

Learning-based SLAM

Prof Tom Drummond

Using deep learning for SLAM

Want to know two things:

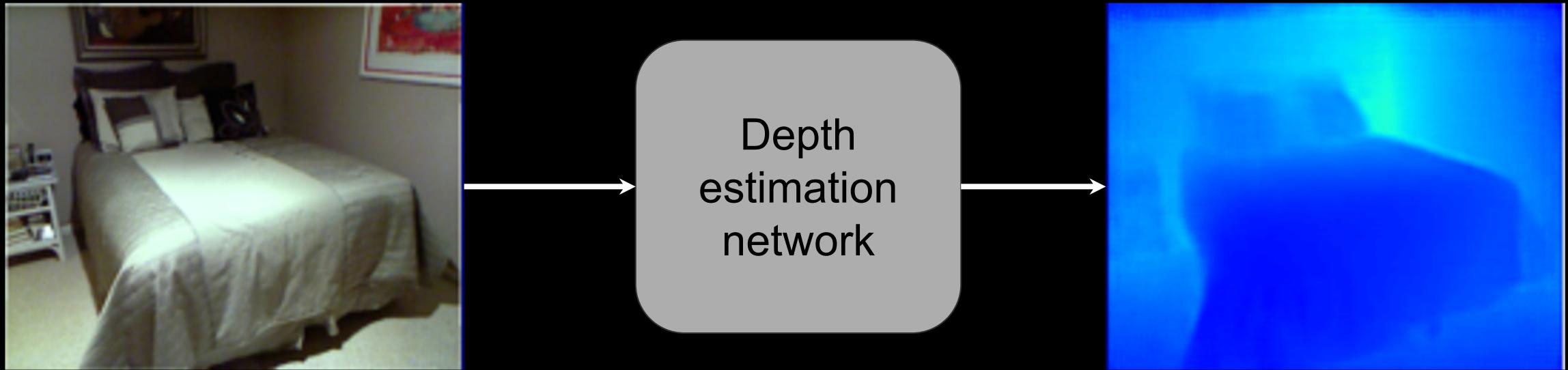
1. Mapping

- What shape is the world?

2. Localisation

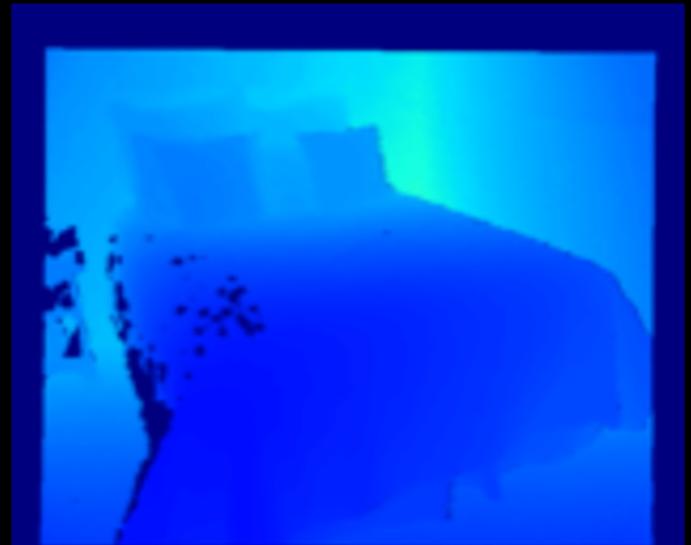
- Where are we?
- What is our trajectory through the world?

What shape is the world?



Minimise L_2 loss to ground truth.

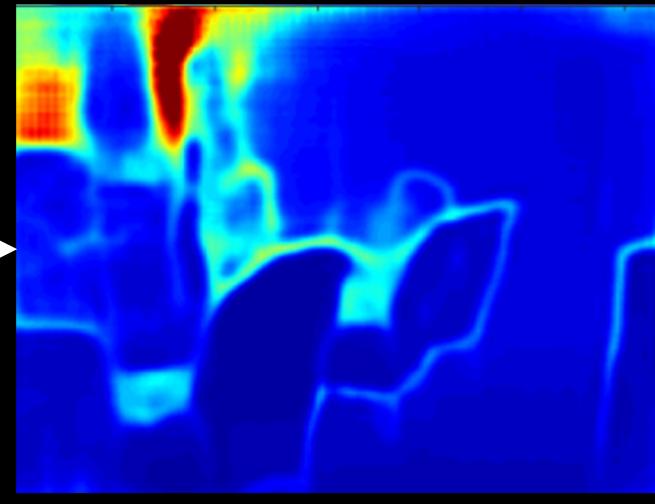
Q: Should we estimate z , $1/z$ or $\log(z)$?



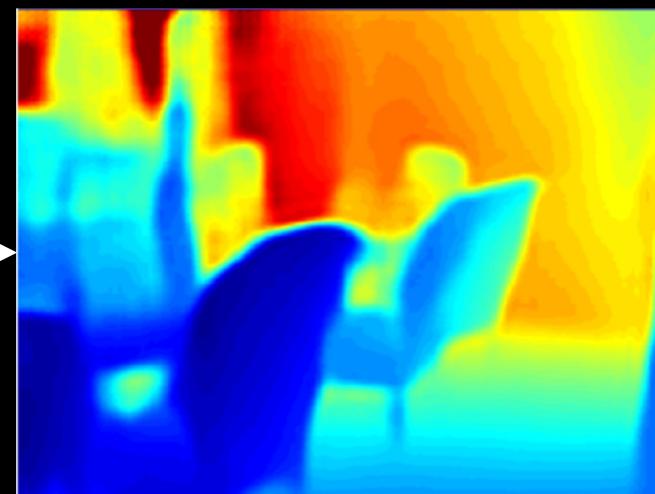
Better to learn uncertainty as well as depth estimates



Depth
estimation
network



Uncertainty (σ)

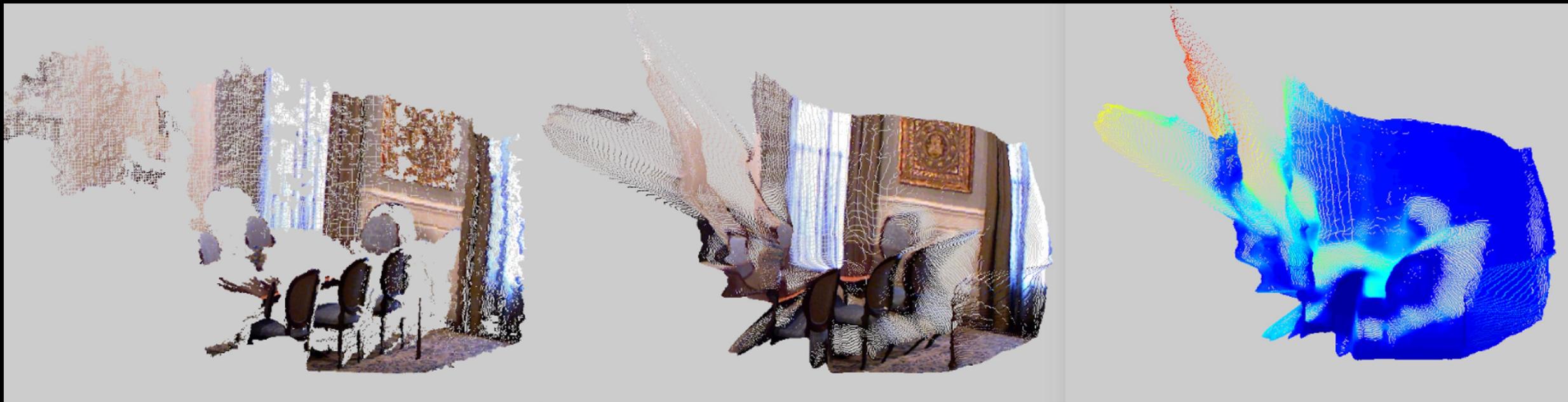


Depth (μ)

