# Supplementary Material for "TESO: Online Tracking of Essential Matrix by Stochastic Optimization"

## A.1. An ablation study: SIFT vs. SuperGlue and Kernel correlation vs. Non-robust loss

This experiment illuminates the robustness introduced by the kernel correlation (KC) principle, demonstrating that the selection of the keypoint detector, feature extractor, and matcher becomes non-critical once the kernelized loss function is used.

We have evaluated TESO performance on three different combinations of keypoints and loss functions:

1. SIFT, w/ KC, 5-NN (standard TESO, see Methods section of the paper) with

$$\mathcal{L}(\theta \,|\, \mathbf{X}, \mathbf{Y}) = -\sum_{\mathbf{x}\in\mathbf{X}} \sum_{\mathbf{y}\in \mathrm{NN}^1(\mathbf{x})} \exp\left[-\frac{\left(\mathbf{y}^\top \mathbf{E}(\theta)\mathbf{x}\right)^2}{2\sigma^2}\right] - \sum_{\mathbf{y}\in\mathbf{Y}} \sum_{\mathbf{x}\in \mathrm{NN}^0(\mathbf{y})} \exp\left[-\frac{\left(\mathbf{y}^\top \mathbf{E}(\theta)\mathbf{x}\right)^2}{2\sigma^2}\right], \tag{1}$$

2. SuperGlue [6] matches $(\mathbf{x}, \mathbf{y}) \in \mathrm{SG}$, w/o KC, i. e.:

$$\mathcal{L}(\theta \,|\, \mathrm{SG}) = -\sum_{(\mathbf{x},\mathbf{y})\in\mathrm{SG}} \left(\mathbf{y}^\top \mathbf{E}(\theta)\mathbf{x}\right)^2, \tag{2}$$

3. SuperGlue matches $(\mathbf{x}, \mathbf{y}) \in \mathrm{SG}$, w/ KC, i. e.:

$$\mathcal{L}(\theta \,|\, \mathrm{SG}) = -\sum_{(\mathbf{x},\mathbf{y})\in\mathrm{SG}} \exp\left[-\frac{\left(\mathbf{y}^\top \mathbf{E}(\theta)\mathbf{x}\right)^2}{2\sigma^2}\right]. \tag{3}$$

SuperGlue provides one-to-one matches with a specific confidence level. If the matcher works correctly, matches with higher confidence should have a higher probability of being inliers of the epipolar geometry. TESO tracking without a robust loss function (Eq. (2)) should therefore show better performance on keypoints with higher confidence levels. The kernel correlation (Eq. (3)) should robustify the results, ensuring that the tracking performance is less dependent on the quality of the matches.

In Fig. 1, we present the average rotational error (ARE) of TESO tracking on two sequences from the MAN TruckScenes dataset (around 160 frames each) under various scenarios. In ARE, lower values indicate better performance, i.e., TESO tracking is closer to the reference parameters (black line). The x-axis represents the minimum confidence level of the SuperGlue matches. SIFT has a constant value (green line) as it does not depend on the SG confidence. For the non-robust version of the loss (Eq. (2), red points), as the confidence of SuperGlue matches increases, the average rotational error decreases. The robust kernel correlation (Eq. (3)) on SuperGlue features (blue points) renders the confidence parameter unimportant and is always more precise than the non-robust variant (Eq. (2)). When TESO uses SIFT (with 5-NN matches, as proposed in the paper), it achieves slightly worse precision on one sequence (left plot) and slightly better precision on the other (right plot) than SuperGlue matches with a kernelized loss function (blue points).

These results suggest that selecting the keypoint detector, feature extractor, and matcher is not crucial for the kernelized loss function.

## A.2. Bias and latency evaluation

In this experiment, we demonstrate that the TESO tracker is unbiased and exhibits low latency in tracking.

