SLAMCast: Large-Scale, Real-Time 3D Reconstruction and Streaming for Immersive Multi-Client Live Telepresence

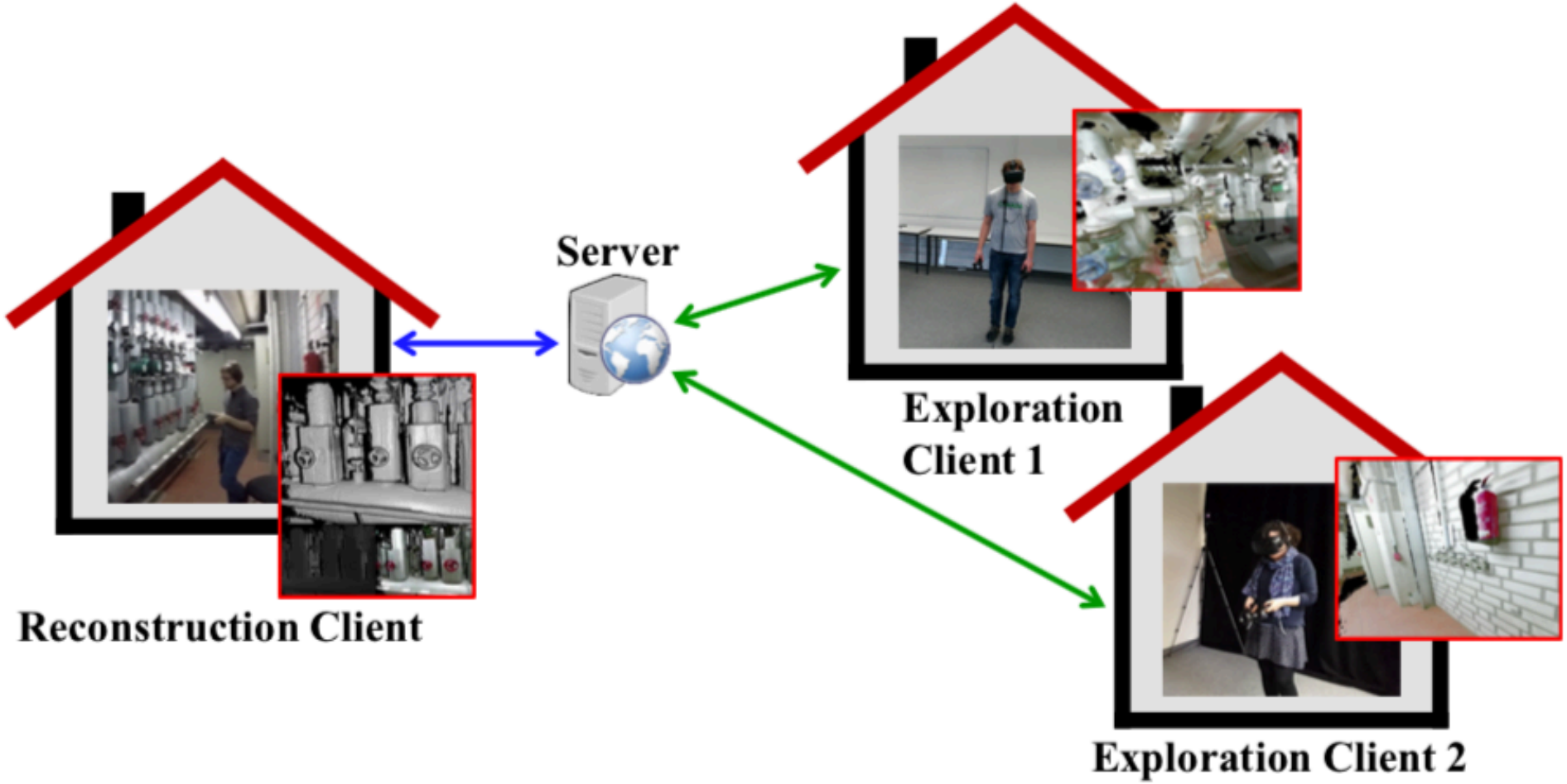
Patrick Stotko, Stefan Krumpen, Matthias B. Hullin,
Michael Weinmann, Reinhard Klein

eprint arXiv:1805.03709 05/2018

Contribution

1. Efficient 3D reconstruction for multi-client telepresence
2. The first thread-safe GUP hash map
   - Concurrent retrieval, insertion, removal entries
3. Low bandwidth remote connection
4. Novel scene representation
   - Reconstruct geometry at exploration client
5. Lightweight projective texture map
   - Overcome limited resolution of voxel-based scene representation