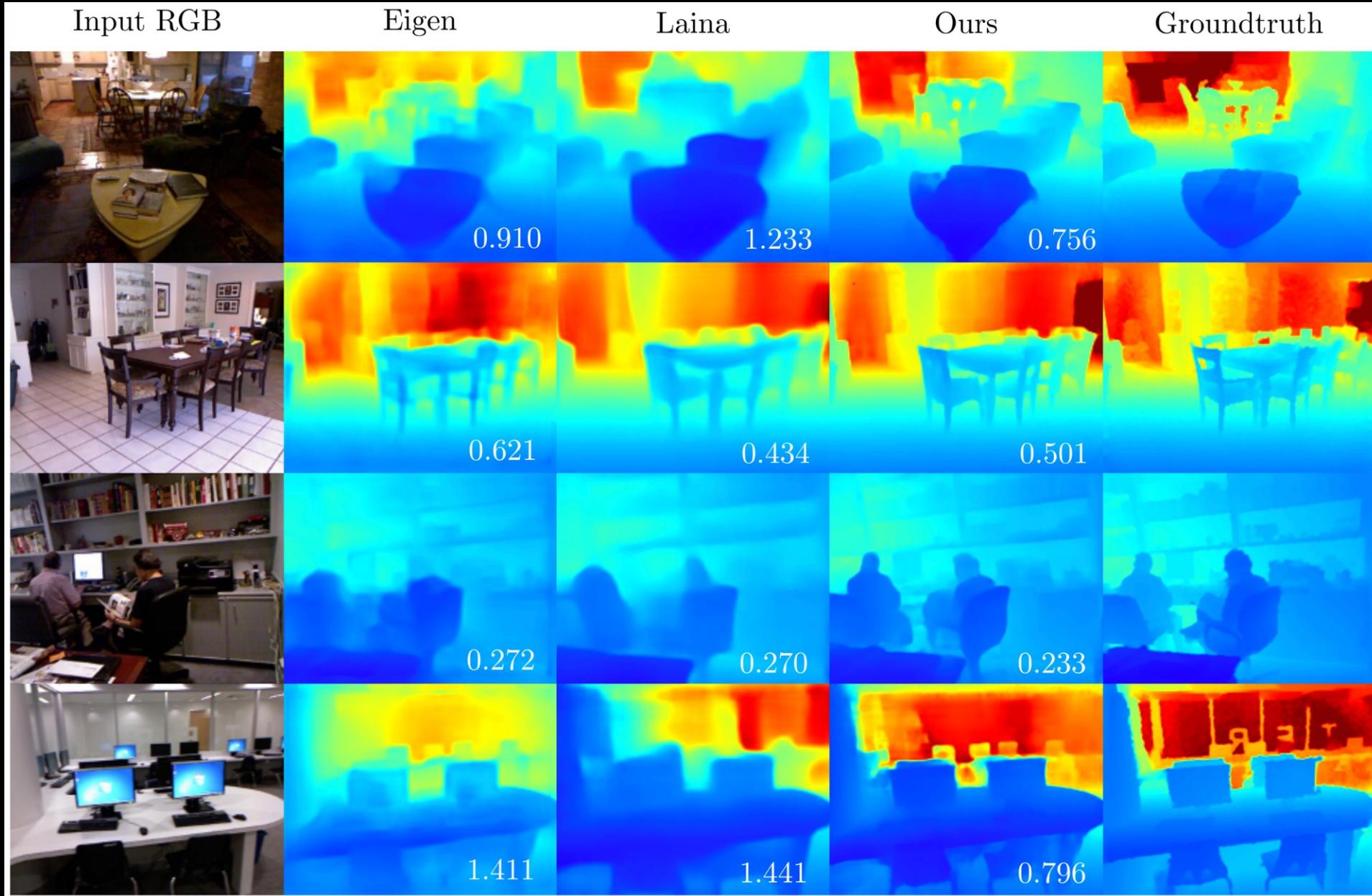
Q: How can we learn σ if we only have ground truth for μ ?

