Fusing Physical and Virtual Worlds into Interactive Mixed Reality

Ruofei Du | Google, San Francisco | me@duruofei.com
Virtual | Mobile Immersive Computing by Prof. Bo Han

Self Intro

www.duruofei.com

Publications

Projects

Videos

Talks

Artsy

© Ruofei Du (杜若飞) | 2020 | DuWebKit

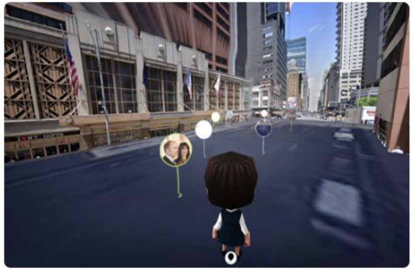
Featured Publications

DepthLab: Real-Time 3D Interaction With Depth Maps for Mobile Augmented Reality

Ruofei Du, Eric Turner, Maksym Dzitsiuk, Luca Prasso, Ivo Duarte, Jason Dourgarian, Joao Afonso, Jose Pascoal, Josh Gladstone, Nuno Cruces, Shahram Izadi, Adarsh Kowdle, Konstantine Tsotsos, and David Kim

Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology (UIST), 2020.

pdf, lowres | website, code, demo, supp | cite

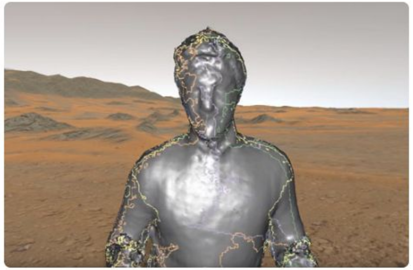


Geollery: A Mixed Reality Social Media Platform *Juried Demo at CHI 2019*

Ruofei Du, David Li, and Amitabh Varshney

Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI), 2019.

pdf, doi | website, video, slides, demo | cite



Montage4D: Real-Time Seamless Fusion and Stylization of Multiview Video Textures

Ruofei Du, Ming Chuang, Wayne Chang, Hugues Hoppe, and Amitabh Varshney

Journal of Computer Graphics Techniques (JCGT), 2019.

Social Street View: Blending Immersive Street Views With Geo-Tagged Social Media *Best Paper Award*

Ruofei Du and Amitabh Varshney

Proceedings of the 21st International Conference on Web3D Technology (Web3D), 2016.

Self Intro

Ruofei Du (杜若飞)

PEOPLE ›

Ruofei Du

About

Ruofei Du is a Research Scientist at Google and works on creating novel interactive technologies for VR and AR. Ruofei holds a Ph.D. in Computer Science from University of Maryland, College Park, advised by Prof. Varshney. Their research covers a wide range of topics in VR and AR, including mobile augmented reality (DepthLab), mixed-reality social platforms (Geollery and Social Street View), 4D video-based rendering (Montage4D and VideoFields), foveated rendering (KFR and EFR), and deep learning in graphics (SketchySceneColorization). I am passionate about inventing interactive technologies with computer graphics, 3D vision, and HCI. Feel free to visit my research, arts, blog, youtube, github, and shadertoy demos for fun!

Personal website

Google scholar

Research Areas

- Human-Computer Interaction and Visualization →
- Machine Intelligence →
- Machine Perception →

Authored publications

Google publications

Filters

Research areas +

Year +

Sort by: Year ⌄

9 publications

A Log-Rectilinear Transformation for Foveated 360-degree Video Streaming
David Li, Ruofei Du, Adharsh Babu, Camelia D. Brumar, Amitabh Varshney • *IEEE Transactions on Visualization and Computer Graphics*, pp. 1-10 (to appear)

3D-Kernel Foveated Rendering for Light Fields
Xiaoxu Meng, Ruofei Du, Joseph F. JaJa, Amitabh Varshney • *IEEE Transactions on Visualization and Computer Graphics* (2020)

CollaboVR: A Reconfigurable Framework for Creative Collaboration in Virtual Reality
Zhenyi He, Ruofei Du, Ken Perlin • *2020 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, IEEE, pp. 11

DepthLab: Real-time 3D Interaction with Depth Maps for Mobile Augmented Reality
Ruofei Du, Eric Lee Turner, Maksym Dzitsiuk, Luca Prasso, Ivo Duarte, Jason Dourgarian, Joao Afonso, Jose Pascoal, Josh Gladstone, Nuno Moura e Silva Cruces, Shahram Izadi, Adarsh Kowdle, Konstantine Nicholas John Tsotsos, David Kim • *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*, ACM (2020), pp. 15

Experiencing Real-time 3D Interaction with Depth Maps for Mobile Augmented Reality in DepthLab
Ruofei Du, Eric Lee Turner, Maksym Dzitsiuk, Luca Prasso, Ivo Duarte, Jason Dourgarian, Joao Afonso, Jose Pascoal, Josh Gladstone, Nuno Moura e Silva Cruces, Shahram Izadi, Adarsh Kowdle, Konstantine Nicholas John Tsotsos, David Kim • *Adjunct Publication of the 33rd Annual ACM Symposium on User Interface Software and Technology*, ACM (2020), pp. 3

Eye-dominance-guided Foveated Rendering
Xiaoxu Meng, Ruofei Du, Amitabh Varshney • *IEEE Transaction on Visualization and Computer Graphics*, vol. 26 (2020), pp. 1972-1980

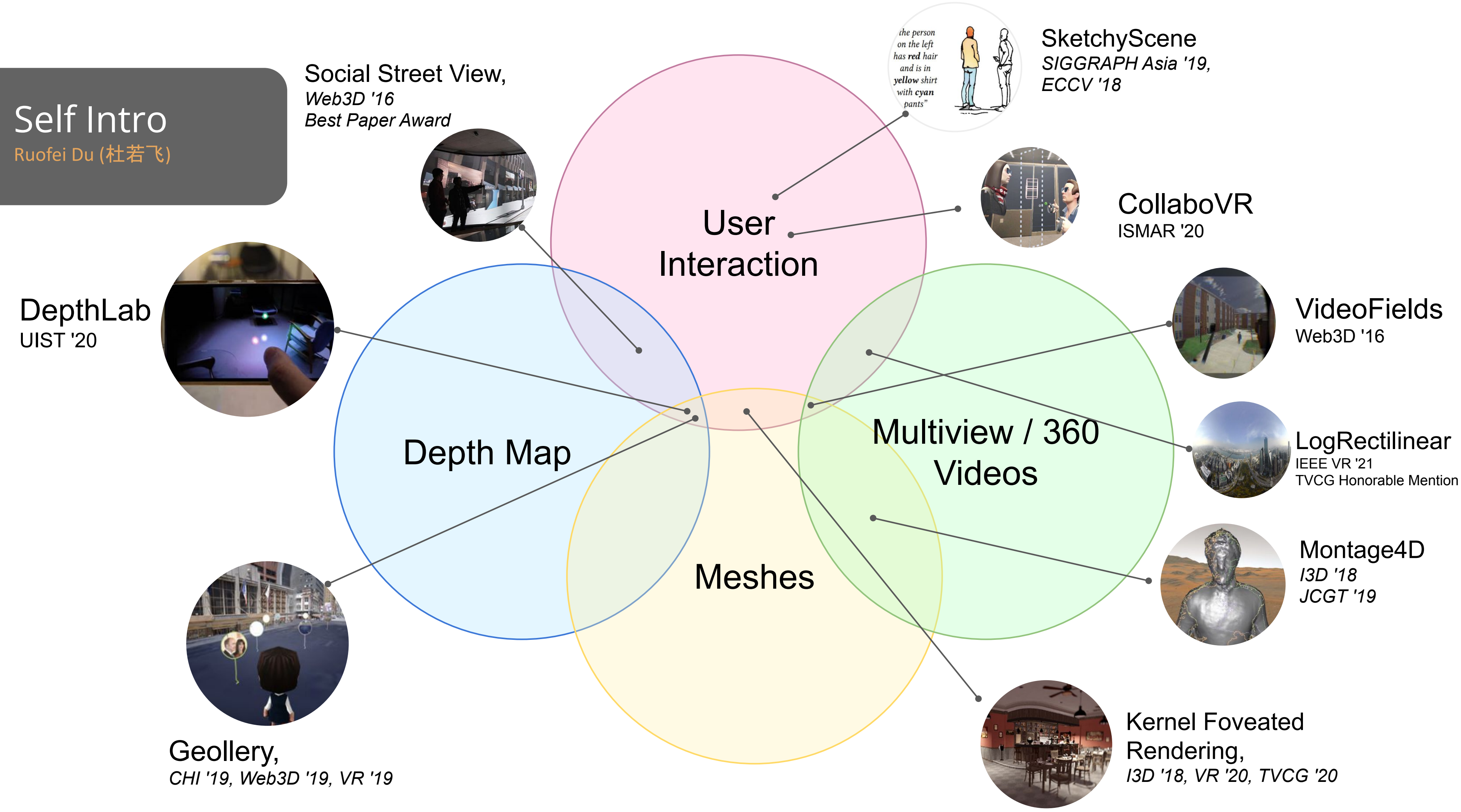
MeteoVis: Visualizing Meteorological Events in Virtual Reality
David Li, Eric Lee, Elijah Schwelling, Mason G. Quick, Patrick Meyers, Ruofei Du, Amitabh Varshney • *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, ACM, Honolulu, Hawaii, pp. 9

Language-based Colorization of Scene Sketches
Changqing Zou, Haoran Mo, Chengying Gao, Ruofei Du, Hongbo Fu • *ACM Transactions on Graphics*, vol. 38, pp. 16

ORC Layout: Adaptive GUI Layout with OR-Constraints
Yue Jiang, Ruofei Du, Christof Lutteroth, Wolfgang Stuerzlinger • *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI 2019)*, ACM, pp. 1-12

Self Intro

Ruofei Du (杜若飞)



DepthLab: Real-time 3D Interaction with Depth Maps for Mobile Augmented Reality

Ruofei Du, Eric Turner, Maksym Dzitsiuk, Luca Prasso, Ivo Duarte,
Jason Dourgarian, Joao Afonso, Jose Pascoal, Josh Gladstone, Nuno Cruces,
Shahram Izadi, Adarsh Kowdle, Konstantine Tsotsos, David Kim

Google | ACM UIST 2020

Introduction

Mobile Augmented Reality

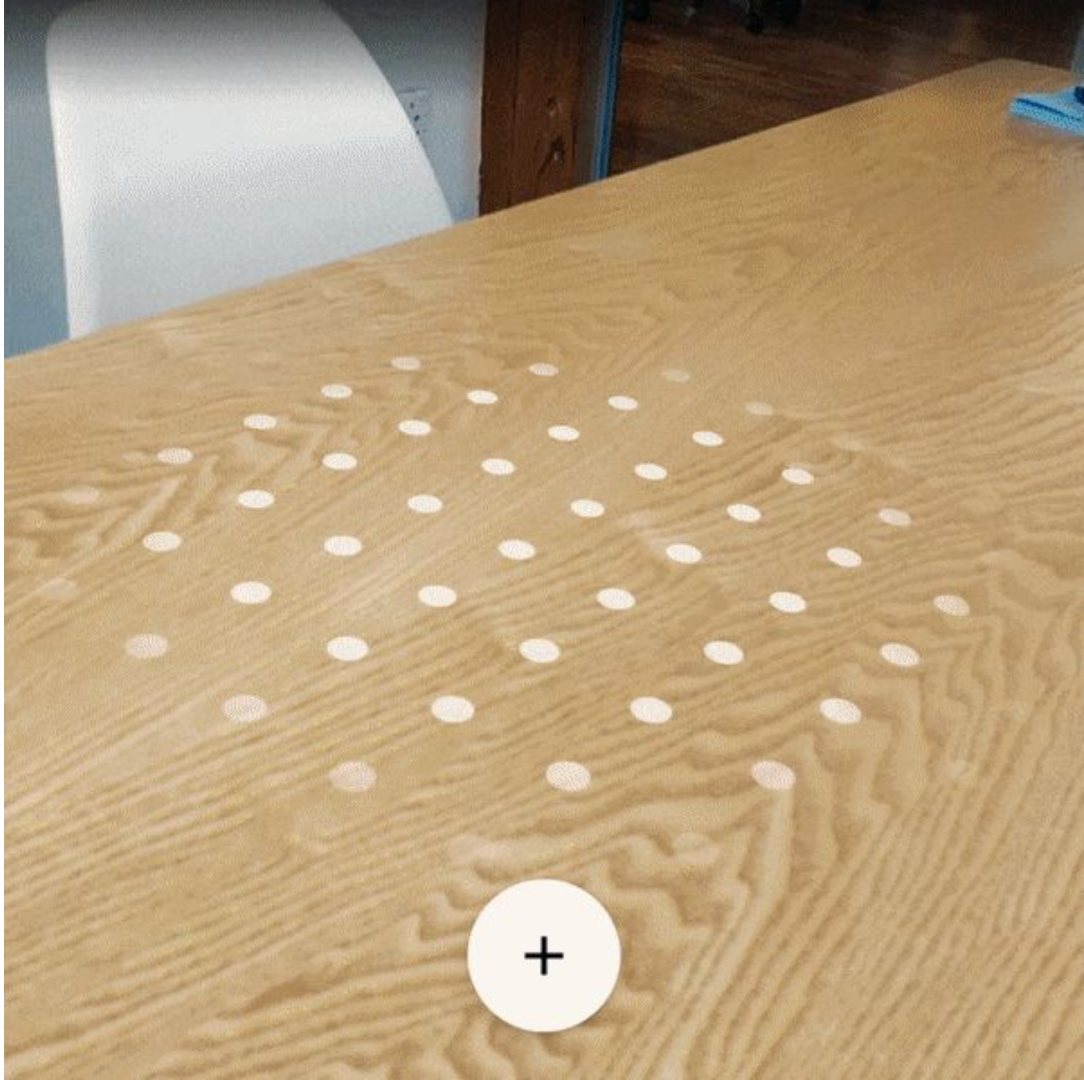
Introduction

Google's ARCore

Introduction

Google's ARCore

ARCore



Introduction

Mobile Augmented Reality

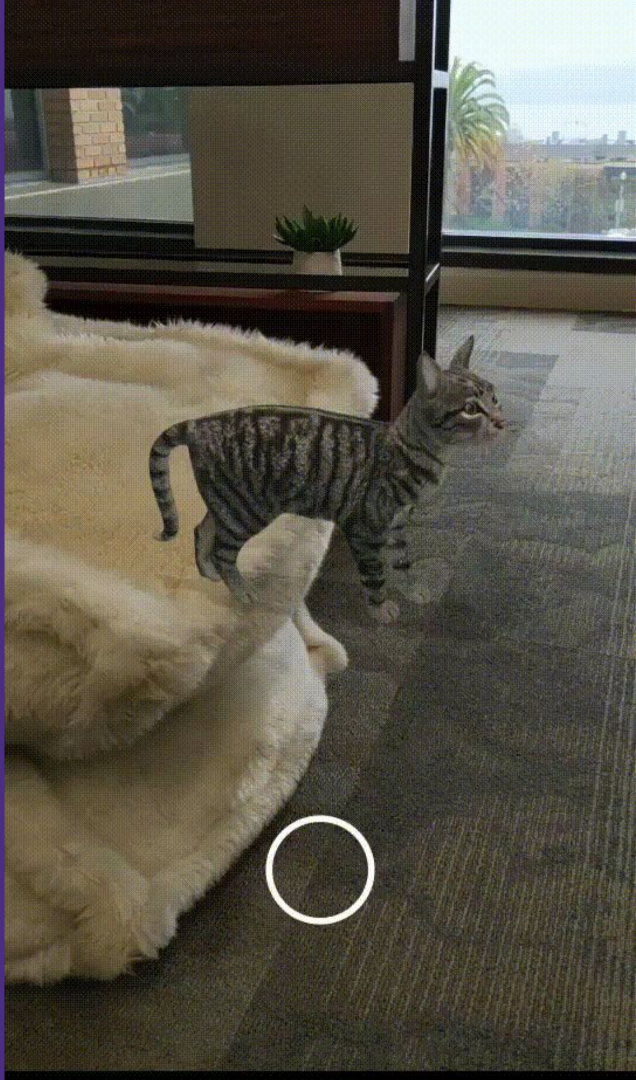
Is direct placement and rendering of 3D objects sufficient for realistic AR experiences?

Not always!

Introduction

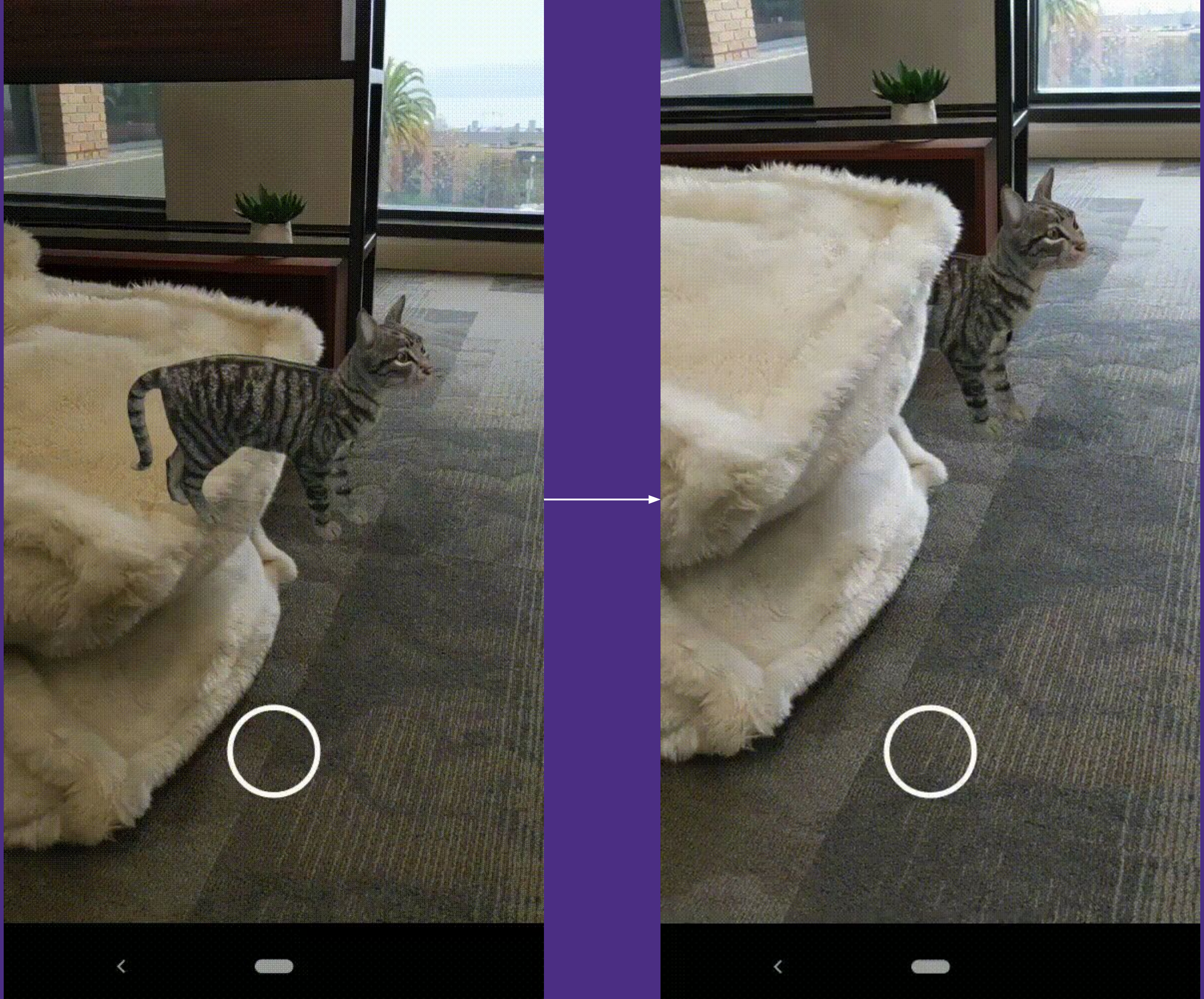
Depth Lab

Virtual content looks like it's *"pasted on the screen"* rather than *"in the world"*!



Introduction

Motivation



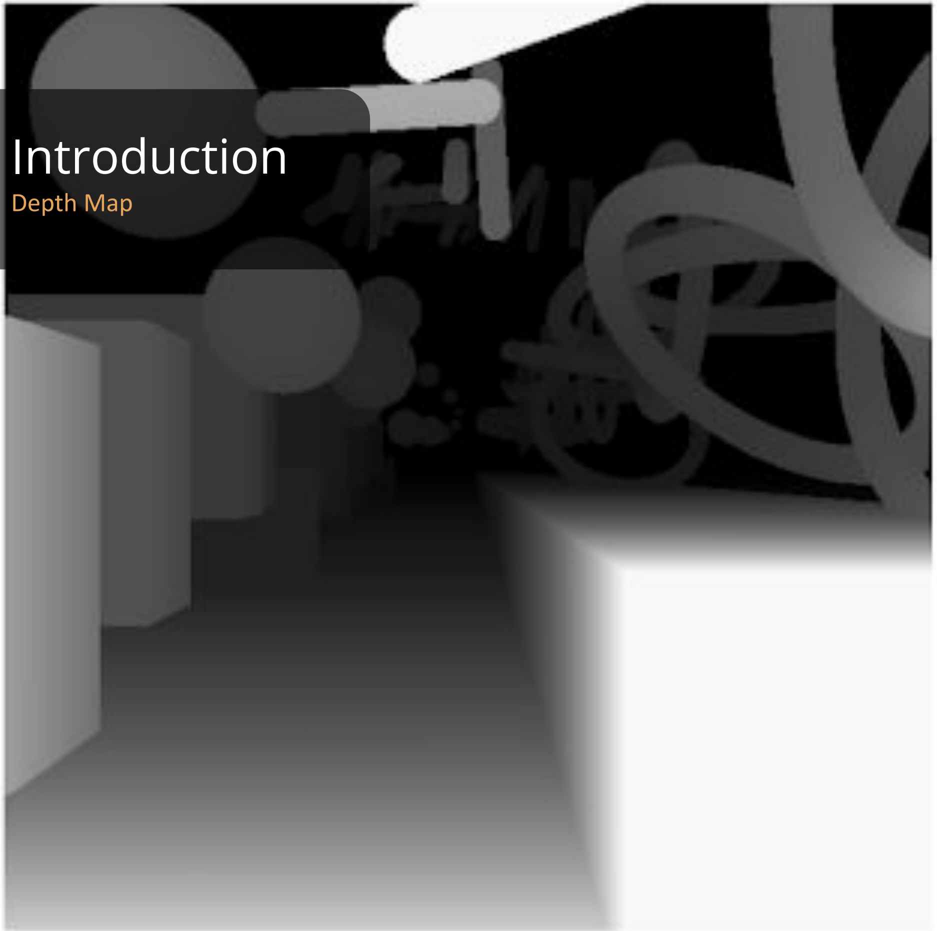
Introduction

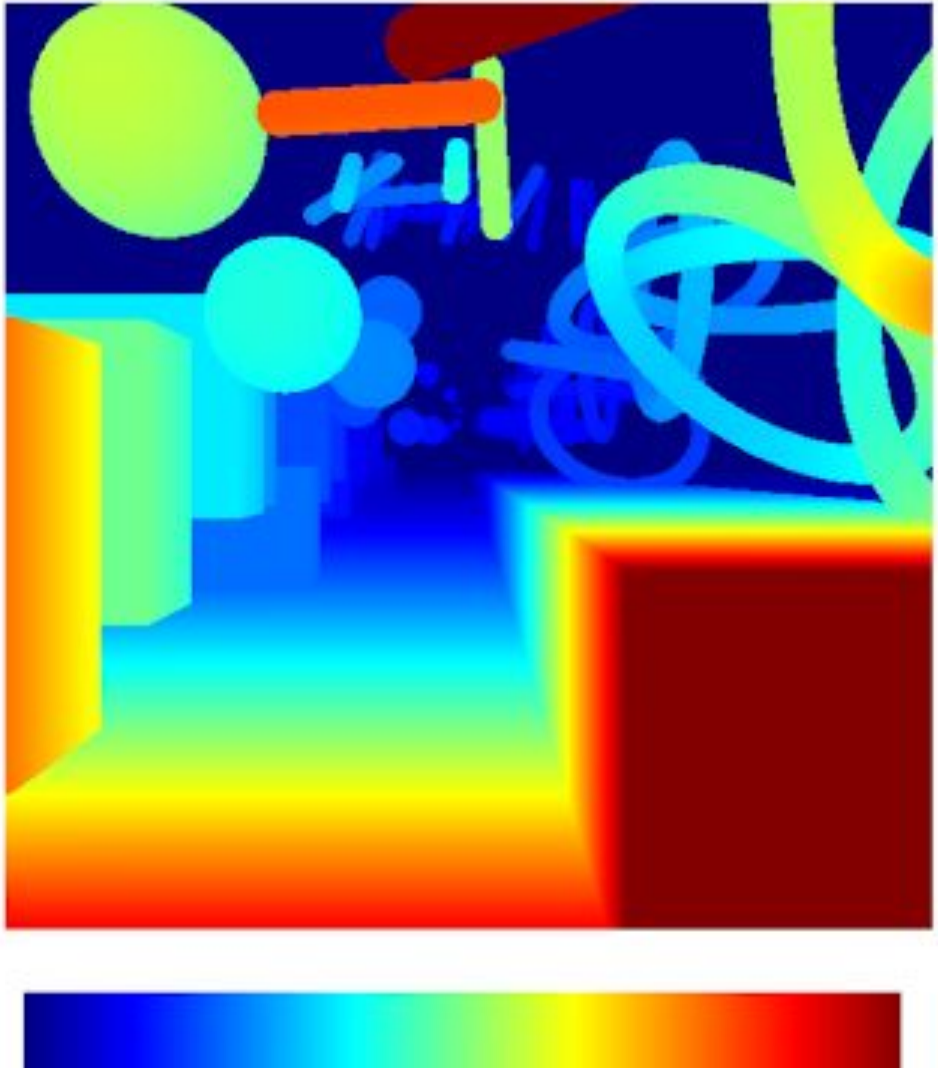
Motivation

How can we bring these advanced features to mobile AR experiences without relying on dedicated sensors or the need for computationally expensive surface reconstruction?

Introduction

Depth Map





Introduction
Depth Lab

Google	• *Pixel 2, Pixel 2 XL, Pixel 3, Pixel 3 XL, Pixel 3a, Pixel 3a XL, Pixel 4, Pixel 4 XL*
Huawei	• *Honor 10, Honor V20, Mate 20 Lite, Mate 20, Mate 20 X, Nova 3, Nova 4, P20, P30, P30 Pro*
LG	• *G8X ThinQ, V35 ThinQ, V50S ThinQ, V60 ThinQ 5G*
OnePlus	• *OnePlus 6, OnePlus 6T, OnePlus 7, OnePlus 7 Pro, OnePlus 7 Pro 5G, OnePlus 7T, OnePlus 7T Pro*
Oppo	• *Reno Ace*
Samsung	• *Galaxy A80, Galaxy Note8, Galaxy Note9, Galaxy Note10, Galaxy Note10 5G, Galaxy Note10+, Galaxy Note10+ 5G, Galaxy S8, Galaxy S8+, Galaxy S9, Galaxy S9+, Galaxy S10e, Galaxy S10, Galaxy S10+, Galaxy S10 5G, Galaxy S20, Galaxy S20+ 5G, Galaxy S20 Ultra 5G*
Sony	• *Xperia XZ2, Xperia XZ2 Compact, Xperia XZ2 Premium, Xperia XZ3*
Xiaomi	• *Pocophone F1*

And growing…

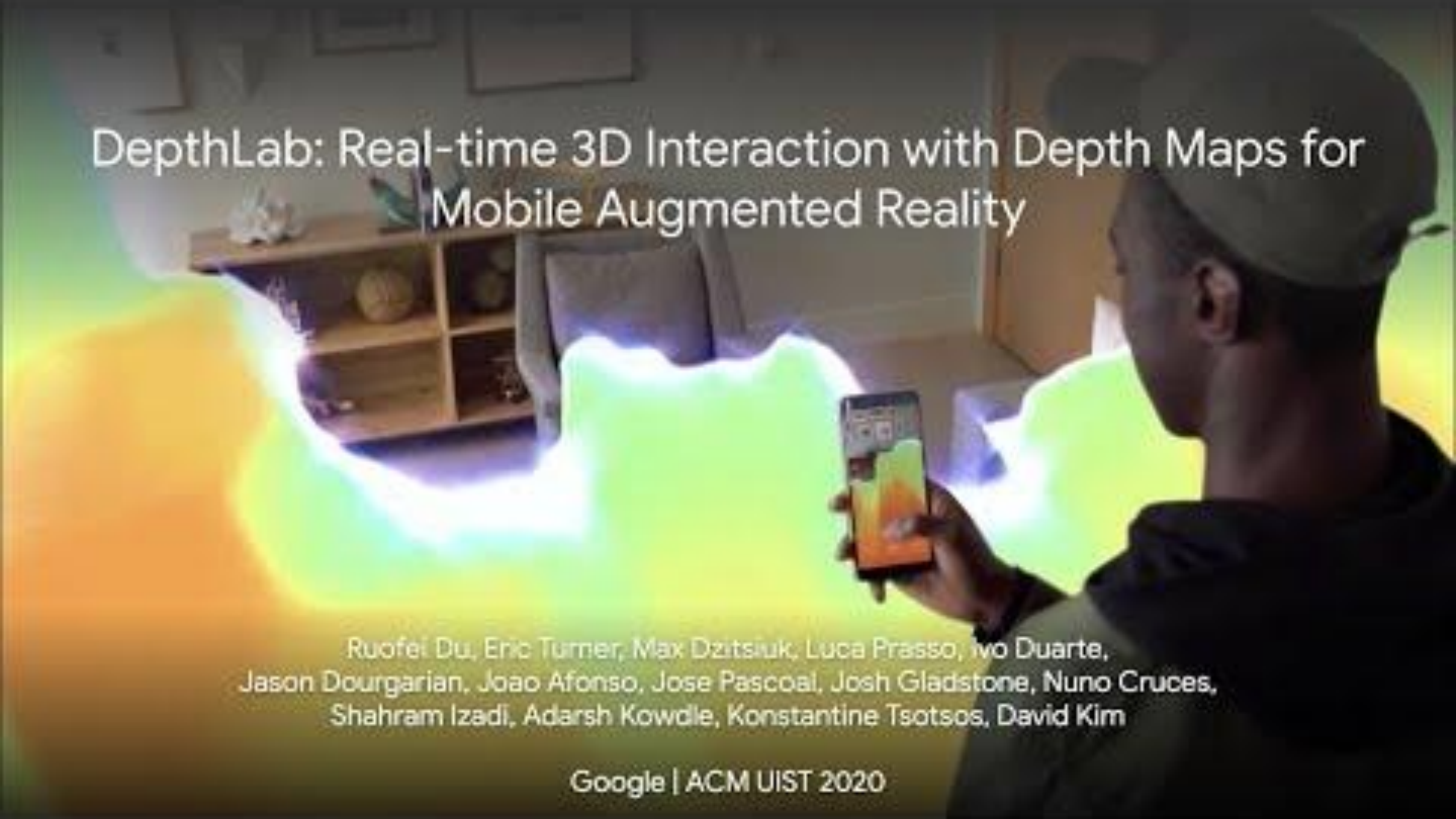
https://developers.google.com/ar/discover/supported-devices

Is there *more* to realism than occlusion?

