NeRF-IBVS: Visual Servo Based on NeRF for Visual Localization and Navigation
- Supplementary Material -

Yuanze Wang[1,2*] **Yichao Yan**[1*] **Dianxi Shi**[1,2†] **Wenhan Zhu**[1] **Jianqiang Xia**[2,3]
Tan Jeff[3] **Songchang Jin**[2] **Ke Gao**[4] **Xiaobo Li**[4] **Xiaokang Yang**[1]
[1]MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University
[2]Intelligent Game and Decision Lab (IGDL), Beijing, China
[3]Tianjin Artificial Intelligence Innovation Center [4]Alibaba Group
{yz.wang,yanyichao,zhuwenhan823,xkyang}@sjtu.edu.cn
dxshi@nudt.edu.cn,jianqiang.xia@foxmail.com,jsc04@tsinghua.org.cn
{xiaobo.lixb,gaoke.gao}@alibaba-inc.com

1 Experimental Details

1.1 Normalized 3D Label Generation Details

For 3D labels used to train coordinate regression network, only the Office5a and Office5b in the 12-Scenes dataset uses a cube with 10 meters for normalization of 3D labels, other scenes use a cube with 7 meters. To reduce the error of the 3D labeling, we crop out information within 40 pixels from the boundary.

1.2 Details of Pose Optimization and Correspondence Selection

To accelerate RABNSAC [2], we uniformly sample 40 correspondences when more correspondences existed after the coordinate distance filtering. At each iteration of the RANSAC, a set of four correspondences is randomly sampled to launch the IBVS module to obtain the optimized pose. The optimized pose is then utilized to project all correspondences from the rendered image to the query image, thereby allowing the coordinate distance of the correspondences to be calculated. Correspondences with a coordinate distance of less than 3 pixels are considered inliers. The optimized pose corresponding to the most inliers is selected as the best-optimized pose. Meanwhile, the correspondences corresponding to the best-optimized pose are selected as the appropriate correspondences for IBVS, which are accurate and non-collinear.

2 Revisit of Image-based Visual Servo (IBVS)

2.1 Basic Components of IBVS

The aim of IBVS [1] is to control the camera move from the current pose to the desired pose to minimize the error e of image features between the current image and the target image. Correspondences are used as image features in this paper. The error e of image features is typically defined by

$$\mathbf{e} = \mathbf{s} - \mathbf{s}^*, \tag{1}$$

where $\mathbf{s}$ denote image coordinates of correspondences in the current image (current pose $\hat{\mathbf{T}}$), $\mathbf{s}^*$ denote image coordinates of correspondences in the target image (desired pose $\mathbf{T}^*$) and is constant. Once $\mathbf{s}$ and $\mathbf{s}^*$ are selected, the design of controlling the camera move from the current pose to the

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

desired pose can be quite simple. The most straightforward approach is to design a velocity controller. To achieve this, we need to establish the relationship between the time variation of $\mathbf{s}$ (image feature velocity) and the camera velocity. Let the velocity of the camera be denoted by $\mathbf{v}_c = (\boldsymbol{v_c}, \boldsymbol{\omega_c})$, with $\boldsymbol{v_c} = (v_x, v_y, v_z)$ the instantaneous linear velocity of the origin of the camera and $\boldsymbol{\omega_c} = (\omega_x, \omega_y, \omega_z)$ the instantaneous angular velocity of the camera. The relationship between $\dot{\mathbf{s}}$ and $\mathbf{v}_c$ is given by:

$$\dot{\mathbf{s}} = \mathbf{L}\mathbf{v}_c, \tag{2}$$

where $\mathbf{L} \in \mathbb{R}^{k\times 6}$ is named the jacobian matrix related to $\mathbf{s}$. Using (1) and (2), we immediately obtain:

$$\dot{\mathbf{e}} = \mathbf{L}\mathbf{v}_c. \tag{3}$$

Considering $\mathbf{v}_c$ as the input to the camera controller, we can aim for an exponential decoupled decrease of the error by setting $\dot{\mathbf{e}} = -\lambda\mathbf{e}$ (the desired image coordinate velocity of correspondences), we obtain using (3):

$$\mathbf{v}_c = -\lambda\mathbf{L}^{+}\mathbf{e}, \tag{4}$$

where $\lambda, \mathbf{L}^{+} \in \mathbb{R}^{6\times k}$ denote proportional factor and generalized inverse matrix of $\mathbf{L}$. Then, we update the current pose by integrating the camera velocity $\mathbf{v}_c$ during unit iteration to obtain the variation of pose $\Delta\mathbf{T}(\mathbf{v}_c(k))$. Finally, the current pose is updated as follows:

$$\hat{\mathbf{T}}(k+1) = \hat{\mathbf{T}}(k)\Delta\mathbf{T}(\mathbf{v}_c(k)). \tag{5}$$

The IBVS iteration proceeds until either the pixel coordinate error of correspondences is less than a certain threshold or the maximum number of iterations is reached.

