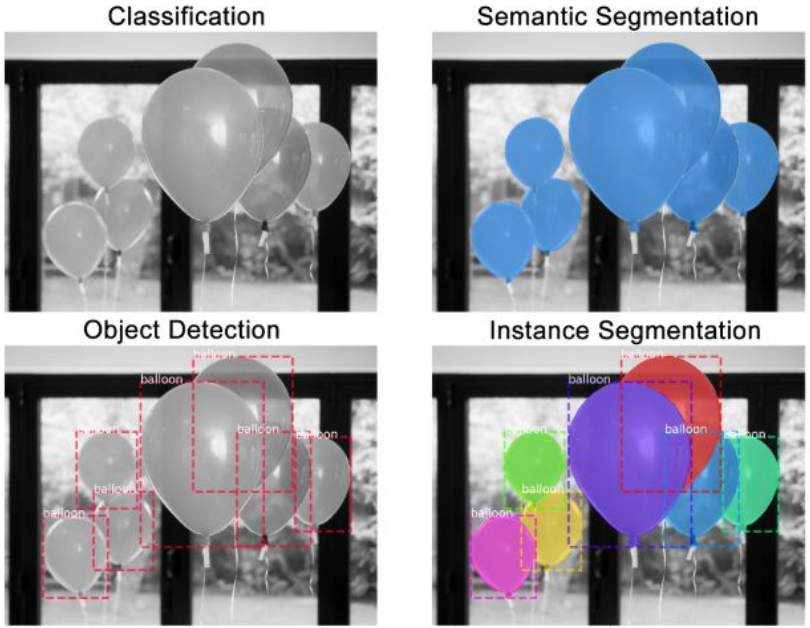
3D Scanning & Spatial Learning: Final presentation

‘Real-Time’ Instance Segmentation

Eissa Mostafa
Mustea Iulia-Otilia
Zidan Youssef

Introduction - What is Instance Segmentation?

- Classification - image belongs to a particular class
- Semantic segmentation - segmenting objects in the image based on classes
- Object detection - classes and the spatial location of those classes
- Instance segmentation - identification of boundaries of the objects
- Panoptic Segmentation - combines instance and semantic



Introduction - Volumetric Fusion

- The fusion of multiple depth maps allows to reconstruct the 3D surface from a set of range images
- Accumulate depth data into ‘model’ using poses (implicit surface representation)
- Image fusion methods incorporating a regular voxel space based on the equation from Curless and Levoy

$$D_{i+1}(\mathbf{x}) = \frac{W_i(\mathbf{x})D_i(\mathbf{x}) + w_{i+1}(\mathbf{x})d_{i+1}(\mathbf{x})}{W_i(\mathbf{x}) + w_{i+1}(\mathbf{x})}$$

$$W_{i+1}(\mathbf{x}) = W_i(\mathbf{x}) + w_{i+1}(\mathbf{x})$$

Task description

- reconstruction pipeline - e.g. VoxelHashing, Voxblox

- instance segmentation in 2D, fuse labels into the volumetric grid - instance merging

Motivation

Why not segment directly on 3D?

- It has the potential to run in realtime, implying it can assist in real-time applications like AR/VR, SLAM, planning, robotics (e.g. grasping)
- Plenty of progress done on 2D segmentation and vision

Experiments - Voxel Hashing

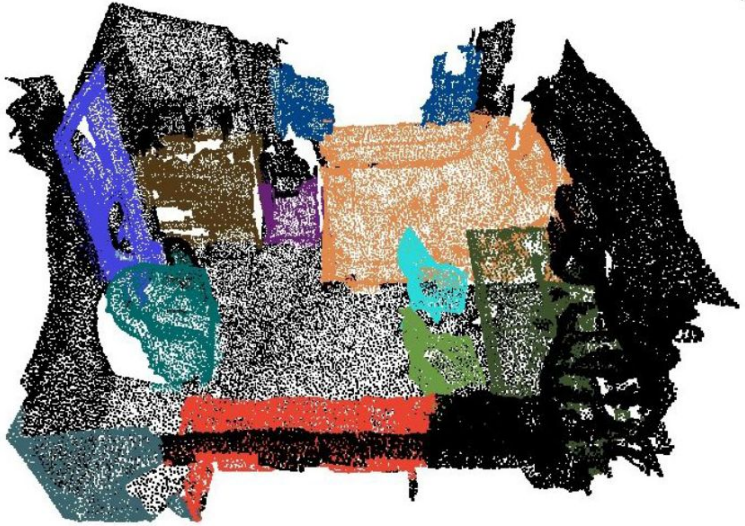
- Input: scene from ScanNet
- Used the VoxelHashing GPU version

VoxelHashing vs. Voxblox vs. ScanReal

1. **VoxelHashing**
   a. Pros
      - Runs in realtime
      - Predicts poses in real time
      - Integration with various sensors
   b. Cons
      - Lack of documentation available
      - Implementation difficulty - CUDA, DirectX
2. **Voxblox**
   a. Pros
      - Documentation available
      - ROS integration
      - Panoptic fusion uses it (has a specific interface for different fusion types)
      - Runs in realtime
   b. Cons
      - Doesn’t predict poses
      - Difficult to apply on custom datasets