Surface interaction?

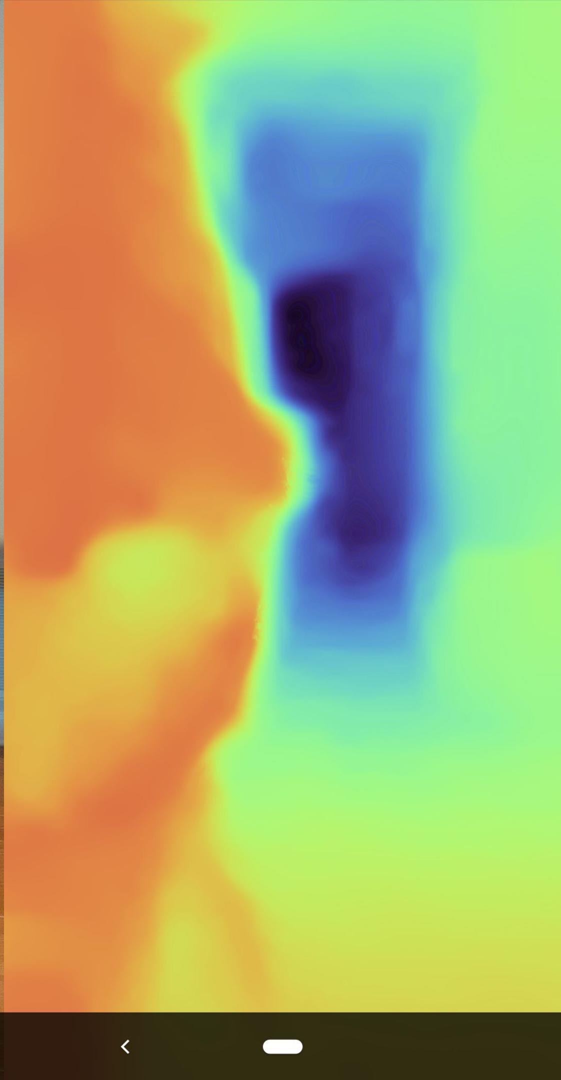
Realistic Physics?

Path Planning?


DepthLab: Real-time 3D Interaction with Depth Maps for Mobile Augmented Reality
Ruofei Du, Eric Turner, Max Dzitsiuk, Luca Prasso, Ivo Duarte,
Jason Dourgarian, Joao Afonso, Jose Pascoal, Josh Gladstone, Nuno Cruces,
Shahram Izadi, Adarsh Kowdle, Konstantine Tsotsos, David Kim
Google | ACM UIST 2020

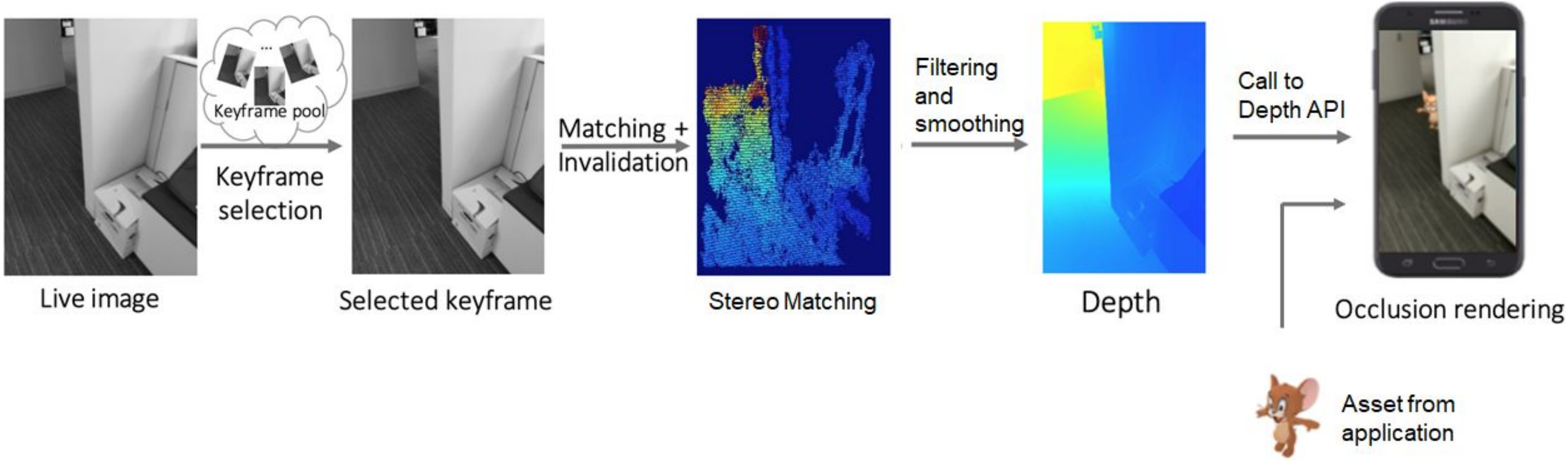
Introduction

Depth Lab



Related Work

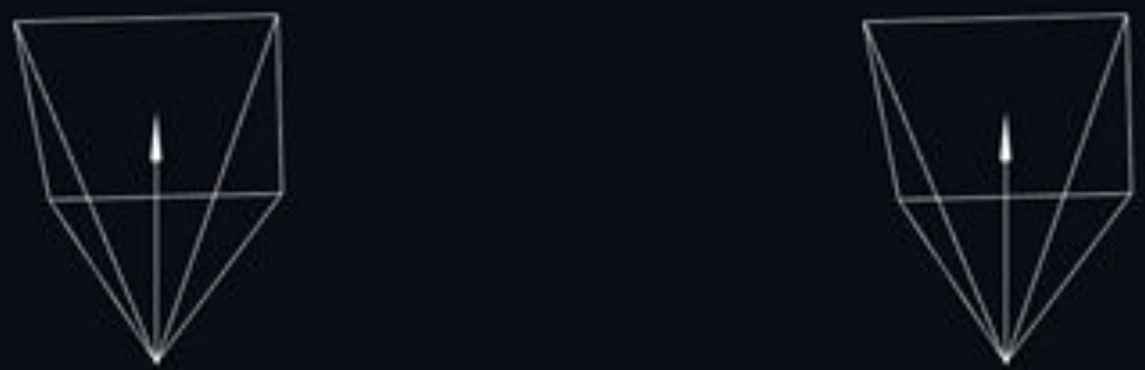
Valentin et al.



Depth from Motion for Smartphone AR (SIGGRAPH Asia 2018)

Introduction

Depth Lab



Introduction

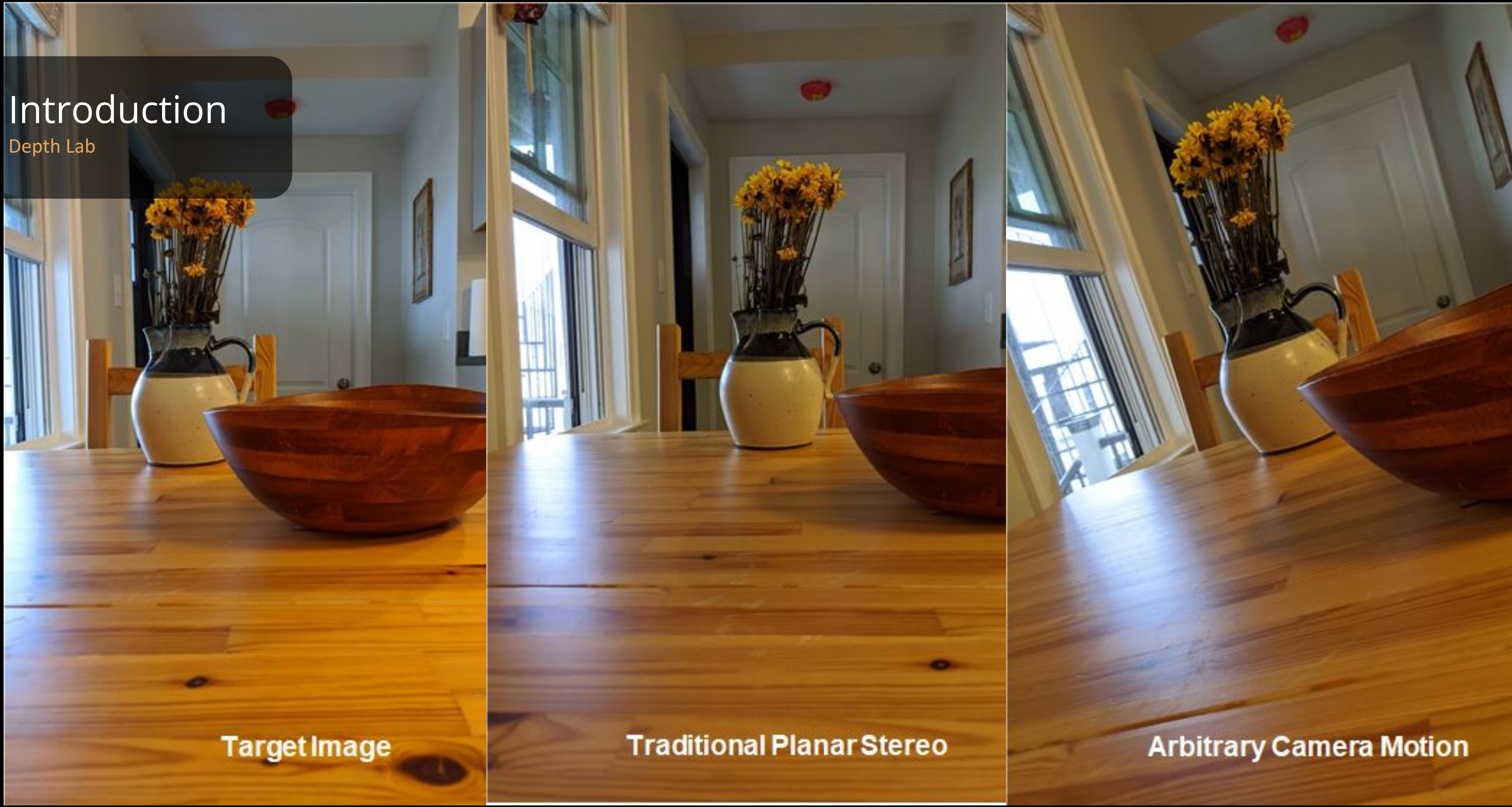
Depth Lab

Introduction

Depth Generation

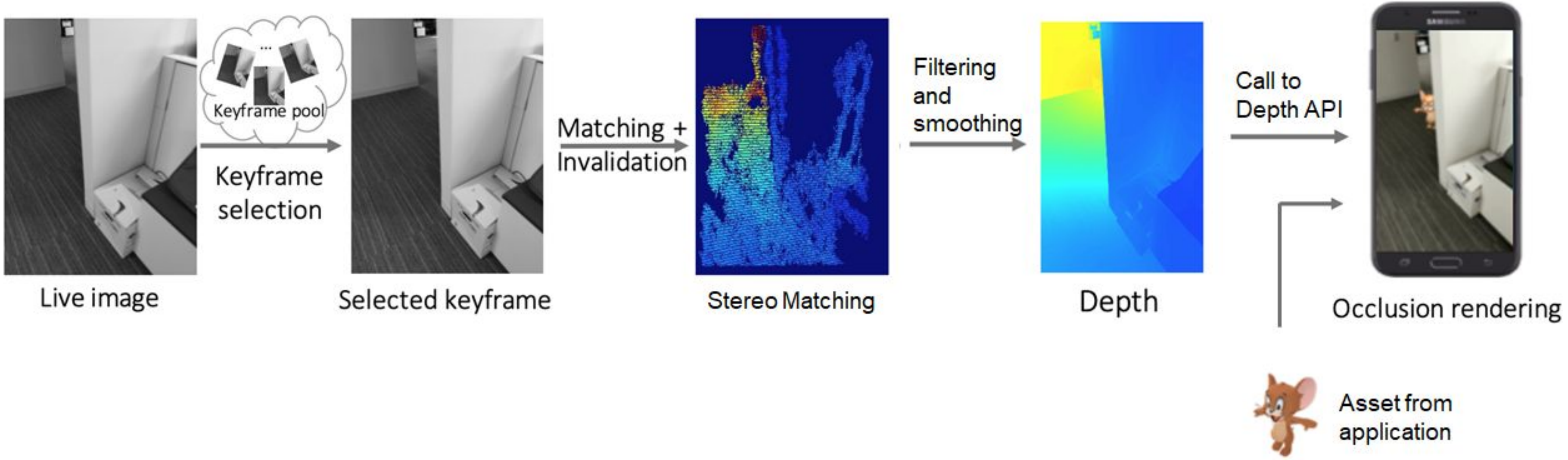
Introduction

Depth Lab



Related Work

Valentin et al.



Depth from Motion for Smartphone AR (SIGGRAPH Asia 2018)

Introduction

Depth Lab

Introduction

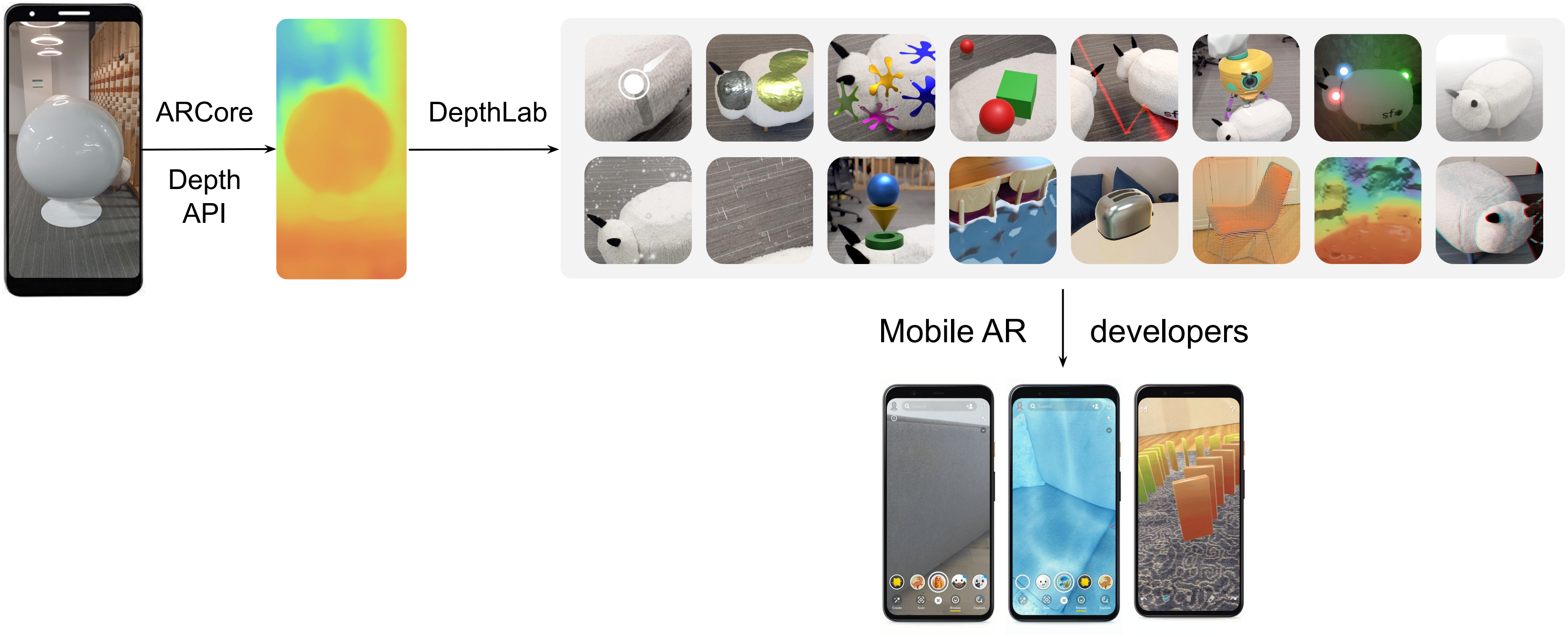
Depth Lab

Up to 8 meters, with
the best within 0.5m to 5m

Motivation

Gap from raw depth to applications

Introduction
Depth Lab



Design Process

3 Brainstorming Sessions

3 brainstorming sessions
18 participants
39 aggregated ideas

Design Process

3 Brainstorming Sessions

Supplementary Material for DepthLab: Real-time 3D Interaction with Depth Maps for Mobile Augmented Reality

Ruofei Du, Eric Turner, Maksym Dzitsiuk, Luca Prasso, Ivo Duarte, Jason Dourgarian, Joao Afonso, Jose Pascoal, Josh Gladstone, Nuno Cruces, Shahram Izadi, Adarsh Kowdle, Konstantine Tsotsos, David Kim†
Google LLC

GEOMETRY-AWARE AR FEATURES
In this section, we list all ideas from our brainstorming sessions and discuss their depth representation requirements, use cases, and whether each is implemented in DepthLab [5]. Note that ideas 9, 21, 24, 25 are not available as open source code yet, but can be easily reproduced with the provided algorithms.

Depth Representation Requirement: Localized Depth

Figure 1. Implementation examples of geometry-aware AR features 1–5 with localized depth use cases. Please refer to the supplementary video for live demonstration.

1. 3D oriented cursor: Render a 3D cursor centered in the screen center. The 3D cursor should change its orientation and scale according to the surface normal and distance when moving along physical surfaces.
Implemented in DepthLab: Yes.

4. Avatar path planning: Navigate a virtual object to move naturally between two points in physical environments.
Implemented in DepthLab: Yes.

5. Collision-aware placement: Test if a virtual object's volume collides with observed environment surfaces.
Implemented in DepthLab: Yes.

Depth Representation Requirement: Surface Depth

Figure 2. Implementation examples of geometry-aware AR features 6–9.

6. Virtual shadows: Render geometry-aware shadows [11] that are cast onto physical surfaces. The shadow may be ...

8. Physical simulation: Simulate physical phenomena for augmented reality objects, e.g. collision.
Implemented in DepthLab: Yes.

9. AR graffiti: Allow the user to touch on the screen and sketch/spray/paint virtual drawings onto physical objects.
Implemented in DepthLab: Yes.

10. AR paintball: ...

Figure 3. Implementation examples of geometry-aware AR features 15–25 with dense depth.

19. Rain effects: Similiar in behavior to the snow effect, the rain particles should also splat on the surface using the estimated normal vector from the localized depth.
Implemented in DepthLab: Yes.

20. Fog effects: Render screen-space post-processing effects, where far objects are overlaid with thicker fog. The user may interactively adjust the fog intensity in real time.
Implemented in DepthLab: Yes.

21. Edge highlighting: Highlight the edges of the observed environment according to the depth map. Unlike edge detection in a color image, highlighting depth edges may offer a clean segmentation of physical objects regardless of their texture.
Implemented in DepthLab: Yes.

22. Depth-based segmentation: Segment the foreground, background, or objects between a certain range of depth values from the color image. It may be useful for teleconferencing tasks.
Implemented in DepthLab: Yes.

23. False-color visualization and animated transition effects: Visualize the depth map based on a specific transfer function and animate the transition ...

35. Enable multitouch on surfaces [12]: User may annotate sticky notes and papers with a pen and "program" them to control smart lights, music, and other digital functions of the environment.

36. Person capture: Enable self-scanning with the frontal camera [1] and teleconference with the rear camera.

37. AR board game: Design an AR-based board game [2] that overlays digital assets upon physical cards with aware of users' gestures and actions.

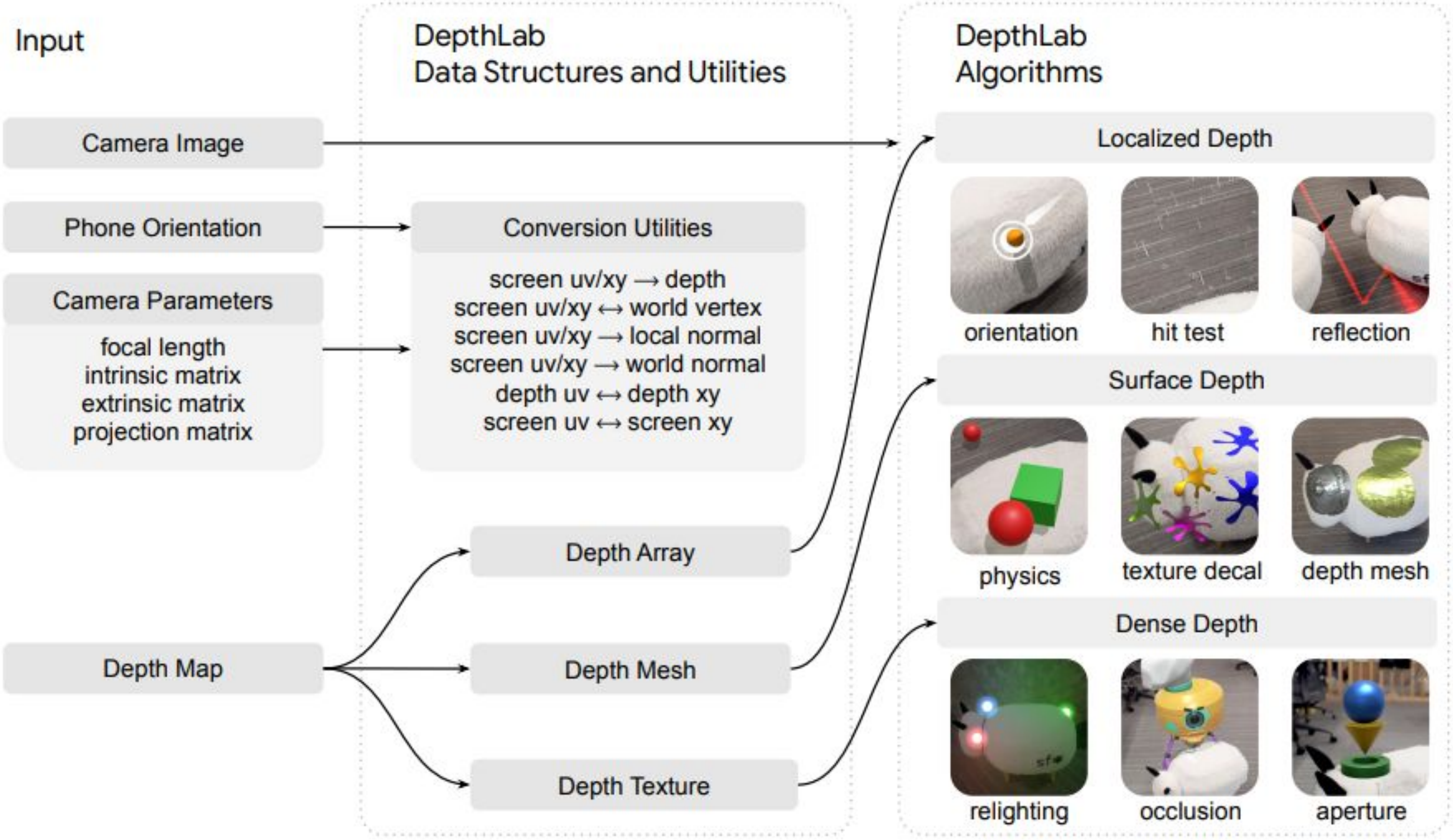
38. Interactive surface editing: Apply simple 3D distortion (pinch, twist, taper, bend) to captured colored voxels of the physical environment [4].

39. Interactive music experience: Design in-air instruments (guitar, piano) with dynamic gesture recognition [8]. Virtual targets are placed in 3D space, such as a drum set, big piano keys, etc. Upon contact detection, the app plays a sound.

REFERENCES
[1] Karan Ahuja, Chris Harrison, Mayank Goel, and Robert Xiao. 2019. MeCap: Whole-Body Digitization for Low-Cost VR/AR Headsets. In Proceedings of the 32Nd Annual ACM Symposium on User Interface Software and Technology (UIST '19). ACM, ACM, New York, NY, USA, 453–462. DOI: http://dx.doi.org/10.1145/3332165.3347889
[2] Troels L Andersen, Sune Kristensen, Bjørn W Nielsen, and Kaj Grønbæk. 2004. Designing an augmented reality board game with children: the battleboard 3D experience. In Proceedings of the 2004 conference on Interaction design and children: building a community. 137–138.
[3] Sujal Bista, Icaro Lins Leitao da Cunha, and Amitabh Varshney. 2017. Kinetic Depth Images: Flexible Generation of Depth Perception. The Visual Computer 33, 10 (01 October 2017), 1357–1369.
[4] Ming Chuang and Michael Kazhdan. 2011. Interactive and anisotropic geometry processing using the screened Poisson equation. In ACM SIGGRAPH 2011 papers. 1–10.
[5] Ruofei Du, Eric Turner, Max Dzitsiuk, Luca Prasso, Ivo Duarte, Jason Dourgarian, Joao Afonso, ...