Estimating uncertainty in learning

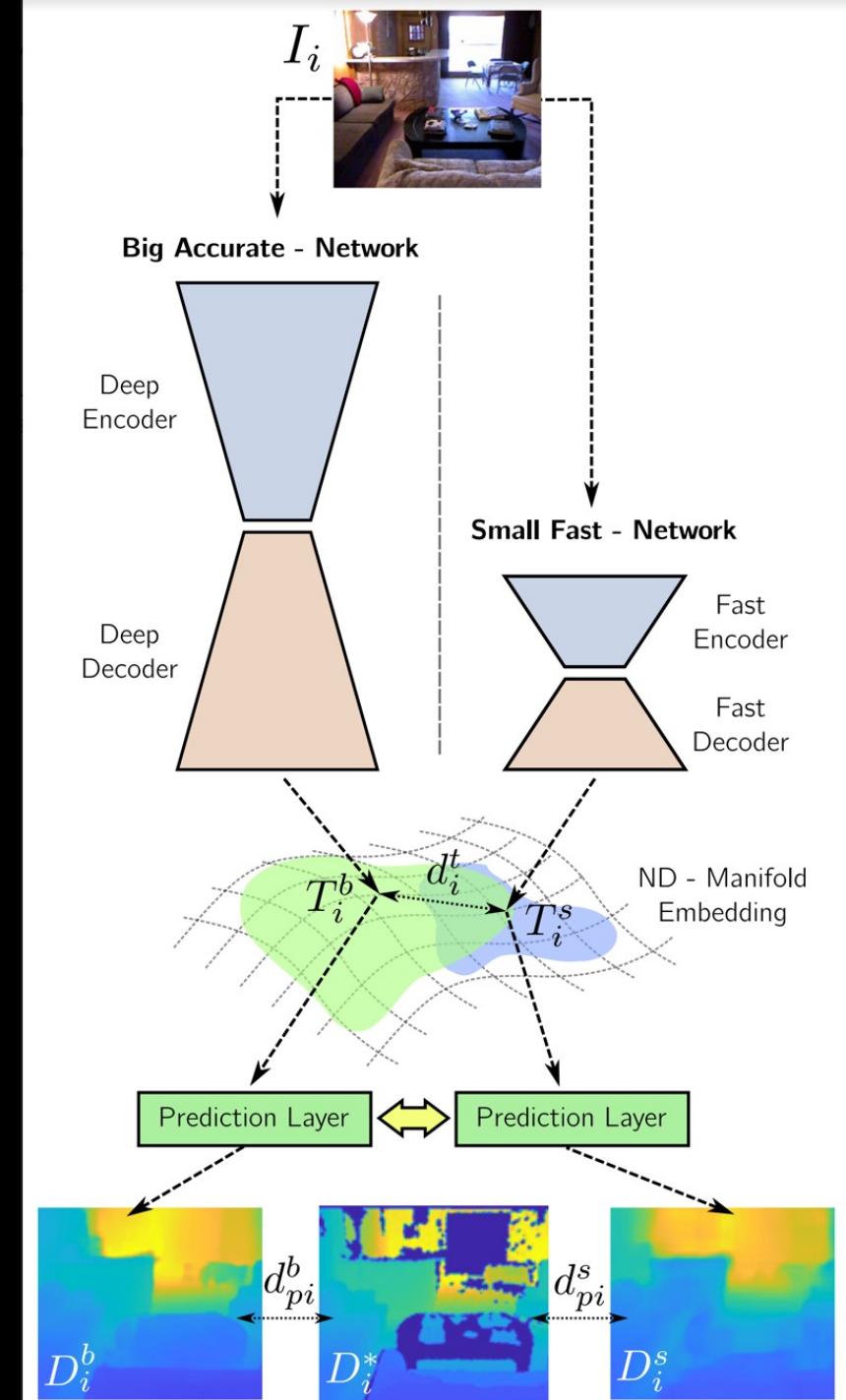
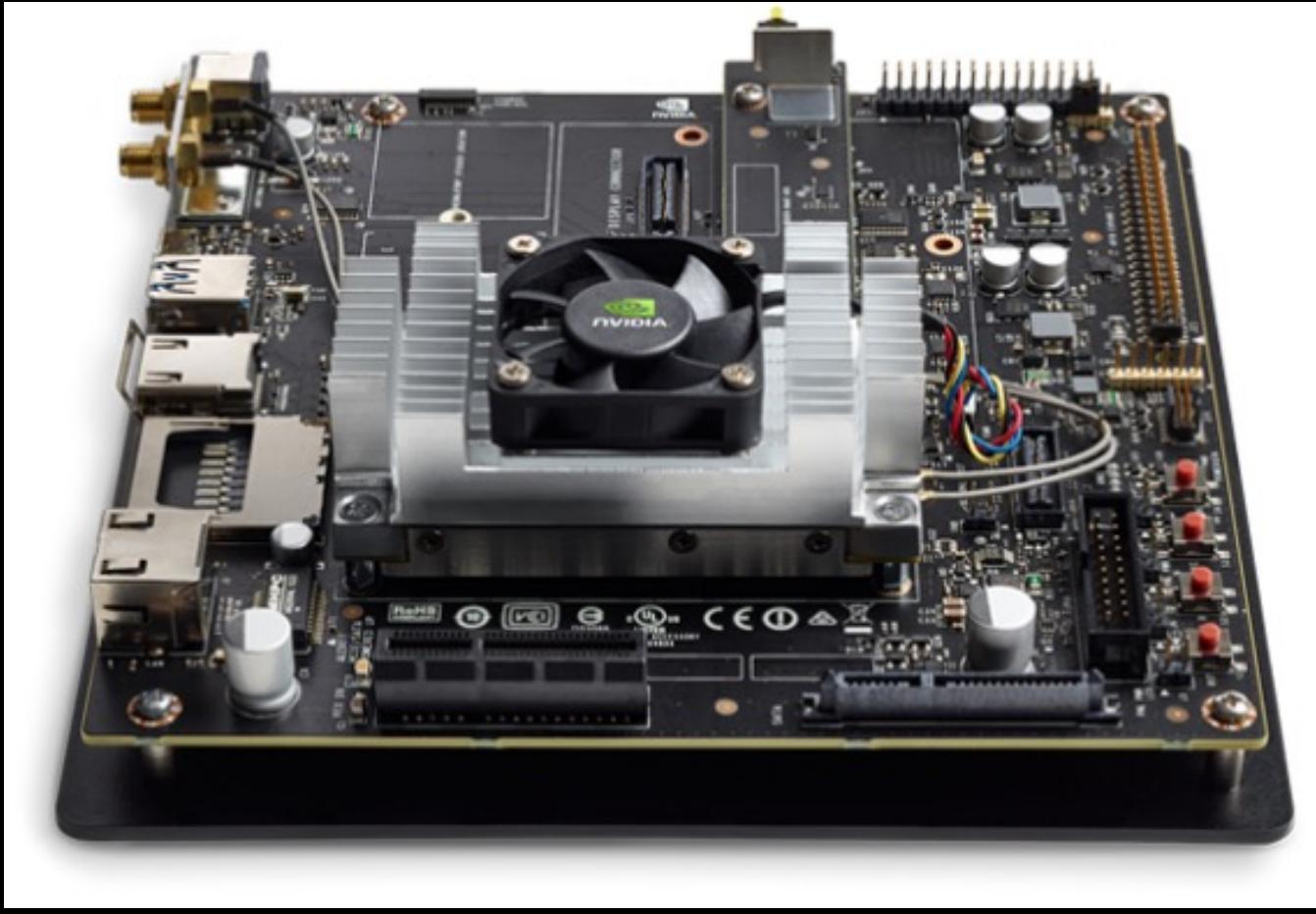
Uncertainty maps reflect the errors



Obligatory comparisons...



