Metric-Semantic Simultaneous Localization and Mapping

Mo Shan

Advisor: Nikolay Atanasov

Existential Robotics Laboratory

University of California, San Diego

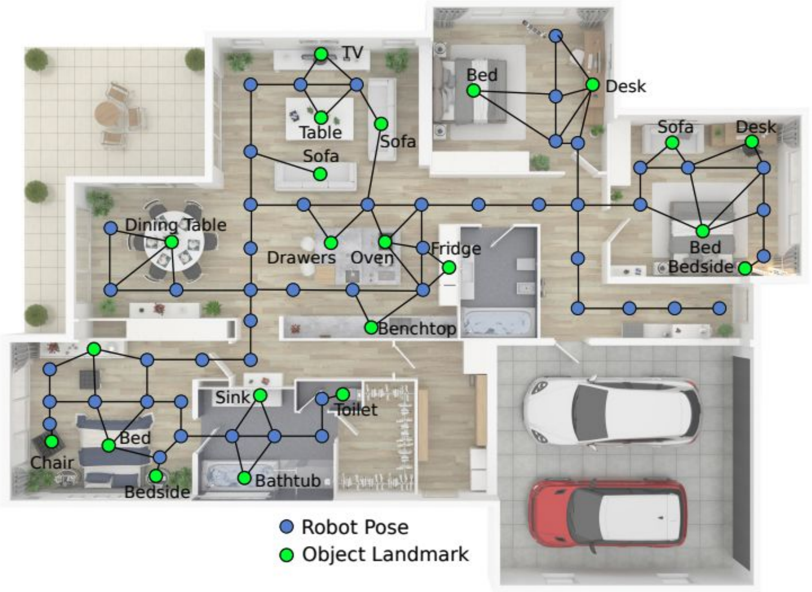
ERL

CONTEXTUAL ROBOTICS INSTITUTE

UC San Diego
JACOBS SCHOOL OF ENGINEERING
Electrical and Computer Engineering

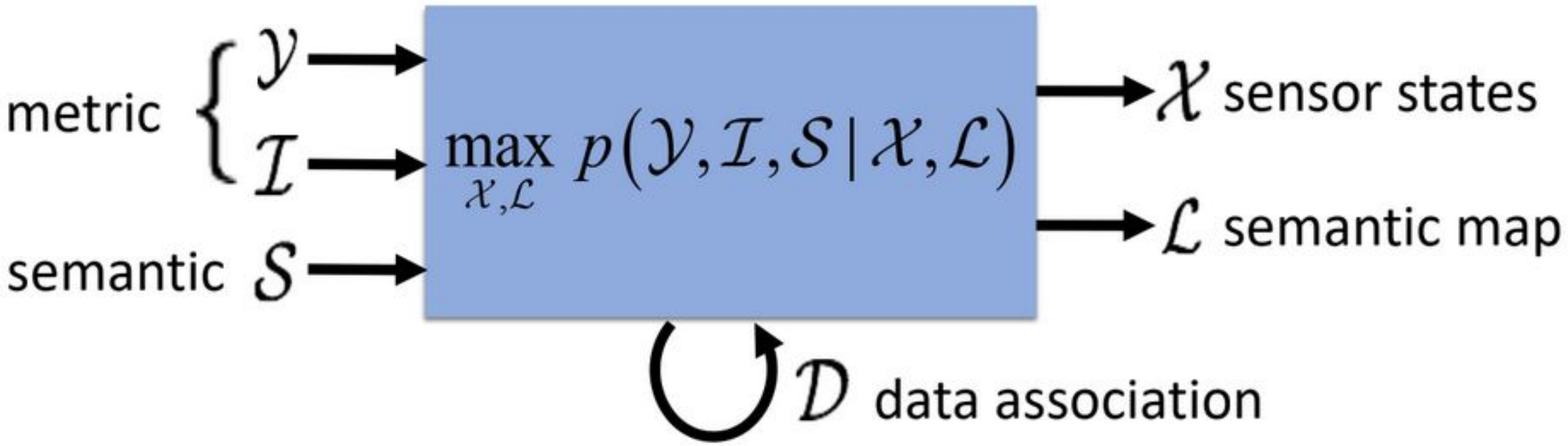
Metric-Semantic SLAM

- Why semantic localization:
  - Enables the robot to do loop closure to correct the drift
  - Can handle large baseline localization in the wild, by matching objects instead of geometric features
  - Execute tasks in terms of object entities



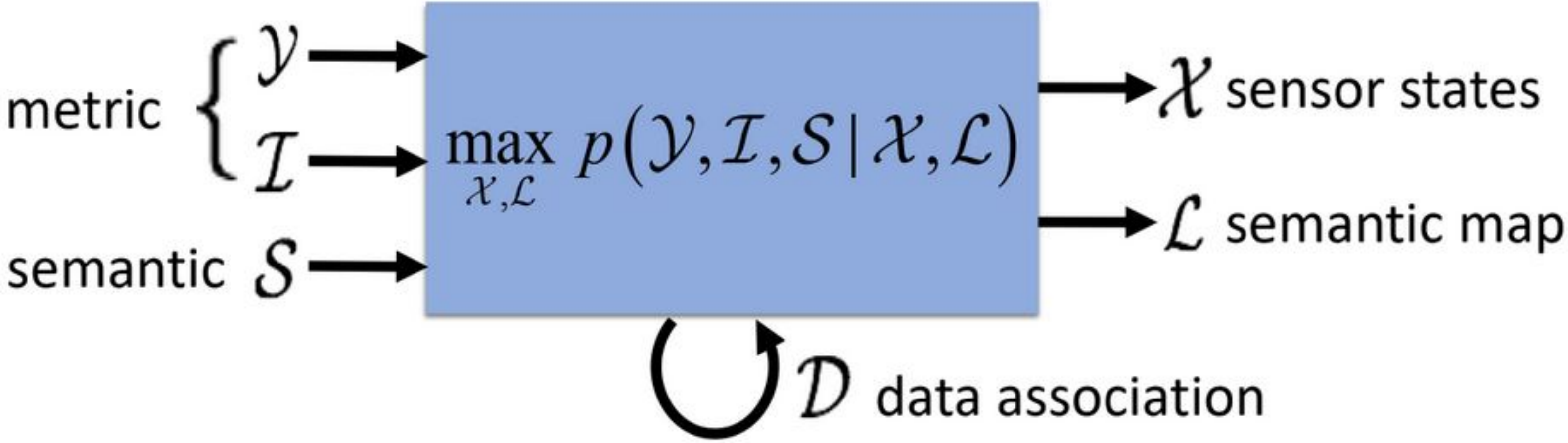
[1] Where are the Keys? – Learning Object-Centric Navigation Policies on Semantic Maps with Graph Convolutional Networks

Metric-Semantic SLAM

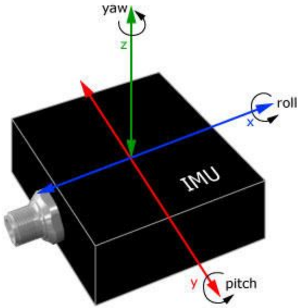


- Unified formulation of SLAM including:
  - Metric information: visual features, inertial measurements
  - Semantic information: object detections, object parts, semantic segmentation
  - Data association: correspondences among observations and landmarks

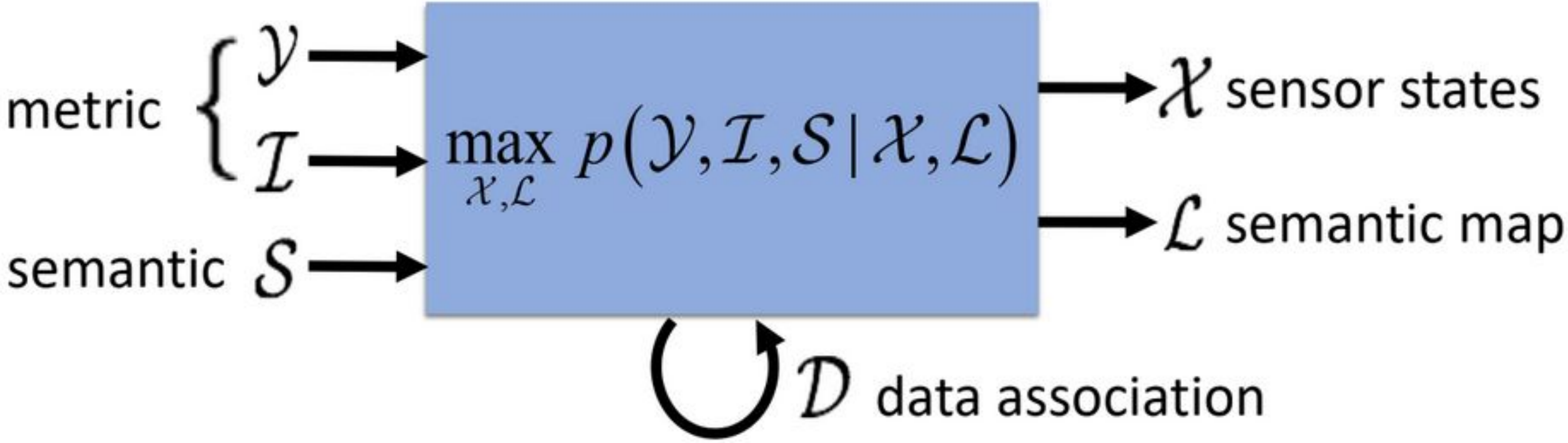
Metric-Semantic SLAM



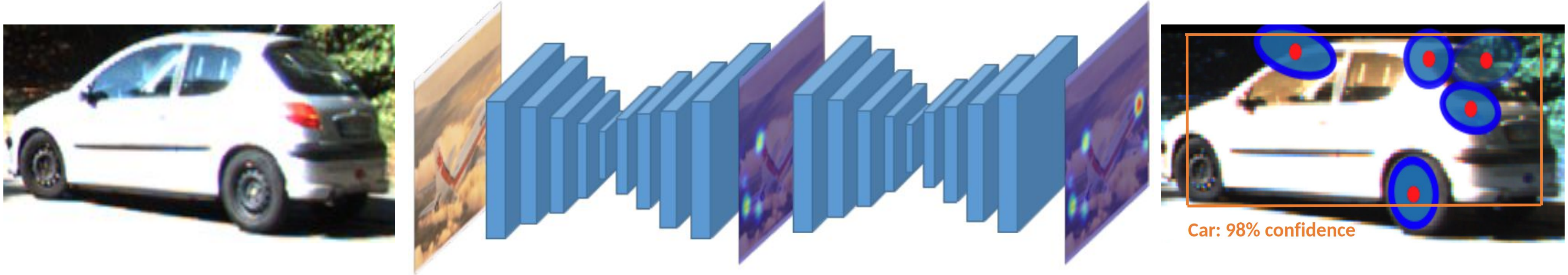
- Metric measurements include geometric features and IMU measurements



Metric-Semantic SLAM

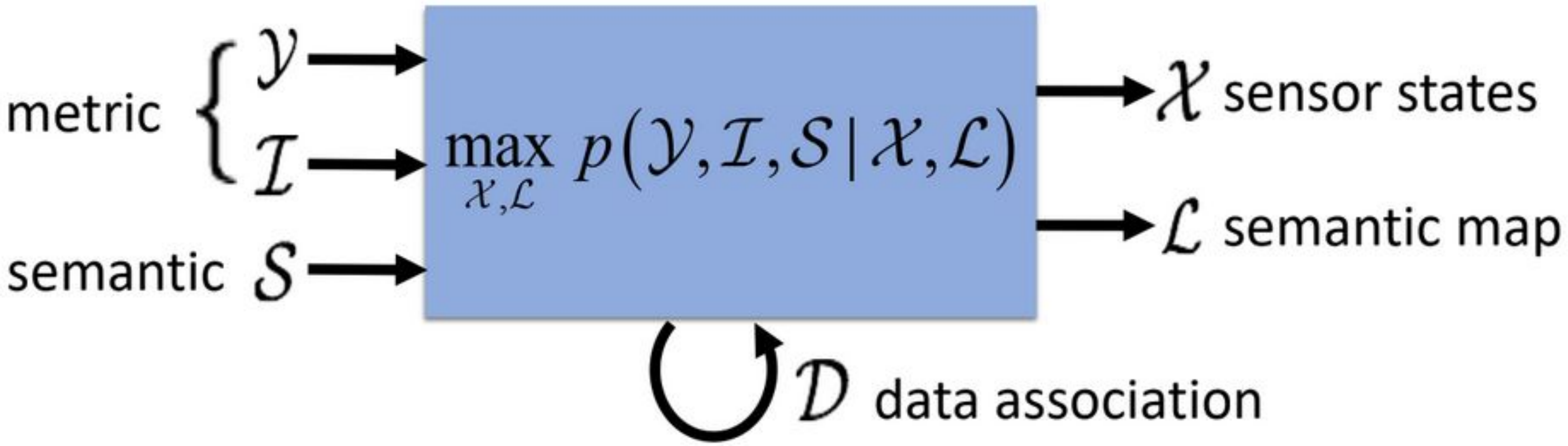


- Semantic measurements are object bounding boxes, semantic keypoints, etc



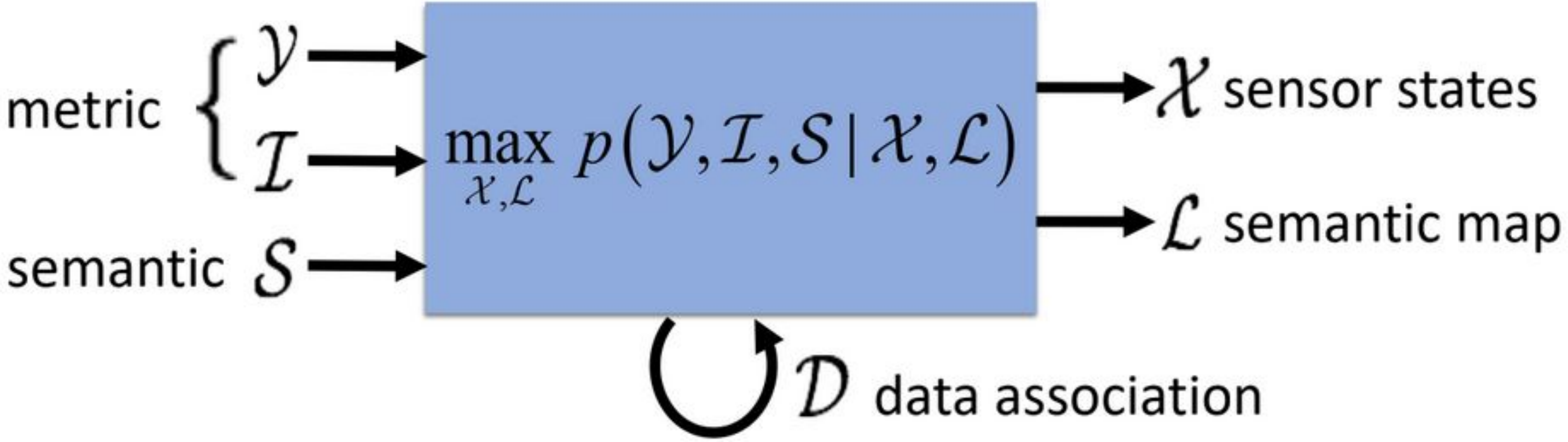
[1] StarMap for Category-Agnostic Keypoint and Viewpoint Estimation

Metric-Semantic SLAM

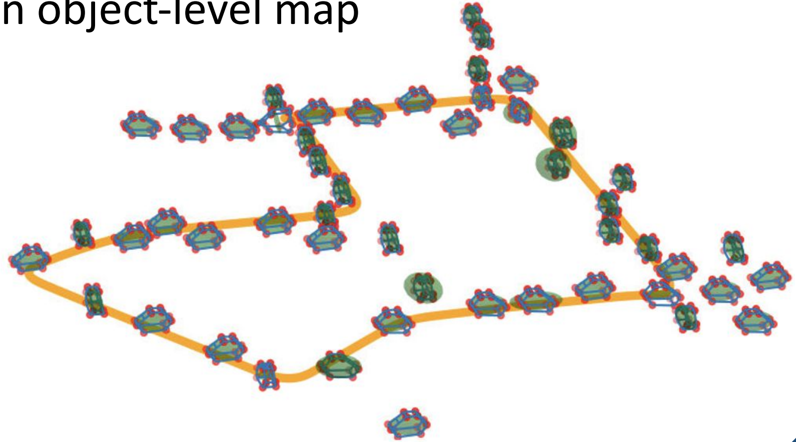


- Data association links the measurements to landmarks

Metric-Semantic SLAM

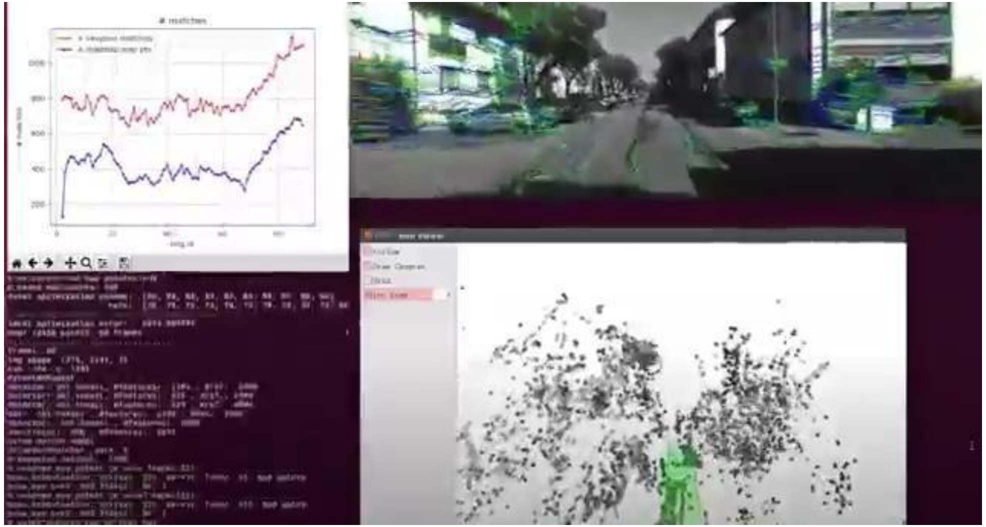


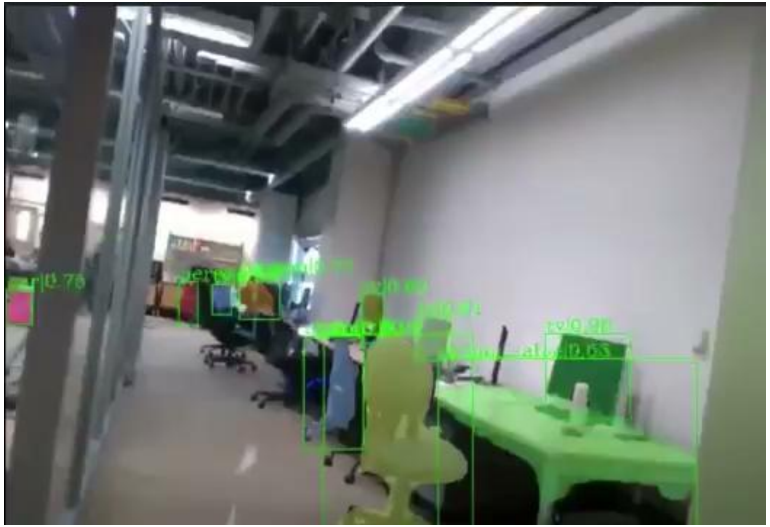
- Sensor states are in SE(3) and semantic map is an object-level map



Object residual constrained VIO

- Harness the strength of both VIO and deep neural networks
- Output geometrically consistent, semantically meaningful maps

