of geometric data within a volume
- + Works for traditional volume rendering, more efficient for volume ray casting
- Performance depends on:
  - ▶ screen-size projection of geometry
  - ▶ penetration depth of geometry within volume



# Summary

- ▶ **Migration artifacts**

can be almost always hidden without introducing synchronization

- ▶ **Interactive light manipulation**

multi-frame rate deferred shading allows interactive manipulation of lights and light properties

- ▶ **Volume rendering support**

combines high-quality volume rendering and interactive object manipulation within a volumetric data set

# Summary

- ▶ **Migration artifacts**

can be almost always hidden without introducing synchronization

- ▶ **Interactive light manipulation**

multi-frame rate deferred shading allows interactive manipulation of lights and light properties

- ▶ **Volume rendering support**

combines high-quality volume rendering and interactive object manipulation within a volumetric data set

# Summary

- ▶ **Migration artifacts**

can be almost always hidden without introducing synchronization

- ▶ **Interactive light manipulation**

multi-frame rate deferred shading allows interactive manipulation of lights and light properties

- ▶ **Volume rendering support**

combines high-quality volume rendering and interactive object manipulation within a volumetric data set

# Future Work

- ▶ Multi-frame rate shadows
- ▶ Multi-frame rate ray tracing
- ▶ Improving navigation support
  - e. g. depth-image warping

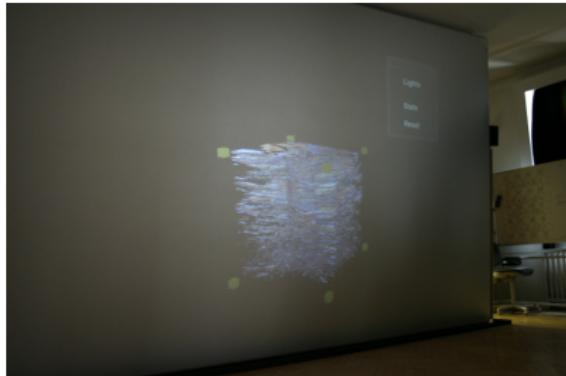
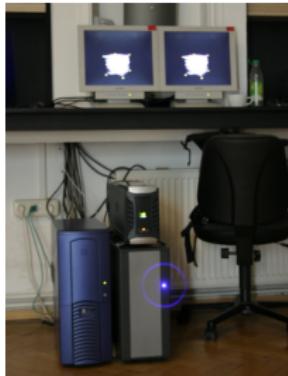
# Future Work

- ▶ Multi-frame rate shadows
- ▶ Multi-frame rate ray tracing
- ▶ Improving navigation support
  - e. g. depth-image warping

# Future Work

- ▶ Multi-frame rate shadows
- ▶ Multi-frame rate ray tracing
- ▶ Improving navigation support
  - e. g. depth-image warping

Thank you for your attention.



# Contents

- Peer-to-Peer Local Network Performance
- ReadPixel/DrawPixel Performance
- End-to-End Latency
- State Transition
- Deferred Shading

# Peer-to-Peer Local Network Performance

[◀ Back](#)

Packet Size in Bytes	Bandwidth in MBit/s	Transfer Time in $\mu$ s	Resolution at 32 Bit Color and 32 Bit Depth
1048576	889.59	8992.90	
1572864	891.94	13453.75	
2097152	892.74	17922.26	
3145728	894.69	26824.90	640 × 512 @ 40 Hz
4194304	895.65	35728.31	
6291456	896.09	53566.20	960 × 768 @ 20 Hz
8388608	896.45	71393.10	
12582912	896.71	107058.40	1280 × 1024 @ 10 Hz
16777216	896.91	142711.80	

- ▶ Cisco Catalyst 3560G, non-jumbo frames
- ▶ NPtcp -I -b 262144 -p 0 -l 1048576 -u 23592960
- ▶ several non-overlapping peers do not degrade performance

# ReadPixel/DrawPixel Performance

[◀ Back](#)

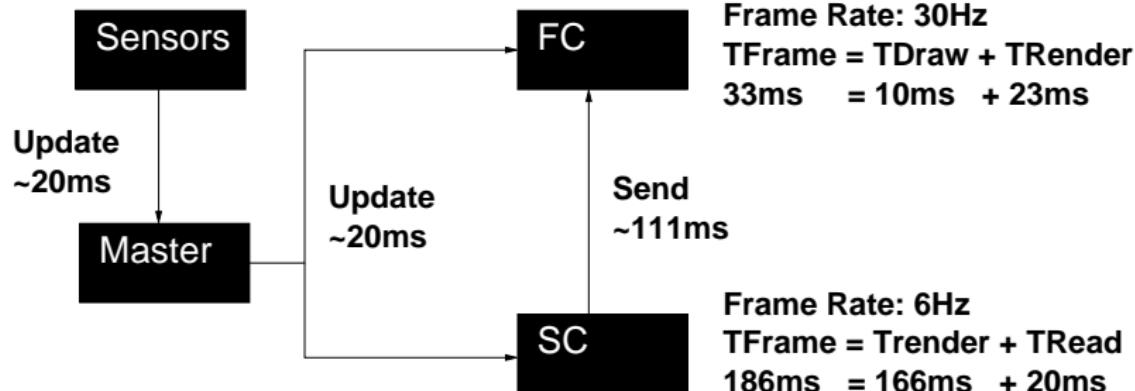
OpenGL: NVIDIA Corporation GeForce 8800 GTX/PCI/SSE2 2.1.0 NVIDIA 97.46