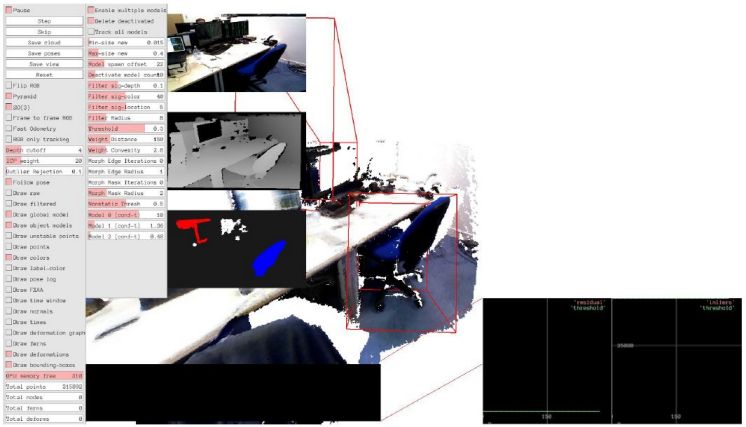
Experiments - 3D-SIS

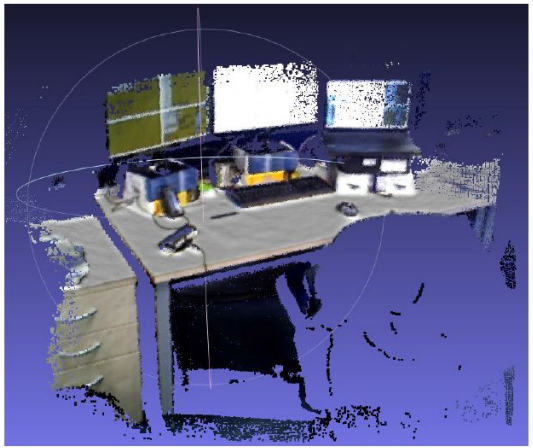
- Instance segmentation on RGB-D scan data
- Fuse both 2D RGB input features with 3D scan geometry features
- Joint feature training

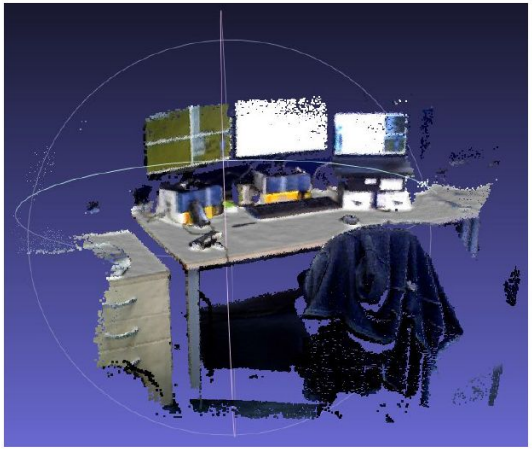


PanopticFusion-inst	0.478 10	0.667 12	0.712 11	0.595 6	0.259 13	0.550 13	0.000 19	0.613 5	0.175 12	0.250 14	0.434 5	0.437 3	0.411 11	0.857 7	0.48
Gaku Narita, Takashi Seno, Tomoya Ishikawa, Yohsuke Kaji: PanopticFusion: Online Volumetric Semantic Mapping at the Level of Stuff and Things. IROS 2019 (to appear)															
ResNet-backbone	0.459 11	**1.000** 1	0.737 8	0.159 17	0.259 12	0.587 11	0.138 4	0.475 12	0.217 9	0.416 8	0.408 8	0.128 12	0.315 13	0.714 12	0.41
MASC P	0.447 12	0.528 15	0.555 14	0.381 11	0.382 8	0.633 8	0.002 17	0.509 10	0.260 8	0.361 9	0.432 6	0.327 7	0.451 6	0.571 13	0.367
Chen Liu, Yasutaka Furukawa: MASC: Multi-scale Affinity with Sparse Convolution for 3D Instance Segmentation.															
3D-SIS P	0.382 13	**1.000** 1	0.432 15	0.245 14	0.190 14	0.577 12	0.013 14	0.263 14	0.033 17	0.320 12	0.240 14	0.075 15	0.422 10	0.857 7	0.117
ji Hou, Angela Dai, Matthias Niessner: 3D-SIS: 3D Semantic Instance Segmentation of RGB-D Scans. CVPR 2019															

Experiments - MaskFusion



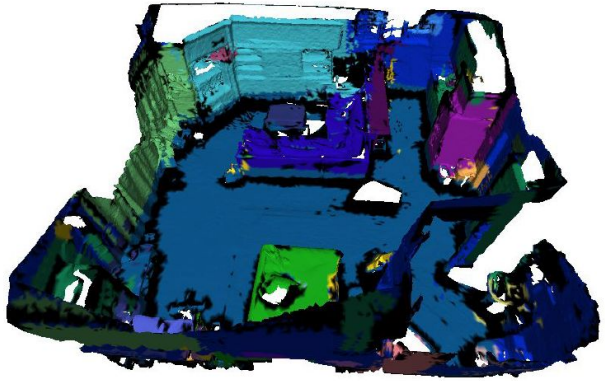




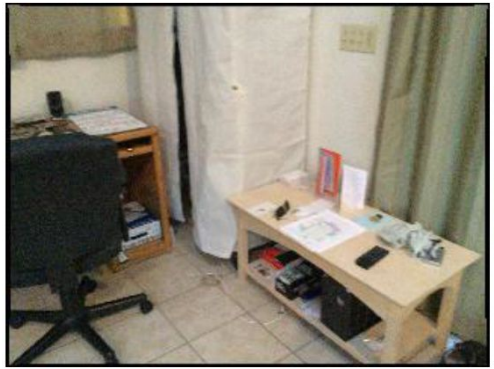
Con : No bechmark on ScanNet for MaskFusion, but PanopticFusion has benchmarks

Experiments

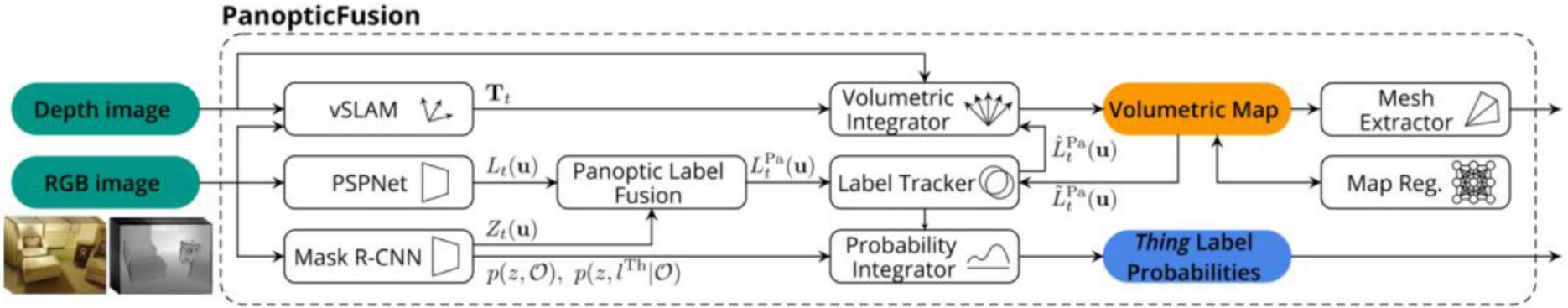
1. Backprojection of 2D masks to form 3D masks using depth and pose information