Speeding it up for robotics



Can use depth to improve ORB SLAM

Monocular Only

Before Loop-closure



After Loop-closure



Can use depth to improve ORB SLAM

Using Depth Predictions

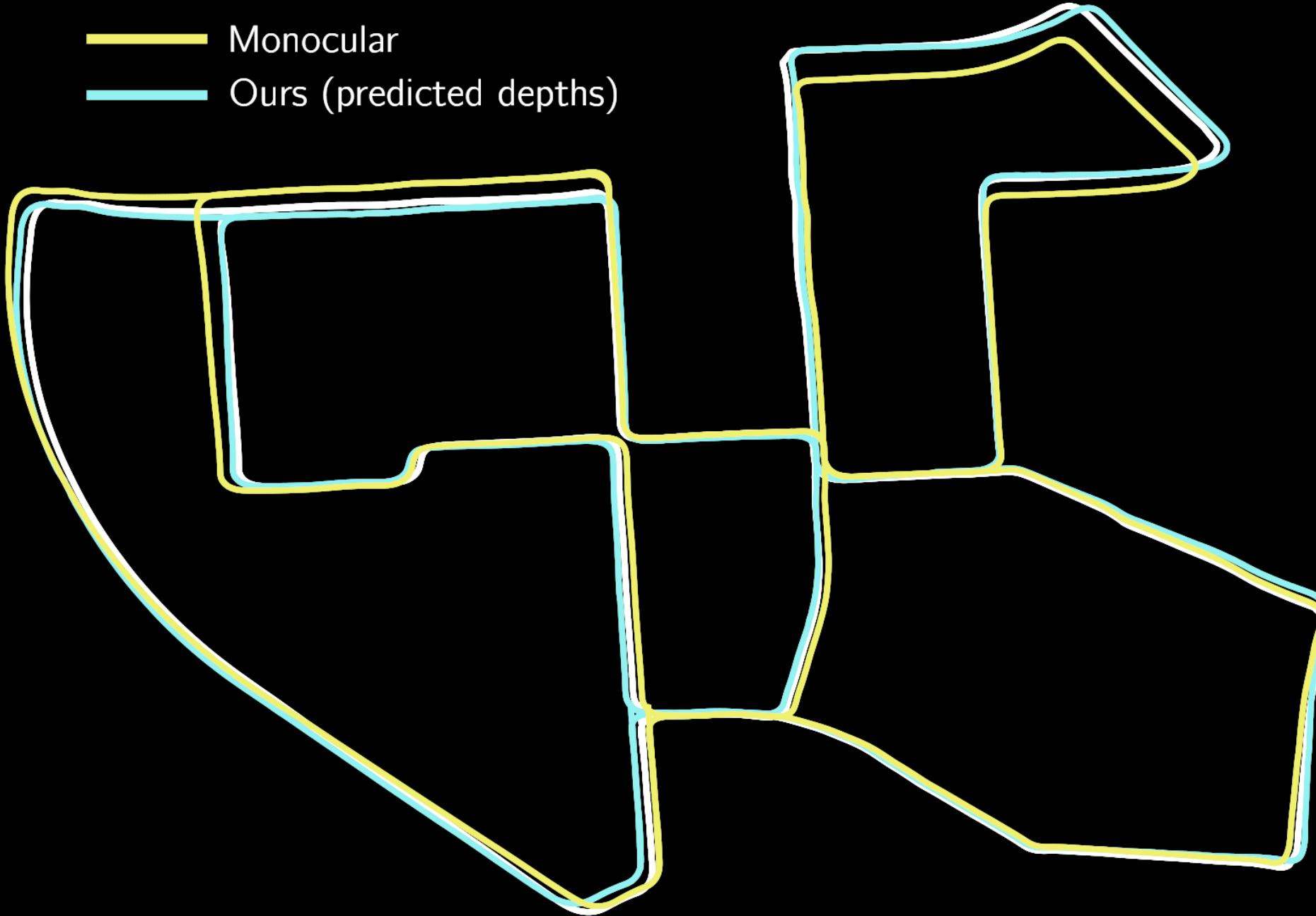
Before Loop-closure