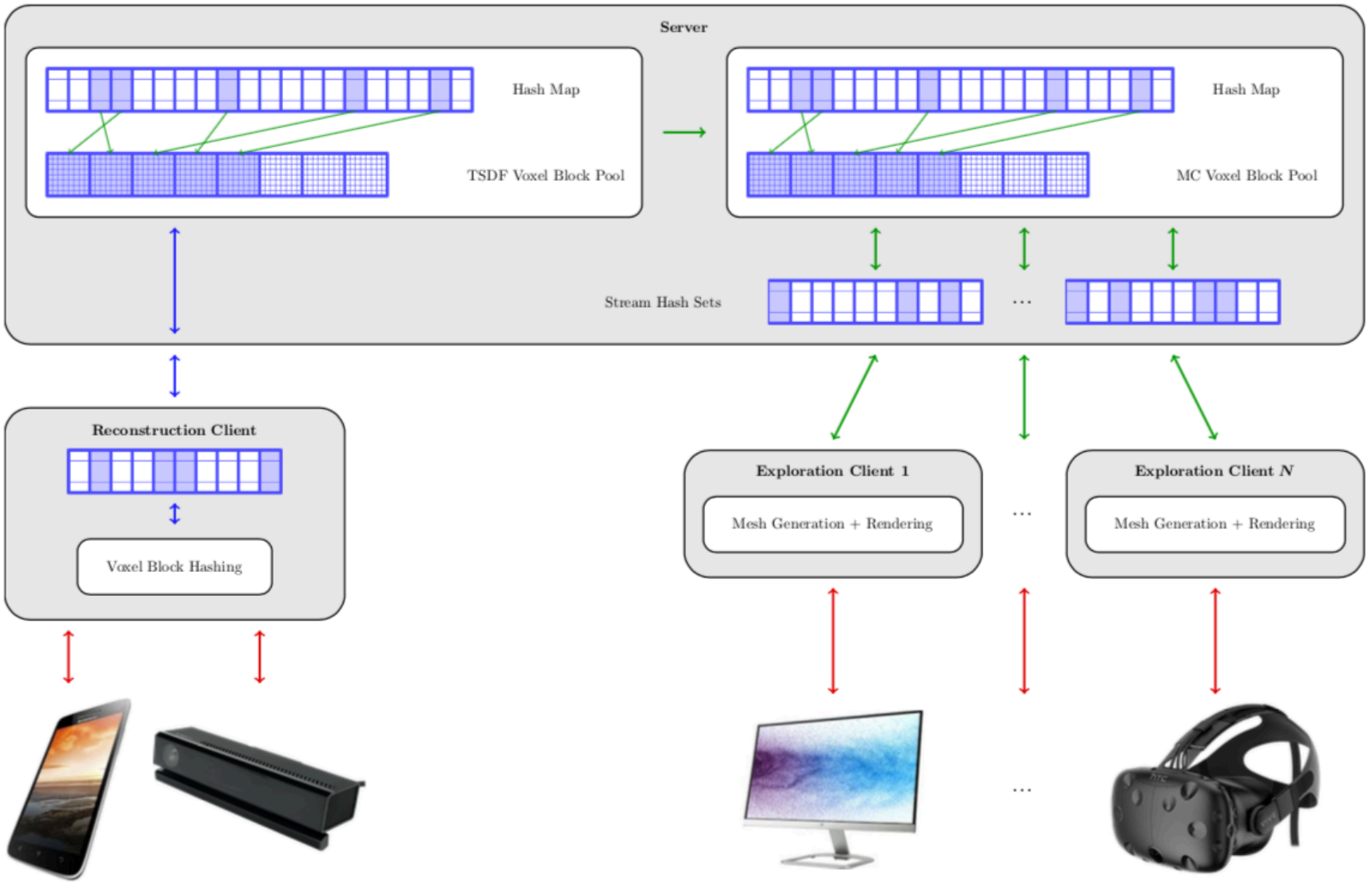
Block Diagram



Design Choices

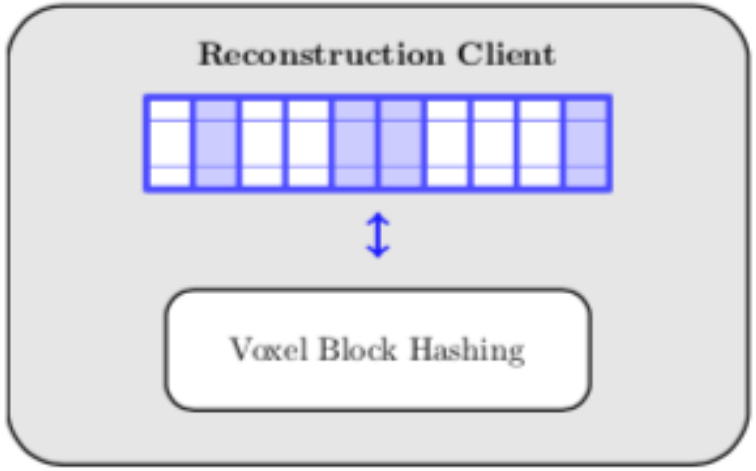
- Collaboration task
  - Users interact with captured scene and observe the other client's interactions
  - Central server is placed between individual clients
- Interactive exploration
  - Transmit RGB-D input sequence
  - Reconstruct scene model at the exploration client's site
  - Stream parts of the fused model independently from the acquisition order
- Hash Data Structure
  - Hash map supports concurrent insertion, removal, retrieval including key preservation -> high reliability

