

Interactions Lab
University of Calgary



A Buffer Framework for Supporting Responsive Interaction in Information Visualization Interfaces

Tobias Isenberg

André Miede

Sheelagh Carpendale



Main Message

The **buffer framework** provides faster and more responsive interaction for **large displays** with many objects and multiple users.



Overview

- Introduction & Motivation
- Buffer Concept
- Realization
- Applications
- Implementation & Results
- Summary & Future Work

Introduction and Motivation



application courtesy of Uta Hinrichs [Hinrichs et al., 2005]

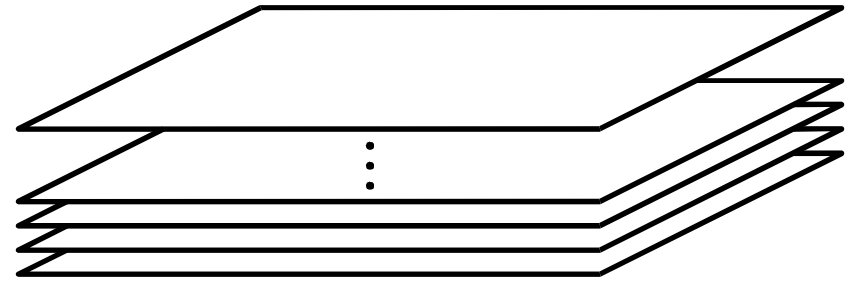


Complexities and Limitations

- number of objects
- complexity of interaction between objects and controlling structures
- complexity of interaction between several controlling structures
- simultaneous user-interactions
- ▶ complexity in run-time and development

Buffer Concept: Borrowing from Computer Graphics

- buffers as means to store data in computer graphics (e.g., z-buffer, G-buffers)

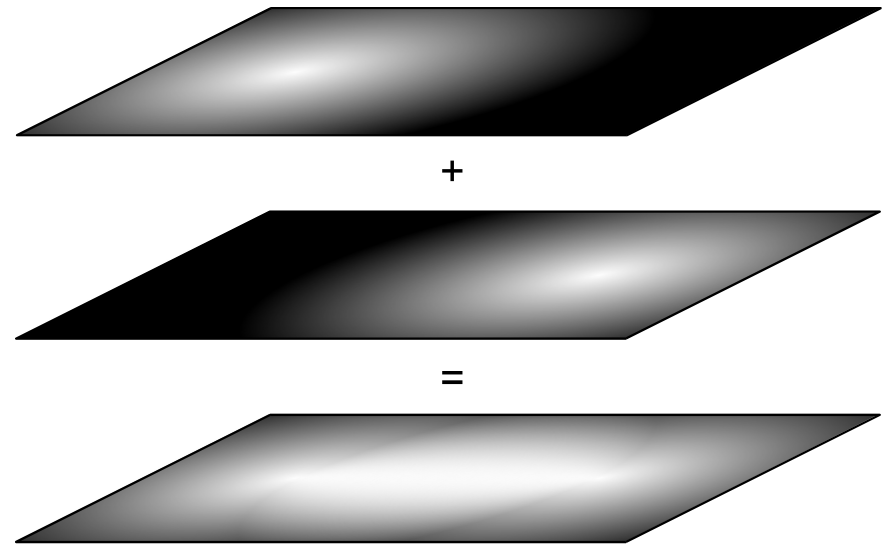


- ▶ **property sampling** on regular grid
- ▶ **fast lookup** of discrete values
- ▶ **interpolation** for smooth animations

Buffer Concept: Borrowing from Physics

- composition of multiple effects
- treat buffers as physical (force) fields

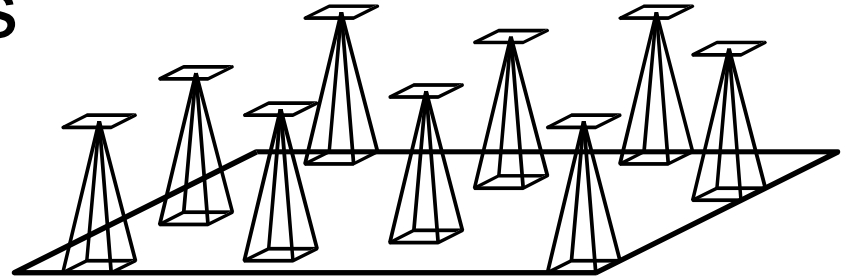
- example:
composition of
intensity buffers
→ treat as fields