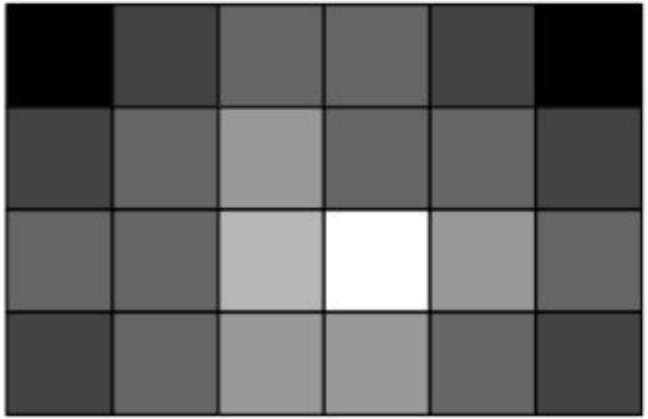
System

Architecture overview



Data Structure

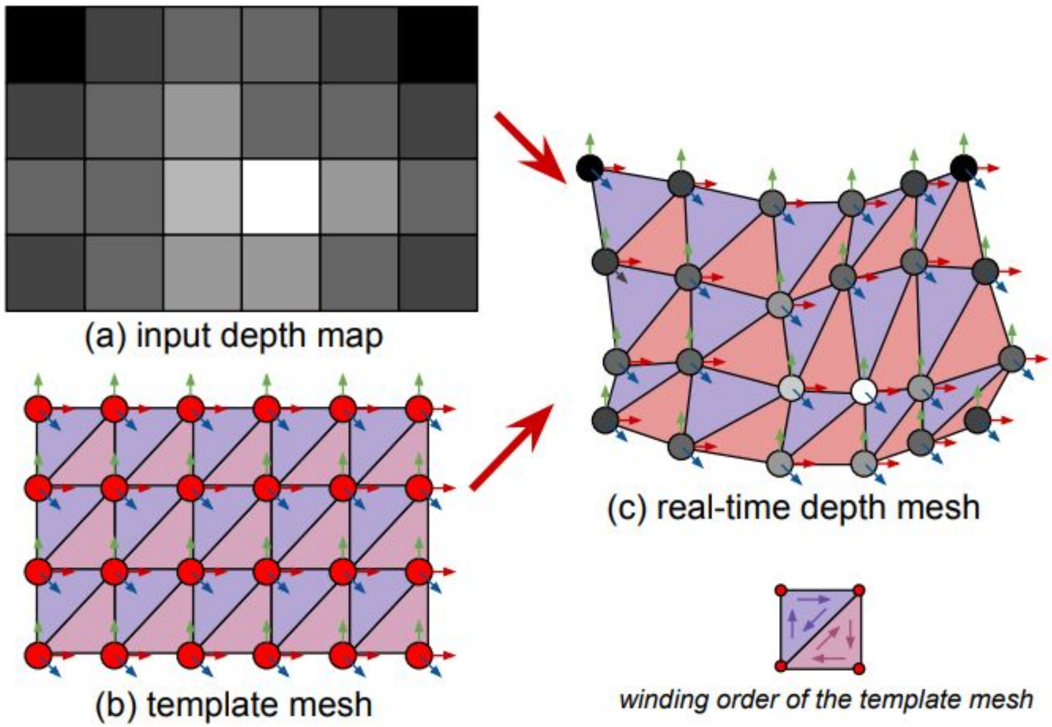
Depth Array



2D array (160x120 and above) of 16-bit integers

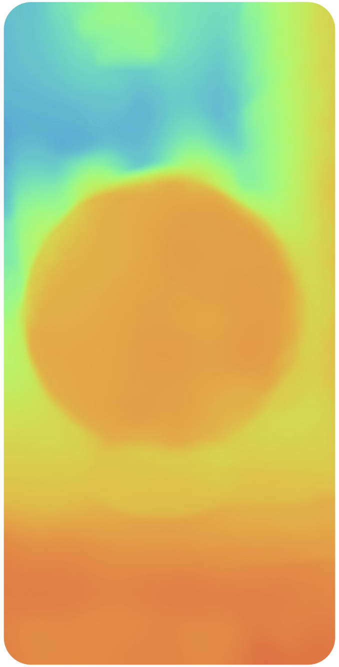
Data Structure

Depth Mesh



Data Structure

Depth Texture



System
Architecture

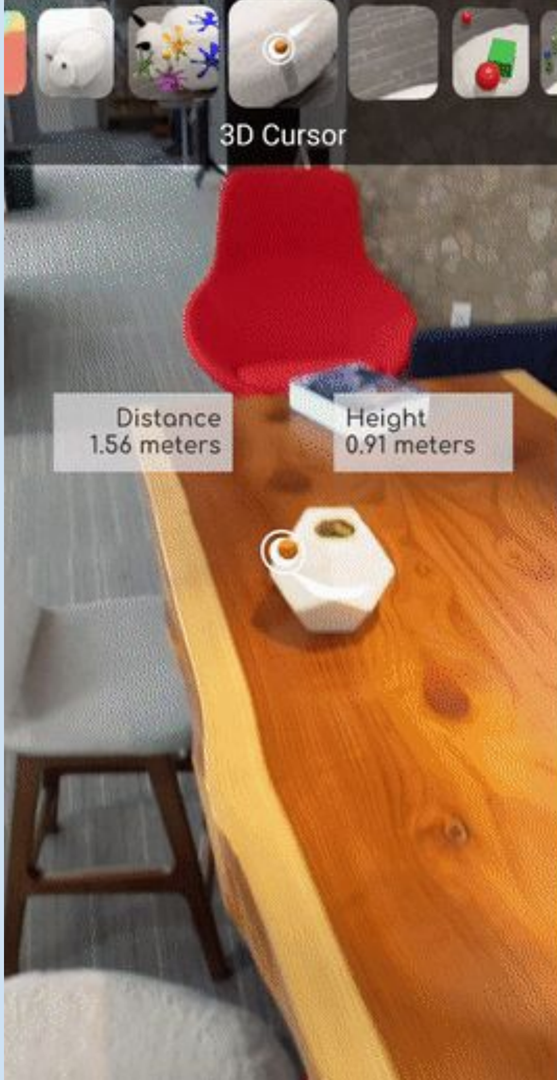
	Localized Depth	Surface Depth	Dense Depth
CPU	✓	✓	✗ (non-real-time)
GPU	N/A	✓ (compute shader)	✓ (fragment shader)
Prerequisite	point projection normal estimation	depth mesh triplanar mapping	anti-aliasing multi-pass rendering
Data Structure	depth array	depth mesh	depth texture
Example Use Cases	physical measure oriented 3D cursor path planning	collision & physics virtual shadows texture decals	scene relighting aperture effects occluded objects

Localized Depth

Coordinate System Conversion

Conversion Utilities

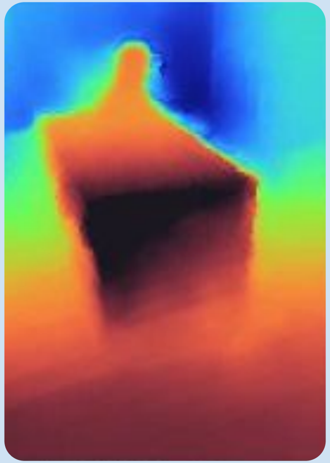
screen uv/xy ⟶ depth
screen uv/xy ⟷ world vertex
screen uv/xy ⟶ local normal
screen uv/xy ⟶ world normal
depth uv ⟷ depth xy
screen uv ⟷ screen xy

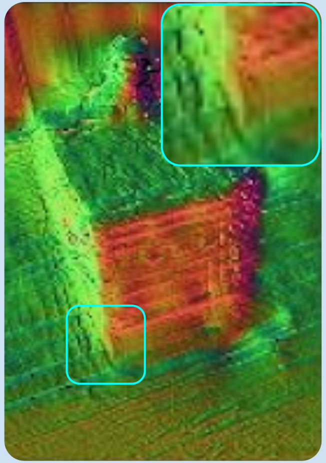


Localized Depth

Normal Estimation

$$\mathbf{n}_\mathbf{p} = \left(\mathbf{v}_\mathbf{p} - \mathbf{v}_{\mathbf{p}+(1,0)}\right) \times \left(\mathbf{v}_\mathbf{p} - \mathbf{v}_{\mathbf{p}+(0,1)}\right)$$





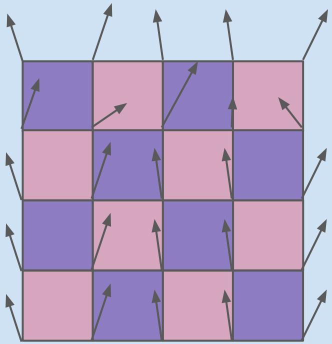
Localized Depth

Normal Estimation

Algorithm 1: Estimation of the Normal Vector of a Screen Point in DepthLab.

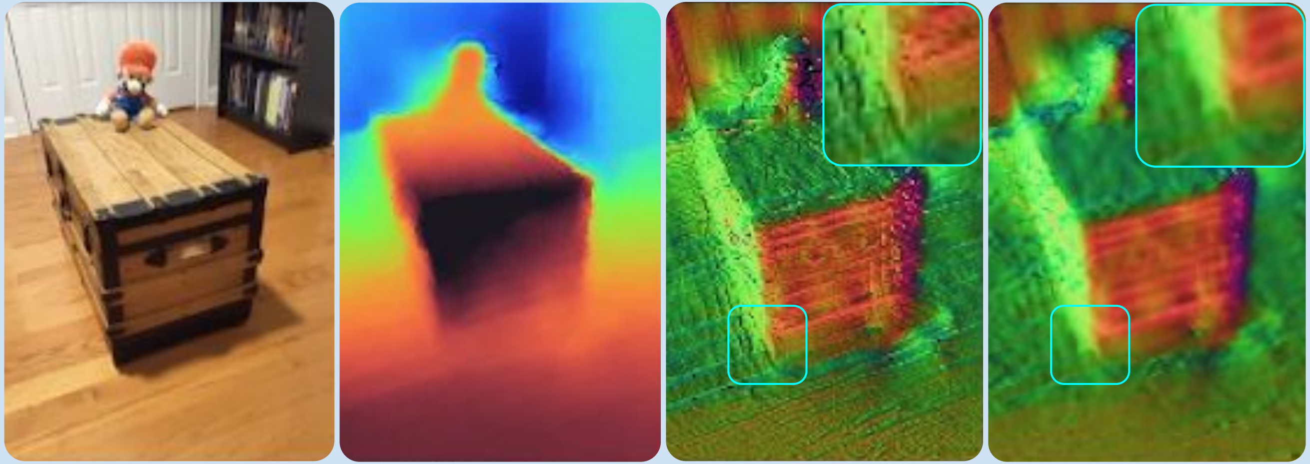
Input : A screen point $\mathbf{p} \leftarrow (x, y)$ and focal length f.
Output : The estimated normal vector $\mathbf{n}$.

Set the sample radius: $r \leftarrow 2$ pixels.
Initialize the counts along two axes: $c_X \leftarrow 0, c_Y \leftarrow 0$.
Initialize the correlation along two axes: $\rho_X \leftarrow 0, \rho_Y \leftarrow 0$.
for $\Delta x \in [-r, r]$ **do**
  for $\Delta y \in [-r, r]$ **do**
    Continue if $\Delta x = 0$ **and** $\Delta y = 0$.
    Set neighbor's coordinates: $\mathbf{q} \leftarrow [x + \Delta x, y + \Delta y]$.
    Set $\mathbf{q}$'s distance in depth: $d_{\mathbf{pq}} \leftarrow \|\mathbf{D}(\mathbf{p}), \mathbf{D}(\mathbf{q})\|$.
    Continue if $d_{\mathbf{pq}} = 0$.
    if $\Delta x \neq 0$ **then**
      $c_X \leftarrow c_X + 1$.
      $\rho_X \leftarrow \rho_X + d_{\mathbf{pq}} / \Delta x$.
    end
    if $\Delta y \neq 0$ **then**
      $c_Y \leftarrow c_Y + 1$.
      $\rho_Y \leftarrow \rho_Y + d_{\mathbf{pq}} / \Delta y$.
    end
  end
end
Set pixel size: $\lambda \leftarrow \frac{\mathbf{D}(\mathbf{p})}{f}$.
return the normal vector $\mathbf{n}$: $\left(-\frac{\rho_Y}{\lambda c_Y}, -\frac{\rho_X}{\lambda c_X}, -1\right)$.



Localized Depth

Normal Estimation

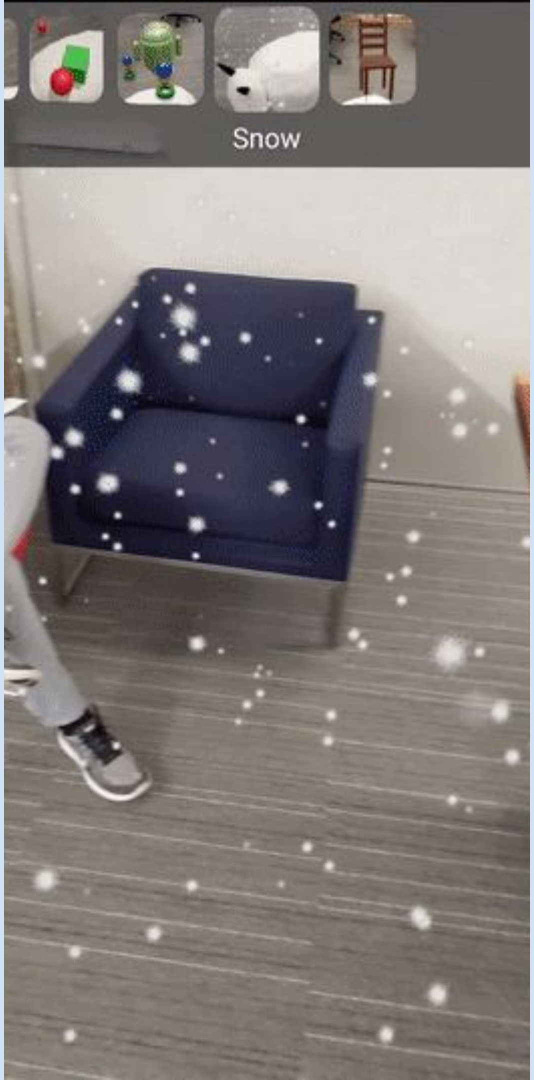


Localized Depth

Avatar Path Planning

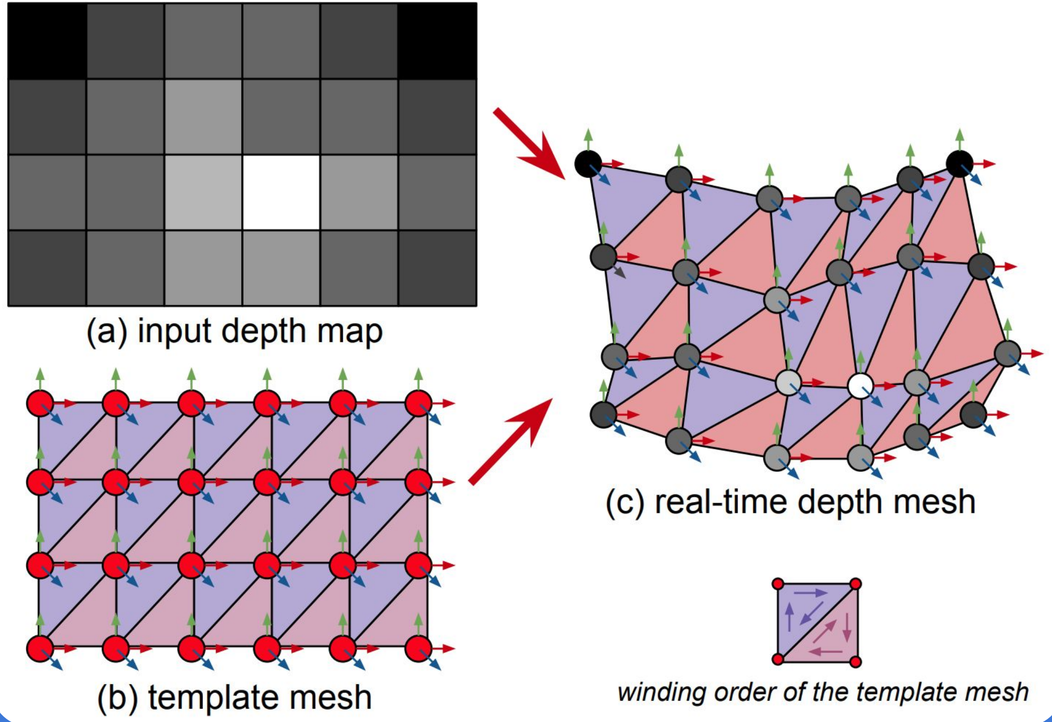
Localized Depth

Rain and Snow



Surface Depth

Use Cases



Surface Depth

Physics collider

Physics with depth mesh.

Surface Depth

Texture decals

Texture decals with depth mesh.

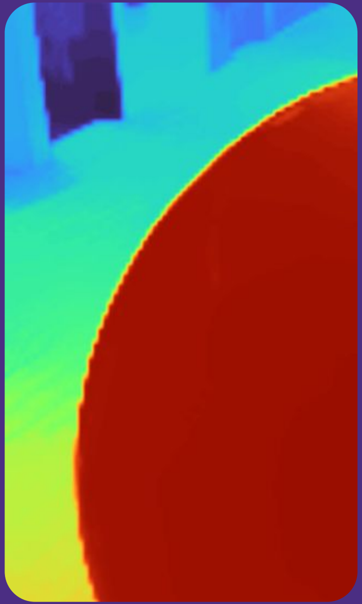
Surface Depth

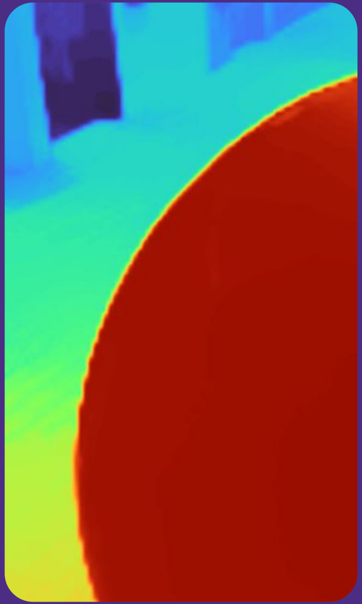
3D Photo

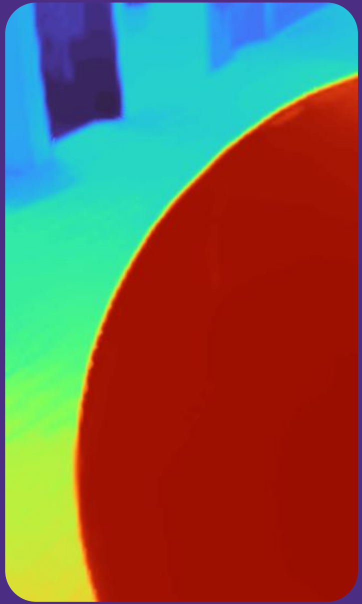
Projection mapping with depth mesh.

Dense Depth

Depth Texture - Antialiasing

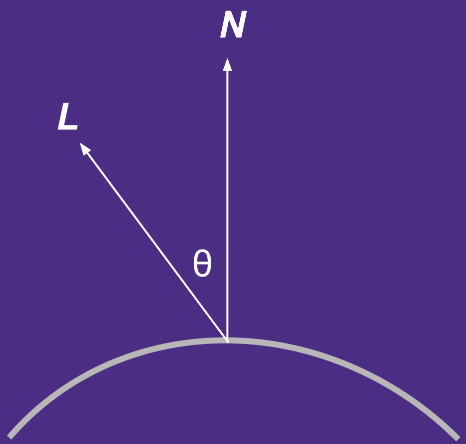






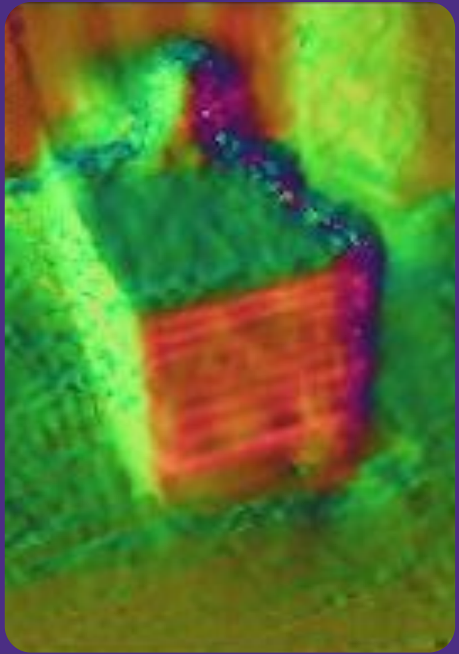
Dense Depth

Real-time relighting



Dense Depth

Why normal map does not work?



Dense Depth

Real-time relighting

Algorithm 3: Ray-marching-based Real-time Relighting.

Input : Depth map $\mathbf{D}$, the camera image $\mathbf{I}$, camera intrinsic matrix $\mathbf{K}$, L light sources $\mathbb{L} = \{\mathscr{L}^i, i \in L\}$ with each light's location $\mathbf{v}_{\mathscr{L}}$ and intensity in RGB channels $\phi_{\mathscr{L}}$.

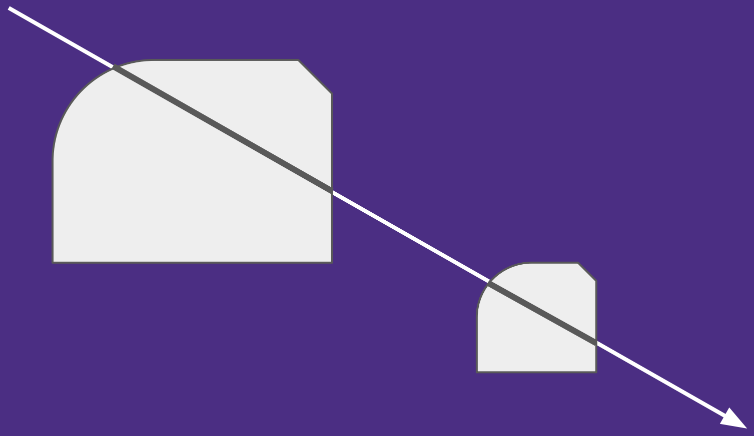
Output : Relighted image $\mathbf{O}$.

for *each image pixel* $\mathbf{p} \in$ *depth map* $\mathbf{D}$ *in parallel* **do**
- Sample $\mathbf{p}$'s depth value $d \leftarrow \mathbf{D}(\mathbf{p})$.
- Compute the corresponding 3D vertex $\mathbf{v_p}$ of the screen point $\mathbf{p}$ using the camera intrinsic matrix $\mathbf{v_p}$ with $\mathbf{K}$: $\mathbf{v_p} = \mathbf{D}(\mathbf{p}) \cdot \mathbf{K}^{-1}[\mathbf{p}, 1]$
- Initialize relighting coefficients of $\mathbf{v_p}$ in RGB: $\phi_\mathbf{p} \leftarrow \mathbf{0}$.
- **for** *each light* $\mathscr{L} \in$ *light sources* $\mathbb{L}$ **do**
  - Set the current photon coordinates $\mathbf{v}_o \leftarrow \mathbf{v_p}$.
  - Set the current photon energy $E_o \leftarrow 1$.
  - **while** $\mathbf{v}_o \neq \mathbf{v}_{\mathscr{L}}$ **do**
    - Compute the weighted distance between the photon to the physical environment $\Delta d \leftarrow \alpha|\mathbf{v}_o^{xy} - \mathbf{v}_{\mathscr{L}}^{xy}| + (1-\alpha)|\mathbf{v}_o^z - \mathbf{v}_{\mathscr{L}}^z|$, $\alpha = 0.5$.
    - Decay the photon energy: $E_o \leftarrow 95\% E_o$
    - Accumulate the relighting coefficients: $\phi_\mathbf{p} \leftarrow \phi_\mathbf{p} + \Delta d E_o \phi_{\mathscr{L}}$.
    - March the photon towards the light source: $\mathbf{v}_o \leftarrow \mathbf{v}_o + (\mathbf{v}_{\mathscr{L}} - \mathbf{v}_o)/S$, here $S = 10$, depending on the mobile computing budget.
  - **end**
- **end**
- Sample pixel's original color: $\Phi_\mathbf{p} \leftarrow \mathbf{I}(\mathbf{p})$.
- Apply relighting effect: $\mathbf{O}(\mathbf{p}) \leftarrow \gamma \cdot |\mathbf{0.5} - \phi_\mathbf{p}| \cdot \Phi_\mathbf{p}^{\mathbf{1.5} - \phi_\mathbf{p}} - \Phi_\mathbf{p}$, here $\gamma \leftarrow 3$.

end

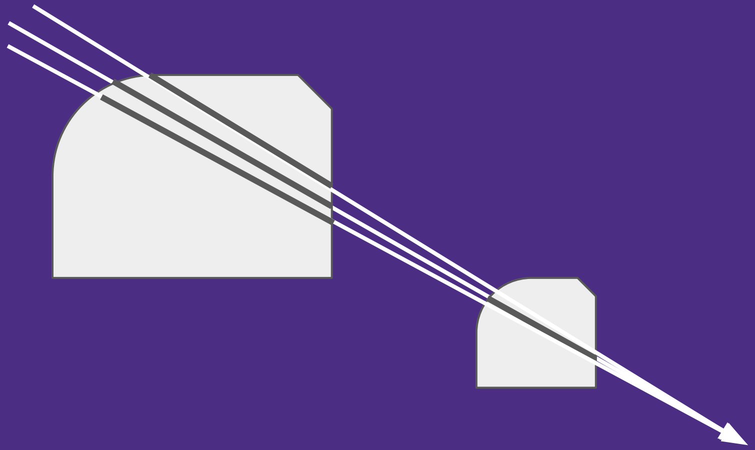
Dense Depth

Real-time relighting



Dense Depth

Real-time relighting



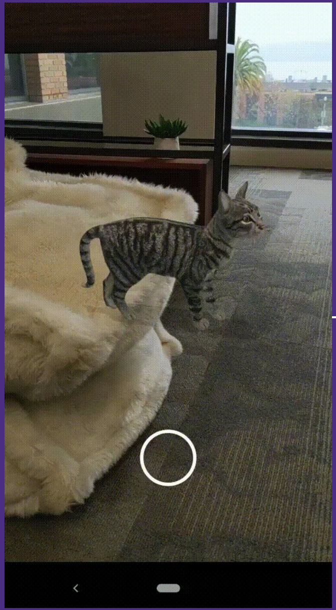
go/realtime-relighting, go/relit

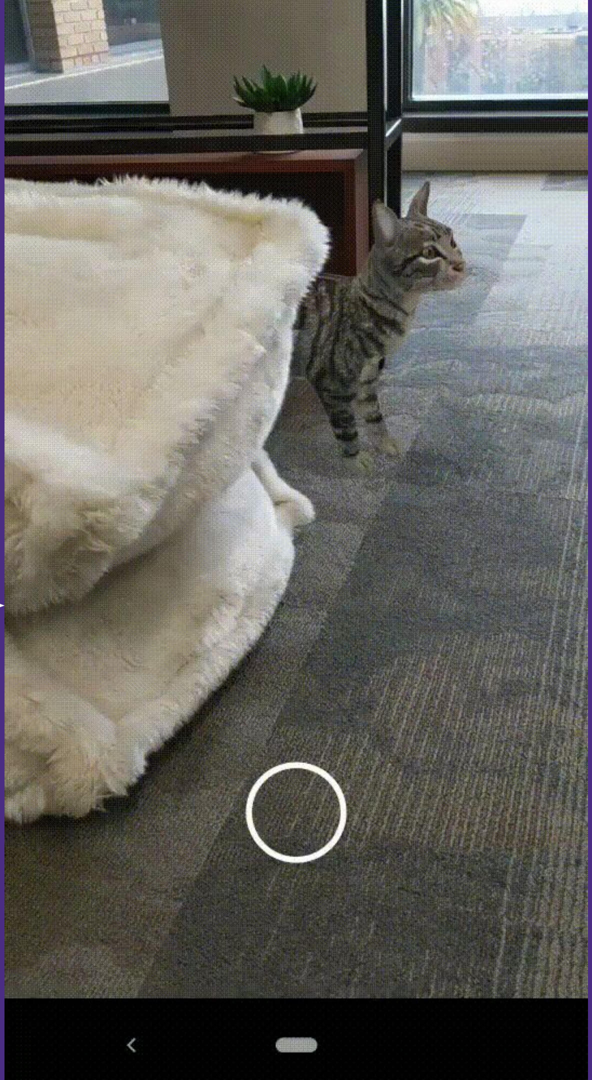
Dense Depth

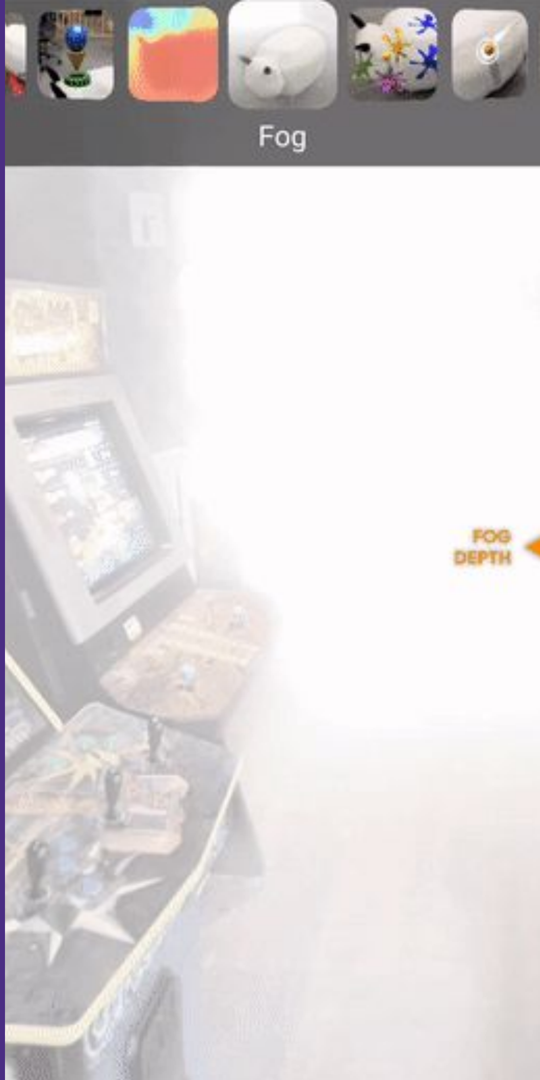
Wide-aperture effect

Dense Depth

Occlusion-based rendering







Experiments

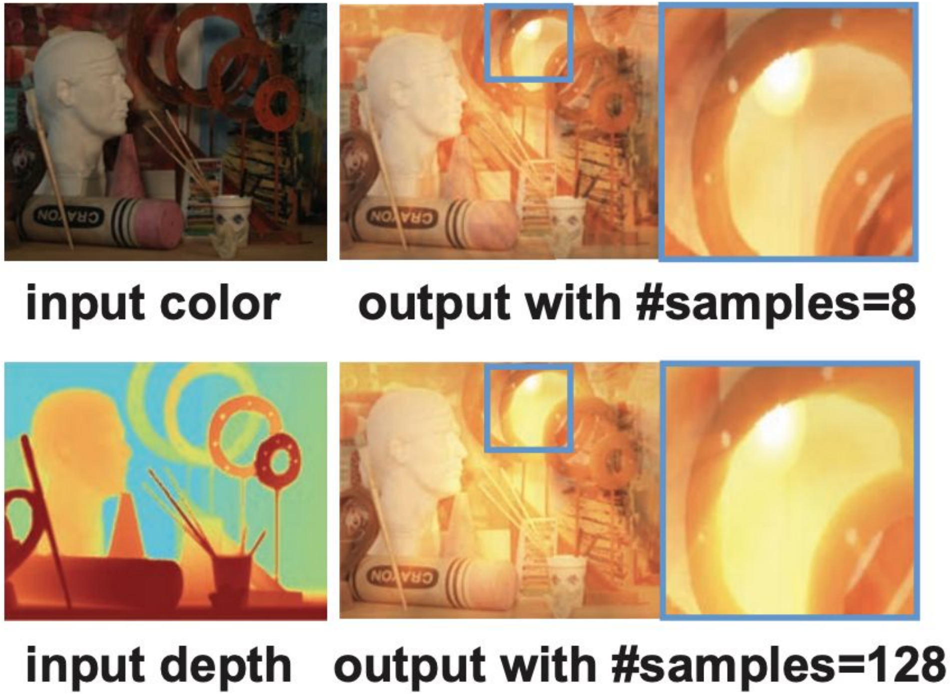
DepthLab minimum viable application

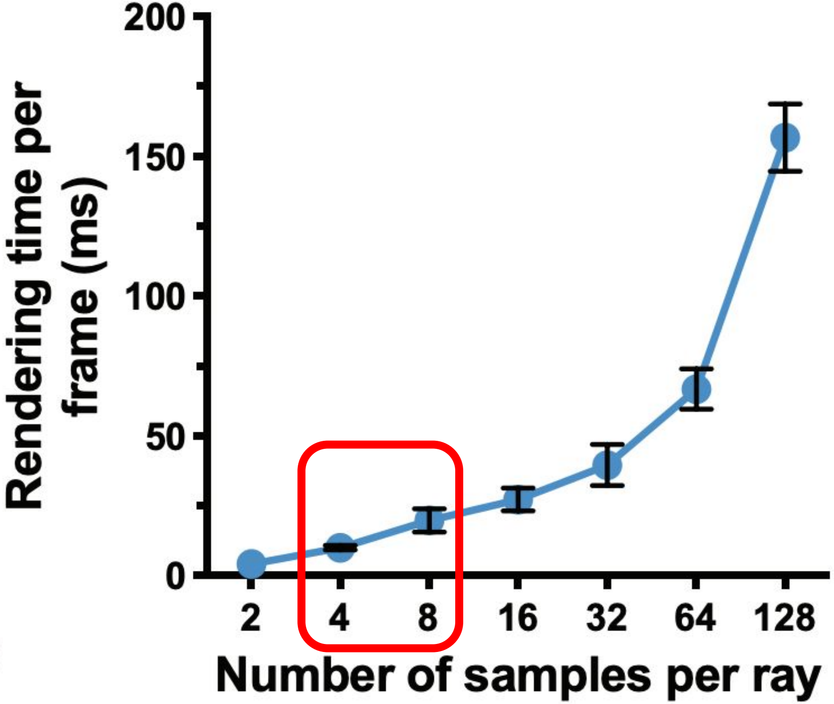
Experiments

General Profiling of MVP

Procedure	Timings (ms)
DepthLab's overall processing and rendering in Unity	8.32
DepthLab's data structure update and GPU uploading	1.63
Point Depth: normal estimation algorithm	< 0.01
Surface Depth: depth mesh update algorithm	2.41
Per-pixel Depth: visualization with single texture fetch	0.32