2. OpenGL pipeline for rendering 3D models compute the overlap between the 2D masks and rendered image



PanopticFusion - General overview



- Proven benchmarks on ScanNet (as opposed to MaskFusion)
- Panoptic labels (semantic + instance)
- Querying volumetric map for instance consistency based on thresholded
- IOU-based overlap

ScanNetV2 Dataset

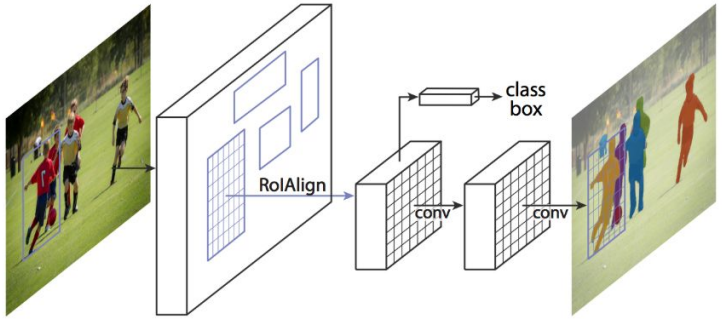
- scans of real-world scenes - reconstructed using BundleFusion
- contains RGB-D scans of 1513 scenes, comprising ≈ 2.5 million RGB-D frames

Contains (per scene):

1. 6DoF pose alingments + reconstructed models (BundleFusion)
2. manually-annotated instance-level semantic segmentation masks on the 3D mesh
3. used reduced 18 classes for instance segmentation evaluation

Instance Segmentation - Mask RCNN

- Flexible and general framework for 2D instance segmentation
- Predicts the segmentation masks and the bounding boxes
- Based on Faster R-CNN



Mask R-CNN framework. Source: https://arxiv.org/abs/1703.06870

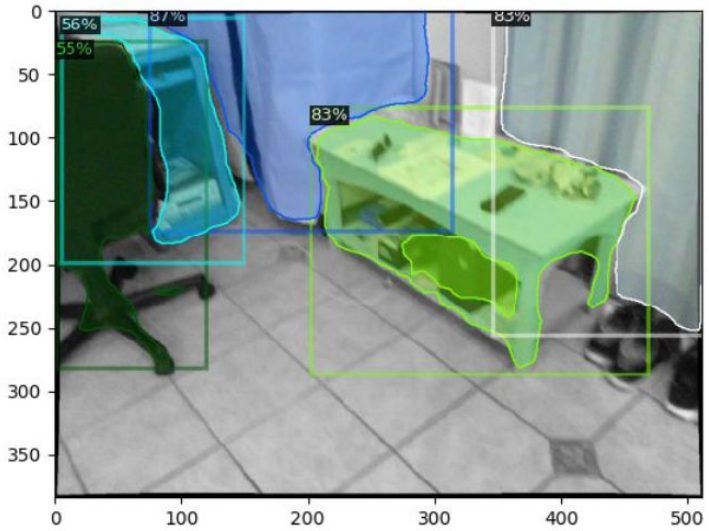
Instance Segmentation - Mask RCNN

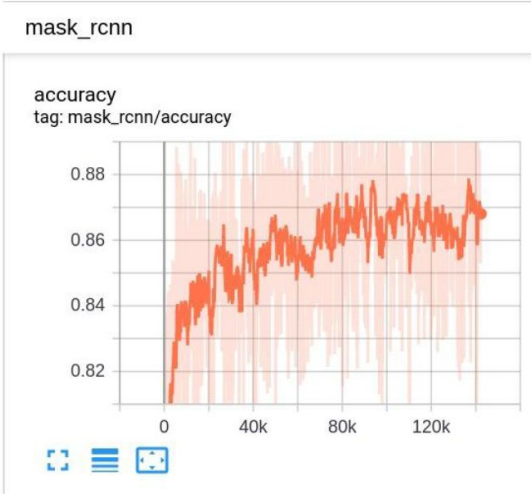
- **Training data pipeline:**
  - Convert ground truth masks to polygons
  - Calculate the bounding boxes from the ground truth
  - Filter the ignored objects and small masks (noise - len polygon >=6)

- **Training scheme:**
  - Mask R-CNN with ResNet-101 backbone
  - Pretrained initialization weights on COCO (Transfer learning)
  - Hyperparameters from PanopticFusion paper
  - 90-10 split of ScanNetV2 scenes (1513)
  - Reduced classes (18)
  - ScanNetV2 RGB images (every 20th frame)
  - Instance masks per object as polygons (every 20th frame)

Instance Segmentation - Mask RCNN

- Results





Instance Segmentation - Mask RCNN

- Validation
  - Difficulties in running default data evaluator implemented in Detectron2 with our dataset (ScanNet)
  - Training was implemented from scratch by calculating true positives, false positives and true negatives according to certain iou threshold to finally get the average precision.
  - Noise and missing objects (from the 18 classes) in the ground truth affect the evaluation (as false positives)

Instance Segmentation - Mask RCNN

- Validation

IOU Threshold	Scene	Value
0.5	scene0639_00	0.4448
0.5	scene0640_00	0.4244
0.5	scene0640_01	0.4341
0.5	scene0640_02	0.418
0.5	Average of scenes (639-648)	0.3987

PanopticFusion - Panoptic Label

$$L_t^{\mathrm{Pa}}(\mathbf{u}) = \begin{cases} Z_t(\mathbf{u}) & Z_t(\mathbf{u}) \neq l_{\mathrm{unk}} \\ L_t(\mathbf{u}) & Z_t(\mathbf{u}) = l_{\mathrm{unk}} \wedge L_t(\mathbf{u}) \in \mathcal{L}^{\mathrm{St}} \\ l_{\mathrm{unk}} & \mathrm{otherwise.} \end{cases} \quad (1)$$

- We use the semantic labels from the ground truth image
- Only evaluate on instance segmentation

