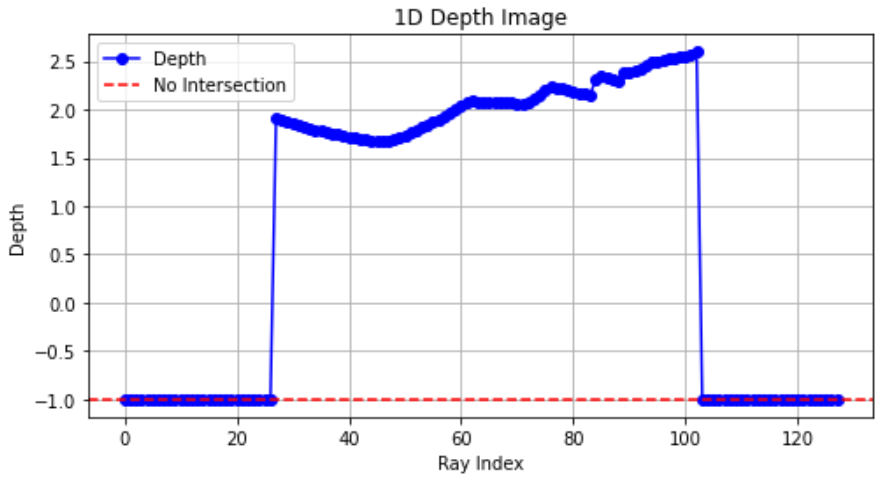
Hacker: 2D KinectFusion

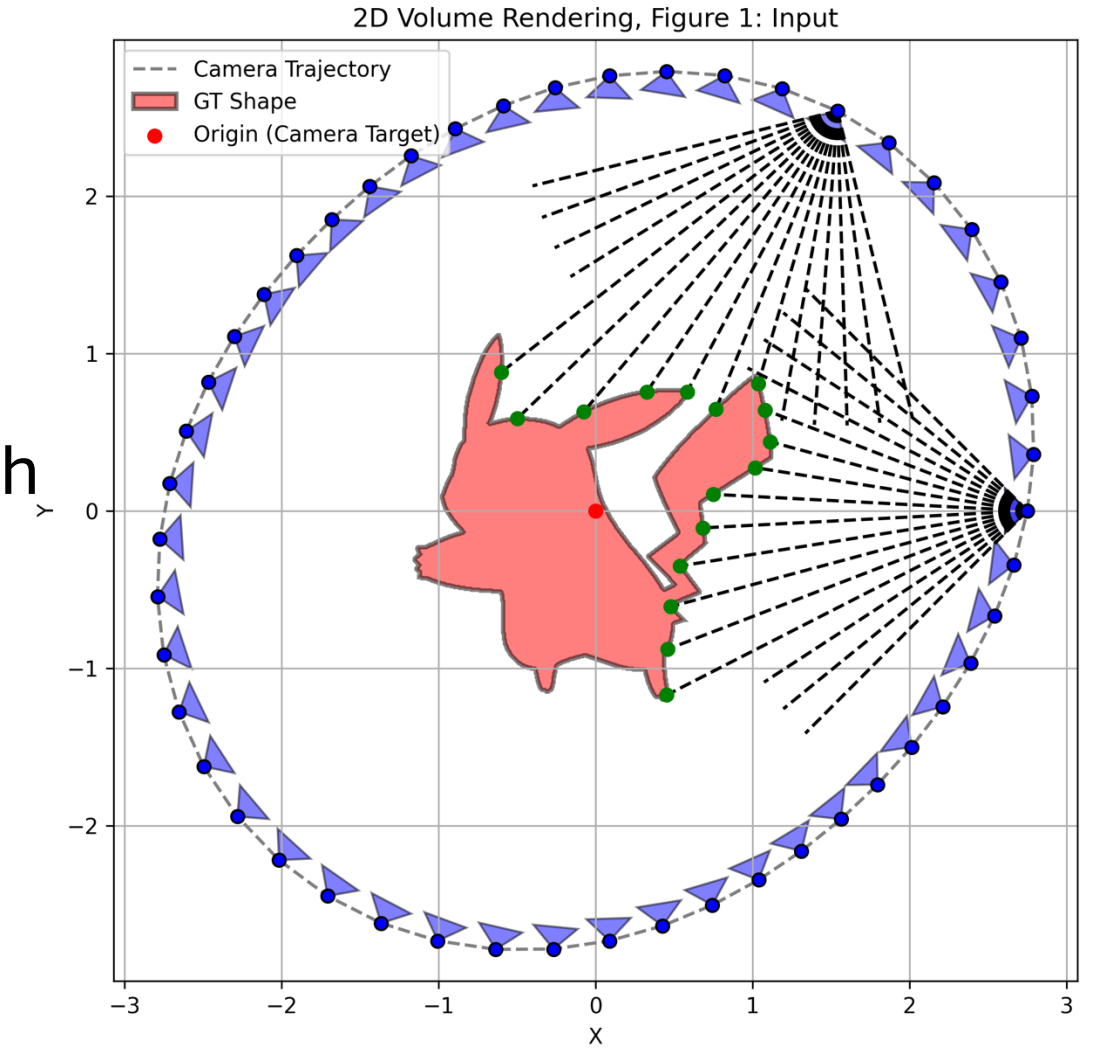
Shenlong Wang

Code will be available on our shared Github repo (Zhi-Hao will share later today)

Kinect Fusion in a 2D World

- Assuming we live in a 2D world, everyone perceives the 2D world through a 1D perspective imaging.
- One day, 2D computer scientists invented a '1.5D' camera, where each pixel captures the depth of the ray.