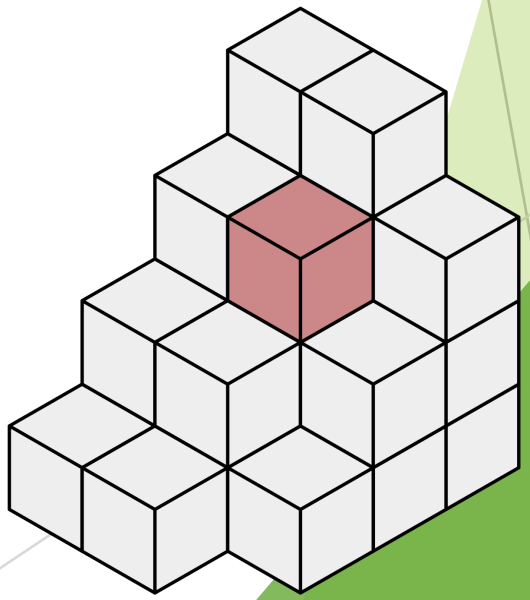
Remote Collaboration System



Reconstruction Client

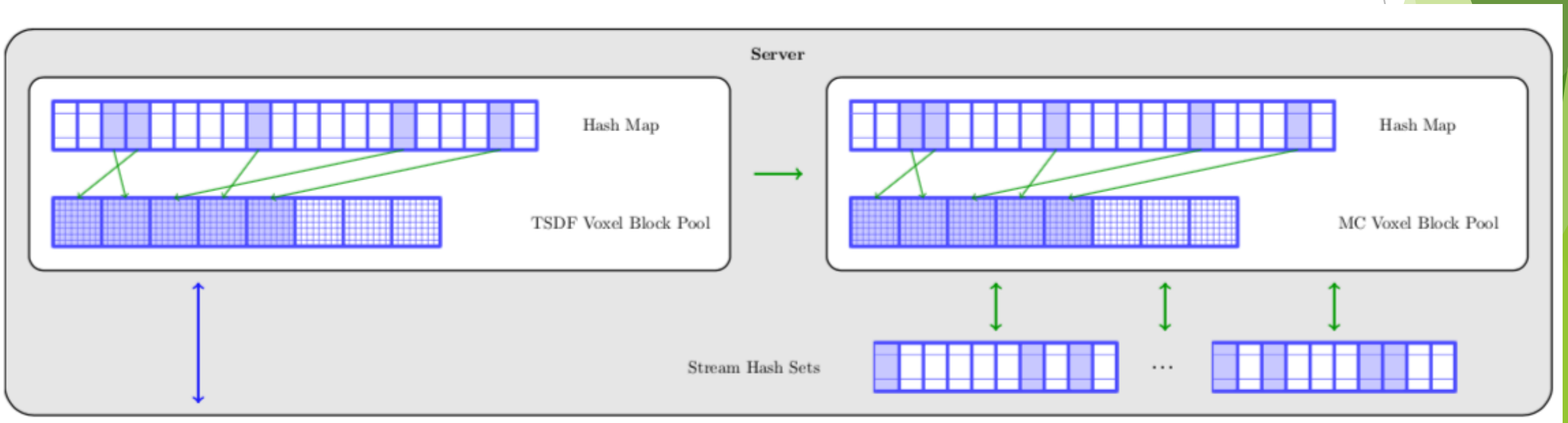
- Receives a stream of RGB-D images
- Use voxel block hashing to reconstruct a virtual 3D model
- Consider only voxel blocks that have been fully constructed (b/w limit)
- Check EMA(exponential moving average) of the stream set size is below a threshold