		500 frames at 1280 × 1024			500 frames at 1800 × 1100		
		avg (μs)	MPix/s	MB/s	avg (μs)	MPix/s	MB/s
glReadPixel	RGBA	10071	130.14	520.55	11579	170.99	683.98
	BGRA	5264	248.99	995.98	8447	234.40	937.59
	BGRA_EXT	5260	249.16	996.65	8436	234.70	938.79
	DEPTH	9278	141.27	565.10	10799	183.34	733.34
	STENCIL	3432	381.86	1527.45	5382	367.87	1471.50
glDrawPixel	RGBA	4039	324.50	1297.99	5610	352.91	1411.65
	BGRA	3387	386.89	1547.55	3933	503.42	2013.70
	BGRA_EXT	2519	520.32	2081.26	3656	541.50	2166.01
	DEPTH	4320	303.35	1213.41	5774	342.86	1371.45
	STENCIL	4056	323.09	1292.37	5409	366.04	1464.16

- ▶ `unsigned char` read/draw for RGBA, BGRA\_EXT, and STENCIL
- ▶ `unsigned int` read/draw for DEPTH

# End-to-End Latency

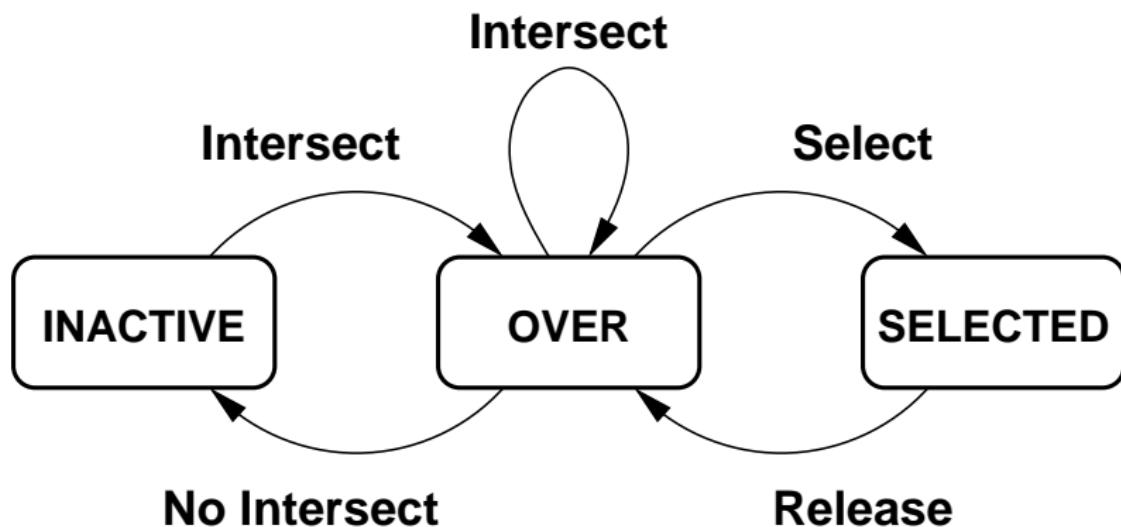
◀ Back



- ▶ Resolution:  $1280 \times 1024$
- ▶ Image compression may not decrease network latency  
see Roth and Reiners, *Sorted Pipeline Image Composition*, EGPGV06

# State Transition

◀ Back



# Deferred Shading

◀ Back

- ▶ digital composition:  
color: 32bits ( $4 \times 8\text{bit}$ )  
depth: 32bits ( $1 \times 32\text{bit}$ )  
 $\rightarrow x\text{-res} \times y\text{-res} \times \text{eyes} \times 8\text{byte}$   
 $\rightarrow 1280 \times 1024 \times 2 \times 8\text{byte} = 20\text{Mbyte}$
- ▶ deferred shading:  
normal: 128bits ( $4 \times 32\text{bit}$ )  
depth: 32bits ( $1 \times 32\text{bit}$ )  
ambient: 32bits ( $4 \times 8\text{bit}$ )  
diffuse: 32bits ( $4 \times 8\text{bit}$ )  
specular: 32bits ( $4 \times 8\text{bit}$ )  
 $\rightarrow x\text{-res} \times y\text{-res} \times \text{eyes} \times 32\text{byte} = 80\text{Mbyte}$
- ▶ digital composition : deferred shading = 1 : 4

This slide intentionally left empty.