2.2 Derivation for Jacobian Matrix of IBVS

We project a 3D point with coordinates $\mathbf{X} = (X, Y, Z)$ in the camera frame on the image plane, which obtains a 2D point with coordinates $\mathbf{n} = (x, y)$. Then we have:

$$x = X/Z,\ y = Y/Z, \tag{6}$$

By differentiating the projection equations (6) with respect to time, we obtain:

$$\begin{cases} \dot{x} = \dot{X}/Z - X\dot{Z}/Z^2 = (\dot{X} - x\dot{Z})/Z \\ \dot{y} = \dot{Y}/Z - Y\dot{Z}/Z^2 = (\dot{Y} - y\dot{Z})/Z. \end{cases} \tag{7}$$

We can establish the relationship between the velocity of the 3D point and the camera velocity using the well-known equation [1]:

$$\dot{\mathbf{X}} = -\boldsymbol{v_c} - \boldsymbol{\omega_c} \times \mathbf{X} \Leftrightarrow \begin{cases} \dot{X} = -v_x - \omega_y Z + \omega_z Y \\ \dot{Y} = -v_y - \omega_z X + \omega_x Z \\ \dot{Z} = -v_z - \omega_x Y + \omega_y X. \end{cases} \tag{8}$$

Injecting (8) in (7), and organizing terms we obtain:

$$\begin{cases} \dot{x} = -\dfrac{v_x}{Z} + x\dfrac{v_z}{Z} + xy\omega_x - (1+x^2)\omega_y + y\omega_z \\ \dot{y} = -\dfrac{v_y}{Z} + y\dfrac{v_z}{Z} + (1+y^2)\omega_x - xy\omega_y - x\omega_z, \end{cases} \tag{9}$$

which can be written as:

$$\dot{\mathbf{n}} = \mathbf{L}\mathbf{v}_c, \tag{10}$$

where the jacobian matrix $\mathbf{L}$ related to $\mathbf{n}$ is

$$\mathbf{L} = \begin{bmatrix} -\frac{1}{Z} & 0 & \frac{x}{Z} & xy & -(1+x^2) & y \\ 0 & -\frac{1}{Z} & \frac{y}{Z} & 1+y^2 & -xy & -x \end{bmatrix}. \tag{11}$$

The value Z in the matrix $\mathbf{L}$ represents the point's depth with respect to the camera frame. Therefore, traditional IBVS techniques that leverage this particular form of the Jacobian matrix must estimate or approximate the value of Z. We take $\mathbf{s} = \mathbf{n} = (x, y)$ in IBVS, the image coordinates of the correspondences.

2.3 Correspondence Selection for IBVS

To control the 6 DOF velocity of the camera, at least three correspondences are necessary according to the equation (4) (i.e., we require $k \geq 6$). If we merely stack jacobian matrices for three correspondences $\mathbf{s} = (s_1, s_2, s_3)$, we obtain:

$$\mathbf{L} = \begin{bmatrix} \mathbf{L_{s_1}} \\ \mathbf{L_{s_2}} \\ \mathbf{L_{s_3}} \end{bmatrix}. \tag{12}$$

In certain configurations, the matrix $\mathbf{L}$ may be singular [4]. Furthermore, there are four unique camera poses where $\mathbf{e} = 0$, and it is impossible to differentiate them [3]. For these reasons, more than three non-collinear correspondences are usually selected. Moreover, the IBVS-based navigation [5] assumes that the correspondences selected can always be detected during the entire servo process. Therefore, the correspondences selected are required to remain in the camera's field of view.

3 Qualitative Results for Visual Localization

Our coarse pose estimation can obtain the poses near the target as shown in Figure 1(a), which provides a good starting point for pose optimization. The pose optimization trajectories in both 3D space and image plane show that optimized pose can accurately reach the desired pose from coarse pose as shown in Figure 1(b). Figure 1(c) shows that the images rendered using the optimized poses are accurately aligned with the query images. They demonstrate that our method can achieve accurate localization.

