OpenARK – Open Source Augmented Reality

Presenter: Joseph Menke

OpenARK

augcog.berkeley.edu/OpenARK/html

AR/VR – The Next Computing Platform

In 2016, Goldman Sachs believed that Augmented Reality (AR) and Virtual Reality (VR) would become the next big computing platform. Their “accelerated uptake” scenario predicted a **182 billion** dollar market by 2025, where AR/VR technology evolved from a niche device to a broader computing platform.

“What is the biggest obstacle to mass adoption of AR/VR technology?”

-2018 VR AR Survey Digital

- User Experience!

OpenARK – Why We Do This?

- Say goodbye to external controllers and interact with virtual objects with your bare hands!
  - Hand tracking and SLAM solutions are changing how we interact with technology and bring AR/VR experiences to life.
- OpenARK aims to build an **open ecosystem** that pushes the AR/VR industry forward and foster numerous startups.
- OpenARK is **the first** open-source project in AR/VR computing platform since 2016 that we believe will play a crucial role in the era of AR/VR, just like Linux to PC/Desktop and Android to Mobile.

OpenARK is an open-source platform includes fundamental tools such as AR-based camera calibration and SLAM, and it also includes higher-level functions to aid human-computer interaction, such as 3D gesture recognition and multi-user collaboration.

Features:

- 3D Hand Tracking
- Building Scale SLAM
- Building Interior 3D Reconstruction
- Articulated Avatar Tracking

Hardware Ecosystem

Support a wide variety of Devices:

- Intel RealSense Serials
- Kinect
- HTC VIVE Focus
- More off-the-shelf depth cameras

> “More than 60 to 70 percent of everything built for machines for virtual reality or augmented reality or any of the XR platforms are built in Unity.”

\- John Riccitiello, Unity CEO

OpenARK can be seamlessly integrated into popular game engines like Unity3D, making it accessible to more developers and end users.

This Unity plugin fully abstracts the communications between OpenARK's native library and the Unity game engine, providing a simple object-oriented interface that allows developers to easily access OpenARK functionality from any Unity project.

“建行杯”
第四届中国互联网+
大学生创新创业大赛
The 4th China College Students' 'Internet Plus'
Innovation and Entrepreneurship Competition
勇立时代潮头敢闯会创 扎根中国大地书写人生华章
冠军争夺赛

