

To the Reviewer:

The authors would like to appreciate the Reviewer's highly relevant comments. The authors, upon further examination of the manuscript, understand the concerns expressed by the Reviewer and have attempted to address them. The Reviewer's original comments are listed below in **Bold and Italic** type, while the responses of the authors are typed in blue ink. Moreover, the revisions are highlighted by yellow color in the revised manuscript. For the Reviewer's quick reference, some revisions are also available in this report.

Comments of Reviewer:

Overall, the article presents a system with frame-to-frame ICP with IMU in the Kalman fusion scheme for UAV applications. I believe almost everything is presented nicely but the system is rather simple when compared to other VO/SLAM solutions and evaluated on short sequences (more is needed in that part of the article). Moreover, I have some concerns:

Reply: The authors are highly appreciative of the Reviewer's comments. Some extra experiments and verifications have been carried out for longer sequences and added to the revised version. The specific comments will be addressed below, point by point:

1. In the introduction, you distinguish between the optimization method and the ICP method. I believe that the wording is unfortunate as ICP is also based on optimization. I suggest direct or image-based optimization for that type of optimization.

Reply: The authors totally agree with the Reviewer that ICP is also based on optimization. The critical step of the ICP method and the direct optimization method is solving a minimization function which share similar form ($\operatorname{argmin}_{trans} \sum_j (model)$). e.g.

ICP (we presented)	minimize the distance of point cloud pairs $\operatorname{argmin}_{\Delta T} \sum_{j=0}^n \ p'_j - p_j \ ^2$
Feature (image) based	minimize the reprojection error

optimization	$\begin{cases} e_{i,j} = x_{i,j} - \pi_i(T, X_{w,j}) \\ \operatorname{argmin}_T \sum_{i,j} \rho_h(e_{i,j}^T \Omega_{i,j}^{-1} e_{i,j}) \end{cases}$ Ref: ORB-SLAM: a Versatile and Accurate Monocular SLAM System
Intensity (direct) based optimization	minimize intensity residual $\operatorname{argmin}_T \sum_i \left(I_{ref}(p_i) - I\left(\omega(p_i, D_{ref}(p_i), T)\right) \right)^2$ Ref: LSD-SLAM: Large-Scale Direct Monocular SLAM

While, comparing to the other two methods in the above table (i.e. the image-based or direct-based optimization method), the ICP is different in the workflow. The argument of the optimizer is set as transformation of camera (T) itself. Then the optimizer uses the gradient descent method to find out the optimal solution which minimizes the objective function. In the ICP workflow, the argument of the optimizer is the increment transformation (ΔT), and one-step Singular value decomposition (SVD) is used to find the optimal solution. Following that, it searches the nearest point to create the new point pairs for the next loop. ICP stops when ΔT is relatively small. Therefore, the authors believe that the current ICP as a special case of image-based optimization comparing to image and intensity (direct) based optimization. To avoid confusion, the authors elaborate more on the difference among the traditional image-based optimization method, the direct optimization method and the ICP method, and modify the relevant parts in introduction section as below.

Revised: Line 31 Page 2 in the Revised Manuscript:

The optimization method models the estimation task as a minimization problem [15], [16]. Generally, two kinds of optimization methods are widely used: Image-based optimization method and direct optimization method. In the traditional image-based optimization method, the objective function is modeled by the re-projection error of the features through feature matching; while in the direct optimization method, with the photometric camera model, the objective function is chosen as the intensity residual between frames. A comprehensive comparison of these two optimization methods can be found in Delmerico and Scaramuzza [17]. The target functions are solved by the gradient descent optimizer. As the image transformation in the traditional image-based method is represented by the Jacobian matrix in the objective function, which is the first order linearization of the real model, this method

performs poorly in scenarios involving significant rotation, due to the implicitly high non-linearity in the real model.

Apart from the traditional image-based optimization method, the ICP method is considered as a relatively new type of its kind. The ICP method relies on the classic registration workflow [18], where the ICP algorithm begins with two sequential point clouds and an initial prediction of the transformation. The correspondence between two images (i.e., the source point cloud and the target point cloud) is found by searching the closest neighborhoods of the source points and estimating the progressive transformation to minimize the distances between the source points and target points. The iteration process is continued until transformation convergence is achieved. The value of the initial guess for the ICP is important, as a poor guess may lead to the iterative solution becoming trapped in a local minimum. Also, the ICP require high computation power especially when the number of points increase.