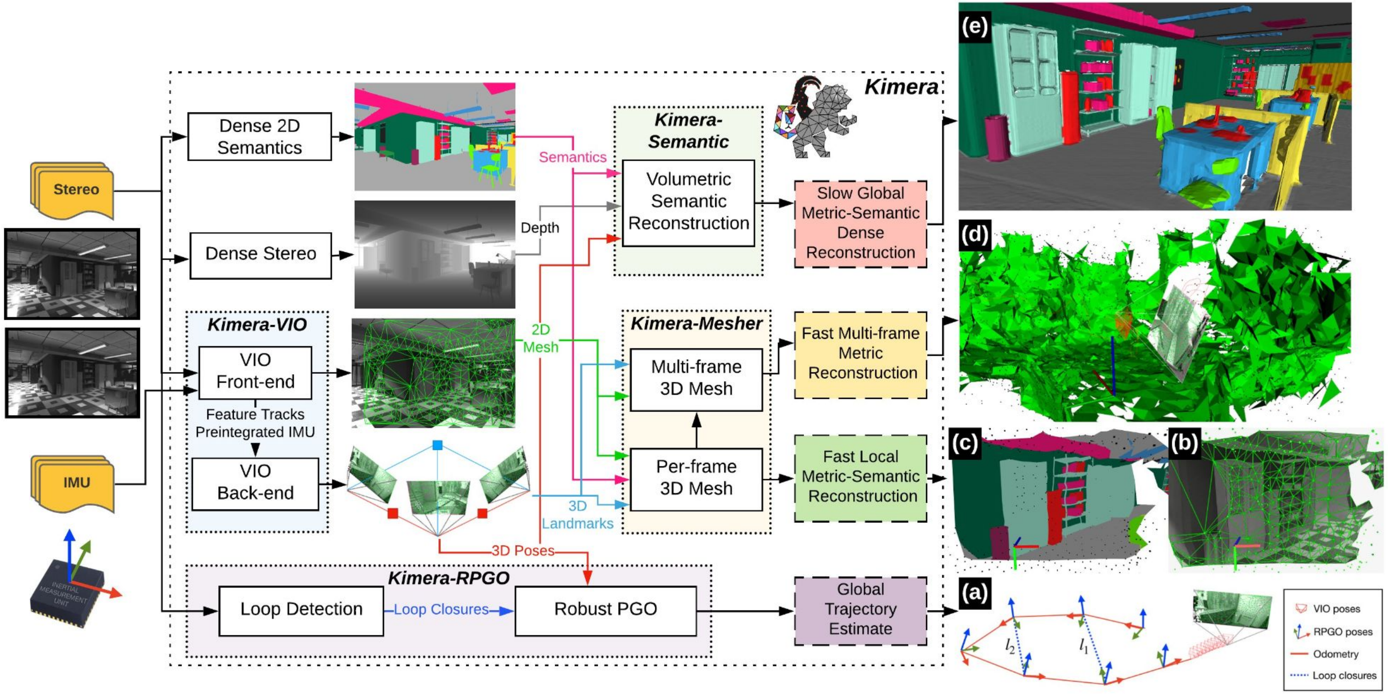




[1] ORB-SLAM: a Versatile and Accurate Monocular SLAM System
[2] Mask R-CNN

Semantic SLAM

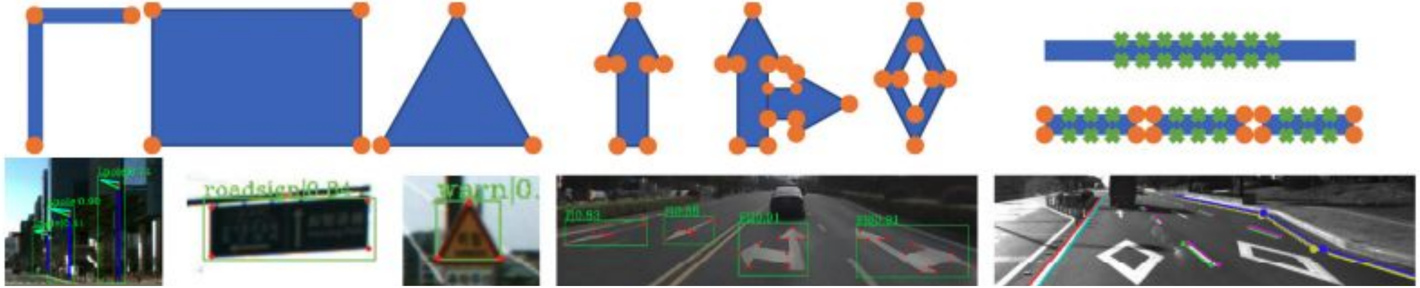
- Kimera uses visual and inertial measurements to build a semantically annotated 3D mesh of the scene

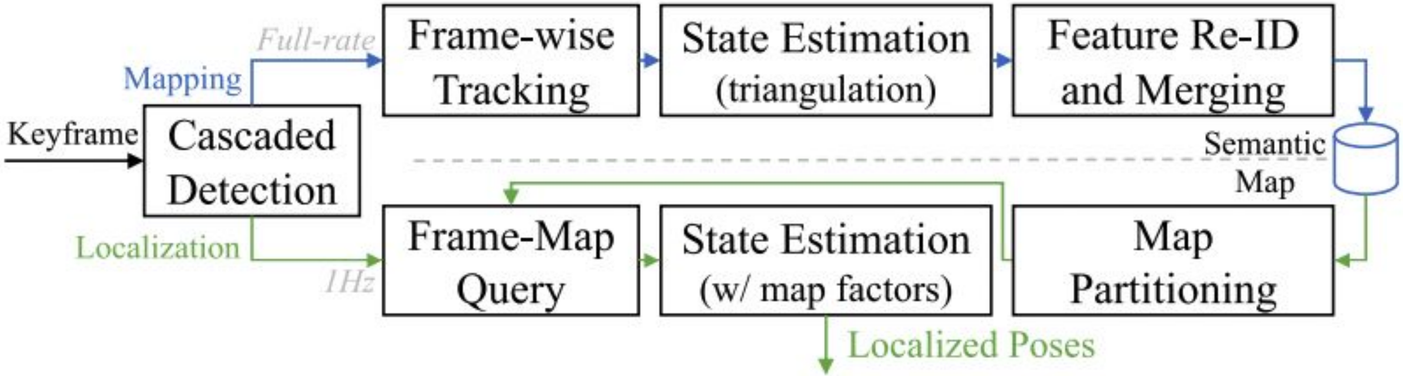


[1] Incremental Visual-Inertial 3D Mesh Generation with Structural Regularities

Semantic SLAM

- This work detects the road elements such as traffic signs, road lanes, and parameterizes the semantic elements to form a compact semantic map

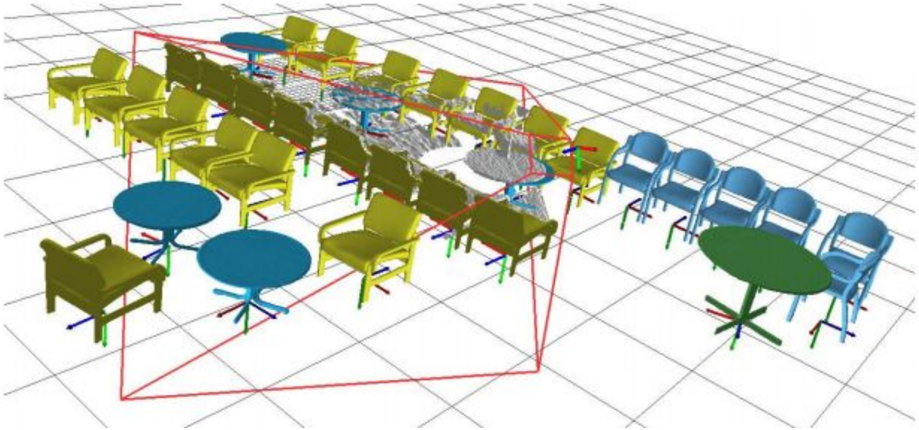


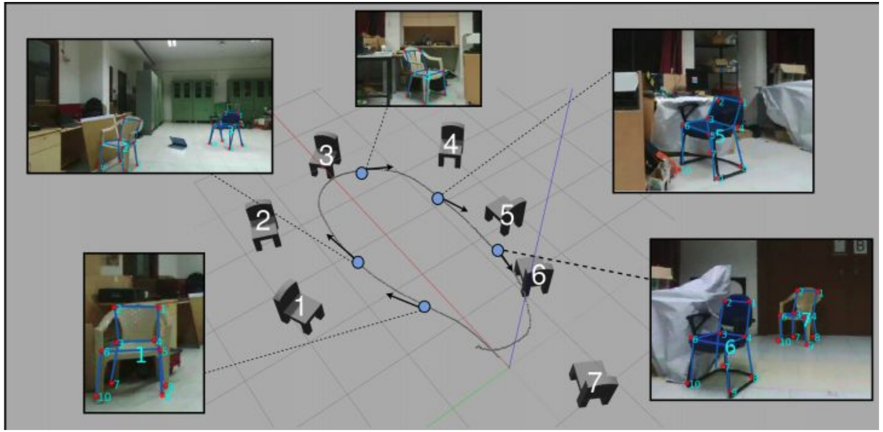


[1] Road Mapping and Localization Using Sparse Semantic Visual Features

Category-specific Object SLAM

- Category-specific approaches optimize the pose and shape of object instances using 3D shape models/semantic keypoints

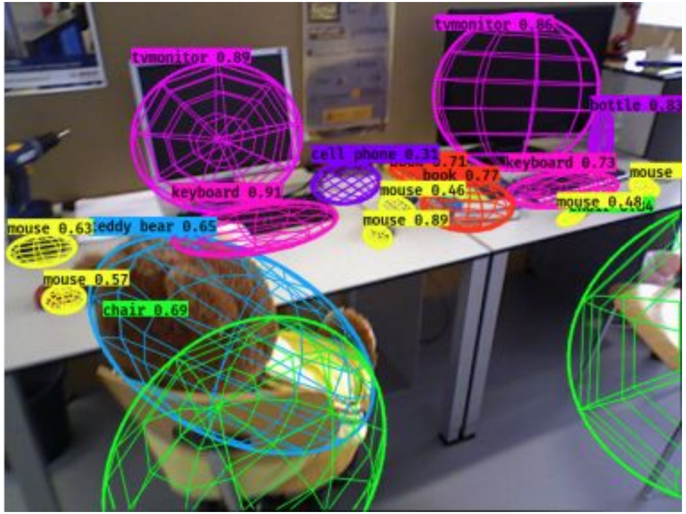


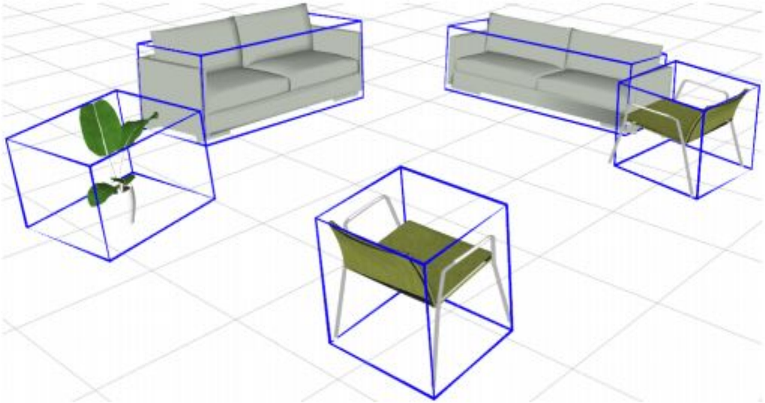


[1] Slam++: Simultaneous localisation and mapping at the level of objects
[2] Constructing category-specific models for monocular object-slam

Category-agnostic Object SLAM

- Category-agnostic approaches use geometric shapes such as ellipsoids or cuboids to represent objects



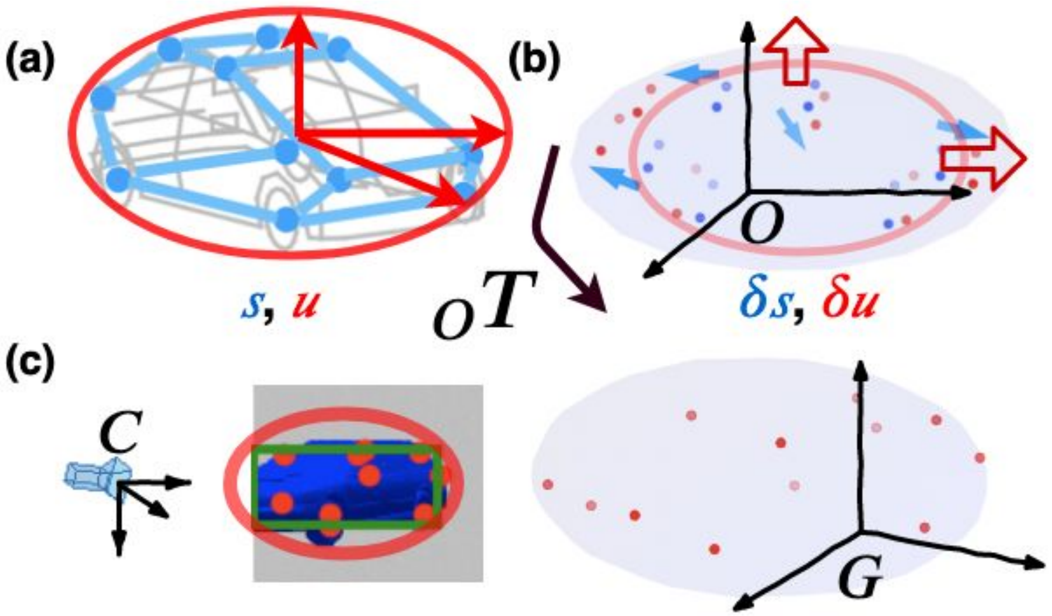


[1] Quadricslam: Dual quadrics from object detections as landmarks in object-oriented slam
[2] Cubeslam: Monocular 3-d object slam

Bi-level Object Model

- Object class: ellipsoid (coarse level) and keypoints (fine level)
- Object instance: deformations of ellipsoid, mean shape and pose

- Object Class: $(\sigma, \boldsymbol{s}, \boldsymbol{u})$
  - $\sigma \in \{car, chair, table, \dots\}$
  - $\boldsymbol{s} \in \mathbb{R}^{3 \times N}$ positions of N semantic landmarks in the object frame
  - $\boldsymbol{u} \in \mathbb{R}^3$ shape: $\mathcal{E}_{\boldsymbol{u}} = \{\boldsymbol{x} \mid \boldsymbol{x}^T diag(\boldsymbol{u})^{-2} \boldsymbol{x} \leq 1\}$
- Object Instance: $({}_O\boldsymbol{T}, \delta\boldsymbol{s}, \delta\boldsymbol{u})$
  - ${}_O\boldsymbol{T} \in SE(3)$ world frame pose
  - $\delta\boldsymbol{s} \in \mathbb{R}^{3 \times N}$ position deformations of the N semantic landmarks in 3D
  - $\delta\boldsymbol{u} \in \mathbb{R}^3$ shape deformation



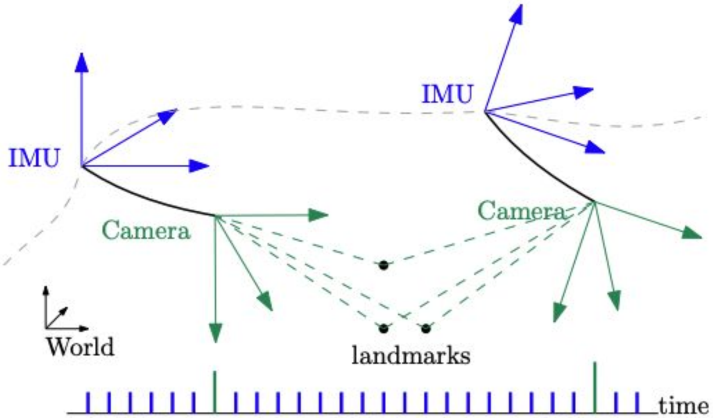
[1] OrcVIO: Object residual constrained Visual-Inertial Odometry

Sensor states

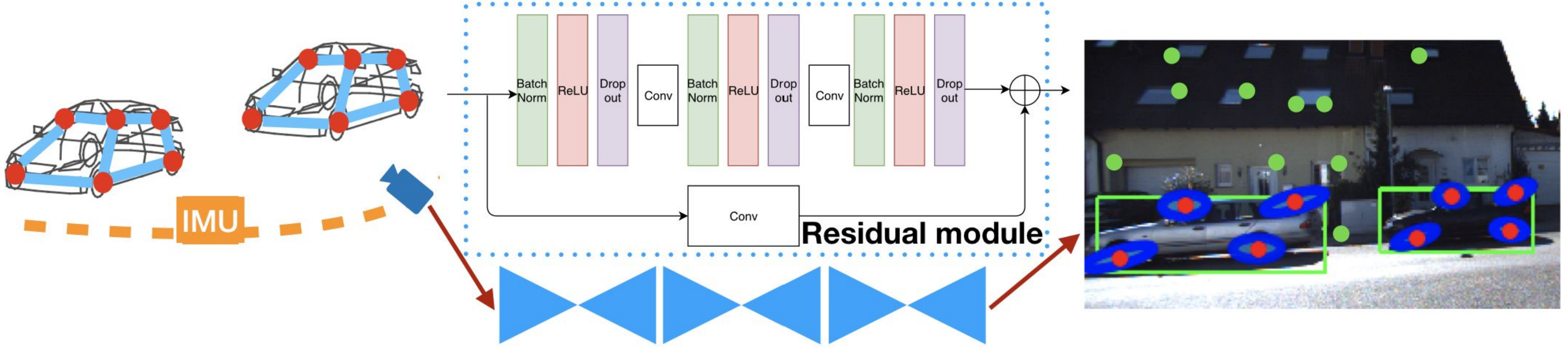
- Sensor states consist of IMU state and camera states

IMU state: ${}_I\boldsymbol{x} = ({}_I\boldsymbol{R}, {}_I\boldsymbol{p}, {}_I\boldsymbol{v}, \boldsymbol{b}_g, \boldsymbol{b}_a)$
- IMU orientation: ${}_I R \in \mathrm{SO}(3)$
- IMU position: ${}_I\boldsymbol{p} \in \mathbb{R}^3$
- IMU velocity: ${}_I\boldsymbol{v} \in \mathbb{R}^3$
- IMU bias: $\boldsymbol{b}_g \in \mathbb{R}^3, \boldsymbol{b}_a \in \mathbb{R}^3$

Camera state: ${}_C\boldsymbol{x}_t$
- Camera pose ${}_C\boldsymbol{T}_t \in SE(3)$
- Cam-to-IMU frame transformation: ${}^I_C\boldsymbol{T} \in SE(3)$



Problem Formulation



(a) 3D scene (b) Stacked hourglass network (c) Visual observation

$$\min_{\{x_t\},\{o_i\},\{\ell_m\}} {}^i w \sum_t \|{}^i e(x_t, x_{t+1}, {}^i z_t)\|^2$$ → Inertial error

$$+ {}^g w \sum_{t,m,n} \left\|{}^g e\left(x_t, \ell_m, {}^g z_{t,n}\right)\right\|^2$$ → Geometric keypoint error

$$+ {}^s w \sum_{t,i,j,k} \left\|{}^s e\left(x_t, o_i, {}^s z_{t,j,k}\right)\right\|^2$$ → Semantic keypoint error

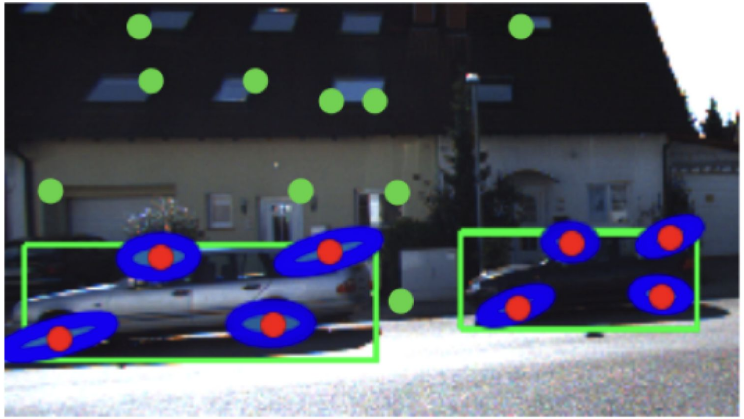
$$+ {}^b w \sum_{t,i,j,k} \left\|{}^b e\left(x_t, o_i, {}^b z_{t,j,k}\right)\right\|^2$$ → Bounding box error

$$+ {}^r w \sum_i \|{}^r e(o_i)\|^2$$ → Object shape regularization

Optimization variables: $\{x_t\}$ — IMU-Camera Trajectory; $\{o_i\}$ — Objects; $\{\ell_m\}$ — Geometric Landmarks

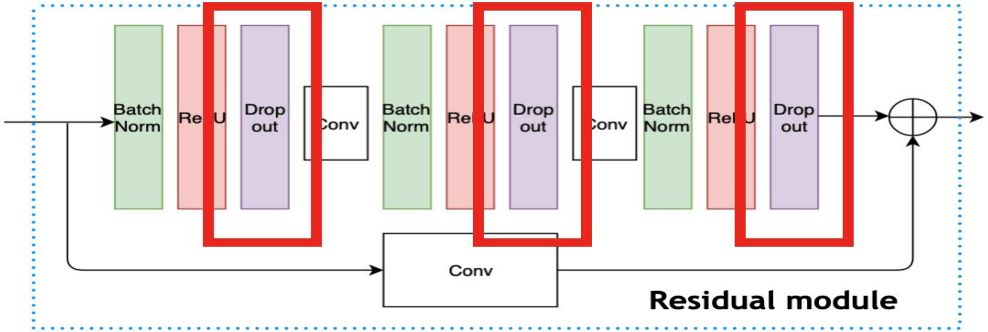
Front End

- Geometric features $g_{\mathbf{z}_{t,n}} \in \mathbb{R}^2$
  - normalized pixel coordinates of n-th keypoint at time t
- Semantic Features ${}^{s}\mathbf{Z}_{t,j,k} \in \mathbb{R}^2$
  - normalized pixel coordinates of j-th keypoint of object detection k at time t
- Bounding box $b_{Z_{t,j,k}} \in \mathbb{R}^2$
  - normalized pixel coordinates of j-th line of object bounding box k



Front End

- StarMap is used to detect semantic keypoints
- We add drop out layers in original network to obtain covariance



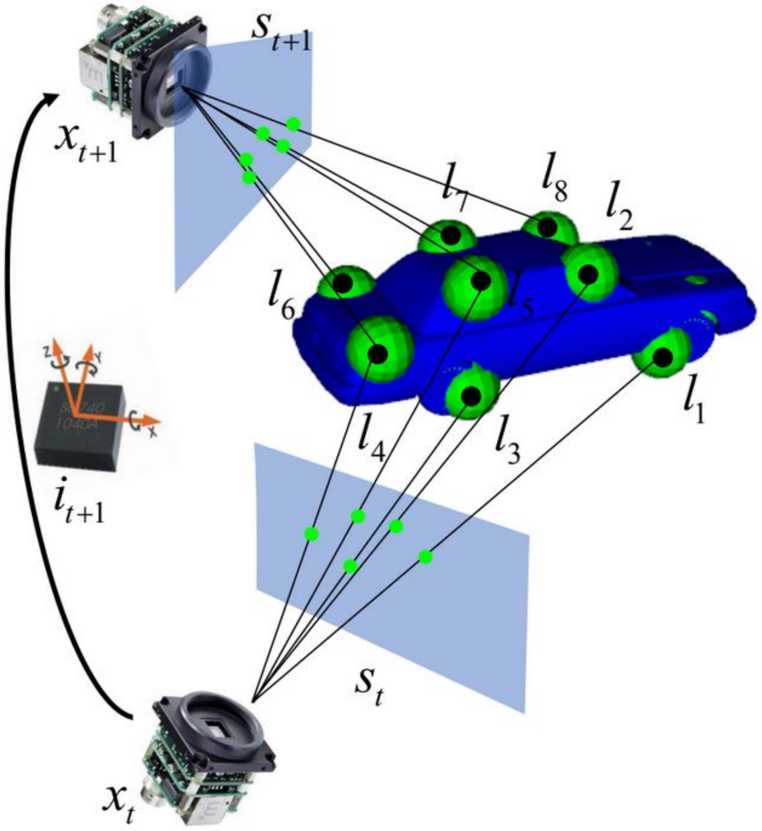
[1] StarMap for Category-Agnostic Keypoint and Viewpoint Estimation