  - Ensure the delayed but complete model is available to the server and exploration clients at all times





Server

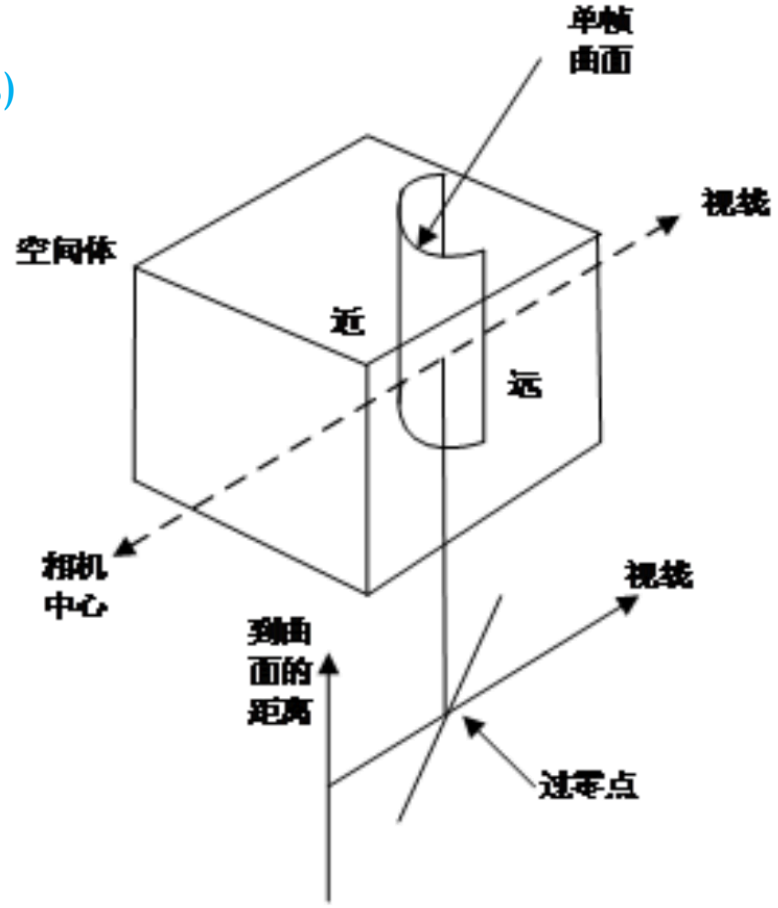
- Manage the global voxel block model
- TSDF Voxel Block (12bytes/voxel) -> MC voxel block (4bytes/voxel)
  - Bandwidth optimized representation



- TSDF (Truncated Signed Distance Field)