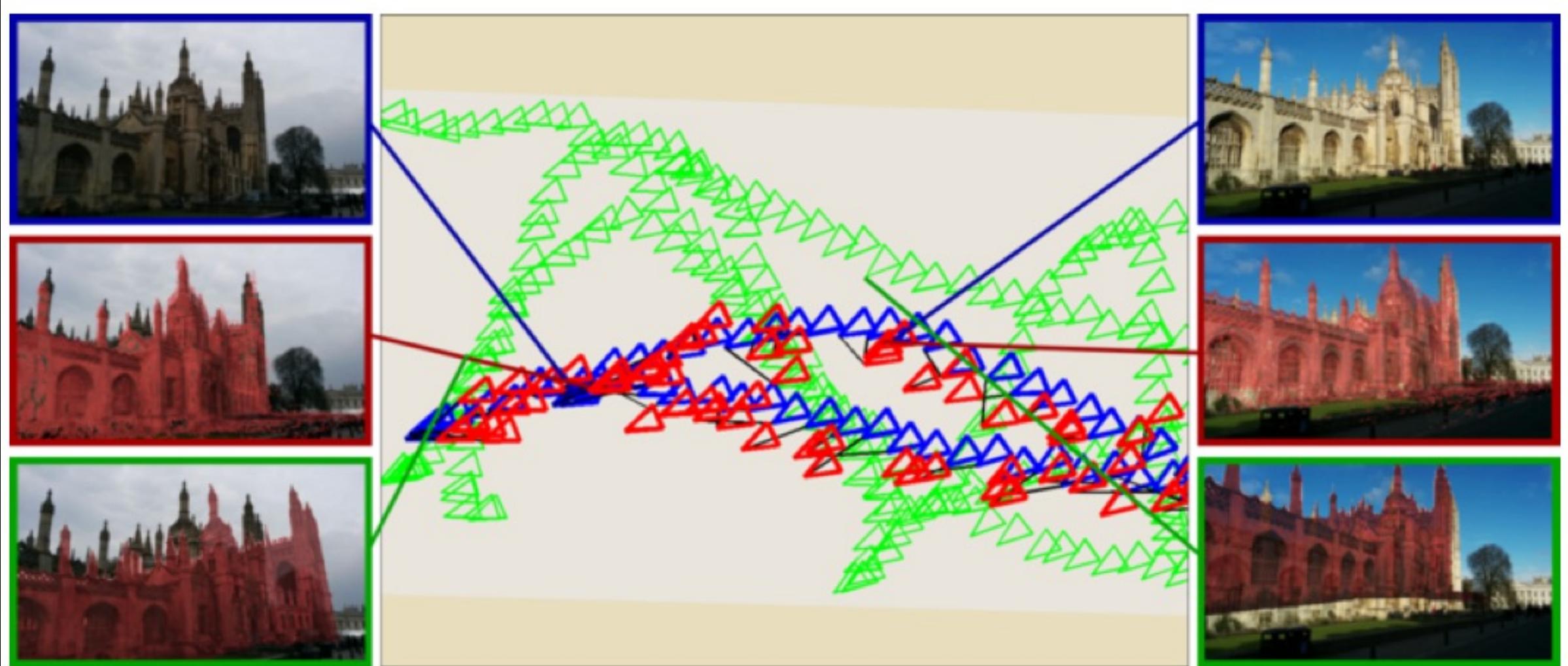
After Loop-closure



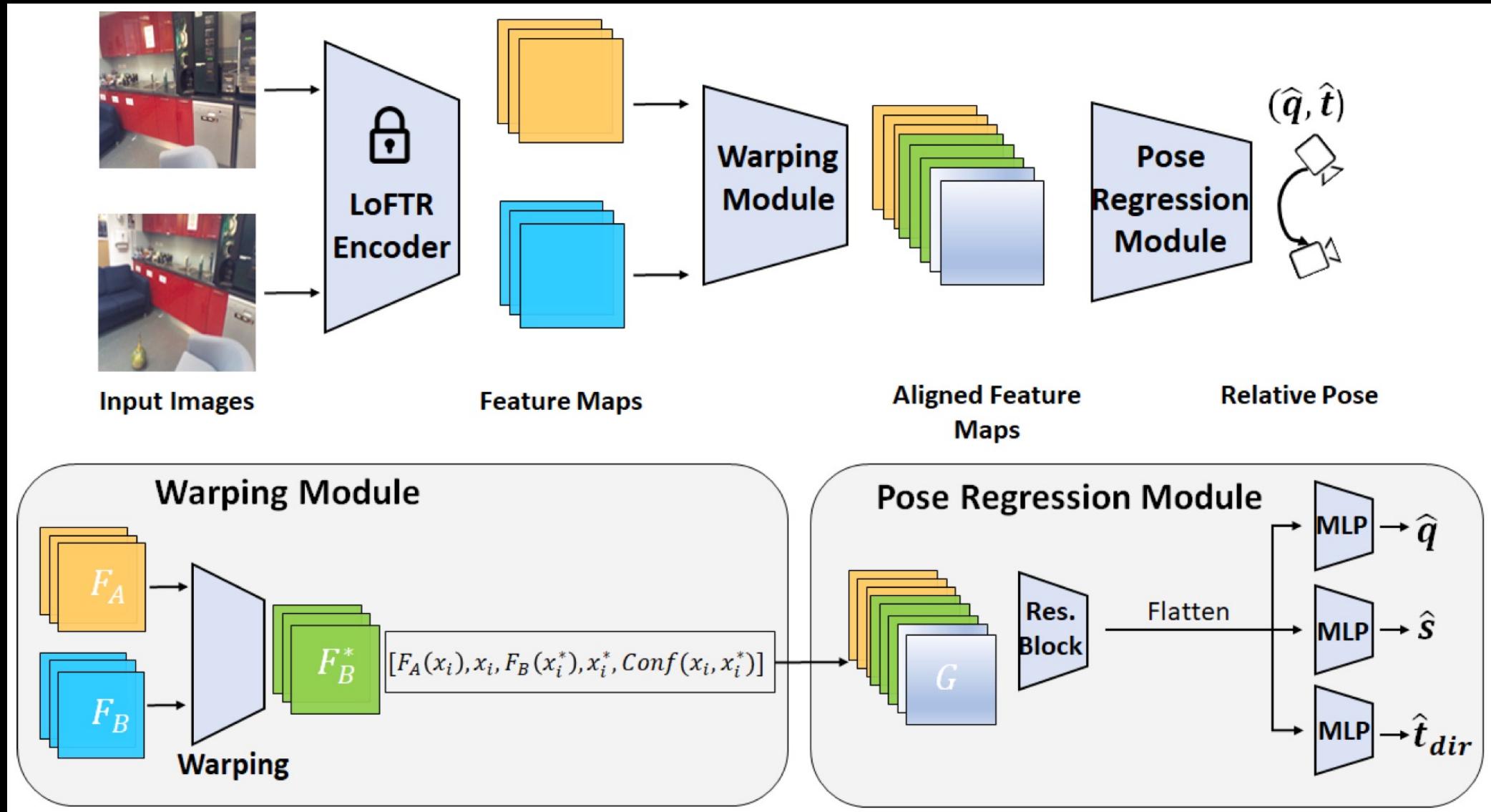
- Ground Truth
- Monocular
- Ours (predicted depths)



What is our trajectory through the world?

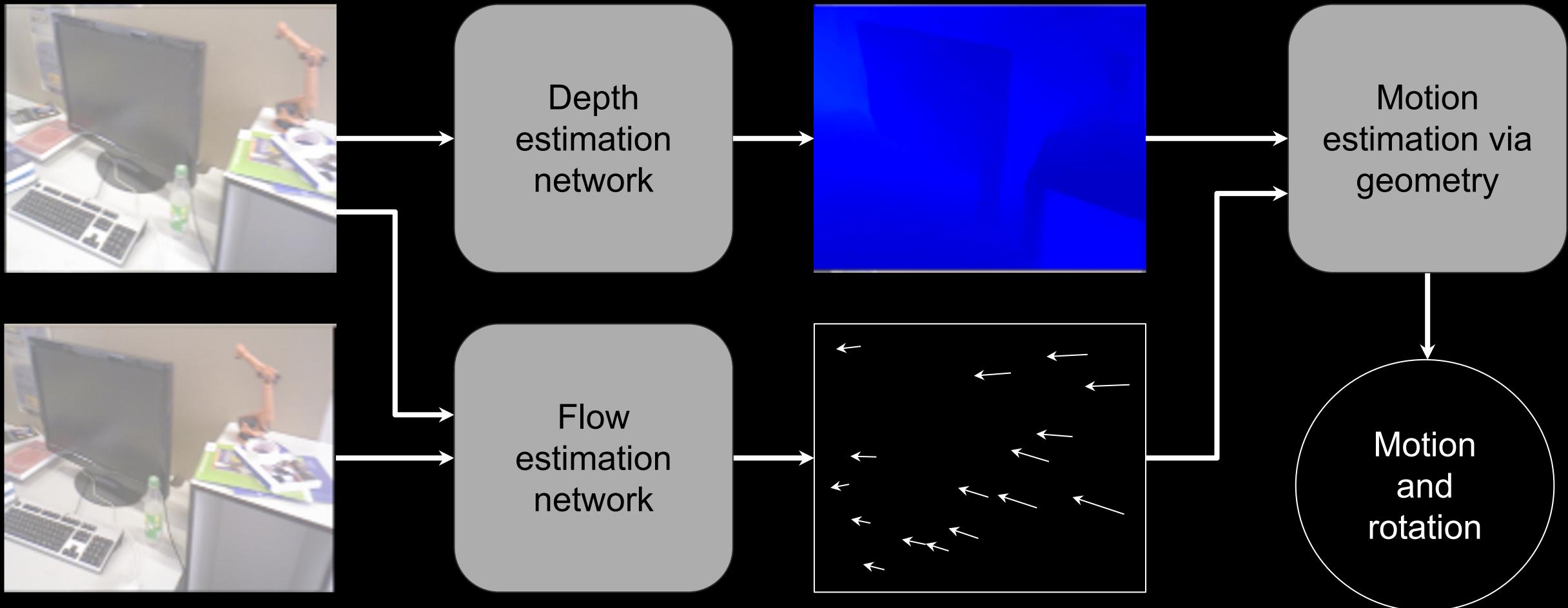


What is our trajectory through the world?