PanopticFusion - Label Tracker (Association)

- IDs from 2D masks are not 3D consistent
- Use the volumetric map as a reference for id association
- Render a “reference” segmentation map (based on current pose)
- Compute association based on IOU overlap between masks in live and reference frame

$$\tilde{L}^{\mathrm{Pa}}_{t-1}(\mathbf{u}) = \mathrm{L}^{\mathrm{Pa}}_{t-1}(\mathbf{T}_t\mathbf{K}^{-1}D_t(\mathbf{u})[\mathbf{u},1]^{\mathrm{T}}).$$

- Usually noisy, especially for large voxel sizes
- Noise due to invalid depth values and unupdated voxels (due to low resolution)
- Filtering techniques like a median filter improve the result
- However the best configuration was to only assign the noisy pixels with their neighbor

PanopticFusion - Label Tracker (Association)

```
Threshold = 0.25
For live_mask in sorted(live_masks)[::-1]:
        Max_iou, argmax
        Found_match = false
        For pred_mask in pred_masks and pred_mask was never associated in the current frame:
                Curr_iou = computeIOU(live_mask, current_mask)
                if(curr_iou > threshold and curr_iou > max_iou
                        Max_iou = curr_iou, argmax = pred_mask
        If not found_match: # This is supposedly the first time we've seen this object
                Create new id and associate it with live_mask
        Else
                Associate pred_mask with argmax
```

Finally, create a "consistent" segmentation map based on the consistent IDs

PanopticFusion - Label Fusion

- Ray cast from sensor origin to voxel space
- Fuse color and depth
- Can’t fuse label IDs in the same manner
- “Voting” mechanism is costly
- Use the update weight to increase or decrease the confidence of a voxel about its label
    - Increase when the casted pixel has the same label as the voxel and vice versa
    - Change the label in the voxel when the update weight is bigger than the voxel’s weight
    - Robustness to noise
    - Improves segmentation over time

$$\mathtt{D}_t(\mathbf{v}) = \frac{\mathtt{W}^{\mathrm{D}}_{t-1}(\mathbf{v})\mathtt{D}_{t-1}(\mathbf{v}) + w_t(\mathbf{v}, \mathbf{p_u}) d_t(\mathbf{v}, \mathbf{p_u}, \mathbf{s})}{\mathtt{W}^{\mathrm{D}}_{t-1}(\mathbf{v}) + w_t(\mathbf{v}, \mathbf{p_u})},$$

$$\mathtt{W}^{\mathrm{D}}_t(\mathbf{v}) = \mathtt{W}^{\mathrm{D}}_{t-1}(\mathbf{v}) + w_t(\mathbf{v}, \mathbf{p_u}).$$

- Probability integration

$$p_{1\cdots t}(z, l^{\mathrm{Th}}) = \frac{\sum_t p_t(z, \mathcal{O}) p_t(z, l^{\mathrm{Th}} | \mathcal{O})}{\sum_t p_t(z, \mathcal{O})}.$$

$$\mathtt{L}^{\mathrm{Pa}}_t(\mathbf{v}) = \mathtt{L}^{\mathrm{Pa}}_{t-1}(\mathbf{v}), \;\; \mathtt{W}^{\mathrm{L}}_t(\mathbf{v}) = \mathtt{W}^{\mathrm{L}}_{t-1}(\mathbf{v}) + w_t(\mathbf{v}, \mathbf{p_u}).$$

PanopticFusion - Additional Details

- Voxel size: 0.024 and truncation distance: 4 * voxel size, as recommended by the paper
- Filtering out small mask sizes (usually noise) in the association step, and in the evaluation (equivalent to post-processing the map)
- Image dimensions used for fusion: 320 x 240
- Threshold:0.25 for IoU, as recommended by the paper

CRF Volumetric Map Regularization

- CRF Represents a conditional distribution P(X|I) defined by the graph's topology (Cliques)
- Dense (fully connected), i.e. every voxel is connected to every other voxel

$$E(\mathbf{x}) = \sum_{v} \psi_u(x_v) + \sum_{v<v'} \psi_p(x_v, x_{v'}). \qquad (12)$$

CRF Volumetric Map Regularization

Unary clique potentials:

$$\psi_u(x_v) = -\log p(x_v).$$

Binary clique potentials:

$$\psi_p(x_v, x_{v'}) = \mu(x_v, x_{v'}) \sum_{---} w^{(m)} k^{(m)}(\mathbf{f}_v, \mathbf{f}_{v'}).$$

$$k^{(1)}(\mathbf{f}_v, \mathbf{f}_{v'}) = \exp\left(-\frac{|\mathbf{v} - \mathbf{v}'|^2}{2\theta_\alpha^2} - \frac{|\mathtt{C}(\mathbf{v}) - \mathtt{C}(\mathbf{v}')|^2}{2\theta_\beta^2} \right), \tag{15}$$

$$k^{(2)}(\mathbf{f}_v, \mathbf{f}_{v'}) = \exp\left(-\frac{|\mathbf{v} - \mathbf{v}'|^2}{2\theta_\alpha^2} \right). \tag{16}$$

$$p(x_v = \mathtt{L}_t^{\mathrm{Pa}}(\mathbf{v})) = \frac{\sum_{t \in \mathcal{T}_+} w_t(\mathbf{v}, \mathbf{p_u})}{\sum_{t \in \mathcal{T}_+} w_t(\mathbf{v}, \mathbf{p_u}) + \sum_{t \in \mathcal{T}_-} w_t(\mathbf{v}, \mathbf{p_u})}$$