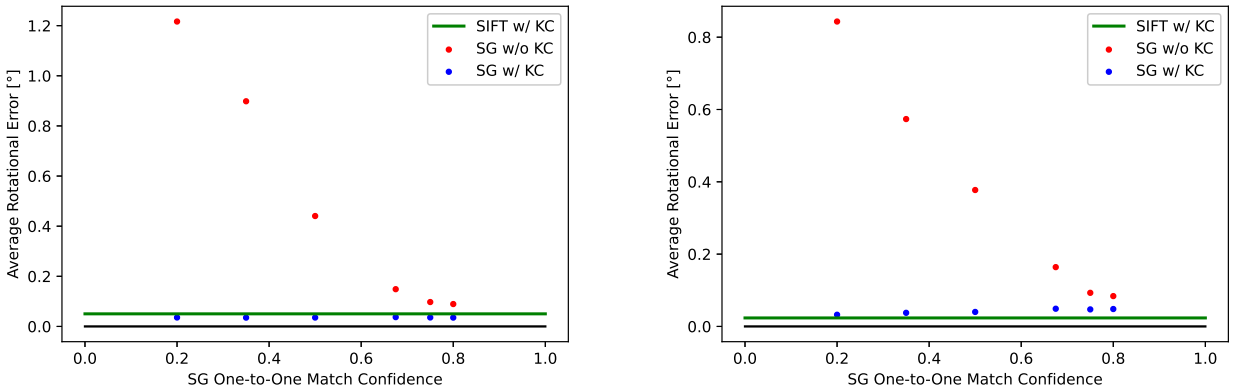



Figure 1. TESO performance visualized as an average rotational error on two sequences from the MAN TruckScenes dataset using different matching algorithms and loss functions. It is evident that with a higher confidence level in SuperGlue matches and a non-robust loss function (represented by the red points), the precision increases. However, using our proposed kernelized loss function with SuperGlue matches (blue points) renders the confidence hyper-parameter unimportant. The difference between SIFT (with 5-NN matching) and SuperGlue, using a kernelized loss function, is also very small, suggesting that the robust loss function renders the selection of keypoint detector and feature extractor uncritical. In all cases, the proposed robust loss function outperforms the non-robust variant.

Instead of examining the mean absolute error of the tracker in rotation (as in Section 5.3 of the paper), we evaluate the actual average tracked parameters on 149 calibrated sequences from the MAN TruckScenes dataset (with no simulated calibration drift). Here, we artificially extend the sequences by repeating them periodically ten times (this yields approximately 1600 frames per sequence). TESO achieved the following mean parameters over all sequences (with standard deviation):

| $R_x$ | $R_y$ | $R_z$ |
|---|---|---|
| $-0.0009$ ($\pm 0.017$) | $-0.0682$ ($\pm 0.110$) | $0.0143$ ($\pm 0.023$) |

As the parameter deviation from zero is not statistically significant, it suggests that the tracker is unbiased.

On the drifted sequences (cumulated drift of $\pm 0.01°$/DoF/frame), we evaluated the cross-correlation profile between the cumulated drift sequence and the TESO tracking sequence. Fig. 2 shows that the maximum correlation in all three degrees of freedom is zero (the sequences are discrete, with a unit of one frame). Although we have anticipated a more substantial latency (at least a frame), this result suggests that the mean latency is less than a frame in the TESO reaction to the cumulative drift. It is not possible to find a sub-frame latency estimate with this correlation method.

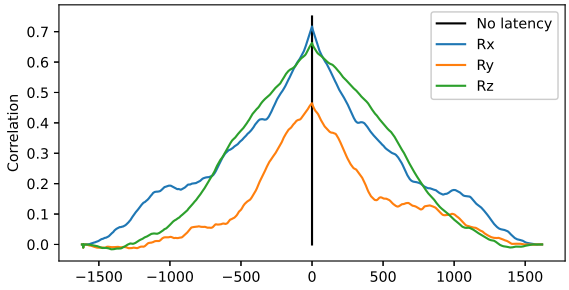



Figure 2. Latency evaluation on MAN TruckScenes dataset. It is estimated as a discrete cross-correlation between the TESO tracking sequence and the cumulative drift sequence for shifts in the range $(-1600, 1600)$. The maxima in all three degrees of freedom are at zero, which suggests the latency is less than one frame.