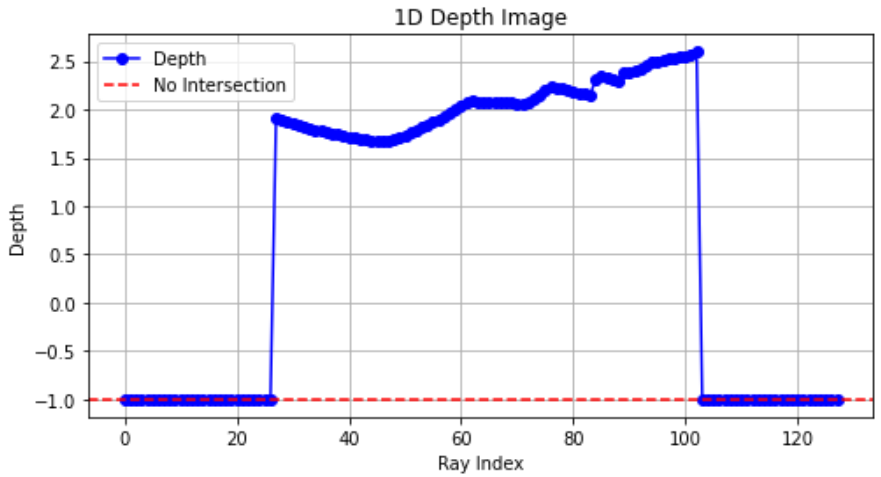


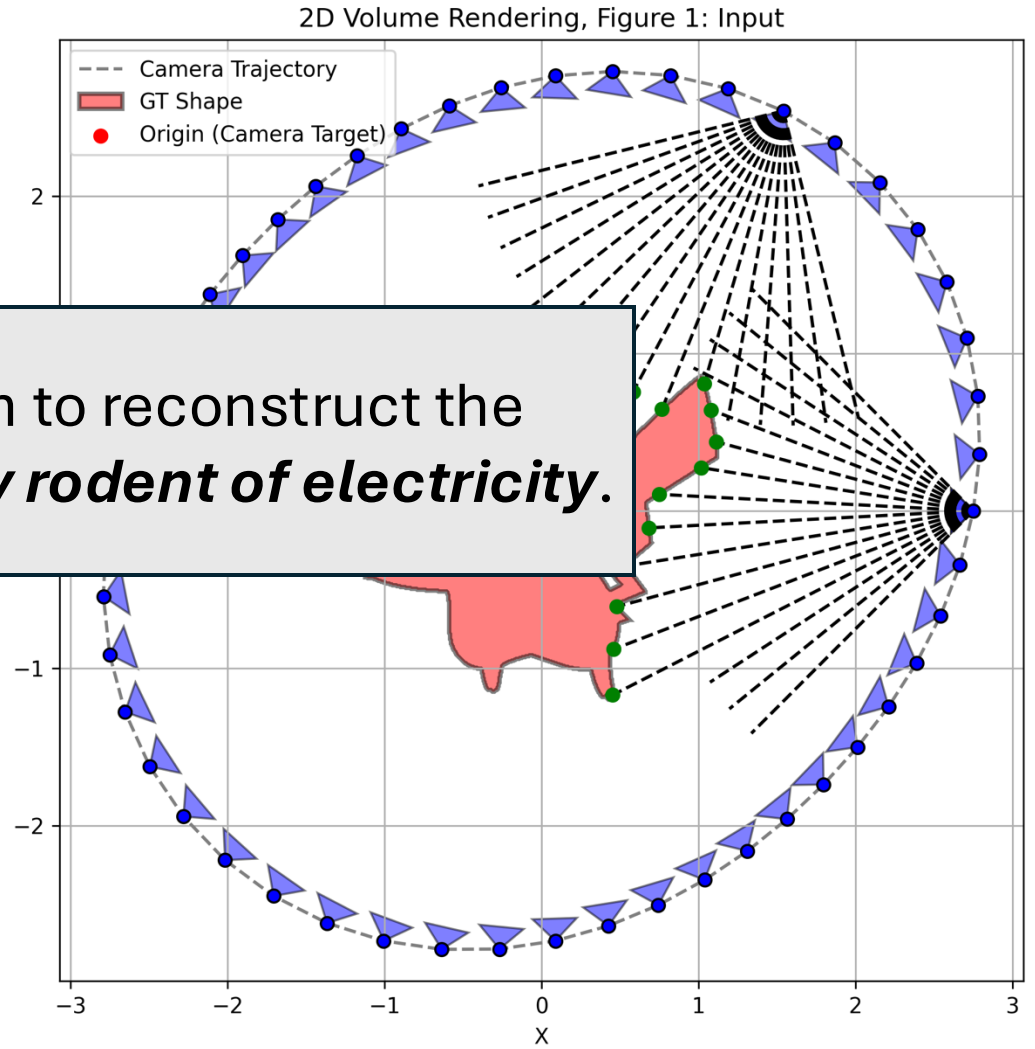


Kinect Fusion in a 2D World

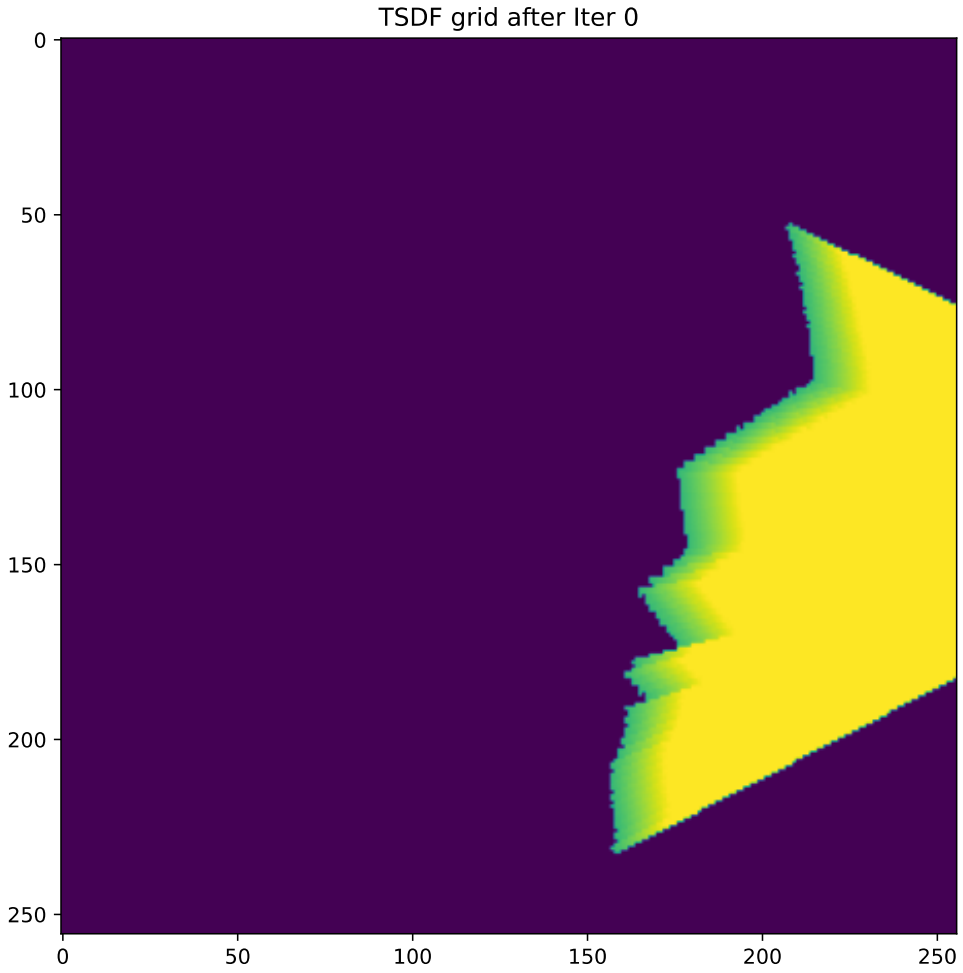
- Assuming we live in a 2D world, everyone perceives the 2D world through a 1D perspective imaging.
- One day, [illegible] invented [illegible] pixel cap[illegible]

They want to use this invention to reconstruct the shape of their giant idol – the ***holy rodent of electricity***.