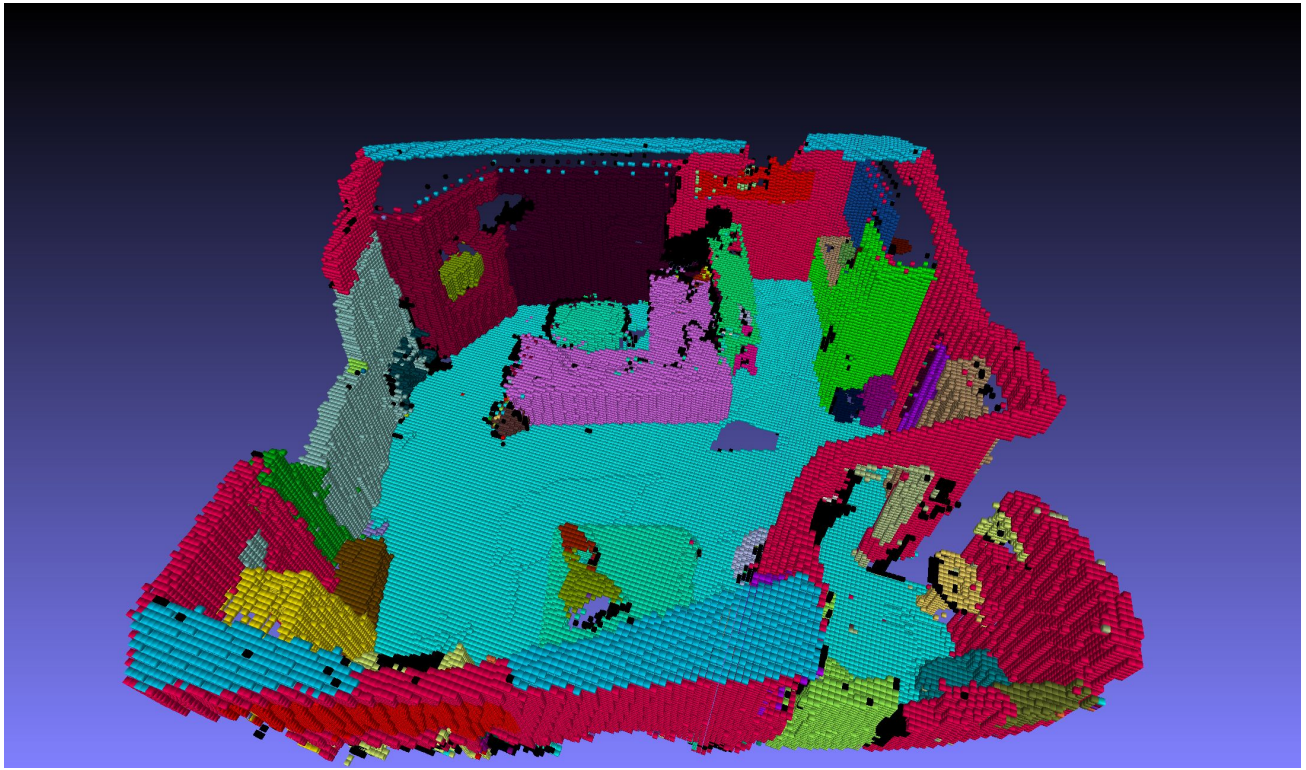
$$\simeq \frac{1}{2}\left(1 + \frac{\mathtt{W}_t^{\mathrm{L}}(\mathbf{v})}{\mathtt{W}_t^{\mathrm{D}}(\mathbf{v})}\right). \tag{19}$$

CRF Volumetric Map Regularization - Implementation

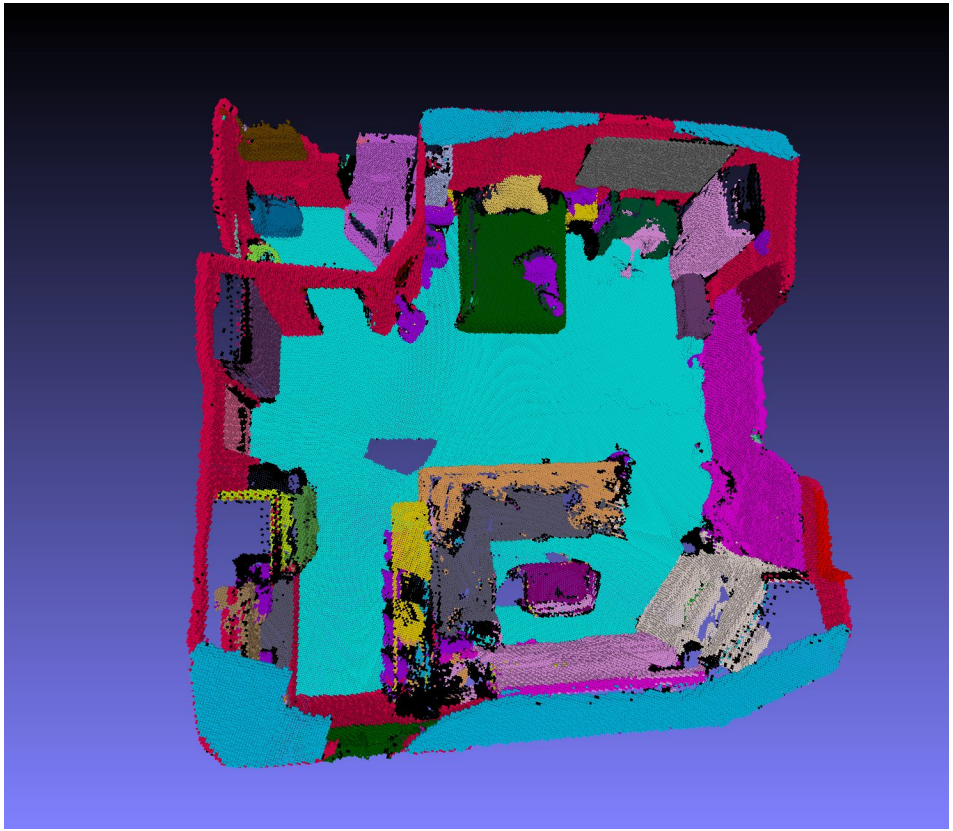
- Used DenseCRF (http://graphics.stanford.edu/projects/drf/) → extended for 3D
- Apply it every 90 frames (simulating 3 seconds like what the paper recommended)
- To make it more memory efficient, we only consider label ids (stored in the voxel grid) that have a mask size bigger than a certain threshold
  - Usually these are a result of some association errors (like a new object appearing) or noise
- And consider only voxels that have been updated before and their 3x3x3 neighborhood