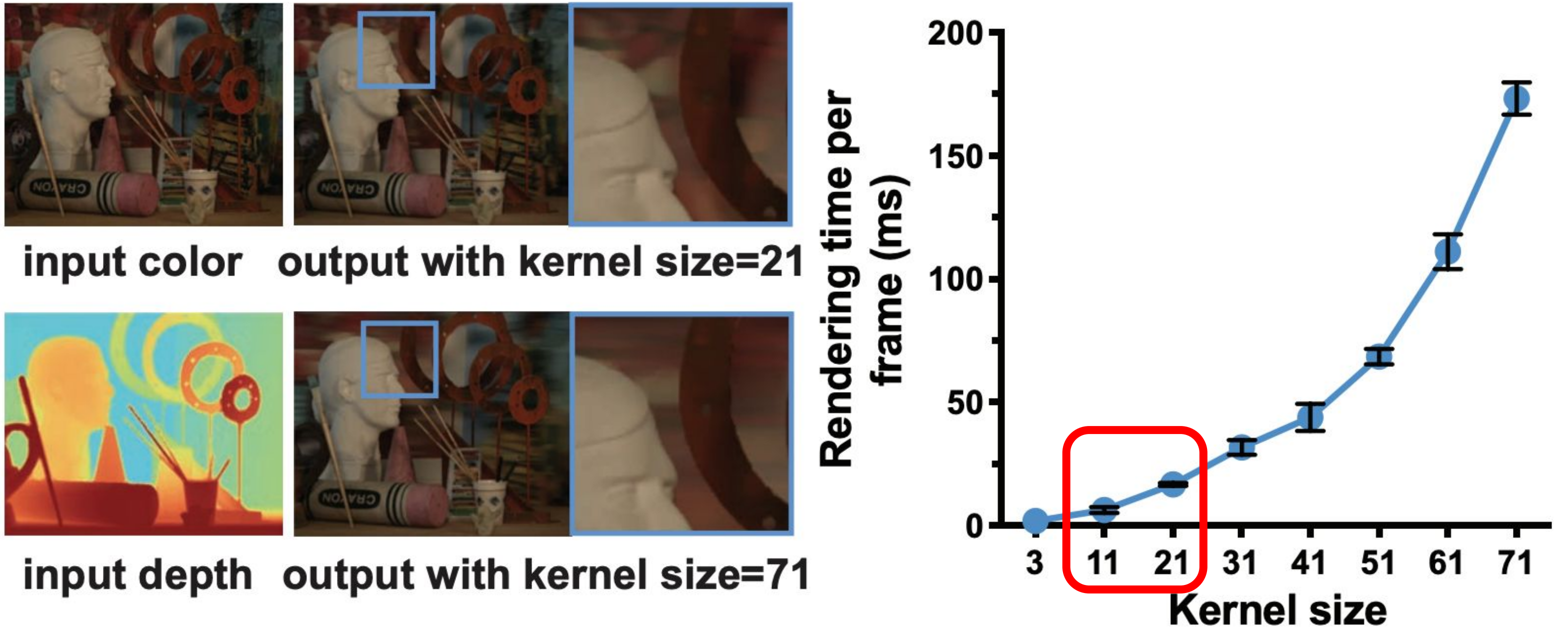
Experiments
Relighting





Experiments

Aperture effects



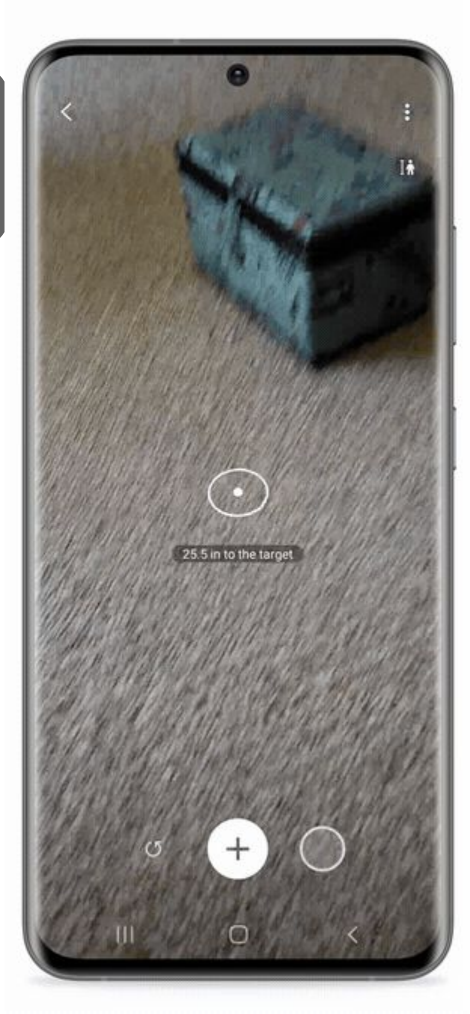
(a) examples of aperture effects (b) performance benchmark

Discussion

Deployment with partners

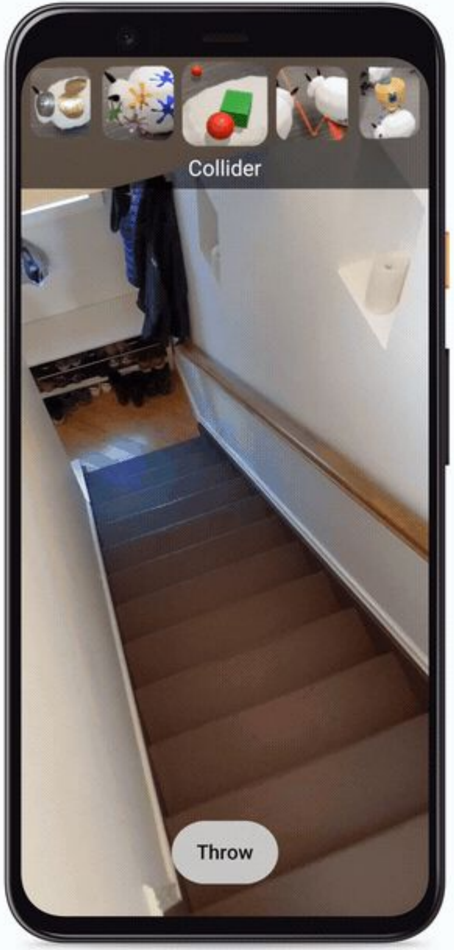
Discussion

Deployment with partners



Discussion

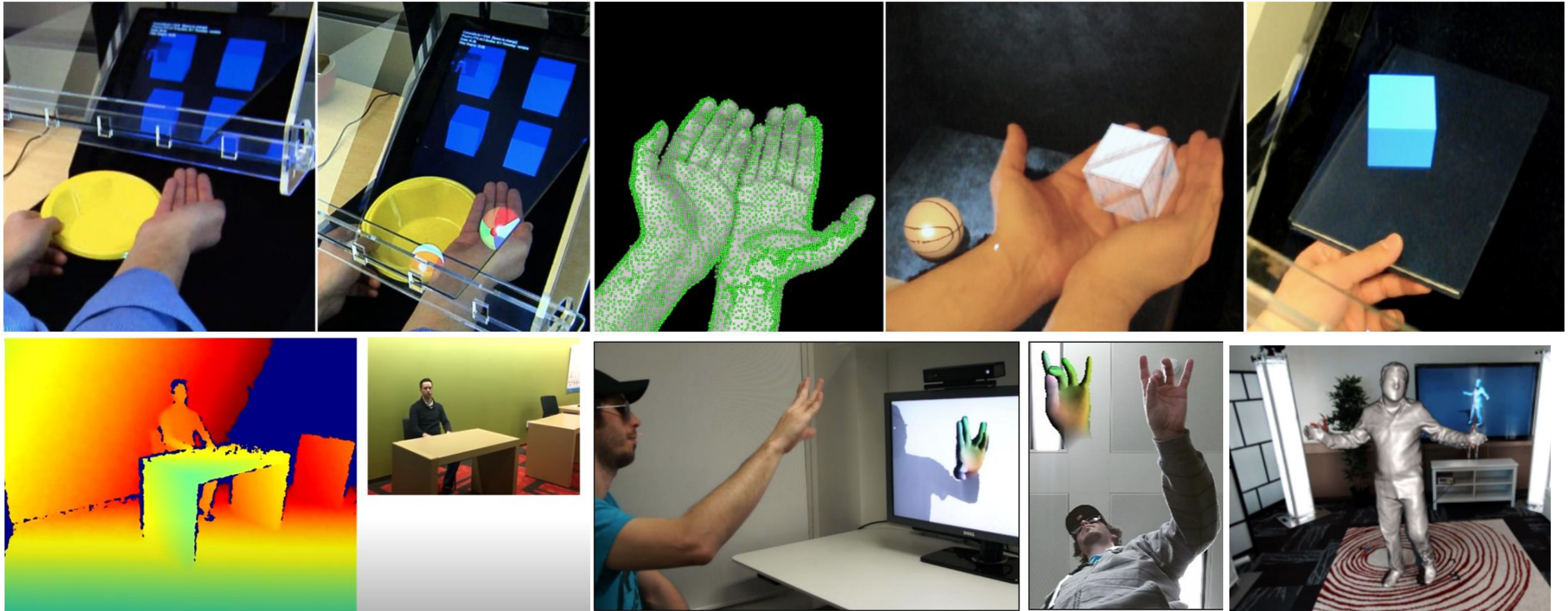
Deployment with partners



Limitations

Design space of dynamic depth

Dynamic Depth? HoloDesk, HyperDepth, Digits, Holoportation for mobile AR?

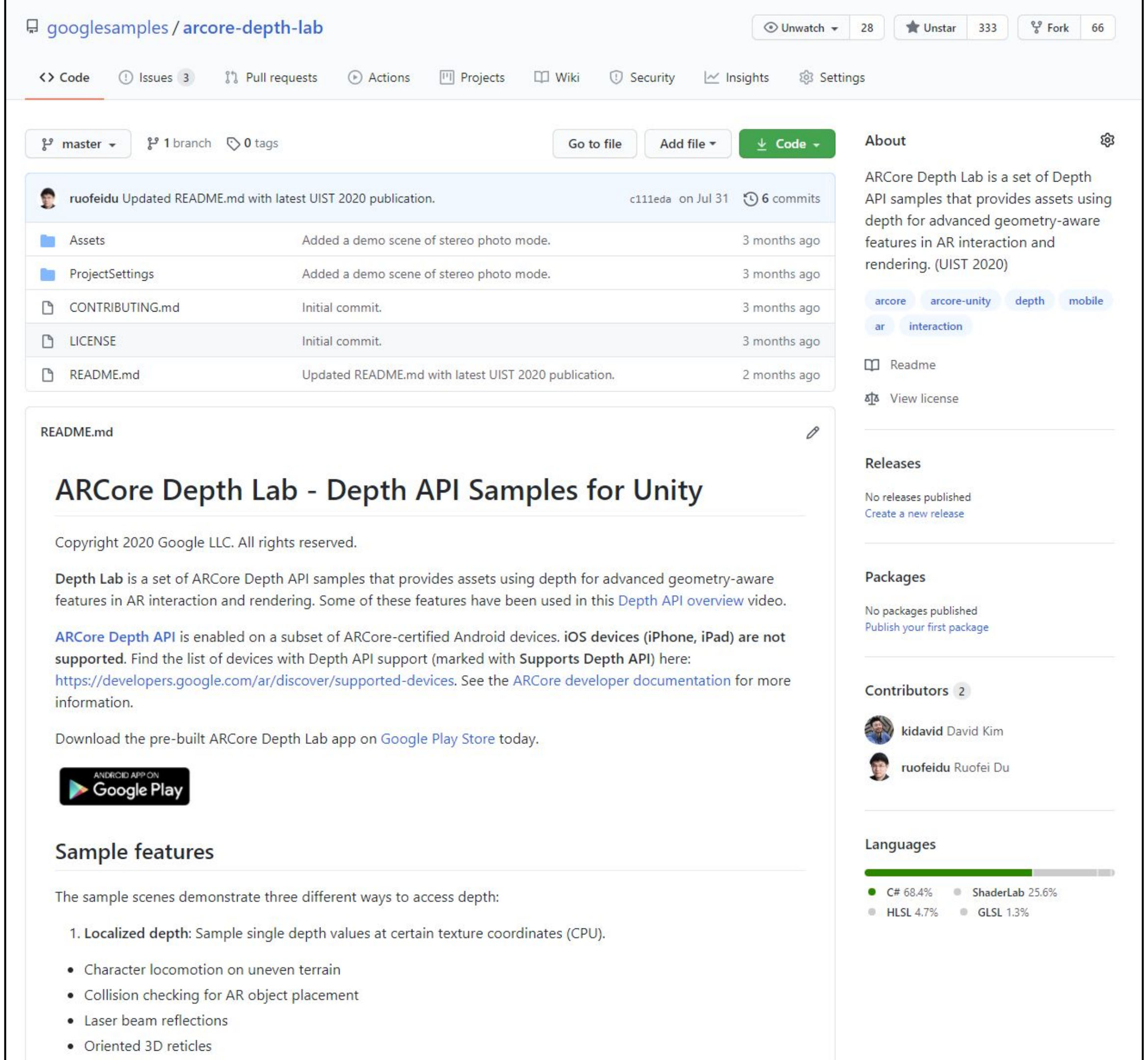


Envision

Design space of dynamic depth

GitHub

Please feel free to fork!



googlesamples / arcore-depth-lab

Unwatch 28 | Unstar 333 | Fork 66

Code | Issues 3 | Pull requests | Actions | Projects | Wiki | Security | Insights | Settings

master | 1 branch | 0 tags | Go to file | Add file | Code

ruofeidu Updated README.md with latest UIST 2020 publication. c111eda on Jul 31 | 6 commits

Assets	Added a demo scene of stereo photo mode.	3 months ago
ProjectSettings	Added a demo scene of stereo photo mode.	3 months ago
CONTRIBUTING.md	Initial commit.	3 months ago
LICENSE	Initial commit.	3 months ago
README.md	Updated README.md with latest UIST 2020 publication.	2 months ago

README.md

ARCore Depth Lab - Depth API Samples for Unity

Depth Lab is a set of ARCore Depth API samples that provides assets using depth for advanced geometry-aware features in AR interaction and rendering. Some of these features have been used in this Depth API overview video.

ARCore Depth API is enabled on a subset of ARCore-certified Android devices. **iOS devices (iPhone, iPad) are not supported**. Find the list of devices with Depth API support (marked with **Supports Depth API**) here: https://developers.google.com/ar/discover/supported-devices. See the ARCore developer documentation for more information.

Download the pre-built ARCore Depth Lab app on Google Play Store today.

ANDROID APP ON Google Play

Sample features

The sample scenes demonstrate three different ways to access depth:

1. **Localized depth**: Sample single depth values at certain texture coordinates (CPU).

- Character locomotion on uneven terrain
- Collision checking for AR object placement
- Laser beam reflections
- Oriented 3D reticles

About

ARCore Depth Lab is a set of Depth API samples that provides assets using depth for advanced geometry-aware features in AR interaction and rendering. (UIST 2020)

arcore | arcore-unity | depth | mobile | ar | interaction

Readme

View license

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package

Contributors 2

kidavid David Kim

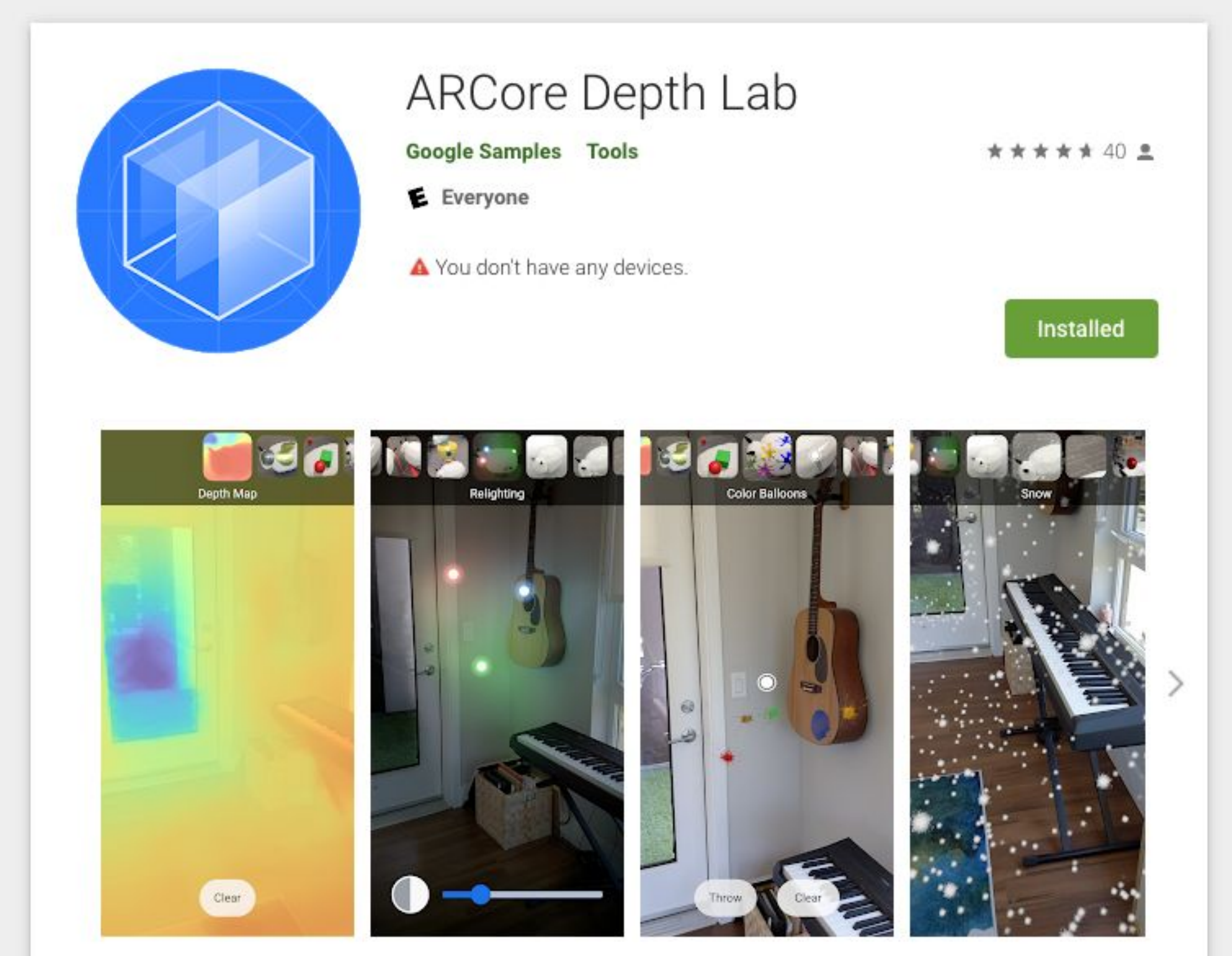
ruofeidu Ruofei Du

Languages

C# 68.4% | ShaderLab 25.6% | HLSL 4.7% | GLSL 1.3%

Play Store

Try it yourself!



DepthLab: Real-time 3D Interaction with Depth Maps for Mobile Augmented Reality

Ruofei Du, Eric Turner, Maksym Dzitsiuk, Luca Prasso, Ivo Duarte,
Jason Dourgarian, Joao Afonso, Jose Pascoal, Josh Gladstone, Nuno Cruces,
Shahram Izadi, Adarsh Kowdle, Konstantine Tsotsos, David Kim

Google | ACM UIST 2020

Thank you!

DepthLab | UIST 2020

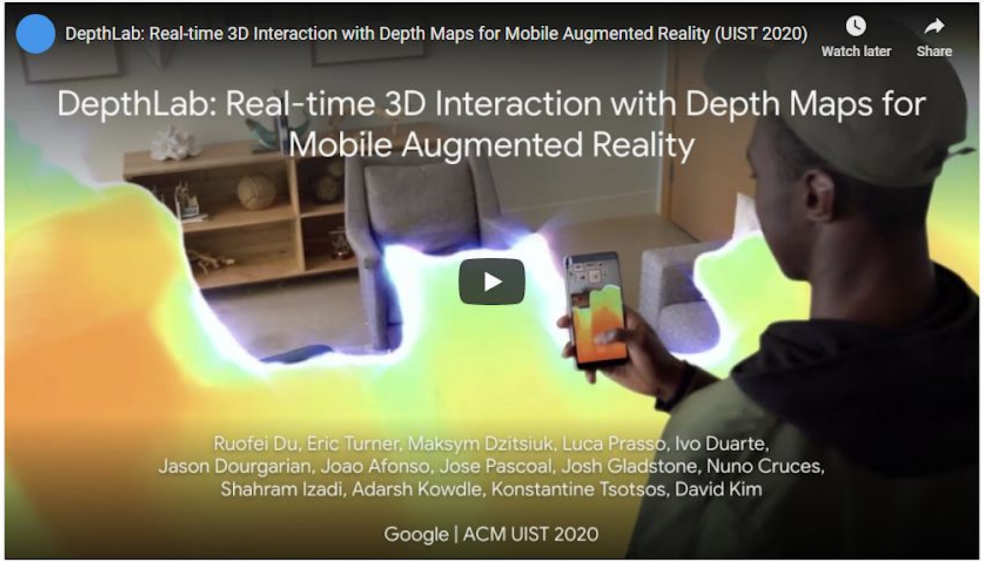
DEPTHLAB: REAL-TIME 3D INTERACTION WITH DEPTH MAPS FOR MOBILE AUGMENTED REALITY

ACM UIST 2020

Download PDF (6 MB) or Low-Res PDF (4 MB)

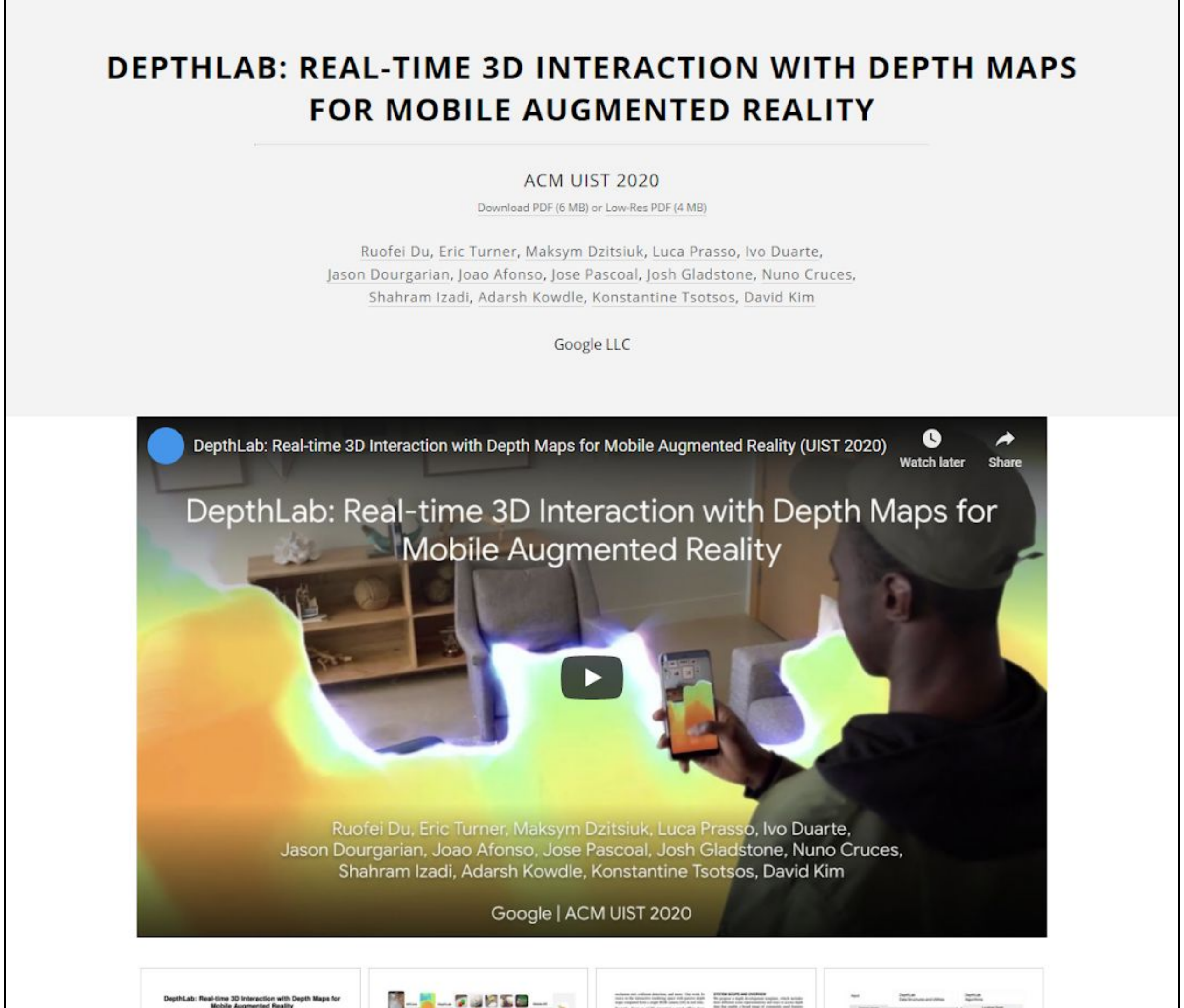
Ruofei Du, Eric Turner, Maksym Dzitsiuk, Luca Prasso, Ivo Duarte,
Jason Dourgarian, Joao Afonso, Jose Pascoal, Josh Gladstone, Nuno Cruces,
Shahram Izadi, Adarsh Kowdle, Konstantine Tsotsos, David Kim

Google LLC



Demo

DepthLab | UIST 2020



Project Geollery.com: Reconstructing a Live Mirrored World With Geotagged Social Media

Ruofei Du[†], David Li[†], and Amitabh Varshney
{ruofei, dli7319, varshney}@umiacs.umd.edu | www.Geollery.com | Web3D 2019, Los Angeles, USA