4 Simulation Results for Visual Navigation

We randomly selected a pair of images from the real 12-Scenes dataset with co-views to serve as the initial state and the desired state for navigation. We first use NeRF-IBVS to select appropriate correspondences between the initial state and the desired state as shown in Figure 2 (a) and (b). A 6-degree-of-freedom navigation simulation is then performed using the camera model. The navigation trajectories in both 3D space and image plane show that the camera can accurately reach the desired state from the initial state as shown in Figure 2 (c). Figure 2 (d) shows that the coordinate error of correspondences can converge. They demonstrate that our method can achieve accurate navigation.

5 Qualitative Results on Correspondence Selection

We qualitatively analyze the effectiveness of correspondence selection. Coordinate error and depth error of correspondences are generally prone to occur in areas with poor rendering quality. As shown in Figure 3, the correspondence selection tends to select correspondences that occur in areas with good rendering quality, which can enhance the robustness of pose estimation.

References

[1] Francois Chaumette and Seth Hutchinson. Visual servo control. i. basic approaches. *IEEE Robotics Automation Magazine*, 13(4):82–90, 2006.

[2] Ondrej Chum and Jiri Matas. Optimal randomized RANSAC. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(8):1472–1482, 2008.

[3] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981.

[4] E Recherche, En Informatique, Henri Michel, and Patrick Rives. Singularities in the determination of the situation of a robot effector from the perspective view of 3 points. 06 1993.

[5] Kunwu Zhang, Yang Shi, and Huaiyuan Sheng. Robust nonlinear model predictive control based visual servoing of quadrotor uavs. *IEEE/ASME Transactions on Mechatronics*, 26(2):700–708, 2021.