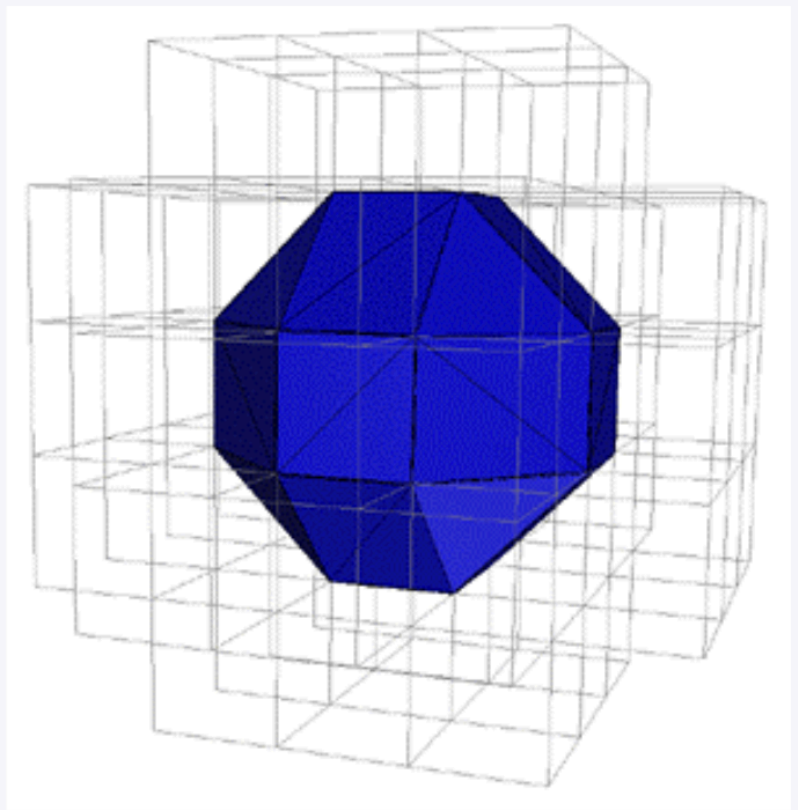
Zumbach and U. Mu¨ller. 2001. Operators on inhomogeneous time series. International Journal of Theoretical and Applied Finance 4, 01 (2001), 147–177

A TSDF Voxel: a TSDF value (4bytes) + a fusion weight (4bytes) + a color (3 bytes + 1 byte alignment)



- MC(Marching Cubes)

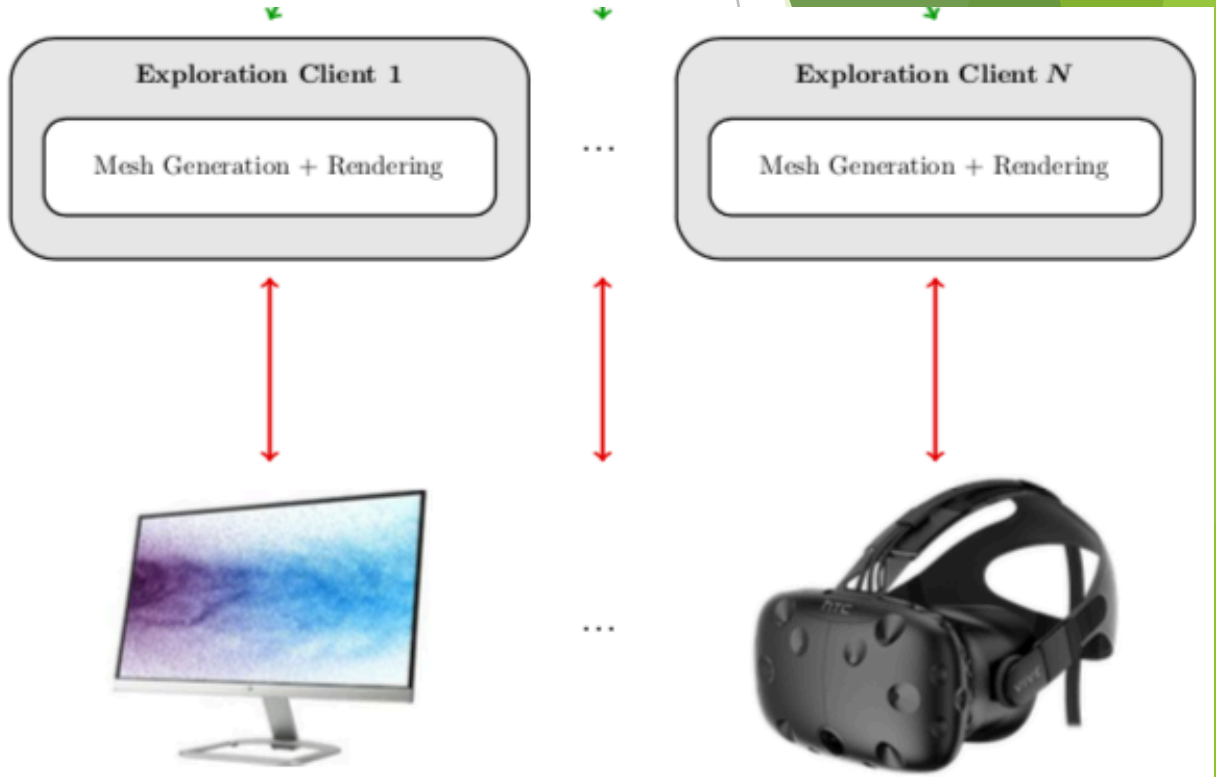
W. E. Lorensen and H. E. Cline. 1987. Marching Cubes: A High Resolution 3D Surface Construction Algorithm. In Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '87). ACM, New York, NY, USA, 163–169. https://doi.org/10.1145/37401.37422