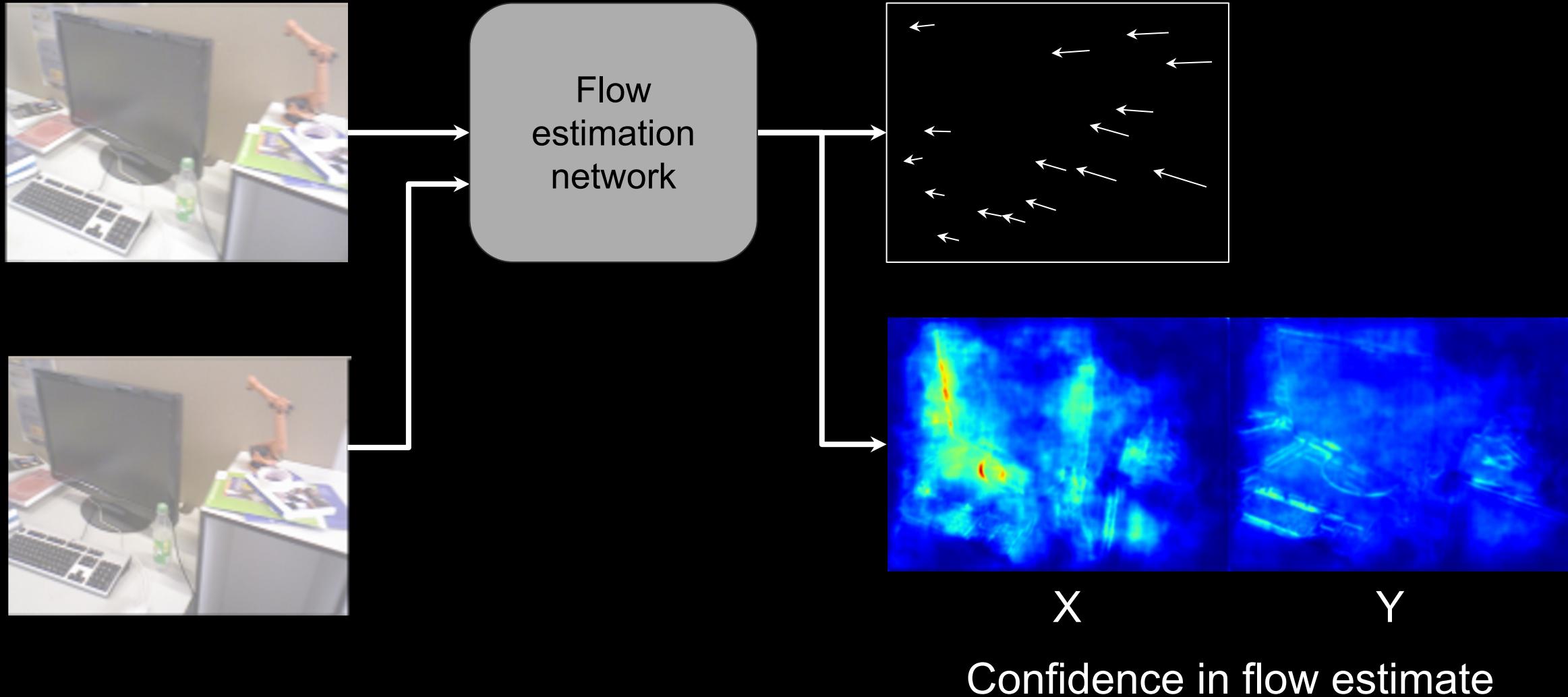


What is our trajectory through the world?

Estimate flow from two images; use this to get camera motion:

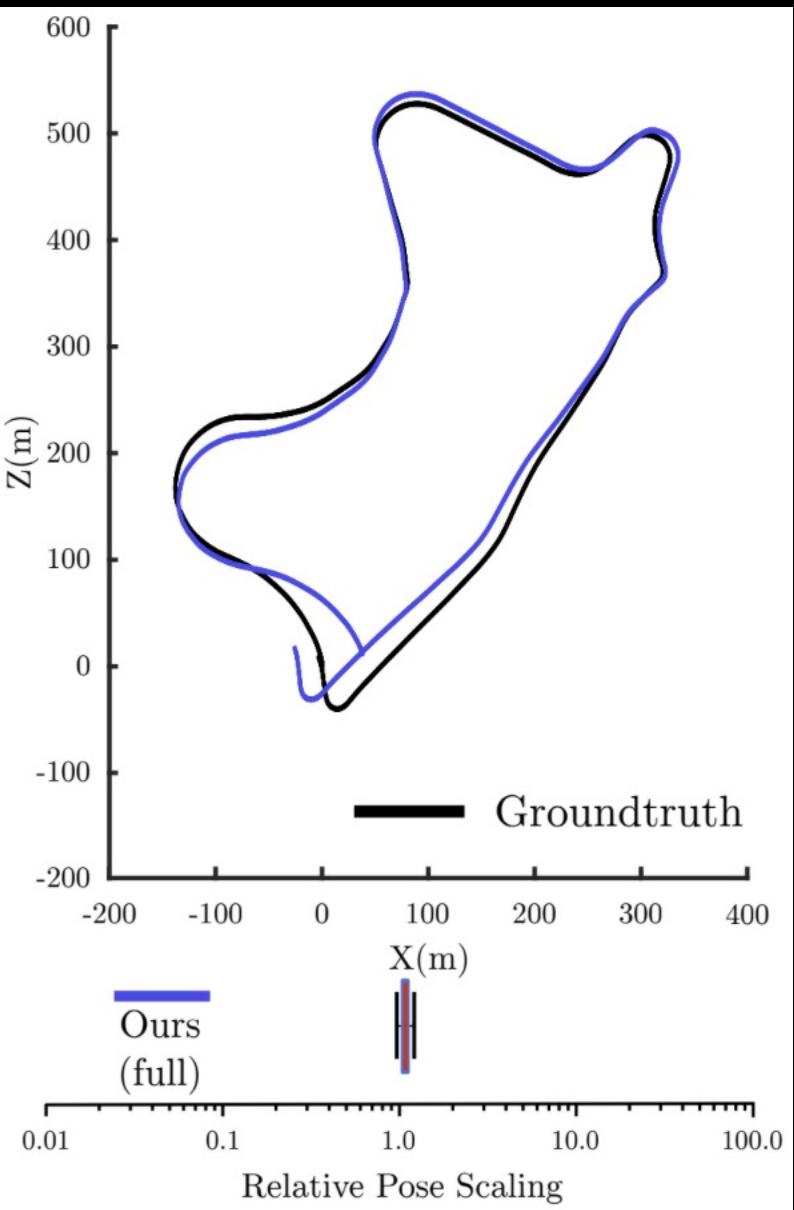
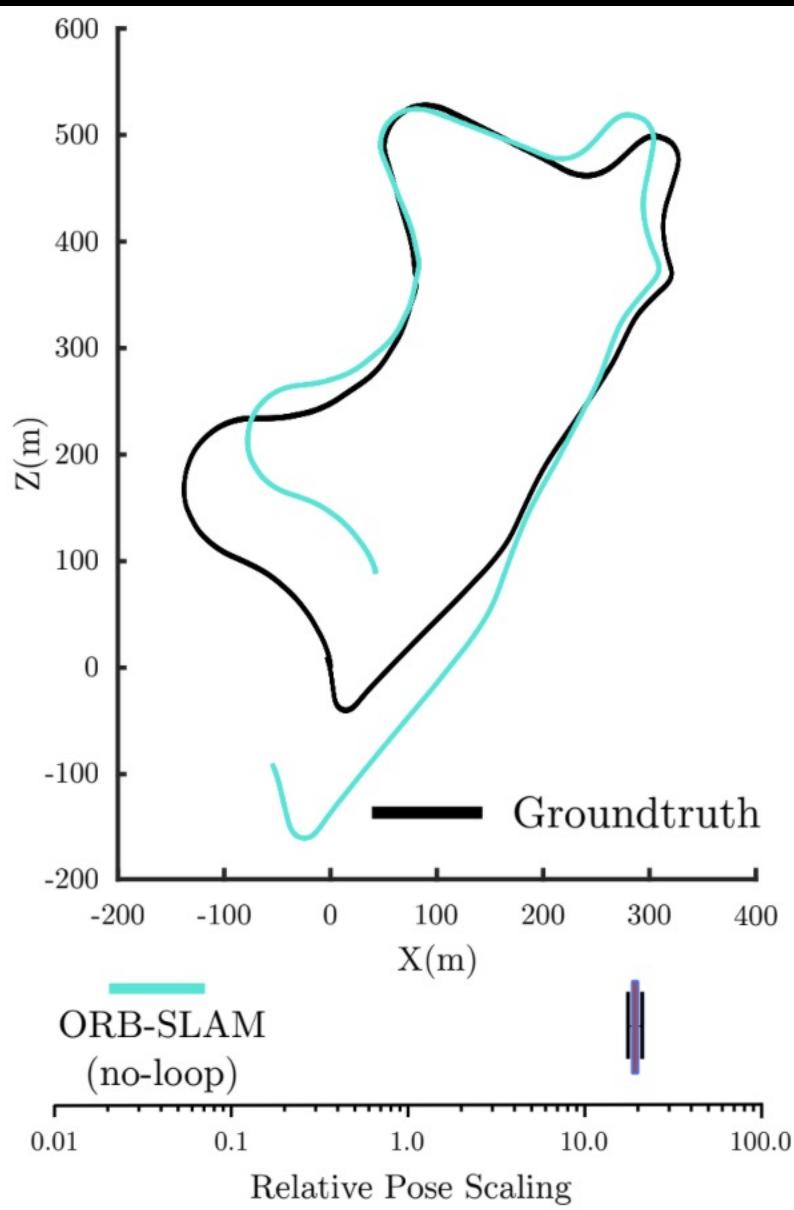
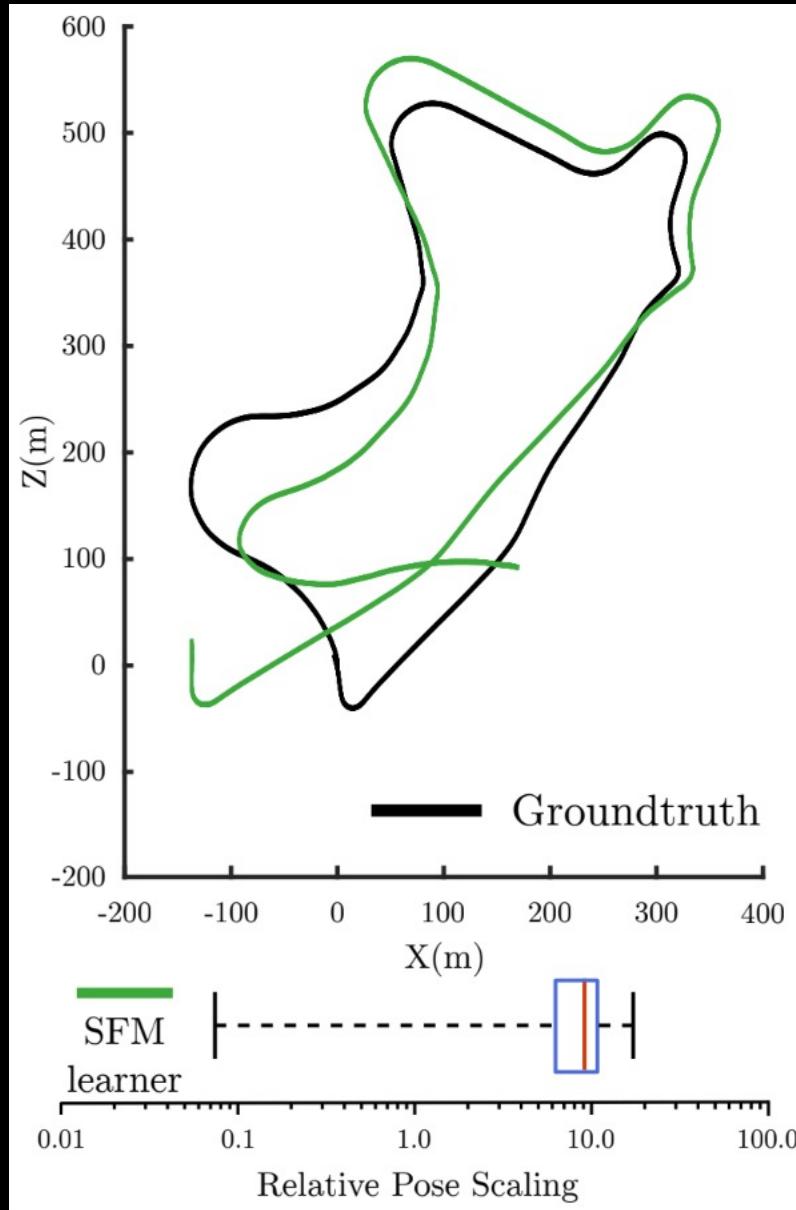