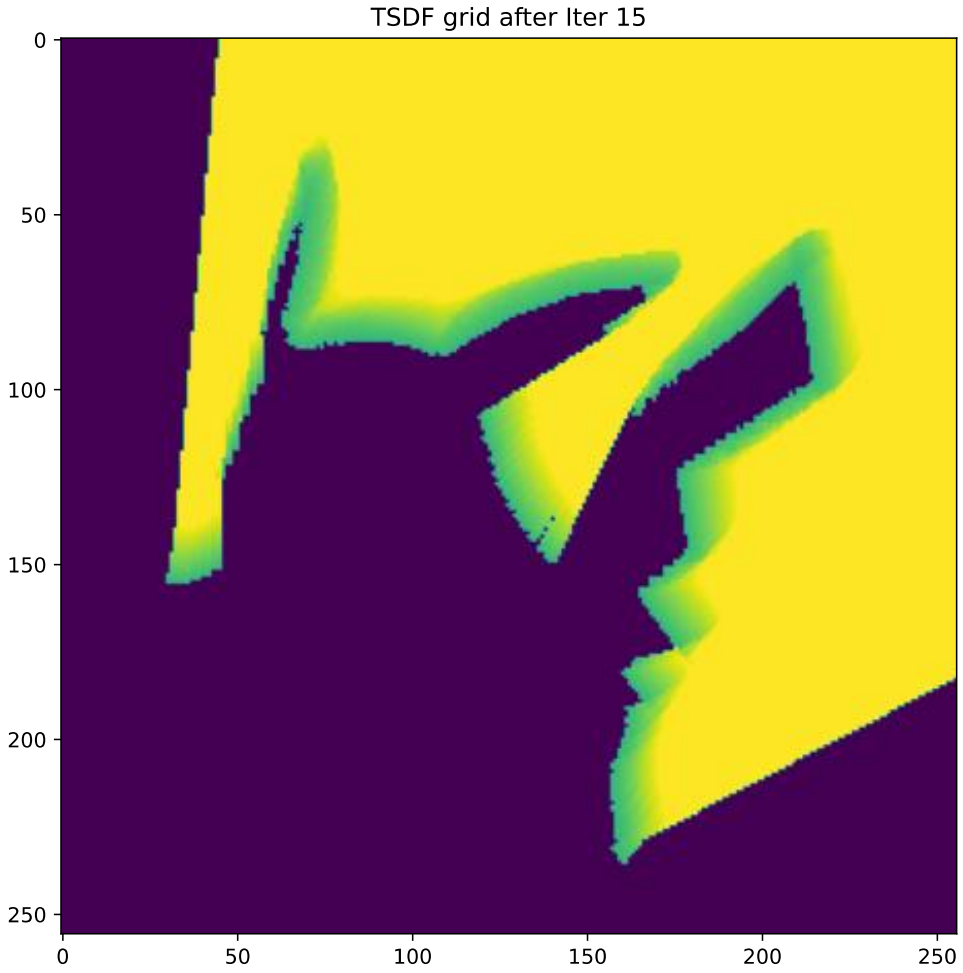




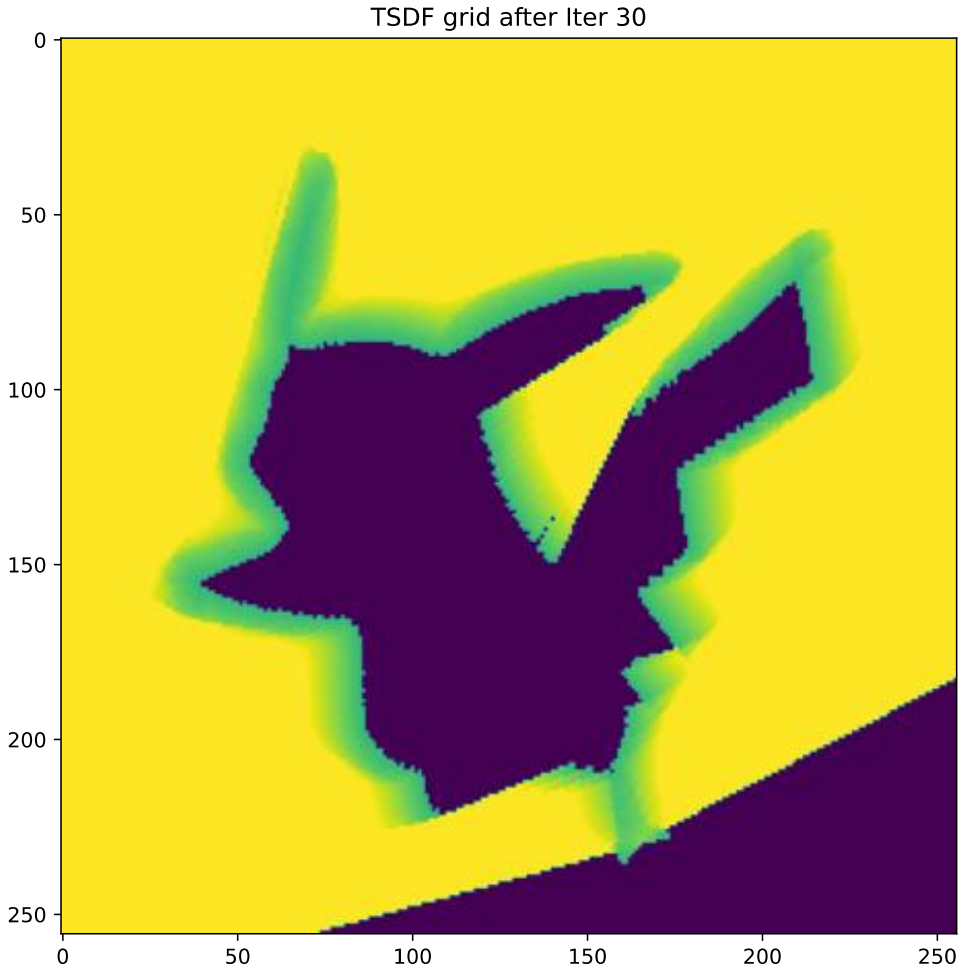
Hacker: Kinect Fusion in a 2D World



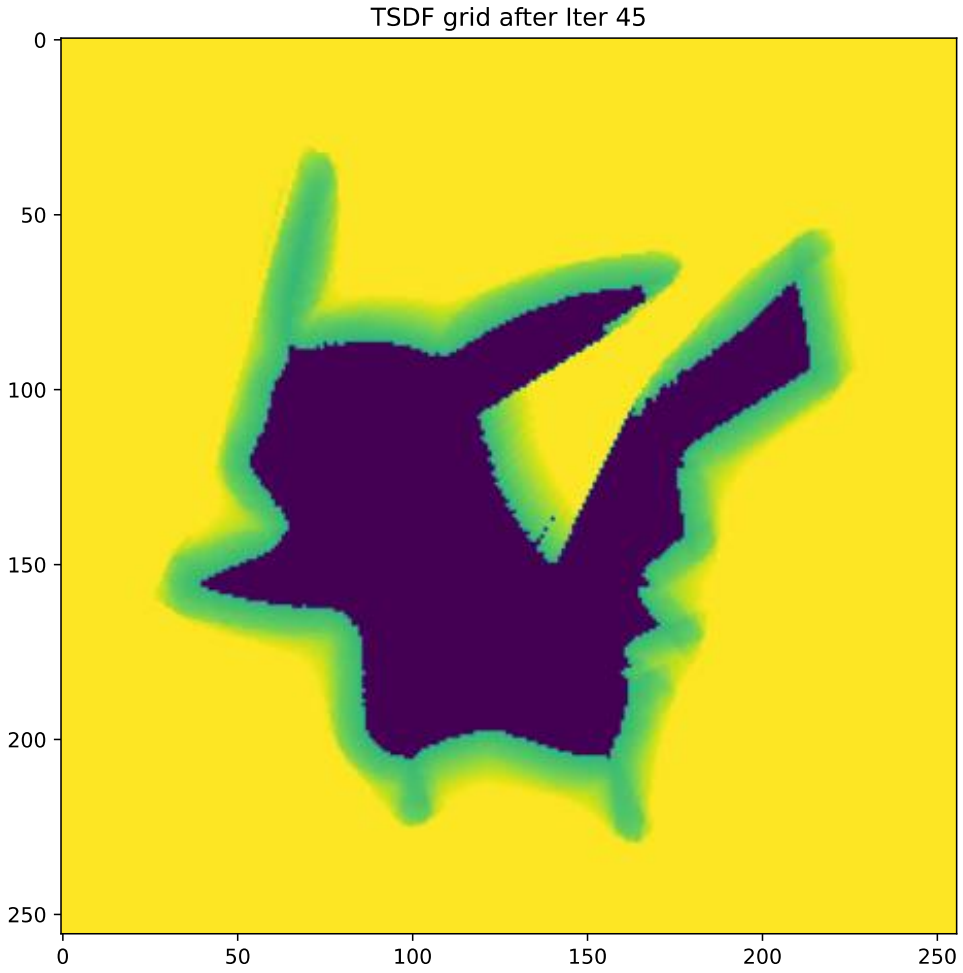
