GAME EVENT LENSES

Focus+Context Visualizations of Computer Game Data

Lars Schütz
l.schuetz@inf.hs-anhalt.de

October 1, 2015

Anhalt University of Applied Sciences, Department of Computer Science and Languages
Otto von Guericke University Magdeburg, Faculty of Computer Science

OUTLINE

BACKGROUND

MOTIVATION AND PERSPECTIVES ON GAMES

- Games are played everywhere, at anytime by everybody

Developers

- Trends and preferences
- Player behavior in solo and group activities
- Further game adjustments, e. g., balance, bugs, content
- Making profit

Players

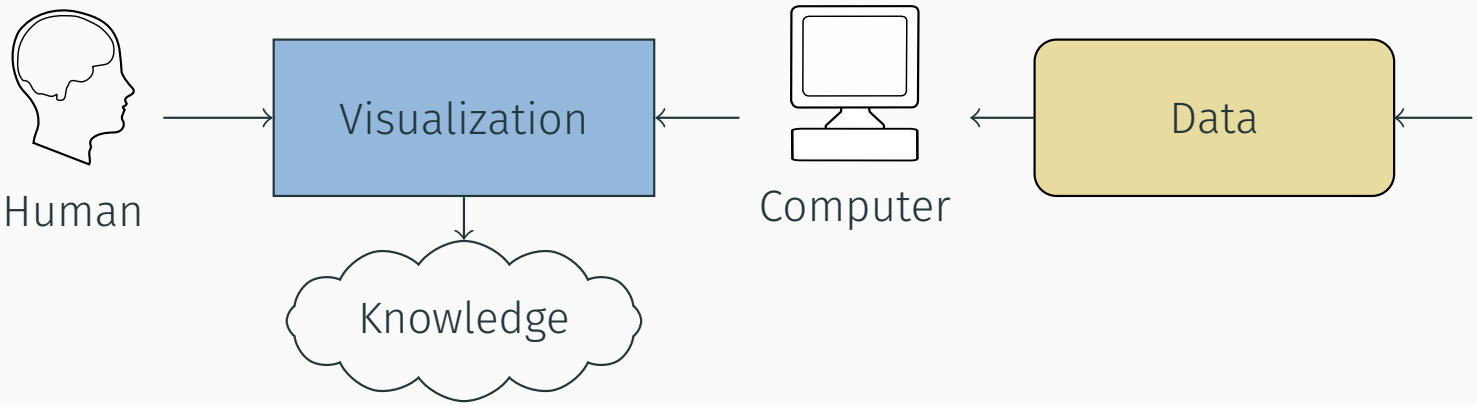
- Part of a competitive environment, e. g., e-sports tournaments
- Interest in best strategies, tactics and performances

$\rightarrow$ Facts and room for improvement in data $\rightarrow$ Look at data

CHALLENGES

- Huge amount of data
- Complex and time-consuming data processing
- Extraction and communication of usable information
- Gain insight into data