- simple buffer adding
- simpler “interface physics”

Buffer Concept: Borrowing from Swarm Intelligence

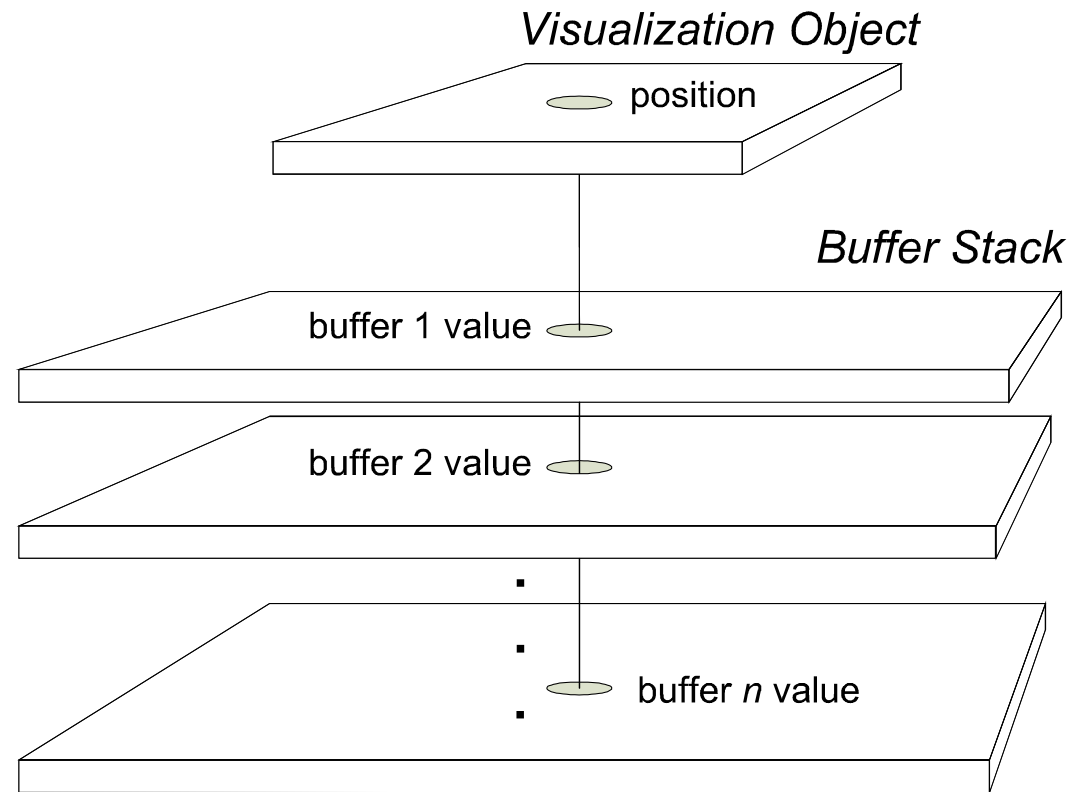
- single swarm entities
 - local awareness
 - local processing
-
- ▶ divide-and-conquer strategy
 - ▶ local aspect of entities & local character of buffers