OpenARK Technical Details

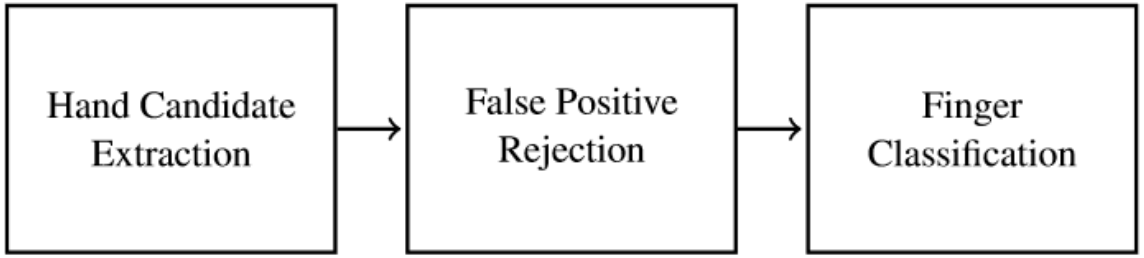
Hand Tracking

Simultaneous Localization And Mapping

OpenARK

augcog.berkeley.edu/OpenARK/html

Overview



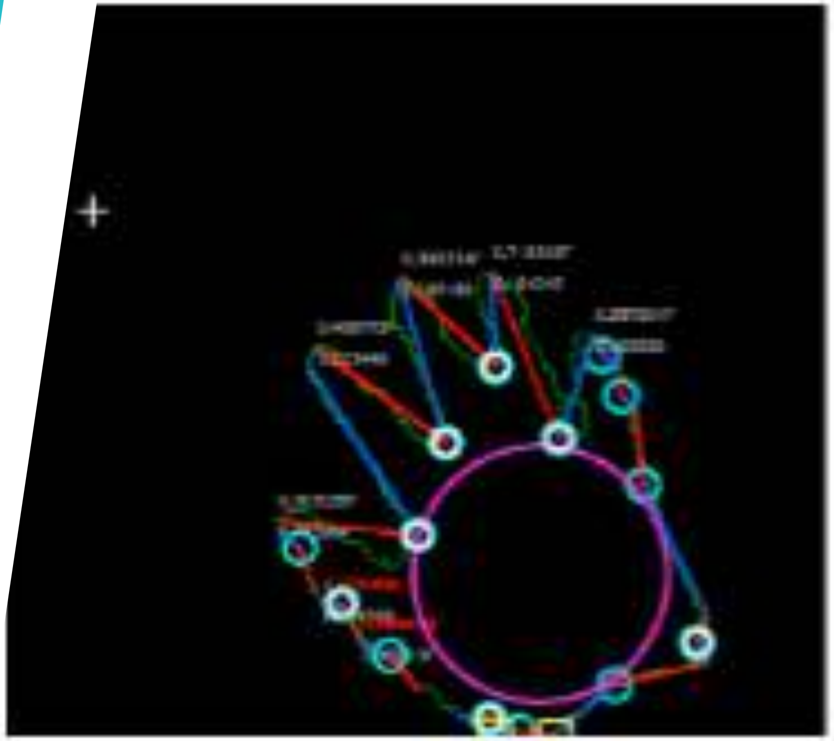
- Low-Latency: Required for using hand models as an input modality in human-computer interface
