Supplemental Material:
HybridPose: 6D Object Pose Estimation under Hybrid Representations

Chen Song*, Jiaru Song*, Qixing Huang
The University of Texas at Austin
song@cs.utexas.edu, jiarus@cs.utexas.edu, huangqx@cs.utexas.edu

This is the supplemental material to "HybridPose: 6D Object Pose Estimation under Hybrid Representations". We provide detailed explanations to our the algorithm used in the initialization sub-module. We also conduct a stability analysis of the refinement sub-module, and show how the optimal solution to the the objective function changes with respect to noise in predicted representations. In addition, we present an ablation study on Linemod [1] dataset. While keypoints alone already achieve reasonable pose estimation performance on Linemod, utilizing symmetry correspondences and edge vectors lead to slight improvements.

1. Initial Solution for Pose Regression

Recall that we denote 3D keypoint coordinates in the canonical coordinate system as $\overline{\boldsymbol{p}}_k, 1 \leq k \leq |\mathcal{K}|$. To make notations uncluttered, we denote output of the first module, i.e., predicted keypoints, edge vectors, and symmetry correspondences as $\boldsymbol{p}_k \in \mathbb{R}^2, 1 \leq k \leq |\mathcal{K}|$, $\boldsymbol{v}_e \in \mathbb{R}^2, 1 \leq e \leq |\mathcal{E}|$, and $(\boldsymbol{q}_{s,1} \in \mathbb{R}^2, \boldsymbol{q}_{s,2} \in \mathbb{R}^2), 1 \leq s \leq |\mathcal{S}|$, respectively. Our formulation also uses the homogeneous coordinates $\hat{\boldsymbol{p}}_k \in \mathbb{R}^3$, $\hat{\boldsymbol{v}_e} \in \mathbb{R}^3$,$\hat{\boldsymbol{q}}_{s,1} \in \mathbb{R}^3$ and $\hat{\boldsymbol{q}}_{s,2} \in \mathbb{R}^3$ of $\boldsymbol{p}_k$, $\boldsymbol{v}_e$, $\boldsymbol{q}_{s,1}$ and $\boldsymbol{q}_{s,2}$ respectively. The homogeneous coordinates are normalized by camera intrinsic matrix.

1.1. Three constraints for object pose.

We seek to generalize the EPnP algorithm which only exploits keypoint 2D-3D correspondences for pose estimation by leveraging hybrid representations, keypoint, edge vector and symmetry correspondence. To this end, we introduce the following difference vectors for each type of predicted elements:

$$\overline{\boldsymbol{r}}^{\mathcal{K}}_{R,\boldsymbol{t}}(\boldsymbol{p}_k) := \hat{\boldsymbol{p}}_k \times (R\overline{\boldsymbol{p}}_k + \boldsymbol{t}), \tag{1}$$

$$\overline{\boldsymbol{r}}^{\mathcal{E}}_{R,\boldsymbol{t}}(\boldsymbol{v}_e, \boldsymbol{p}_{e_s}) := \hat{\boldsymbol{v}}_e \times (R\overline{\boldsymbol{p}}_{e_t} + \boldsymbol{t}) + \hat{\boldsymbol{p}}_{e_s} \times (R\overline{\boldsymbol{v}}_e), \tag{2}$$

$$r^{\mathcal{S}}_{R,\boldsymbol{t}}(\boldsymbol{q}_{s,1}, \boldsymbol{q}_{s,2}) := (\hat{\boldsymbol{q}}_{s,1} \times \hat{\boldsymbol{q}}_{s,2})^T \boldsymbol{R}\overline{\boldsymbol{n}}_r. \tag{3}$$

where e_s and e_t are end vertices of edge e, $\overline{\boldsymbol{v}}_e = \overline{\boldsymbol{p}}_{e_t} - \overline{\boldsymbol{p}}_{e_s} \in \mathbb{R}^3$, and $\overline{\boldsymbol{n}}_r \in \mathbb{R}^3$ is the normal of the reflection symmetry plane in the canonical system.

Proposition 1 *If there is a perfect alignment between the predicted elements and the corresponding 3D keypoint template with respect to the ground-truth pose $R^\star, \boldsymbol{t}^\star$. Then*

$$\overline{\boldsymbol{r}}^{\mathcal{K}}_{R^\star,\boldsymbol{t}^\star}(\boldsymbol{p}_k) = \boldsymbol{0}, \overline{\boldsymbol{r}}^{\mathcal{E}}_{R^\star,\boldsymbol{t}^\star}(\boldsymbol{v}_e \boldsymbol{p}_{e_s}) = \boldsymbol{0}, r^{\mathcal{S}}_{R,\boldsymbol{t}}(\boldsymbol{q}_{s,1}, \boldsymbol{q}_{s,2}) = 0$$

Proof:

1. The proof of the first equality is straight-forward as there exists a "depth" $\lambda_k > 0$ so that

$$\lambda_k \hat{\boldsymbol{p}}_k = R^\star \overline{\boldsymbol{p}}_k + \boldsymbol{t}^\star$$

   It follows that

$$\boldsymbol{0} = \lambda_k \hat{\boldsymbol{p}}_k \times \hat{\boldsymbol{p}}_k = \hat{\boldsymbol{p}}_k \times (R^\star \overline{\boldsymbol{p}}_k + \boldsymbol{t}^\star)$$

2. The proof of the second equality follows the first equality. So we have

$$\hat{\boldsymbol{p}}_{e_s} \times (R^\star \overline{\boldsymbol{p}}_{e_s} + \boldsymbol{t}^\star) = \boldsymbol{0} \quad \hat{\boldsymbol{p}}_{e_t} \times (R^\star \overline{\boldsymbol{p}}_{e_t} + \boldsymbol{t}^\star) = \boldsymbol{0}$$

   Replacing $\hat{\boldsymbol{p}}_{e_t}$ by $\hat{\boldsymbol{v}_e} + \hat{\boldsymbol{p}}_{e_s}$, we have

$$\hat{\boldsymbol{v}_e} \times (R^\star \overline{\boldsymbol{p}}_{e_t} + \boldsymbol{t}^\star) + \hat{\boldsymbol{p}}_{e_s} \times (R^\star \overline{\boldsymbol{p}}_{e_t} + \boldsymbol{t}^\star) = \boldsymbol{0}$$

   Replacing the second $\overline{\boldsymbol{p}}_{e_t}$ by $\overline{\boldsymbol{v}}_e + \overline{\boldsymbol{p}}_{e_s}$ in the above equation, we have

$$\hat{\boldsymbol{v}_e} \times (R^\star \overline{\boldsymbol{p}}_{e_t} + \boldsymbol{t}^\star) + \hat{\boldsymbol{p}}_{e_s} \times R^\star \overline{\boldsymbol{v}}_e = \boldsymbol{0}$$