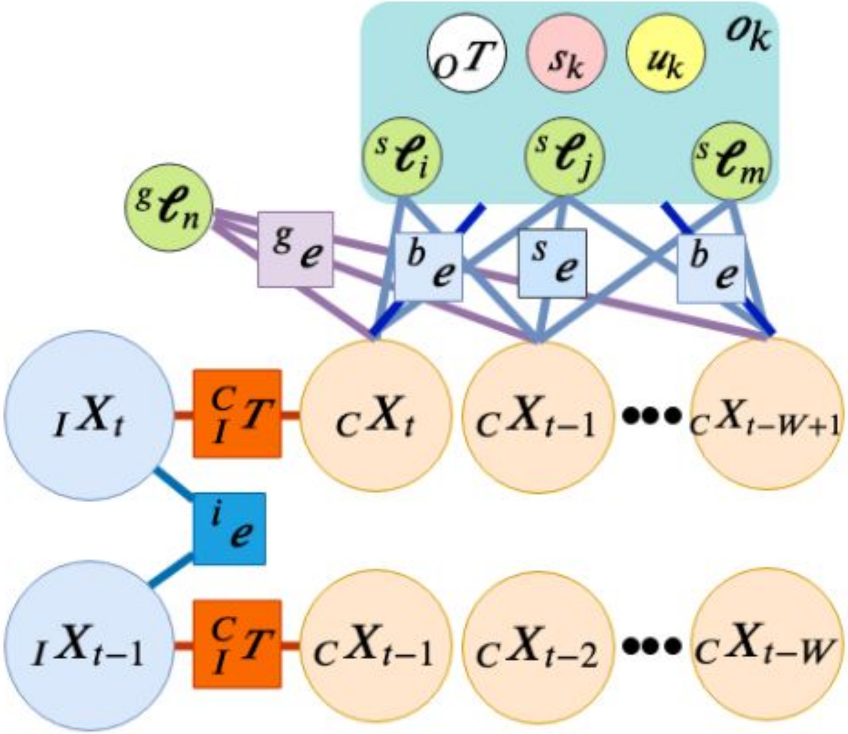
Front End

- Kalman filter tracks semantic keypoints on an object level

[1] OrcVIO: Object residual constrained Visual-Inertial Odometry

Back End



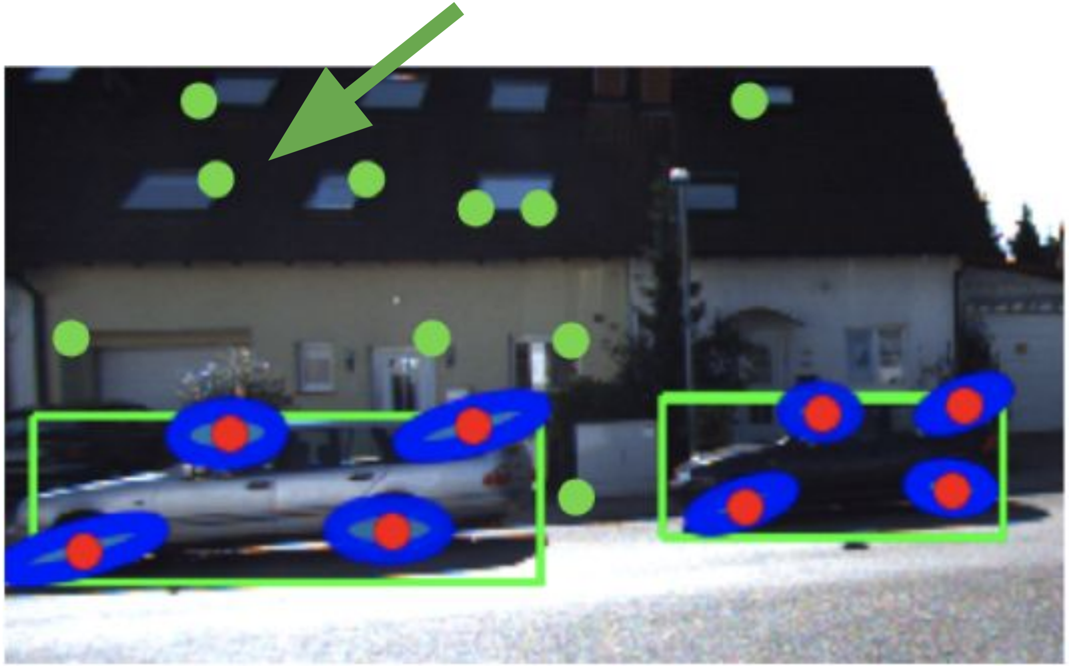


[1] OrcVIO: Object residual constrained Visual-Inertial Odometry

Residual functions

- **Geometric keypoint error**: an observed geometric keypoint should be equal to the image plane projection of its corresponding 3D landmark

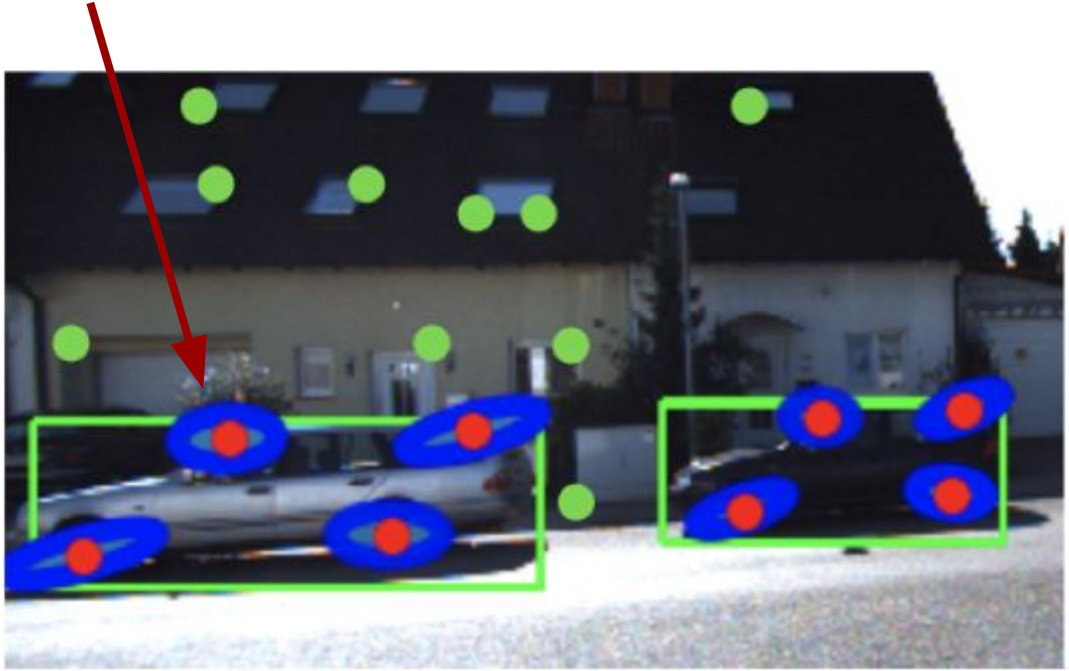
$$g_e(x_t, \ell, g_Z) = P\pi(c_t^{-1}\ell) - g_Z$$



Residual functions

- **Semantic keypoint error**: an observed semantic keypoint should be equal to the image plane projection of its corresponding semantic landmark

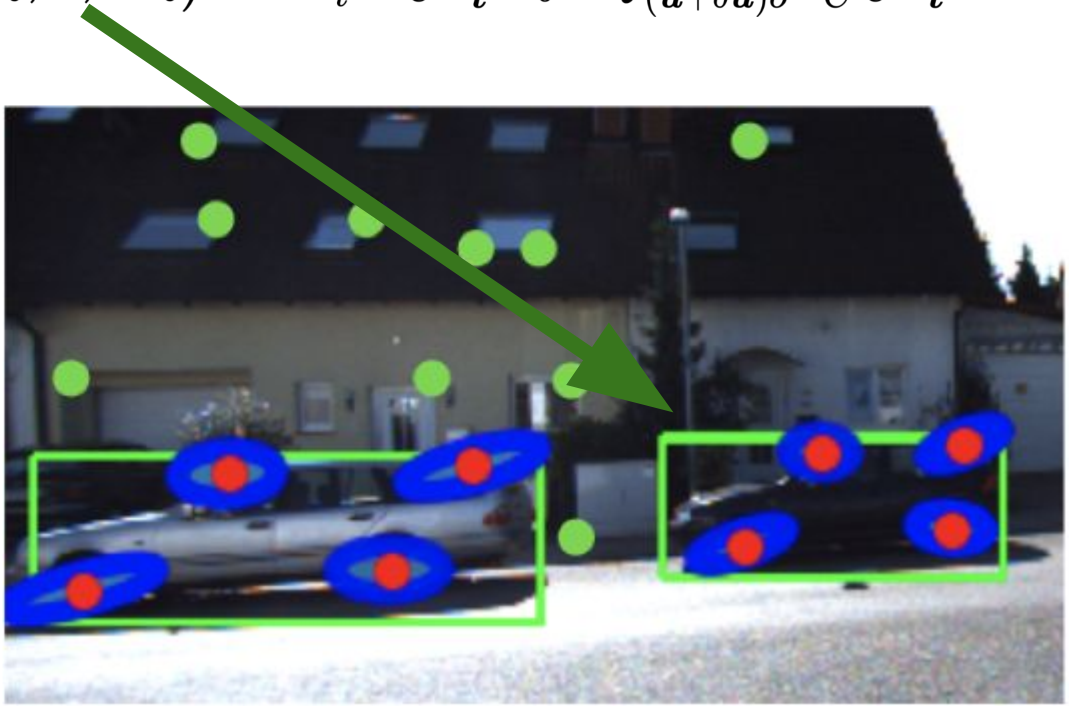
$$^{s}\boldsymbol{e}(\boldsymbol{x}_t, \boldsymbol{o}, {}^{s}\boldsymbol{Z}_{t,j}) = \boldsymbol{P}\boldsymbol{\pi}\left({}_{c}\boldsymbol{T}_{\boldsymbol{t}}^{-1}\,{}_{0}\boldsymbol{T}(\boldsymbol{s}_j + \delta\boldsymbol{s}_j)\right) - {}^{s}\boldsymbol{Z}_{\boldsymbol{t},\boldsymbol{j}}$$



Residual functions

- **Bounding box error**: the bounding box lines should be tangent to the conic projection of the object quadric surface

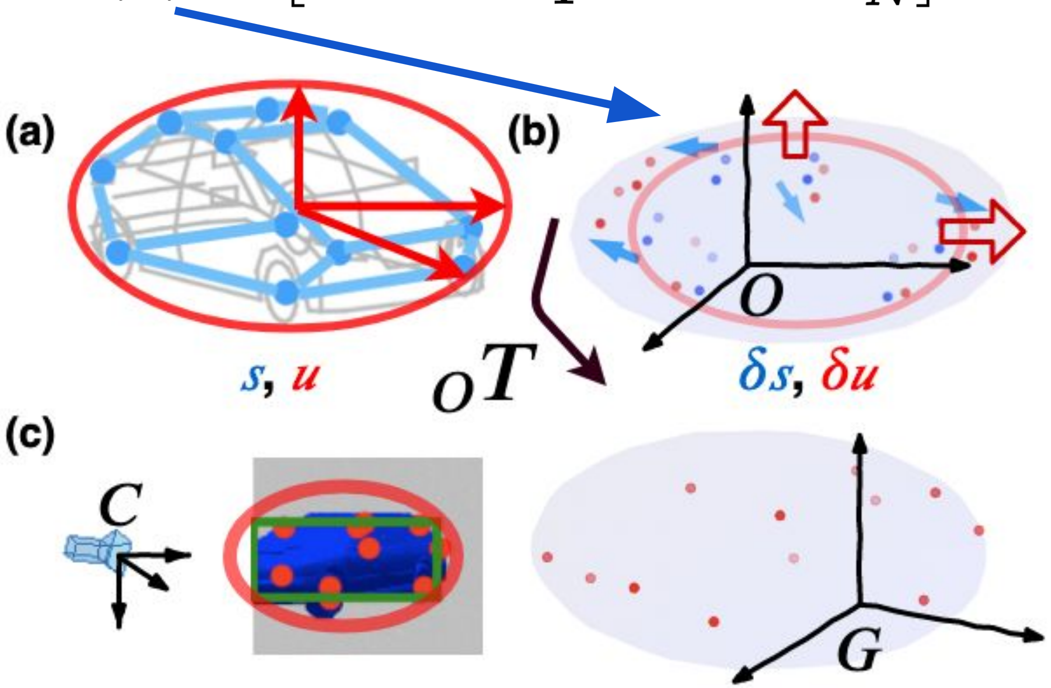
$$ {}^{b}\boldsymbol{e}\left(\boldsymbol{x}_{\boldsymbol{t}}, \boldsymbol{o}, {}^{b}\boldsymbol{Z}_{t}\right) = {}^{b}\boldsymbol{Z}_{t}^{T} \boldsymbol{P}_{C} \boldsymbol{T}_{\boldsymbol{t}}^{-\mathbf{1}}\, {}_{0}\boldsymbol{T} \boldsymbol{Q}^{*}_{(\boldsymbol{u}+\delta\boldsymbol{u})o} \boldsymbol{T}_{C}^{T}\, {}_{C}\boldsymbol{T}_{\boldsymbol{t}}^{-T} \boldsymbol{P}^{T_b} \boldsymbol{Z}_{t} $$



Residual functions

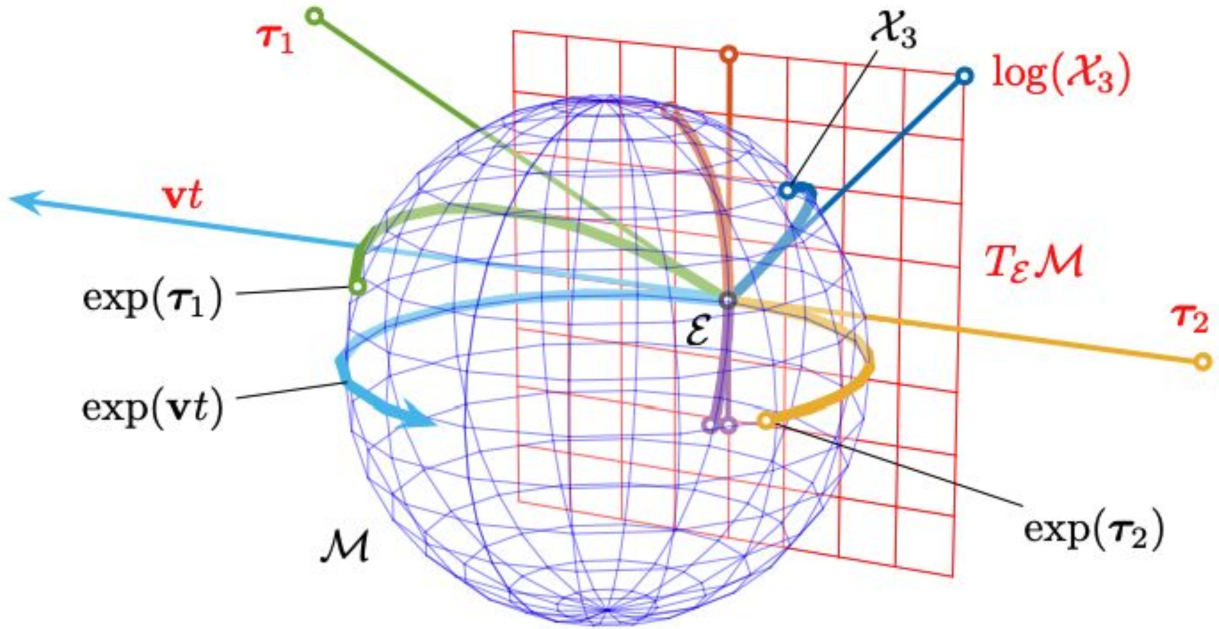
- **Object shape regularization**: penalize the deviation of the reconstructed shape from the average class shape

$$^{r}\boldsymbol{e}(\boldsymbol{o}) = \begin{bmatrix} \delta \boldsymbol{u}^{\boldsymbol{T}} & \delta \boldsymbol{s}_{\boldsymbol{1}}^{\boldsymbol{T}} & \cdots & \delta \boldsymbol{s}_{\boldsymbol{N}}^{\boldsymbol{T}} \end{bmatrix}$$



Lie Group

- Poses have a manifold structure, need to derive the Jacobians in the tangent space



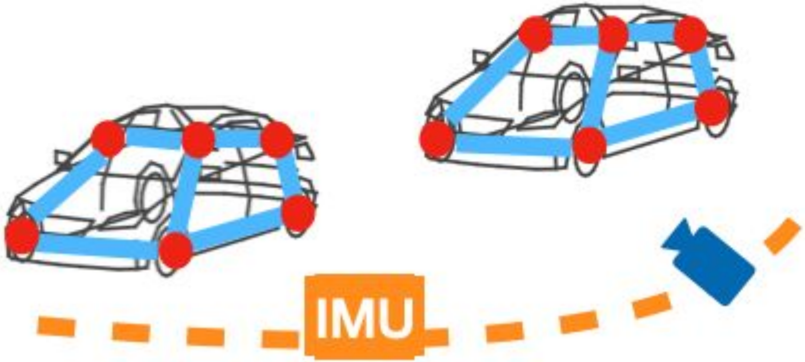
Semantic keypoint residual

$$\underline{\mathbf{x}}^{\odot} \triangleq \begin{bmatrix} \mathbf{I}_3 & -\mathbf{x}_\times \\ \mathbf{0}^\top & \mathbf{0}^\top \end{bmatrix} \in \mathbb{R}^{4\times 6}$$

$$\frac{\partial^s \mathbf{e}}{\partial_C \boldsymbol{\xi}_t} = -\frac{\partial^s \mathbf{e}}{\partial_O \boldsymbol{\xi}} \in \mathbb{R}^{2\times 6}$$

$$\frac{\partial^s \mathbf{e}}{\partial_O \boldsymbol{\xi}} = \mathbf{P}\frac{d\pi}{d\underline{\mathbf{s}}}\left({}_C\hat{\mathbf{T}}_t^{-1}\,{}_O\hat{\mathbf{T}}\left(\underline{\mathbf{s}}_j + \delta\hat{\underline{\mathbf{s}}}\right)_j\right){}_C\hat{\mathbf{T}}_t^{-1}\left[{}_O\hat{\mathbf{T}}\left(\underline{\mathbf{s}}_j + \delta\hat{\underline{\mathbf{s}}}\right)_j\right]^{\odot}$$

$$\frac{\partial^s \mathbf{e}}{\partial \delta\tilde{\mathbf{s}}_j} = \mathbf{P}\frac{d\pi}{d\underline{\mathbf{s}}}\left({}_C\hat{\mathbf{T}}_t^{-1}\,{}_O\hat{\mathbf{T}}\left(\underline{\mathbf{s}}_j + \delta\hat{\underline{\mathbf{s}}}\right)_j\right){}_C\hat{\mathbf{T}}_t^{-1}\,{}_O\hat{\mathbf{T}}\begin{bmatrix} \mathbf{I}_3 \\ \mathbf{0}^\top \end{bmatrix} \in \mathbb{R}^{2\times 3}.$$



Semantic bounding box residual

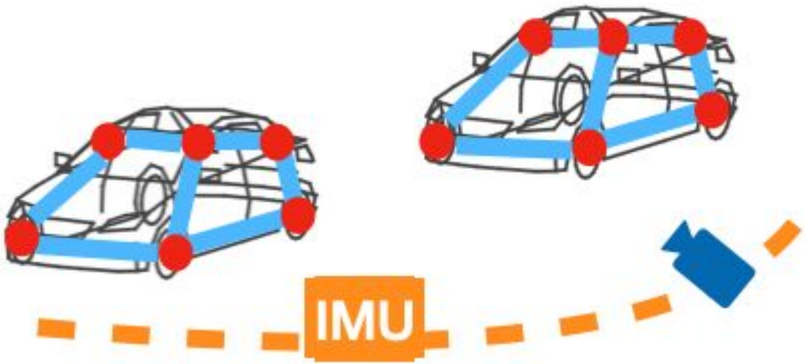
$$\mathbf{x}^{\circledcirc} \triangleq \begin{bmatrix} \mathbf{0} & \mathbf{x} \\ -\mathbf{x}_{\times} & \mathbf{0} \end{bmatrix} \in \mathbb{R}^{6\times 4}$$

$$\frac{\partial^b \mathbf{e}}{\partial_C \boldsymbol{\xi}_t} = -\frac{\partial^b \mathbf{e}}{\partial_O \boldsymbol{\xi}} \in \mathbb{R}^{1\times 6}$$

$$\frac{\partial^b \mathbf{e}}{\partial_O \boldsymbol{\xi}} = 2^b\underline{\mathbf{z}}^\top \mathbf{P}_C\hat{\mathbf{T}}_t^{-1}{}_O\hat{\mathbf{T}}\hat{\mathbf{Q}}^*_{(\mathbf{u}+\delta\hat{\mathbf{u}})}{}_O\hat{\mathbf{T}}^\top \left[{}_C\hat{\mathbf{T}}_t^{-\top}\mathbf{P}^{\top b}\underline{\mathbf{z}}\right]^{\circledcirc\top}$$

$$\frac{\partial^b \mathbf{e}}{\partial \delta\tilde{\mathbf{u}}} = (2(\mathbf{u}+\delta\hat{\mathbf{u}})\odot\mathbf{y}\odot\mathbf{y})^\top \in \mathbb{R}^{1\times 3}$$

$$\mathbf{y} \triangleq [\mathbf{I}_3 \quad \mathbf{0}]\,{}_O\hat{\mathbf{T}}^\top{}_C\hat{\mathbf{T}}_t^{-\top}\mathbf{P}^{\top b}\underline{\mathbf{z}}.$$



Visual-inertial Odometry

- Filtering based multi-state constraint Kalman filter (MSCKF):
    - Batch optimization over object/landmark when track is lost
    - Null-space trick: the optimized object/landmark state is used for a Kalman filter update to the sensor pose but is not retained in the filter state

Algorithm 1 Multi-State Constraint Filter

Propagation: For each IMU measurement received, propagate the filter state and covariance (cf. Section III-B).