Hacker: Kinect Fusion in a 2D World



Hacker: Kinect Fusion in a 2D World

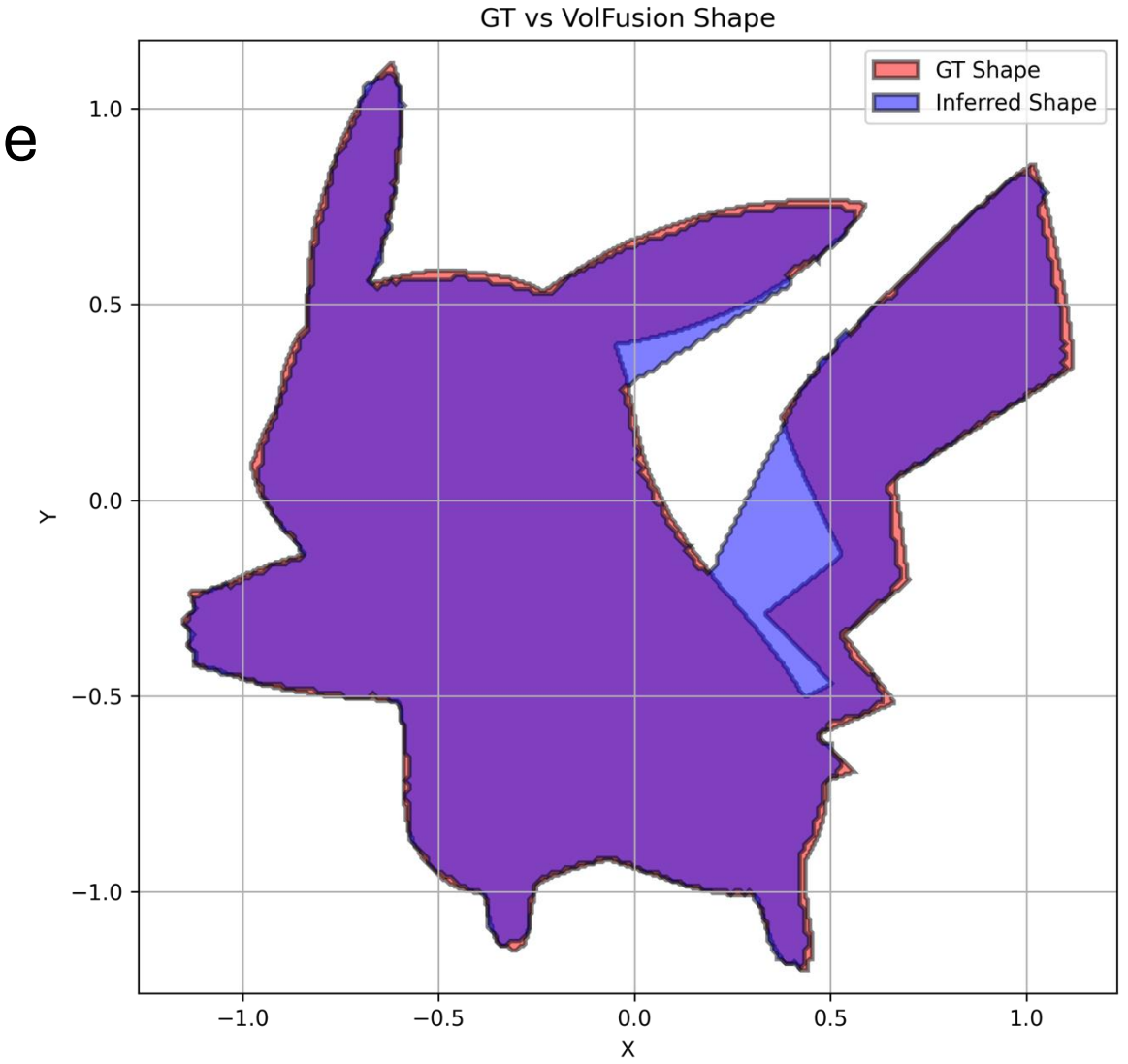


Hacker: Kinect Fusion in a 2D World



Hacker: Kinect Fusion in a 2D World

- Depth sensor has no noise but why it’s not perfect?
- Any idea to improve that?