- Light Weight: Both compute and physical
- Robust to Cluttered Background: the solutions must work well under complex lighting conditions and a variety of backgrounds

Plane Detection

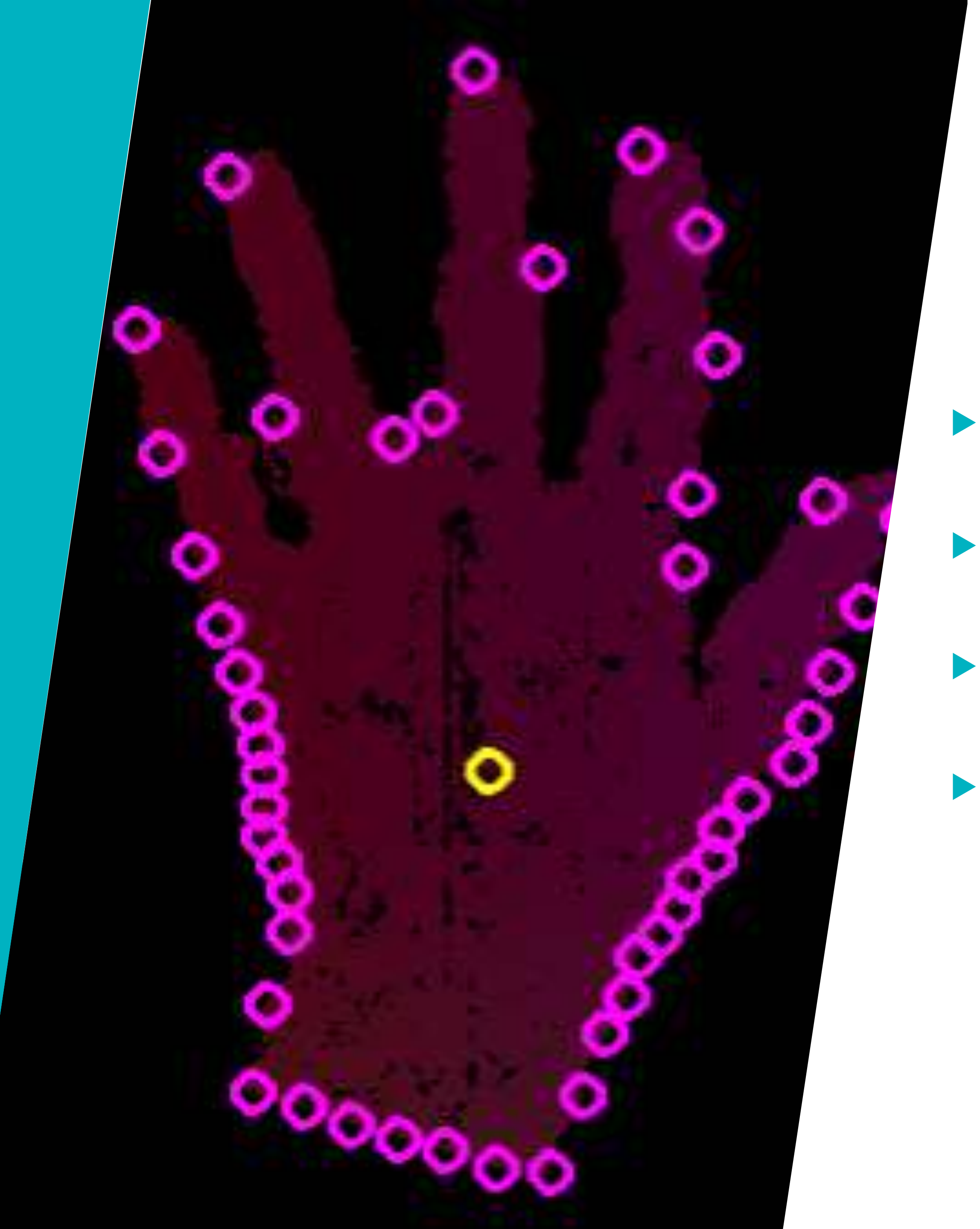
- Compute the normal vector for each pixel
- Choose a random seed
- Grow region along similar normal vectors and depth