→ Visualization as an interface for data exploration



GAME EVENTS

COMPONENTS

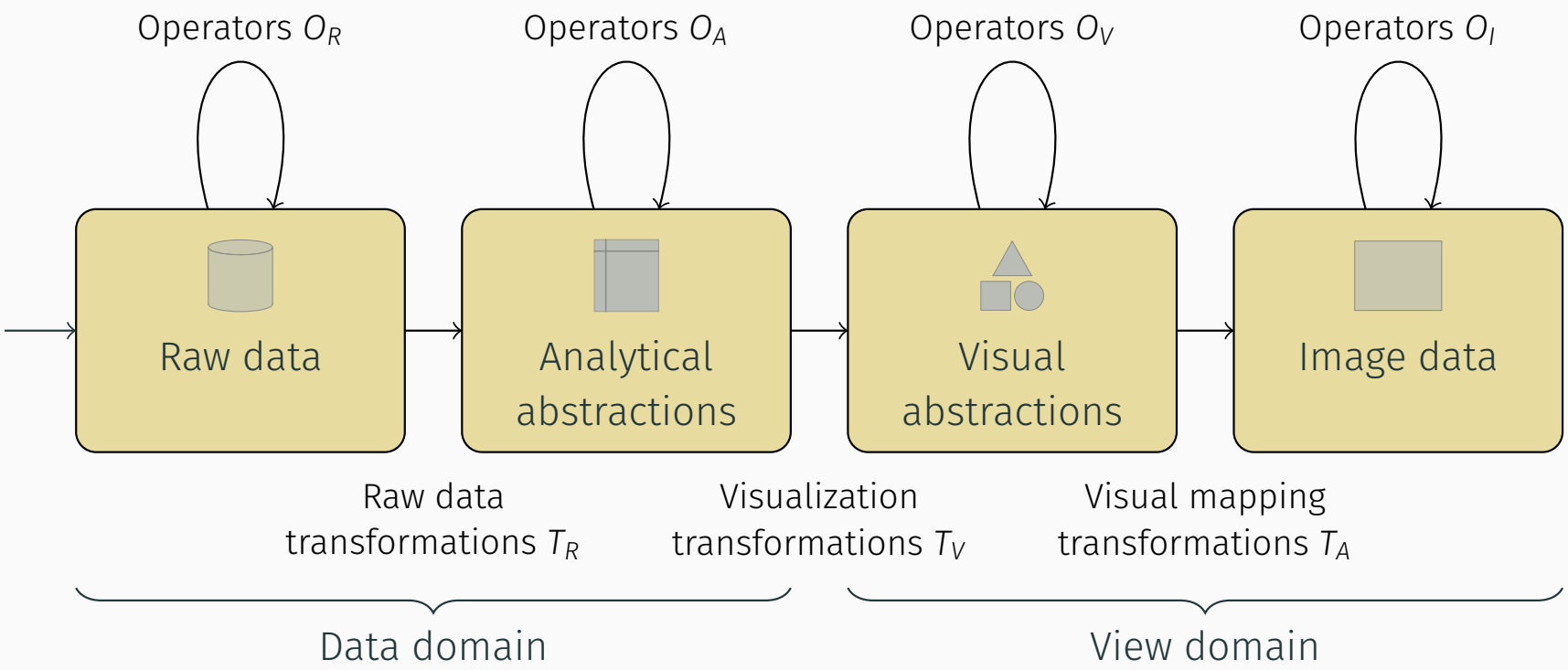
Event type	The event's class
Timestamp	Point in time the event happened
Spatial data	Several spatial information
Sender	Information about the event's trigger
Receiver	Data of the affected game related entity
Type data	Specific information based on the event type

EXAMPLE

Component	Gameplay game event
Event type	Attack
Timestamp	2015-03-07T23:01:14Z
Spatial data	Area: 23, x: 234, y: 102, z: 20
Sender	Player: 3, level: 80, class: 4
Receiver	Player: 5, health points: 34;
	Player: 14, health points: 56
Type data	Damage to player 5: 11;
	Damage to player 14: 5