3. To prove the third equality, define the depths of $\hat{\boldsymbol{q}}_{s,1}$ and $\hat{\boldsymbol{q}}_{s,2}$ as $\lambda_{s,1}$ and $\lambda_{s,2}$ and the corresponding 3D model points in the canonical system as $\overline{\boldsymbol{q}}_{s,1}$ and $\overline{\boldsymbol{q}}_{s,2}$. $\overline{\boldsymbol{p}}_s$ is a point on the reflectional symmetry plane, whose normal is $\overline{\boldsymbol{n}}_r$. Given a symmetry correspondence pair ($\overline{\boldsymbol{q}}_{s,1}$,$\overline{\boldsymbol{q}}_{s,2}$), we have

$$\overline{\boldsymbol{q}}_{s,2} = (I_3 - 2\overline{\boldsymbol{n}}_r \overline{\boldsymbol{n}}_r^{\mathrm{T}}) \overline{\boldsymbol{q}}_{s,1} + 2\overline{\boldsymbol{n}}_r \overline{\boldsymbol{n}}_r^{\mathrm{T}} \overline{\boldsymbol{p}}_s \tag{4}$$

*Authors contributed equally

Let $R_s = I_3 - 2\overline{\boldsymbol{n}}_r\overline{\boldsymbol{n}}_r^{\mathrm{T}}, \boldsymbol{t}_s = 2\overline{\boldsymbol{n}}_r\overline{\boldsymbol{n}}_r^{\mathrm{T}}\overline{\boldsymbol{p}}_s$. Following the camera perspective model, we have

$$\lambda_{s,1}\hat{\boldsymbol{q}}_{s,1} = R^\star\overline{\boldsymbol{q}}_{s,1} + t^\star$$
$$\lambda_{s,2}\hat{\boldsymbol{q}}_{s,2} = R^\star R_s\overline{\boldsymbol{q}}_{s,1} + R^\star\boldsymbol{t}_s + t^\star$$

Subtracting these two equations, we have

$$\lambda_{s,2}\hat{\boldsymbol{q}}_{s,2} - \lambda_{s,1}\hat{\boldsymbol{q}}_{s,1} = R^\star(R_s\overline{\boldsymbol{q}}_{s,1} + \boldsymbol{t}_s - \overline{\boldsymbol{q}}_{s,1})$$

Left multiply both sides of the equation by $\hat{\boldsymbol{q}}_{s,2}\times$ yields

$$-\lambda_{s,1}\hat{\boldsymbol{q}}_{s,2}\times\hat{\boldsymbol{q}}_{s,1} = \hat{\boldsymbol{q}}_{s,2}\times[R^\star(R_s\overline{\boldsymbol{q}}_{s,1} + \boldsymbol{t}_s - \overline{\boldsymbol{q}}_{s,1})] \tag{5}$$

Geometrically, (5) reveals that $\hat{\boldsymbol{q}}_{s,2}\times\hat{\boldsymbol{q}}_{s,1}$ is perpendicular to the plane with span of $\{\boldsymbol{q}_{s,2}, R^\star(R_s\overline{\boldsymbol{q}}_{s,1} + \boldsymbol{t}_s - \overline{\boldsymbol{q}}_{s,1})\}$, thus we have

$$(\hat{\boldsymbol{q}}_{s,2}\times\hat{\boldsymbol{q}}_{s,1})^{\mathrm{T}}R^\star(R_s\overline{\boldsymbol{q}}_{s,1} + \boldsymbol{t}_s - \overline{\boldsymbol{q}}_{s,1}) =$$
$$2(\overline{\boldsymbol{n}}_r^{\mathrm{T}}(\overline{\boldsymbol{p}}_s - \overline{\boldsymbol{q}}_{s,1}))(\hat{\boldsymbol{q}}_{s,2}\times\hat{\boldsymbol{q}}_{s,1})^{\mathrm{T}}R^\star\overline{\boldsymbol{n}}_r = 0$$

Since $2(\overline{\boldsymbol{n}}_r^{\mathrm{T}}(\overline{\boldsymbol{p}}_s - \overline{\boldsymbol{q}}_{s,1}))$ is a non-zero scalar, we can delete this term and finally get

$$(\hat{\boldsymbol{q}}_{s,2}\times\hat{\boldsymbol{q}}_{s,1})^{\mathrm{T}}R^\star\overline{\boldsymbol{n}}_r = 0$$

1.2. Pose solution in eigenvector space.

A nice feature shared by (1), (2) and (3) is that all constraints are linear in the elements of R and $\boldsymbol{t}$. This allows us to derive a closed-form solution of R and $\boldsymbol{t}$ in the affine transformation space. Specifically, we can define $\boldsymbol{x} = (\boldsymbol{r}_1^{\mathrm{T}}, \boldsymbol{r}_2^{\mathrm{T}}, \boldsymbol{r}_3^{\mathrm{T}}, \boldsymbol{t}^{\mathrm{T}})^{\mathrm{T}}_{12\times1}$ as a vector that contains rotation and translation parameters in affine space. Expanding constraint (1) and constraint (2) yields three linear equations for each predicted element respectively for $\boldsymbol{x}$, and expanding constraint (3) yields one linear equation. By concatenating all linear equations of predicted elements together, we can generate a linear system of the form $A\boldsymbol{x} = \boldsymbol{0}$, where A is matrix and its dimension is $(3|\mathcal{K}| + 3|\mathcal{E}| + |\mathcal{S}|)\times 12$.

To model the relative importance among keypoints, edge vectors, and symmetry correspondences, we rescale (2) and (3) by hyper-parameters α_E and α_S, respectively, to generate A. As discussed in the body of this paper, we calculate α_E and α_S by solving an optimization problem using finite-difference and back-track line search.

Then following EPnP [2], we compute $\boldsymbol{x}$ as

$$\boldsymbol{x} = \sum_{i=1}^{N}\gamma_i\boldsymbol{v}_i \tag{6}$$

where $\boldsymbol{v}_i$ is the i^{th} smallest right singular vector of A. Ideally, when predicted elements are noise-free, $N = 1$ with $\boldsymbol{x} = \boldsymbol{v}_1$ is an optimal solution. However, this strategy performs poorly given noisy predictions. Same as EPnP [2], we choose $N = 4$.

1.3. Optimize a good linear combination.

To compute the optimal $\boldsymbol{x}$, we optimize latent variables γ_i and the rotation matrix R with following objective function:

$$\min_{R\in\mathbb{R}^{3\times3},\gamma_i} \|\sum_{i=1}^{4}\gamma_i R_i - R\|_{\mathcal{F}}^2 \tag{7}$$

where $R_i \in \mathbb{R}^{3\times3}$ is reshaped from the first 9 elements of $\boldsymbol{v}_i$. We solve this optimization problem with the following alternating procedure:

1. Fix γ_i and solve for R by SVD. i.e. $R = U\mathrm{diag}(1,1,1)V^{\mathrm{T}}$ given $\sum_{i=1}^{4}\gamma_i R_i = U\Sigma V^{\mathrm{T}}$[1],
2. Fix R and solve for γ_i's by optimizing a linear system $\sum_{i=1}^{4}\gamma_i R_i = R$ in an element-wise manner.