Realization of the Framework

- *visualization objects* carry information

- *buffer stack* for several properties of interface

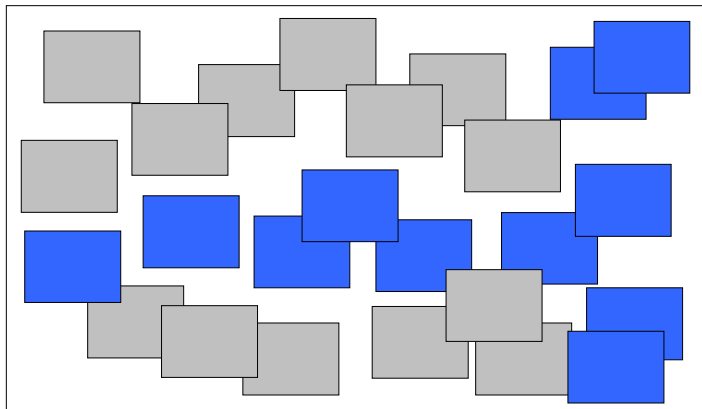


- ▶ buffer contents controls **object behavior**

Examples for Buffer Control

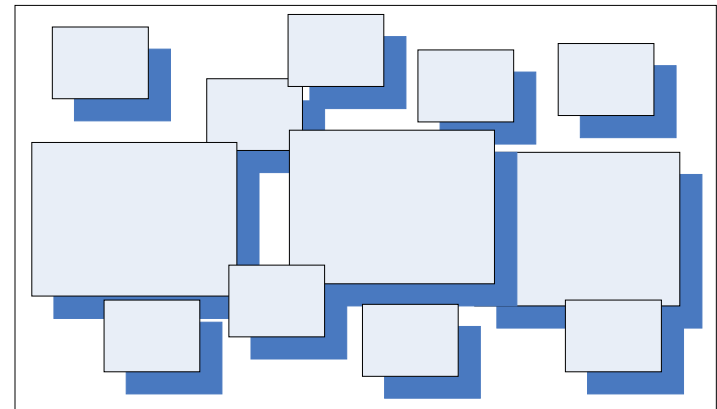
color

0	0	0	1
1	1	1	1
0	0	0	1



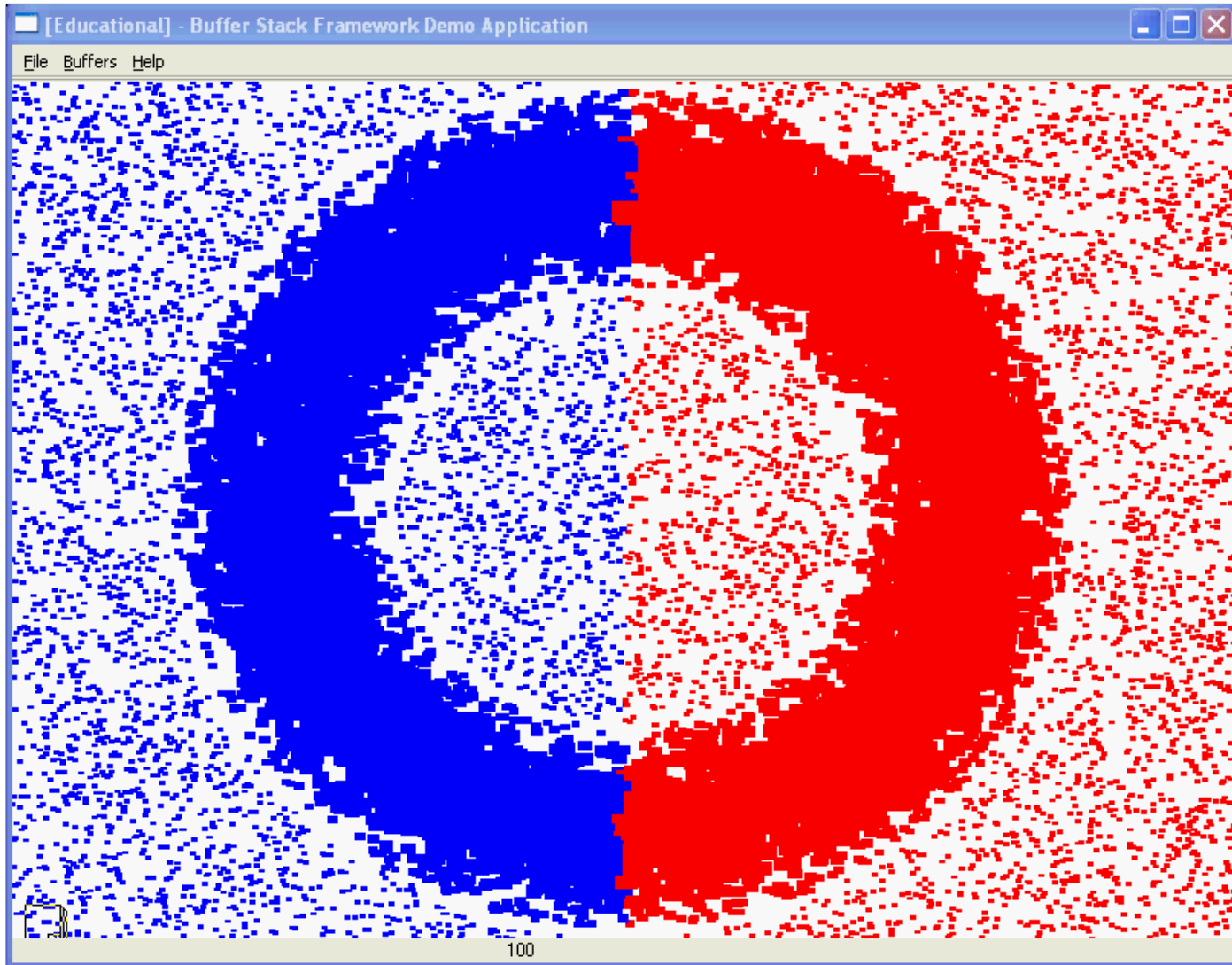
size

1	1	1	1
2	2	2	2
1	1	1	1



- resolution of buffers & display independent

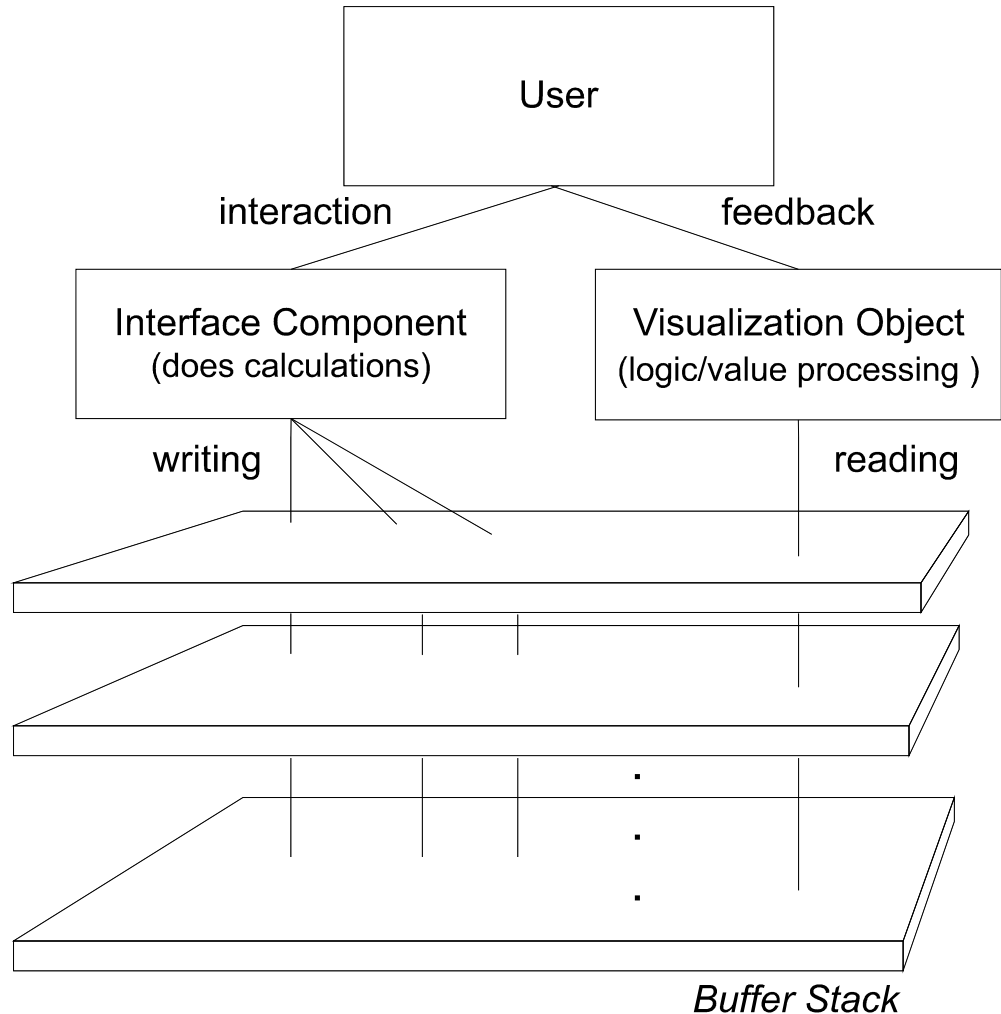
Examples for Buffer Control



