DynCam: A Reactive, Distributed Point-Cloud Pipeline

Christoph Schröder
University of Bremen, Germany
cgvr.informatik.uni-bremen.de

VRIC, April 2018
Telepresence

Motivation

Methods

Results

Conclusion
Telepresence

Motivation

Methods

Results

Conclusion

Altspace VR
Vision
Collaborative Virtual Environments

Motivation

Methods

Results

Conclusion
Our Longterm Goal

- Requirements
  - Realistic visualization
  - Shared virtual environment
  - Precise interaction

- Technologies
  - Multiple 3D cameras
  - Merge point clouds
  - Realistic rendering
Related Work

- Fusion Kit\(^1\)
- Fusion4D\(^2\)
- ROS (Robot Operating System)\(^3\)

\[1\] Michael Rietzler, et al. 2016. *FusionKit*. ACM SIGCHI Symp. 8 - EICS '16
Our Contribution

- Framework for distributed point cloud processing
  - Easy-to-extend C++ pipeline
  - Implicit parallelism
  - Different camera sources
  - Platform independent
- Basic visualization in Unreal Engine 4
- Improved end-to-end latency measurement method
Reactive Architecture

- Updates
  - Automatic
  - Lazy
  - Implicitly parallel

*ACM SIGPLAN international conference on Functional programming - ICFP ’97*
Reactive Architecture

Motivation

Methods

Results

Conclusion
Reactive Architecture

Motivation

Methods

Results

Conclusion
Reactive Architecture

Motivation

Methods

Results

Conclusion
Reactive Architecture
Unreal Engine Integration

Motivation

Methods

Results

Conclusion
Unreal Engine Integration

**Motivation**

**Methods**

**Results**

**Conclusion**

Vertex shader

\[
X = 0 \\
Y = 0 \\
Z = i_d, \ i_d = 0, \ldots, N-1
\]

Point \((X,Y,Z)\)

32 bit texture

Index using z-coordinate

\[(X,Y,Z)\]
Latency Measurement

- Based on method by Anthony Steed\textsuperscript{4}

Latency Measurement

- Converted to Python
- Extract LEDs with OpenCV blob detection
- Outlier removal (moving average)
- Estimate frequency and phase angle with fft
- Optimize via curve fit
  \[ y(x) = A \sin(2\pi f x + \Phi) \]
- Latency \[ \Delta t = \frac{\Delta \Phi}{2\pi f r} \]
Latency

Latency in ms

Protonect 2D
DynCam 2D

Configuration
Latency

![Graph showing latency comparison between Protonect 2D and DynCam 2D]

**Configuration**
Latency

![Box plot comparing latency in ms for different configurations: Protonect 2D, DynCam 2D, MSKinect 2D, DynCam local, ROS local. The graph shows the distribution of latency values for each configuration.]
Latency

![Box plot showing latency in ms for different configurations: Protonect 2D, DynCam 2D, MSKinect 2D, DynCam local, and ROS local. The plot displays the distribution of latency with quartiles and outliers.](image-url)
Latency
Latency

![Box plot showing latency in milliseconds for different configurations. The box plot includes configurations like Protonect 2D, DynCam 2D, MSKinect 2D, DynCam local, ROS local, DynCam LAN uncompressed, DynCam LAN compressed, and ROS LAN uncompressed. The latency values range from 100 to 200 ms, with the following configurations having specific latency values: Protonect 2D (773 Mbps), DynCam 2D (314 Mbps), MSKinect 2D (901 Mbps).]
Conclusion

- Distributed point cloud library
  - Platform independent
  - Implicit parallelism
  - Easy to extend
  - 30% lower latency than ROS
- Unreal Engine integration
- Improved latency measurements
- Code
Conclusion and Future Work

- Enhanced rendering as splats or surface
- Improved compression
- Temporal resolution enhancement
- Automatic VR/AR alignment
Thank you!