Image registration: Every time a new image is recorded,

- augment the state and covariance matrix with a copy of the current camera pose estimate (cf. Section III-C).
- image processing module begins operation.

Update: When the feature measurements of a given image become available, perform an EKF update (cf. Sections III-D and III-E).

[1] A Multi-State Constraint Kalman Filter for Vision-aided Inertial Navigation

Visual-inertial Odometry

- Split the IMU dynamics into deterministic nominal and stochastic error dynamics via the perturbations
- **Nominal dynamics**: integrate in closed-form (assuming constant input) to obtain predicted mean

$$
{}_I\dot{\hat{\mathbf{R}}} = {}_I\hat{\mathbf{R}}\left({}^i\boldsymbol{\omega} - \hat{\mathbf{b}}_g\right)_\times, \qquad \dot{\hat{\mathbf{b}}}_g = \mathbf{0}, \qquad \dot{\hat{\mathbf{b}}}_a = \mathbf{0},
$$

$$
{}_I\dot{\hat{\mathbf{v}}} = {}_I\hat{\mathbf{R}}\left({}^i\mathbf{a} - \hat{\mathbf{b}}_a\right) + \mathbf{g}, \qquad {}_I\dot{\hat{\mathbf{p}}} = {}_I\hat{\mathbf{v}},
$$

- **Stochastic error dynamics**: integrate to obtain covariance

$$
{}_I\dot{\boldsymbol{\theta}} = -\left({}^i\boldsymbol{\omega} - \hat{\mathbf{b}}_g\right)_\times {}_I\boldsymbol{\theta} - \left(\tilde{\mathbf{b}}_g + \mathbf{n}_{\boldsymbol{\omega}}\right),
$$

$$
{}_I\dot{\tilde{\mathbf{v}}} = -{}_I\hat{\mathbf{R}}\left({}^i\mathbf{a} - \hat{\mathbf{b}}_a\right)_\times {}_I\boldsymbol{\theta} - {}_I\hat{\mathbf{R}}\left(\tilde{\mathbf{b}}_a + \mathbf{n}_{\mathbf{a}}\right),
$$

$$
{}_I\dot{\tilde{\mathbf{p}}} = {}_I\tilde{\mathbf{v}}, \qquad \dot{\tilde{\mathbf{b}}}_g = \mathbf{n}_g, \qquad \dot{\tilde{\mathbf{b}}}_a = \mathbf{n}_a.
$$

Prediction Step

- **Closed-form solutions** to the linear time invariant (LTI) ordinary differential equations (ODEs) from nominal dynamics

$$
\begin{aligned}
{}_I\mathbf{R} &= {}_I\hat{\mathbf{R}} \exp\left({}_I\boldsymbol{\theta}_\times\right) & {}_I\mathbf{p} &= {}_I\tilde{\mathbf{p}} + {}_I\hat{\mathbf{p}} & {}_I\mathbf{v} &= {}_I\tilde{\mathbf{v}} + {}_I\hat{\mathbf{v}} \\
{}_C\mathbf{T} &= {}_C\hat{\mathbf{T}} \exp\left({}_C\boldsymbol{\xi}_\times\right) & \mathbf{b}_g &= \tilde{\mathbf{b}}_g + \hat{\mathbf{b}}_g & \mathbf{b}_a &= \tilde{\mathbf{b}}_a + \hat{\mathbf{b}}_a \\
{}_O\mathbf{T} &= {}_O\hat{\mathbf{T}} \exp\left({}_O\boldsymbol{\xi}_\times\right) & \delta\mathbf{s} &= \delta\tilde{\mathbf{s}} + \delta\hat{\mathbf{s}} & \delta\mathbf{u} &= \delta\tilde{\mathbf{u}} + \delta\hat{\mathbf{u}},
\end{aligned}
$$

$$
\begin{aligned}
{}_I\dot{\hat{\mathbf{R}}} &= {}_I\hat{\mathbf{R}}\left({}^i\boldsymbol{\omega} - \hat{\mathbf{b}}_g\right)_\times, & \dot{\hat{\mathbf{b}}}_g &= \mathbf{0}, & \dot{\hat{\mathbf{b}}}_a &= \mathbf{0}, \\
{}_I\dot{\hat{\mathbf{v}}} &= {}_I\hat{\mathbf{R}}\left({}^i\mathbf{a} - \hat{\mathbf{b}}_a\right) + \mathbf{g}, & {}_I\dot{\hat{\mathbf{p}}} &= {}_I\hat{\mathbf{v}},
\end{aligned}
$$

Proposition 4. The nominal dynamics (23) can be integrated in *closed-form* to obtain the predicted mean $\hat{\mathbf{x}}^p_{k+1}$:

$$
\begin{aligned}
{}_I\hat{\mathbf{R}}^p_{k+1} &= {}_I\hat{\mathbf{R}}_k \exp\left(\tau_k\left({}^i\boldsymbol{\omega}_k - \hat{\mathbf{b}}_{g,k}\right)_\times\right), \\
{}_I\hat{\mathbf{v}}^p_{k+1} &= {}_I\hat{\mathbf{v}}_k + \mathbf{g}\tau_k + {}_I\hat{\mathbf{R}}_k\mathbf{J}_L\left(\tau_k\left({}^i\boldsymbol{\omega}_k - \hat{\mathbf{b}}_{g,k}\right)\right)\left({}^i\mathbf{a}_k - \hat{\mathbf{b}}_{a,k}\right)\tau_k, \\
{}_I\hat{\mathbf{p}}^p_{k+1} &= {}_I\hat{\mathbf{p}}_k + {}_I\hat{\mathbf{v}}_k\tau_k + \mathbf{g}\frac{\tau_k^2}{2} + {}_I\hat{\mathbf{R}}_k\mathbf{H}_L\left(\tau_k\left({}^i\boldsymbol{\omega}_k - \hat{\mathbf{b}}_{g,k}\right)\right)\left({}^i\mathbf{a}_k - \hat{\mathbf{b}}_{a,k}\right)\tau_k^2, \\
\hat{\mathbf{b}}^p_{g,k+1} &= \hat{\mathbf{b}}_{g,k}, \qquad \hat{\mathbf{b}}^p_{a,k+1} = \hat{\mathbf{b}}_{a,k}, \qquad\qquad (25) \\
{}_I\hat{\mathbf{T}}^p_k &= \begin{bmatrix} {}_I\hat{\mathbf{R}}_k & {}_I\hat{\mathbf{p}}_k \\ \mathbf{0}^\top & 1 \end{bmatrix}, {}_I\hat{\mathbf{T}}^p_{k-1} = {}_I\hat{\mathbf{T}}_{k-1}, \ldots, {}_I\hat{\mathbf{T}}^p_{k-W+1} = {}_I\hat{\mathbf{T}}_{k-W+1},
\end{aligned}
$$

where $\mathbf{J}_L(\boldsymbol{\omega}) \triangleq \mathbf{I}_3 + \frac{\boldsymbol{\omega}_\times}{2!} + \frac{\boldsymbol{\omega}^2_\times}{3!} + \ldots$ is the left Jacobian of $SO(3)$ and $\mathbf{H}_L(\boldsymbol{\omega}) \triangleq \frac{\mathbf{I}_3}{2!} + \frac{\boldsymbol{\omega}_\times}{3!} + \frac{\boldsymbol{\omega}^2_\times}{4!} + \ldots$. Both $\mathbf{J}_L(\boldsymbol{\omega})$ and $\mathbf{H}_L(\boldsymbol{\omega})$ admit closed-form (Rodrigues) expressions:

$$
\mathbf{J}_L(\boldsymbol{\omega}) = \mathbf{I}_3 + \frac{1 - \cos\|\boldsymbol{\omega}\|}{\|\boldsymbol{\omega}\|^2}\boldsymbol{\omega}_\times + \frac{\|\boldsymbol{\omega}\| - \sin\|\boldsymbol{\omega}\|}{\|\boldsymbol{\omega}\|^3}\boldsymbol{\omega}^2_\times \qquad (26)
$$

$$
\mathbf{H}_L(\boldsymbol{\omega}) = \frac{1}{2}\mathbf{I}_3 + \frac{\|\boldsymbol{\omega}\| - \sin\|\boldsymbol{\omega}\|}{\|\boldsymbol{\omega}\|^3}\boldsymbol{\omega}_\times + \frac{2(\cos\|\boldsymbol{\omega}\| - 1) + \|\boldsymbol{\omega}\|^2}{2\|\boldsymbol{\omega}\|^4}\boldsymbol{\omega}^2_\times.
$$

Prediction Step

- **Closed-form solutions** to the **linear time variant (LTV)** stochastic differential equation (SDE) from stochastic error dynamics

$$
\begin{aligned}
{}_I\dot{\boldsymbol{\theta}} &= -\left({}^i\boldsymbol{\omega} - \hat{\mathbf{b}}_g\right)_\times {}_I\boldsymbol{\theta} - \left(\tilde{\mathbf{b}}_g + \mathbf{n}_\omega\right), \\
{}_I\dot{\tilde{\mathbf{v}}} &= -{}_I\hat{\mathbf{R}}\left({}^i\mathbf{a} - \hat{\mathbf{b}}_a\right)_\times {}_I\boldsymbol{\theta} - {}_I\hat{\mathbf{R}}\left(\tilde{\mathbf{b}}_a + \mathbf{n}_\mathbf{a}\right), \\
{}_I\dot{\tilde{\mathbf{p}}} &= {}_I\tilde{\mathbf{v}}, \qquad \dot{\tilde{\mathbf{b}}}_g = \mathbf{n}_g, \qquad \dot{\tilde{\mathbf{b}}}_a = \mathbf{n}_a.
\end{aligned}
$$

$$
{}_I\dot{\tilde{\mathbf{x}}} = \mathbf{F}(t)\,{}_I\tilde{\mathbf{x}} + {}_I\mathbf{n}, \qquad {}_I\tilde{\mathbf{x}}(0) \sim \mathcal{N}(\mathbf{0}, {}_I\boldsymbol{\Sigma}_k) \qquad (27)
$$

$$
\begin{aligned}
{}_I\boldsymbol{\Sigma}^p_{k+1} &= \mathbb{E}\left[{}_I\tilde{\mathbf{x}}(\tau_k)\,{}_I\tilde{\mathbf{x}}(\tau_k)^\top\right] \qquad (28) \\
&= \boldsymbol{\Phi}(\tau_k,0)\,{}_I\boldsymbol{\Sigma}_k\boldsymbol{\Phi}(\tau_k,0)^\top + \int_0^{\tau_k}\boldsymbol{\Phi}(\tau_k,s)\mathbf{Q}\boldsymbol{\Phi}(\tau_k,s)^\top ds
\end{aligned}
$$

Proposition 5. The LTV SDE in (27) has a *closed-form* transition matrix:

$$
\boldsymbol{\Phi}(t,0) = \begin{bmatrix}
\exp\left(-t\boldsymbol{\omega}_\times\right) & \mathbf{0} & \mathbf{0} & -t\mathbf{J}_L\left(-t\boldsymbol{\omega}\right) & \mathbf{0} \\
\boldsymbol{\Phi}_{\mathbf{v}\boldsymbol{\theta}}(t) & \mathbf{I}_3 & \mathbf{0} & \boldsymbol{\Phi}_{\mathbf{v}\boldsymbol{\omega}}(t) & \boldsymbol{\Phi}_{\mathbf{va}}(t) \\
\boldsymbol{\Phi}_{\mathbf{p}\boldsymbol{\theta}}(t) & t\mathbf{I}_3 & \mathbf{I}_3 & \boldsymbol{\Phi}_{\mathbf{p}\boldsymbol{\omega}}(t) & \boldsymbol{\Phi}_{\mathbf{pa}}(t) \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}_3 & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}_3
\end{bmatrix}
$$

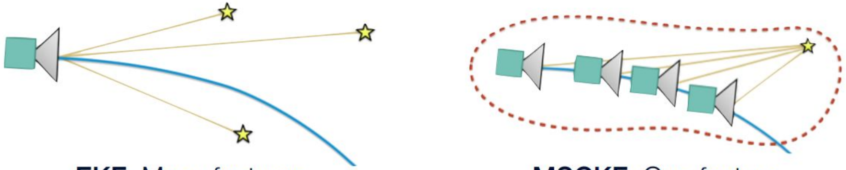
where $\mathbf{w} = {}^i\boldsymbol{\omega}_k - \hat{\mathbf{b}}_{g,k}$, $\mathbf{a} = {}^i\mathbf{a}_k - \hat{\mathbf{b}}_{a,k}$ and the blocks are:

$$
\begin{aligned}
\boldsymbol{\Phi}_{\mathbf{v}\boldsymbol{\theta}}(t) &= -t\,{}_I\hat{\mathbf{R}}_k\left[\mathbf{J}_L\left(t\boldsymbol{\omega}\right)\mathbf{a}\right]_\times \\
\boldsymbol{\Phi}_{\mathbf{v}\boldsymbol{\omega}}(t) &= {}_I\hat{\mathbf{R}}_k\Delta(t)\frac{\mathbf{a}_\times}{\|\boldsymbol{\omega}\|^2}\left(\mathbf{I}_3 + \frac{\boldsymbol{\omega}_\times^2}{\|\boldsymbol{\omega}\|^2}\right) \\
&\quad + t\,{}_I\hat{\mathbf{R}}_k\left(\frac{\boldsymbol{\omega}\mathbf{a}^\top}{\|\boldsymbol{\omega}\|^2}\left(\mathbf{J}_L(-t\boldsymbol{\omega}) - \mathbf{I}_3\right) + \frac{\mathbf{a}^\top\boldsymbol{\omega}}{\|\boldsymbol{\omega}\|^2}\left(\mathbf{J}_L(t\boldsymbol{\omega}) - \mathbf{I}_3\right)\right) \\
\boldsymbol{\Phi}_{\mathbf{va}}(t) &= -t\,{}_I\hat{\mathbf{R}}_k\mathbf{J}_L\left(t\boldsymbol{\omega}\right) \qquad (29) \\
\boldsymbol{\Phi}_{\mathbf{p}\boldsymbol{\theta}}(t) &= -t^2\,{}_I\hat{\mathbf{R}}_k\left[\mathbf{H}_L\left(t\boldsymbol{\omega}\right)\mathbf{a}\right]_\times \\
\boldsymbol{\Phi}_{\mathbf{p}\boldsymbol{\omega}}(t) &= {}_I\hat{\mathbf{R}}_k\left(t\mathbf{J}_L\left(t\boldsymbol{\omega}\right) - \frac{\boldsymbol{\omega}_\times}{\|\boldsymbol{\omega}\|^2}\Delta(t) - t\mathbf{I}_3\right)\frac{\mathbf{a}_\times}{\|\boldsymbol{\omega}\|^2}\left(\mathbf{I}_3 + \frac{\boldsymbol{\omega}_\times^2}{\|\boldsymbol{\omega}\|^2}\right) \\
&\quad + \frac{t^2}{2}\,{}_I\hat{\mathbf{R}}_k\left(\frac{\boldsymbol{\omega}\mathbf{a}^\top}{\|\boldsymbol{\omega}\|^2}\left(2\mathbf{H}_L(-t\boldsymbol{\omega}) - \mathbf{I}_3\right) + \frac{\mathbf{a}^\top\boldsymbol{\omega}}{\|\boldsymbol{\omega}\|^2}\left(2\mathbf{H}_L(t\boldsymbol{\omega}) - \mathbf{I}_3\right)\right) \\
\boldsymbol{\Phi}_{\mathbf{pa}}(t) &= -t^2\,{}_I\hat{\mathbf{R}}_k\mathbf{H}_L\left(t\boldsymbol{\omega}\right)
\end{aligned}
$$

Update Step

- When an object/landmark track is lost, optimize its state given the current sensor state

$$\min_{\ell_m} {}^g\boldsymbol{W} \sum_{t,n} \left\| {}^g\boldsymbol{e}(\boldsymbol{x_t}, \ell_{\boldsymbol{m}}, {}^g\boldsymbol{Z}_{t,n}) \right\|^2$$



EKF: Many features constrain one state.

MSCKF: One feature constrains many states.

$$\min_{o_i} \boldsymbol{s}_{\boldsymbol{W}} \sum_{t,j,\boldsymbol{k}} \left\| {}^s\boldsymbol{e}(\boldsymbol{x_t}, \boldsymbol{o_i}, {}^s\boldsymbol{Z}_{t,j,k}) \right\|^2 + {}^b\boldsymbol{w} \sum_{t,j,k} \left\| {}^b\boldsymbol{e}\left(\boldsymbol{x_t}, \boldsymbol{o_i}, {}^b\boldsymbol{Z}_{t,j,k}\right) \right\|^2 + {}^r\boldsymbol{w} \left\| {}^r\boldsymbol{e}(\boldsymbol{o_i}) \right\|^2$$

- Levenberg-Marquardt with error Jacobians obtained via object state perturbation:

$$_0\boldsymbol{T} = \exp\left({}_{\boldsymbol{o}}\zeta_{\mathrm{x}}\right){}_{\boldsymbol{o}}\widehat{\boldsymbol{T}} \qquad \boldsymbol{\delta s} = \widetilde{\boldsymbol{\delta s}} + \widehat{\boldsymbol{\delta s}} \qquad \boldsymbol{\delta u} = \widetilde{\boldsymbol{\delta u}} + \widehat{\boldsymbol{\delta u}}$$

- Eliminate object state from sensor state residual via left-nullspace matrix

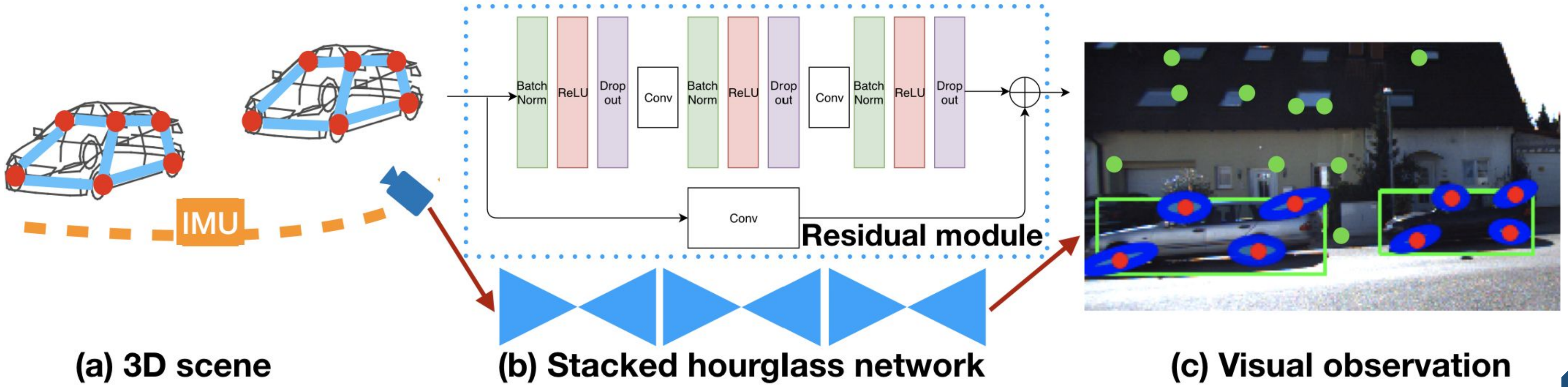
0

$$N_i^T e_i \approx N_i^T \hat{e}_i + N_i^T \frac{\partial \hat{e}_i}{\partial \tilde{x}_{t+1}} \tilde{x}_{t+1} + N_i^T \frac{\partial \hat{e}_i}{\partial \tilde{o}_i} \tilde{o}_i + N_i^T n_i \quad \Big\}$$

innovation for KF update

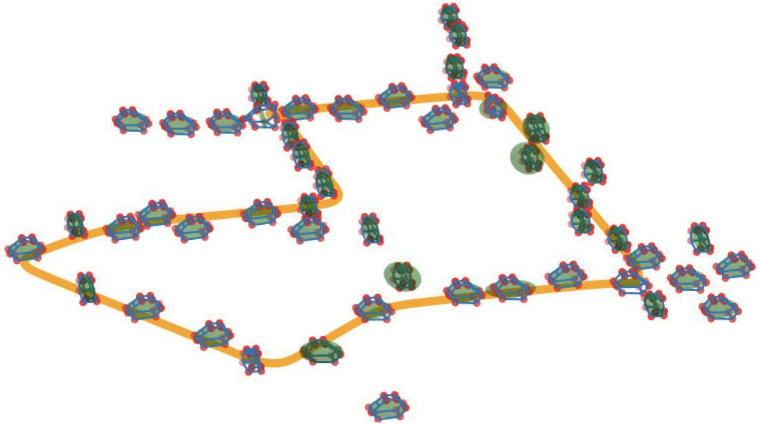
Contributions

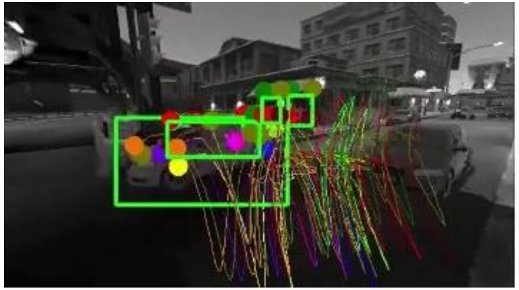
- We introduce **object states** in the formulation SLAM, with coarse ellipsoid shape, and fine semantic-keypoint shape
- We define **residuals** relating object states and IMU camera states to inertial measurements, geometric features, object semantic features, and object bounding-box detections
- We propose **closed-form** mean and covariance propagation over the SE(3) pose and velocity manifold of the IMU-camera states