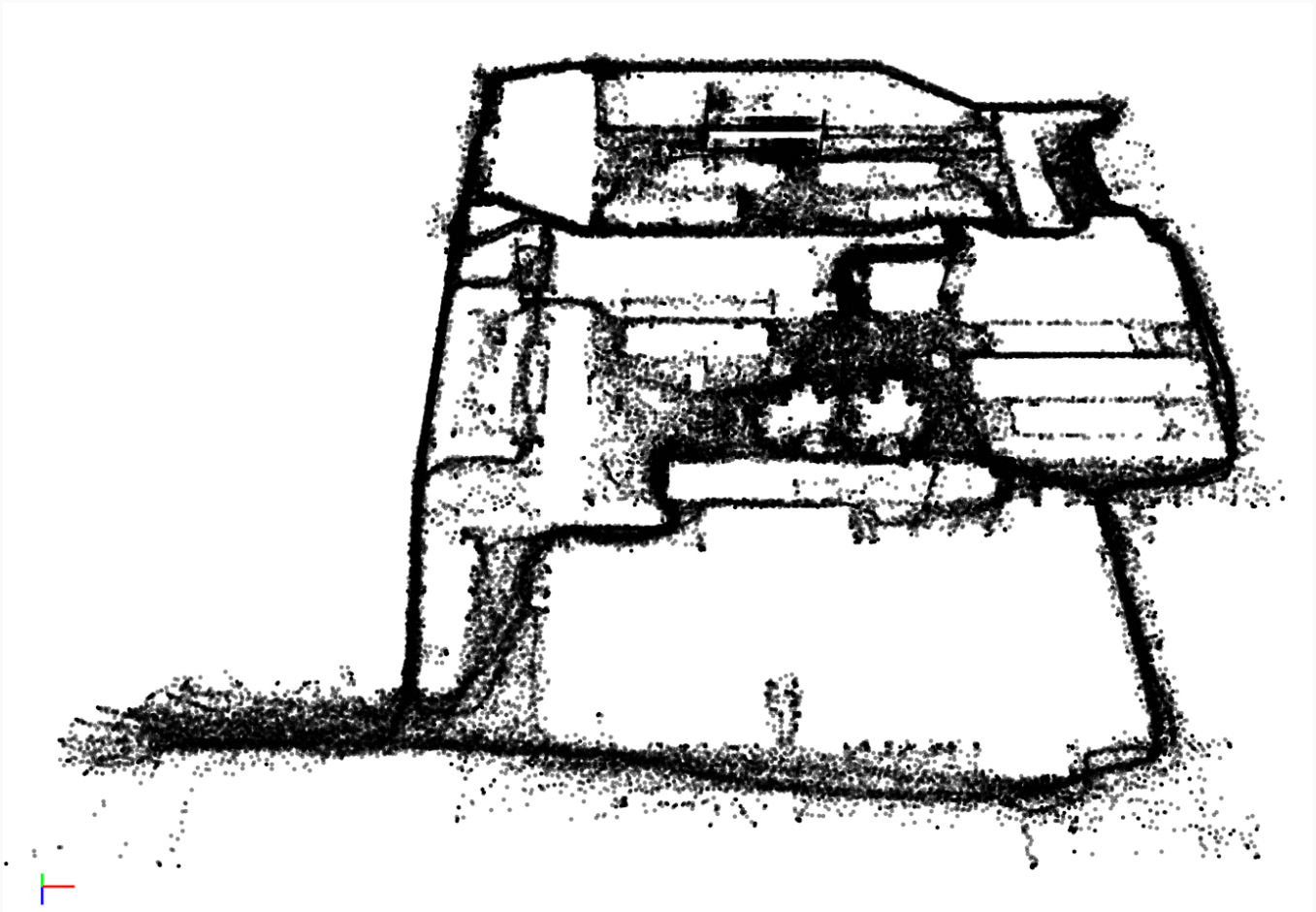
DATA STATE REFERENCE MODEL AND LENSES

DATA STATE REFERENCE MODEL



EXAMPLE

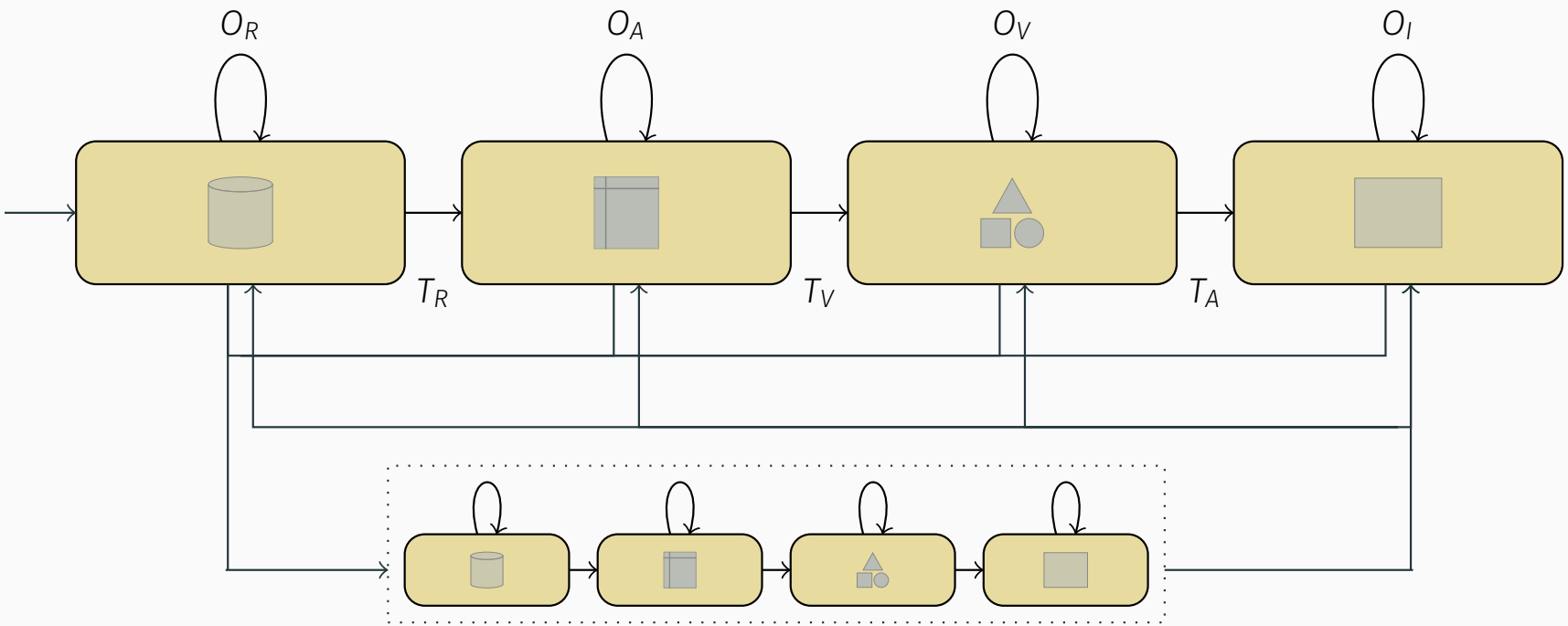
- CS:GO – Position reports of all players