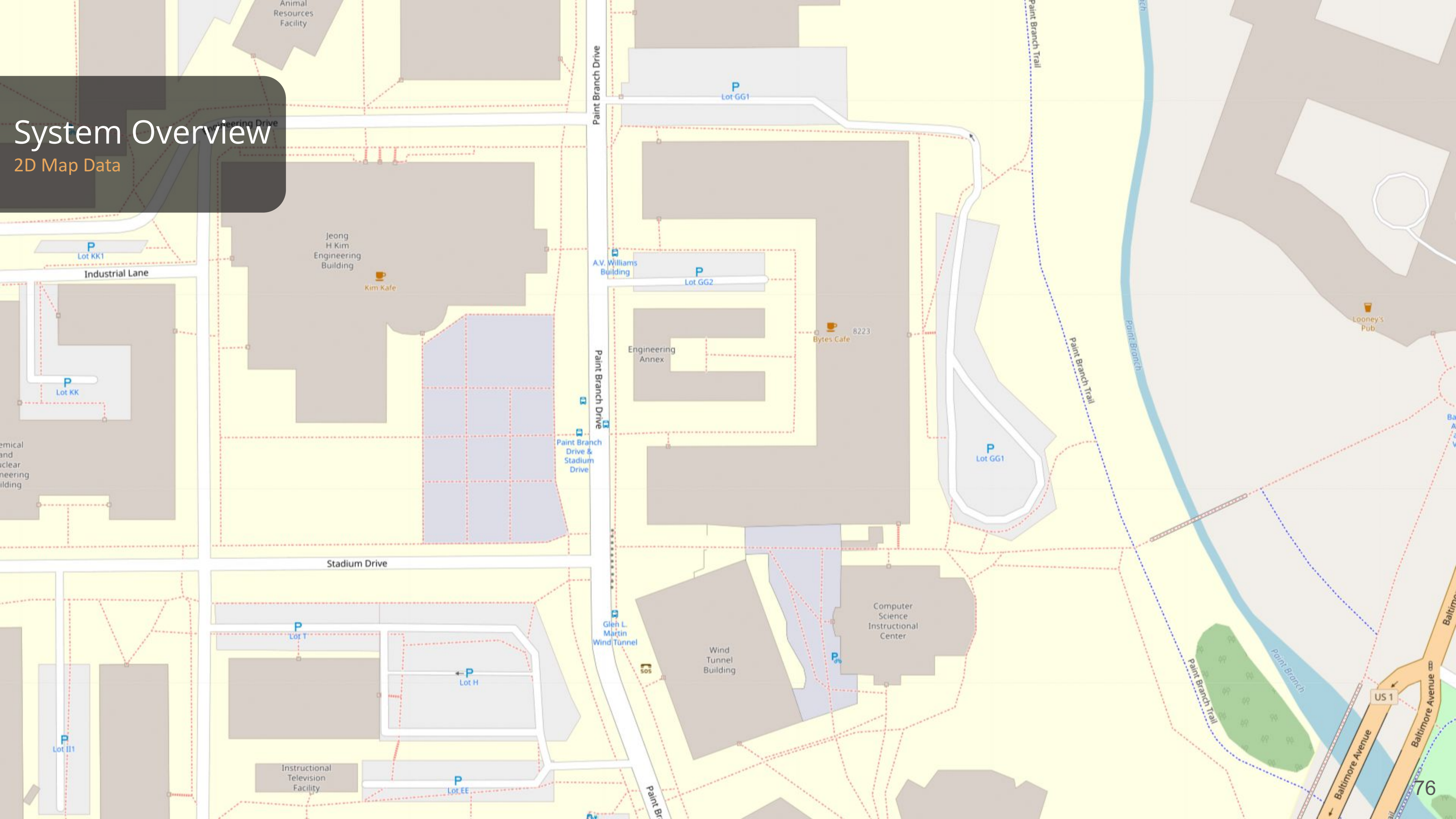
UNIVERSITY OF MARYLAND
UMIACS
THE AUGMENTARIUM
VIRTUAL AND AUGMENTED REALITY LAB
AT THE UNIVERSITY OF MARYLAND
COMPUTER SCIENCE
UNIVERSITY OF MARYLAND, COLLEGE PARK

Geollery.com

v2: a major leap

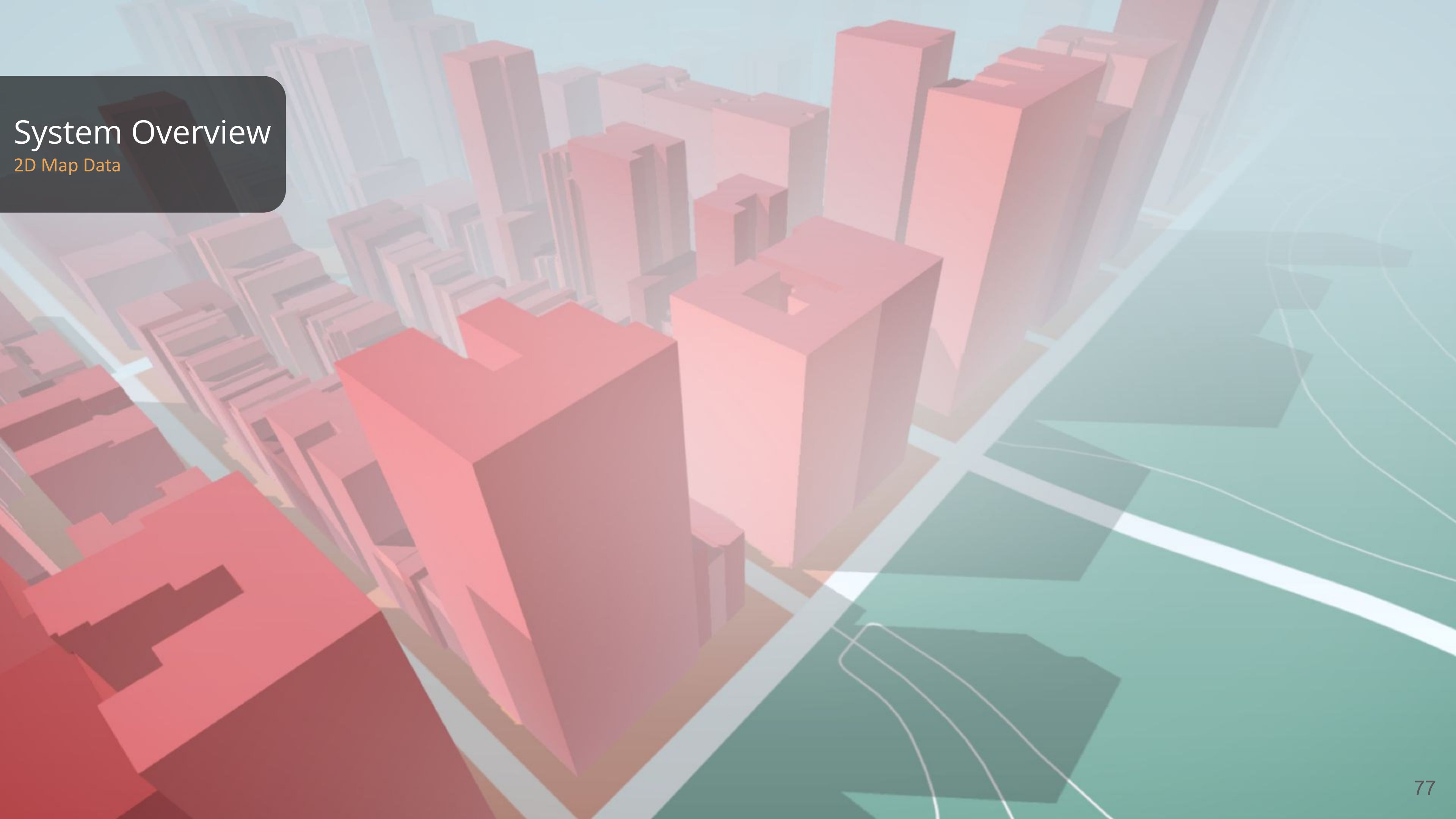
System Overview

2D Map Data



System Overview

2D Map Data



System Overview

+Avatar +Trees +Clouds

System Overview

+Avatar +Trees +Clouds +Night

System Overview

Street View Panoramas

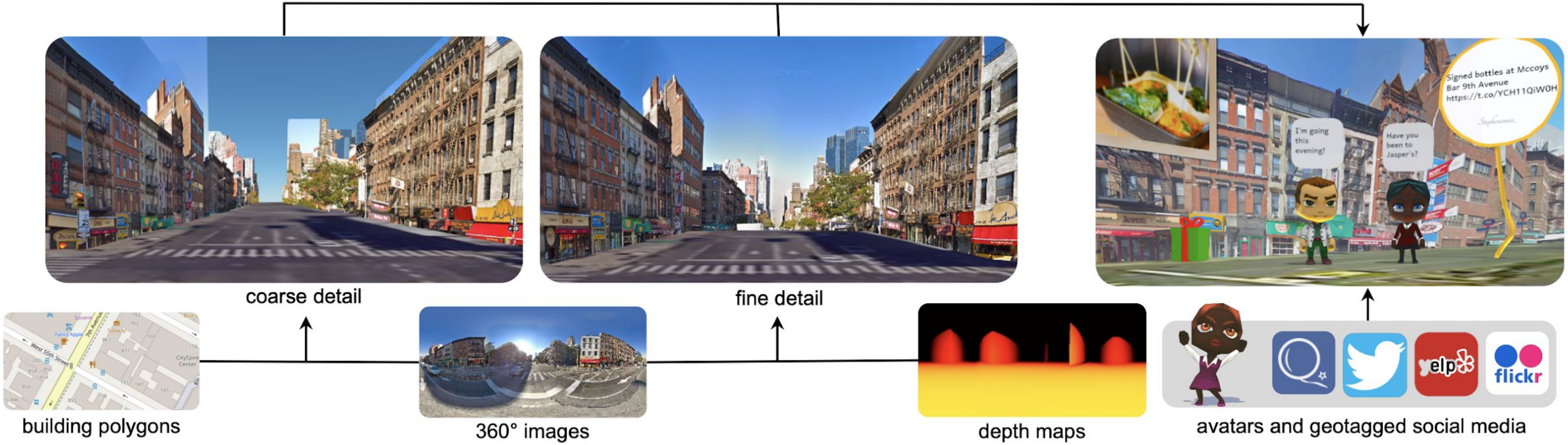
System Overview

Street View Panoramas

System Overview

Street View Panoramas

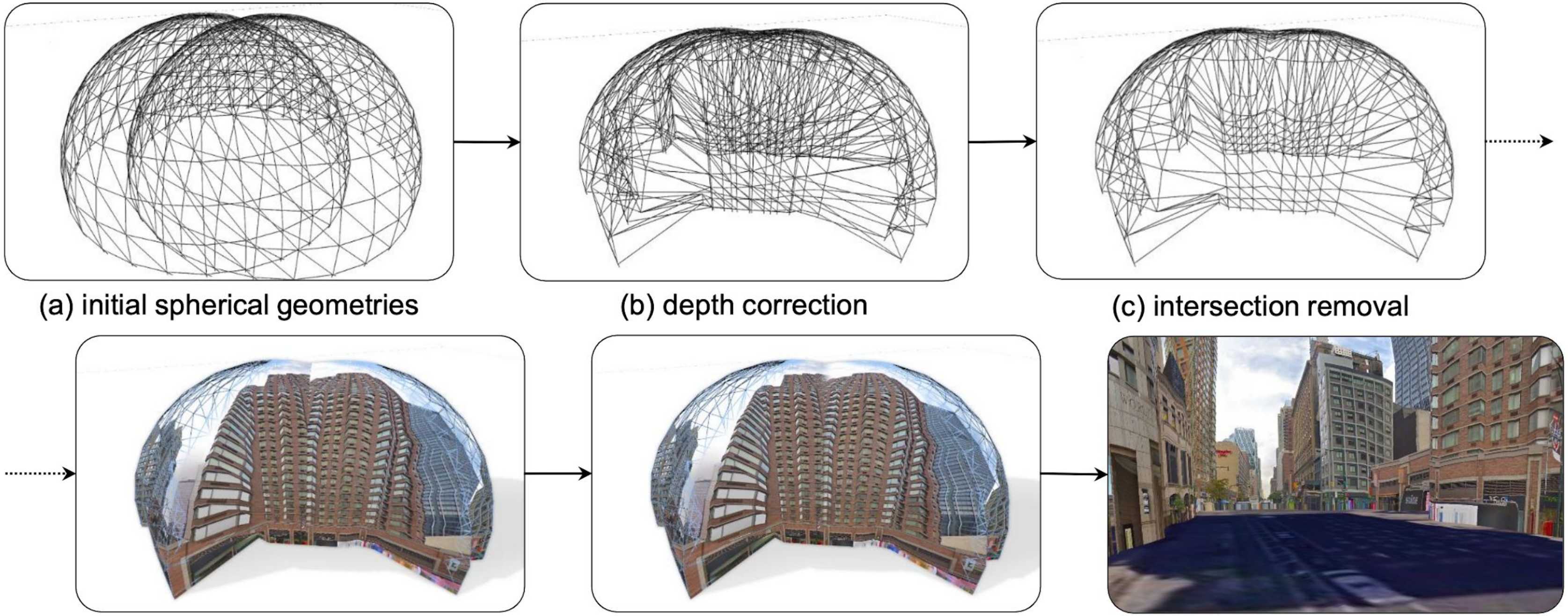
System Overview
Geollery Workflow



All data we used is publicly and widely available on the Internet.

Rendering Pipeline

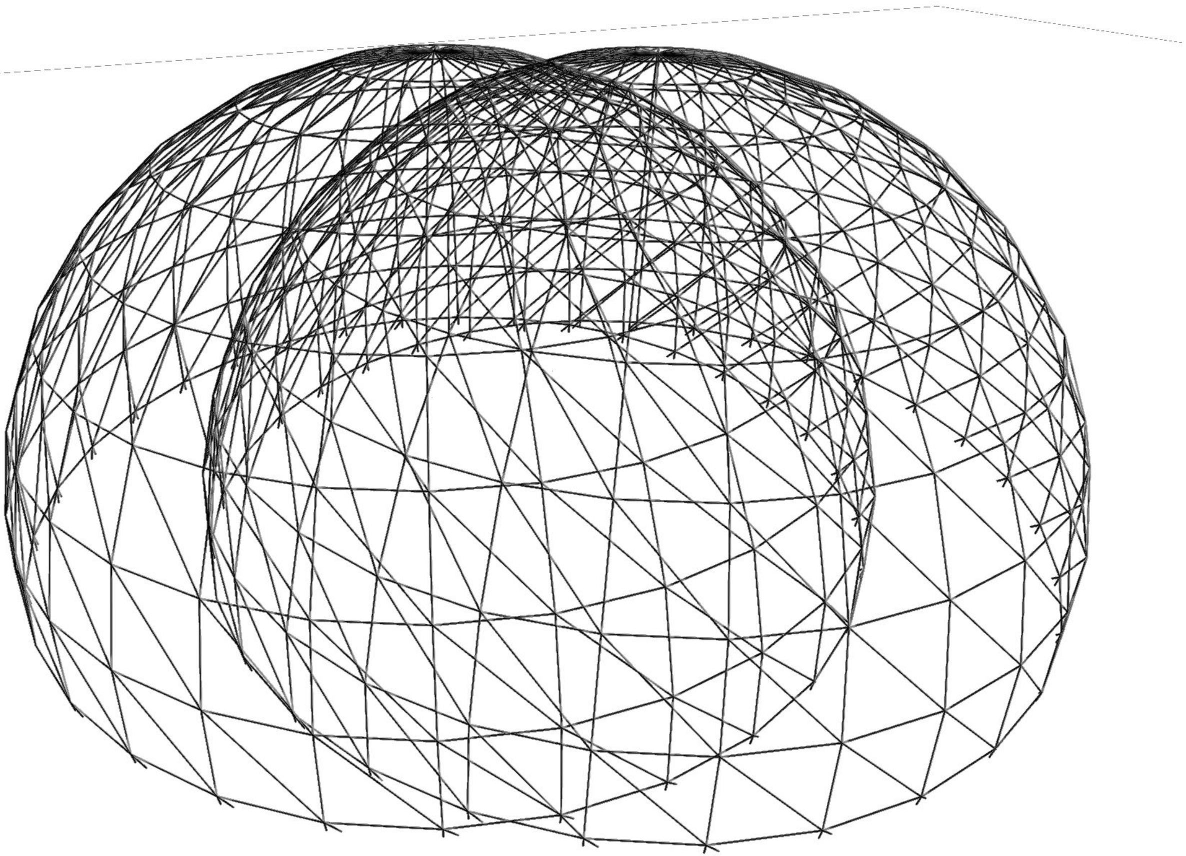
Close-view Rendering



(a) initial spherical geometries
(b) depth correction
(c) intersection removal
(d) texturing individual geometry
(e) texturing with alpha blending
(f) rendering results in fine detail