Egocentric Viewpoint and Area Filtering

- Run flood-fill clustering on depth image
- Remove clusters that don't intersect with bottom of the image.
- Remove clusters whose area is not similar to that of a hand.



Contour Detection



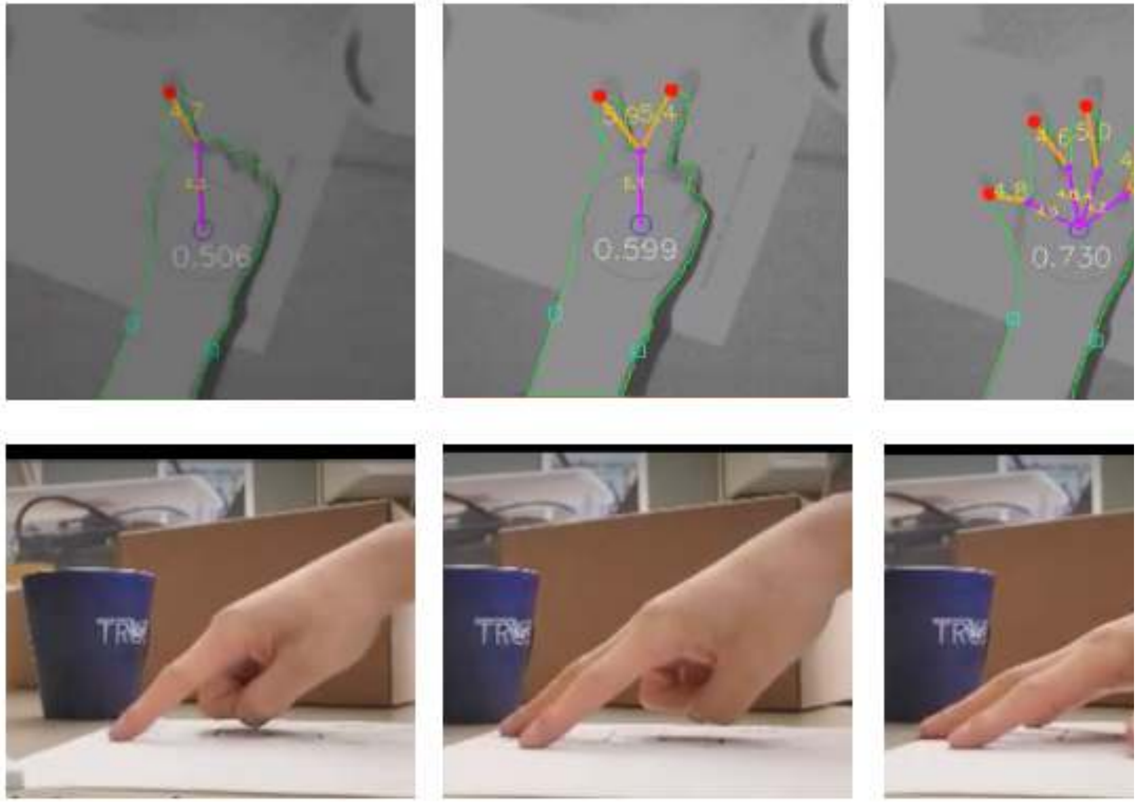
Support Vector Machine Classification

- Enforce Rotation and Translation Invariance
- Find 48 Key Points by casting rays from palm center
- Create Features using Euclidean distance to palm center
- Trained on 18k examples

Finger Classification

- Finger Length: Distance between D1 and D2.
- Distance between defect to palm: Distance between D2 and P.
- Slope of Finger and nearest defect: Slope of the line between D1 and D2.
- Slope of finger and palm center: Slope of the line between D1 and P.
- Local curvature: the curvature of the contour between C1, D3, and C2.
- Non-local curvature: the curvature of the contour between C3, D3, and C4, to contrast with *local curvature*.

Plane Interactions



- By removing the plane first, we can detect finger positions even when in contact with planes
- By also estimating plane parameters we can determine when fingers contact planes
- Turn any surface into a touch screen!

Results

- Tested on CVAR Hand Dataset
- 2800 unique hand poses
- P6 & P7 contain hand interacting with non-planar objects
- Speed performance evaluated our algorithm Intel i5-9600k and Intel i3- 8350k CPU with an Intel Realsense SR300 depth camera.

Table 1: Finger Classification Accuracy

Dataset	P1	P2	P3	P4	P5	P6	P7
Accuracy (%)	98.7	95.7	94.1	93.8	94.2	86.2	84.1

Table 2: Speed Performance

Configuration	Hand Only	Hand + Clutter	Hand + Clutter + Plane
i5-9600k (FPS)	45	42	36
i3-8350k (FPS)	35	32	31

Limitations

- Detection will only be as good as the depth image you put in.
- Cannot detect occluded fingers
- Does not detect exact joint positions
- Will not work with IR absorbent materials. (Black cloth)

OpenARK

vivecenter.berkeley.edu/OpenARK

Simultaneous Localization And Mapping

Depth Camera Properties

- Global Shutter
- Stereo
- Time Synchronized IMU
- Structured Light IR Projector

Visual Inertial Odometry

$$J(\mathbf{x}) := \underbrace{\sum_{i=1}^{I} \sum_{k=1}^{K} \sum_{j \in \mathcal{J}(i,k)} \mathbf{e}_{\mathrm{r}}^{i,j,k\,T} \mathbf{W}_{\mathrm{r}}^{i,j,k} \mathbf{e}_{\mathrm{r}}^{i,j,k}}_{\text{visual}} + \underbrace{\sum_{k=1}^{K-1} \mathbf{e}_{\mathrm{s}}^{k\,T} \mathbf{W}_{\mathrm{s}}^{k} \mathbf{e}_{\mathrm{s}}^{k}}_{\text{inertial}}$$

Visual: $$\mathbf{e}_{\mathrm{r}}^{i,j,k} = \mathbf{z}^{i,j,k} - \mathbf{h}_i \left(\boldsymbol{T}_{C_i S} \boldsymbol{T}_{SW}^{k} \boldsymbol{l}_{W}^{WL,j} \right)$$