To initialize γ_i's for the above optimization problem, we calculate γ_i with $i = 1...3$ by enforcing that $\sum_{i=1}^{3}\gamma_i R_i$ is an orthogonal matrix[2]:

$$(\sum_{i=1}^{3}\gamma_i R_i)^{\mathrm{T}}\sum_{i=1}^{3}\gamma_i R_i = I_3 \tag{8}$$

Since I_3 is a symmetric matrix, expanding (8) yields 6 nonlinear constraints for $\boldsymbol{\gamma} = (\gamma_1,\gamma_2,\gamma_3)^{\mathrm{T}}$, which is however uneasy to solve. We then define a new vector $\boldsymbol{y} = (y_1, y_2, y_3, y_4, y_5, y_6)^{\mathrm{T}} = (\gamma_1^2, \gamma_1\gamma_2, \gamma_1\gamma_3, \gamma_2^2, \gamma_2\gamma_3, \gamma_3^2)^{\mathrm{T}}$ and form a linear system $C\boldsymbol{y} = \boldsymbol{z}$ which has the unique solution with $\boldsymbol{z}$ generated from I_3. Afterwards, it is easy to recover γ_i from $\boldsymbol{y}$ and optimize from initialized γ_i alone with $\gamma_4 = 0$.

After optimization, we again apply SVD to project $\sum_{i=1}^{4}\gamma_i R_i$ onto the space of SO(3), i.e., $R^{init} = U\mathrm{diag}(1,1,1)V^{\mathrm{T}}$ and enforce $\det(R^{init}) > 0$ where $R^{init} = U\Sigma V^T$. Leveraging $A\boldsymbol{x} = 0$ defined in section (1.2), the corresponding translation $\boldsymbol{t}^{init}$ is

$$\boldsymbol{t}^{init} = -(A_2^{\mathrm{T}}A_2)^{-1}A_2^{\mathrm{T}}A_1\boldsymbol{r}^{init} \tag{9}$$

where $A_1 = A_{[:,1:9]}$, $A_2 = A_{[:,10:12]}$, $\boldsymbol{r}^{init}_{9\times1}$ is reshaped from R^{init}.

2. Ablation Study on Linemod Dataset

Table 1 summarizes the performance of HybridPose using different predicted intermediate representations on Linemod dataset. The overall relative performance is similar to that on Occlusion-Linemod. Specifically, adding symmetry correspondences can boost the performance of rotations. Adding edge vectors can significantly boost the performance of both rotations and translations. Moreover, such

[1] If $\det(R) < 0$ we enforce $\det(R) > 0$ by defining $R = U\mathrm{diag}(1,1,-1)V^{\mathrm{T}}$.

[2] The reason of initializing 3 γ_i's is that (8) is unable to provide enough linear constraints for 4 γ_i's and this initialization ensures the convergence of optimization.

	keypoints		keypoints + symmetries		full model	
	Rotation	Translation	Rotation	Translation	Rotation	Translation
ape	1.122°	0.085	1.064°	0.090	0.808°	0.055
benchvise	1.319°	0.039	1.194°	0.037	0.657°	0.015
cam	1.310°	0.058	1.203°	0.058	0.716°	0.025
can	1.323°	0.053	1.210°	0.053	0.696°	0.024
cat	1.127°	0.062	1.031°	0.062	0.696°	0.029
driller	1.387°	0.037	1.294°	0.034	0.792°	0.019
duck	1.052°	0.080	1.038°	0.080	0.710°	0.044
eggbox	1.599°	0.072	1.317°	0.056	0.740°	0.029
glue	1.064°	0.053	1.063°	0.053	0.759°	0.026
holepuncher	1.351°	0.076	1.188°	0.073	0.709°	0.034
iron	1.629°	0.038	1.456°	0.039	0.769°	0.016
lamp	1.606°	0.036	1.321°	0.036	0.740°	0.020
phone	1.093°	0.038	1.093°	0.038	0.695°	0.021
mean	1.306°	0.056	1.190°	0.055	0.730°	0.028

Table 1. **Qualitative evaluation with different intermediate representations (Linemod).** We report errors using two metrics: the median of absolute angular error in rotation, and the median of relative error in translation with respect to object diameter.

improvements are consistent when starting from only using keypoints and when starting from combing keypoints and edge vectors.

3. Stability Analysis for Pose Refinement

In this section, we provide a local stability analysis of the pose regression procedure, which amounts to solving the following optimization problem:

$$\min_{R,\boldsymbol{t}} \sum_{k=1}^{|\mathcal{K}|} \rho(\|\boldsymbol{r}_{R,\boldsymbol{t}}^{\mathcal{K}}(\boldsymbol{p}_k)\|, \beta_{\mathcal{K}})\|\boldsymbol{r}_{R,\boldsymbol{t}}^{\mathcal{K}}(\boldsymbol{p}_k)\|_{\Sigma_k}^2 + \frac{|\mathcal{K}|}{|\mathcal{E}|}\sum_{e=1}^{|\mathcal{E}|} \rho(\|\boldsymbol{r}_{R,\boldsymbol{t}}^{\mathcal{E}}(\boldsymbol{v}_e)\|, \beta_{\mathcal{E}})\|\boldsymbol{r}_{R,\boldsymbol{t}}^{\mathcal{E}}(\boldsymbol{v}_e)\|_{\Sigma_e}^2 + \frac{|\mathcal{K}|}{|\mathcal{S}|}\sum_{s=1}^{|\mathcal{S}|} \rho(r_{R,\boldsymbol{t}}^{\mathcal{S}}(\boldsymbol{q}_{s,1}, \boldsymbol{q}_{s,2}), \beta_{\mathcal{S}}) \tag{10}$$

When predictions are accurate, then the optimal solution of the objective function described above should recover the underlying ground-truth. However, when the predictions possess noise, then the optimal object pose can drift from the underlying ground-truth. Our focus is local analysis, which seeks to understand the interplay between different objective terms defined by keypoints, edge vectors, and symmetry correspondences. Therefore, we assume the noise level of the input is small, and the perturbation of the output is well captured by low-order Taylor expansion of the output.

Our goal is to characterize the relation between the variance of the input noise and the variance of the output pose. We show that incorporating edge vectors and symmetry correspondences generally help to reduce the variance of the output.

The remainder of this section is organized as follows. In Section 3.1, we provide a local stability analysis framework for regression problems. In Section 3.2, we describe the structure of the pose regression and apply this framework to provide a preliminary analysis of the stability of pose regression. In Section 3.3, we provide further analysis on a specific example, which indicates the interactions among keypoints, edge vectors, and symmetry correspondences. Finally, Section 3.4 provide proofs of the propositions in this analysis.