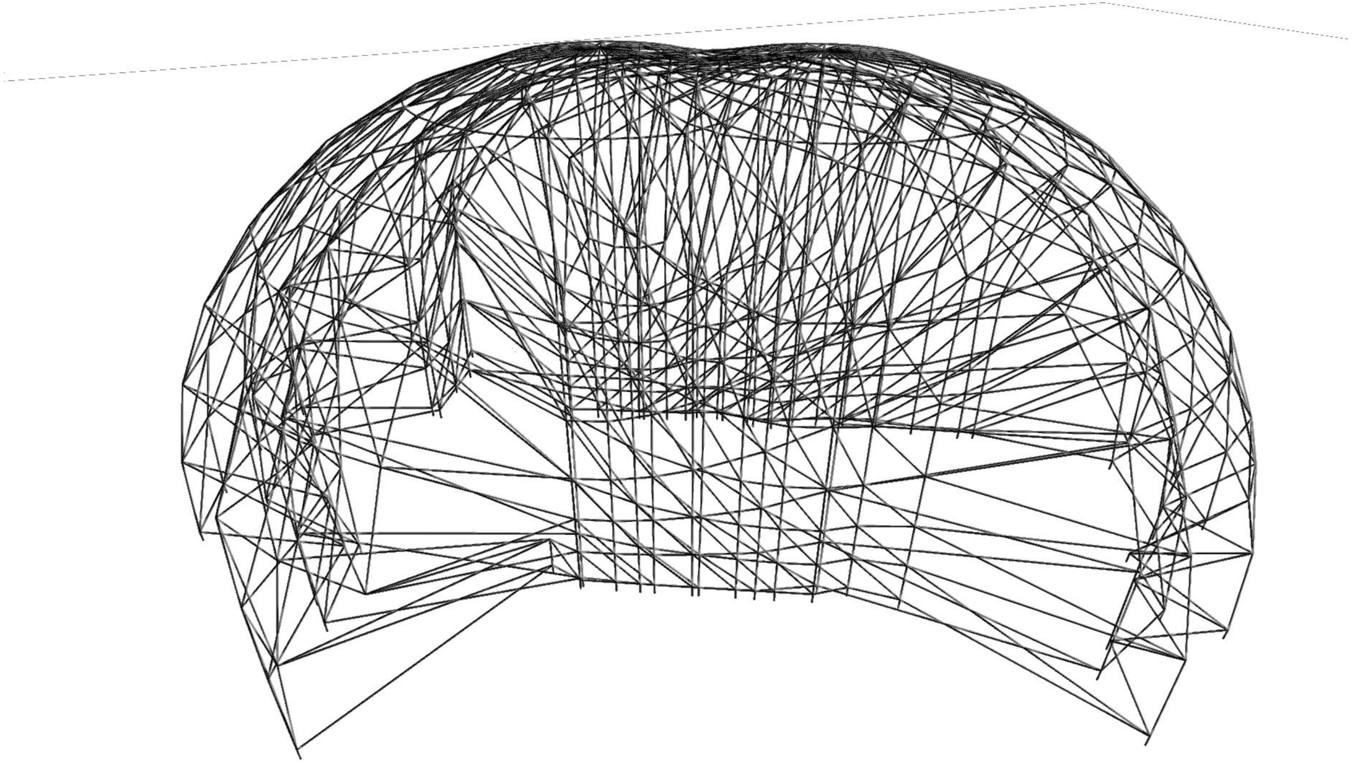
Rendering Pipeline

Initial spherical geometries

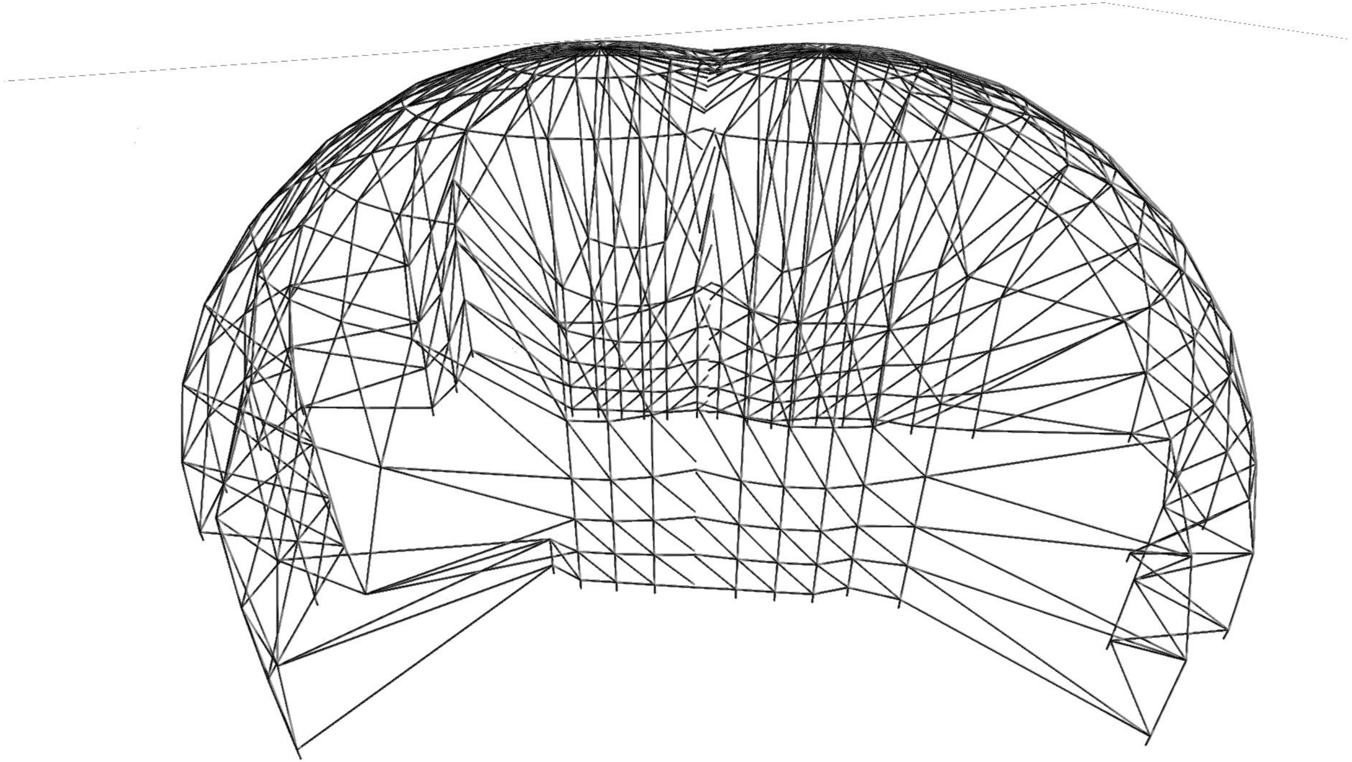


Rendering Pipeline

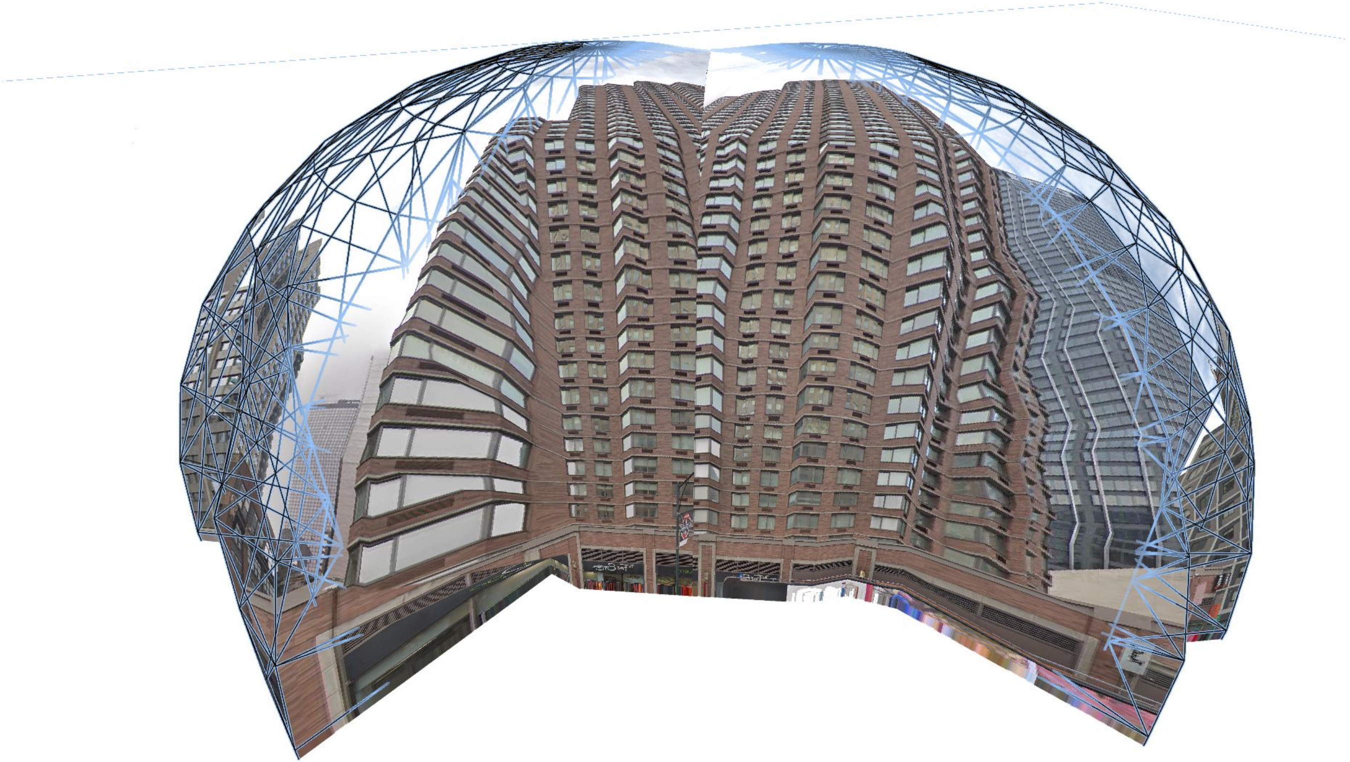
Depth correction



Rendering Pipeline
Intersection removal

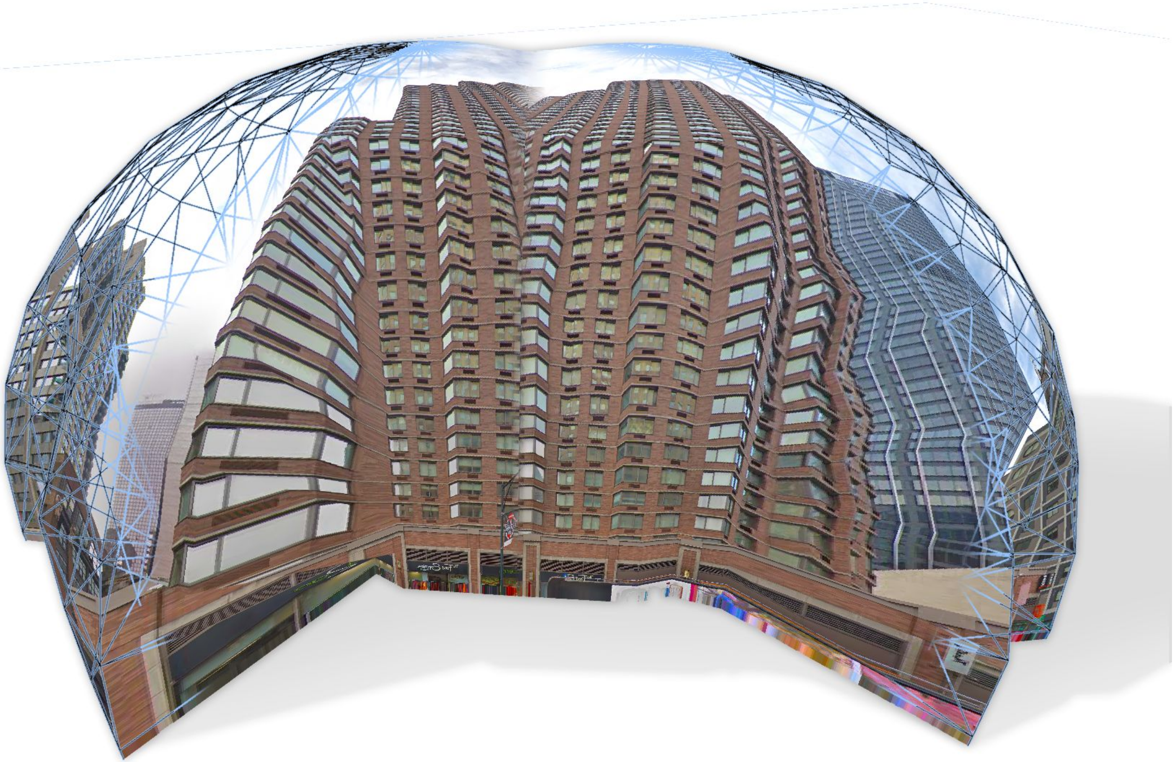


Rendering Pipeline
Texturing individual geometry



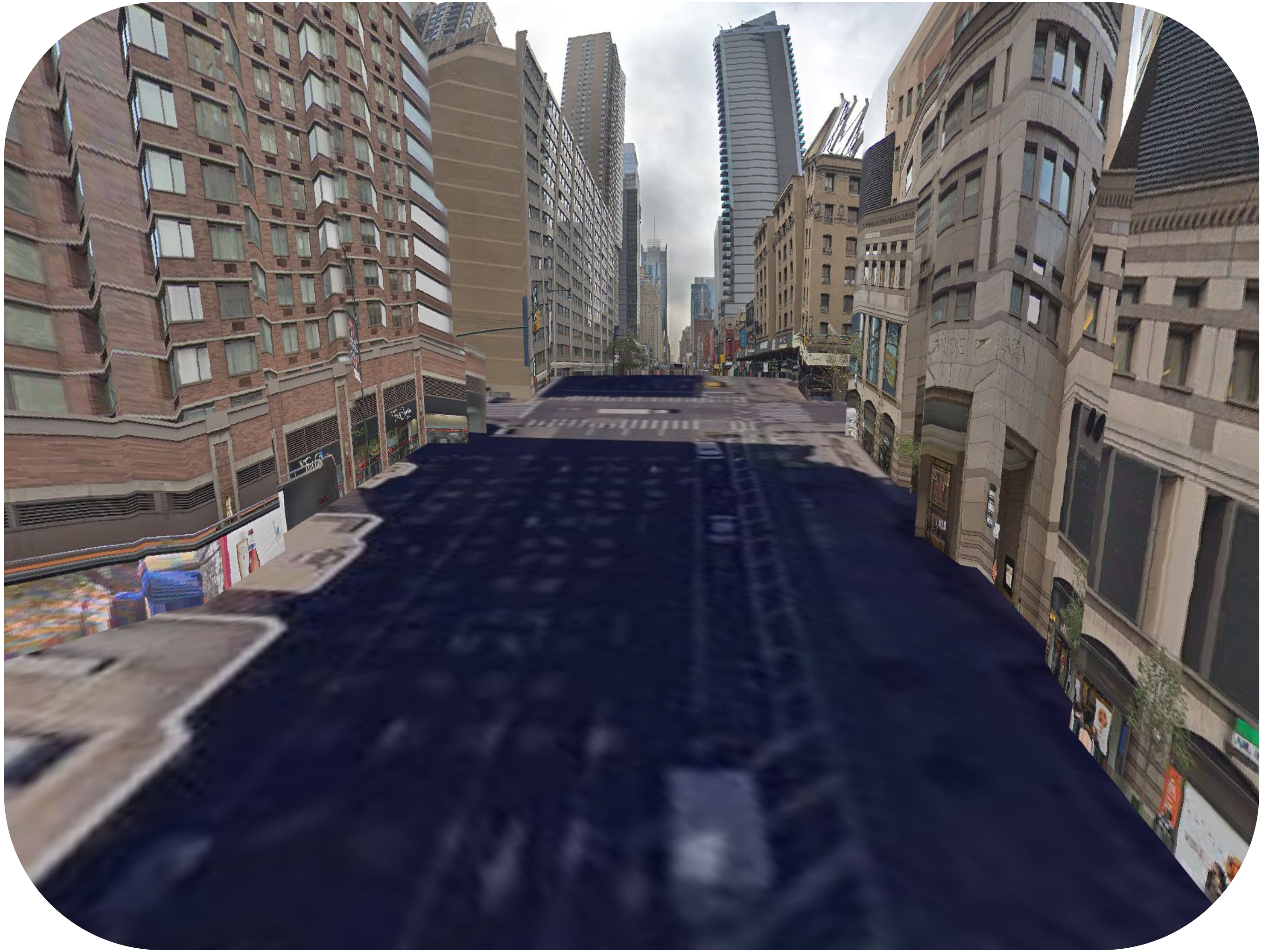
Rendering Pipeline

Texturing with alpha blending



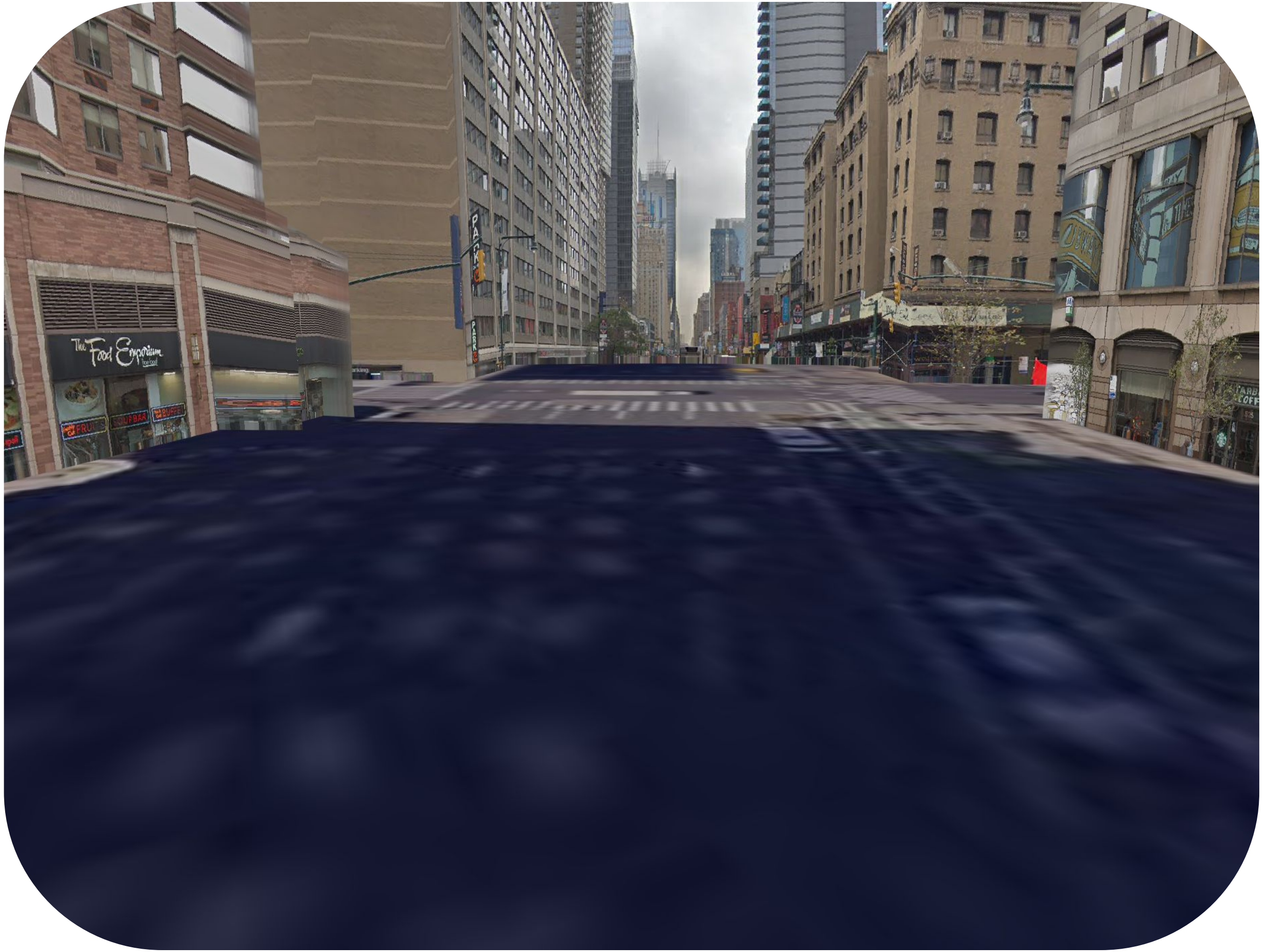
Rendering Pipeline

Rendering result in the fine detail



Rendering Pipeline

Rendering result in the fine detail



Rendering Pipeline

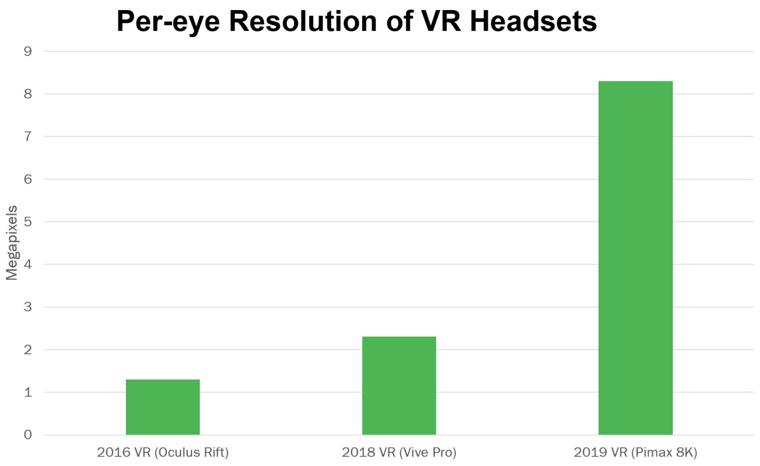
Rendering result in the fine detail

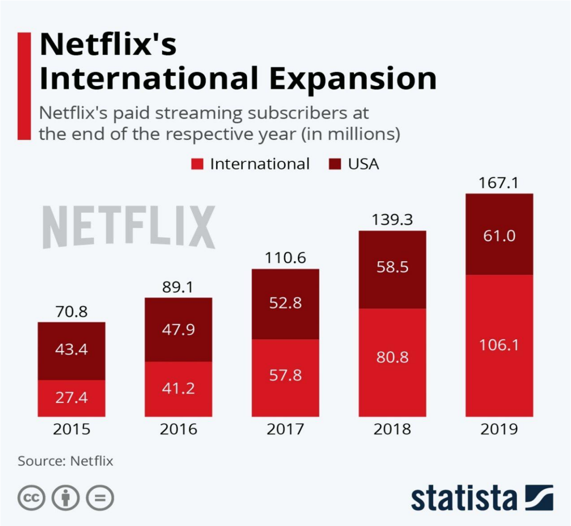
Rendering Pipeline
Experimental Features


A Log-Rectilinear Transformation for Foveated
360-Degree Video Streaming
David Li, Ruofei Du, Adharsh Babu, Camelia Brumar, and Amitabh Varshney
IEEE Transactions on Visualization and Computer Graphics (TVCG), 2021.

Introduction
VR headset & video streaming

- 360 Cameras and VR headsets are increasing in resolution.
- Video streaming is quickly increasing in popularity.





Introduction
VR + eye tracking

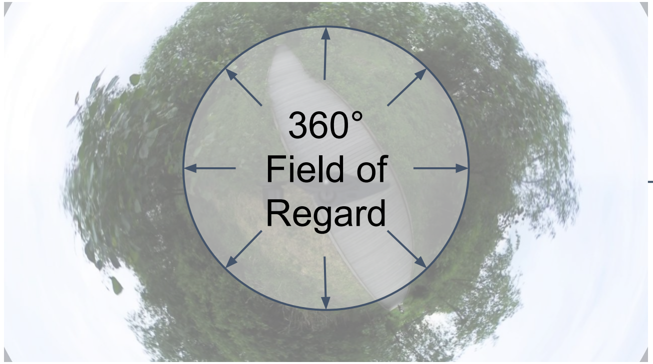
- Commercial VR headsets are getting eye-tracking capabilities.

HTC Vive Eye

Varjo VR-3

Fove

- 360 cameras capture the scene in every direction with a full 360 degree spherical field of regard.
- These videos are typically stored in the equirectangular projection parameterized by spherical coordinates (θ, φ).



Scene

Captured 360 Video

- When viewed in a VR headset, 360° videos cover the entire field-of-view for more immersive experiences.
- However, transmitting the full field-of-regard either has worse perceived quality or requires far more bandwidth than for conventional videos.

Captured 360 Video

Projection to Field of View

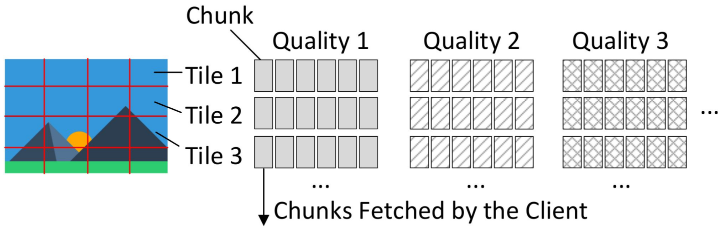
- Existing work in 360° streaming focuses on viewport dependent streaming by using tiling to transmit only visible regions based on the user's head rotation.


Flare: Practical Viewport-Adaptive 360-Degree Video Streaming for Mobile Devices

Pano: Optimizing 360° Video Streaming with a Better Understanding of Quality Perception

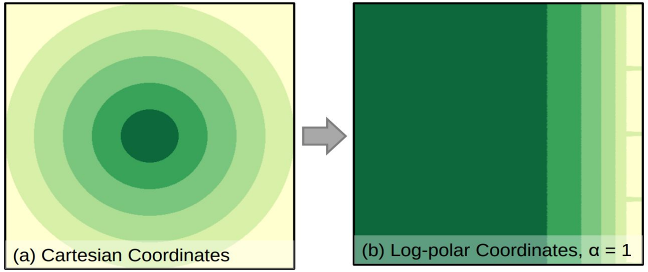
Adaptive 360 VR Video Streaming: Divide and Conquer!

360° Innovations for Panoramic Video Streaming



Tiling Illustration
Image from (Liu *et al. with Prof. Bo*, 2017)

- Foveated rendering renders the fovea region of the viewport at a high-resolution and the peripheral region at a lower resolution.
- Kernel Foveated Rendering (Meng *et al.,* PACMCGIT 2018) uses a log-polar transformation to render foveated images in real-time.



Log-polar Transformation,
Image from (Meng *et al.*, 2018)

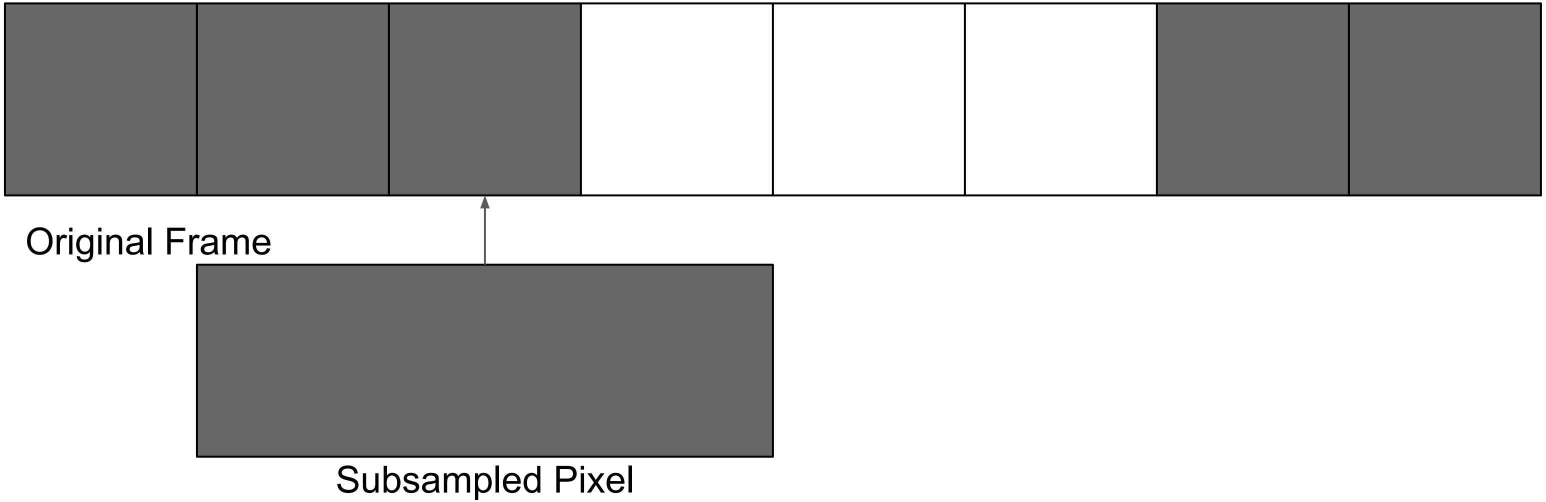
- Applying log-polar subsampling to videos results in flickering and aliasing artifacts in the foveated video.

Research Questions

- Can foveation techniques from rendering be used to optimize 360 video streaming?
- How can we reduce foveation artifacts by leveraging the full original video frame?

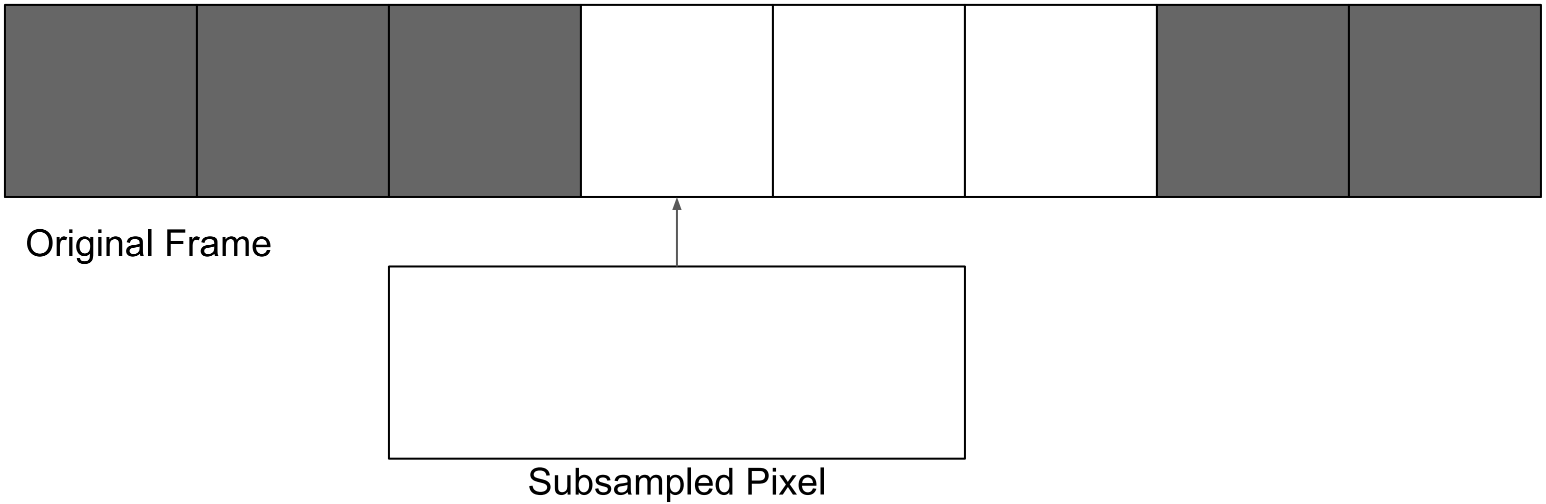
Log-Polar Foveated Streaming

- Artifacts are caused by subsampling of the original video frame.



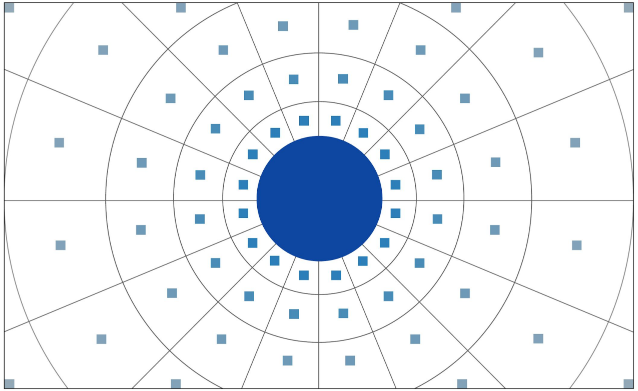
Log-Polar Foveated Streaming

- Artifacts are caused by subsampling of the original video frame.



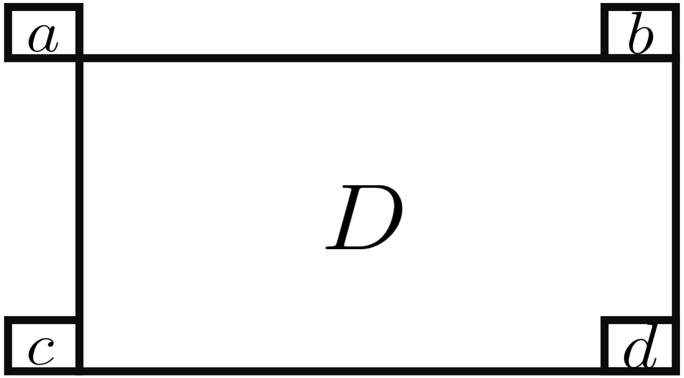
Log-Polar Foveated Streaming

- Subsampled pixels should represent an average over an entire region of the original video frame.
- Computationally, this would take O(region size) time to compute for each sample.



Summed-Area Tables

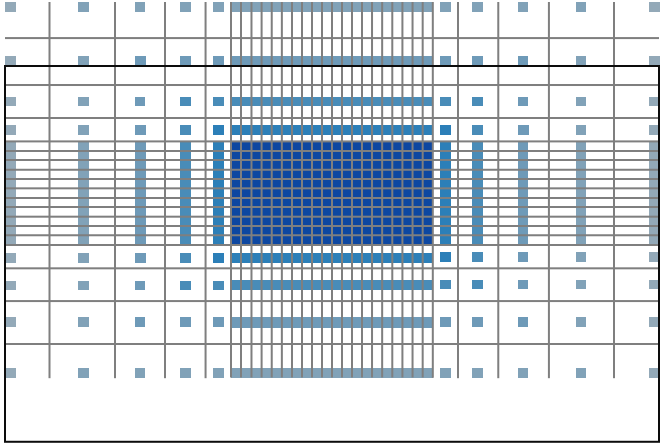
- One way to compute averages quickly is using summed-area tables, also known as integral images.
- Sampling a summed area table only takes O(1) time.



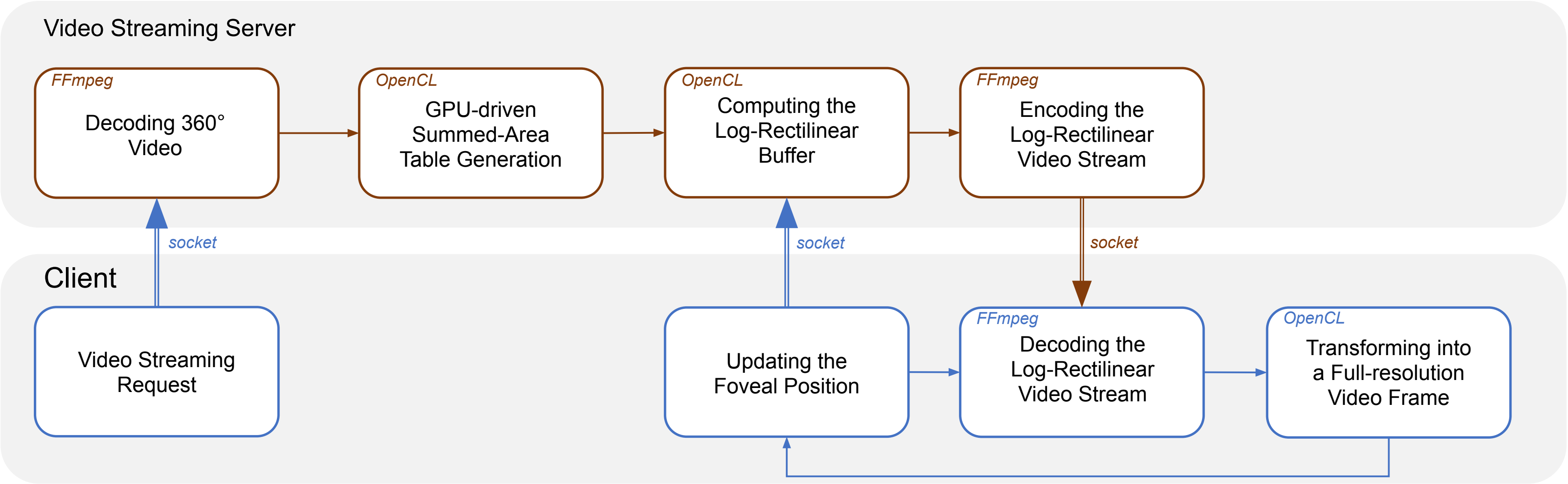
$$\mathrm{Sum}(D) = a - b - c + d$$

Log-Rectilinear Transformation

- Apply exponential drop off along x-axis and y-axis independently.
- Rectangular regions allow the use of summed area tables for subsampling.
- A one-to-one mapping near the focus region preserves the resolution of the original frame.



Foveated Streaming



Qualitative Results

- Shown with gaze at the center of the viewport

Quantitative Results

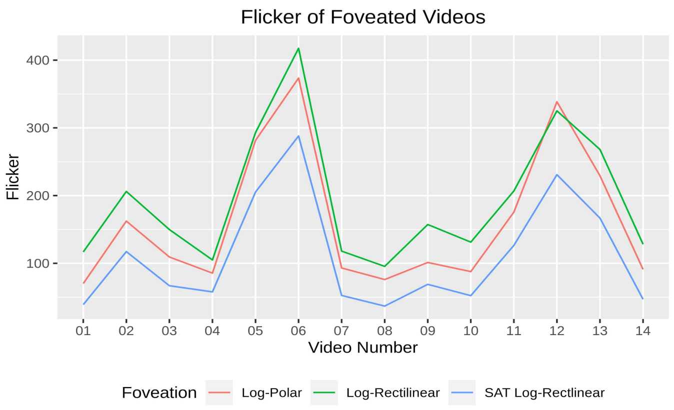
We perform quantitative evaluations comparing the log-rectilinear transformation and the log-polar transformation in 360° video streaming.

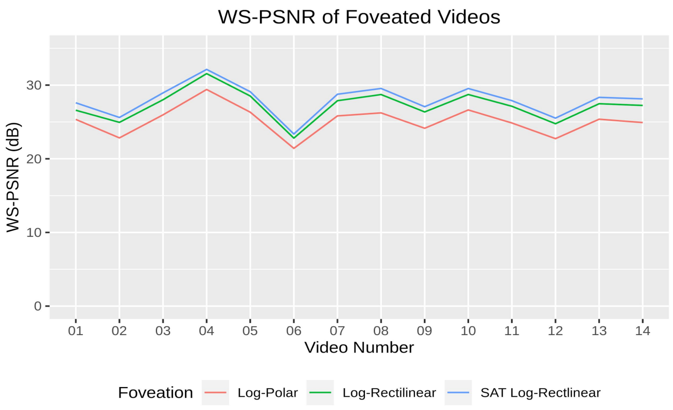
- Performance overhead of summed-area tables.
- Full-frame quality.
- Bandwidth usage.

Sampling Method	Decoding (ms)	Processing (ms)	Sampling (ms)	Encoding (ms)	Total (ms)
Log-Polar	6.14	1.91	0.55	2.86	11.46
Log-Rectilinear	6.13	1.91	0.53	2.85	11.43
SAT Log-Rectilinear	6.14	3.00	0.46	2.84	12.44

Quantitative Results

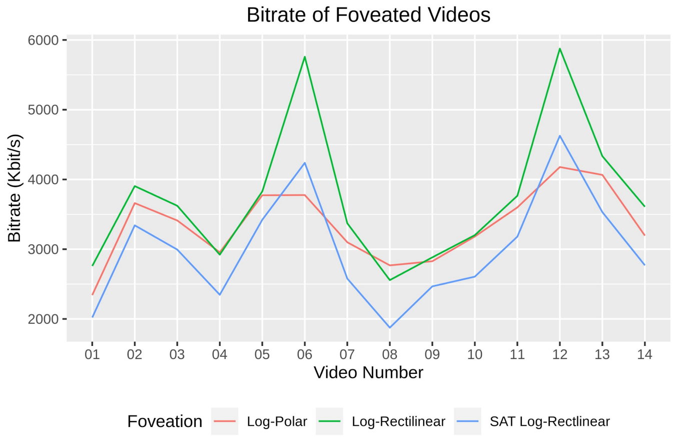
- Pairing the log-rectilinear transformation with summed area table filtering yields lower flickering while also reducing bandwidth usage and returning high weighted-to-spherical signal to noise ratio (WS-PSNR) results.





Quantitative Results

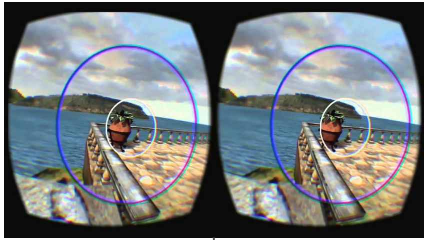
- Pairing the log-rectilinear transformation with summed area table filtering yields lower flickering while also reducing bandwidth usage and returning high weighted-to-spherical signal to noise ratio (WS-PSNR) results.



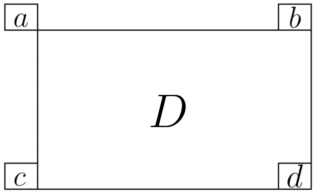
Conclusion

- We present a log-rectilinear transformation which utilizes foveation, summed-area tables, and standard video codecs for foveated 360° video streaming.