Better if uncertainty in flow is incorporated too



Parameterising flow uncertainty

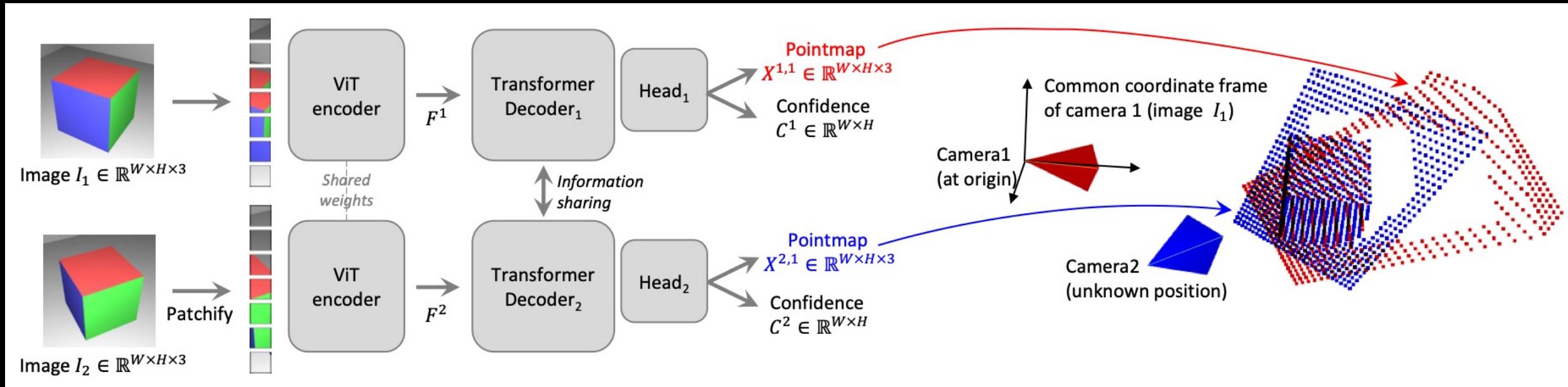
Visual Odometry (just prior to loop closure)