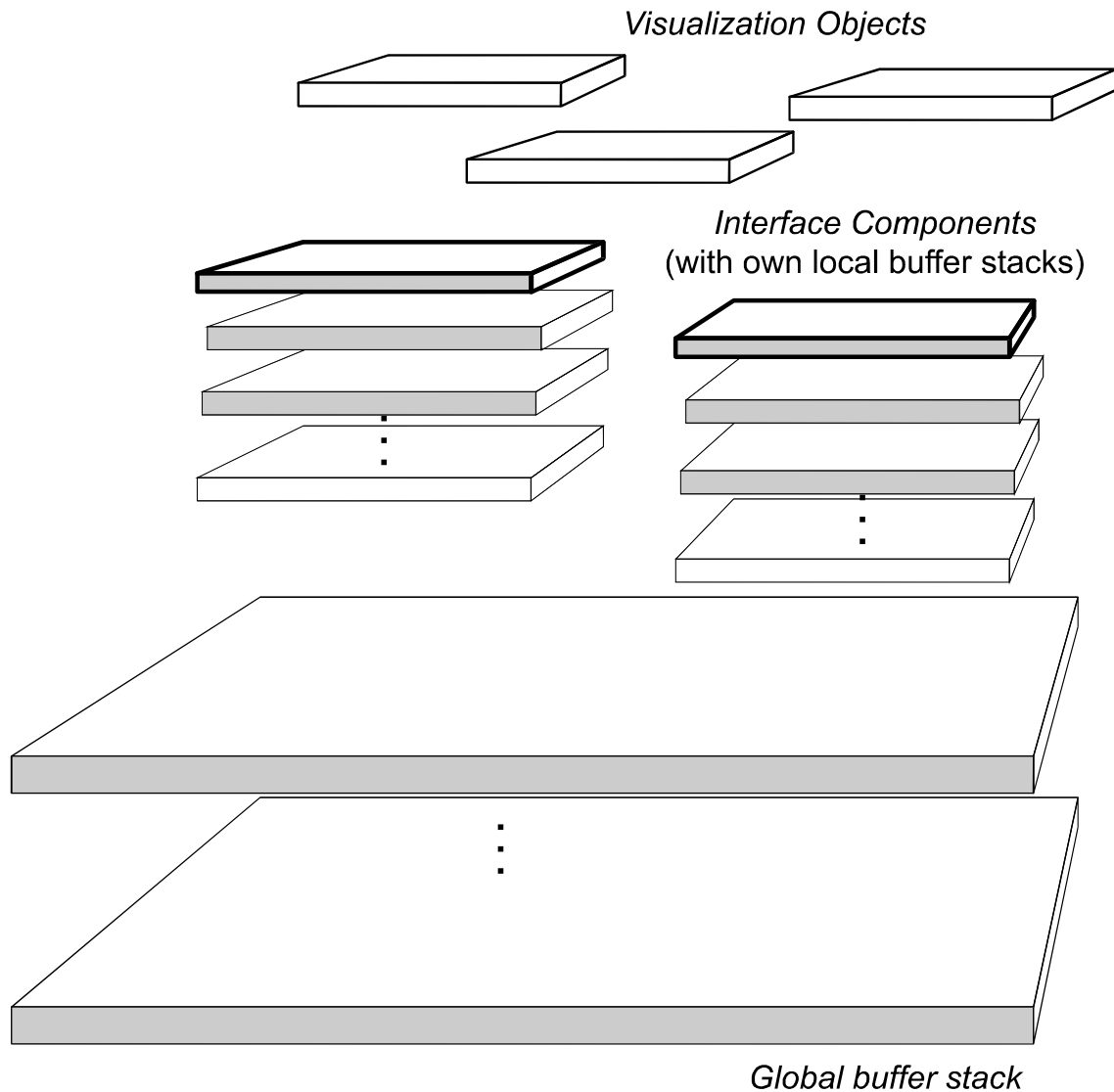
Realization of the Framework

- *interface components* organize objects

- interface logic:
interface components
↓
visualization objects



Final Framework Layout



Applications

- support of responsive interfaces on large displays
- extension of systems to support many more objects
- seamless integration of input support



Applications





Implementation and Hardware

- OpenGL + Trolltech Qt
- hardware support for rendering tasks