Foveation



Summed-Area Tables



Standard Video Codecs

H.264
MPEG-4/AVC

Foveated 360° Video Streaming

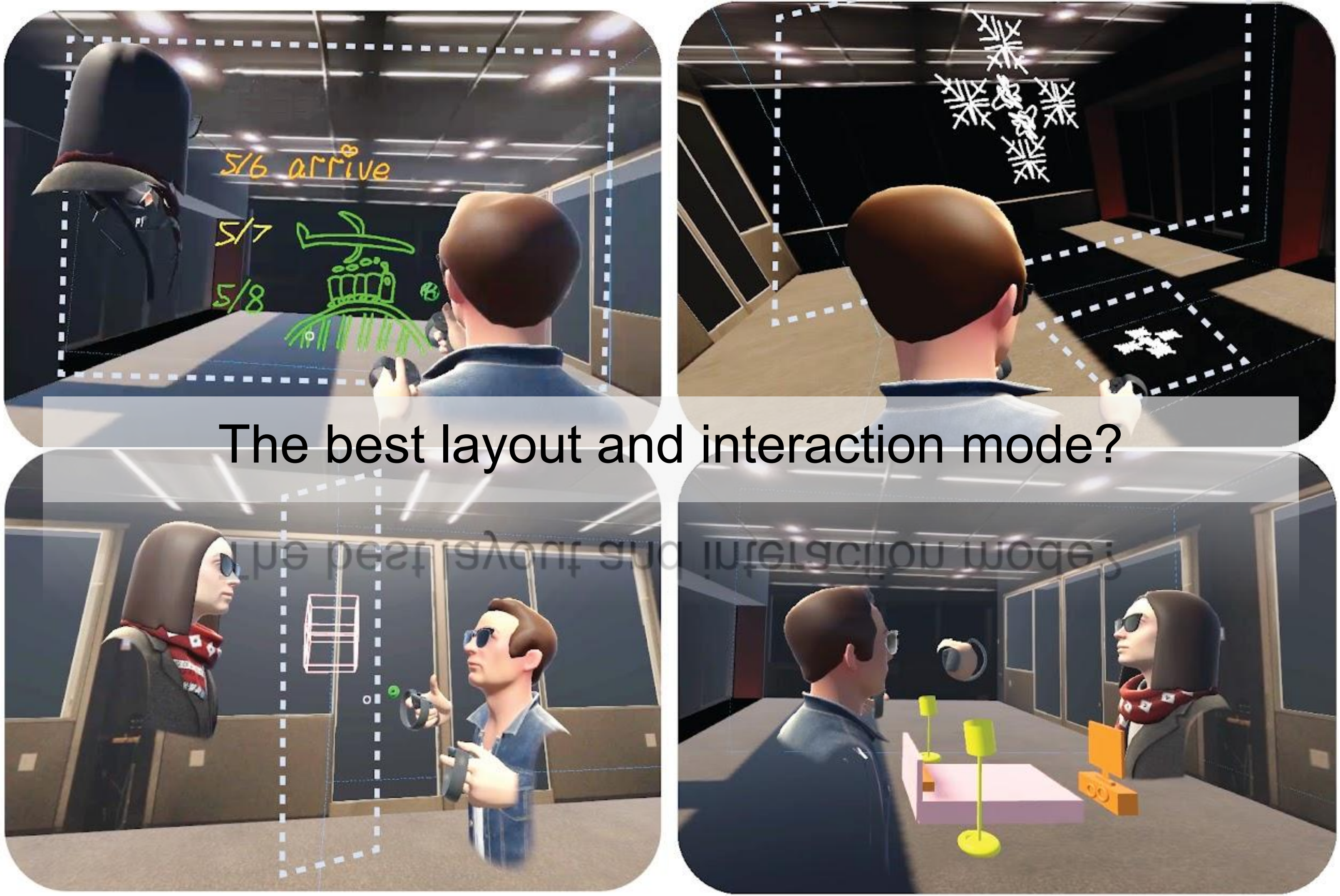
CollaboVR: A Reconfigurable Framework for Creative Collaboration in Virtual Reality

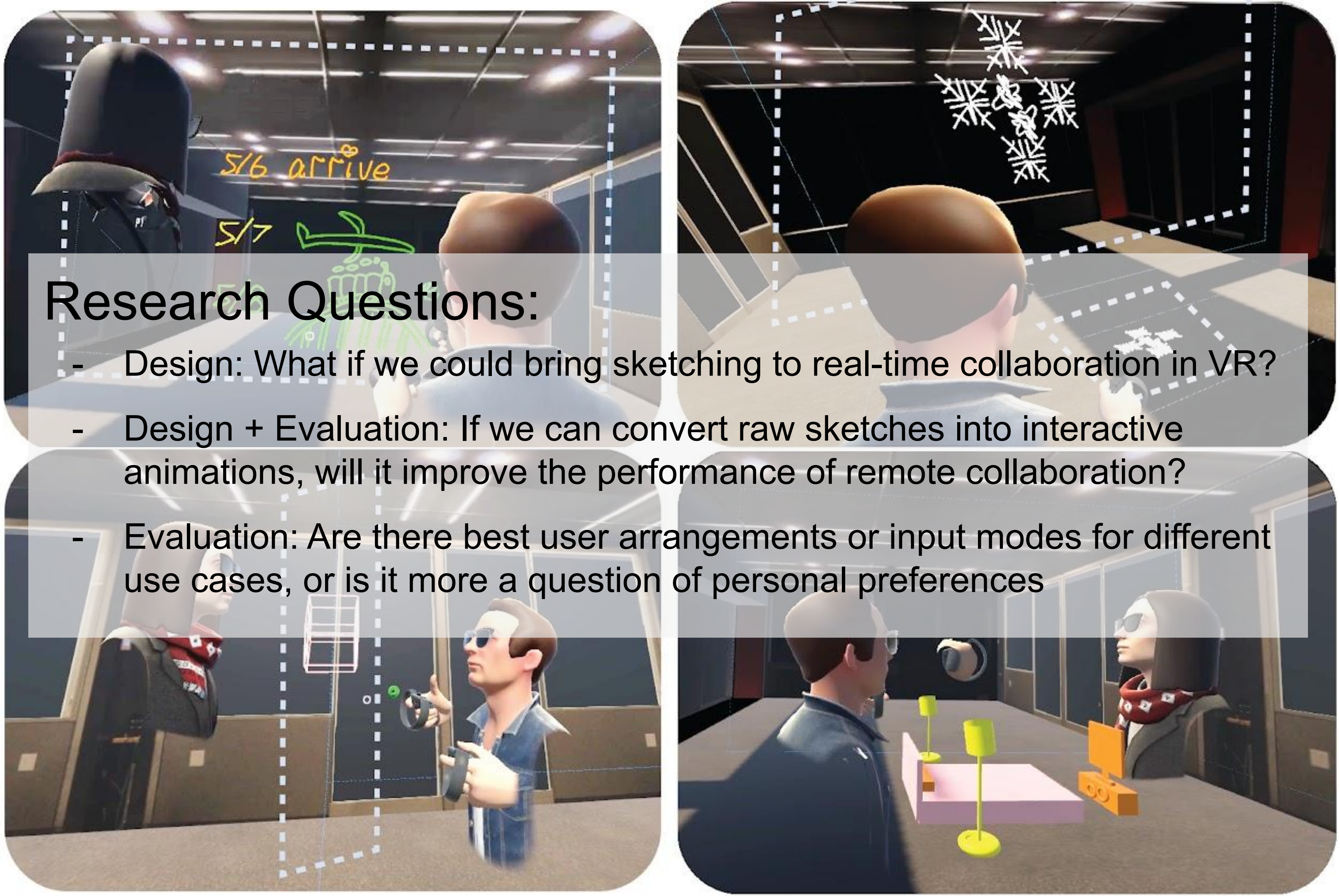
Zhenyi He[*] Ruofei Du[†] Ken Perlin[*]

[*]Future Reality Lab, New York University [†]Google LLC


ERASE

The best layout and interaction mode?





Research Questions:

- Design: What if we could bring sketching to real-time collaboration in VR?
- Design + Evaluation: If we can convert raw sketches into interactive animations, will it improve the performance of remote collaboration?
- Evaluation: Are there best user arrangements or input modes for different use cases, or is it more a question of personal preferences

CollaboVR: A Reconfigurable Framework for Creative Collaboration in Virtual Reality

(a) Discussing travel schedules in *integrated layout* with remote participants.

(b) Presenting the topic of four dimensional shapes in *mirrored layout*.

(c) Sketching a baroque pattern in *projective layout* to remote users.

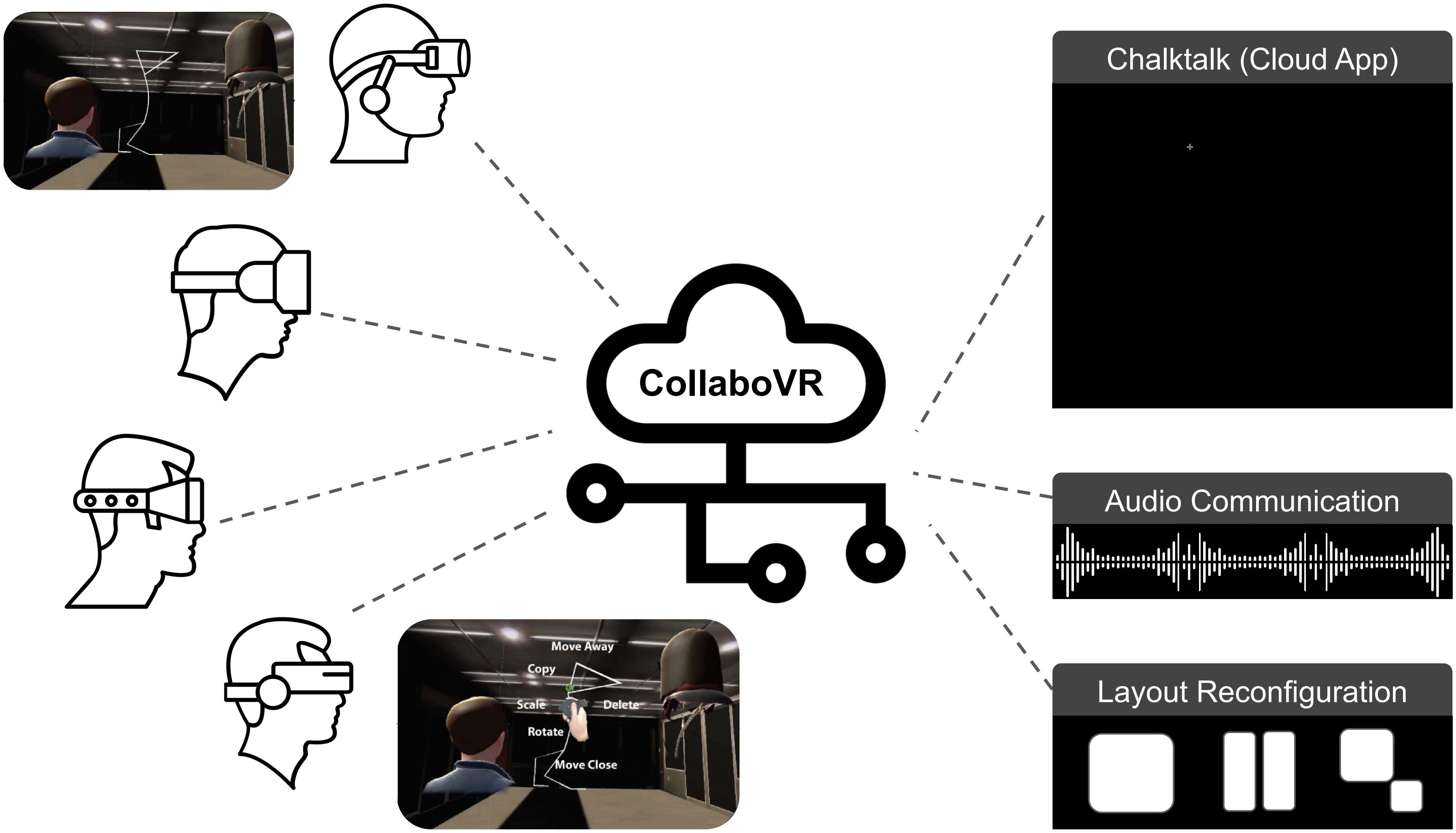
(d) Collaborative design session of furniture and apartment arrangements.

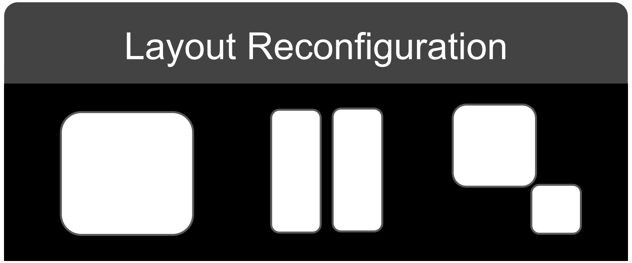
ISMAR2020
IEEE International Symposium on Mixed and Augmented Reality

NYU

FUTURE REALITY LAB

Google


CollaboVR
Chalktalk (Cloud App)
Audio Communication
Layout Reconfiguration
Move Away
Copy
Scale
Delete
Rotate
Move Close

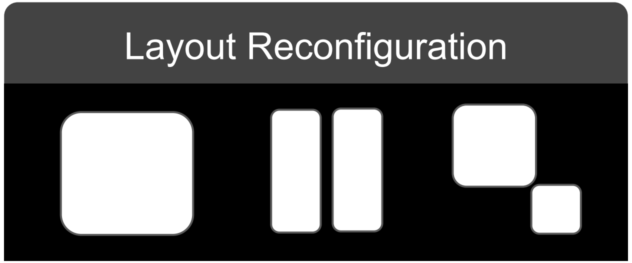


User Arrangements

(1) side-by-side
(2) face-to-face
(3) hybrid

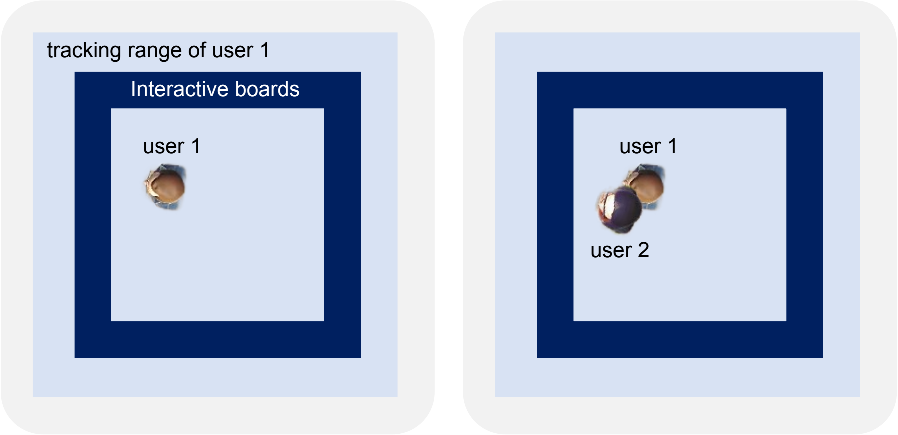
Input Modes

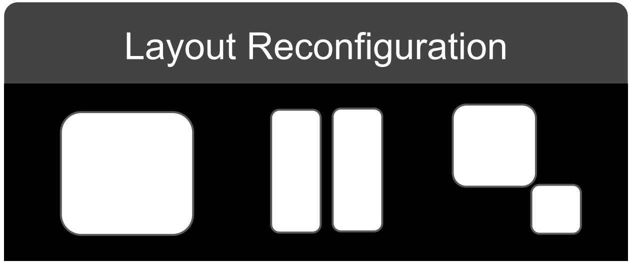
(1) direct
(2) projection



User Arrangements

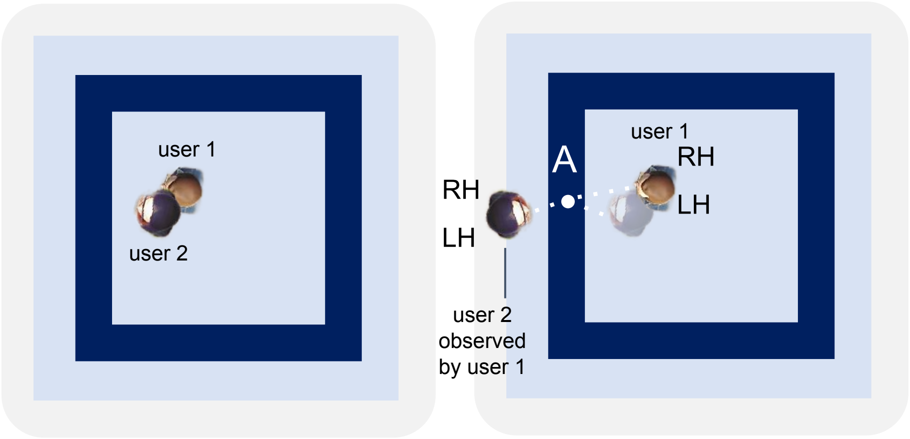
(1) side-by-side

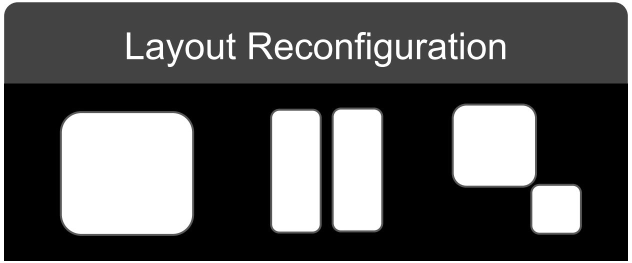




User Arrangements

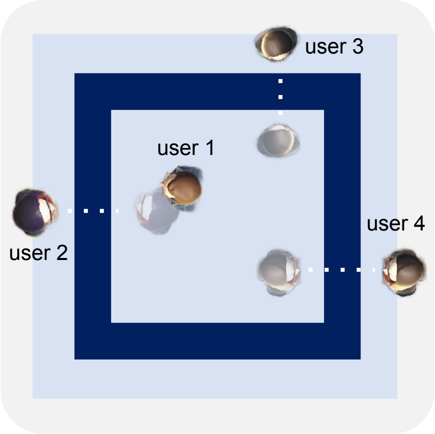
(1) side-by-side
(2) face-to-face

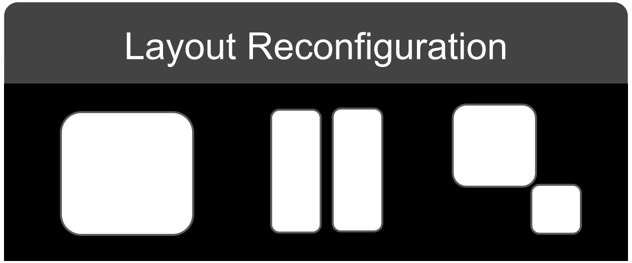




User Arrangements

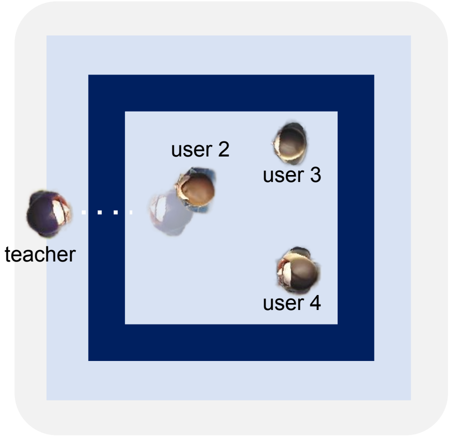
(1) side-by-side
(2) face-to-face

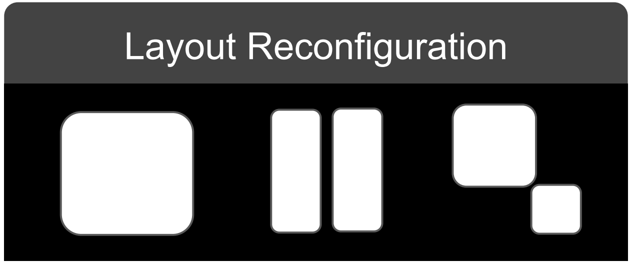




User Arrangements

(1) side-by-side
(2) face-to-face
(3) hybrid

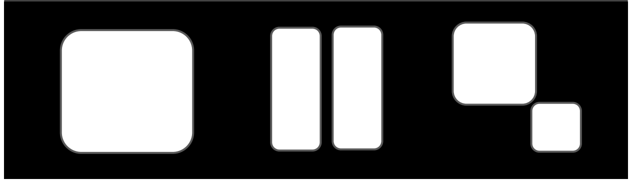




Input Modes

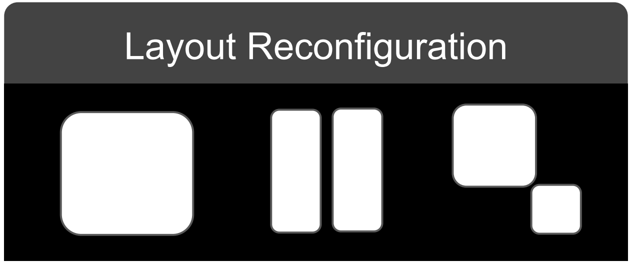
(1) direct
(2) projection

Layout Reconfiguration



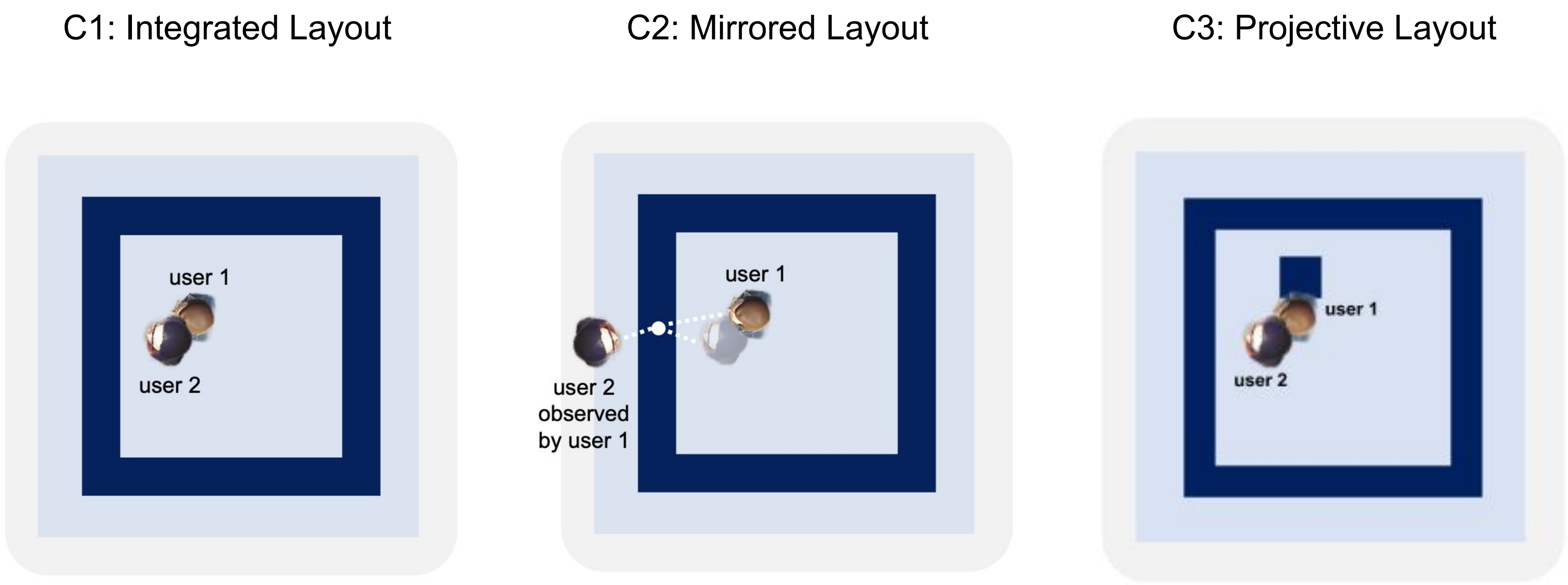
Input Modes

(1) direct
(2) projection



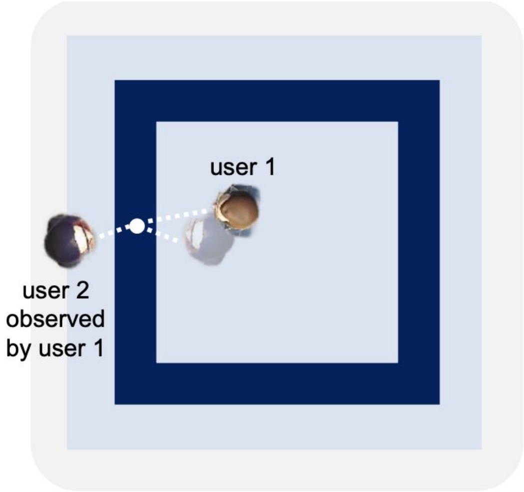
Input Modes

(1) direct
(2) projection

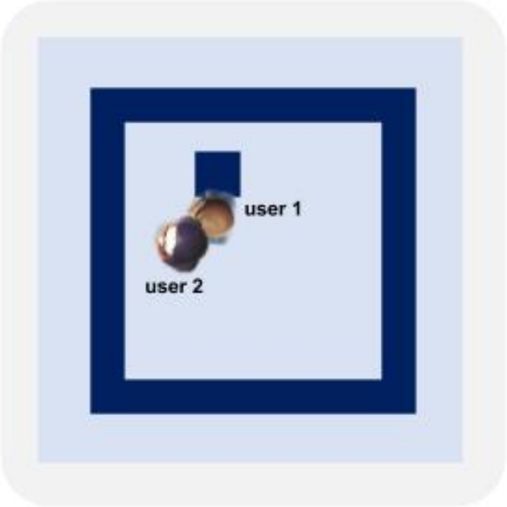

C1: Integrated Layout
C2: Mirrored Layout
C3: Projective Layout
user 1
user 2
user 1
user 2
observed
by user 1
user 1
user 2

C1: Integrated Layout

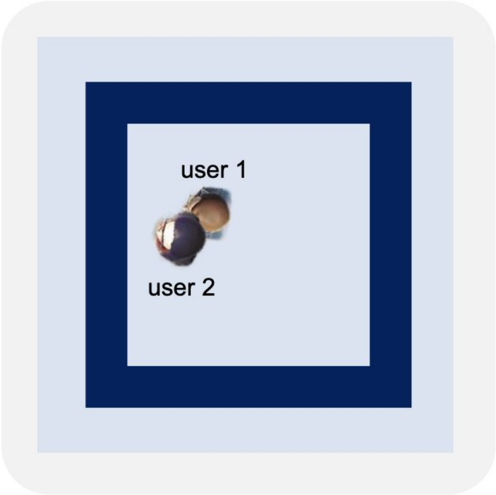
C2: Mirrored Layout



C3: Projective Layout

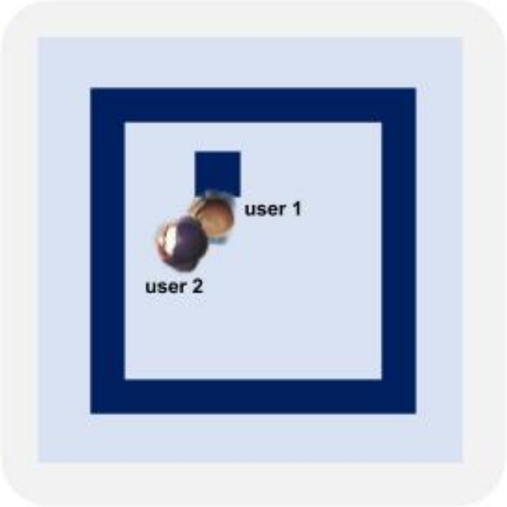


C1: Integrated Layout

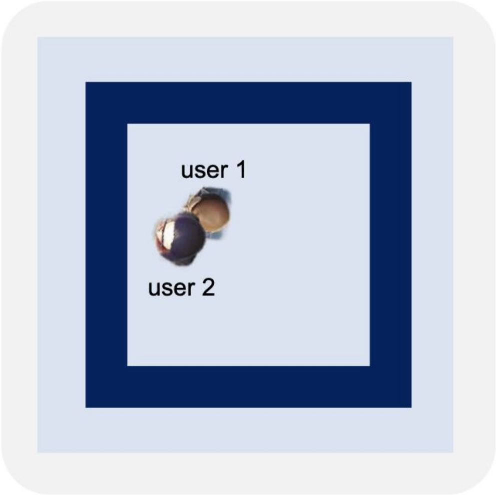


C2: Mirrored Layout

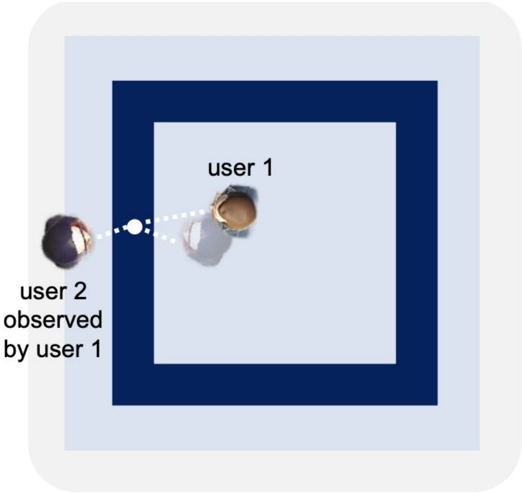
C3: Projective Layout



C1: Integrated Layout

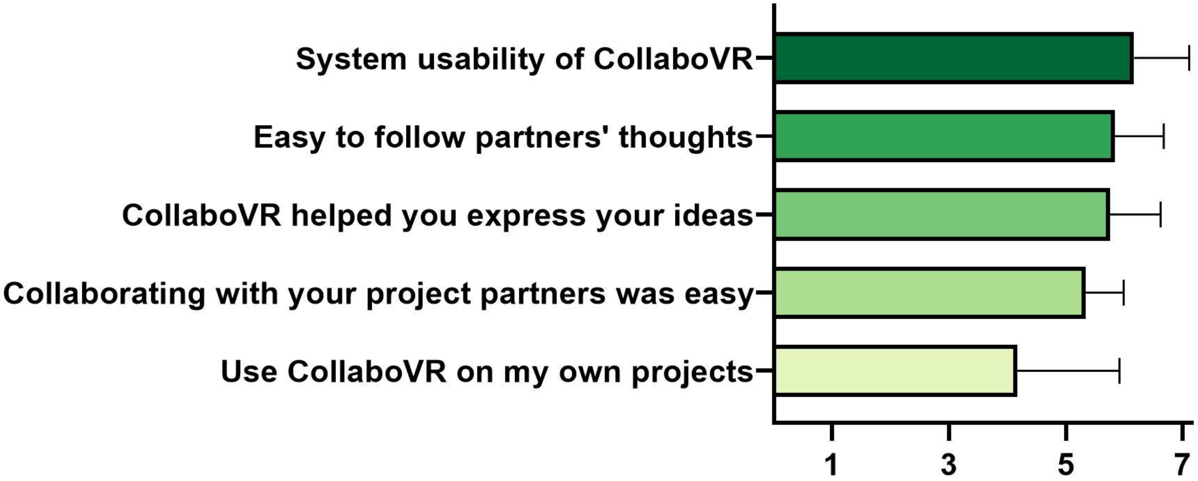


C2: Mirrored Layout



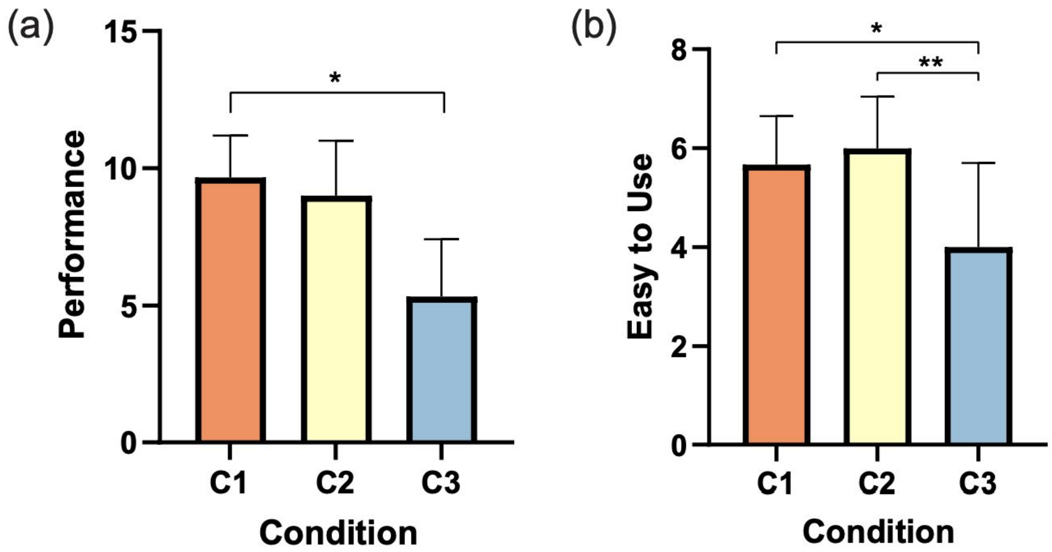
C3: Projective Layout

Evaluation

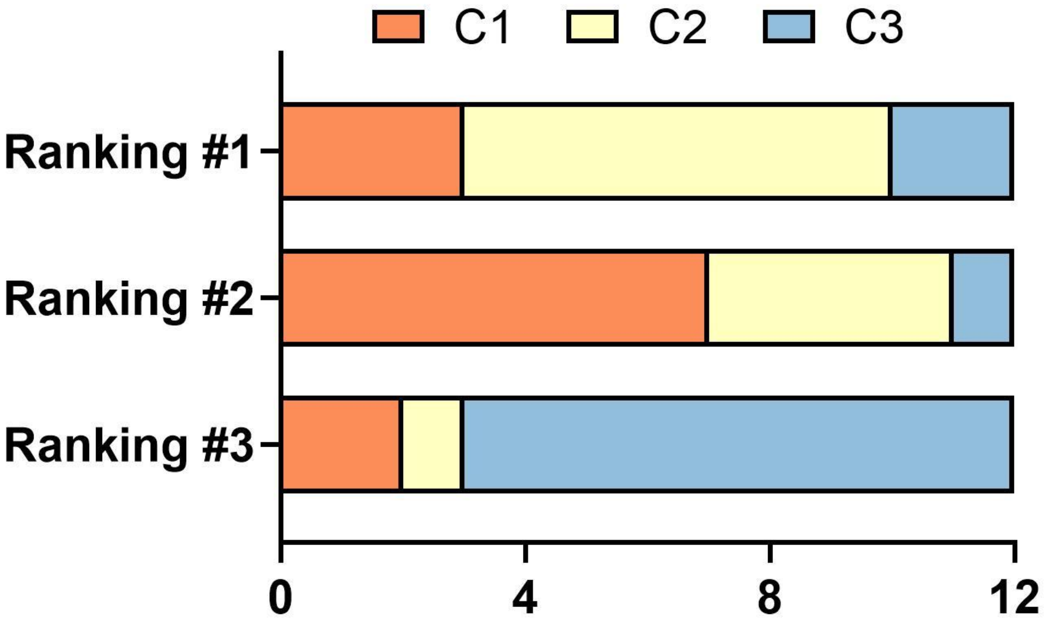


Overview of subjective feedback on CollaboVR

Evaluation


(a)
15
10
5
0
Performance
*
C1
C2
C3
Condition
(b)
8
6
4
2
0
Easy to Use
*
**
C1
C2
C3
Condition