## A.3. Memory and time efficiency of TESO

### Memory

Throughout the stochastic optimization, TESO needs to temporally store the following variables:
- a vector of filtered gradients $\mathbf{g} \in \mathbb{R}^5$,
- a vector of filtered variance of gradients $\mathbf{v} \in \mathbb{R}^5$
- a vector of filtered diagonal of the Hessian matrix $\mathbf{h} \in \mathbb{R}^5$,
- memory size of the filter $\mathbf{m} \in \mathbb{R}^5$ and
- orthogonal matrices that make up the essential manifold $\mathbf{U}, \mathbf{V} \in \mathbb{R}^{3\times3}$.

That is, a total of 38 parameters.

### Time

We run a Python implementation of TESO on a desktop CPU AMD Ryzen 5 9600X. On one sequence from the CARLA–Drift dataset (with 1024×512 px resolution and extracting at most 1000 keypoints per frame), we achieved the following per-frame time efficiency:

| | Time [ms] |
|---|---|
| Keypoint detection and feature extraction [1, 5] | 70.16 |
| Nearest neighbors search [3] | 8.50 |
| Gradients and the diagonal of the Hessian matrix estimation | 7.24 |
| Filter and manifold update | 0.11 |
| **Total** | **86.01** |

The most time-consuming part, as expected, is keypoint detection and feature extraction. As shown in Sec. A.1, the use of a robust kernelized loss function renders the choice of a keypoint detector uncritical, allowing for the use of a more time-efficient variant. Our CPU implementation of TESO achieves a runtime similar to that of the fastest SotA method [4] (79 ms vs. ours 86 ms), whose matcher (41 ms vs. ours 79 ms) runs on a GPU and utilizes data-driven training.

## A.4. Keypoint detector selection

The keypoint detector is a building block of the TESO pipeline that may influence its precision and robustness. In this experiment, we have tested several kinds of keypoint detectors and feature extractors. You can see the TESO tracking average rotational error (ARE; lower is better) across three tested datasets in the table below. We did not find statistically significant differences between SIFT and BRISK across the datasets and all sequences; both performed best. This is consistent with the related work [7]. If a faster execution is needed, ORB can be used, but a drop in precision should be expected.

| | Average Rotational Error [°] | | | |
|---|---|---|---|---|
| | SIFT | BRISK | ORB | STAR + BRIEF64 |
| CARLA–Drift | **0.022** | **0.022** | 0.037 | 0.027 |
| KITTI, 00-01, [2] | **0.014** | **0.014** | 0.025 | 0.020 |
| MAN, 0.01° Drift | **0.049** | 0.052 | 0.062 | 0.057 |

## A.5. Hyper-parameters selection

Our method has two hyper-parameters – $\sigma$ and $k$. The parameter $\sigma$ represents a trade-off between the basin of attraction width and the variance of the tracked parameters; see the table below. In calibrated data (*CARLA–Drift, Calibrated* row), $\sigma$ increases the TESO variance, but it also helps to track the drift (*CARLA–Drift, 0.02° Drift* row). This depends on the expected decalibration speed for the setup and the sensor resolution. We have selected $\sigma = 0.001$ for CARLA–Drift and MAN TruckScenes dataset, which seems to be a good trade-off between the basin of attraction and process variance. For KITTI, we selected a slightly lower value of $\sigma = 0.00075$ due to a lower angular resolution of the pixels ($0.05°/\text{px}$) compared to CARLA ($0.068°/\text{px}$).

| | Average Rotational Error [°] | | | | |
|---|---|---|---|---|---|
| | $\sigma = 0.00025$ | $\sigma = 0.0005$ | $\sigma = 0.001$ | $\sigma = 0.002$ | $\sigma = 0.004$ |
| CARLA–Drift, Calibrated | 0.004 | 0.005 | 0.007 | 0.009 | 0.015 |
| CARLA–Drift, 0.02° Drift | 0.051 | 0.033 | 0.029 | 0.029 | 0.032 |