3.1. Local Stability Analysis Framework

We begin with a general result regarding an optimization problem of the following form

$$\boldsymbol{x}^{\star}(\boldsymbol{y}) := \underset{\boldsymbol{y}}{\text{argmin}}\ f(\boldsymbol{x}, \boldsymbol{y}). \tag{11}$$

In the context of this paper, $\boldsymbol{y}$ encodes the noise associated with the predictions, i.e., keypoints, edge vectors, and symmetry correspondences. $\boldsymbol{x} \in \mathbb{R}^6$ provides a local parameterization of the output, i.e., the object pose. The specific expressions of $\boldsymbol{y}$ and $\boldsymbol{x}$ will be described in Section 3.2.

Without losing generality, we further assume that f satisfies the following assumptions (which are valid in the context of this paper):

- $f(\boldsymbol{x}, \boldsymbol{y}) \geq 0$. Moreover, $f(\boldsymbol{x}, \boldsymbol{y}) = 0$ if and only if $\boldsymbol{x} = 0$ and $\boldsymbol{y} = 0$. This means $\boldsymbol{x}^{\star}(\boldsymbol{0}) = \boldsymbol{0}$, and $(\boldsymbol{0}, \boldsymbol{0})$ is the strict global optimal solution.
- f is smooth and at least C^3 continuous.
- The following Hessian matrix is positive definite in some local neighborhood of $(\boldsymbol{0}, \boldsymbol{0})$:

$$\begin{bmatrix} \frac{\partial^2 f}{\partial^2 \boldsymbol{x}} & \frac{\partial^2 f}{\partial \boldsymbol{x} \partial \boldsymbol{y}} \\ \frac{\partial^2 f}{\partial \boldsymbol{x} \partial \boldsymbol{y}} & \frac{\partial^2 f}{\partial^2 \boldsymbol{y}} \end{bmatrix}.$$

Our analysis will utilize the following partial derivative of $\boldsymbol{x}^\star$ with respect to $\boldsymbol{y}$.

Proposition 2 *Under the assumptions described above, $\boldsymbol{x}^\star(\boldsymbol{y})$ is unique in the local neighborhood of $\mathbf{0}$, and*

$$\frac{\partial \boldsymbol{x}^\star}{\partial \boldsymbol{y}}(\boldsymbol{y}) := -\Big(\frac{\partial^2 f}{\partial^2 \boldsymbol{x}}(\boldsymbol{x}^\star(\boldsymbol{y}),\boldsymbol{y})\Big)^{-1}\frac{\partial^2 f}{\partial \boldsymbol{x}\partial \boldsymbol{y}}(\boldsymbol{x}^\star(\boldsymbol{y}),\boldsymbol{y}). \tag{12}$$

Proof. See Section 3.4.2. □

Since we are interested in local stability analysis, we assume the magnitude of $\boldsymbol{y}$ is small. Thus,

$$\boldsymbol{x}^\star(\boldsymbol{y}) \approx \frac{\partial \boldsymbol{x}^\star}{\partial \boldsymbol{y}}(\mathbf{0})\cdot \boldsymbol{y}. \tag{13}$$

If we further assume $\boldsymbol{y}$ follows some random distribution whose variance matrix if $\mathrm{Var}(\boldsymbol{y})$. Then the variance of the output $\boldsymbol{x}^\star$ is given by

$$\begin{aligned}&\mathrm{Var}(\boldsymbol{x}^\star(\boldsymbol{y}))\\ \approx&\Big(\frac{\partial^2 f}{\partial^2 \boldsymbol{x}}\Big)^{-1}\cdot\frac{\partial^2 f}{\partial \boldsymbol{x}\partial \boldsymbol{y}}\cdot \mathrm{Var}(\boldsymbol{y})\Big(\frac{\partial^2 f}{\partial \boldsymbol{x}\partial \boldsymbol{y}}\Big)^T\Big(\frac{\partial^2 f}{\partial^2 \boldsymbol{x}}\Big)^{-1}.\end{aligned} \tag{14}$$

Note that in our problem, f consists of non-linear least squares, i.e.,

$$f = \sum \frac{\beta_{i,1}^2\cdot \|\boldsymbol{r}_i\|_{\Sigma_i}^2}{\beta_{i,2}^2 + \|\boldsymbol{r}_i\|^2}. \tag{15}$$

The following proposition characterizes how to compute $\frac{\partial^2 f}{\partial^2 \boldsymbol{x}}$ and $\frac{\partial^2 f}{\partial \boldsymbol{x}\partial \boldsymbol{y}}$.

Proposition 3 *Under the expression described in (15), the second-order derivatives $\frac{\partial^2 f}{\partial^2 \boldsymbol{x}}$ and $\frac{\partial^2 f}{\partial \boldsymbol{x}\partial \boldsymbol{y}}$ at $(\mathbf{0},\mathbf{0})$ are given by*

$$\frac{\partial^2 f}{\partial^2 \boldsymbol{x}} = \sum_i \frac{\beta_{i,1}^2}{\beta_{i,2}^2}\frac{\partial \boldsymbol{r}_i}{\partial \boldsymbol{x}}\Sigma_i\frac{\partial \boldsymbol{r}_i}{\partial \boldsymbol{x}}^T \tag{16}$$

$$\frac{\partial^2 f}{\partial \boldsymbol{x}\partial \boldsymbol{y}} = \sum_i \frac{\beta_{i,1}^2}{\beta_{i,2}^2}\frac{\partial \boldsymbol{r}_i}{\partial \boldsymbol{x}}\Sigma_i\frac{\partial \boldsymbol{r}_i}{\partial \boldsymbol{y}}^T \tag{17}$$

Proof. See Section 3.4.3. □

3.2. Structure of Pose Stability

We begin by rephrasing the pose-regression problem described in the main paper.

Ground-truth setup. We use the same definition of variables as that in full paper. Recall that $\overline{\boldsymbol{p}}_k$ is coordinates of keypoint in canonical system. Let R^{gt} and $\boldsymbol{t}^{gt}$ be the ground-truth pose. Then the ground-truth 3D location of $\overline{\boldsymbol{p}}_k$ in the camera coordinate system is

$$\overline{\boldsymbol{p}}_k^{gt} = R^{gt}\overline{\boldsymbol{p}}_k + \boldsymbol{t}^{gt}, \qquad 1\le k\le|\mathcal{K}|.$$

Let $\overline{\boldsymbol{p}}_k^{gt} = (p_{k,x}^{gt,3D}, p_{k,y}^{gt,3D}, p_{k,z}^{gt,3D})^T$. Then the ground-truth image coordinates of the projected keypoint $\boldsymbol{p}_k^{gt}\in\mathbb{R}^2$ is given by

$$\boldsymbol{p}_k^{gt} = \Big(\frac{p_{k,x}^{gt,3D}}{p_{k,z}^{gt,3D}}, \frac{p_{k,y}^{gt,3D}}{p_{k,z}^{gt,3D}}\Big)^T = (p_{k,x}^{gt}, p_{k,y}^{gt})^T.$$