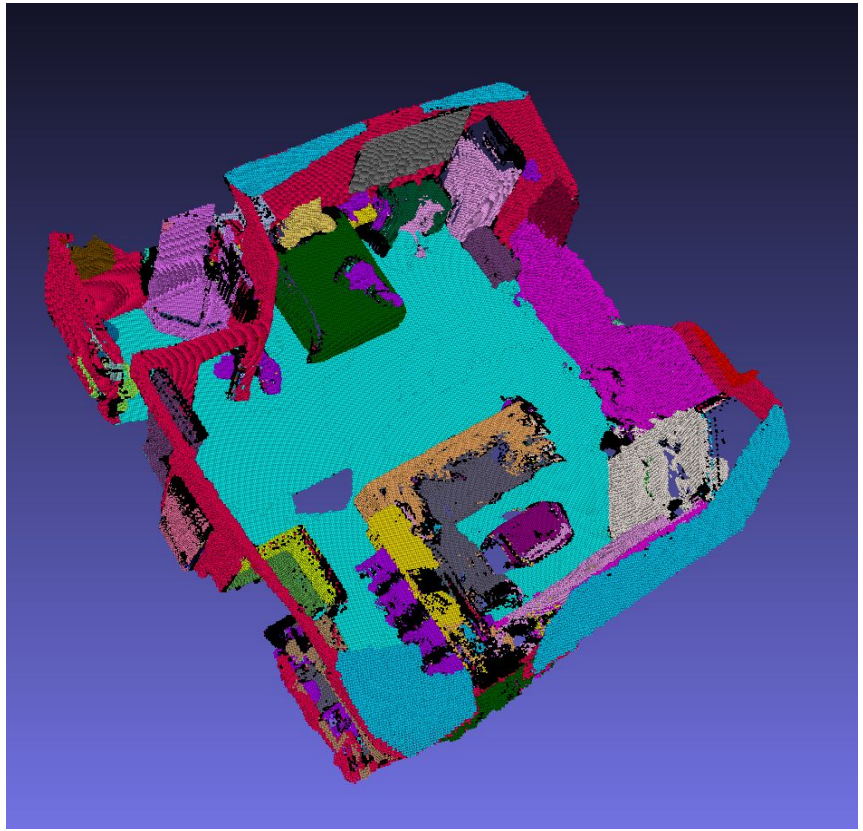
Panoptic fusion - with ground truth labels (without CRF)

- Implemented on ScanReal
- Result: labels + instances taken from the ScanNetV2 labels + instances (ground truth)

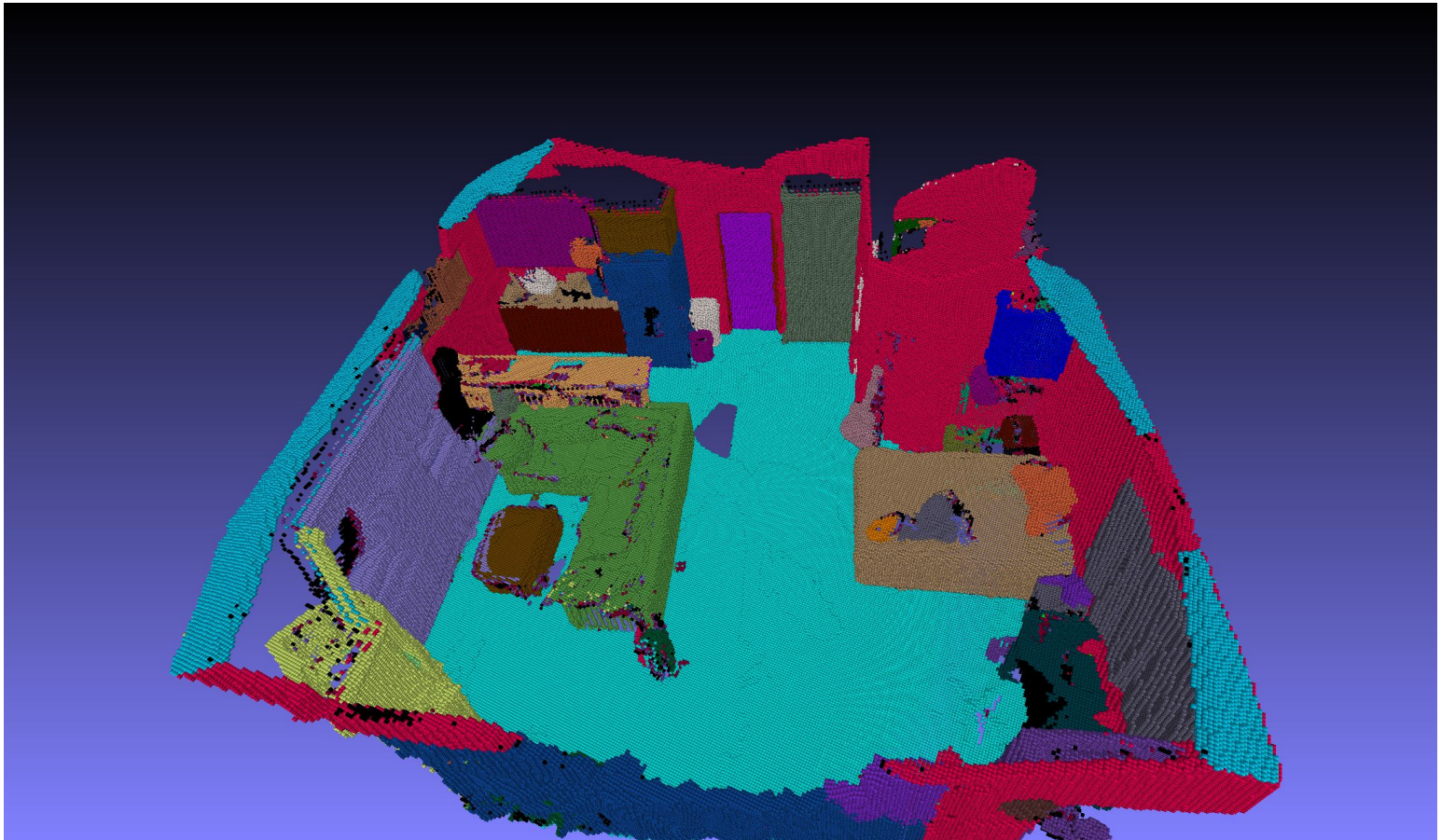


Panoptic fusion - with labels from Detectron2 (without CRF)