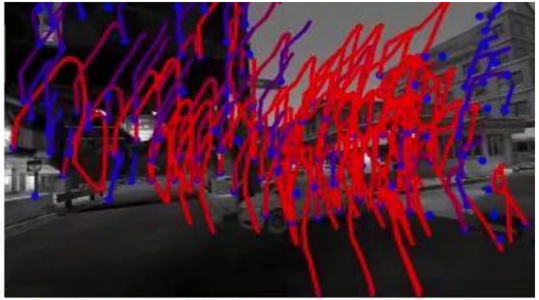
Evaluation

- Object-level map and reprojected object states on KITTI odom 07

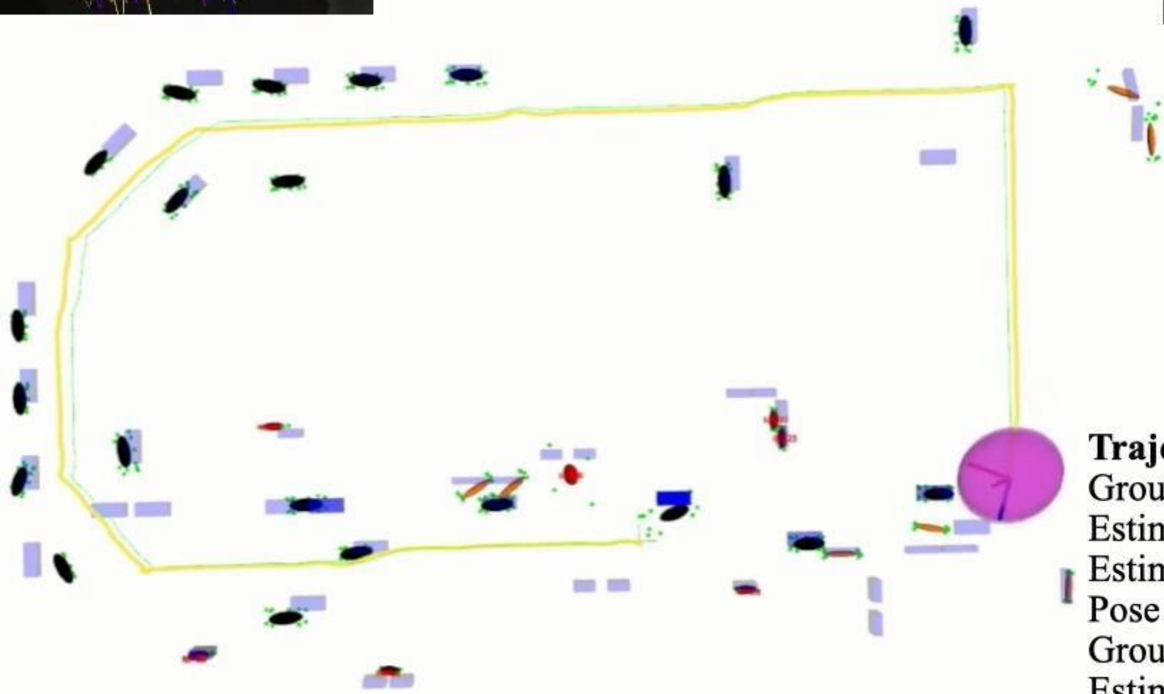




Semantic features
Bounding boxes are green
Semantic keypoints are colored dots
Semantic keypoint tracks are colored lines



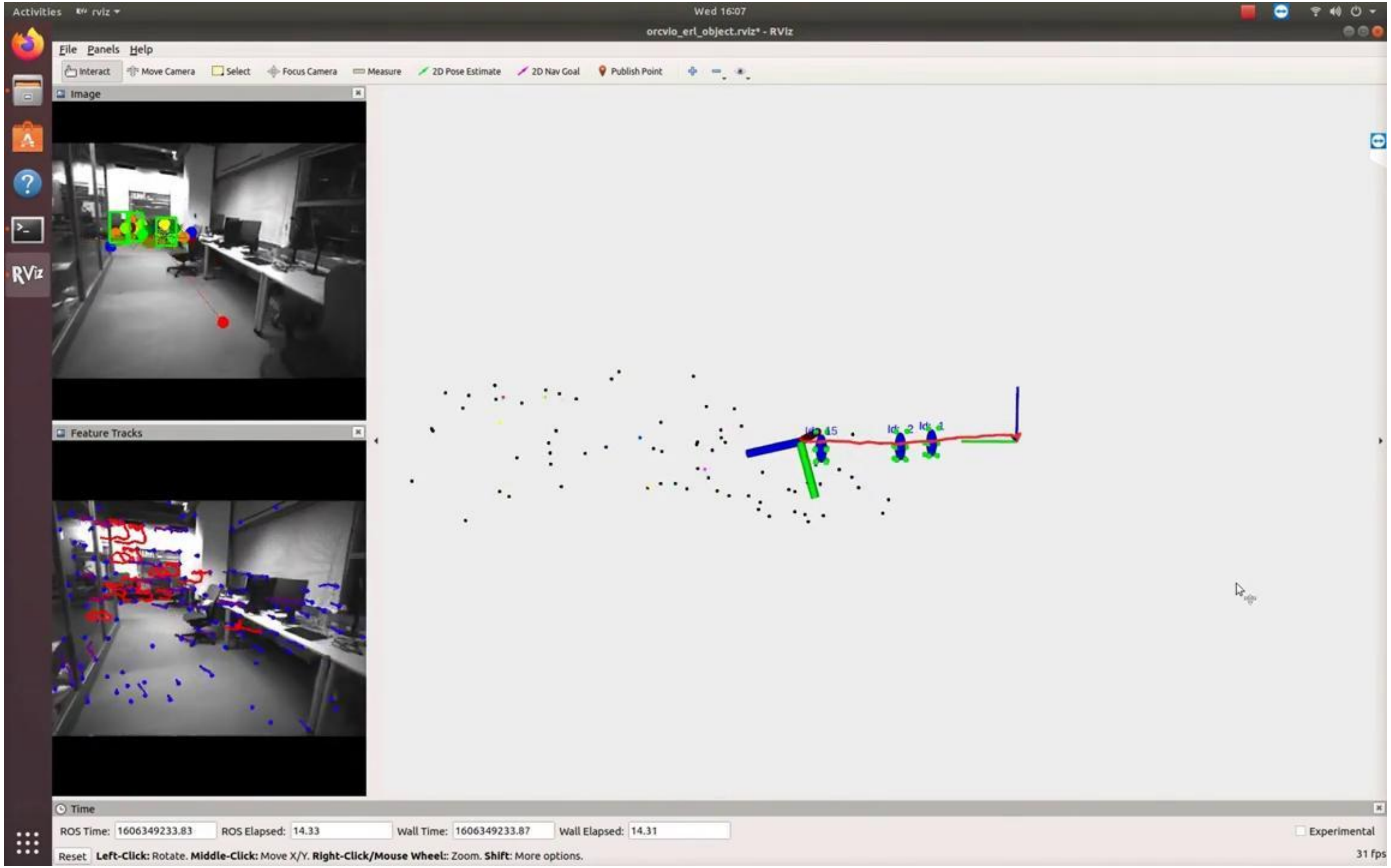
Geometric features
Geometric features are blue
Geometric feature tracks are red



Trajectory and object map
Groundtruth trajectory is the green line
Estimated trajectory is the yellow line
Estimated pose is the axes
Pose covariance is the purple ellipsoid
Groundtruth objects are blue meshes
Estimated objects are colored ellipsoids
Semantic keypoints are green dots

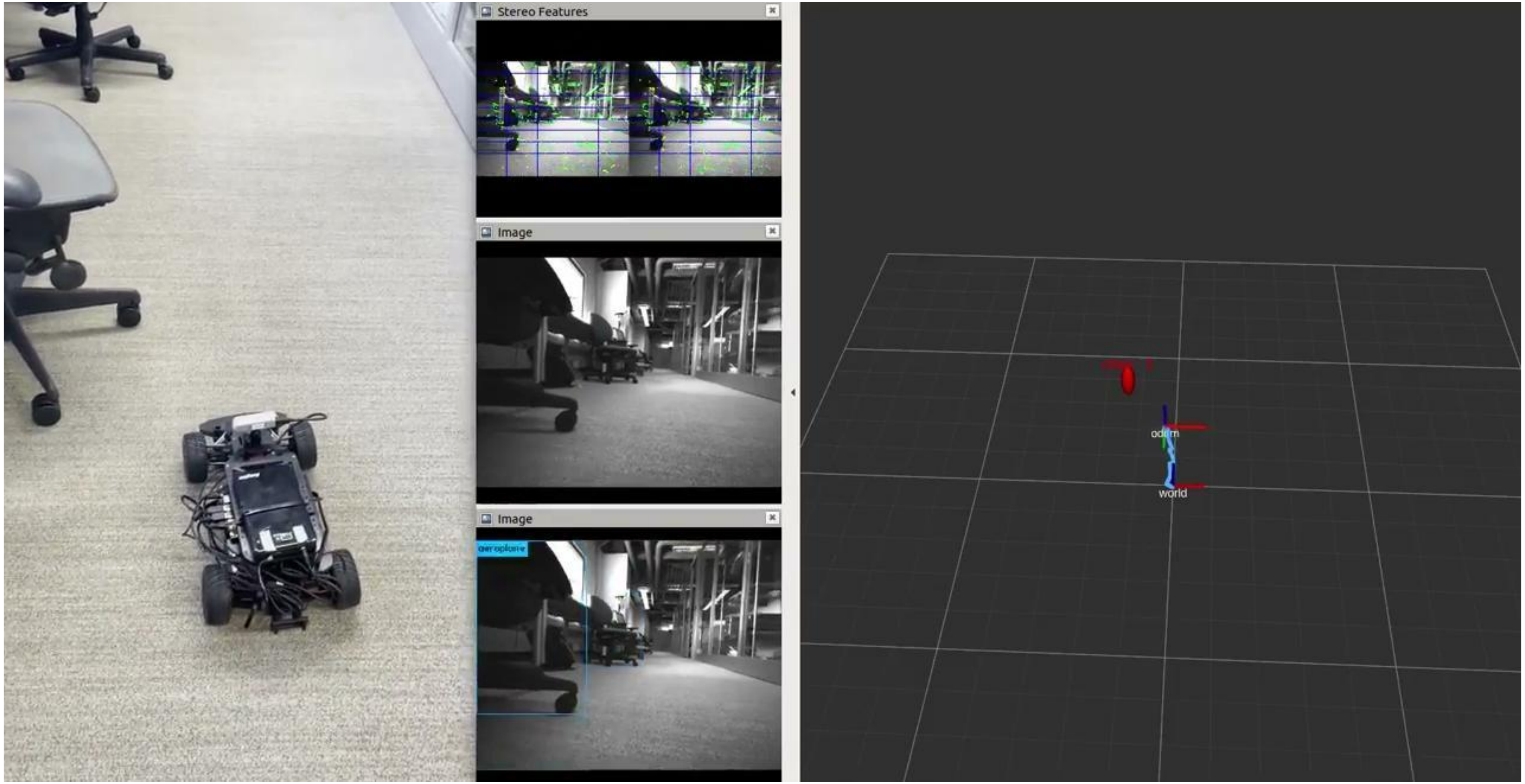
Evaluation

- Indoor scene with hand-held VI sensor to map chairs



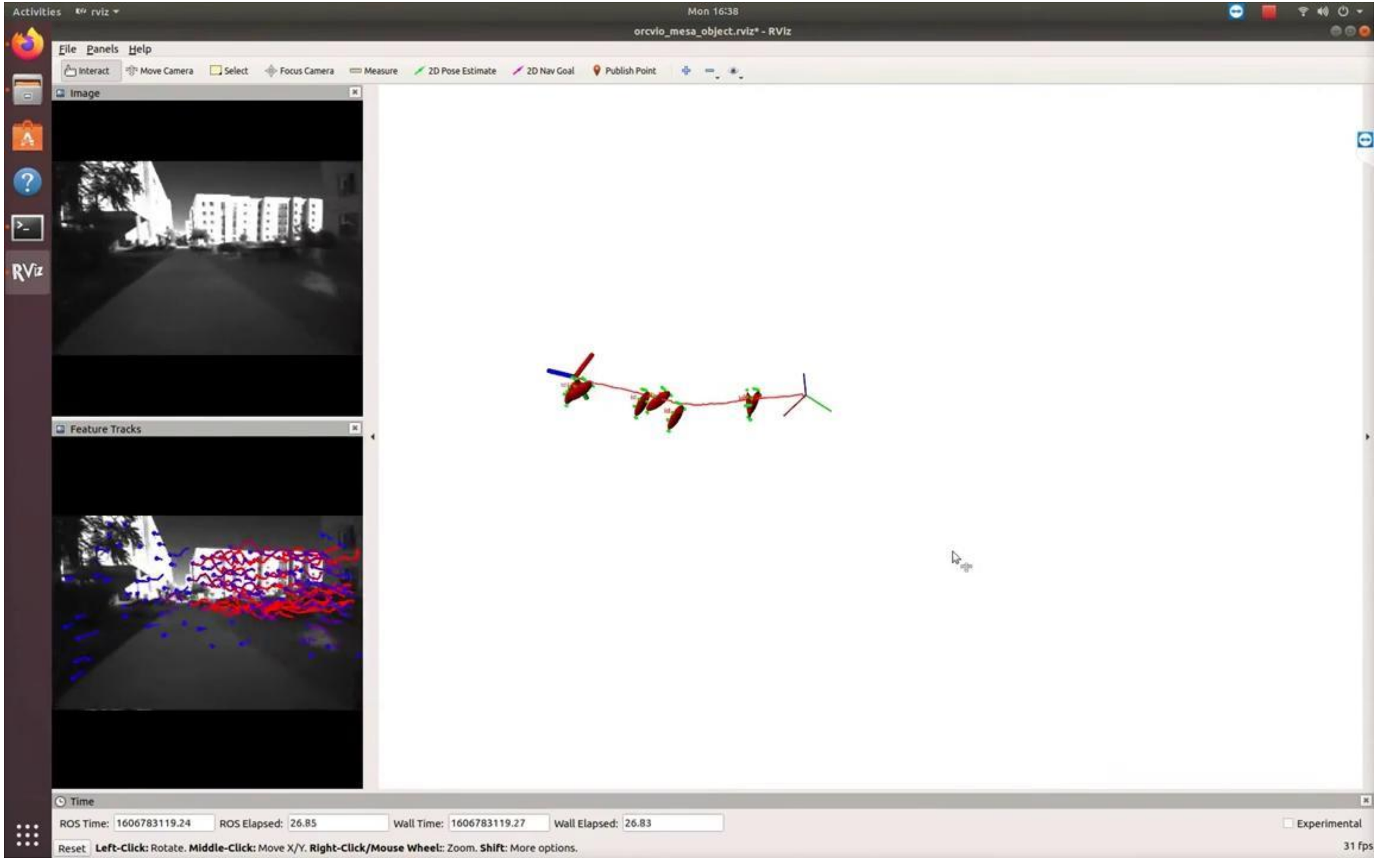
Evaluation

- Indoor scene with VI sensor on robotic car to map chairs



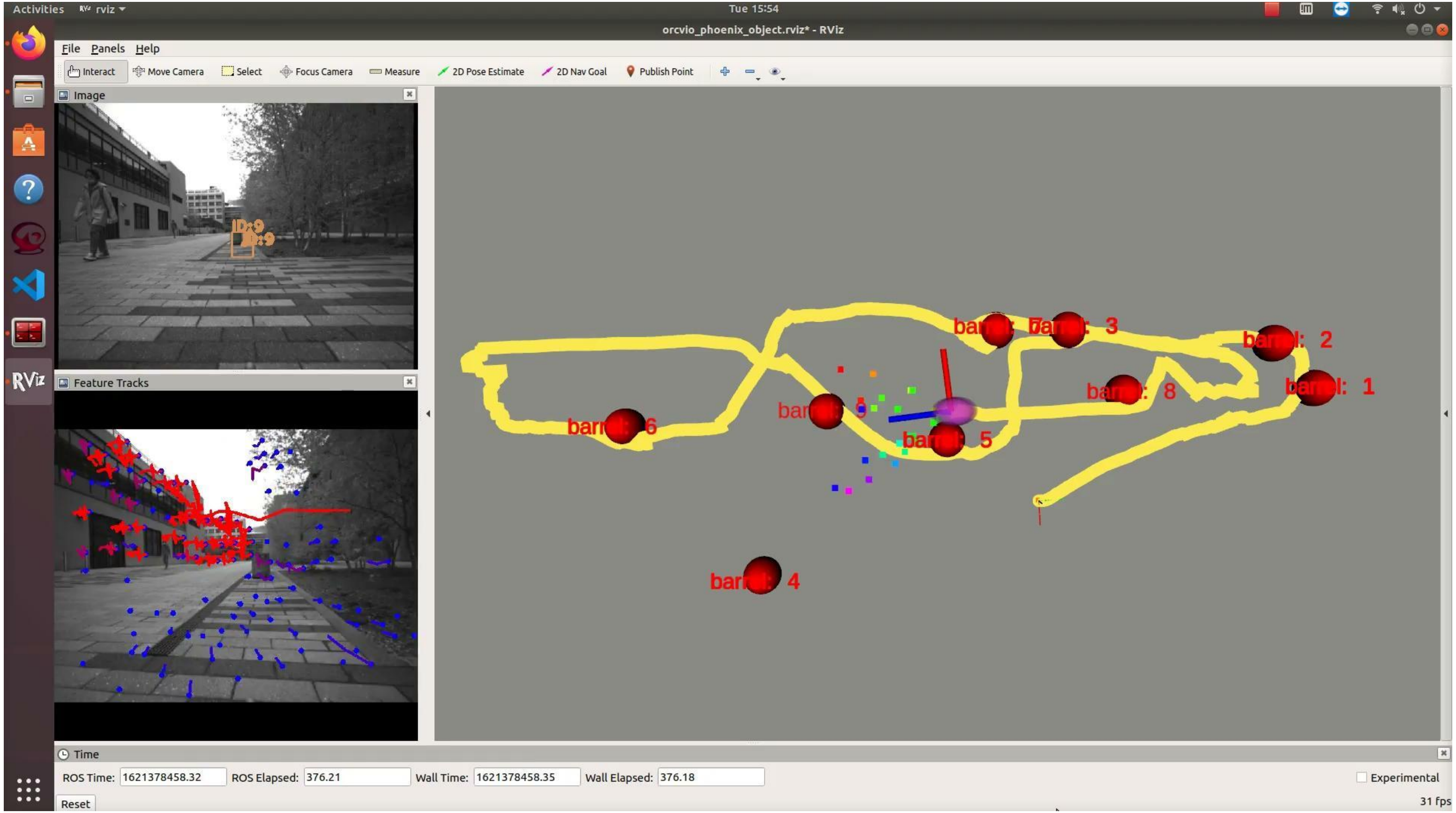
Evaluation

- Outdoor scene with hand-held VI sensor to map chairs, bikes, cars



Evaluation

- Outdoor scene with VI sensor on robotic car to map barrels



Evaluation

- Quantitative results comparable with SOTA

TABLE II: Object Detection and Pose Estimation on the KITTI Object Sequences

Metric	KITTI Sequence →	22	23	36	39	61	64	95	96	117	Mean
3D IoU	SingleView [90]	0.52	0.32	0.50	0.54	**0.54**	0.43	0.40	0.26	0.25	0.42
	CubeSLAM [61]	**0.58**	0.35	**0.54**	**0.59**	0.50	**0.48**	**0.52**	0.29	**0.35**	**0.47**
	OrcVIO	0.51	**0.55**	0.53	0.55	0.53	0.46	0.29	0.31	0.23	0.44
Trans. error (%)	CubeSLAM [61]	**1.68**	1.72	**2.93**	1.61	1.24	**0.93**	**1.49**	1.81	2.21	1.74
	OrcVIO	**1.68**	**1.50**	2.95	**1.44**	**1.22**	1.02	**1.49**	**1.59**	**1.92**	**1.65**

TABLE IV: Trajectory RMSE (m) on the KITTI Odometry Sequences

KITTI Sequence →	00	02	04	05	06	07	08	09	10	Mean
Object BA [58]	73.4	55.5	10.7	50.8	73.1	47.1	72.2	31.2	53.5	51.9
CubeSLAM [61]	13.9	26.2	1.1	**4.8**	7.0	2.7	**10.7**	10.7	8.4	9.5
OrcVIO	**10.9**	**18.9**	**0.8**	5.5	**4.5**	**2.5**	14.1	**6.6**	**5.3**	**7.7**

Evaluation

- Stereo VIO trajectory accuracy comparable with SOTA

Dataset	VINS	S-MSCKF	ORB SLAM	SVO2	Stereo OrcVIO
Sensor	Mono+IMU	Stereo+IMU	Stereo	Stereo	Stereo+IMU
MH_01_easy	0.156025	x	**0.037896**	0.111732	0.231
MH_02_easy	0.178418	0.152133	**0.044086**	x	0.416
MH_03_medium	0.194874	0.289593	**0.040688**	0.360784	0.279
MH_04_difficult	0.346300	0.210353	**0.088795**	2.891935	0.320
MH_05_difficult	0.302346	0.293128	**0.067401**	1.199866	0.453
V1_01_easy	0.088925	0.070955	0.087481	0.061025	**0.056**
V1_02_medium	0.110438	0.126732	0.079843	**0.081536**	0.168
V1_03_difficult	**0.187195**	0.203363	0.284315	0.248401	0.203
V2_01_easy	0.086263	**0.065962**	0.077287	0.076514	0.073
V2_02_medium	0.157444	0.175961	**0.117782**	0.204471	0.208
V2_03_difficult	**0.277569**	x	x	x	x