Likewise, recall $(\overline{\boldsymbol{q}}_{s1}, \overline{\boldsymbol{q}}_{s2})$ are symmetry correspondence in the world coordinate system, and let

$$\begin{aligned}\overline{\boldsymbol{q}}_{s1}^{gt} &:= R^{gt}\overline{\boldsymbol{q}}_{s1} + \boldsymbol{t}^{gt}\\ \overline{\boldsymbol{q}}_{s2}^{gt} &:= R^{gt}\overline{\boldsymbol{q}}_{s2} + \boldsymbol{t}^{gt}\end{aligned}$$

denote the transformed points in the camera coordinate system, where $\overline{\boldsymbol{q}}_{si}^{gt} = (q_{si,x}^{gt,3D}, q_{si,y}^{gt,3D}, q_{si,z}^{gt,3D})^T$. So the image coordinates of each symmetry correspondence are given by

$$\boldsymbol{q}_{s1}^{gt} = \Big(\frac{q_{s1,x}^{gt,3D}}{q_{s1,z}^{gt,3D}}, \frac{q_{s1,y}^{gt,3D}}{q_{s1,z}^{gt,3D}}\Big)^T, \quad \boldsymbol{q}_{s2}^{gt} = \Big(\frac{q_{s2,x}^{gt,3D}}{q_{s2,z}^{gt,3D}}, \frac{q_{s2,y}^{gt,3D}}{q_{s2,z}^{gt,3D}}\Big)^T.$$

Noise model. we proceed to describe the noise model used in the stability analysis. In this analysis, we assume each input keypoint is perturbed from the ground-truth location by $\boldsymbol{y}_k = (y_{k,x}, y_{k,y})^T$, i.e.,

$$\boldsymbol{p}_k = \boldsymbol{p}_k^{gt} + \boldsymbol{y}_k.$$

Likewise, we assume each input edge vector is perturbed from the ground-truth edge vector by $\boldsymbol{y}_e = (y_{e,x}, y_{e,y})^T$, i.e.,

$$\boldsymbol{v}_e = \boldsymbol{p}_{e_s}^{gt} - \boldsymbol{p}_{e_t}^{gt} + \boldsymbol{y}_e.$$

Finally, for symmetry correspondences, we assume that $\boldsymbol{q}_{s1}$ is not perturbed), and $\boldsymbol{q}_{s2}$ is perturbed by $\boldsymbol{y}_s = (y_{s,x}, y_{s,y})^T$, i.e.,

$$\boldsymbol{q}_{s1} = \boldsymbol{q}_{s1}^{gt}, \quad \boldsymbol{q}_{s2} = \boldsymbol{q}_{s2}^{gt} + \boldsymbol{y}_s.$$

Local parameterization. We parameterize the 6D object pose locally using exponential map with coefficients ($\boldsymbol{c}\in\mathbb{R}^3, \overline{\boldsymbol{c}}\in\mathbb{R}^3$), i.e.,

$$R = \exp(\boldsymbol{c}\times)\cdot R^{gt}, \quad \boldsymbol{t} = \boldsymbol{t}^{gt} + \overline{\boldsymbol{c}}.$$

Note this parameterization is quite standard for rigid transformations.

Now consider the three terms used in pose regression[3]:

$$\begin{aligned}\boldsymbol{r}_k^{\mathcal{K}} &:= \boldsymbol{p}_k - \mathcal{P}_{R,\boldsymbol{t}}(\overline{\boldsymbol{p}}_k),\\ \boldsymbol{r}_e^{\mathcal{E}} &:= \boldsymbol{v}_e - (\mathcal{P}_{R,\boldsymbol{t}}(\overline{\boldsymbol{p}}_{e_s}) - \mathcal{P}_{R,\boldsymbol{t}}(\overline{\boldsymbol{p}}_{e_t})),\\ r_s^{\mathcal{S}} &:= (\hat{\boldsymbol{q}}_{s,1}\times\hat{\boldsymbol{q}}_{s,2})^T R\overline{\boldsymbol{n}}_r.\end{aligned}$$

The following proposition characterizes the derivatives between each term and the parameters of the noise model and the parameters of the local parameterization.

[3]For convenience, we negate both $\boldsymbol{r}_k^{\mathcal{K}}$ and $\boldsymbol{r}_e^{\mathcal{E}}$ defined in the body of this paper.

Proposition 4 *Define*

$$J_{k,\boldsymbol{c}} = \begin{pmatrix} -p_{k,x}^{gt} p_{k,y}^{gt} & 1 + p_{k,x}^{gt\ 2} & -p_{k,y}^{gt} \\ -1 - p_{k,y}^{gt\ 2} & p_{k,x}^{gt} p_{k,y}^{gt} & p_{k,x}^{gt} \end{pmatrix}$$

$$J_{k,\overline{\boldsymbol{c}}} = \frac{1}{p_{k,z}^{gt,3D}} \begin{pmatrix} 1 & 0 & -p_{k,x}^{gt} \\ 0 & 1 & -p_{k,y}^{gt} \end{pmatrix}$$

The derivatives of $\boldsymbol{r}_{R,\boldsymbol{t}}^{\mathcal{K}}(\boldsymbol{p}_k) = \boldsymbol{r}_k^{\mathcal{K}}$ are given by

$$\left(\frac{\partial \boldsymbol{r}_k^{\mathcal{K}}}{\partial \boldsymbol{y}_k}, \frac{\partial \boldsymbol{r}_k^{\mathcal{K}}}{\partial \boldsymbol{c}}, \frac{\partial \boldsymbol{r}_k^{\mathcal{K}}}{\partial \overline{\boldsymbol{c}}}\right) = (I_2, -J_{k,\boldsymbol{c}}, -J_{k,\overline{\boldsymbol{c}}})$$

The derivatives of $\boldsymbol{r}_{R,\boldsymbol{t}}^{\mathcal{E}}(\boldsymbol{v}_e) = \boldsymbol{r}_e^{\mathcal{E}}$ are given by

$$\left(\frac{\partial \boldsymbol{r}_e^{\mathcal{E}}}{\partial \boldsymbol{y}_e}, \frac{\partial \boldsymbol{r}_e^{\mathcal{E}}}{\partial \boldsymbol{c}}, \frac{\partial \boldsymbol{r}_e^{\mathcal{E}}}{\partial \overline{\boldsymbol{c}}}\right) = (I_2, J_{e_t,\boldsymbol{c}} - J_{e_s,\boldsymbol{c}}, J_{e_t,\overline{\boldsymbol{c}}} - J_{e_s,\overline{\boldsymbol{c}}}).$$

Moreover, the derivatives of r_s^S are given by

$$\frac{\partial r_s^S}{\partial \boldsymbol{y}_s} = \begin{pmatrix} n_y^{gt} - n_z^{gt} q_{s1,y}^{gt} \\ -(n_x^{gt} - n_z^{gt} q_{s1,x}^{gt}) \end{pmatrix}^T, \tag{18}$$

$$\frac{\partial r_s^S}{\partial \boldsymbol{c}} = \left(\overline{\boldsymbol{n}}_r \times (\hat{\boldsymbol{q}}_{s1}^{gt} \times \hat{\boldsymbol{q}}_{s2}^{gt})\right)^T. \tag{19}$$

where $\overline{\boldsymbol{n}}^{gt} = R\overline{\boldsymbol{n}}_r = (n_x^{gt}, n_y^{gt}, n_z^{gt})^T$, $\hat{\boldsymbol{q}}_{si}^{gt}$ is homogeneous coordinate of $\boldsymbol{q}_{si}^{gt}$ normalized by camera intrinsic matrix.