| | Data | Count | $R_x$ [°] | $R_y$ [°] | $R_z$ [°] | $T_x$ [mm] | $T_y$ [mm] | $T_z$ [mm] |
|---|---|---|---|---|---|---|---|---|
| | All | 149 | 0.014 (±0.01) | 0.115 (±0.07) | 0.024 (±0.02) | 1.0 (±0.7) | 2.5 (±2.7) | 7.1 (±5.3) |
| Daytime | Morning | 45 | 0.013 | 0.119 | 0.020 | 1.1 | 2.7 | 8.1 |
| | Noon | 97 | 0.016 | 0.106 | 0.023 | 0.9 | 2.4 | 6.2 |
| | Evening | 7 | 0.015 | 0.208 | 0.054 | 1.7 | 3.8 | 11.7 |
| Location | Terminal | 12 | 0.024 | 0.057 | 0.034 | 1.5 | 5.1 | 10.6 |
| | Highway | 103 | 0.014 | 0.131 | 0.024 | 1.0 | 2.4 | 7.2 |
| | Residential | 5 | 0.016 | 0.084 | 0.018 | 0.8 | 3.3 | 5.7 |
| | City | 7 | 0.015 | 0.054 | 0.027 | 0.4 | 1.6 | 3.1 |
| | Parking | 2 | 0.006 | 0.097 | 0.015 | 0.8 | 0.6 | 5.8 |
| | Rural | 20 | 0.011 | 0.097 | 0.017 | 0.9 | 1.9 | 6.2 |
| Weather | Clear | 64 | 0.015 | 0.120 | 0.026 | 1.0 | 2.6 | 6.9 |
| | Overcast | 52 | 0.010 | 0.100 | 0.014 | 0.9 | 1.6 | 6.6 |
| | Rain | 30 | 0.018 | 0.129 | 0.032 | 1.1 | 3.7 | 7.8 |
| | Fog | 1 | 0.027 | 0.078 | 0.037 | 0.6 | 9.4 | 5.3 |
| | Other | 2 | 0.013 | 0.129 | 0.051 | 1.9 | 4.9 | 13.0 |

(a) Calibrated

| | Data | Count | $R_x$ [°] | $R_y$ [°] | $R_z$ [°] | $T_x$ [mm] | $T_y$ [mm] | $T_z$ [mm] |
|---|---|---|---|---|---|---|---|---|
| | All | 149 | 0.021 (±0.01) | 0.126 (±0.07) | 0.032 (±0.02) | 1.3 (±0.8) | 4.4 (±2.9) | 9.4 (±5.6) |
| Daytime | Morning | 45 | 0.019 | 0.132 | 0.027 | 1.4 | 4.1 | 10.1 |
| | Noon | 97 | 0.021 | 0.119 | 0.031 | 1.3 | 4.4 | 9.1 |
| | Evening | 7 | 0.024 | 0.191 | 0.060 | 1.4 | 5.1 | 9.8 |
| Location | Terminal | 12 | 0.027 | 0.068 | 0.039 | 1.5 | 5.2 | 10.4 |
| | Highway | 103 | 0.021 | 0.142 | 0.032 | 1.4 | 4.4 | 9.7 |
| | Residential | 5 | 0.021 | 0.100 | 0.026 | 1.0 | 5.0 | 6.8 |
| | City | 7 | 0.019 | 0.079 | 0.034 | 0.8 | 2.9 | 5.7 |
| | Parking | 2 | 0.012 | 0.117 | 0.018 | 0.8 | 4.8 | 5.7 |
| | Rural | 20 | 0.017 | 0.106 | 0.025 | 1.4 | 3.9 | 9.8 |
| Weather | Clear | 64 | 0.021 | 0.126 | 0.033 | 1.1 | 3.9 | 8.0 |
| | Overcast | 52 | 0.016 | 0.109 | 0.021 | 1.5 | 3.9 | 10.6 |
| | Rain | 30 | 0.027 | 0.145 | 0.043 | 1.4 | 5.9 | 9.9 |
| | Fog | 1 | 0.038 | 0.441 | 0.058 | 1.6 | 8.0 | 12.1 |
| | Other | 2 | 0.020 | 0.135 | 0.058 | 2.7 | 5.1 | 17.8 |

(b) 0.01° calibration drift

Table 1. Full results of TESO on the MAN Dataset based on the time of the day, location, and weather of each sequence. It shows results for sequences with correct calibration (a) and for sequences with synthetic calibration drift in all rotational DoFs (b). Red shows a statistically significantly worse precision than the tracking results over all sequences.