Open-sourced OrcVIO

	Python	C++	ROS support	Mapping	Requires	Note
OrcVIO		✓	✓	✓	**Mono imgs Bounding boxes Semantic kps**	**Original**
OrcVIO Lite		✓	✓	✓	**Mono imgs Bounding boxes**	**Simplified mapper**
OrcVIO Stereo	✓	✓	✓	**External mapper**	**Stereo imgs Bounding boxes**	**More robust VIO**
External mapper					**Mono imgs Bounding boxes Camera poses**	**Compatible with all OrcVIO**

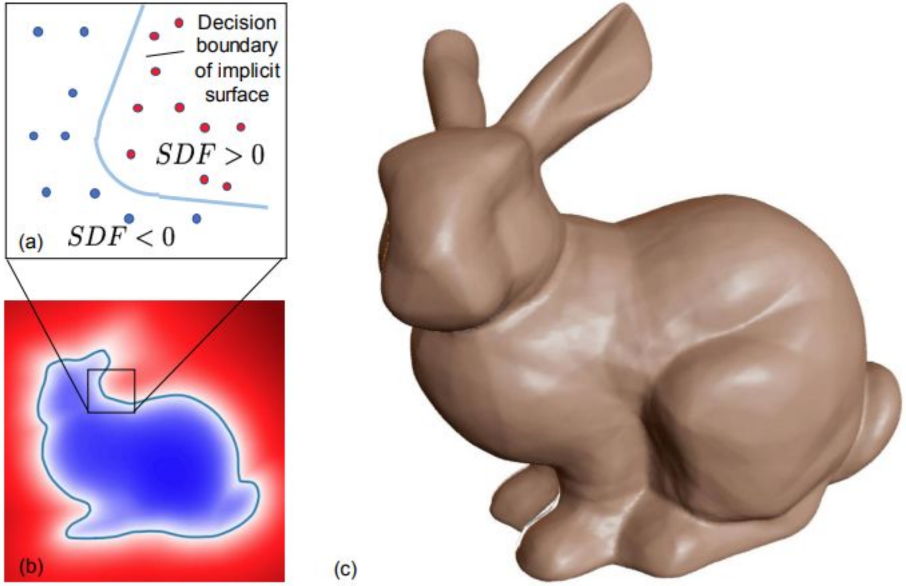
https://github.com/shanmo?tab=repositories

- The surface can be implicitly represented by the zero-level set

The fine shape of a rigid body is represented as $\{\mathbf{x} \in \mathbb{R}^3 \mid f(\mathbf{x}) \leq 0\}$ using the *signed distance field* of a set $\mathcal{S} \subset \mathbb{R}^3$:

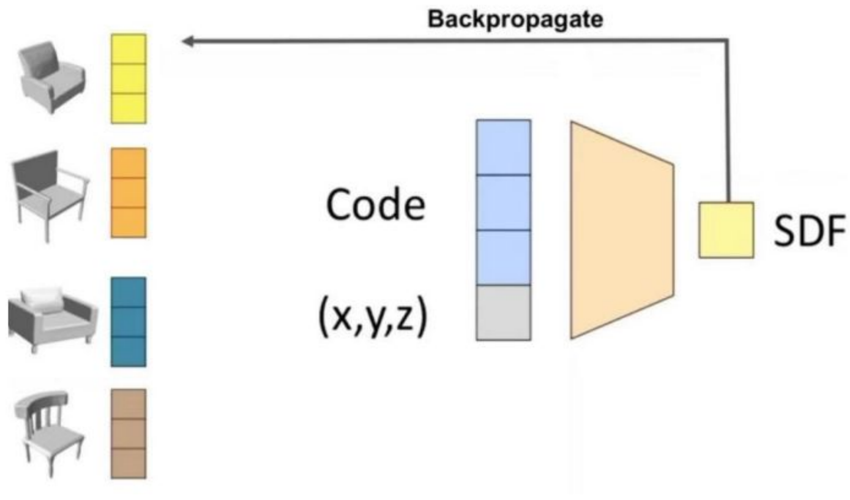
$$f(\mathbf{x}) = \begin{cases} -d(\mathbf{x}, \partial\mathcal{S}), & \mathbf{x} \in \mathcal{S}, \\ d(\mathbf{x}, \partial\mathcal{S}), & \mathbf{x} \notin \mathcal{S}, \end{cases} \tag{4}$$

where $d(\mathbf{x}, \partial\mathcal{S})$ denotes the Euclidean distance from a point $\mathbf{x} \in \mathbb{R}^3$ to the boundary $\partial\mathcal{S}$ of $\mathcal{S}$.



Review

- DeepSDF directly regresses SDF
- Latent vectors are optimized along with the decoder weights through standard backpropagation
- During inference, decoder weights are fixed, and an optimal latent vector is estimated



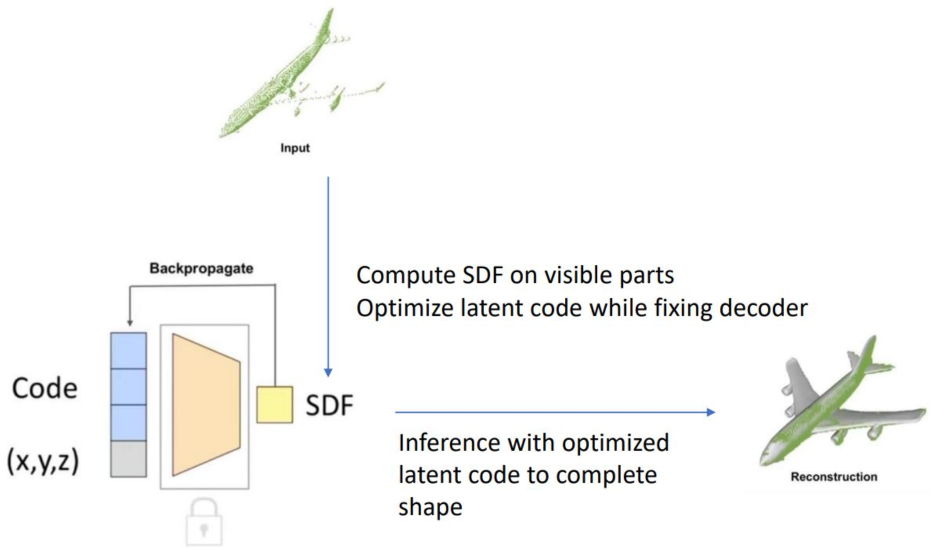
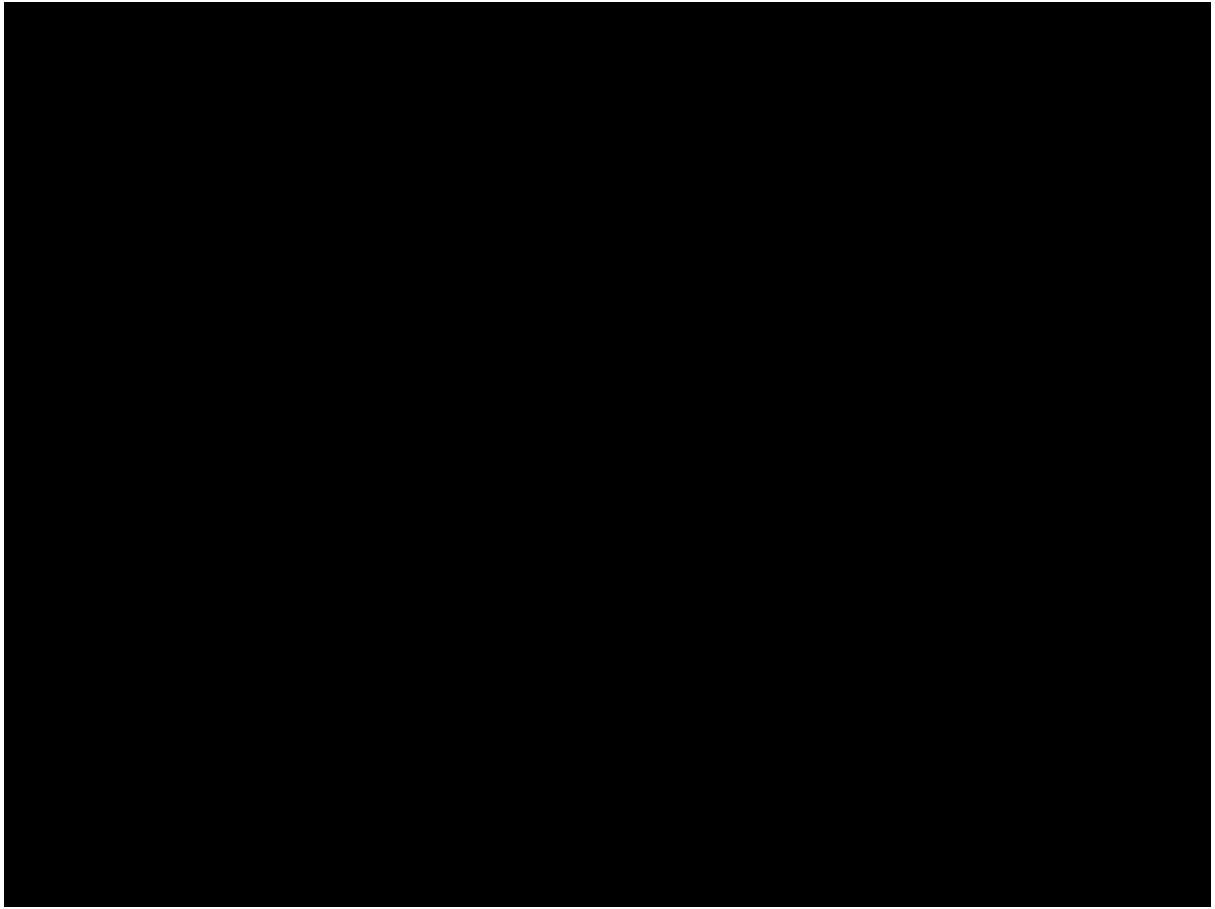


Image source: http://www.pair.toronto.edu/csc2547-w21/assets/slides/DeepSDF_TianchangShen.pdf

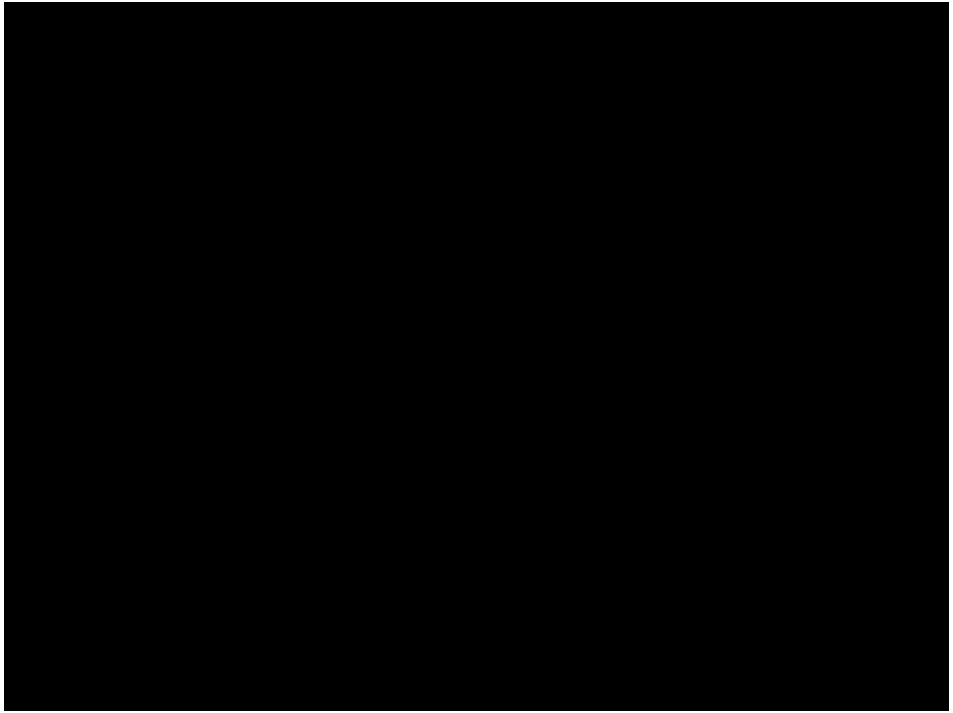
Latent space traversal

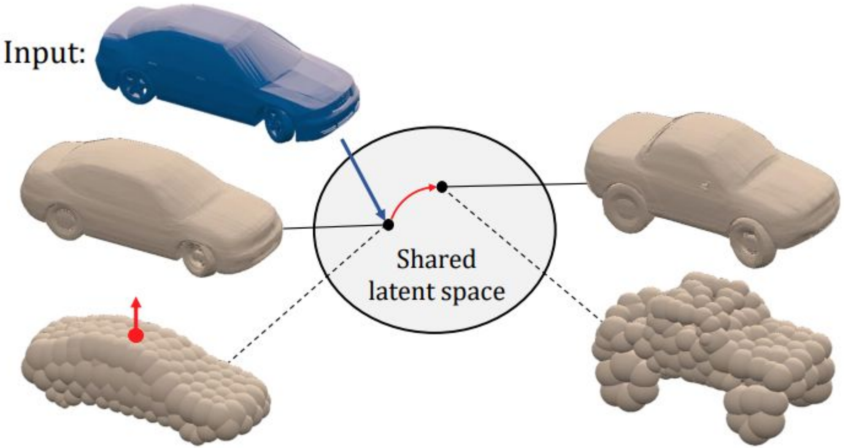


[1] Generative Adversarial Networks and Autoencoders for 3D Shapes

DualSDF

- DualSDF expresses shapes at two levels of granularity
  - Fine level captures fine details
  - Coarse level represents an abstracted proxy shape using simple and semantically consistent shape primitives

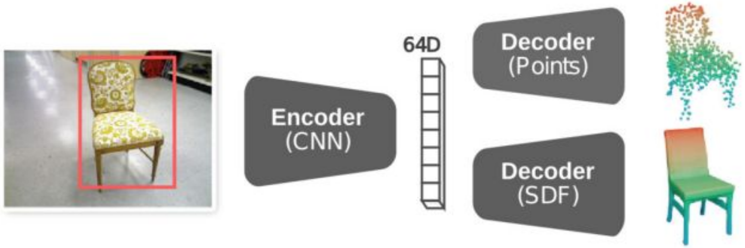




[1] DualSDF: Semantic Shape Manipulation using a Two-Level Representation

FroDO

- FroDO uses joint shape embedding
  - sparse point-based (efficiency)
  - dense surface (expressiveness) object shape representations



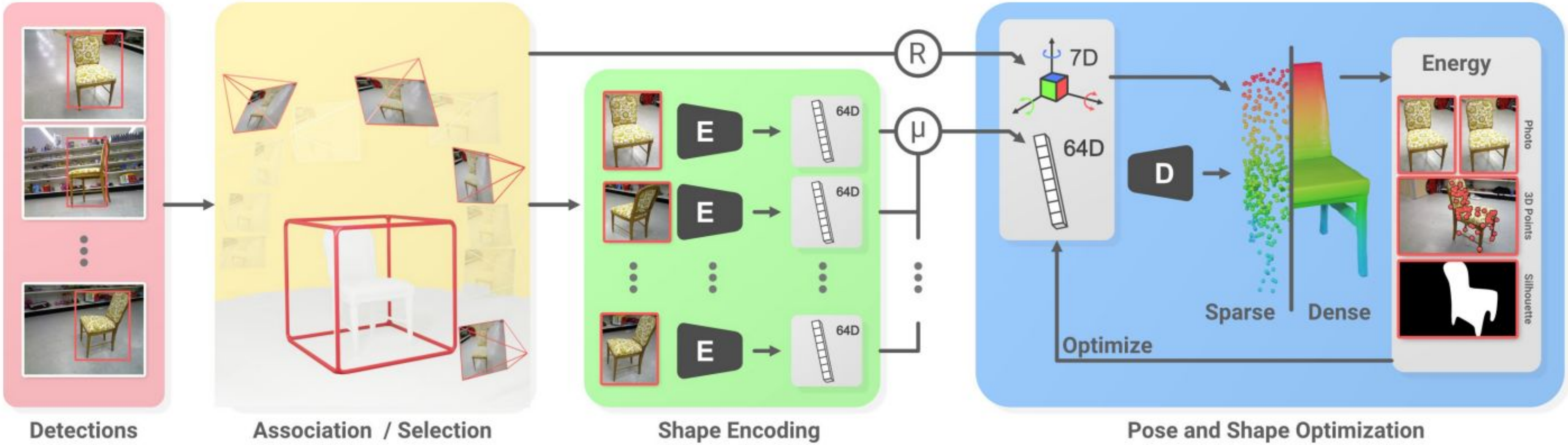
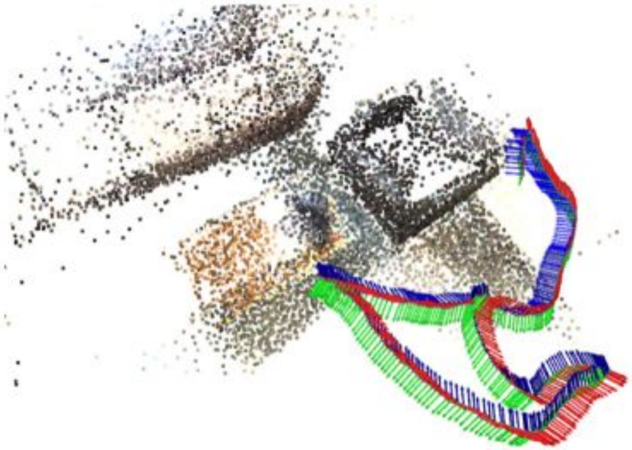


Figure 2: Given a sequence of calibrated, and localized RGB images, FroDO detects objects and infers their shape code and per-frame poses in a coarse-to-fine manner. We demonstrate FroDO on challenging sequences from real-world datasets that contain a single object (Redwood-OS) or multiple objects (ScanNet).

[1] FroDO: From Detections to 3D Objects

Motivation

- Right balance between faithful object reconstruction and a compact object representation
- A **bi-level object model** with coarse and fine levels, to enable joint optimization of object pose and shape. The two levels are coupled via a shared latent space
  - **Coarse-level** uses a primitive shape for robust pose and scale initialization
  - **Fine-level** uses SDF residual directly to allow accurate shape modeling
- A cost function to measure the mismatch between the bi-level object model and the segmented **RGB-D observations** in the world frame



[1] ELLIPSDF: Joint Object Pose and Shape Optimization with a Bi-level Ellipsoid and Signed Distance Function Description

Problem Formulation

- Overall cost = coarse shape error + fine shape error + regularization

$$e(\mathbf{T}, \delta\mathbf{z}, \boldsymbol{\theta}, \phi; \{\mathcal{X}_k(\mathbf{p})\}) \triangleq \alpha e_r(\delta\mathbf{z}) \tag{6}$$

$$+ \sum_{k=1}^{K} \sum_{\mathbf{p} \in \Omega_k^2} \sum_{(\mathbf{x}, d) \in \mathcal{X}_k(\mathbf{p})} \beta e_{\boldsymbol{\theta}}(\mathbf{x}, d, \mathbf{T}, \delta\mathbf{z}) + \gamma e_{\boldsymbol{\phi}}(\mathbf{x}, d, \mathbf{T}, \delta\mathbf{z}),$$