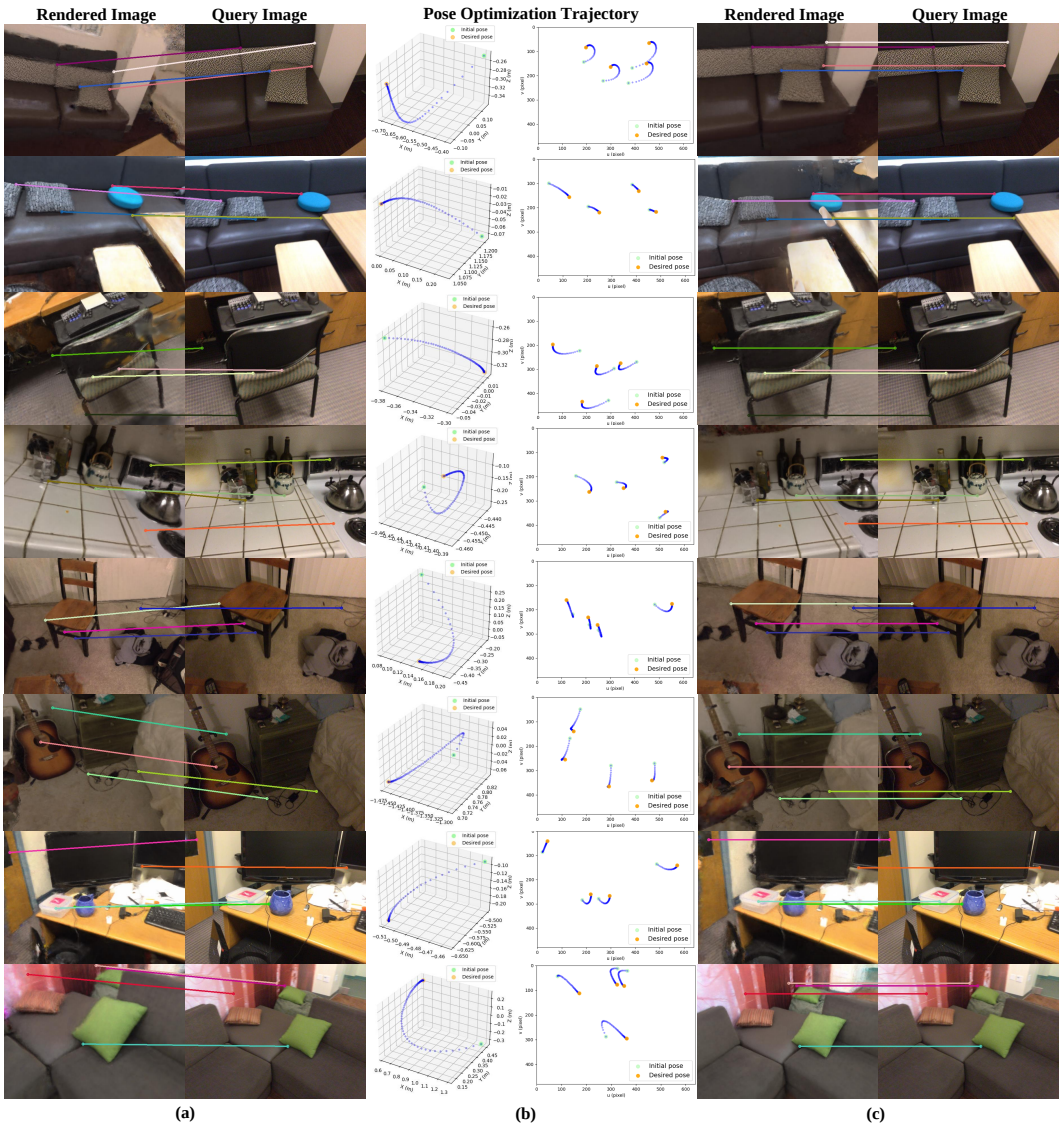

Figure 1: Qualitative results on 12-Scenes dataset. (a) the correspondences between images rendered using coarse poses and query images. (b) the pose optimization trajectories in 3D space and image plane. (c) the correspondences between images rendered using optimized poses and query images.