A MC Voxel: a Marching Cubes index (1 byte), and a color value (3 bytes)

Exploration Client

- Generate surface geometry
- Update reconstructed model
- Render the model
- Reduce the number of draw calls to graphic API
  - Merge 15^3 voxel blocks into a mesh block (small geographic unit)



Experiment Setup

- 1 reconstruction client, 1 server, 2 exploration clients
- RGBD sensor: Microsoft kinetic v2 (512x424 pixels , 30Hz), Asus Zenfone AR(224x172, 10Hz)
- HTC Vive HMD
- Local network

Bandwidth analysis

compared the mean (and maximum) bandwidths of our optimized MC voxel structure with 128-1024 blocks/request to the standard TSDF voxel one with 512 blocks/request

Dataset	Voxel Size [mm]	Bandwidth [MBit/s]					Total Voxel Blocks
		MC 128	MC 256	MC 512	MC 1024	TSDF 512	
heating_room	5	4.5 (8.0)	8.8 (12.3)	17.5 (30.9)	32.7 (71.3)	561.5 (938.8)	897×10^3
pool	5	4.6 (7.1)	9.0 (14.0)	17.8 (29.7)	29.3 (54.5)	489.3 (937.0)	637×10^3
fr1/desk2	5	8.1 (11.6)	16.2 (23.8)	32.6 (46.8)	61.0 (95.0)	764.0 (938.6)	134×10^3
fr1/room	5	12.3 (23.6)	16.4 (23.6)	32.1 (42.2)	57.6 (87.9)	739.7 (938.0)	467×10^3
heating_room	10	5.1 (7.6)	9.2 (14.4)	14.6 (27.8)	20.2 (63.7)	216.8 (937.1)	147×10^3
pool	10	5.6 (8.5)	9.9 (16.0)	13.6 (27.2)	16.9 (52.3)	176.3 (937.0)	104×10^3
fr1/desk2	10	8.7 (11.2)	14.3 (21.8)	19.6 (39.2)	24.4 (71.3)	170.1 (436.4)	23×10^3
fr1/room	10	9.2 (12.5)	15.7 (23.5)	22.9 (46.1)	28.5 (88.8)	207.8 (936.6)	86×10^3

saved more than 90% of the bandwidth (MC 512 vs. TSDF 512) and scales linearly with the request rate.

Visual Quality

evaluated the model completeness during transmission for our novel hash map data structure in comparison to previous techniques that allow failures

(a) Original Voxel Block Hashing Data Structure [Nießner et al. 2013].

(b) Our Hash Data Structure.

Conclusion

- Limitation: users move relatively fast resulting in high angular and linear velocities as well as potential motion blur
- First thread-safe GPU hash map
- Efficient streaming by transmitting representation in terms of MC indices
- Overcome inherently limited resolution of voxel-based scene representations with a lightweight projective texture mapping approach