- Training cost

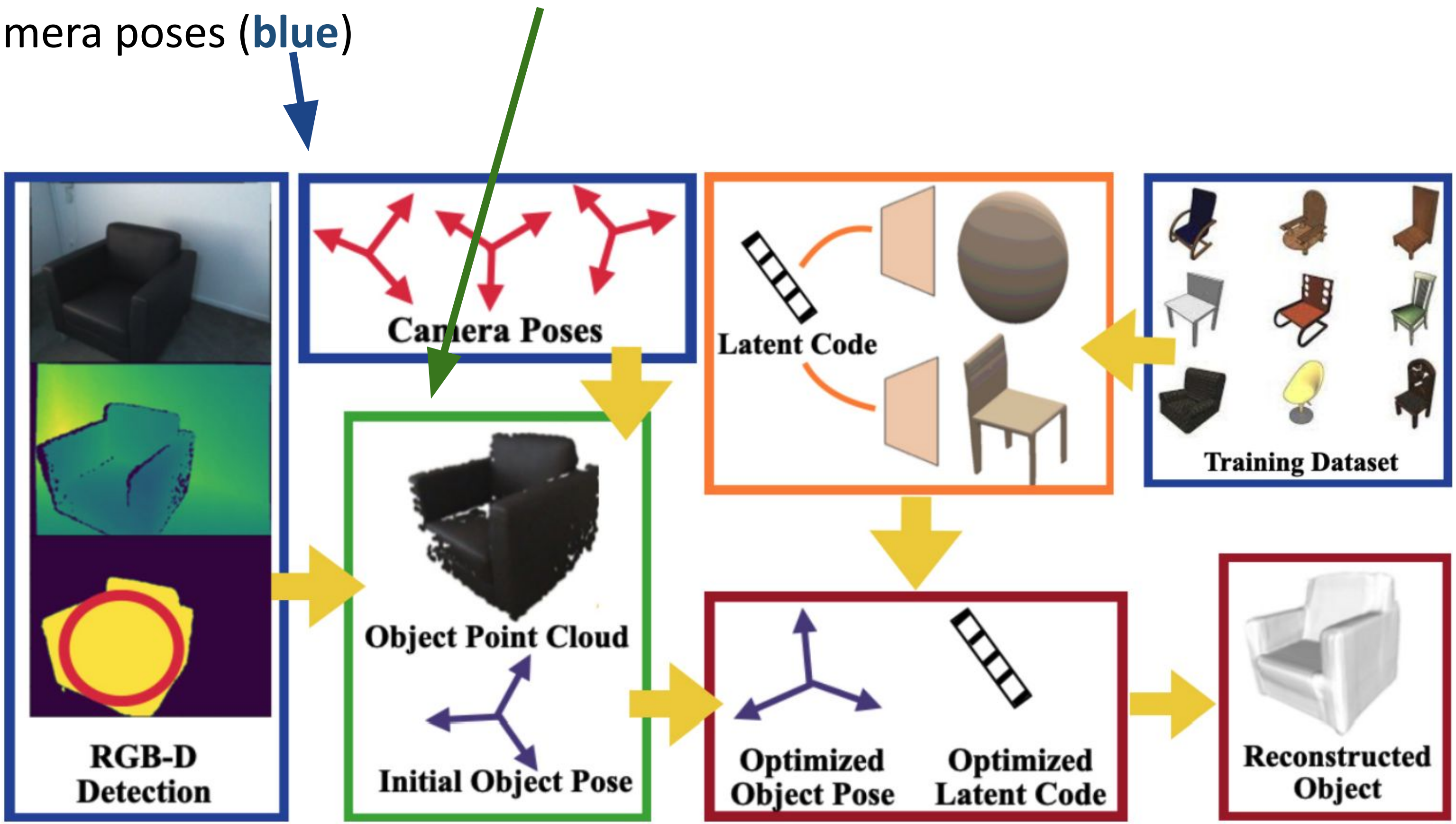
$$\min_{\{\delta\mathbf{z}_n\}, \boldsymbol{\theta}, \boldsymbol{\phi}} \sum_n e(\mathbf{I}_4, \delta\mathbf{z}_n, \boldsymbol{\theta}, \phi; \{\mathcal{X}_{n,k}(\mathbf{p})\}). \tag{7}$$

- Testing cost

$$\min_{\mathbf{T}, \delta\mathbf{z}} e(\mathbf{T}, \delta\mathbf{z}, \boldsymbol{\theta}^*, \phi^*; \{\mathcal{X}_k(\mathbf{p})\}). \tag{8}$$

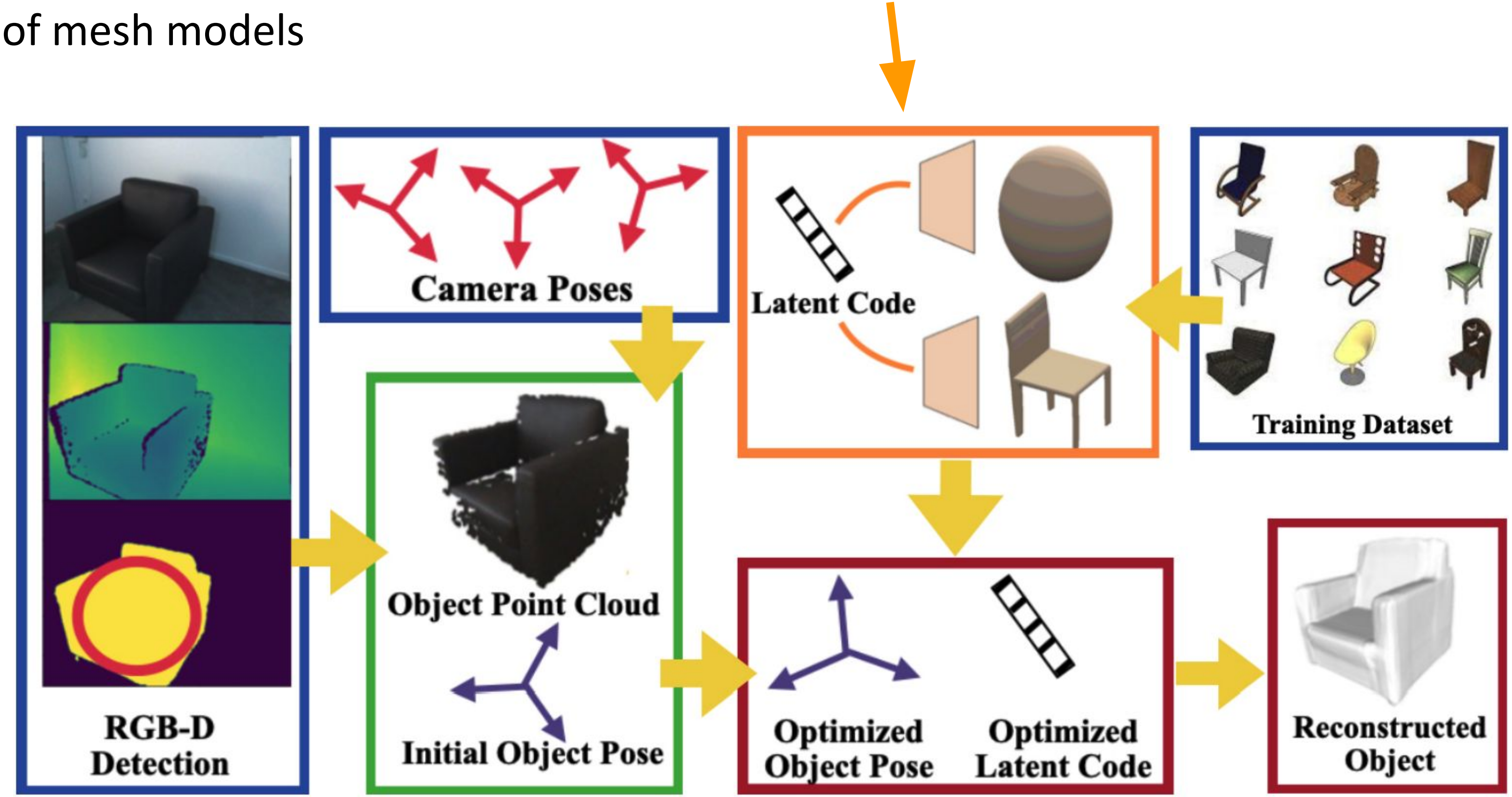
Overview

- Point cloud & initial pose (**green**) obtained from RGB-D detections with known camera poses (**blue**)



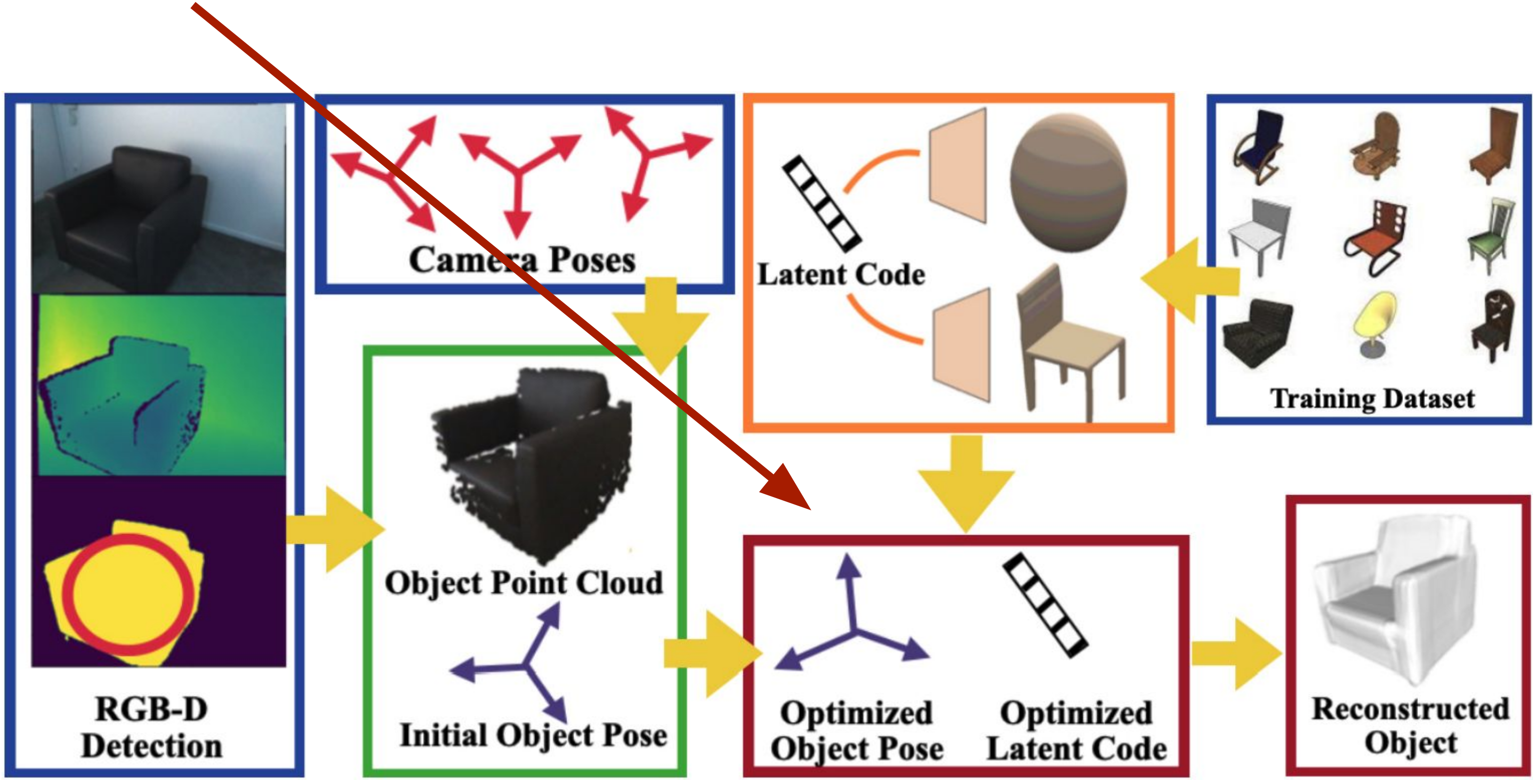
Overview

- A bi-level category shape description, consisting of a latent shape code, a coarse shape decoder, and a fine shape decoder (orange), is trained offline using a dataset of mesh models



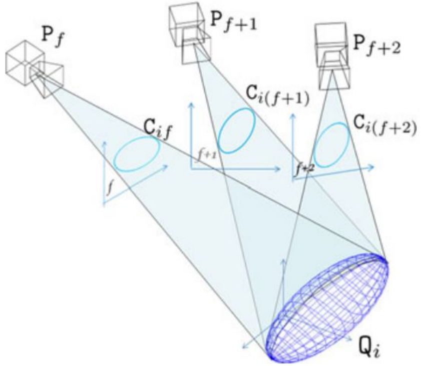
Overview

- Given the observed point cloud, the pose and shape deformation of the newly seen instance are optimized jointly online, achieving shape reconstruction in the global frame (**red**)



Object Pose Initialization

- Reconstruct ellipsoids from ellipses for initial object pose



$$\beta_{if} C^*_{if} = P_f Q^*_i P_f^\top$$

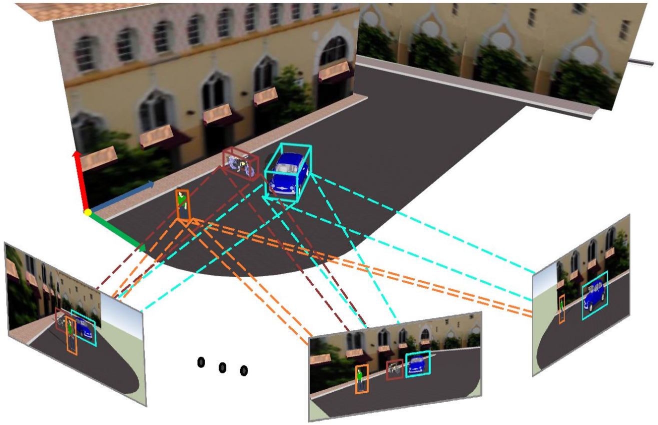
$$G_f = D(P \otimes P)E$$

$$\beta_{if} \mathbf{c}^*_{if} = \mathbf{G}_f \mathbf{v}^*_i$$

$$M_i \mathbf{w}_i = \mathbf{0}_{6F}$$

$$M_i = \begin{bmatrix} G_1 & -\mathbf{c}^*_{i1} & \mathbf{0}_6 & \mathbf{0}_6 & \dots & \mathbf{0}_6 \\ G_2 & \mathbf{0}_6 & -\mathbf{c}^*_{i2} & \mathbf{0}_6 & \dots & \mathbf{0}_6 \\ G_3 & \mathbf{0}_6 & \mathbf{0}_6 & -\mathbf{c}^*_{i3} & \dots & \mathbf{0}_6 \\ \vdots & \mathbf{0}_6 & \mathbf{0}_6 & \mathbf{0}_6 & \ddots & \mathbf{0}_6 \\ G_F & \mathbf{0}_6 & \mathbf{0}_6 & \mathbf{0}_6 & \dots & -\mathbf{c}^*_{iF} \end{bmatrix} \qquad \mathbf{w}_i = \begin{bmatrix} \mathbf{v}^*_i \\ \beta_i \end{bmatrix}$$

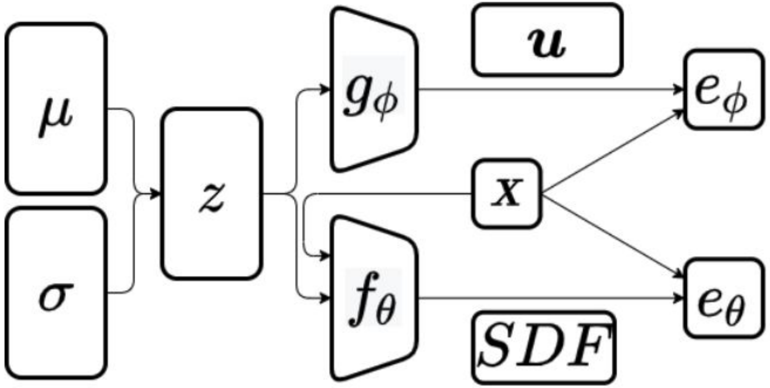
$$\tilde{\mathbf{w}}_i = \arg\min_{\mathbf{w}} \left\| \tilde{\mathbf{M}}_i \mathbf{w} \right\|_2^2 \qquad \|\mathbf{w}\|_2^2 = 1$$



[1] 3D Object Localisation from Multi-view Image Detections

Object Pose & Shape Optimization

- **Training phase**: optimize parameters of object class using offline data, from known meshes
- **Testing phase**: optimize the pose T and shape deformation δz of a previously unseen instance from the same category using online distance data from an RGB-D camera
  - Residuals relate both the **object pose** and shape to the SDF residual to enable joint optimization
  - Solve joint object pose and shape optimization via **gradient descent**

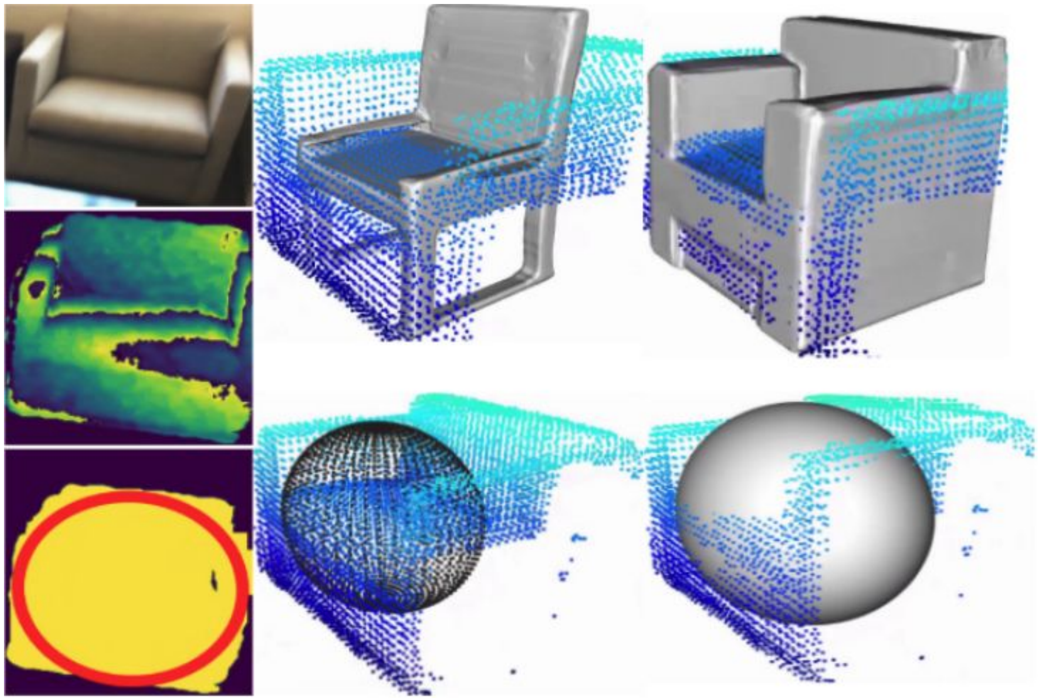


$$e_{\boldsymbol{\theta}}(\mathbf{x}, d, \mathbf{T}, \delta\mathbf{z}) \triangleq \rho(s f_{\boldsymbol{\theta}}(\mathbf{P}\mathbf{T}\underline{\mathbf{x}}; \mathbf{z} + \delta\mathbf{z}) - d). \quad (9)$$

$$e_{\boldsymbol{\phi}}(\mathbf{x}, d, \mathbf{T}, \delta\mathbf{z}) \triangleq \rho(sh(\mathbf{P}\mathbf{T}\underline{\mathbf{x}}, g_{\boldsymbol{\phi}}(\mathbf{z} + \delta\mathbf{z})) - d). \quad (12)$$

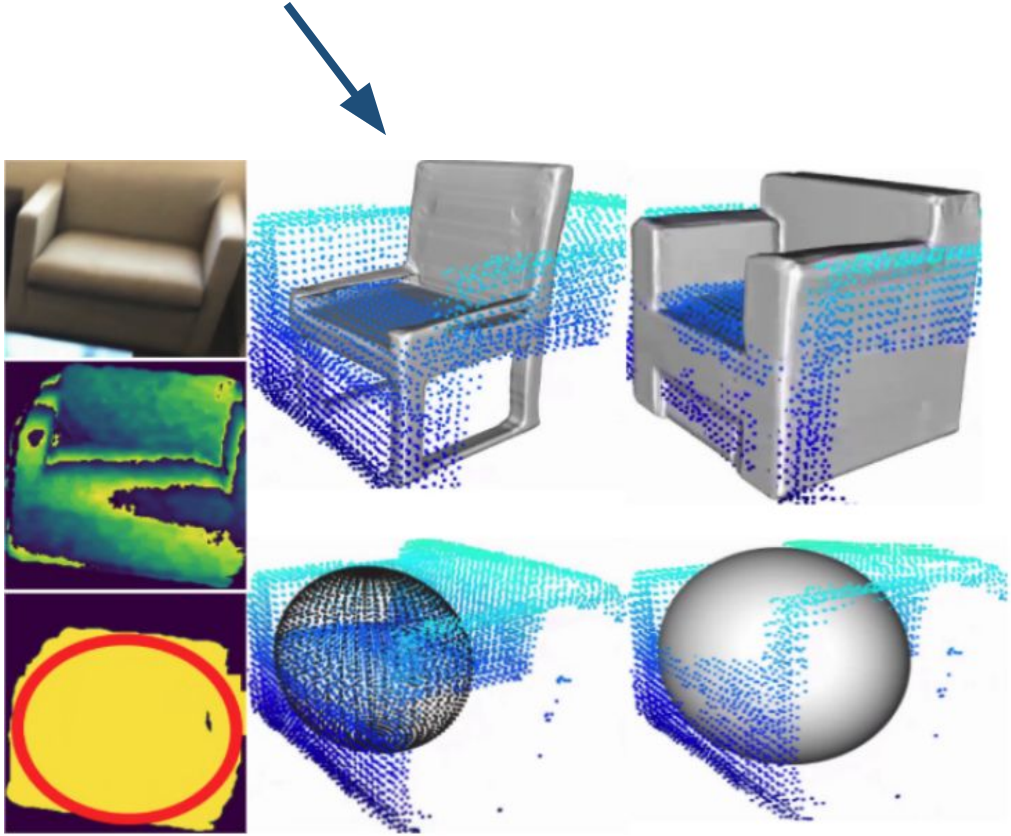
Qualitative results

- ELLIPSDF decoder model trained on synthetic CAD models in **ShapeNet**, visualization shows:
    - RGB image, depth image, instance segmentation (yellow), fitted ellipse (red)



