Paper Reading (EPro-PnP)

Lin Li

August 2, 2025

Overview

1. EPro-Pnp for 3d pose estimation

EPro-PnP: Generalized End-to-End Probabilistic Perspective-n-Points for Monocular Object Pose Estimation

- proposed a probabilistic PnP layer for general end-to-end pose estimation via learnable 2D-3D correspondences
- EPro-PnP can easily reach toptier performance for 6DoF pose estimation by simply inserting it into the CDPN framework.

Overview

Goal: For each proposal object, predict a set

$$X = \{x^3D, x^2D, w^2D\}\text{where } i = 1, \cdots, N$$

corresponding points, with 3D object coordinates $x^3D \in R^3$, 2D image coordinates $x^2D \in R^2$, and 2D weights $w^2D \in R^2$

PnP layer Goal

Find the best pose y(expanded as rotation matrix R and translation vector t) to minimize the error

$$\arg\max_{y} \frac{1}{2} \sum_{i=1}^{N} \| w_i^{2D}(\pi(Rx_i^{3D} + t)) - x_i^{2D} \|^2$$

where we define

$$f_i(y) := w_i^{2D}(\pi(Rx_i^{3D} + t)) - x_i^{2D}$$

Bayesian Distribution

$$P(X|y) = \exp -\frac{1}{2}\sum_{i=1}^{N} \|f_i(y)\|^2$$

Using uninformation prior for pose y

$$P(X|y) = \frac{\exp -\frac{1}{2}\sum_{i=1}^{N} \|f_i(y)\|^2}{\int \exp -\frac{1}{2}\sum_{i=1}^{N} \|f_i(y)\|^2 dy}$$

KL loss

$$L_{KL} = \int -t(y) \log(P(X|y))dy + \log \int P(X|y)dy$$

set target distribution $t(y)$ as a Dirac-like function at ground truth y_{gt}

$$L_{KL} = \frac{1}{2}\sum_{i=1}^{N} \|f_i(y_{gt})\|^2 + \log \int \exp -\frac{1}{2}\sum_{i=1}^{N} \|f_i(y)\|^2 dy$$

The first term: loss in reproject at gt pose
The second term: loss in reproject at predicted pose

reproject loss in predict pose

$$L_{pred} \approx \log \frac{1}{K} \sum_{i=1}^{N} \frac{\exp -\frac{1}{2} \sum_{i=1}^{N} \|f_i(y)\|^2}{q(y_i)}$$

Choice of proposal distribution

- For position, we adopt the 3DoF multivariate t-distribution
- For 1D yaw-only orientation, we use a mixture of von Mises and uniform distribution

EPro-PnP pipeline

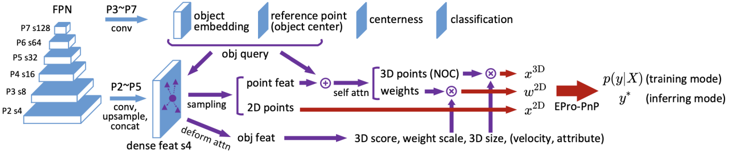

Figure 5. **The deformable correspondence network** based on the FCOS3D [47] detector. Note that the sampled point-wise features are shared by the point-level subnet and the deformable attention layer that aggregates the features for object-level predictions.

Training mode: Using correspond 3D points and 2D points to estimation 3D pose distribution Infering mode: Using least square method to compute the best pose estimation

AMIS-based Monte Carlo pose loss

Algorithm 1: AMIS-based Monte Carlo pose loss

Input : $X = \{x_i^{\mathrm{3D}}, x_i^{\mathrm{2D}}, w_i^{\mathrm{2D}}\}$

Output: L_{pred}

- $y^*, \Sigma_{y^*} \leftarrow PnP(X)$ // Laplace approximation
- Fit $q_1(y)$ to y^*, Σ_{y^*} // initial proposal
- **for** $1 \leq t \leq T$ **do**
  - Generate K' samples $y_{j=1\cdots K'}^t$ from $q_t(y)$
  - **for** $1 \leq j \leq K'$ **do**
    - $P_j^t \leftarrow \exp -\frac{1}{2}\sum_{i=1}^{N}\left\| f_i(y_j^t)\right\|^2$ // eval integrand
  - **for** $1 \leq \tau \leq t$ **and** $1 \leq j \leq K'$ **do**
    - $Q_j^\tau \leftarrow \frac{1}{t}\sum_{m=1}^{t} q_m(y_j^\tau)$ // eval proposal mix
    - $v_j^\tau \leftarrow P_j^\tau / Q_j^\tau$ // importance weight
  - **if** $t < T$ **then**
    - Estimate $q_{t+1}(y)$ from all weighted samples $\{y_j^\tau, v_j^\tau \mid 1 \leq \tau \leq t, 1 \leq j \leq K'\}$
- $L_{\mathrm{pred}} \leftarrow \log \frac{1}{TK'}\sum_{t=1}^{T}\sum_{j=1}^{K'} v_j^t$