Proof. See Section 3.4.1. □

Let $\boldsymbol{y}$ collect all the random variables in a vector. Let $J_{\mathcal{K}}$, $J_{\mathcal{E}}$, and $J_{\mathcal{S}}$ collect the Jacobi matrices for the predicted elements under each type in its column. Note that the size of $J_{\mathcal{S}}$ is 3×3 according to the derivations above. To facilitate the definition below, we reshape $J_{\mathcal{S}}$ as a 6×6 matrix by placing original elements to the upper-left corner, and zeros to elsewhere. Denote $\beta_{\mathcal{E}}$ and $\beta_{\mathcal{S}}$ as the weight in front of each term (without loss of generality, we set $\beta_{\mathcal{K}} = 1$). Then the variance matrix $\mathrm{Var}(\boldsymbol{c}, \overline{\boldsymbol{c}})$ can be approximated by

$$\mathrm{Var}(\boldsymbol{c}, \overline{\boldsymbol{c}}) \approx A^{-1} B \mathrm{Var}(\boldsymbol{y}) B^T A^{-1} \tag{20}$$

where

$$A := J_{\mathcal{K}} J_{\mathcal{K}}^T + \beta_{\mathcal{E}} J_{\mathcal{E}} J_{\mathcal{E}}^T + \beta_{\mathcal{S}} J_{\mathcal{S}} J_{\mathcal{S}}^T$$
$$B := (J_{\mathcal{K}}, \beta_{\mathcal{E}} J_{\mathcal{E}}, \beta_{\mathcal{S}} J_{\mathcal{S}})$$

It we consider $A^{-1} B \mathrm{Var}(\boldsymbol{y}) B^T A^{-1}$ as a function of $\beta_{\mathcal{E}}$ and $\beta_{\mathcal{S}}$ and compute its derivatives at $\beta_{\mathcal{E}} = 0$ and $\beta_{\mathcal{S}} = 0$, we obtain

$$\begin{aligned} &\frac{\partial A^{-1} B \mathrm{Var}(\boldsymbol{y}) B^T A^{-1}}{\partial \beta_{\mathcal{E}}} \\ :=& A^{-1}\big(J_{\mathcal{E}} \Sigma_{\mathcal{E}\mathcal{K}} J_{\mathcal{K}}^T + J_{\mathcal{K}} \Sigma_{\mathcal{K}\mathcal{E}} J_{\mathcal{E}}^T \\ &- J_{\mathcal{E}} J_{\mathcal{E}}^T A^{-1} J_{\mathcal{K}} \Sigma_{\mathcal{K}\mathcal{K}} J_{\mathcal{K}}^T - J_{\mathcal{K}} \Sigma_{\mathcal{K}\mathcal{K}} J_{\mathcal{K}}^T A^{-1} J_{\mathcal{E}} J_{\mathcal{E}}^T\big) A^{-1}. \end{aligned}$$

where $\Sigma_{\mathcal{K}\mathcal{K}}$ and $\Sigma_{\mathcal{E}\mathcal{K}}$ are the corresponding components in $\mathrm{Var}(\boldsymbol{y})$. This means whenever

$$\begin{aligned} &J_{\mathcal{E}} \Sigma_{\mathcal{E}\mathcal{K}} J_{\mathcal{K}}^T + J_{\mathcal{K}} \Sigma_{\mathcal{K}\mathcal{E}} J_{\mathcal{E}}^T \\ \prec& J_{\mathcal{E}} J_{\mathcal{E}}^T A^{-1} J_{\mathcal{K}} \Sigma_{\mathcal{K}\mathcal{K}} J_{\mathcal{K}}^T + J_{\mathcal{K}} \Sigma_{\mathcal{K}\mathcal{K}} J_{\mathcal{K}}^T A^{-1} J_{\mathcal{E}} J_{\mathcal{E}}^T, \end{aligned} \tag{21}$$

increasing the value of $\beta_{\mathcal{E}}$ from zero is guaranteed to obtain a positive reduction in the variance matrix (in terms of both the trace-norm and the spectral-norm).

(21) is satisfied when $\Sigma_{\mathcal{K}\mathcal{K}} = I$ and $\Sigma_{\mathcal{K}\mathcal{E}} = 0$. In general, when $\boldsymbol{y}_k$ and $\boldsymbol{y}_e$ are uncorrelated, then it is likely that increasing its value can lead to reduction in the output variance matrix.

A very similar argument can be applied to $\beta_{\mathcal{S}}$, and we omit the details for brevity.

3.3. An Example

We proceed to provide an example that explicitly shows how the variance of $\mathrm{Var}([\boldsymbol{c}, \overline{\boldsymbol{c}}])$ is reduced by incorporating edge vectors and symmetry correspondences. To this end, we consider a simple object that is given by a square, whose normal direction is along the z-axis in the camera coordinate system. We assume this square object has eight keypoints, whose z coordinates are all 1, i.e., $p_{k,z}^{gt,3D} = 1, 1 \leq k \leq 8$. Their x and y image coordinates are:

$$\boldsymbol{p}_1^{gt} = (\delta, \delta), \quad \boldsymbol{p}_2^{gt} = (\delta, 0), \quad \boldsymbol{p}_3^{gt} = (\delta, -\delta),$$
$$\boldsymbol{p}_4^{gt} = (0, \delta), \quad \boldsymbol{p}_5^{gt} = (0, -\delta), \quad \boldsymbol{p}_6^{gt} = (-\delta, \delta),$$
$$\boldsymbol{p}_7^{gt} = (-\delta, 0), \quad \boldsymbol{p}_8^{gt} = (-\delta, -\delta)$$

Moreover, assume that the normal to the reflection plane is $(1, 0, 0)$. The ground-truth symmetry correspondences are dense, and they are in the form of (x, y) and $(-x, y)$, where $0 \leq x \leq 1, -1 \leq y \leq 1$.