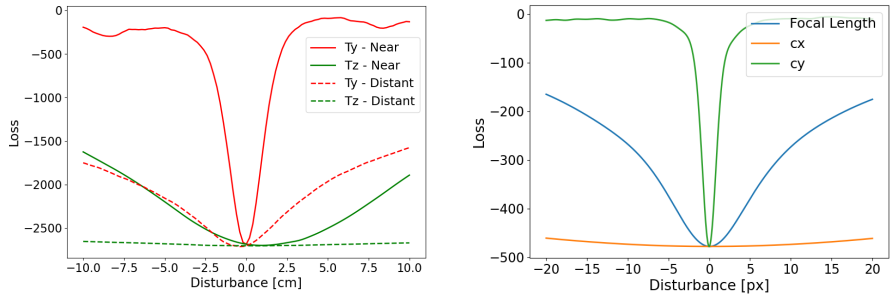
The parameter $k$ in kNN further robustifies the method for challenging (repetitive) scenes. In high-quality data, $k = 1$ will be the most precise; we have chosen $k = 5$, which is more conservative (a closer bound on the theoretical kernel correlation [8]), but may reduce geometric precision.

## A.6. Analysis of TESO results on the MAN TruckScenes dataset

To further analyze the quality of TESO tracking in challenging scenarios, we present results from the MAN dataset, broken down by daytime, location, and weather for each sequence, in Tab. 1. In both calibrated and drifted sequences, tracking is statistically significantly worse at night, in fog, or in tunnels (i.e., weather is 'Other'). This is expected behavior, as universal keypoint detectors (such as SIFT [5]) struggle under low-light conditions. Such scenarios likely require specialized, pre-trained keypoint detectors. A universal SIFT keypoint detector appears sufficient for any driving location, including rainy weather.

## A.7. Translation tracking precision

Throughout this work, we have focused on rotation precision only. This is because the translation is not as observable as the rotations in automotive scenarios, given the very distant scene. Shown in Fig. 3a is the kernel correlation loss evaluation: Narrow curves indicate a higher sensitivity of the loss function (y-axis) to decalibration (x-axis). Except in the Y translation in near scenes, this shows that achieving centimeter-level precision with feature-based optimization methods in typical driving scenes is difficult (as opposed to tight calibration rooms). Position calibrations obtained from vehicle drawings are inherently more accurate than those from online calibration methods.




(a) Near vs. Distant Scene

(b) Intrinsics (non-parallel stereo)

Figure 3. Kernel corelation loss evaluations. Narrower is better.

## A.8. Intrinsics calibration

Our experiments show that the focal length and the vertical position of the principal point are observable; see Fig. 3b. Note that the sensitivity of the focal length is quite high, five pixels are less than 1 % of the image size.

However, adding it to TESO is not straightforward, as these are not manifold parameters. It would require changing the optimization scheme used in TESO. See also the last paragraph of Section 5.3 of the paper.

## References

[1] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[2] I. Cvišić, I. Marković, and I. Petrović. Recalibrating the KITTI Dataset Camera Setup for Improved Odometry Accuracy. In *Eur. Conf. Mobile Robots (ECMR)*, 2021.

[3] M. Douze, A. Guzhva, C. Deng, et al. The Faiss Library. *IEEE Trans. Big Data*, 12(2):346–361, 2026.

[4] R. Gong, K. H. Yap, W. Liu, et al. Rectification-specific Supervision and Constrained Estimator for Online Stereo Rectification. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 22348–22358, 2025.

[5] D. G. Lowe. Object Recognition from Local Scale-Invariant Features. In *Int. Conf. Comput. Vis. (ICCV)*, pages 1150–1157, 1999.

[6] P. E. Sarlin, D. DeTone, T. Malisiewicz, et al. SuperGlue: Learning Feature Matching with Graph Neural Networks. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 4938–4947, 2020.

[7] S. A. K. Tareen and Z. Saleem. A Comparative Analysis of SIFT, SURF, KAZE, AKAZE, ORB, and BRISK. In *Int. Conf. Comput. Math. Eng. Technol. (iCoMET)*, pages 1–10, 2018.

[8] Y. Tsin and T. Kanade. A Correlation-Based Approach to Robust Point Set Registration. In *Eur. Conf. Comput. Vis. (ECCV)*, pages 558–569, 2004.