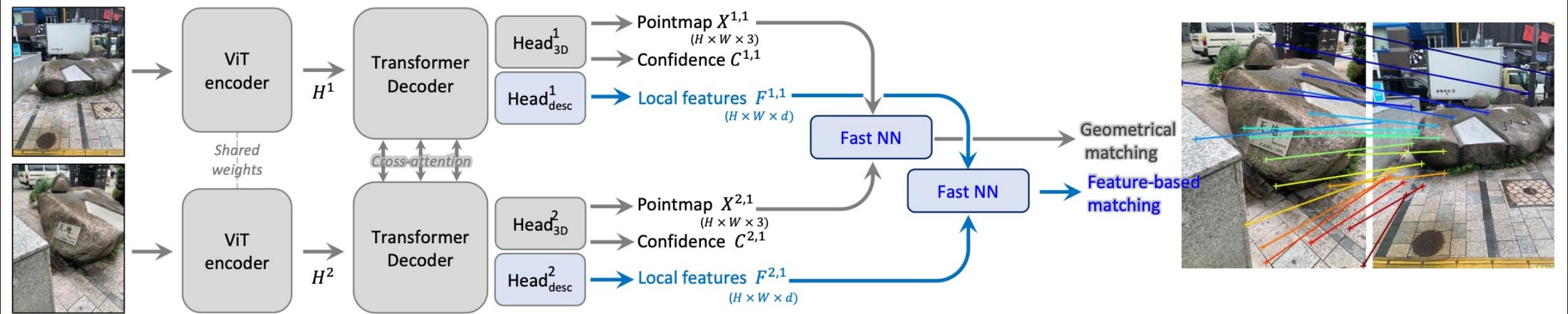
Interestingly, system learns to recognise independently moving objects and give them less confidence



Dust3r

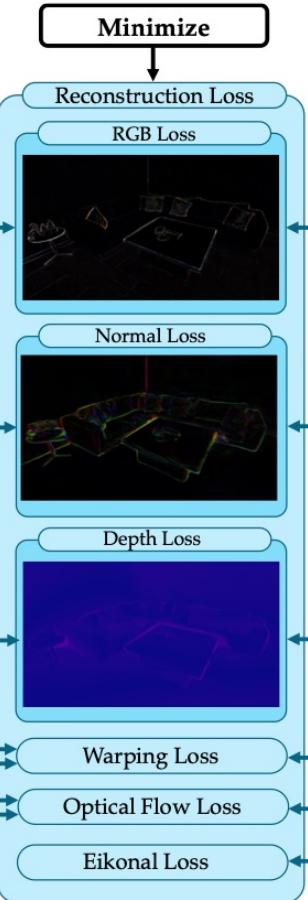
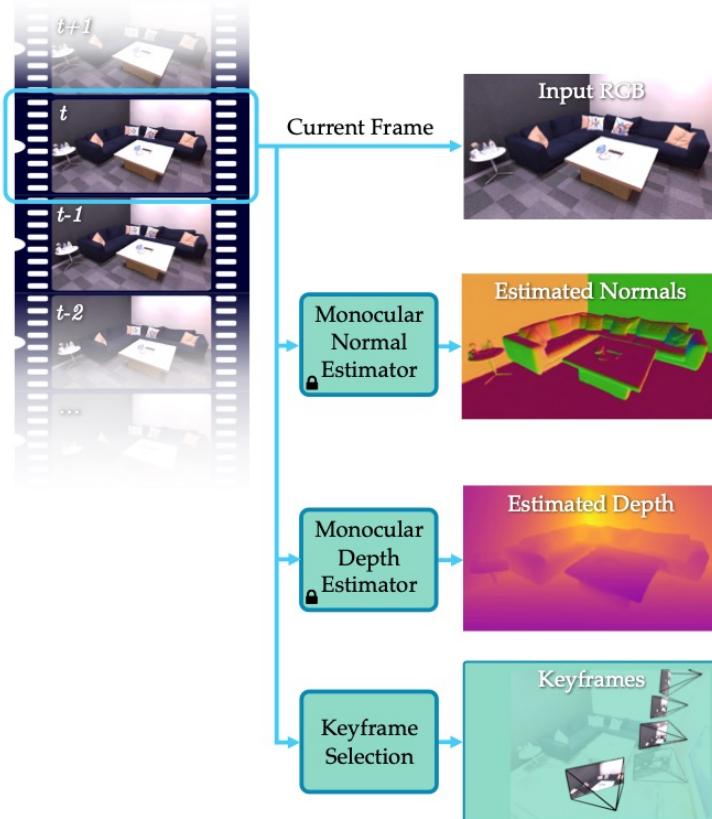


Mast3r

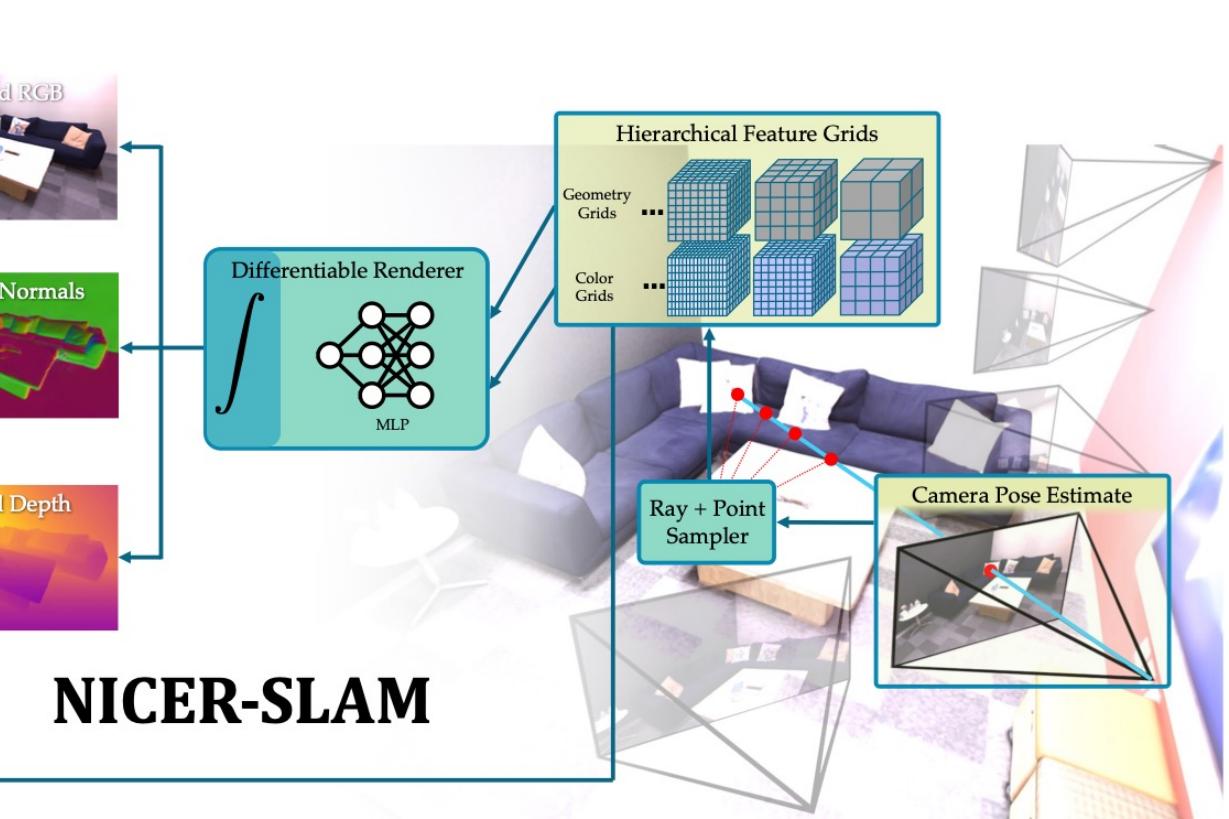


NICER-SLAM

Input RGB Stream



Mapping and Tracking Output



NICER-SLAM