With this setup and after simple calculations, we have

$$H_{\mathcal{K}} := \frac{1}{8} \sum_{k=1}^{8} J_k^T J_k = \begin{pmatrix} c_1(\delta) & 0 & 0 & 0 & -c_2(\delta) & 0 \\ 0 & c_1(\delta) & 0 & c_2(\delta) & 0 & 0 \\ 0 & 0 & \frac{12\delta^2}{8} & 0 & 0 & 0 \\ 0 & c_2(\delta) & 0 & 1 & 0 & 0 \\ -c_2(\delta) & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{12\delta^2}{8} \end{pmatrix}$$

where $c_1(\delta) = \frac{8 + 12\delta^2 + 10\delta^4}{8}$, and $c_2(\delta) = \frac{8 + 6\delta^2}{8}$.

Likewise, we have

$$
\begin{aligned}
H_{\mathcal{E}} :=& \frac{1}{28}\sum_{e=1}^{28} J_e^T J_e \\
=& \begin{pmatrix}
\frac{11}{7}\delta^4 & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{11}{7}\delta^4 & 0 & 0 & 0 & 0 \\
0 & 0 & \frac{24}{7}\delta^2 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \frac{24}{7}\delta^2
\end{pmatrix}
\end{aligned}
$$

Finally, we have

$$
\begin{aligned}
H_{S} :=& [\int_0^1 (\int_{-1}^1 J_s^T J_s dy) dx, 0; 0, 0] \\
=& \begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{4}{3}\delta^2 & 0 & 0 & 0 & 0 \\
0 & 0 & \frac{4}{9}\delta^4 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}
\end{aligned}
$$

We proceed to assume the following noise model for the input:

$$
\mathrm{Var}(\boldsymbol{y}_k) = \sigma_{\mathcal{K}}^2 I_2, \quad \mathrm{Var}(\boldsymbol{y}_e) = \sigma_{\mathcal{E}}^2 I_2, \quad \mathrm{Var}(y_s) = \sigma_{\mathcal{S}}^2.
$$

In other words, noises in different predictions are independent.

Applying Prop. 3.4.3, we have that

$$
\begin{aligned}
& \mathrm{Var}([\boldsymbol{c}, \overline{\boldsymbol{c}}]) \\
\approx & (H_{\mathcal{K}} + \lambda H_{\mathcal{E}} + \mu H_{\mathcal{S}})^{-1} \cdot \\
& (\sigma_{\mathcal{K}}^2 H_{\mathcal{K}} + \lambda^2 \sigma_{\mathcal{E}}^2 H_{\mathcal{E}} + \mu^2 \sigma_{\mathcal{S}}^2 H_{\mathcal{S}}) \cdot \\
& (H_{\mathcal{K}} + \lambda H_{\mathcal{E}} + \mu H_{\mathcal{S}})^{-1} \\
= & \begin{pmatrix}
a_1 & 0 & 0 & 0 & a_2 & 0 \\
0 & a_3 & 0 & a_4 & 0 & 0 \\
0 & 0 & a_5 & 0 & 0 & 0 \\
0 & a_4 & 0 & a_6 & 0 & 0 \\
a_2 & 0 & 0 & 0 & a_7 & 0 \\
0 & 0 & 0 & 0 & 0 & a_8
\end{pmatrix}
\end{aligned} \tag{22}
$$

$a_i, 1 \leq i \leq 8$ are functions of $\delta, \beta_{\mathcal{E}}, \beta_{\mathcal{S}}$ and $\sigma_{\mathcal{K}}^2, \sigma_{\mathcal{E}}^2, \sigma_{\mathcal{S}}^2$. For simplify, we only analyze a_8, which is

$$
a_8 = \frac{\sigma_{\mathcal{K}}^2 \frac{12}{8}\delta^2 + \beta_{\mathcal{E}}^2 \sigma_{\mathcal{E}}^2 \frac{24}{7}\delta^2}{\left(\frac{12}{8}\delta^2 + \frac{24}{7}\beta_{\mathcal{E}}\delta^2\right)^2}
$$

It is easy to check that to minimize a_8, the optimal value for $\beta_{\mathcal{E}}$ is given by

$$
\beta_{\mathcal{E}} = \frac{\sigma_{\mathcal{K}}^2}{\sigma_{\mathcal{E}}^2}.
$$

In other words, incorporating edge vectors is helpful for reducing the velocity of the third dimension of the rotational component.

Similar analysis can be done for other a_i. As the rationale is similar, we omit them for brevity.

Contributions of keypoints, edge vectors, and symmetry correspondences. It is very interesting to study the structure of (22). First of all, all elements are relevant to keypoints. Edge vectors provide full constraints on the underlying rotation. Symmetry correspondences also provide constraints on two dimensions of the underlying rotation. However, by analyzing the structure of $H_{\mathcal{E}}$ and $H_{\mathcal{K}}$, one can see that they do not provide constraints on two dimensions of the underlying translation (albeit on this simple model). This explains why only using edge vectors and symmetry correspondences leads to poor results on object translations.

3.4. Proof of Propositions

3.4.1 Proof of Proposition 4

Derivatives of $\boldsymbol{r}_k^{\mathcal{K}}$ and $\boldsymbol{r}_e^{\mathcal{E}}$. It is straightforward to compute the derivatives of $\boldsymbol{r}_k^{\mathcal{K}}$ and $\boldsymbol{r}_e^{\mathcal{E}}$ with respect to $\boldsymbol{y}_k$ and $\boldsymbol{y}_e$, respectively. In the following, we focus on the derivatives of $\boldsymbol{r}_k^{\mathcal{K}}$ with respect to $(\boldsymbol{c}, \overline{\boldsymbol{c}})$. The derivatives of $\boldsymbol{r}_e^{\mathcal{E}}$ can be obtained by subtracting those of $\boldsymbol{r}_{e_s}^{\mathcal{K}}$ and those of $\boldsymbol{r}_{e_t}^{\mathcal{K}}$.

Recall the local parameterization $R = \exp(\boldsymbol{c}\times)R^{gt}$ and $\boldsymbol{t} = \boldsymbol{t}^{gt} + \overline{\boldsymbol{c}}$. We have

$$
\begin{aligned}
\frac{\partial \overline{\boldsymbol{p}}_k}{\partial(\boldsymbol{c}, \overline{\boldsymbol{c}})} =& \frac{\partial(\overline{\boldsymbol{p}}_k^{gt} + \boldsymbol{c} \times \overline{\boldsymbol{p}}_k^{gt} + \overline{\boldsymbol{c}})}{\partial(\boldsymbol{c}, \overline{\boldsymbol{c}})} \\
=& (-\overline{\boldsymbol{p}}_k^{gt} \times, I_3).
\end{aligned}
$$