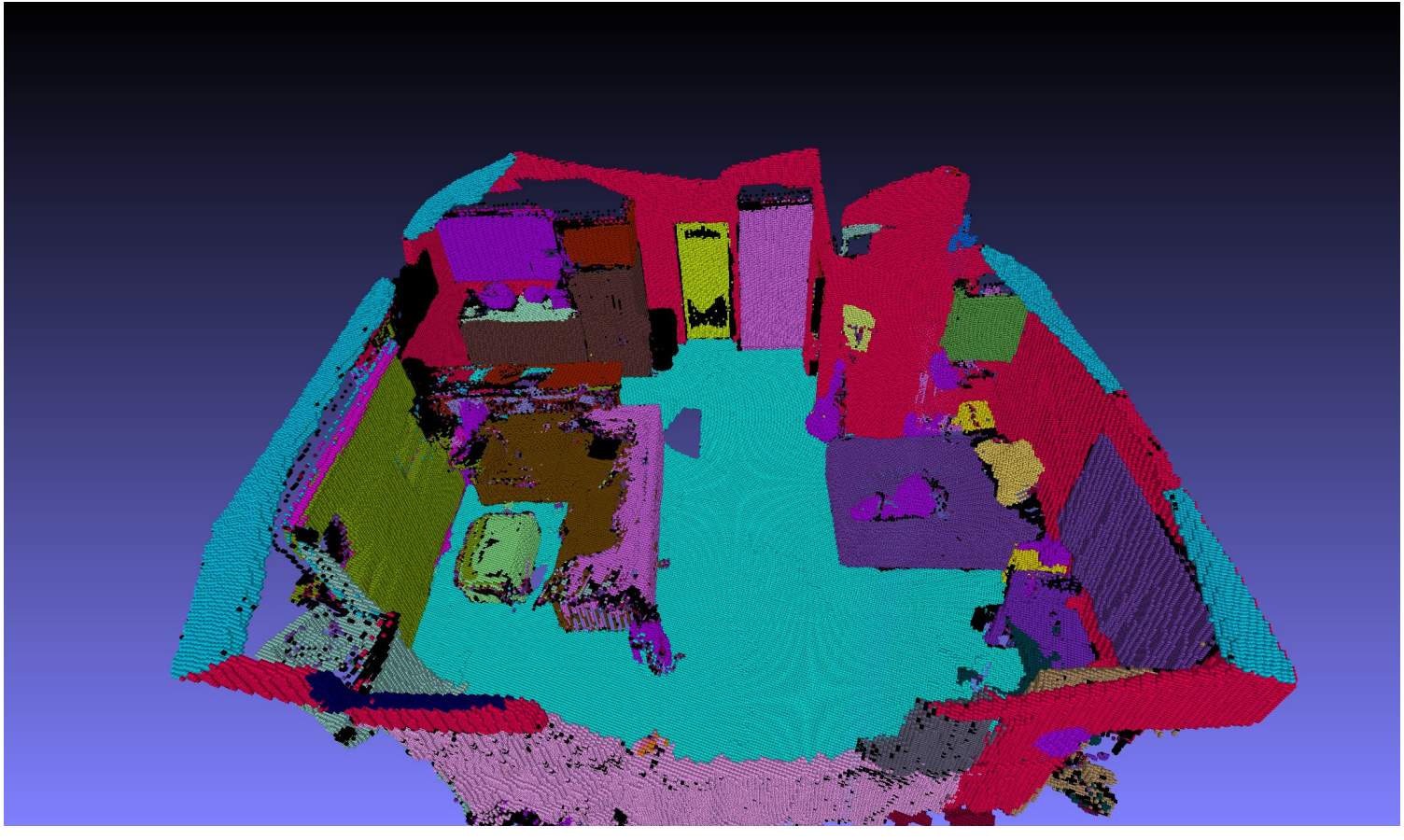




Panoptic fusion - with ground truth labels with CRF



Panoptic fusion - with labels from Detectron2 with CRF



Results - Benchmarking

- Implemented COCO Average precision metric for 3D evaluation from scratch
- For 3D, used ground truth 2D labels and reconstruct the voxel grid using ScanReal (the 2D labels GT are 3D consistent)

AP@50: 0.45 - for scene_0

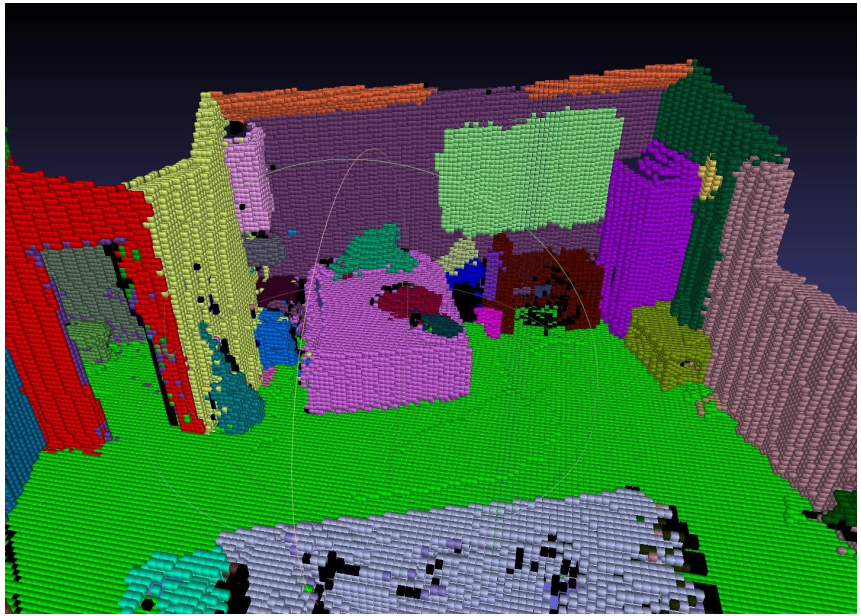
AP@50: 0.46 - for scene_0 with CRF

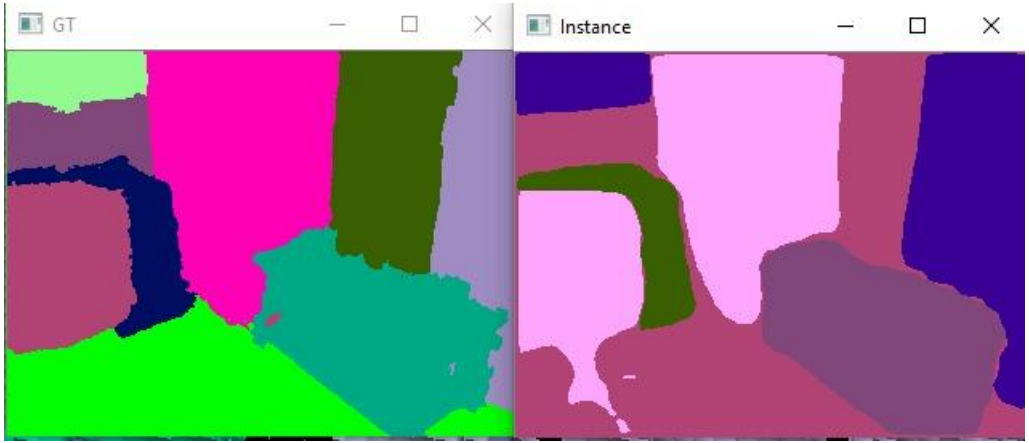
PanopticFusion-inst	0.478 10	0.667 12	0.712 11	0.595 6	0.259 13	0.550 13	0.000 19	0.613 5	0.175 12	0.250 14	0.434 5	0.437 3	0.411 11	0.857 7	0.48

Gaku Narita, Takashi Seno, Tomoya Ishikawa, Yohsuke Kaji: PanopticFusion: Online Volumetric Semantic Mapping at the Level of Stuff and Things. IROS 2019 (to appear)

Results - Benchmarking Issues

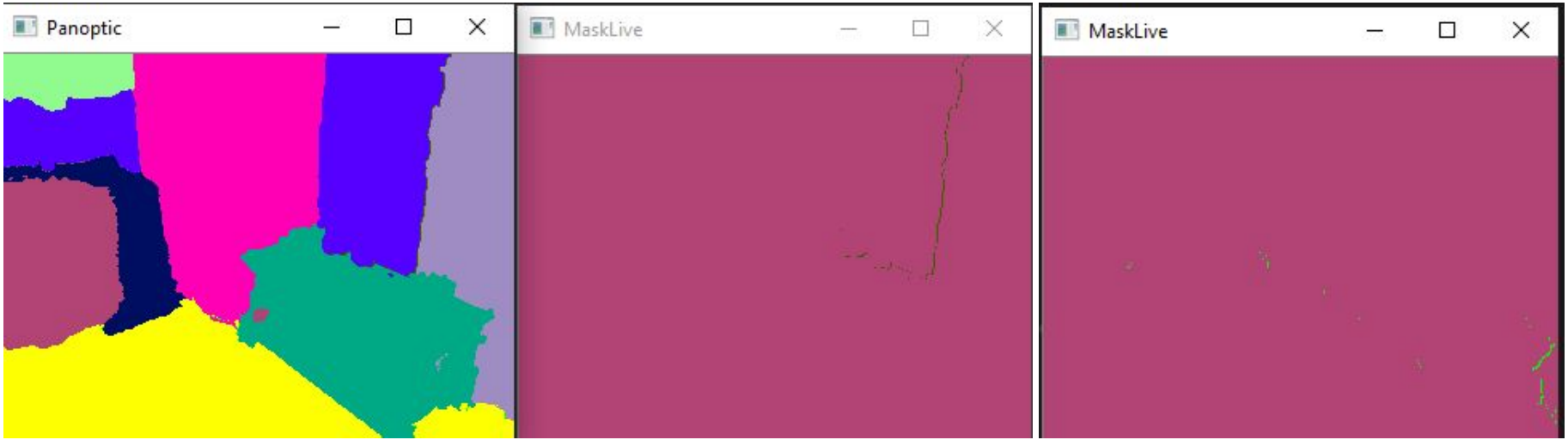
- Removed objects from the 18 classes (e.g. “chair”)





Results - Benchmarking Issues

- Noise in the ground truth



Directions for Future Work

- Inspired by “Self supervised modality adaptation”, combine RGB and Depth maps for 2D instance segmentation
- To support real-time capabilities, instead of segmenting each frame using MaskRCNN (or similar), segment every k-th frame, and use pose information, optical flow, depth, etc. to infer segmentation maps in other frames (i.e. tracking objects)
- Learn on how to associate IDs instead of relying purely on heuristics (should be similar to adding weights on the calculated iou and finding optimal threshold value)
- Incorporate semantic information in pose estimation
- Train Detectron2 on all high resolution images

CU-Hybrid-2D Net	**0.636** 1	**0.825** 1	**0.820** 1	0.179 13	**0.648** 1	**0.463** 1	**0.549** 1	**0.742** 1	**0.676** 1	**0.628** 1	**0.961** 1
DMMF_3d	0.605 2	0.651 4	0.744 5	**0.782** 1	0.637 2	0.387 2	0.536 2	0.732 2	0.590 3	0.540 2	0.856 9
DMMF	0.597 3	0.543 8	0.755 4	0.749 2	0.585 4	0.338 4	0.494 4	0.704 4	0.598 2	0.494 8	0.911 4
MCA-Net	0.595 4	0.533 9	0.756 3	0.746 3	0.590 3	0.334 6	0.506 3	0.670 5	0.587 4	0.500 6	0.905 6
RFBNet	0.592 5	0.616 5	0.758 2	0.659 4	0.581 5	0.330 7	0.469 5	0.655 8	0.543 7	0.524 3	0.924 2
DCRedNet	0.583 6	0.682 3	0.723 6	0.542 7	0.510 8	0.310 9	0.451 6	0.668 6	0.549 6	0.520 4	0.920 3
SSMA C	0.577 7	0.695 2	0.716 8	0.439 9	0.563 6	0.314 8	0.444 7	0.719 3	0.551 5	0.503 5	0.887 8

Acknowledgements

Ji Hou

Thank you for your attention!

Q&A