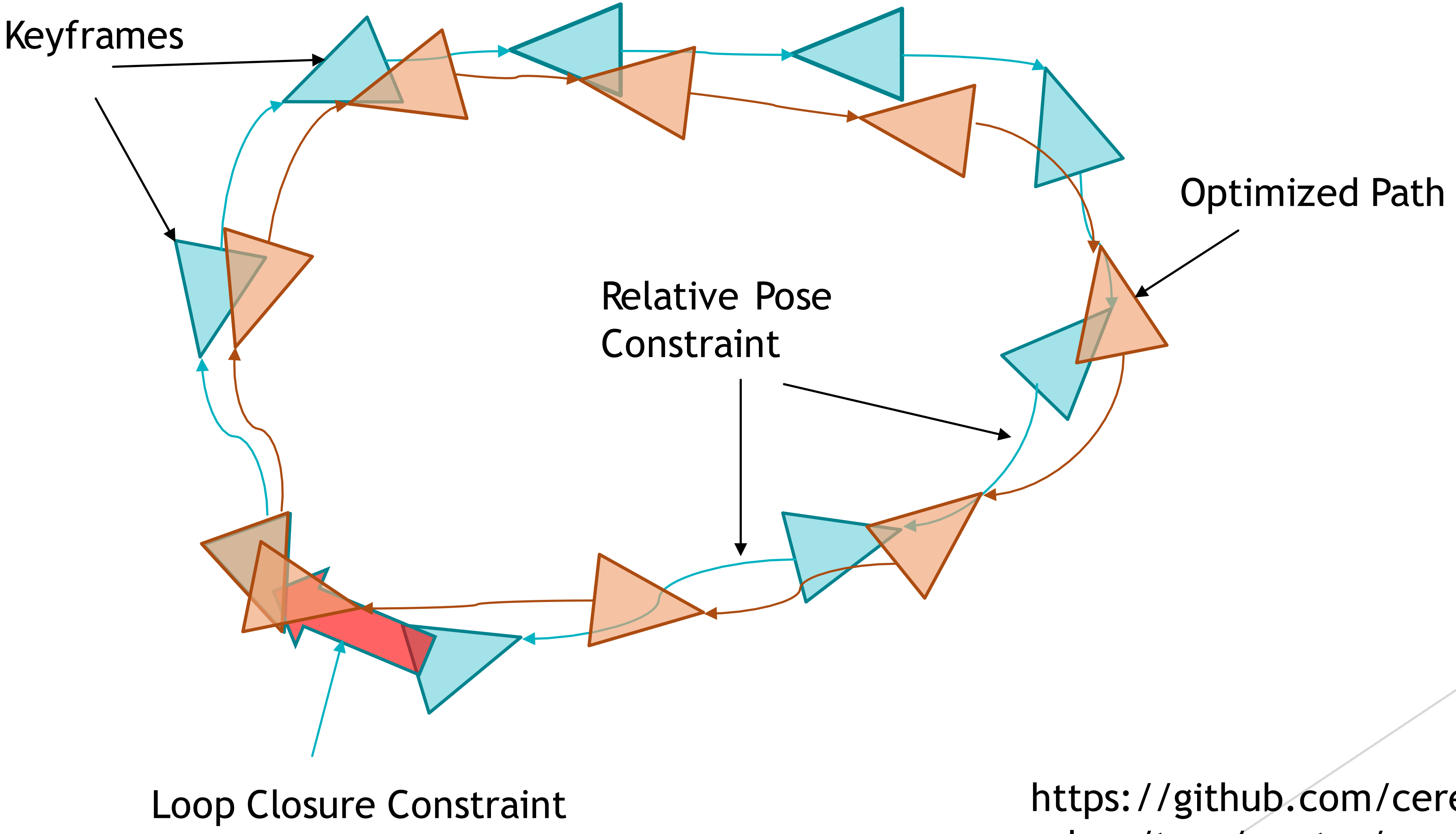
Inertial: $$\mathbf{e}_{\mathrm{s}}^{k}\left(\mathbf{x}_{\mathrm{R}}^{k}, \mathbf{x}_{\mathrm{R}}^{k+1}, \mathbf{z}_{\mathrm{s}}^{k}\right) = \begin{bmatrix} \hat{\mathbf{p}}_{W}^{WS^{k+1}} - \mathbf{p}_{W}^{WS^{k+1}} \\ 2\left[\hat{\mathbf{q}}_{WS}^{k+1} \otimes \mathbf{q}_{WS}^{k+1^{-1}}\right]_{1:3} \\ \hat{\mathbf{x}}_{\mathrm{sb}}^{k+1} - \mathbf{x}_{\mathrm{sb}}^{k+1} \end{bmatrix}$$