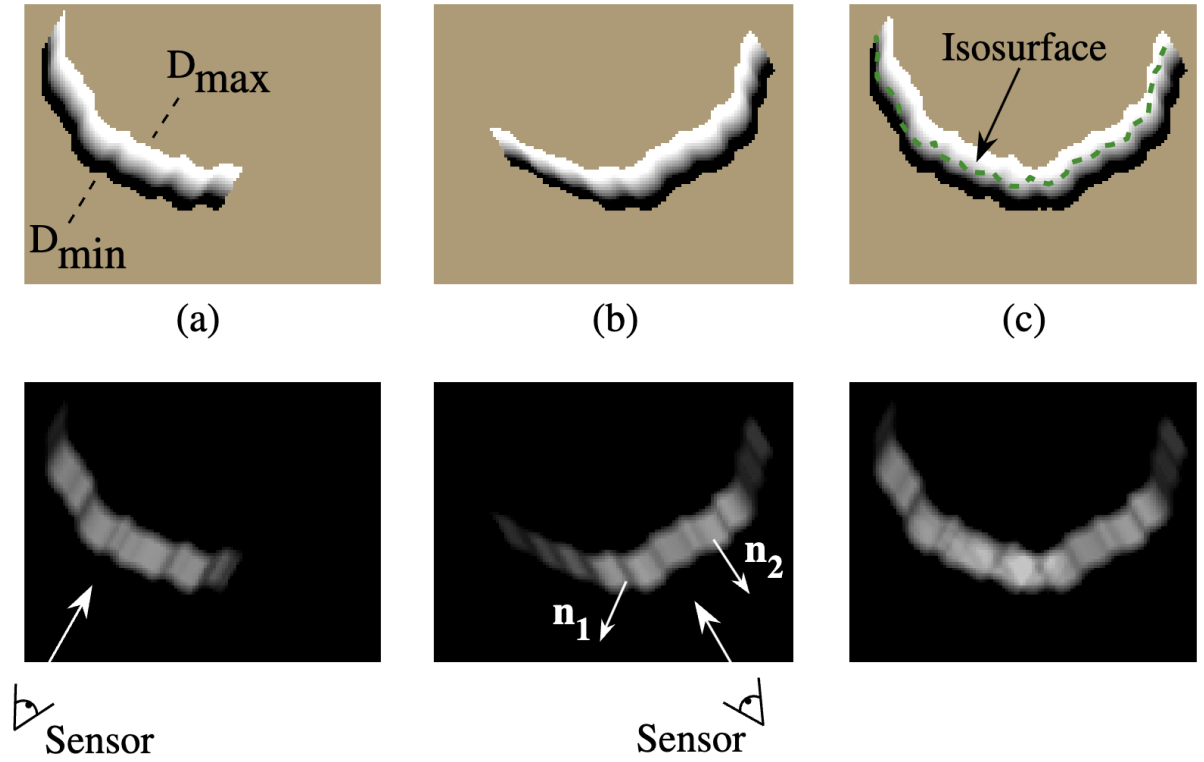
Where I could improve

I kept updating all free space & use a simple weight.

```python
def update_tsdf(tsdf_grid, weight_map, sdf, visibility_mask, trunc_threshold=0.1):
    """
    Update the TSDF grid and the weight map based on the new SDF values.

    Parameters:
    - tsdf_grid: The current TSDF grid.
    - weight_map: The current weight map.
    - sdf: The signed distance function values for the visible points.
    - visibility_mask: Mask of visible points in the grid.
    - trunc_threshold: The truncation threshold for the SDF.

    Returns:
    - Updated TSDF grid and weight map.
    """
    mask = visibility_mask & (sdf > -trunc_threshold)
    tsdf_grid[mask] += sdf[mask]
    weight_map[mask] += 1
    return tsdf_grid, weight_map
```



Points with normal facing sensor should get higher weight, why?

Lessons learned

- Depth sensor has no noise
  but why it’s not perfect?

- Any idea to improve that?