Using chain rule, we have

$$
\begin{aligned}
\frac{\partial \boldsymbol{r}_k^{\mathcal{K}}}{\partial(\boldsymbol{c}, \overline{\boldsymbol{c}})} =& -\frac{1}{p_{k,z}^{gt,3D}} \left(\begin{pmatrix} \frac{\partial p_{k,x}^{3D}}{\partial(\boldsymbol{c},\overline{\boldsymbol{c}})} \\ \frac{\partial p_{k,y}^{3D}}{\partial(\boldsymbol{c},\overline{\boldsymbol{c}})} \end{pmatrix} - \begin{pmatrix} p_{k,x} \\ p_{k,y} \end{pmatrix} \cdot \frac{\partial p_{k,z}^{3D}}{\partial(\boldsymbol{c}, \overline{\boldsymbol{c}})} \right) \\
=& -\begin{pmatrix} 0 & 1 & -p_{k,y}^{gt} & \frac{1}{p_{k,z}^{gt,3D}} & 0 & 0 \\ -1 & 0 & p_{k,x}^{gt} & 0 & \frac{1}{p_{k,z}^{gt,3D}} & 0 \end{pmatrix} \\
& + \begin{pmatrix} p_{k,x}^{gt} \\ p_{k,y}^{gt} \end{pmatrix} \cdot \begin{pmatrix} p_{k.y}^{gt} & -p_{k,x}^{gt} & 0 & 0 & 0 & \frac{1}{p_{k,z}^{gt,3D}} \end{pmatrix}
\end{aligned}
$$

Derivatives of r_s. Again using chain rule, we have

$$
\frac{\partial r_s^S}{\partial \boldsymbol{y}_s} = \begin{pmatrix} (\hat{\boldsymbol{q}}_{s1}^{gt} \times \boldsymbol{e}_1)^T \boldsymbol{n}^{gt} \\ (\hat{\boldsymbol{q}}_{s1}^{gt} \times \boldsymbol{e}_2)^T \boldsymbol{n}^{gt} \end{pmatrix} = \begin{pmatrix} n_y^{gt} - n_z^{gt} q_{s1,y}^{gt} \\ -(n_x^{gt} - n_z^{gt} q_{s1,x}^{gt}) \end{pmatrix}
$$

Moreover,

$$
\frac{\partial r_s^S}{\partial \boldsymbol{c}} = \frac{\partial \det((\hat{\boldsymbol{q}}_{s1}^{gt} \times \hat{\boldsymbol{q}}_{s2}^{gt}, \boldsymbol{c}, \boldsymbol{n}))}{\partial \boldsymbol{c}} = \boldsymbol{n} \times (\hat{\boldsymbol{q}}_{s1}^{gt} \times \hat{\boldsymbol{q}}_{s2}^{gt}).
$$

3.4.2 Proof of Proposition 2

Proof: First of all, any optimal solution $\boldsymbol{x}^\star(\boldsymbol{y})$ is a critical point of f. Therefore, it shall satisfy:

$$\frac{\partial f}{\partial \boldsymbol{x}}(\boldsymbol{x}^\star, \boldsymbol{y}) = 0. \tag{23}$$

Consider a neighborhood, where $\|\boldsymbol{y}\| \leq \epsilon_1$, and $\|\boldsymbol{x}\| \leq \epsilon_2$. ϵ_2 is chosen so that it contains for each $\boldsymbol{y}$, the critical point with the smallest norm. Assume that $\frac{\partial^2 f}{\partial^2 \boldsymbol{x}}$ is positive semidefinite in this neighborhood.

By contradiction, suppose there exists two distinctive local minimums $\boldsymbol{x}_1(\boldsymbol{y})$ and $\boldsymbol{x}_1(\boldsymbol{y})$ for a given $\boldsymbol{y}$, i.e.,

$$\frac{\partial f}{\partial \boldsymbol{x}}(\boldsymbol{x}_1(\boldsymbol{y}), \boldsymbol{y}) = \frac{\partial f}{\partial \boldsymbol{x}}(\boldsymbol{x}_2(\boldsymbol{y}), \boldsymbol{y}) = 0. \tag{24}$$

Through integration, (24) yields

$$\begin{aligned} 0 &= \int_0^1 \frac{\partial^2 f}{\partial^2 \boldsymbol{x}}(\boldsymbol{x}_1 + t(\boldsymbol{x}_2 - \boldsymbol{x}_1), \boldsymbol{y})(\boldsymbol{x}_2 - \boldsymbol{x}_1)dt \\ &= \Big(\int_0^1 \frac{\partial^2 f}{\partial^2 \boldsymbol{x}}(\boldsymbol{x}_1 + t(\boldsymbol{x}_2 - \boldsymbol{x}_1), \boldsymbol{y})dt\Big) \cdot (\boldsymbol{x}_2 - \boldsymbol{x}_1) \end{aligned}$$

Since the weighted sum of positive definite matrices is also positive definite. It follows that

$$\int_0^1 \frac{\partial^2 f}{\partial^2 \boldsymbol{x}}(\boldsymbol{x}_1 + t(\boldsymbol{x}_2 - \boldsymbol{x}_1), \boldsymbol{y})dt \succ 0.$$

In other words, it cannot have a zero eigenvalue, with non-zero eigenvector $\boldsymbol{x}_2 - \boldsymbol{x}_1$. In other words, the critical point is unique. Since the second order derivatives are positive definite, then each critical point is also a local minimum.

Computing the derivatives of (23) with respect to $\boldsymbol{y}$, we obtain

$$\frac{\partial^2 f}{\partial^2 \boldsymbol{x}} \cdot \frac{\partial \boldsymbol{x}^\star}{\partial \boldsymbol{y}} + \frac{\partial^2 f}{\partial \boldsymbol{x} \partial \boldsymbol{y}} = 0 \tag{25}$$

3.4.3 Proof of Proposition 3

The proof is straight-forward as $\boldsymbol{r}_i(\boldsymbol{0}, \boldsymbol{0}) = 0$, and

$$\begin{aligned} \frac{\partial\big(\frac{\beta_{i,1}\boldsymbol{r}_i}{\sqrt{\beta_{i,2}^2 + \|\boldsymbol{r}_i\|^2}}\big)}{\partial \boldsymbol{x}} &= \frac{\beta_{i,1}}{\beta_{i,2}} \cdot \frac{\partial \boldsymbol{r}_i}{\partial \boldsymbol{x}}, \\ \frac{\partial\big(\frac{\beta_{i,1}\boldsymbol{r}_i}{\sqrt{\beta_{i,2}^2 + \|\boldsymbol{r}_i\|^2}}\big)}{\partial \boldsymbol{y}} &= \frac{\beta_{i,1}}{\beta_{i,2}} \cdot \frac{\partial \boldsymbol{r}_i}{\partial \boldsymbol{y}}. \end{aligned}$$

References

[1] Stefan Hinterstoisser, Vincent Lepetit, Slobodan Ilic, Stefan Holzer, Gary Bradski, Kurt Konolige, and Nassir Navab. Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes. In *Proceedings of the 11th Asian Conference on Computer Vision - Volume Part I*, ACCV'12, pages 548–562, Berlin, Heidelberg, 2013. Springer-Verlag. 1

[2] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Epnp: An accurate o (n) solution to the pnp problem. *International journal of computer vision*, 81(2):155, 2009. 2