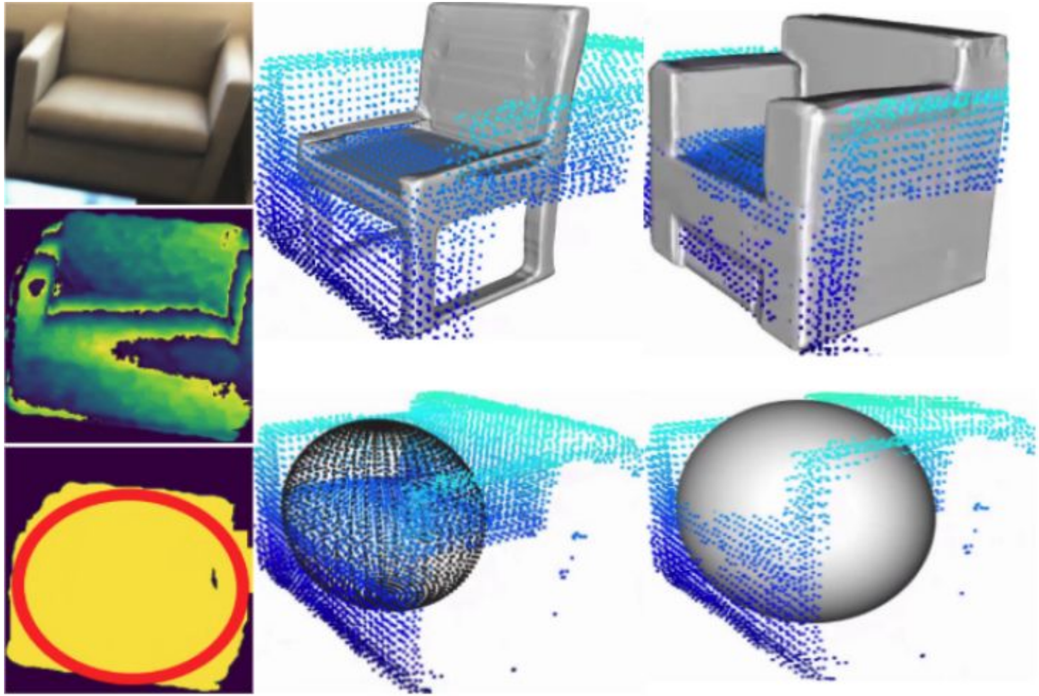
Qualitative results

- ELLIPSDF decoder model trained on synthetic CAD models in **ShapeNet**, visualization shows:
  - Mean shape and ellipsoid with initial pose



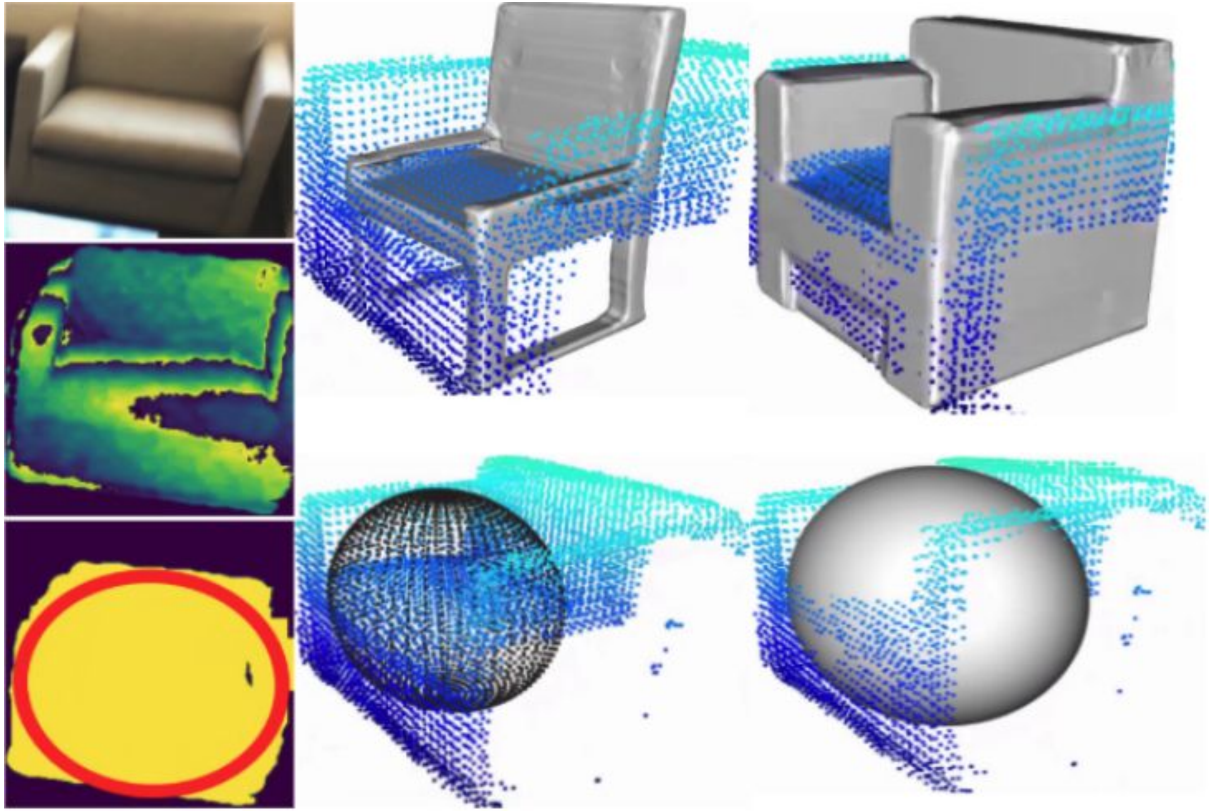
Qualitative results

- ELLIPSDF decoder model trained on synthetic CAD models in **ShapeNet**, visualization shows:
    - Optimized fine-level and coarse-level shapes with optimized pose



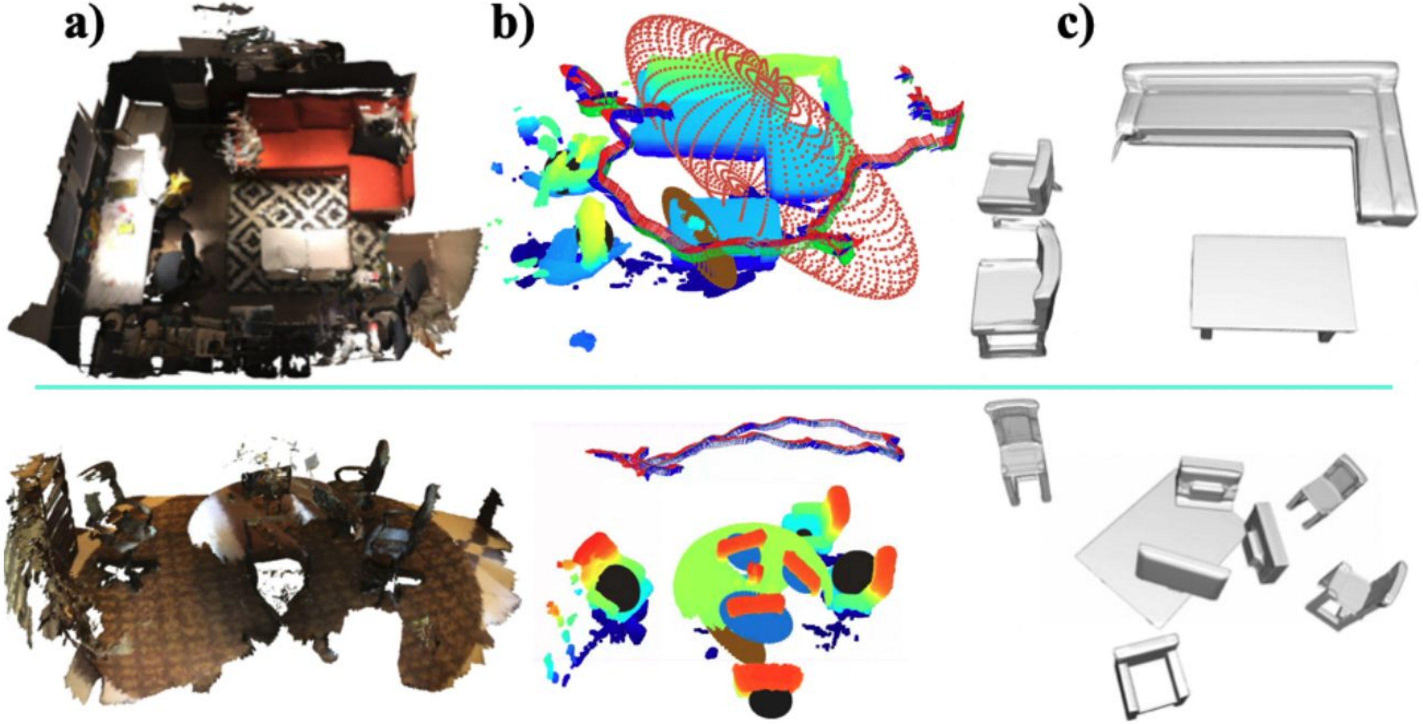
Qualitative results

- Optimization step improves the scale and shape estimates notably on **ScanNet**, e.g. by transforming the four-leg mean shape into an armchair



Qualitative results

- Reconstruction for a scene with multiple objects



Quantitative results

- Large-scale evaluation on ScanNet
  - Optimization step improves pose estimation accuracy
  - Coarse+fine model outperforms fine-model-only for shape estimation
  - ELLIPSDF is comparable with SOTA

Quantitative results for pose estimation on ScanNet:

Scan2CAD	Vid2CAD	ELLIPSDF (init)	ELLIPSDF (opt)
31.7	38.3	31.5	**39.6**

Quantitative results for shape evlaution on ScanNet:

Method # intances	cabinet 132	chair 820	display 209	table 146	avg. 327
ELLIPSDF (fine)	88.4	88.3	90.6	76.2	85.9
ELLIPSDF (coarse+fine)	**91.0**	**90.6**	**96.9**	**77.3**	**89.0**

Comparison of 3D detection results on ScanNet:

mAP @ IoU=0.5	Chair	Table	Display
FroDO	0.32	0.06	0.04
MOLTR	0.39	0.06	0.10
ELLIPSDF (fine)	0.42	0.26	0.25
ELLIPSDF (coarse+fine)	**0.43**	**0.27**	**0.31**

[1] Frodo: From detections to 3d objects
[2] MOLTR: Multiple Object Localization, Tracking and Reconstruction From Monocular RGB Videos

Timing

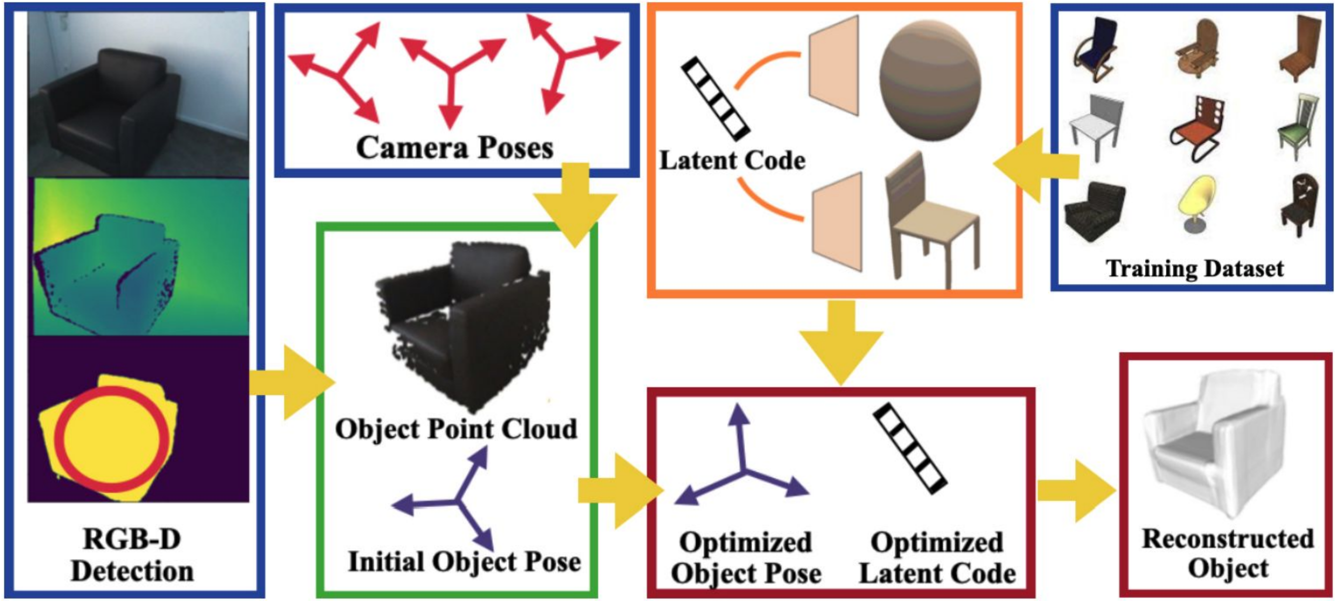
- Init is the pose initialization for 100 views
- Latent Code Opt and SIM(3) Opt are a single SGD step with respect to δz and T respectively using 10000 points as batch size
- SDF Decoding and Meshing are optional steps that generate SDF predictions over 2563 points and apply Marching Cubes to generate a mesh

Table 4. ELLIPSDF timing breakdown (sec)

Init	Latent Code Opt	SIM(3) Opt	SDF Decoding	Meshing
0.04	0.13	0.58	1.38	2.34

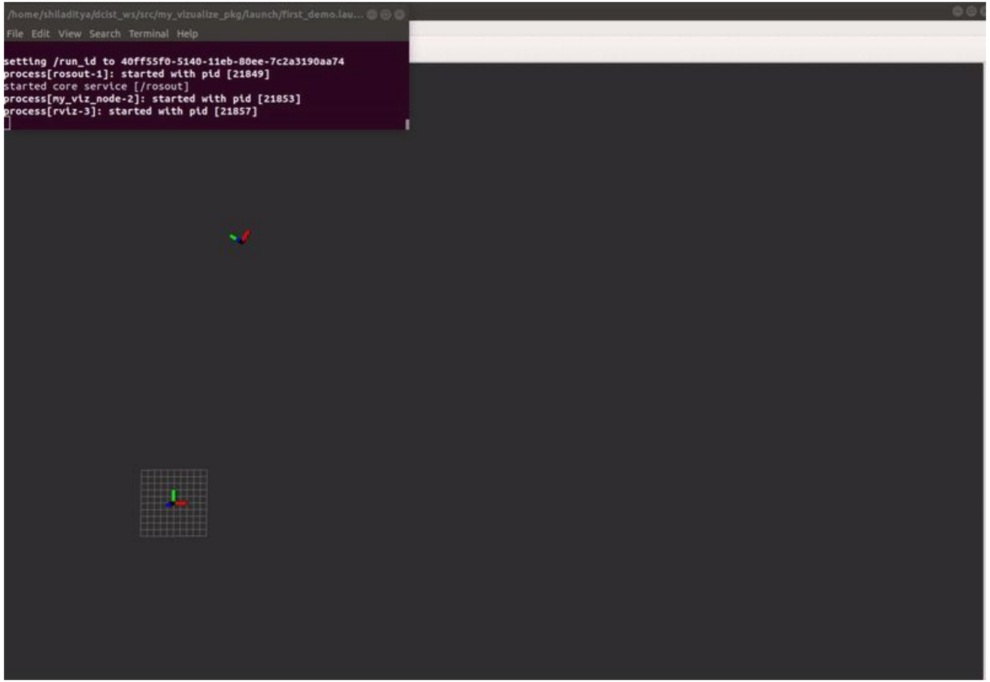
Contributions

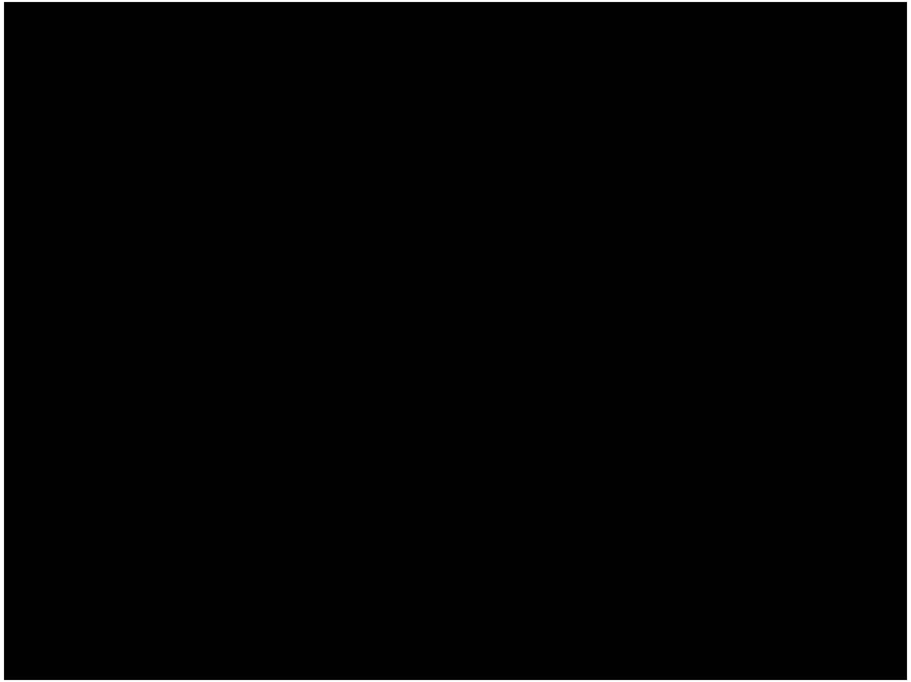
- To summarize, the main contribution of this work is the design of
  - a **bi-level object model** with coarse and fine levels, enabling joint optimization of object pose and shape
  - a **cost function** to measure the mismatch between the bi-level object model and the segmented RGB-D observations in the world frame



Future Directions

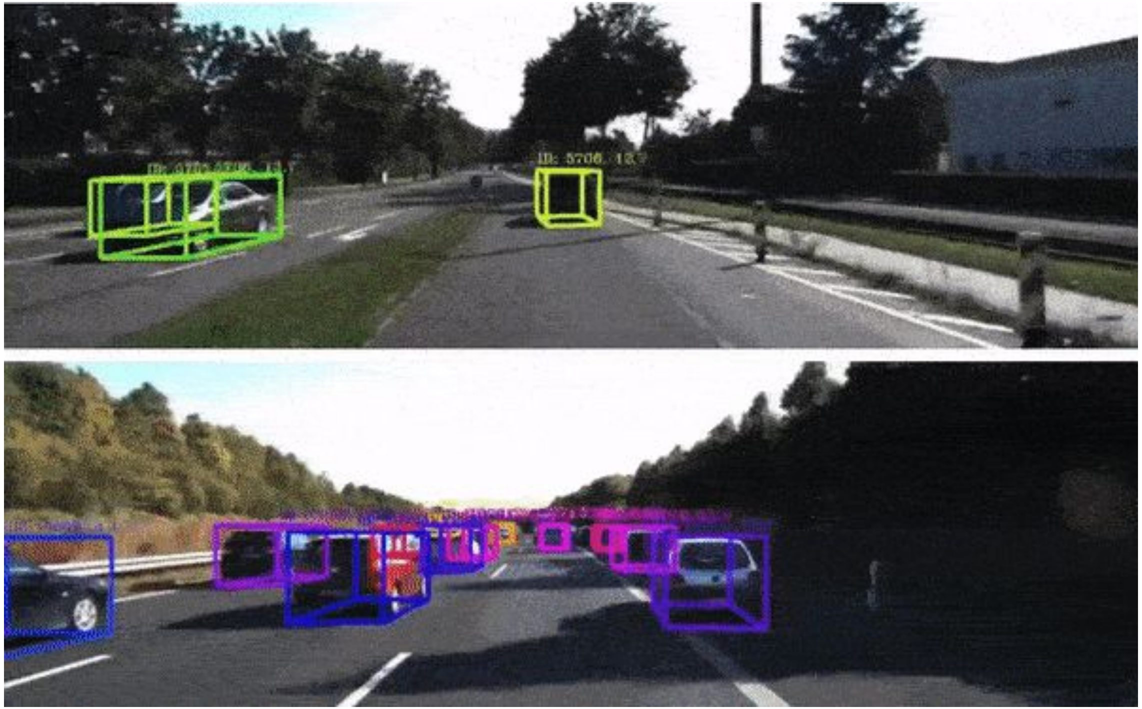
- Data association for loop closure for cars and quadrotors





Future Directions

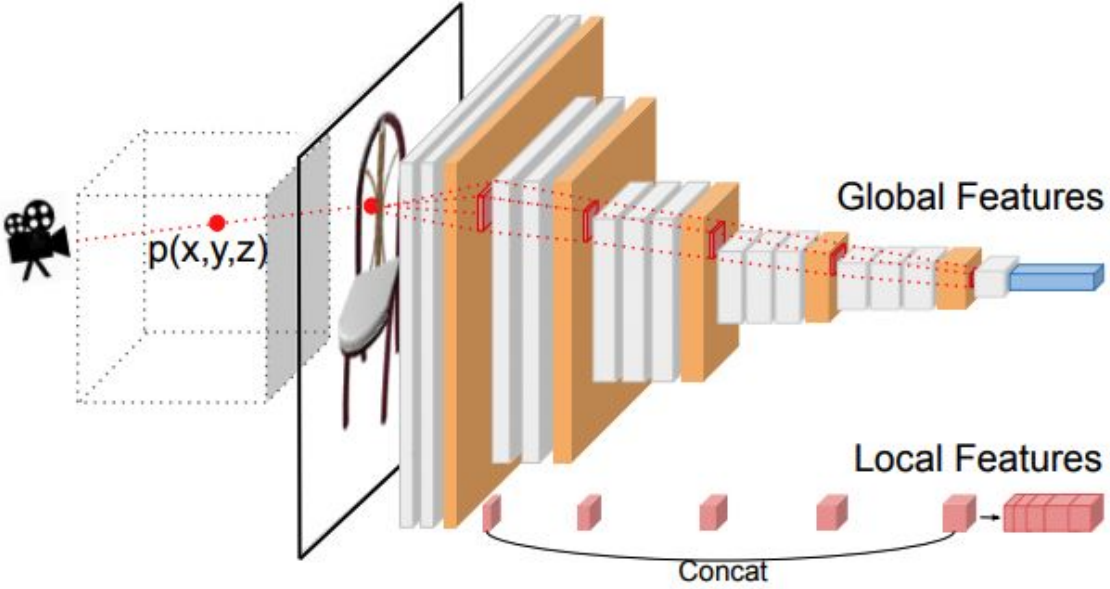
- Dynamic object tracking



[1] 3D Multi-Object Tracking: A Baseline and New Evaluation Metrics

Future Direction

- Vision only object shape optimization



[1] DISN: Deep Implicit Surface Network for High-quality Single-view 3D Reconstruction