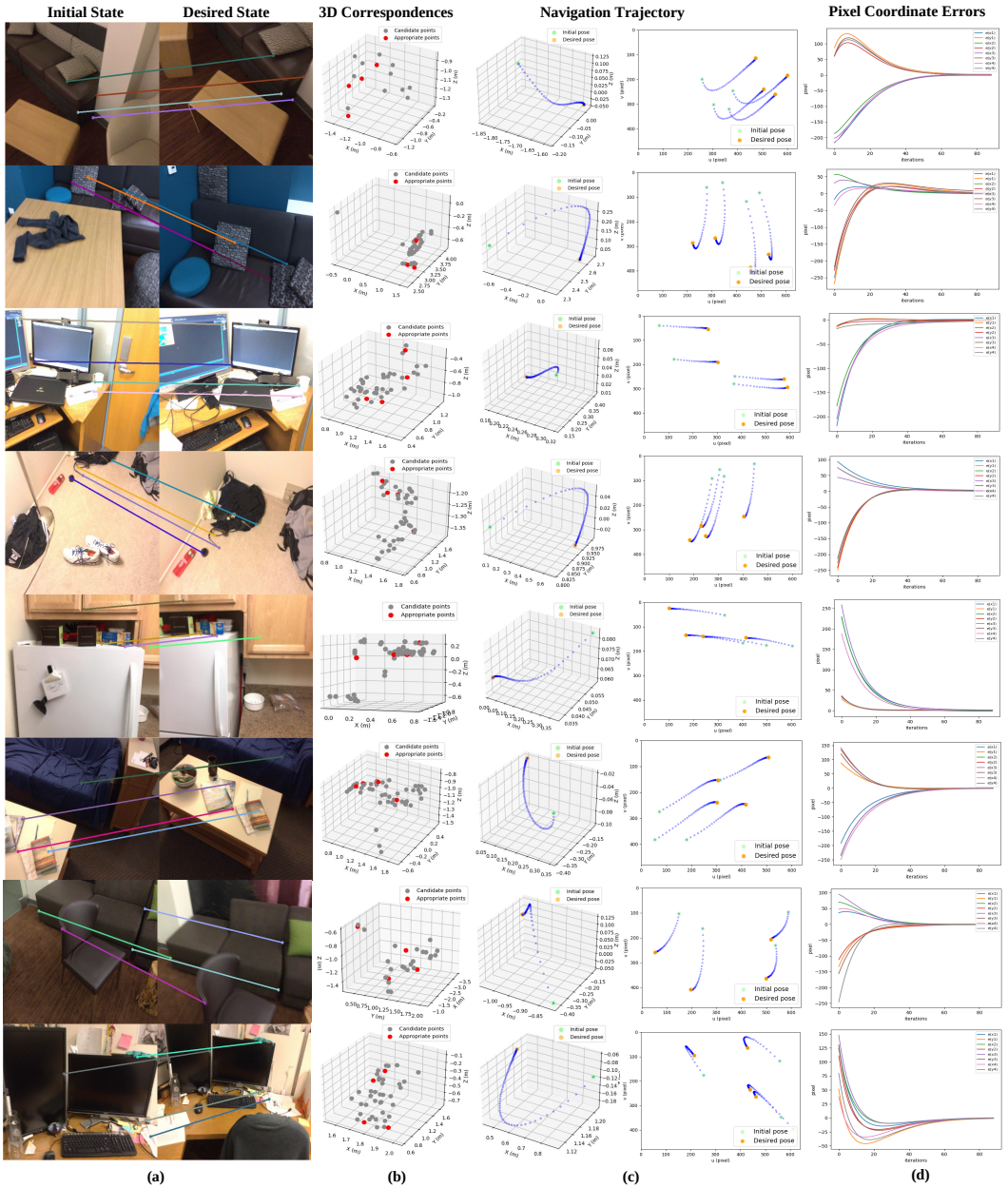

Figure 2: Navigation simulation on 12-Scenes dataset. (a) and (b) the appropriate correspondences between initial state and desired state are selected by NeRF-IBVS. (c) the navigation trajectories in 3D space and image plane. (d) the coordinate error of correspondences between the current view and the desired view during navigation.

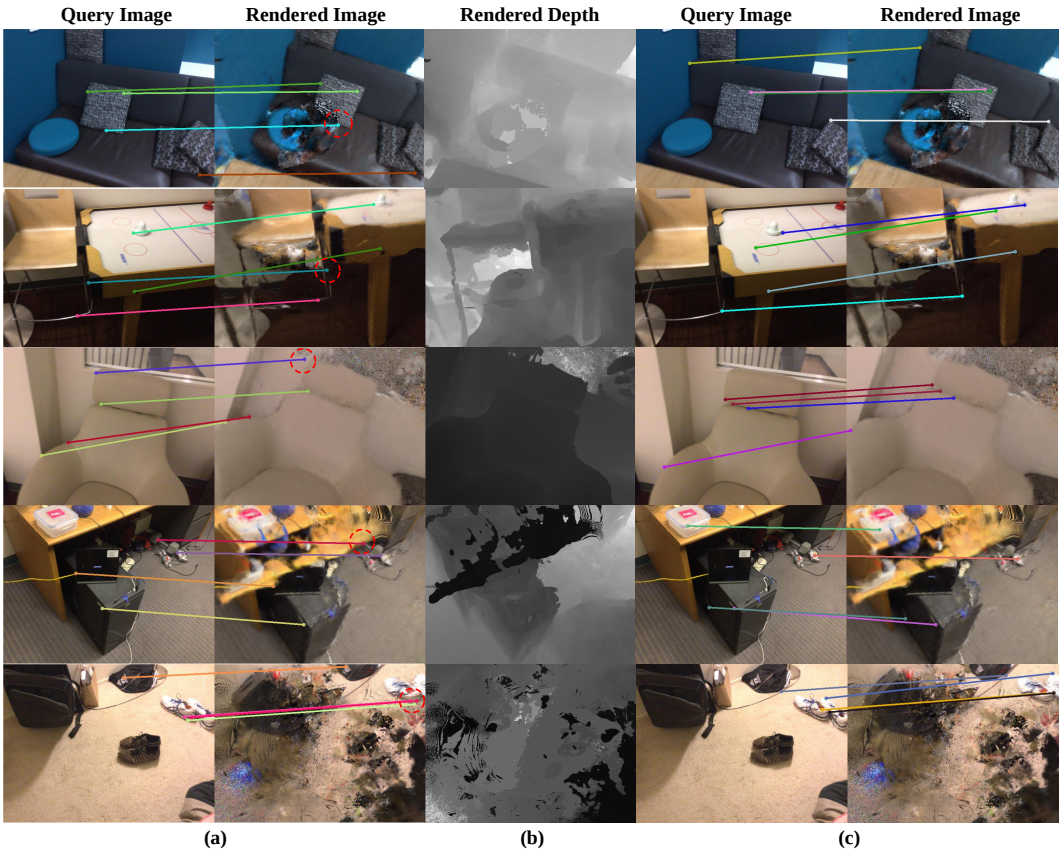

Figure 3: Qualitative comparison on correspondence selection. (a) the correspondences are randomly selected. The dashed red circles indicate the correspondences in the areas with poor rendering quality. (b) the rendered depth is used to indicate depth error of correspondences. (c) the correspondences are selected using correspondence selection.