Evaluation



Takeaways

1. Developing CollaboVR, a reconfigurable end-to-end collaboration system.
2. Designing custom configurations for real-time user arrangements and input modes.
3. Quantitative and qualitative evaluation of CollaboVR.
4. Open-sourcing our software at https://github.com/snowymo/CollaboVR.

more live demos...


Create Board
Draw
Manipulate Sketch
Color Palette
DrawToggle

CollaboVR: A Reconfigurable Framework for Creative Collaboration in Virtual Reality



Zhenyi He[*]　　Ruofei Du[†]　　Ken Perlin[*]

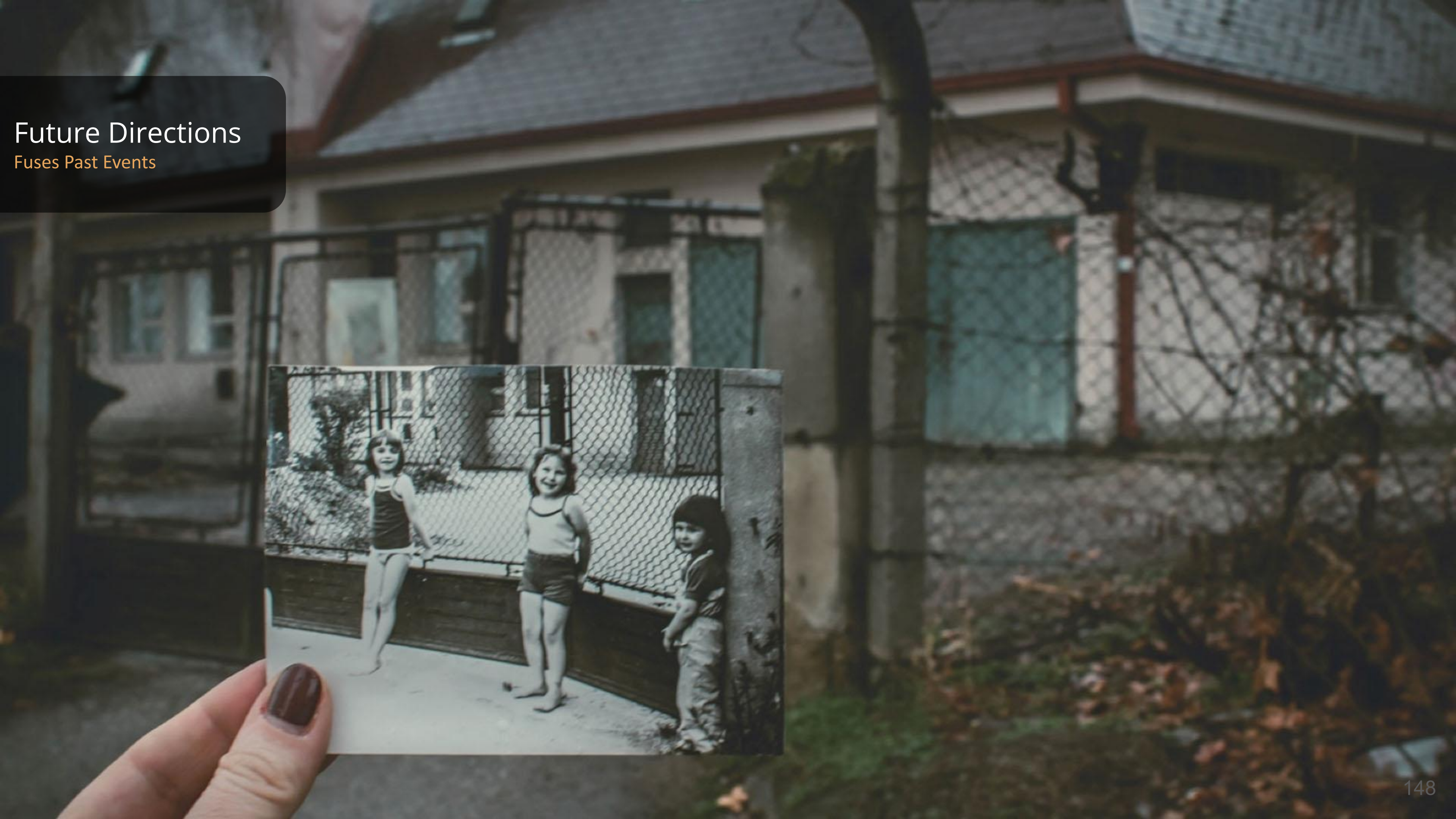
[*]Future Reality Lab, New York University　[†]Google LLC

Future Directions

The Ultimate XR Platform

Future Directions

Fuses Past Events



Future Directions
With the present

Future Directions

And look into the future

Future Directions

Change the way we communicate in 3D and consume the information

Future Directions

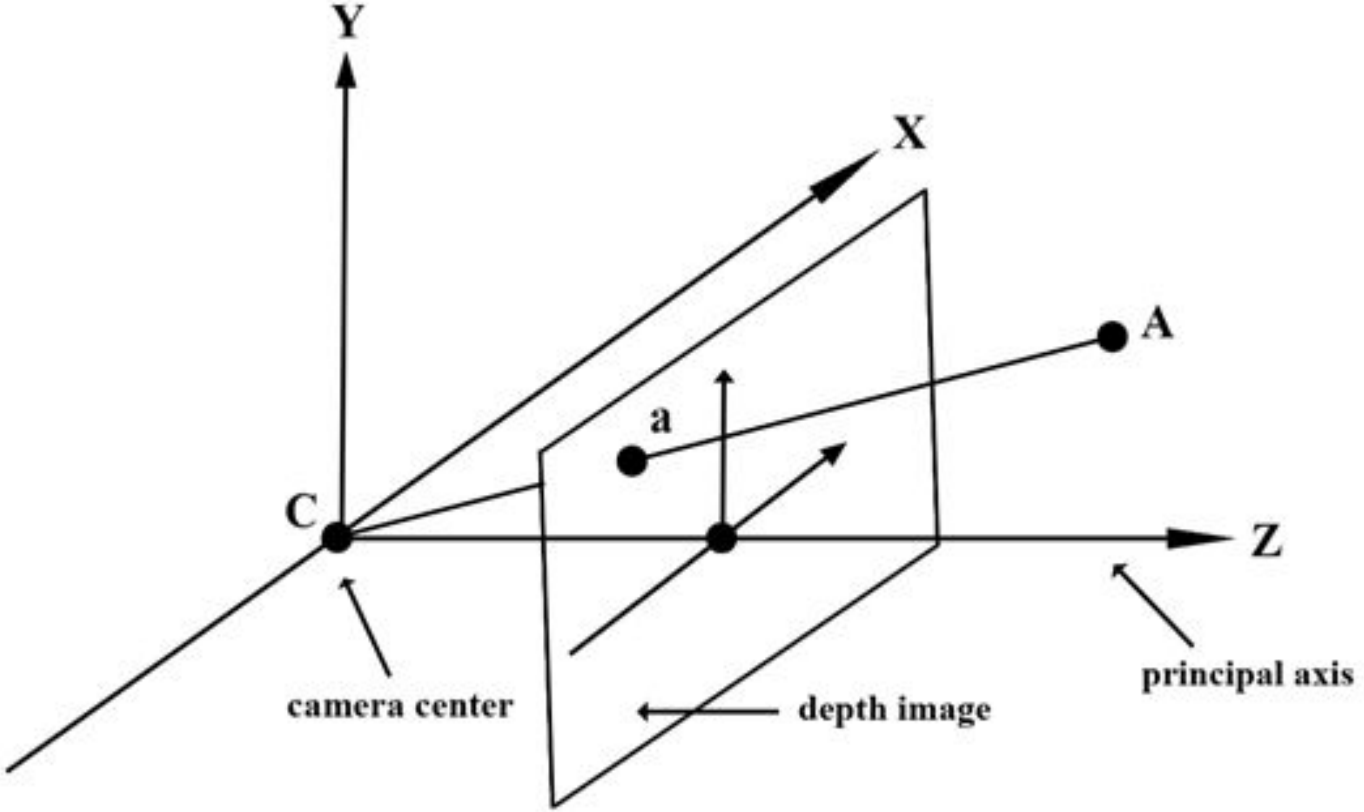
Consume the information throughout the world

Fusing Physical and Virtual Worlds into Interactive Mixed Reality

Ruofei Du | Google, San Francisco | me@duruofei.com
Virtual | Mobile Immersive Computing by Prof. Bo Han | March 19, 2021

Introduction

Depth Map



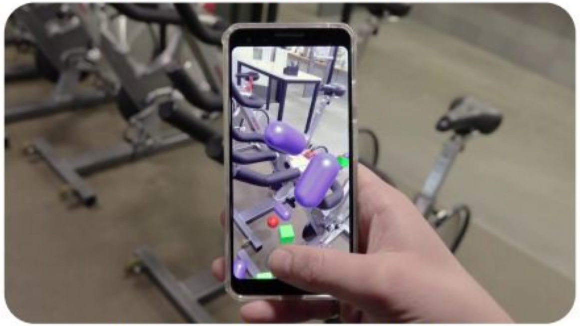
Introduction

Depth Lab

(a) oriented reticles and splats

(b) ray-marching-based scene relighting

(c) depth visualization and particles



(d) geometry-aware collisions

(e) 3D-anchored focus and aperture effect

(f) occlusion and path planning

Thank you!

www.duruofei.com

DepthLab: Real-Time 3D Interaction With Depth Maps for Mobile Augmented Reality

Ruofei Du, Eric Turner, Maksym Dzitsiuk, Luca Prasso, Ivo Duarte, Jason Dourgarian, Joao Afonso, Jose Pascoal, Josh Gladstone, Nuno Cruces, Shahram Izadi, Adarsh Kowdle, Konstantine Tsotsos, and David Kim

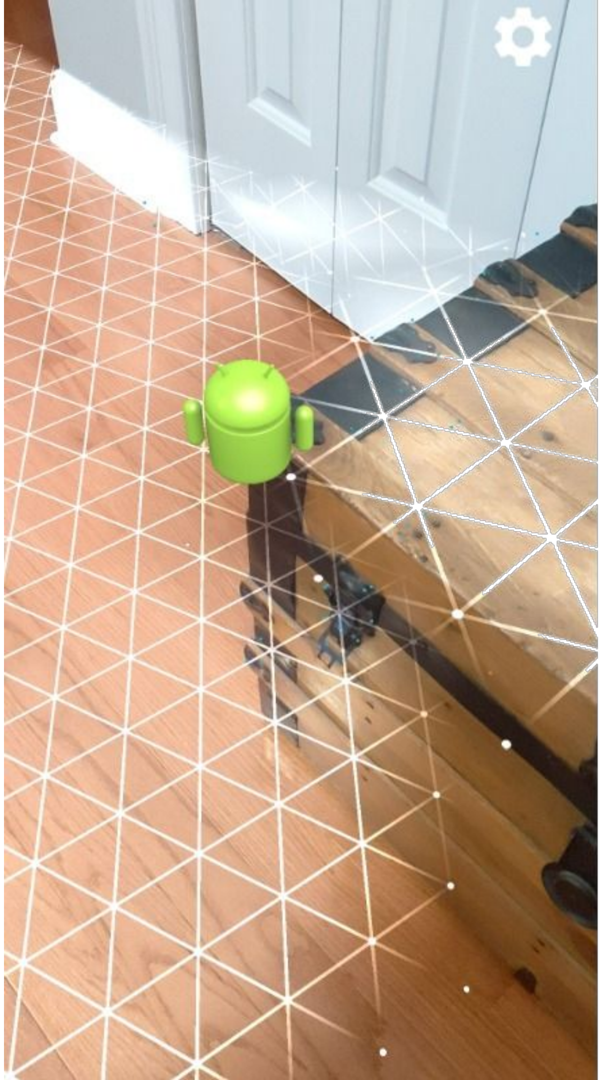
Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology (UIST), 2020.

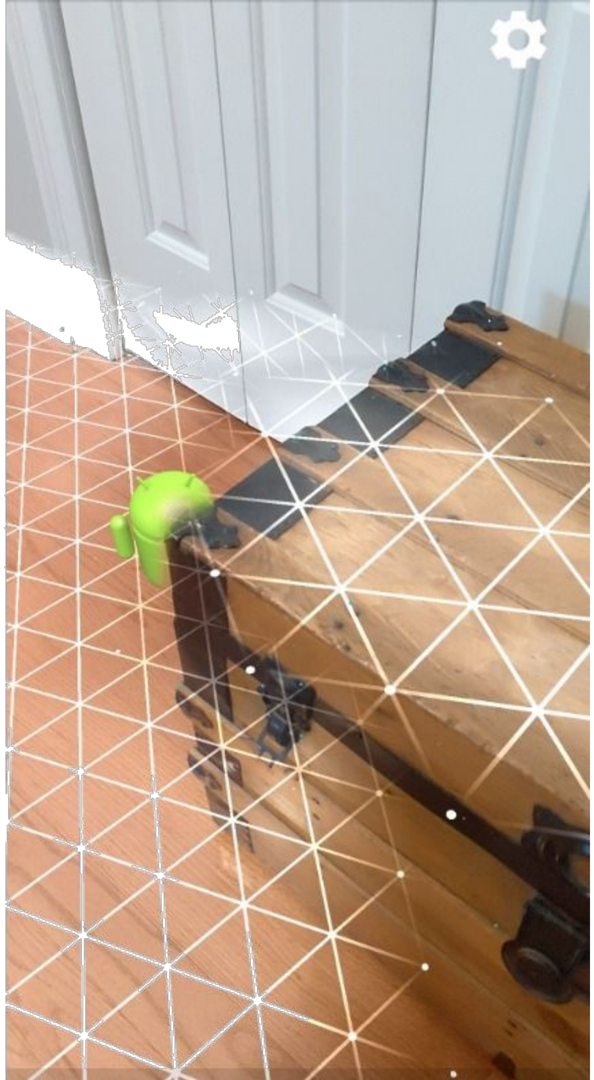
pdf, lowres | website, code, demo, supp | cite

Occlusion is a critical component for AR realism!

Correct occlusion helps ground content in reality, and makes virtual objects feel as if they are actually in your space.

Introduction
Motivation





Depth Mesh
Generation

Algorithm 2: Real-time Depth Mesh Generation.

Input : Depth map $\mathbf{D}$, its dimension $w \times h$, and depth discontinuity threshold $\Delta d_{\max} = 0.5$.

Output : Lists of mesh vertices $\mathbb{V}$ and indices $\mathbb{I}$.

In the initialization stage on the CPU:

for $x \in [0, w-1]$ **do**
    for $y \in [0, h-1]$ **do**
        Set the pivot index: $I_0 \leftarrow y \cdot w + x$.
        Set the neighboring indices:
         $I_1 \leftarrow I_0 + 1,\ I_2 \leftarrow I_0 + w,\ I_3 \leftarrow I_2 + 1$.
        Add the vertex $(x/w, y/h, 0)$ to $\mathbb{V}$.
    end
end

In the rendering stage on the CPU or GPU:

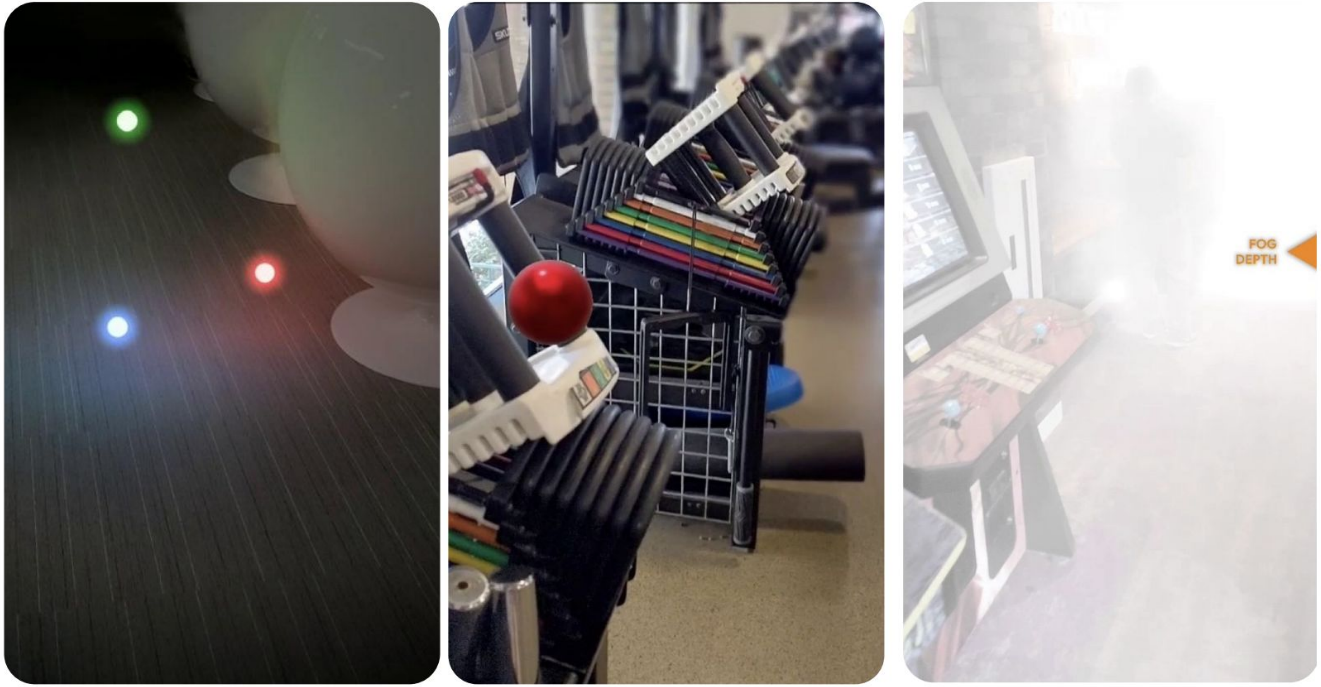
for *each vertex* $v \in \mathbb{V}$ **do**
    Look up v's corresponding screen point $\mathbf{p}$.
    Fetch v's depth value $d_0 \leftarrow \mathbf{D}(\mathbf{p})$.
    Fetch v's neighborhoods' depth values:
     $d_1 \leftarrow \mathbf{D}(\mathbf{p} + (1,0)),\ d_2 \leftarrow \mathbf{D}(\mathbf{p} + (0,1))$,
     $d_3 \leftarrow \mathbf{D}(\mathbf{p} + (1,1))$.
    Compute average depth $\bar{d} \leftarrow \frac{d_0 + d_1 + d_2 + d_3}{4}$.
    Let $\mathbf{d} \leftarrow [d_0, d_1, d_2, d_3]$.
    if $\textbf{\textit{any}}\left(\textbf{\textit{step}}\left(\Delta d_{\max}, |\mathbf{d} - \bar{d}|\right)\right) = 1$ **then**
        Discard v due to large depth discontinuity.
    end
end

Localized Depth

Avatar Path Planning

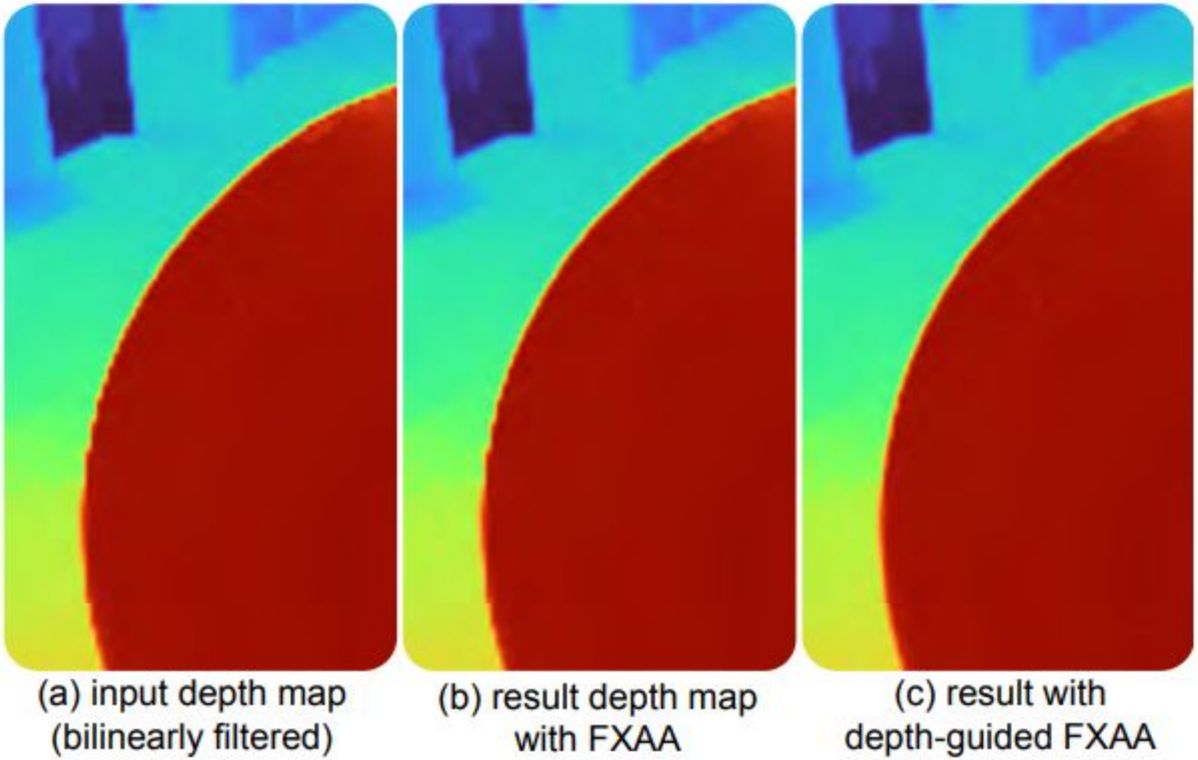
Dense Depth

Depth Texture



(a) relighting effect (b) aperture effect (c) fog effect

Introduction
Depth Map



Taxonomy

Depth Usage

Depth-based Interaction Design Space		
Geometry-aware ...	Depth Interaction ...	Visual Effects of ...
Rendering	**Interface**	**Static**
occlusion shadows relighting	3D cursor bounding-box region selection ...	Bokeh effect triplanar mapping aligned AR text ...
Actions	**Gestures**	**Dynamic**
physics path planning free-space check ...	static hand dynamic motion 3D touch ...	depth transition light painting surface ripples ...

Introduction

Depth Map

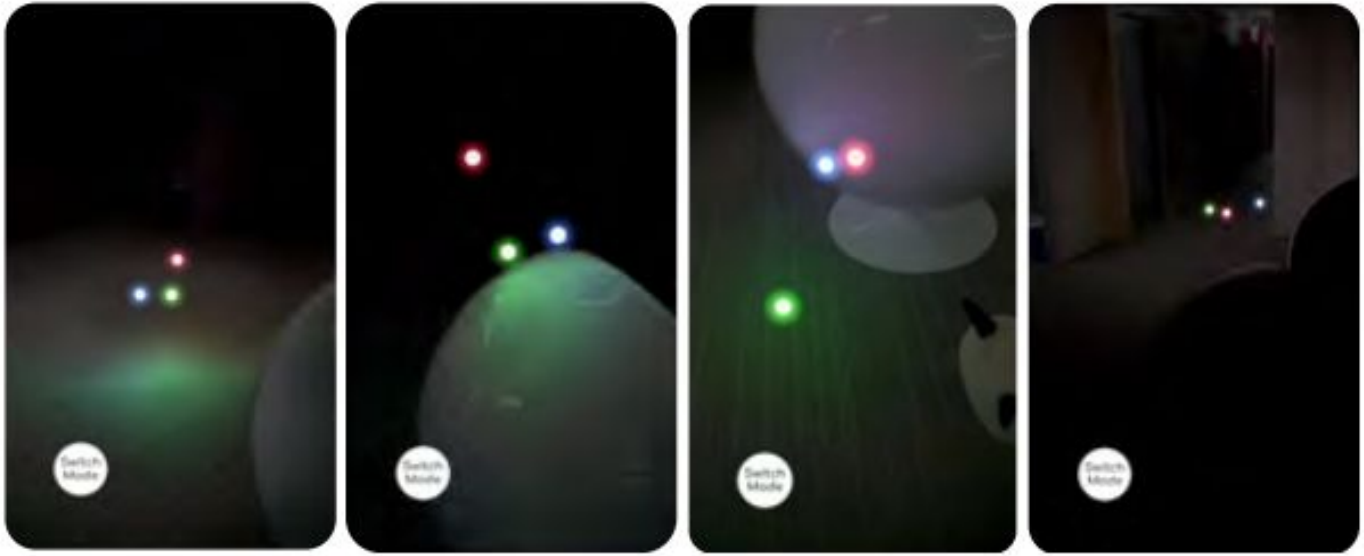


Figure 12. Given a dense depth texture, a camera image, and virtual light sources, we altered the lighting of the physical environment by tracing occlusions along the light rays in real time.

Introduction

Depth Map

Figure 13. Wide-aperture effect focused on a world-anchored point on a flower from different perspectives. Unlike traditional photography software, which only anchors the focal plane to a screen point, DepthLab allows users to anchor the focal point to a physical object and keep the object in focus from even when the viewpoint changes. Please zoom in to compare the focus and out-of-focus regions.