LENSES

- General or basic information to give an overview (context)
- Detailed or other information in region of interest (focus)
- Lens properties position, shape, size and orientation
- Lens function controls the focus effect

LENS INTEGRATION



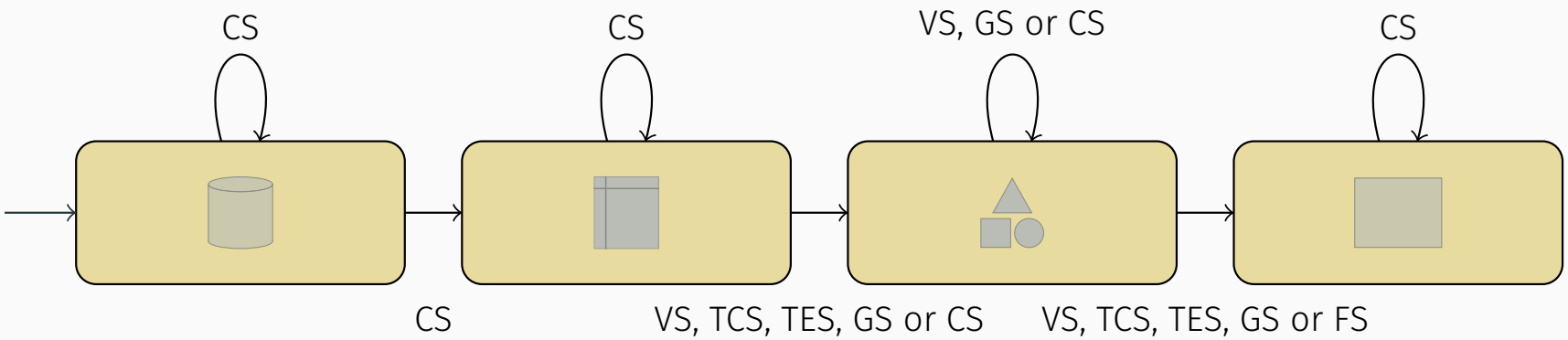
1. Create + render the context visualization, i. e., omit lens region
2. Create + render the focus region
3. Optionally render the lens itself, e. g., by drawing its boundary

REALIZATION OF THE APPROACH

GRAPHICS SYSTEM OF GAME EVENT LENSES

- OpenGL (pipeline) is foundation
- Shader programs simulate operators of the DSRM
- “Traditional shaders” mainly for view (graphical) domain
- Compute shaders mainly for data (semantic) domain
- Raw data coherently packed in SSBOs