- SMART DViT: two concurrent inputs

Table Setup 1: $1280 \times 2048 \approx 2.6 \text{ Mp}$



Table Setup 2: 2800 x 2100 \approx 5.9 Mp



Results



1280x2048
2.6 Mp

ca. 100
objects @
25-30 fps

1000
objects @
25-30 fps



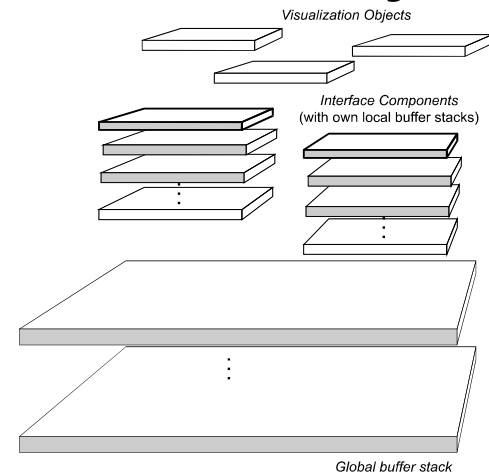
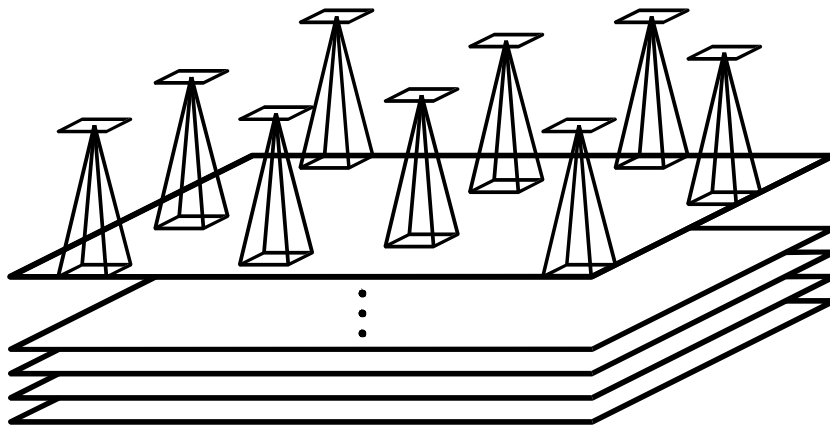
2800x2100
5.9 Mp

ca. 100
objects @
5-10 fps

200/400/1500
objects @
30/20/7.5 fps

Summary

- framework for responsive interaction
- speed gains through storing data in buffers
- sampled, discrete values fast to look up
- local awareness and local processing
- application logic in autonomous objects



Future Work

- continue development of prototype and its comparison with previous applications



- explore new application domains that take full advantage of the buffer framework



Main Message

The **buffer framework** provides faster and more responsive interaction for **large displays** with many objects and multiple users.

Thanks for your attention!

Thanks to our
funding agencies
and sponsors:

