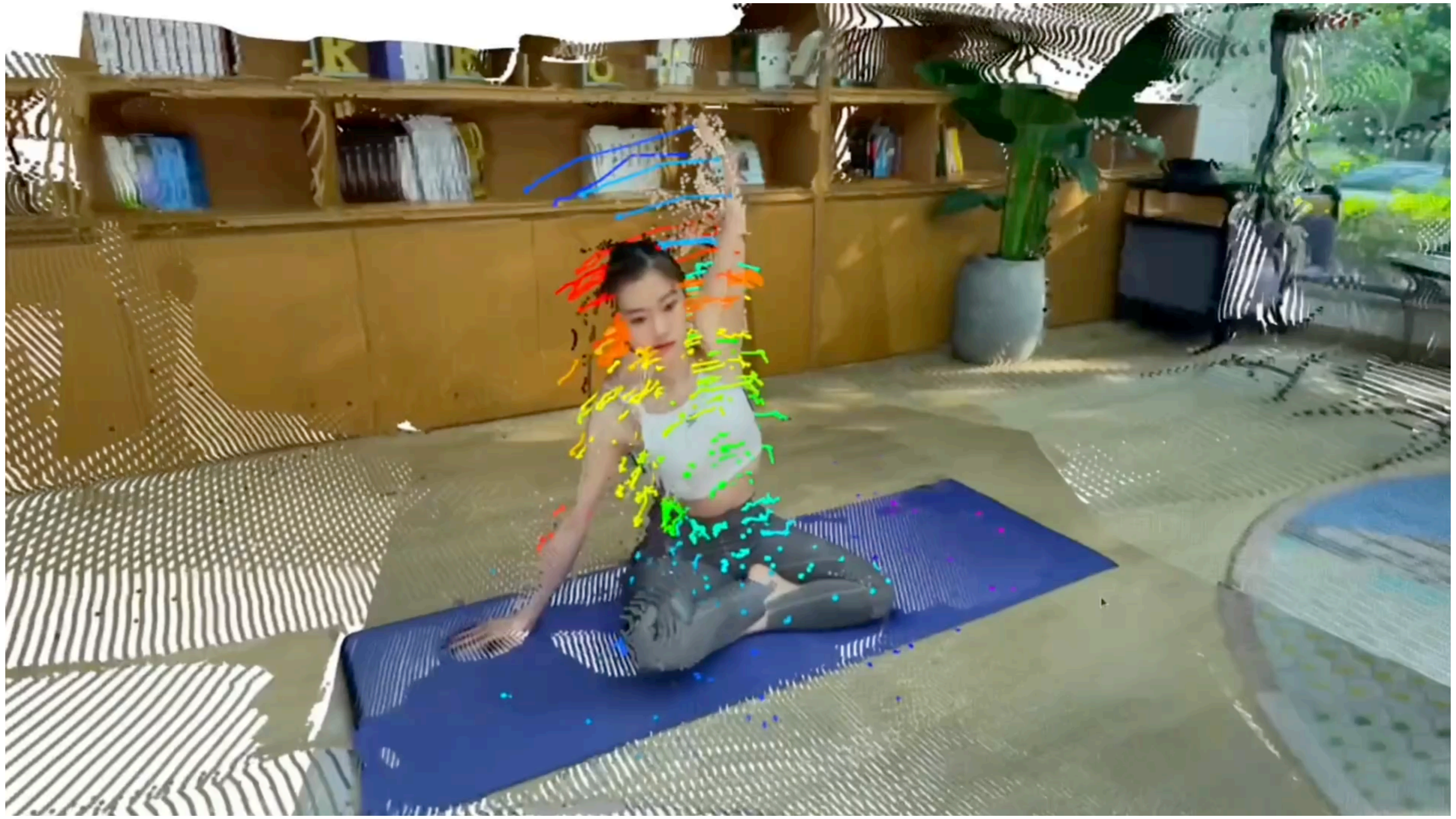


MVTracker: Multi-View 3D Point Tracking

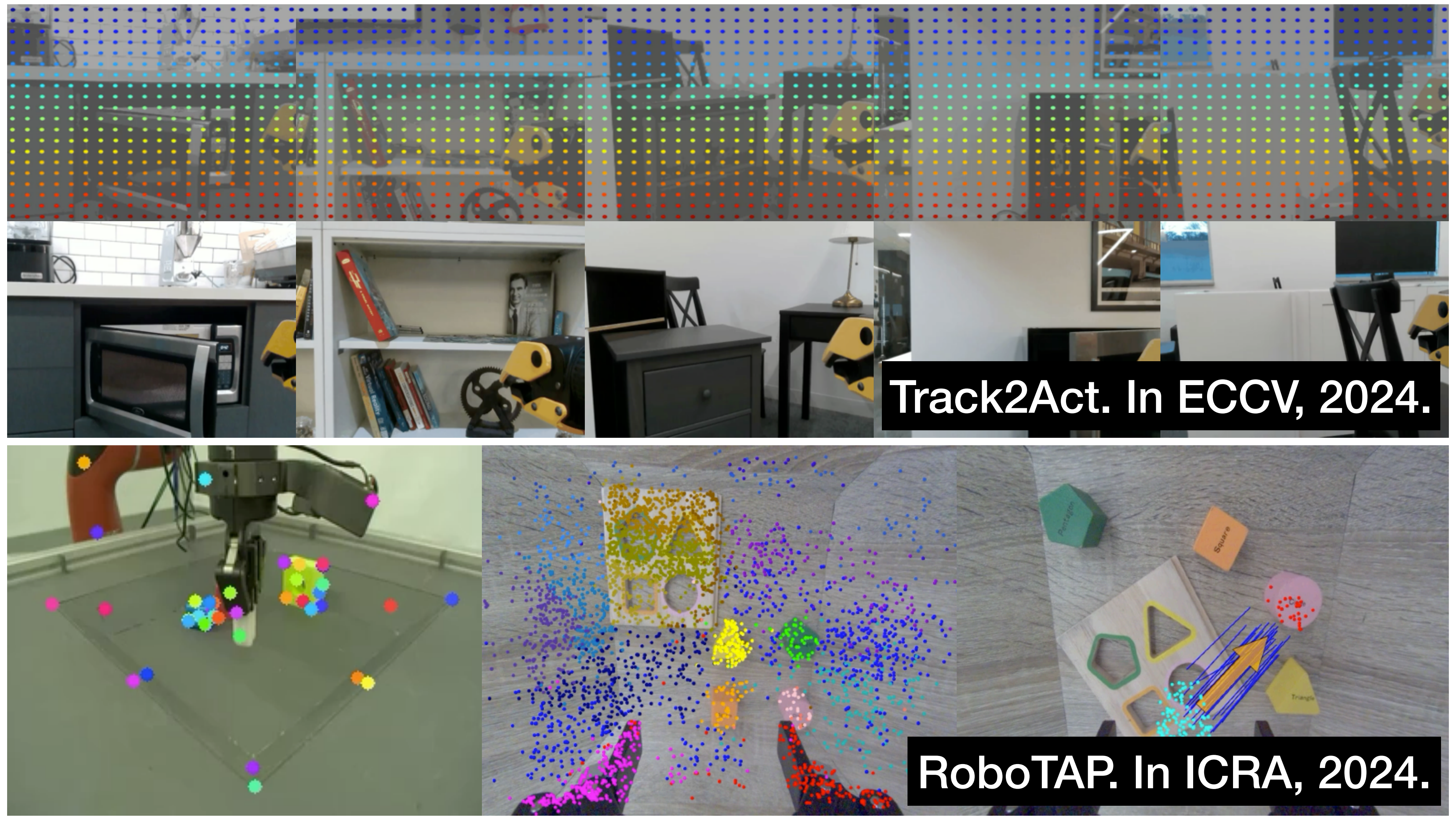
Frano Rajič[1], Haofei Xu[1], Marko Mihajlović[1], Siyuan Li[1], Irem Demir[1],
Emircan Gündoğdu[1], Lei Ke[2], Sergey Prokudin[1,3], Marc Pollefeys[1,4], Siyu Tang[1]
[1]ETH Zürich, [2]Carnegie Mellon University, [3]Balgrist University Hospital, [4]Microsoft

the world is dynamic

Sources: Pexels, Pixabay.

Source: Toyota Research Institute: Teaching Robots New Behaviors.

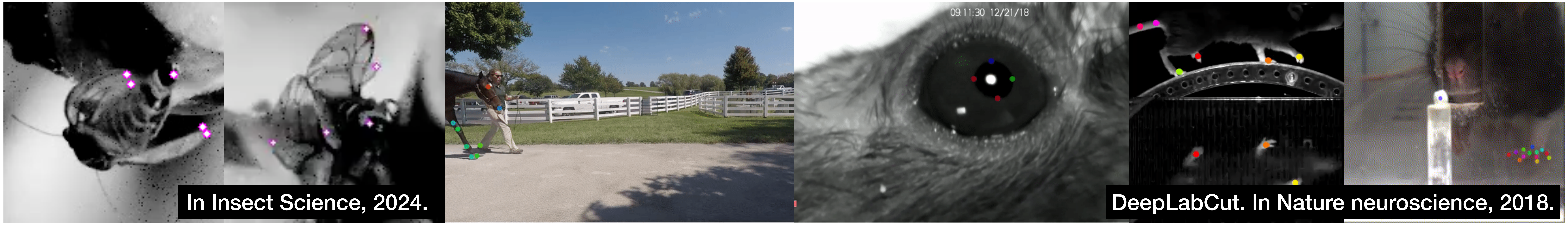
Dynamics perception in robotics and beyond


Track2Act. In ECCV, 2024.
RoboTAP. In ICRA, 2024.

Robotics: Steering actions. Less data needed.


CORI Surgical System. MyMichigan Health (CC BY-NC-ND).

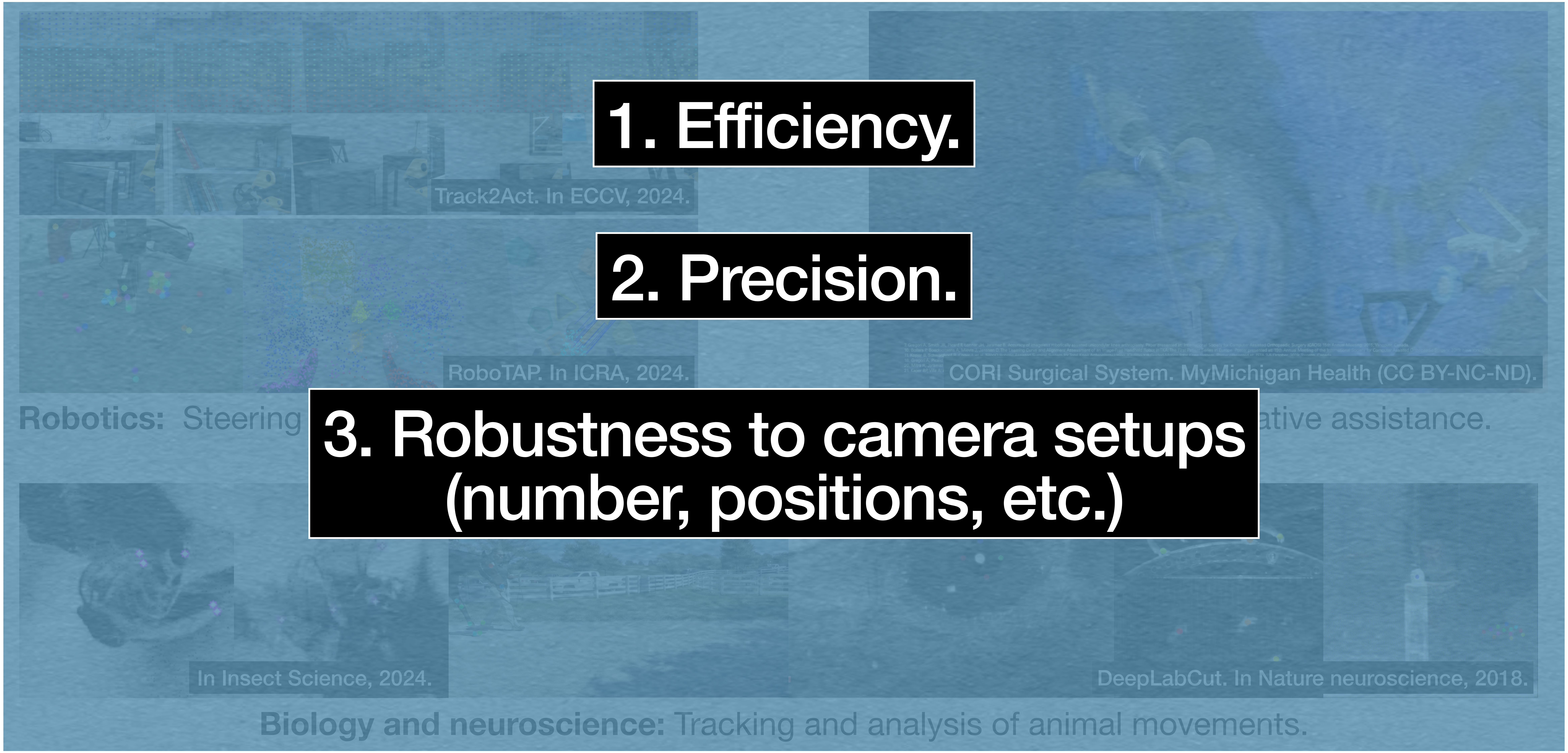
Surgical room: Intra-operative assistance.



In Insect Science, 2024.
DeepLabCut. In Nature neuroscience, 2018.

Biology and neuroscience: Tracking and analysis of animal movements.

Dynamics perception in robotics and beyond



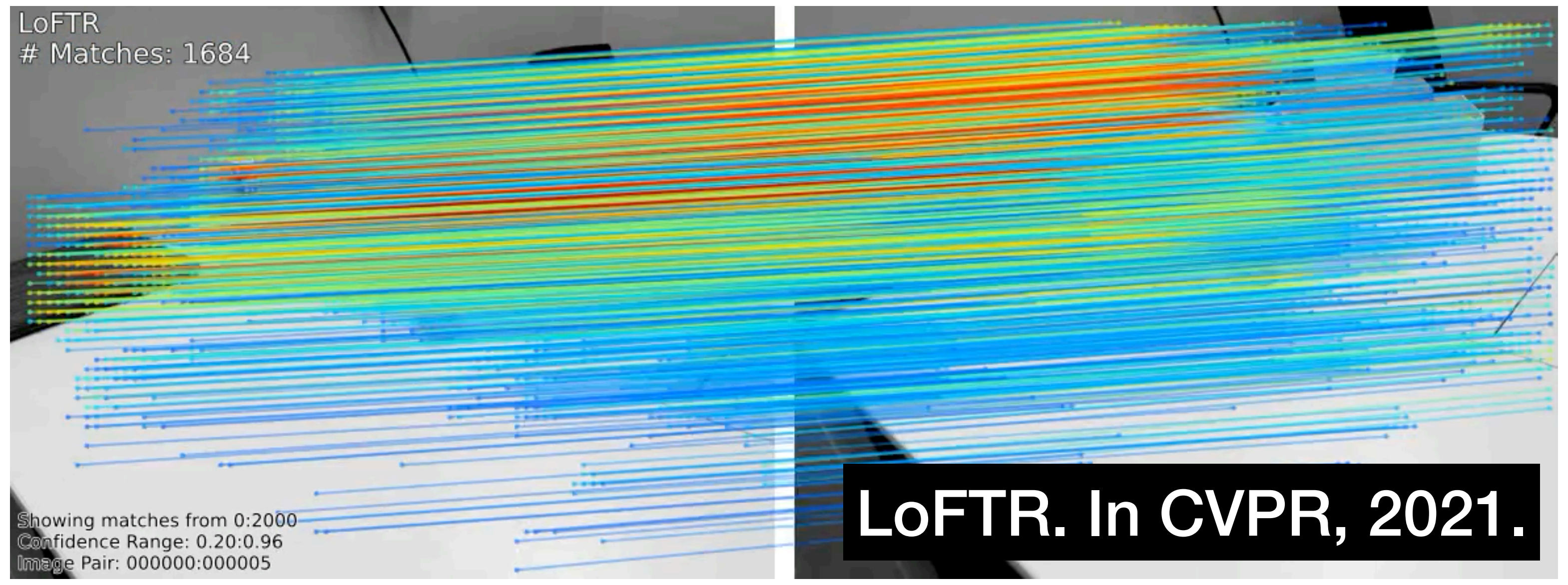
1. Efficiency.

2. Precision.

3. Robustness to camera setups (number, positions, etc.)

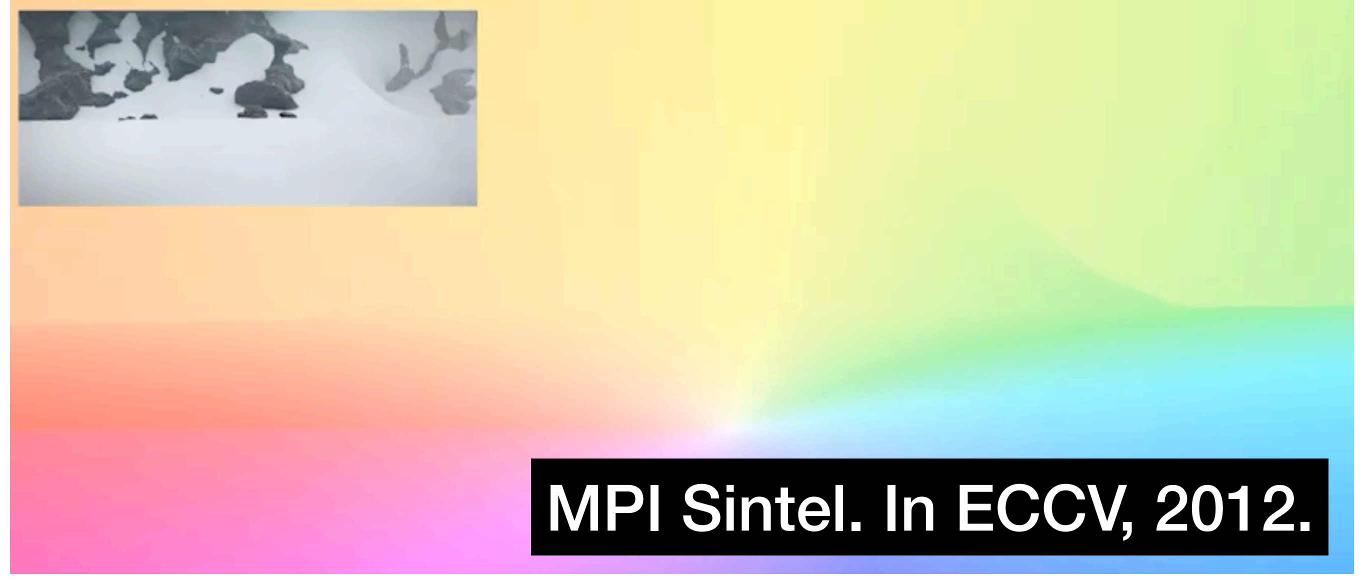
Overview of related problems

Point correspondences in 2D



LoFTR. In CVPR, 2021.

Feature matching.

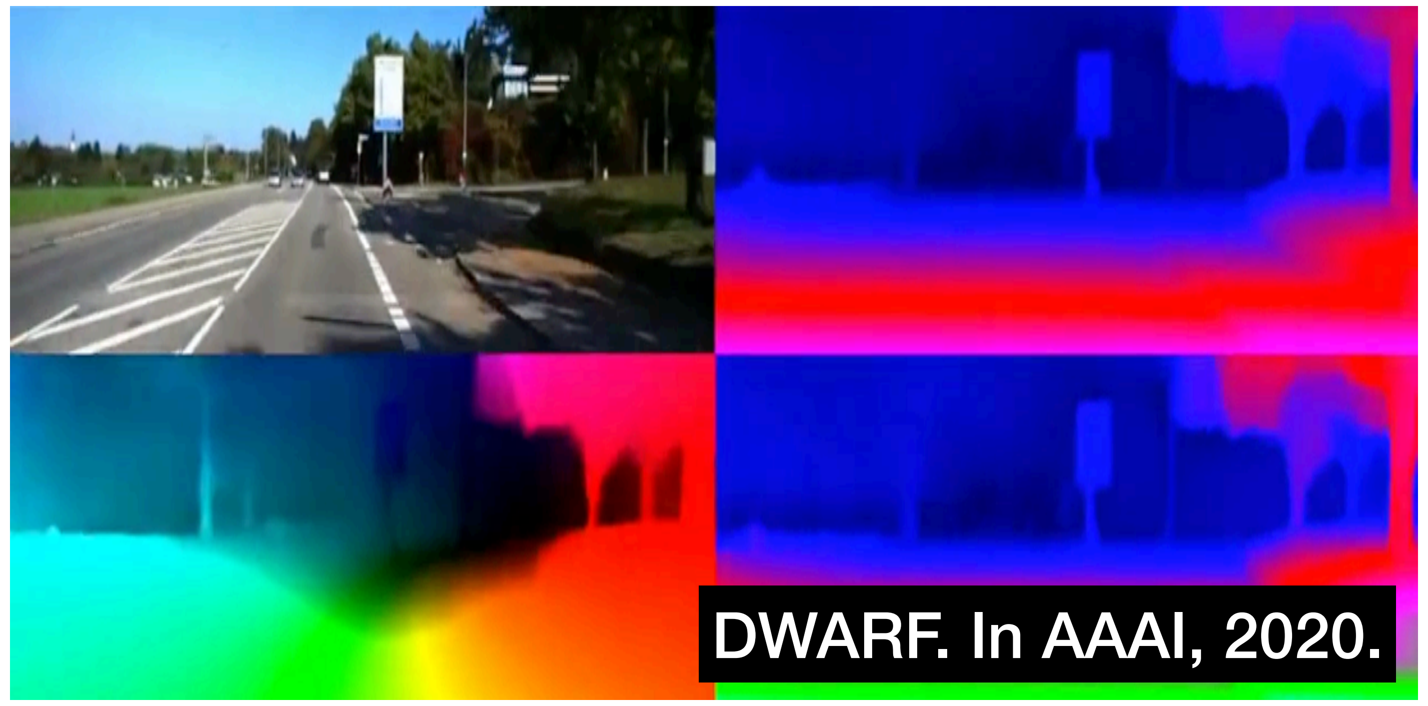

MPI Sintel. In ECCV, 2012.

Optical flow.

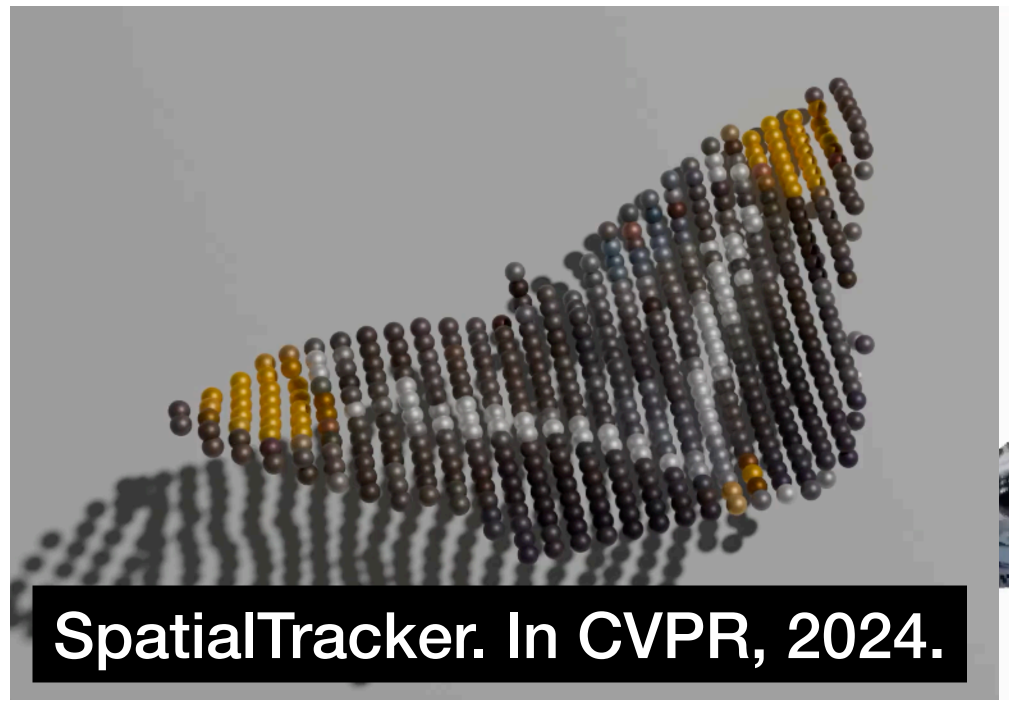

AllTracker. In ICCV, 2025.

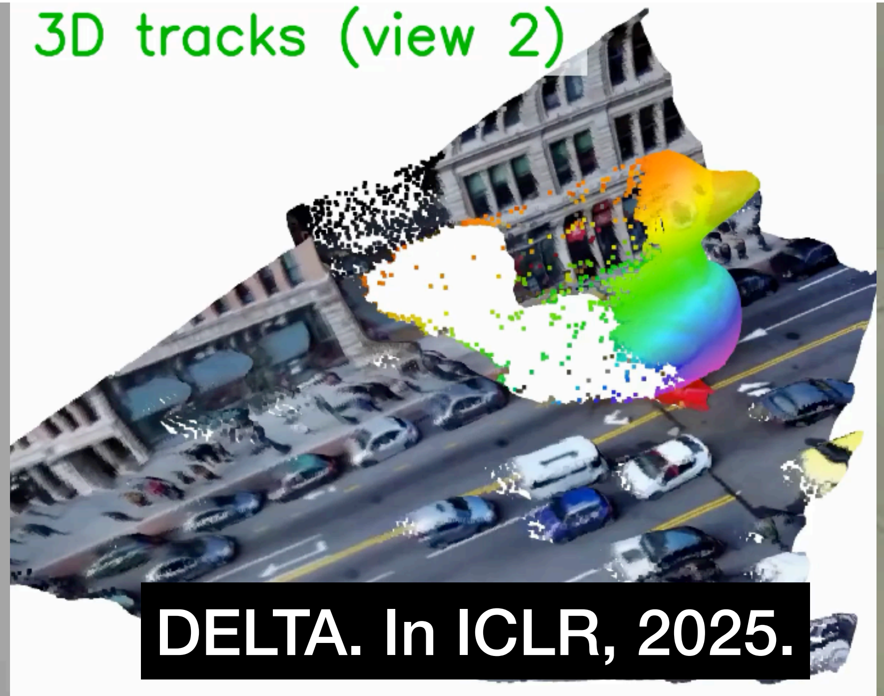
Point tracking in 2D.

Point correspondences in 3D


DWARF. In AAAI, 2020.

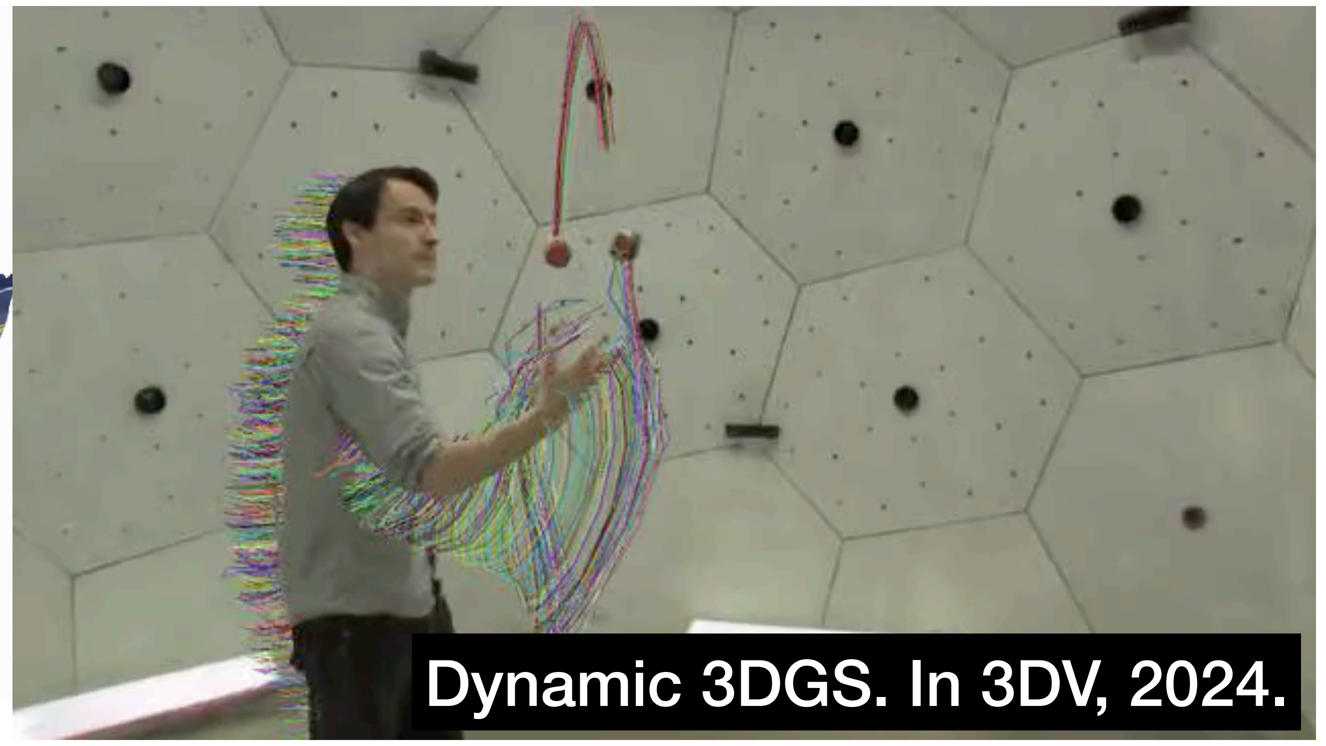
Scene flow.


SpatialTracker. In CVPR, 2024.



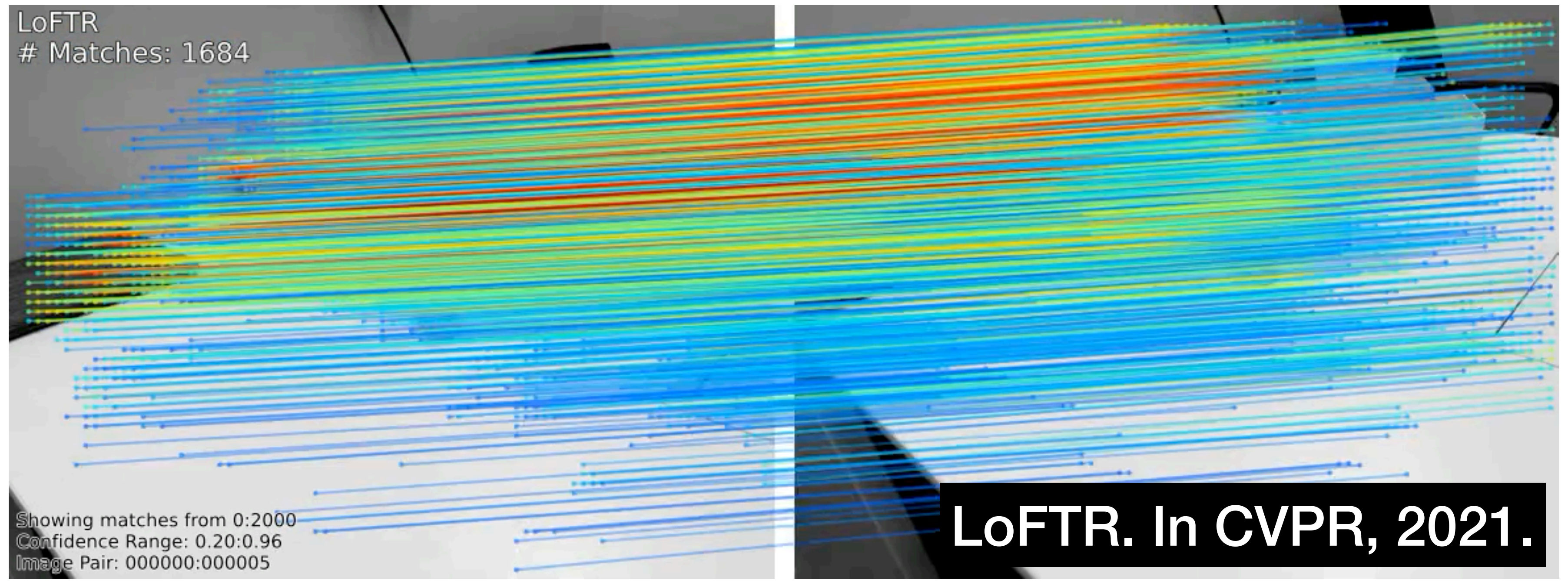
DELTA. In ICLR, 2025.

<u>Monocular</u> 3D point tracking.


Dynamic 3DGS. In 3DV, 2024.

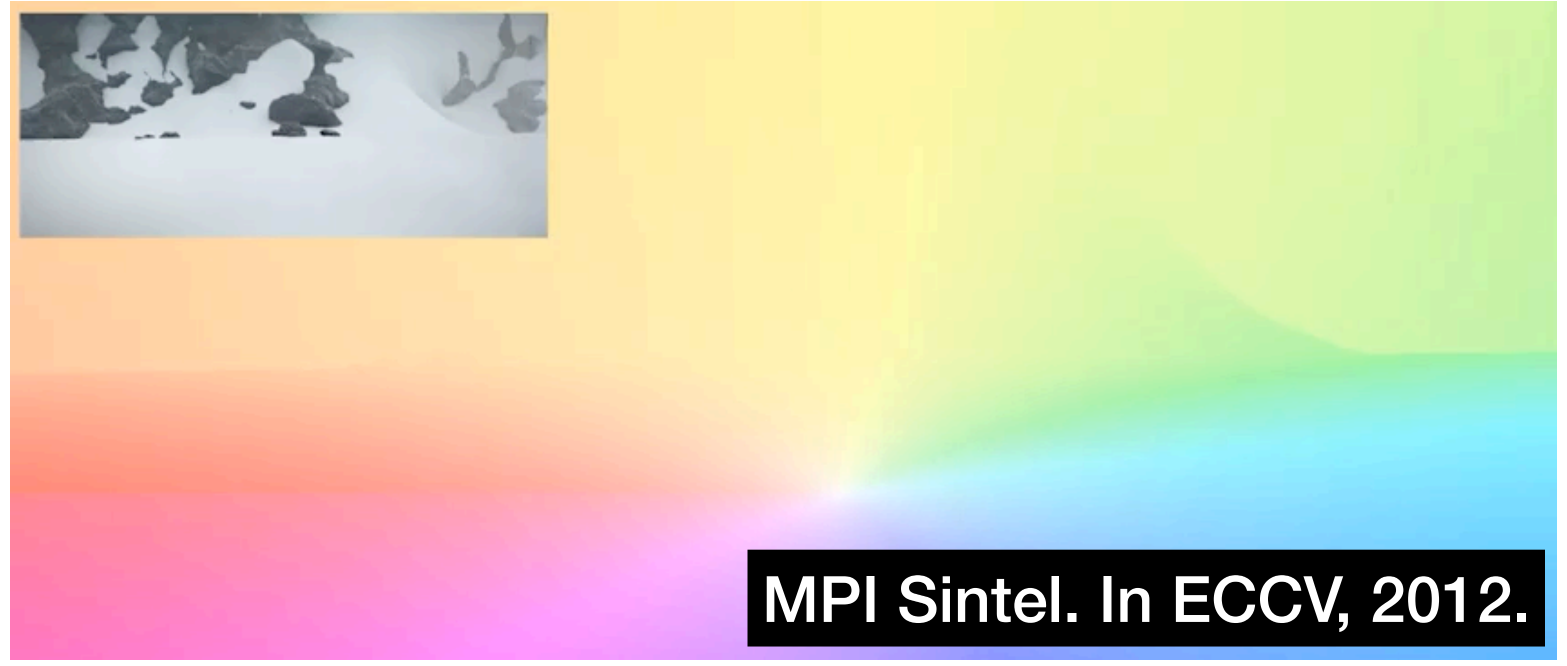
<u>Multi-view</u> 3D point tracking.

Point correspondences in 2D



LoFTR. In CVPR, 2021.

Feature matching: Tracking static points.

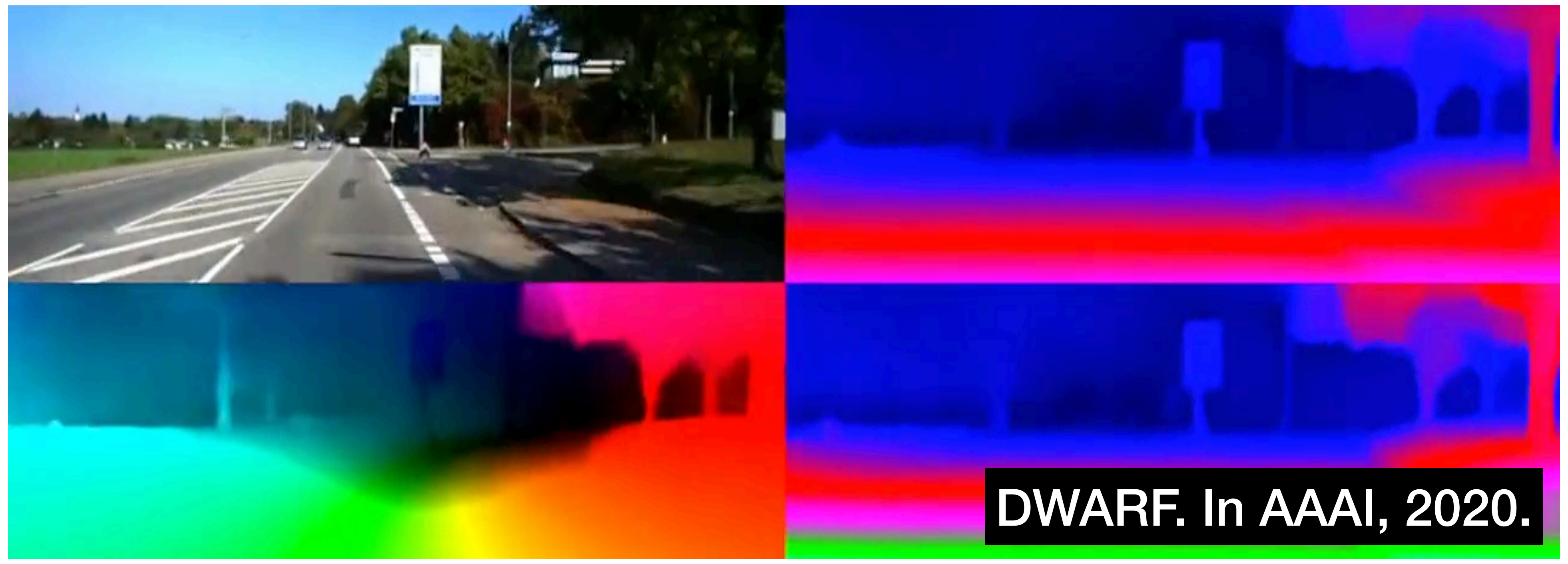

MPI Sintel. In ECCV, 2012.

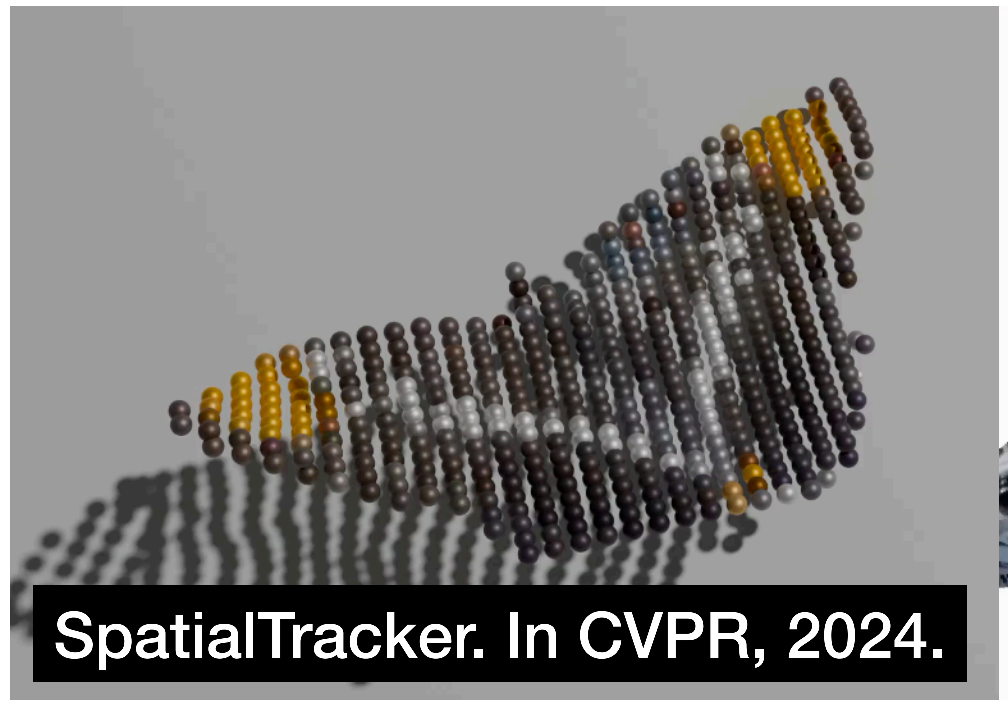
Optical flow: Dynamic, short-term.

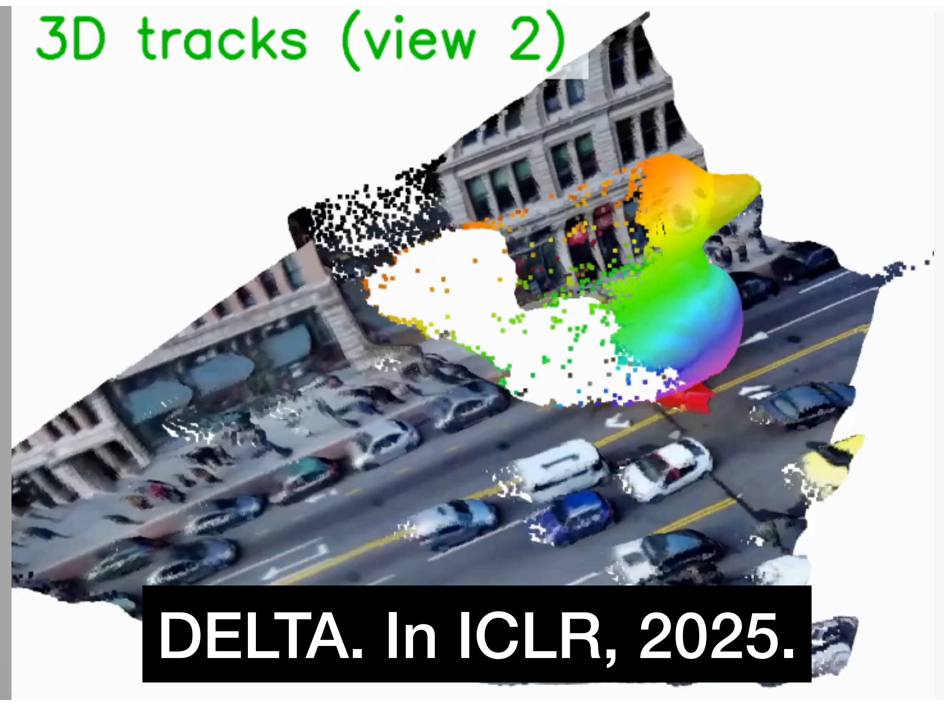

AllTracker. In ICCV, 2025.

Point tracking (2D): Dynamic points, long-term.

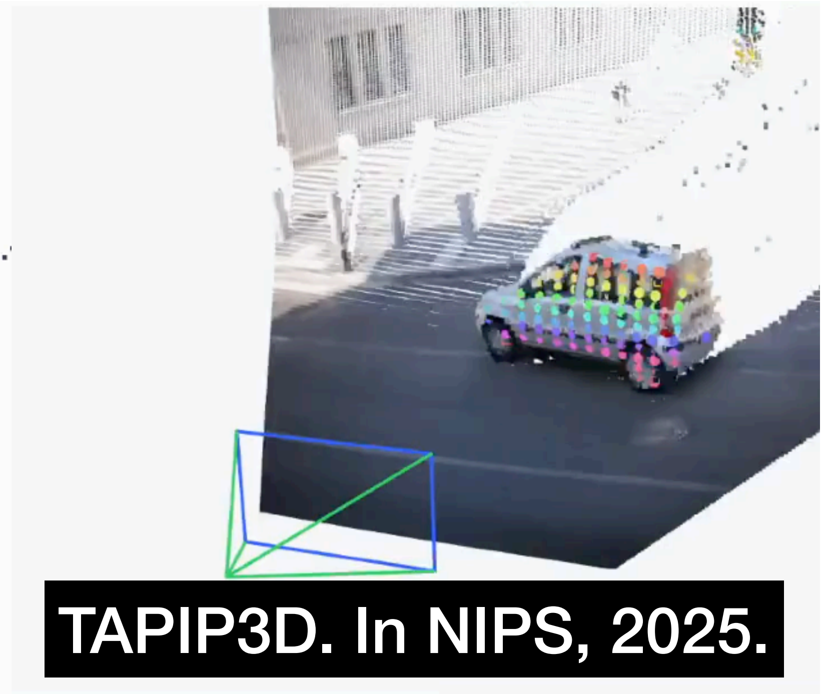
Point correspondences in 3D


DWARF. In AAAI, 2020.


SpatialTracker. In CVPR, 2024.



DELTA. In ICLR, 2025.


TAPIP3D. In NIPS, 2025.

Scene flow: Extension of optical flow to 3D.

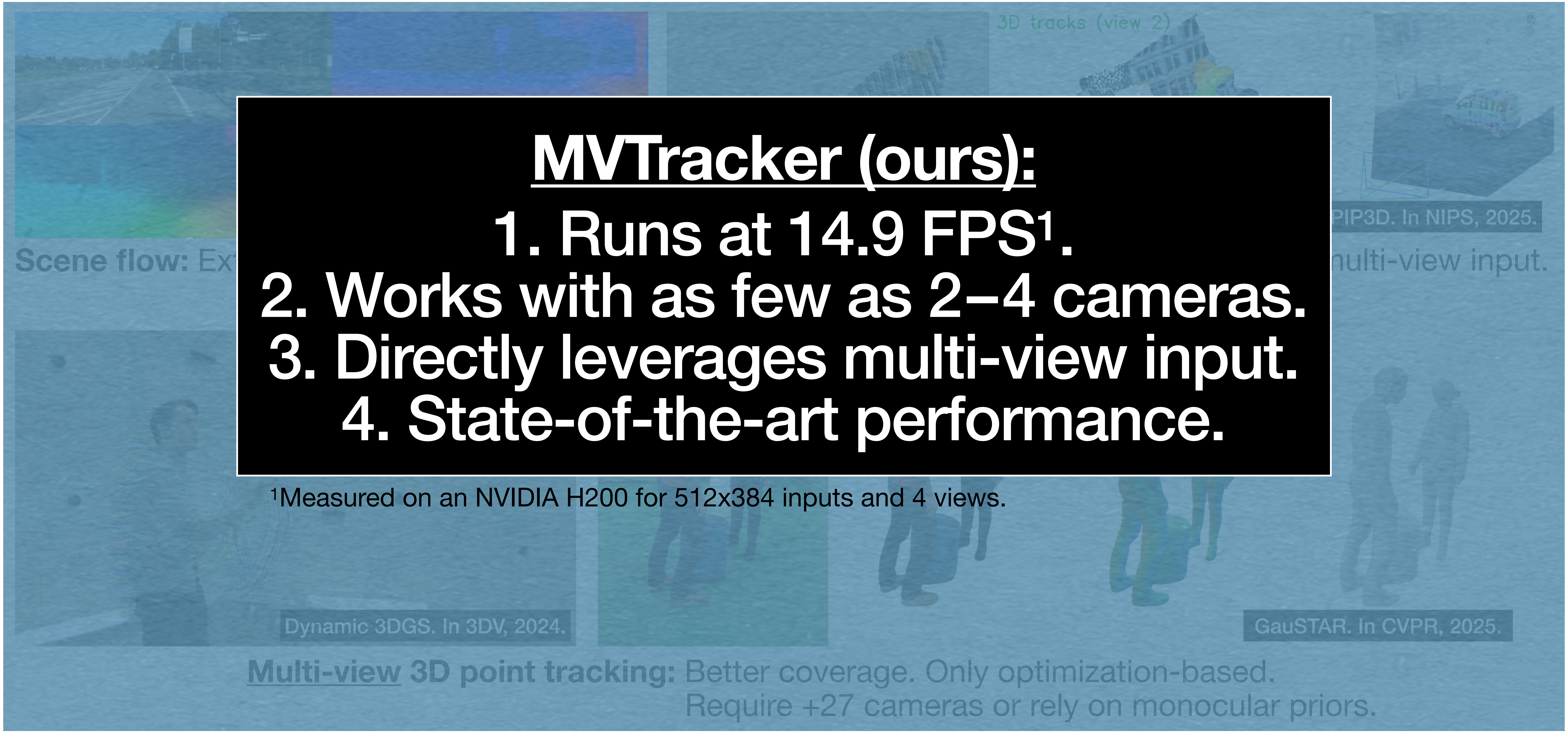
Monocular 3D point tracking: Can't leverage multi-view input.


Dynamic 3DGS. In 3DV, 2024.


GauSTAR. In CVPR, 2025.

Multi-view 3D point tracking: Better coverage. Only optimization-based.
Require +27 cameras or rely on monocular priors.

Point correspondences in 3D



MVTracker (ours):

1. Runs at 14.9 FPS[1].
2. Works with as few as 2–4 cameras.
3. Directly leverages multi-view input.
4. State-of-the-art performance.

[1]Measured on an NVIDIA H200 for 512x384 inputs and 4 views.

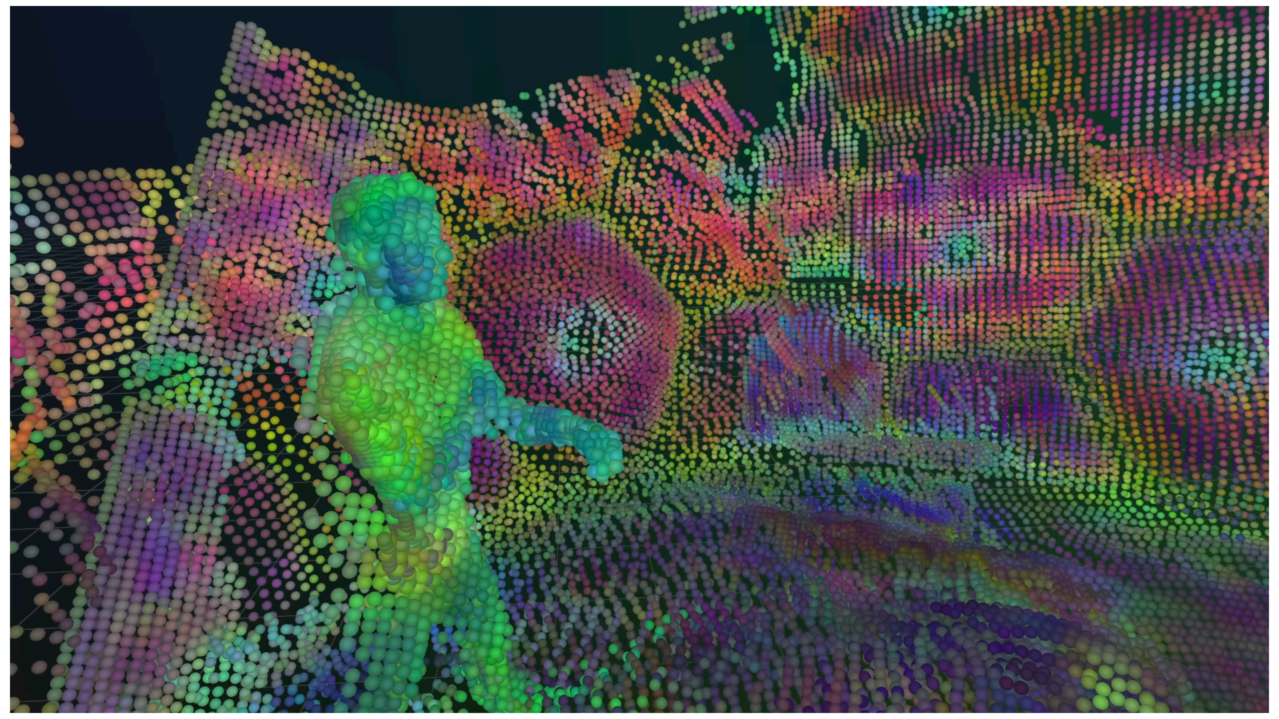
Fusing multi-view features into a point cloud

Colors visualize top 3 PCA components of the features.

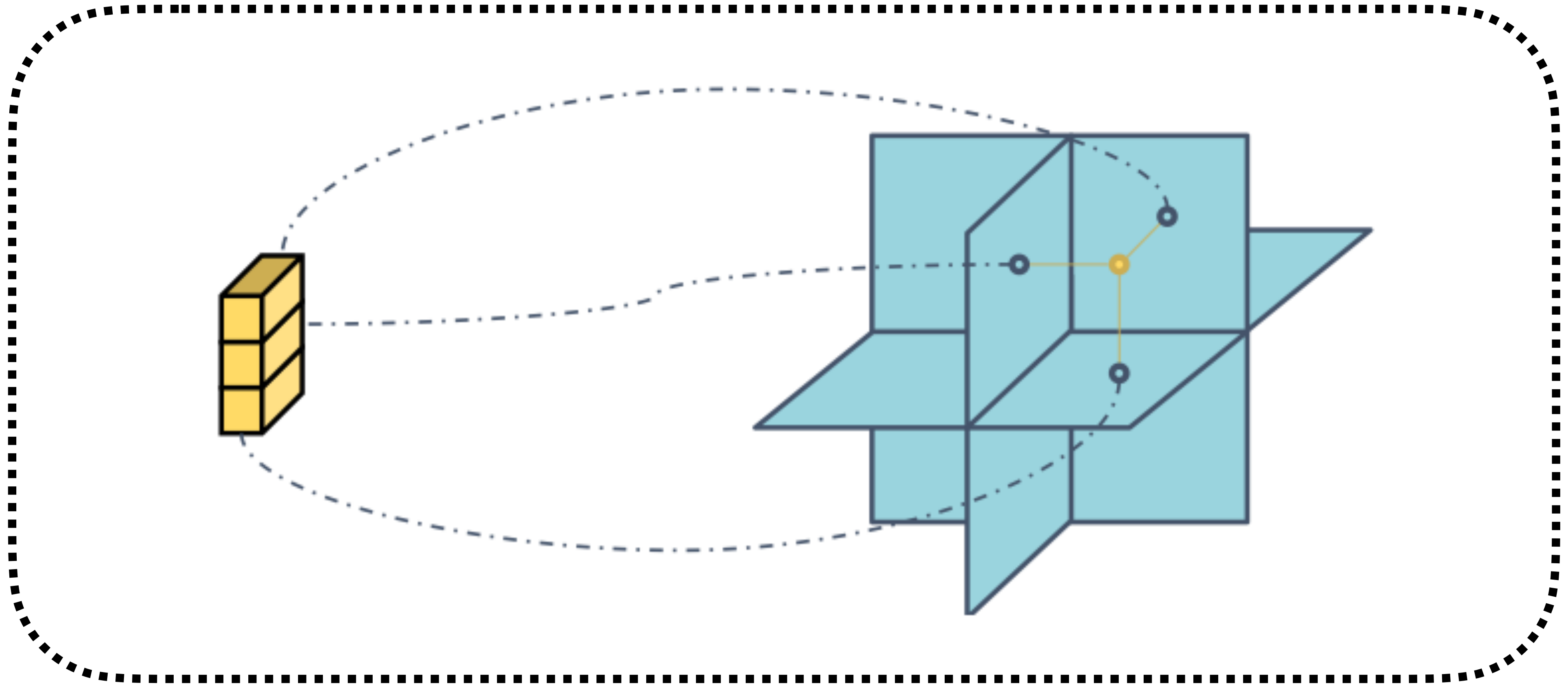
Fusing multi-view features into a point cloud

MVTracker (ours):

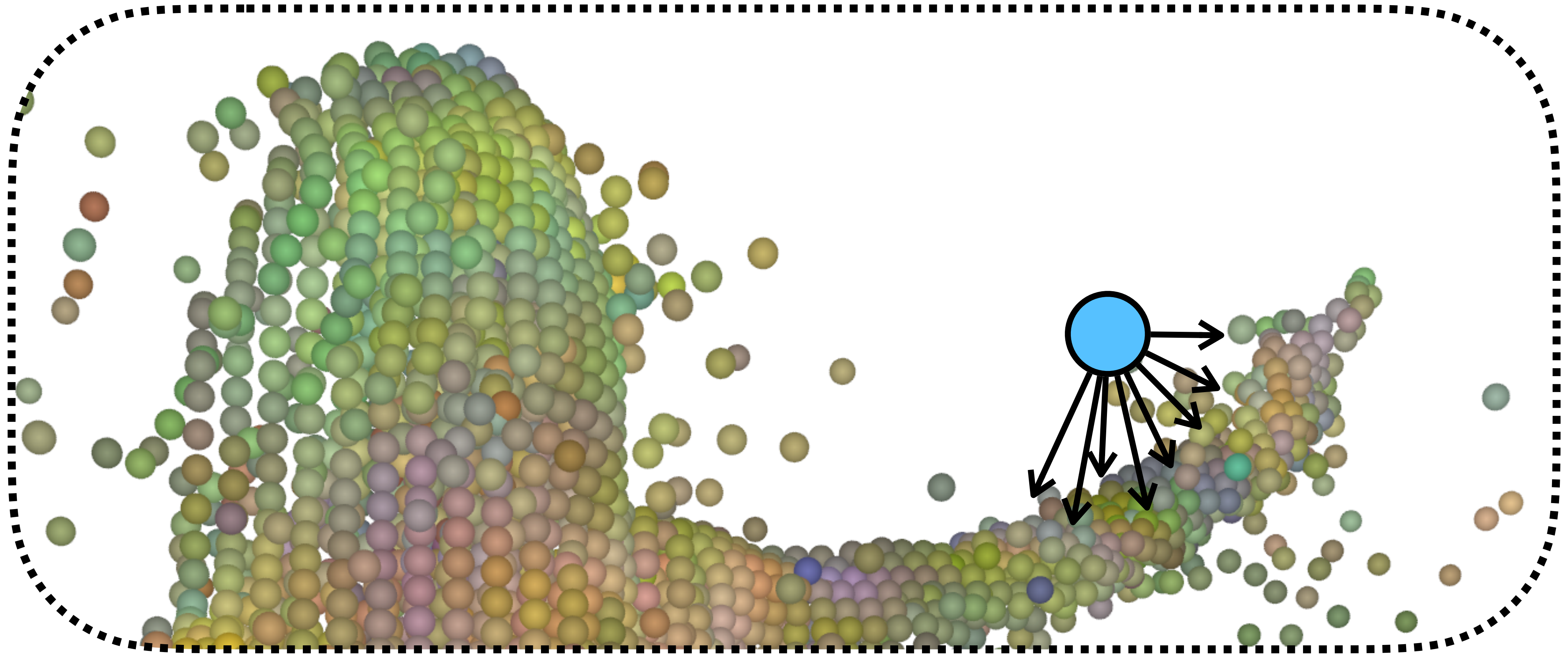
- **Data-driven** prior using a transformer.
- Contribute train. and eval. **datasets**.
- **Flexible** to a different number of cameras, their arrangements, and the depth source.



Fused 3D feature point cloud.



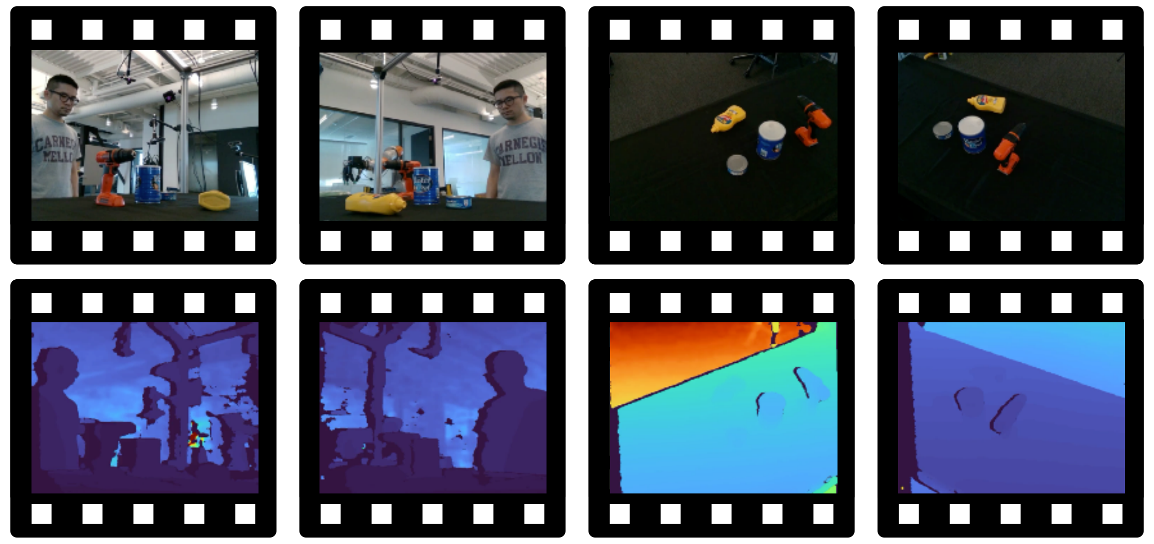
SpaTracker (CVPR'24): Triplane correlation.
Overlap and compression.→Information loss.



Ours: Efficient kNN correlation in point cloud.

Multi-view 3D point tracking

4 Kinect Cameras
(RGB + Depth)

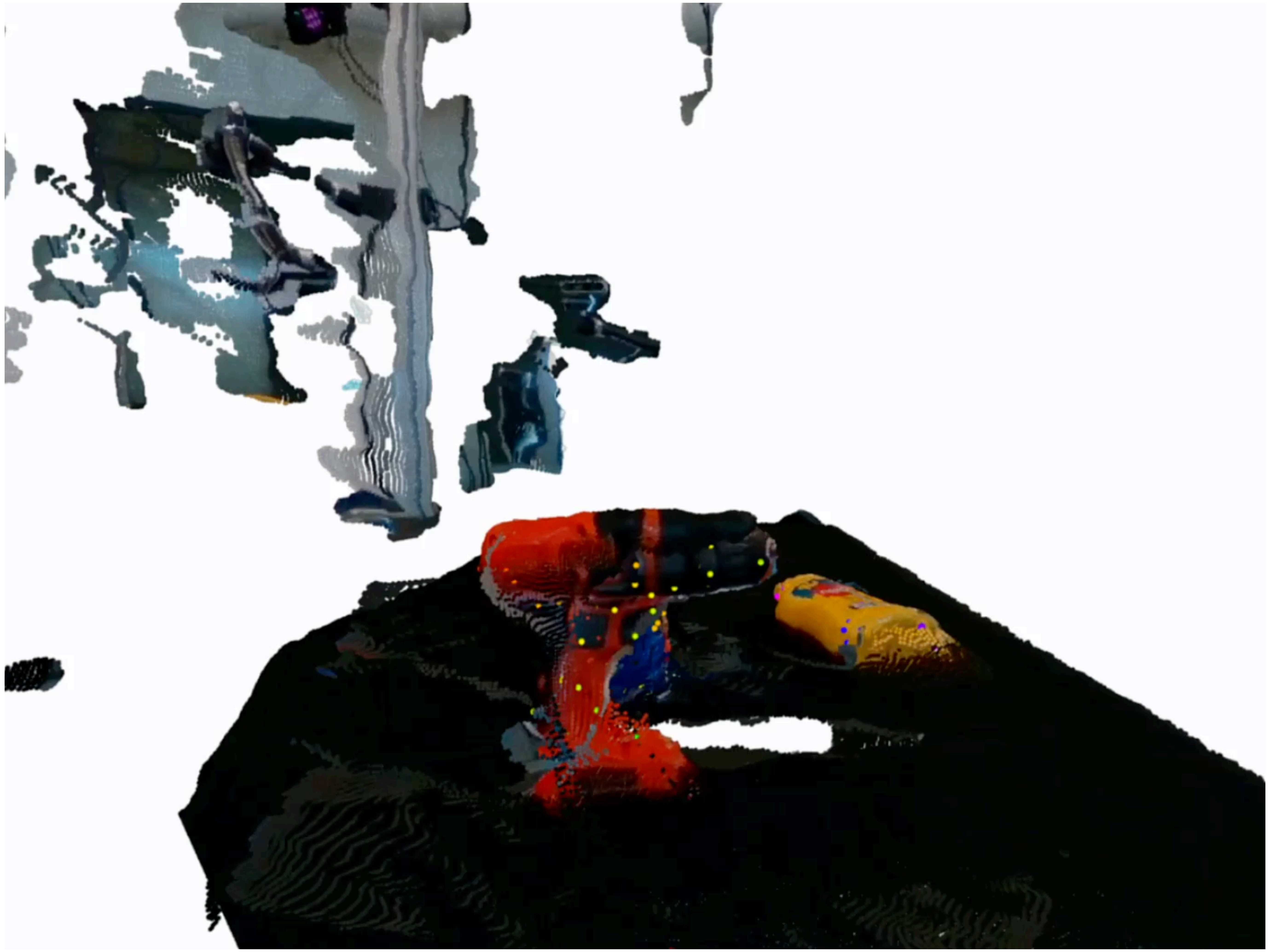


Dataset: DexYCB

+

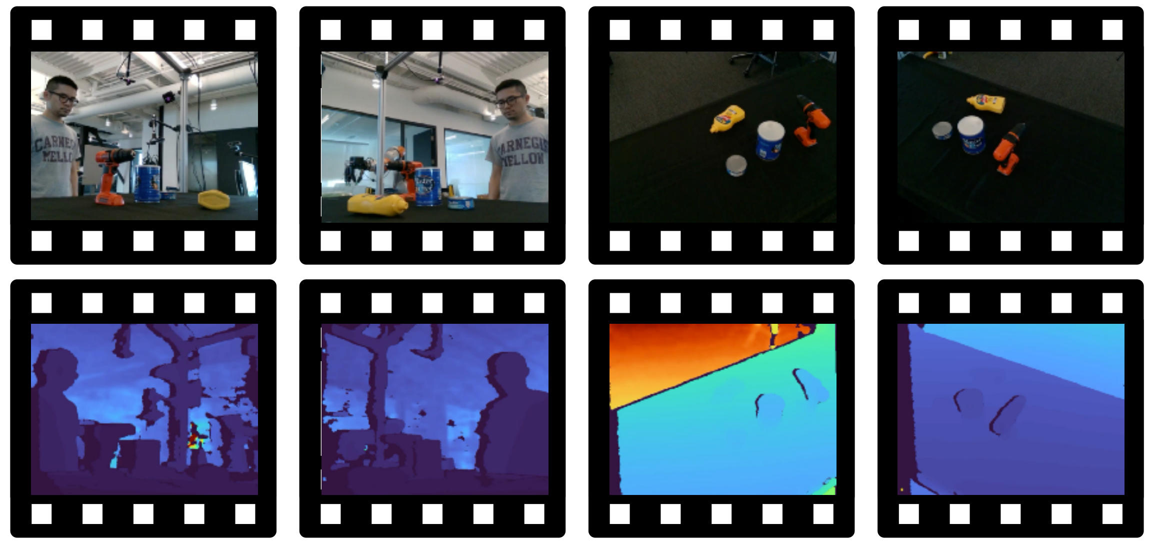
Camera Poses (calibrated)

MVTracker



Multi-view 3D point tracking

4 Kinect Cameras
(RGB + Depth)

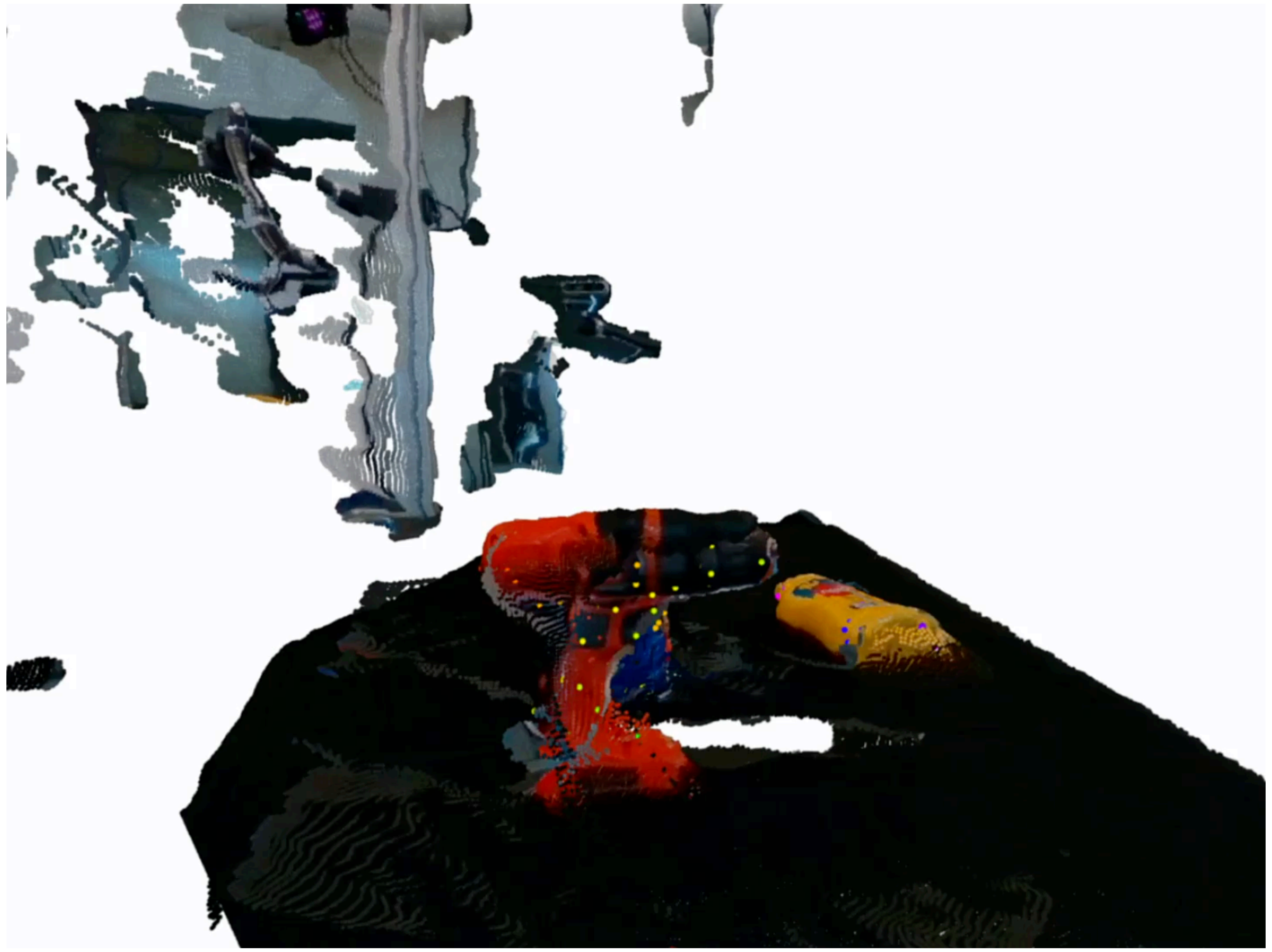


Dataset: DexYCB

+

Camera Poses (calibrated)

MVTracker



Inputs:
- Multi-view RGB: (V, T, H×W×3).
- Depth maps: (V, T, H×W×1).
- Intrinsics: (V, T, 3, 3).
- Extrinsics: (V, T, 3, 4).
- 3D query points: (N, 4) as txyz.

Outputs:
- Predicted tracks: (N, T, 3).
- Predicted visibility: (N, T, 1).

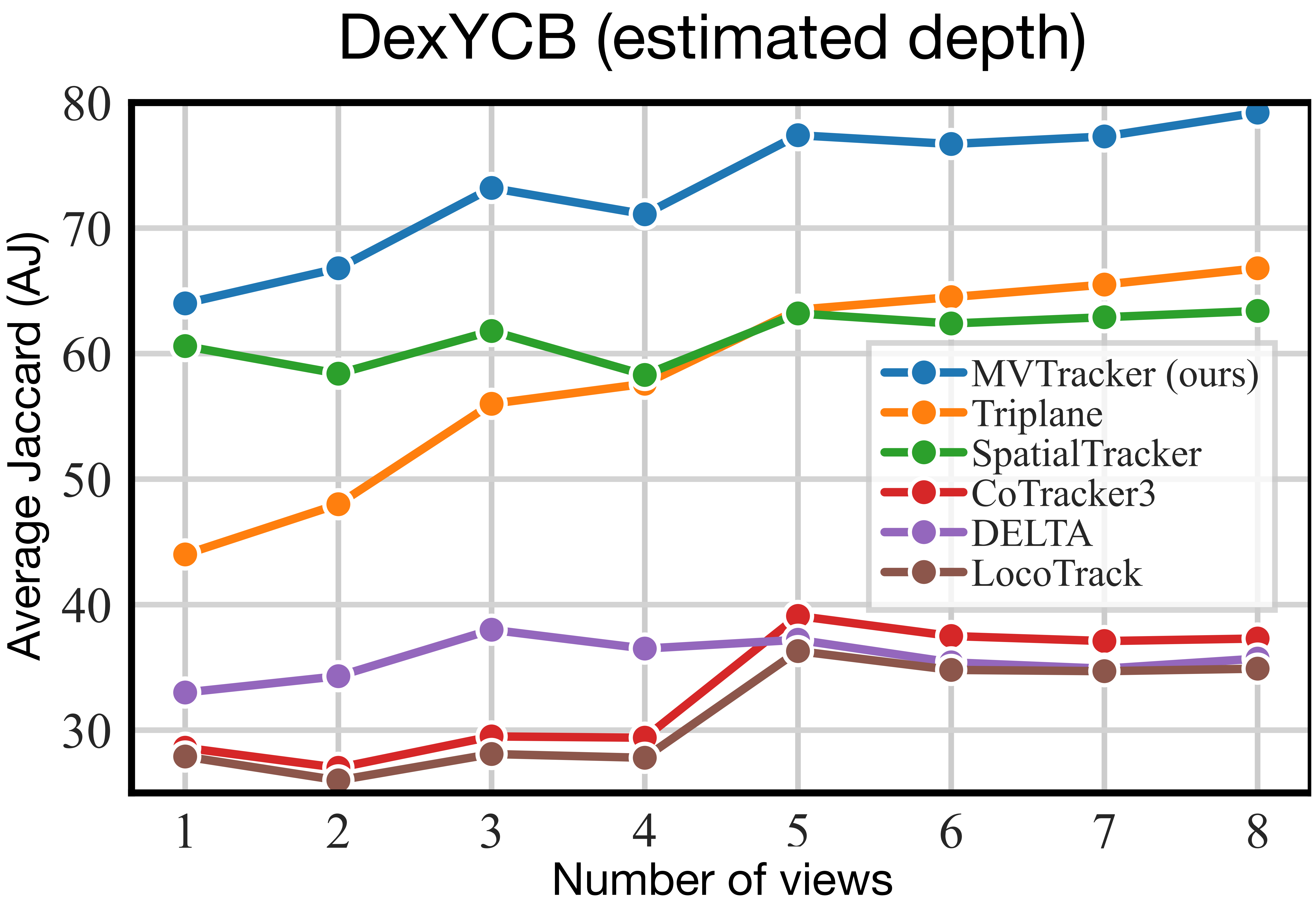
Notation: V views; T frames; N tracks.

Fused 3D feature point cloud + transformer

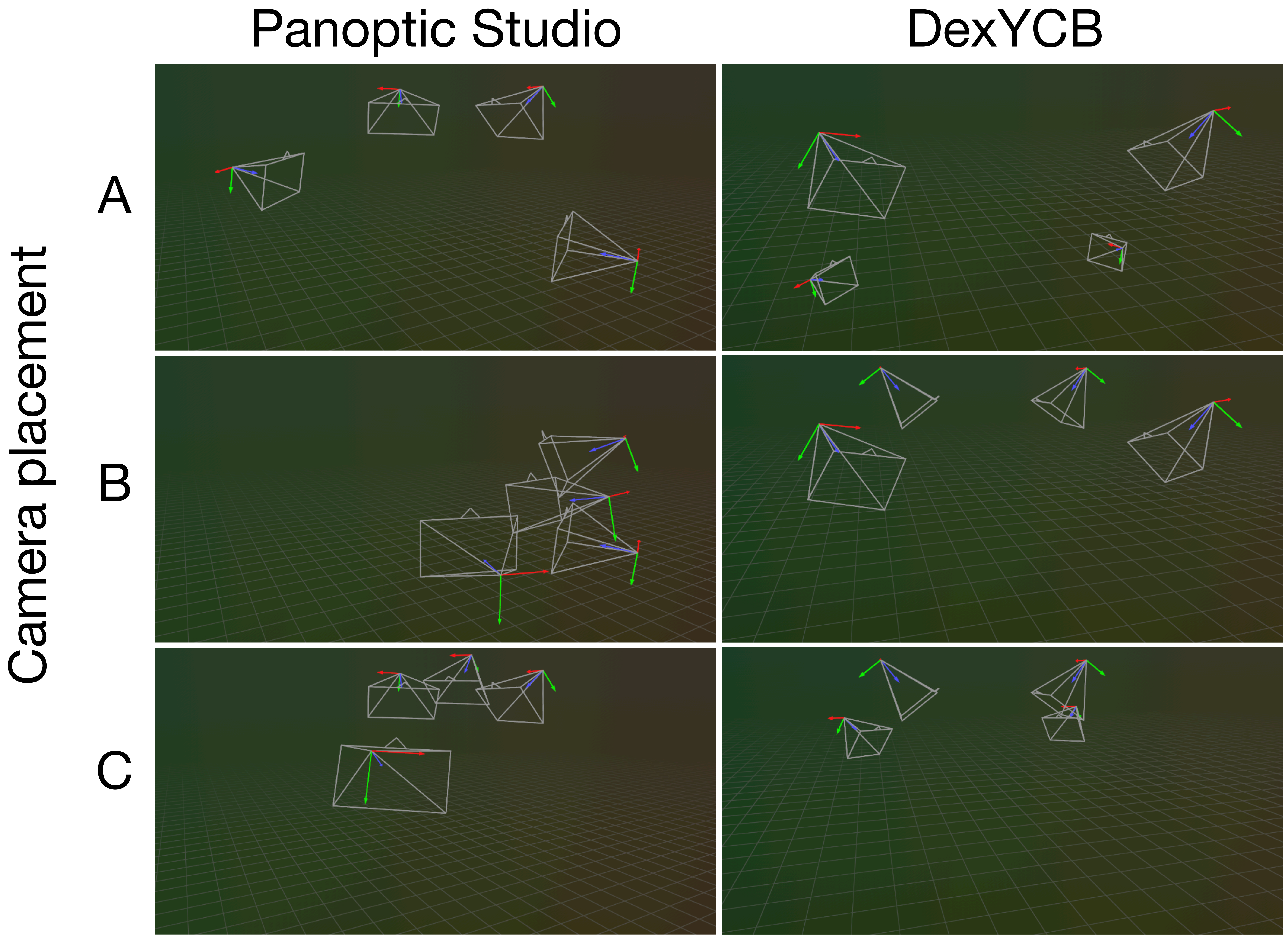
Main comparison (4 cameras)

*concurrent work

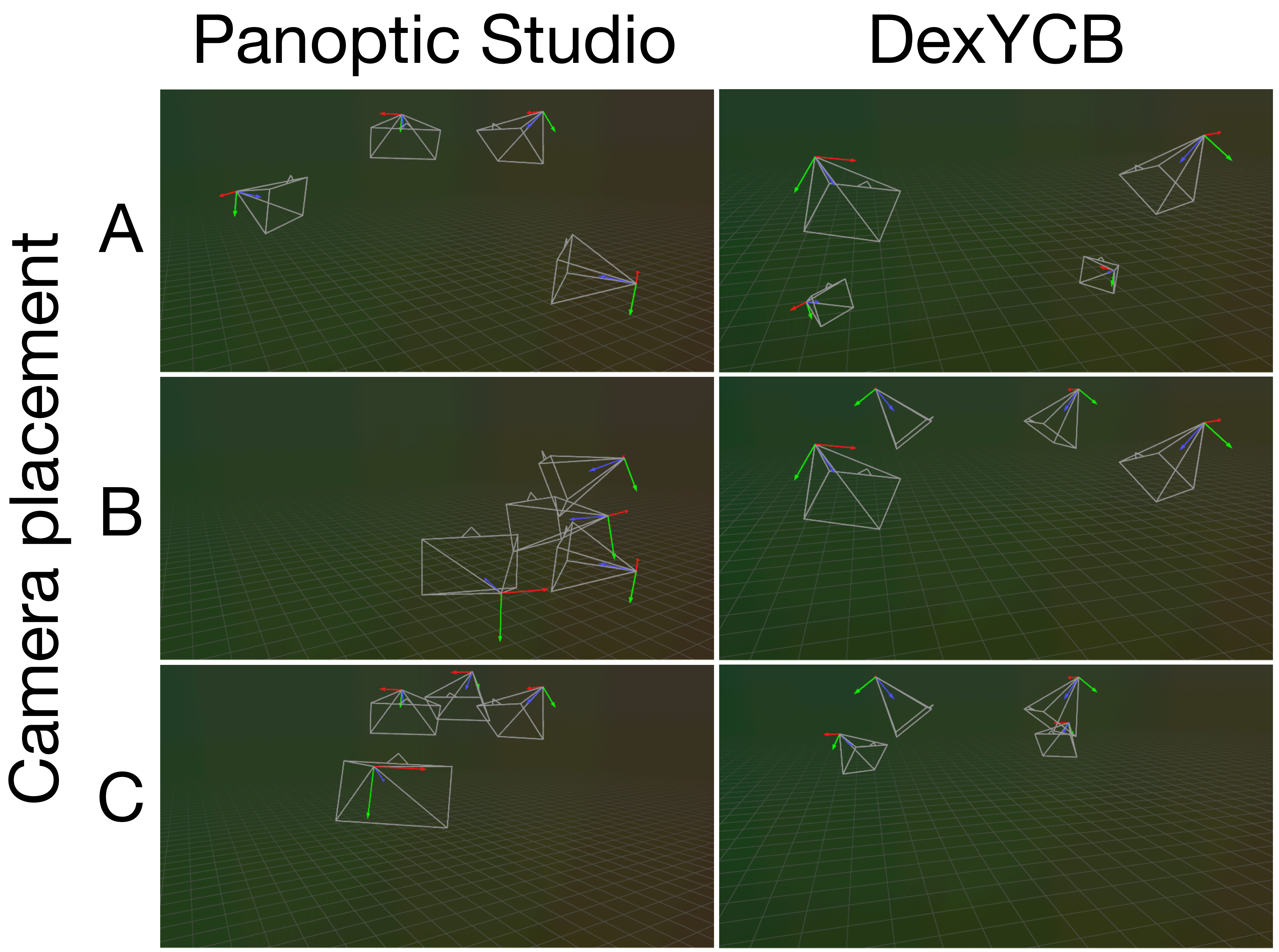
Accuracy improves with more cameras



Stable results across camera placements



Stable results across camera placements



Method	PStudio [18]			DexYCB [3]		
	A	B	C	A	B	C
Dynamic 3DGS [23]	66.5	50.8	56.6	45.7	–	–
Shape of Motion [35]	72.6	64.3	66.8	36.2	–	–
LocoTrack [5]	65.8	57.9	63.7	27.8	40.9	42.9
DELTA [24]	68.1	61.1	65.9	36.5	43.3	47.6
CoTracker2 [16]	69.5	62.3	66.4	28.8	42.0	44.4
CoTracker3 [17]	74.5	66.3	70.9	29.4	43.8	46.3
SpaTracker [38]	61.5	54.8	57.8	58.3	57.9	63.8
Triplane Baseline	65.1	59.9	63.5	57.5	62.0	66.3
MVTracker (ours)	**86.0**	**75.7**	**83.2**	**71.0**	**71.2**	**78.3**

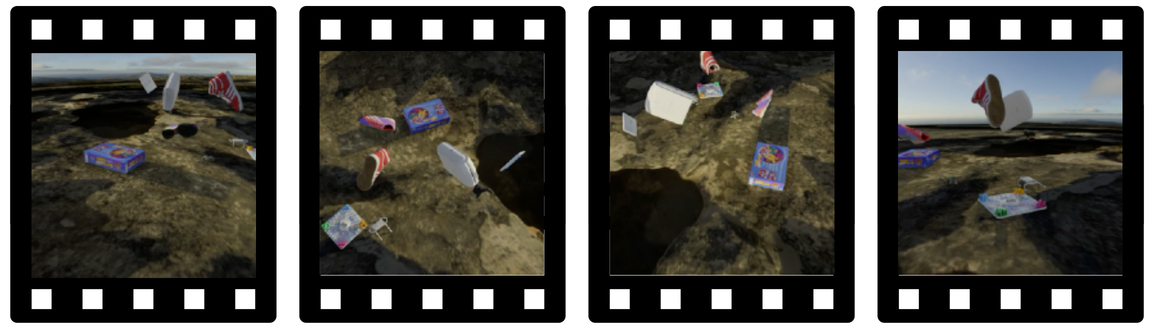
+15.4% (74.5 → 86.0, PStudio A)
+18.1% (66.3 → 78.3, DexYCB C)


Ground Truth
MVTracker (ours)
SpatialTrackerV1
Triplane Baseline
Red lines indicate distance to ground truth.

Ground Truth
MVTracker (ours)
SpatialTrackerV1
Triplane Baseline
Red lines indicate distance to ground truth.

Depth source:
Blender simulation

4 Cameras (synchronized)

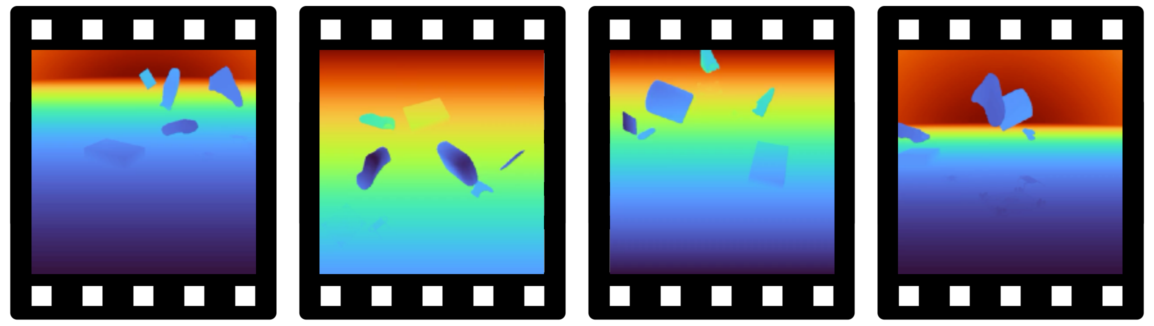


+

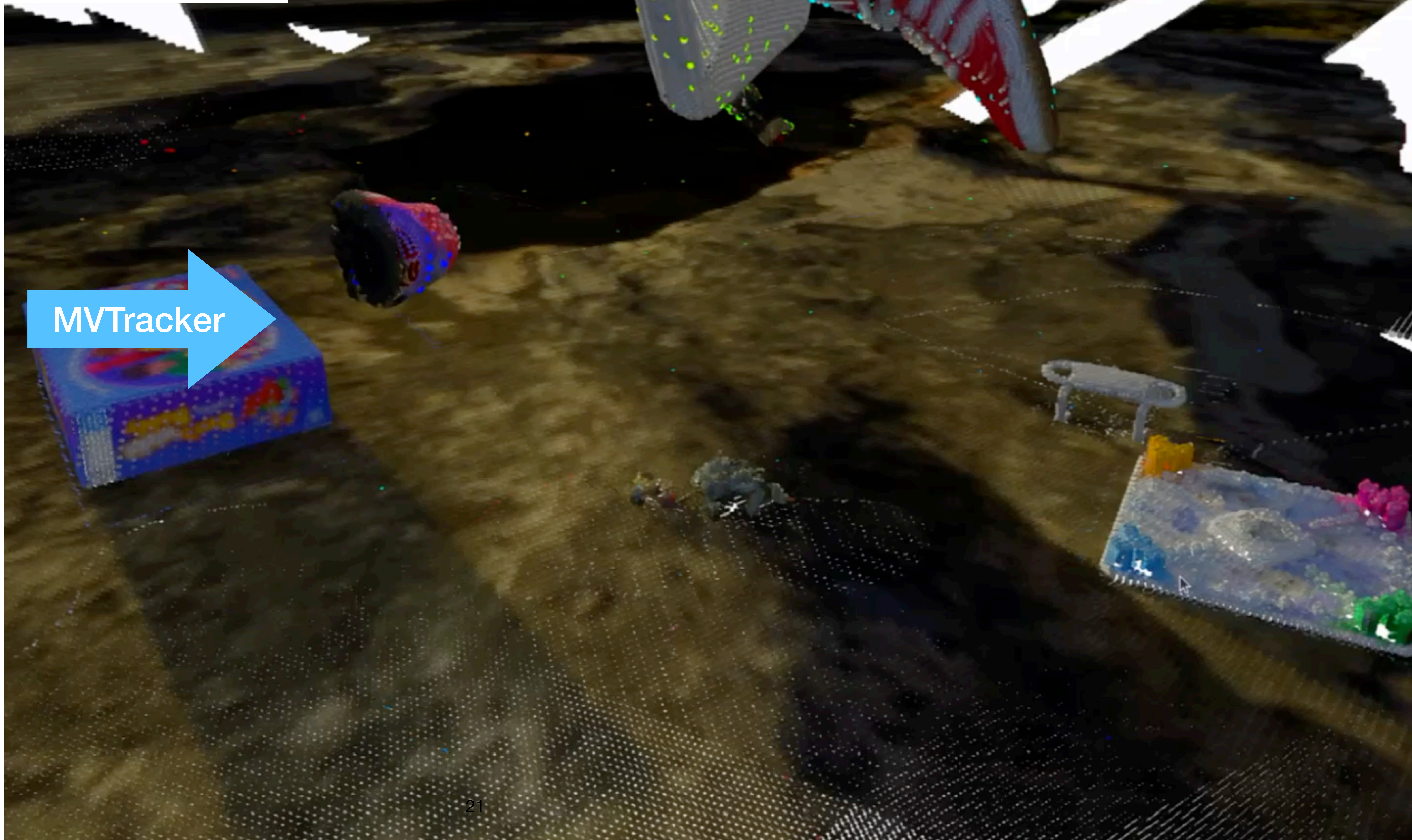
Camera Poses (calibrated)

+

Blender-simulated Depth

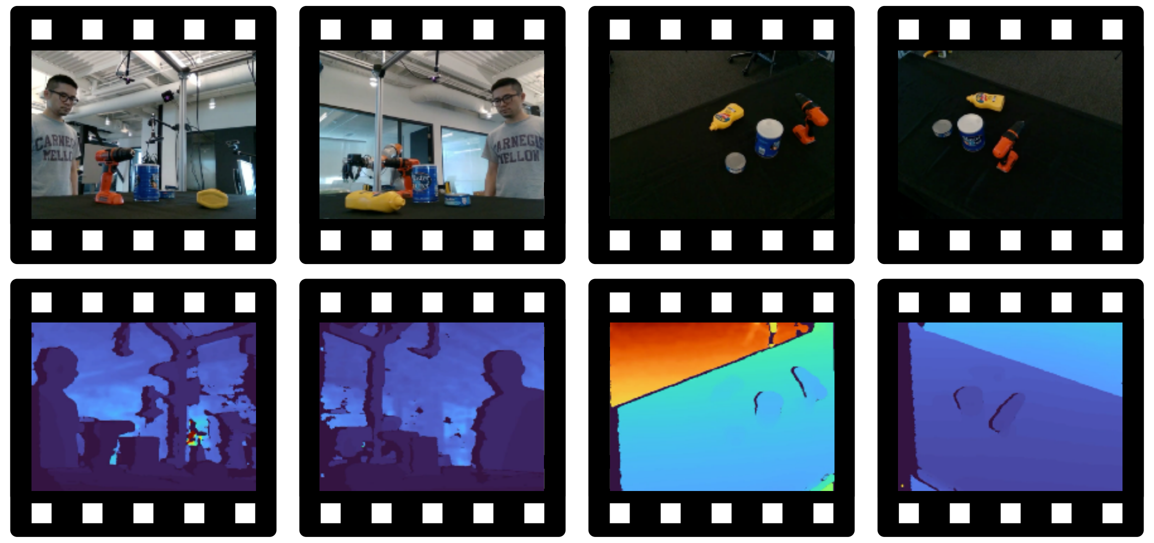


Dataset: MV-Kubric



Depth source:
Kinect camera

4 Kinect Cameras
(RGB + Depth)



+

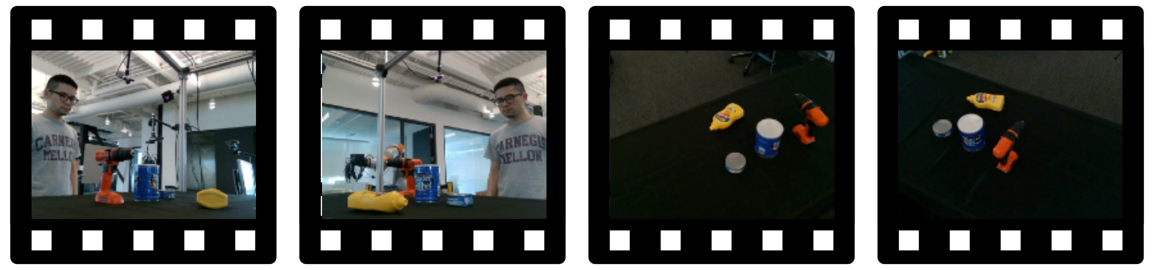
Camera Poses (calibrated)

MVTracker

Dataset: DexYCB

Depth source:
DUSt3R[1] estimates

4 Cameras (synchronized)

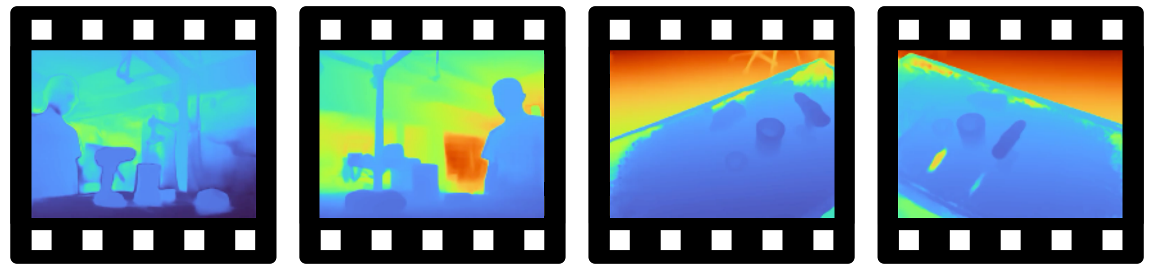


+

Camera Poses (calibrated)

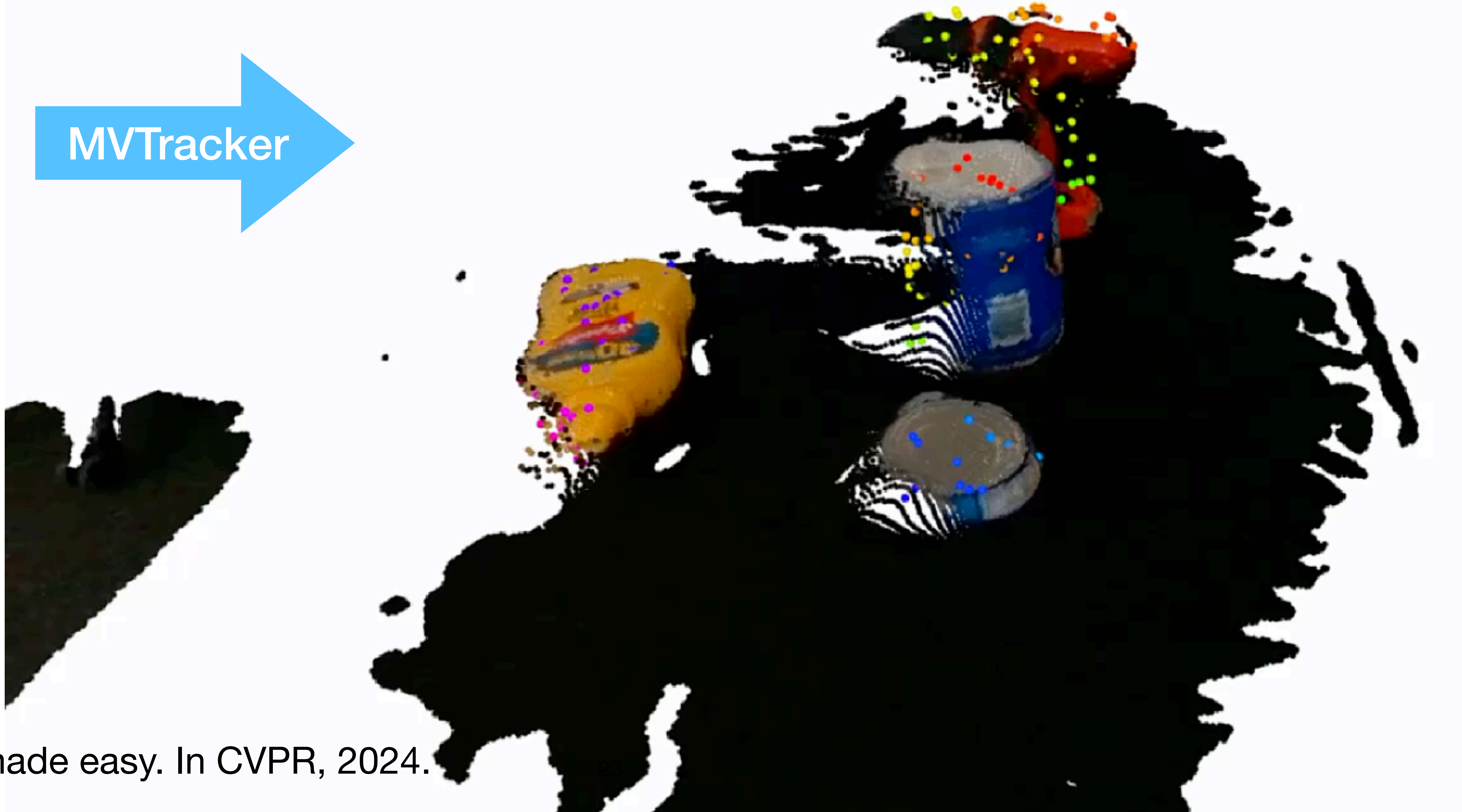
↓

DUSt3R[1]-estimated Depth



Dataset: DexYCB

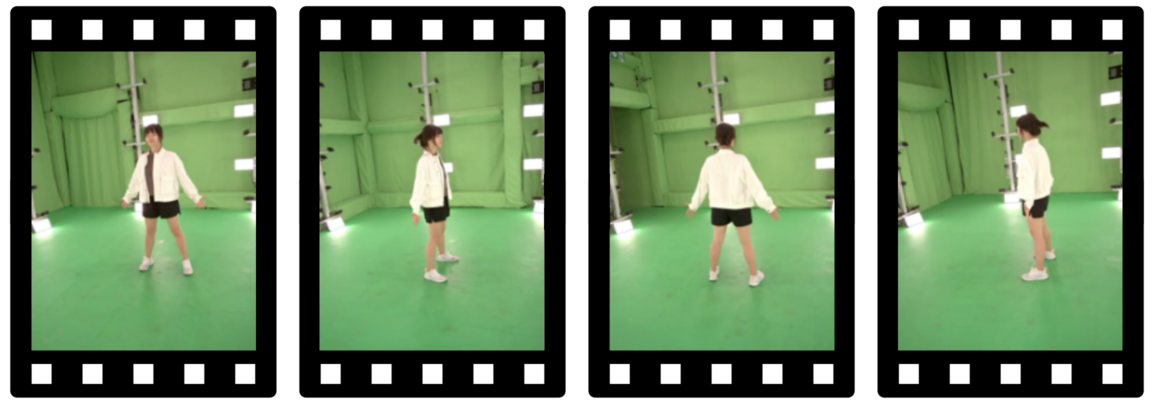
MVTracker →



[1]DUSt3R: Geometric 3D vision made easy. In CVPR, 2024.

Depth source:
Studio capture

4 Cameras (synchronized)

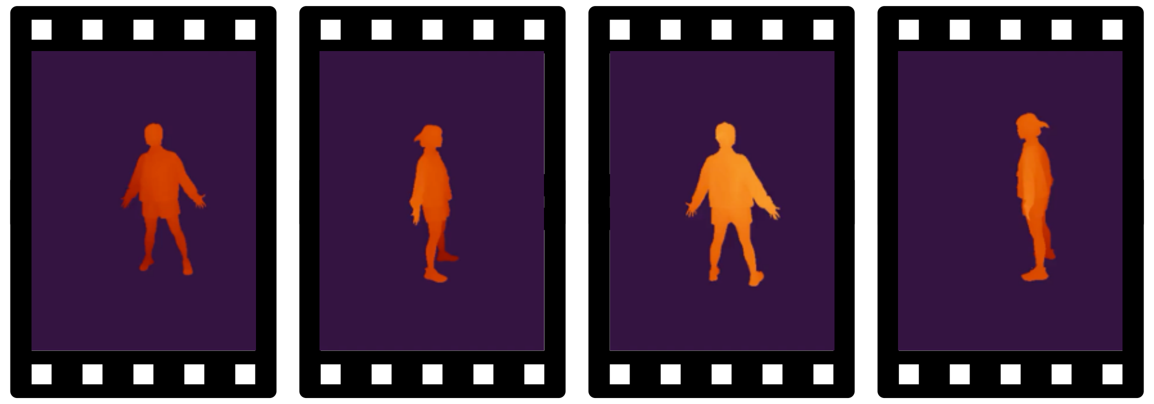


+

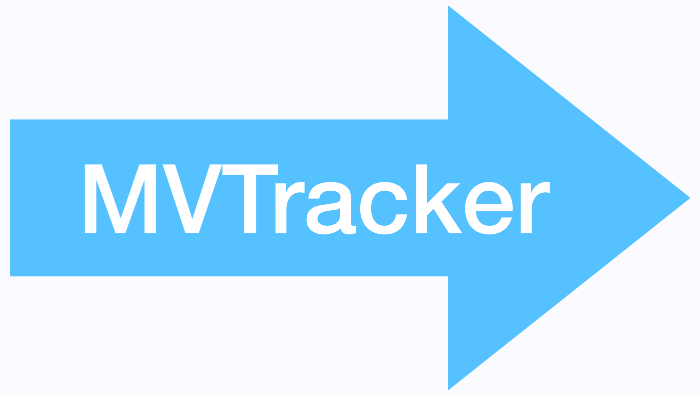
Camera Poses (calibrated)

+

Studio Capture Depth



Dataset: 4D-DRESS



Conclusion and key takeaways

MVTracker (ours):

- First **data-driven multi-view** point tracker.
- Fuses multi-view features into a **point cloud**.

- **Robust** to number of cameras, camera rigging, depth source and noise.
- **Significant gains** over monocular and multi-view baselines.

- **Main limitations:**
  - Dependence on multi-view **depth estimators**.
  - Tested only within **bounded scenes**.
  - We need **more data**. More is more.

Frano Rajič[1], Haofei Xu[1], Marko Mihajlović[1], Siyuan Li[1], Irem Demir[1], Emircan Gündoğdu[1], Lei Ke[2], Sergey Prokudin[1,3], Marc Pollefeys[1,4], Siyu Tang[1]

[1]ETH, [2]CMU, [3]Balgrist, [4]Microsoft

Frequently asked questions

1. Can MVTracker run **online**? Yes, at 14.9 FPS using sliding windows.
2. Can tracked points only be added **at the first frame**? No, at any frame.
3. What is the **scale of the training dataset**? 5000 sequences of MV-Kubric.
4. Can MVTracker be used with a **stereo camera** / small baseline? Yes, but this would amount to monocular 3D point tracking (after stereo depth estimation).
5. Does MVTracker regress **invisible point locations**?
   During training yes, but we didn't benchmark this.

https://ethz-vlg.github.io/mvtracker/