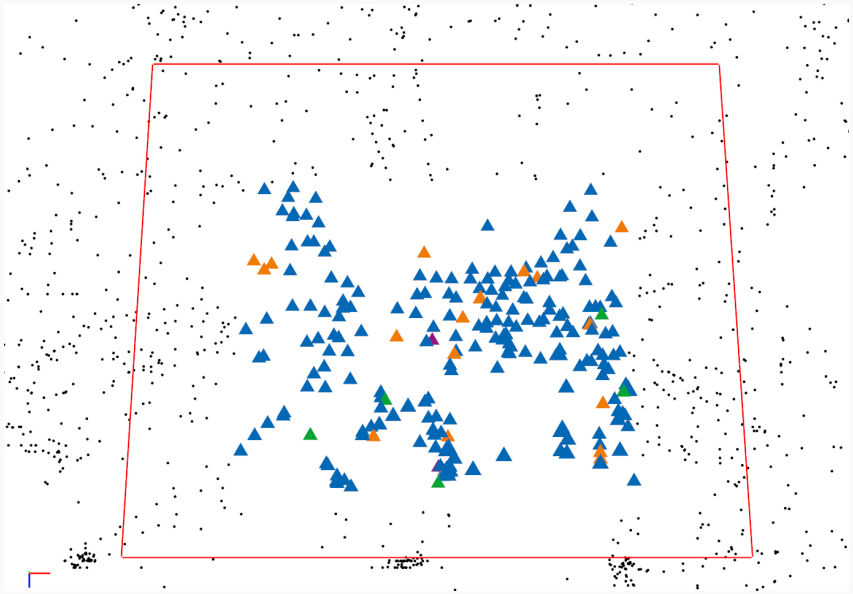
$$(v_a^e) = (v_1^1, v_2^1, \ldots, v_n^1, v_1^2, v_2^2, \ldots, v_n^2, \ldots, v_1^m, v_2^m, \ldots, v_n^m)$$



EXAMPLE

- CS:GO – Weapon Category Usage for One Team

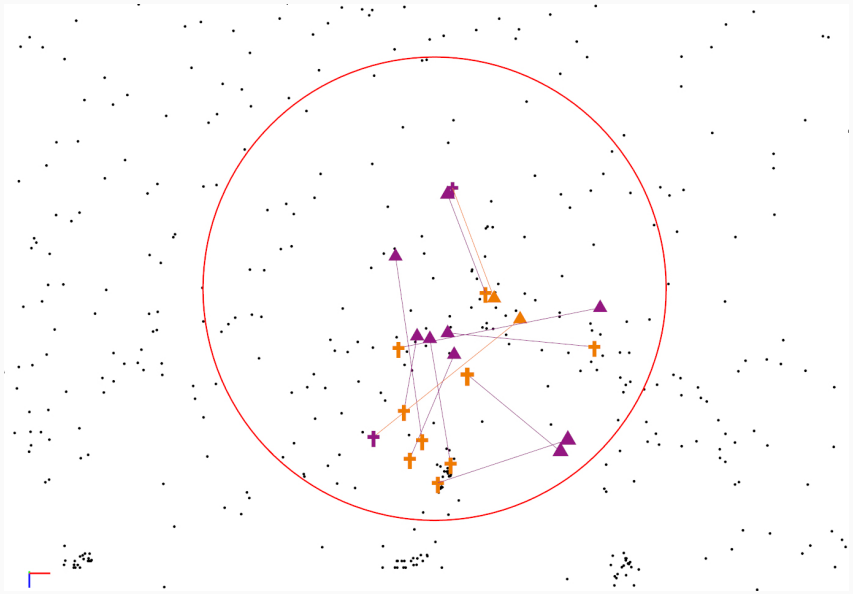
O	S	Usage
T_R	CS	Event pass-through
T_V	VS	Position, color
O_V	GS	Points $\rightarrow$ Triangles
T_A	FS	Color pass-through



EXAMPLE

- CS:GO – Virtual Kills and Deaths of Players

O	S	Usage
T_R	CS	Event pass-through
T_V	VS	Position, color
O_V	GS	Cross or triangle
T_A	FS	Color pass-through
T_V	VS	Position, color
O_V	GS	Points $\rightarrow$ Lines
T_A	FS	Color pass-through



CONCLUSION AND FUTURE WORK

SUMMARY AND FURTHER DIRECTIONS

- Game Event Lenses base on the DSRM
- Shader-based approach for GPU-accelerated operators
  (combination with CPU-based operators not excluded)
- Game event instances are mapped to geometric primitives
- Working approach for first visual data exploration
  (in fact not restricted to game events)

- Design and combination of operators
- Overall visualization design
- Visual clutter
- Lens interaction

Thank you very much!
Questions?

Lars Schütz
l.schuetz@inf.hs-anhalt.de