S. Leutenegger et al "Keyframe-based visual-inertial odometry using nonlinear optimization" 2015

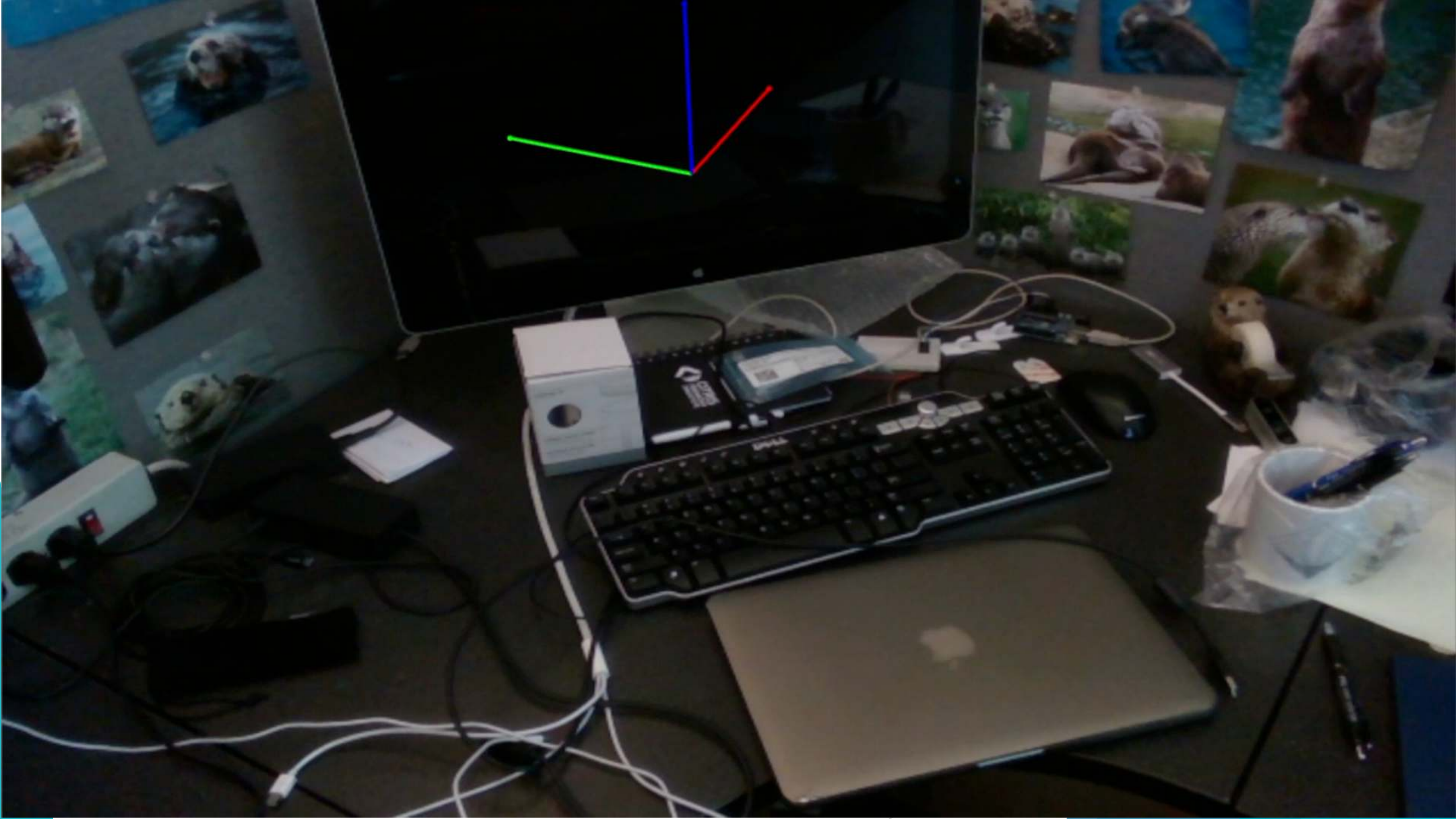
Loop Closure Detection

- **Bag of Binary Words for Fast Place Recognition in Image Sequences**
  D. Galvez-Lopez and J. Tardos
  2012

Pose Graph Optimization



https://github.com/ceres-solver/ceres-solver/tree/master/examples/slam/pose_graph_3d



Current Work

- SLAM reset when no features are detected
- Relocalize to a previously collected map
- Unity integration of SLAM

Questions?

Questions?