2. How many ICP iterations do you perform? I assume that it does have to be much considering the motion prediction with IMU.

Reply: It varied from less than 3 (no motion) to 15 iterations. The authors have added a table (Table 5 in the revised manuscript) to show the statistical results of our algorithm. The relevant explanations will be given in the reply to the Reviewer's comment No. 3 below. Yes, the Reviewer is correct that the motion prediction of IMU provides the initial guess of the ICP algorithm.

Line 324 Page 16 in the Revised Manuscript:

In general, 3 (no motion) to 15 ICP loops were performed and the motion prediction of IMU provided the initial guess of the ICP algorithm. The refinement process continues until the corresponding transformation change (ΔT) is relatively small or the maximum loops (15 iterations in our experiments) are reached.

3. Your experimental section contains rather short sequences. Can you present the influence of some elements of your system on the finally obtained results? For example, what would be the result if the weights for points for the ICP were set to equal instead of t distribution?

Reply: Upon the Reviewer's suggestion, the authors have added the comparison of our ICP method (with the weights for points set to t distribution) and the conventional ICP method (with the weights for points set to equal) in detail.

Due to there is no existing public ToF-IMU dataset, we first conducted the handheld test and UAV test and compared the results with those of the motion capture system, which serve as the ground truth benchmarks. The accuracy of the odometry is presented by the root mean square error (RMSE) of translational drift. Both the absolute trajectory error (ATE) and the one-step relative pose error (RPE) cases are considered [26]. The definitions of ATE and RPE are shown below:

$$\mathbf{E}_{ATE,i} = \mathbf{T}_{gt,i}^{-1} \mathbf{S} \mathbf{T}_{est,i} \quad (40)$$

$$\mathbf{E}_{RPE,i} = \left(\mathbf{T}_{gt,i}^{-1} \mathbf{T}_{gt,i+1} \right)^{-1} \left(\mathbf{T}_{est,i}^{-1} \mathbf{T}_{est,i+1} \right) \quad (41)$$

$$RMSE(\mathbf{E}_{1:n}) = \left(\frac{1}{n} \sum_{i=1}^n \| trans(\mathbf{E}_i) \|^2 \right)^{\frac{1}{2}} \quad (42)$$

where $\mathbf{T}_{gt,i}$ is the transformation of ground truth of the frame i , $\mathbf{T}_{est,i}$ the estimate transformation of the frame i and $\mathbf{S}$ the least-squares solution that maps the estimated trajectory $\mathbf{T}_{est,1:n}$ onto the ground truth trajectory $\mathbf{T}_{gt,1:n}$. We then carried out a field test and compared the results with those obtained by RealSense T265 VIO sensors. The data is provided in rosbag and it is compatible with the TUM dataset [27].

5.1. Handheld Test

Based on this dataset, Figure 11 shows the comparison of ground truth trajectory with those using algorithms of the conventional ICP (icp), the ICP with the salient point selection criteria (icp+s), the ICP with the salient point selection criteria and robust weighting factor (icp+s+w), the random sub-sampled based ICP (icp+rd) and the ICP with the RANSAC pipeline (icp+r) [28]. The corresponding accuracy and processing time of each algorithm are also evaluated and listed in Table 5. It can be seen that the conventional ICP workflow can provide the accurate pose estimation but the processing time is unfavorable to the real-time applications. Both the random sub-sampling based ICP and the ICP with our salient point selection method can dramatically reduce processing time. Nevertheless, the former induced a large ATE error. The ICP with the RANSAC method achieves best in the RPE error but with the largest ATE error and a second largest average processing time. Notably, the conventional ICP, icp+s, icp+r, and icp+rd are all with weights for points set to equal. Together with the robust weighting factor (the weights for points set to the t distribution), the proposed method (icp+s+w) in this work reaches almost the same ATE accuracy and 26% improvement in RPE accuracy when compared with the conventional ICP. In the meanwhile, the processing time of icp+s+w is only 25% that

of the conventional ICP. Although, the average processing time of icp+s+w is a little bit higher than icp+s, the improvement of ATE and RPE errors compensates. The superiority of icp+s+w over the other methods, when considering the improvement of ATE and RPE errors and the average processing time simultaneously, is clearly illustrated.

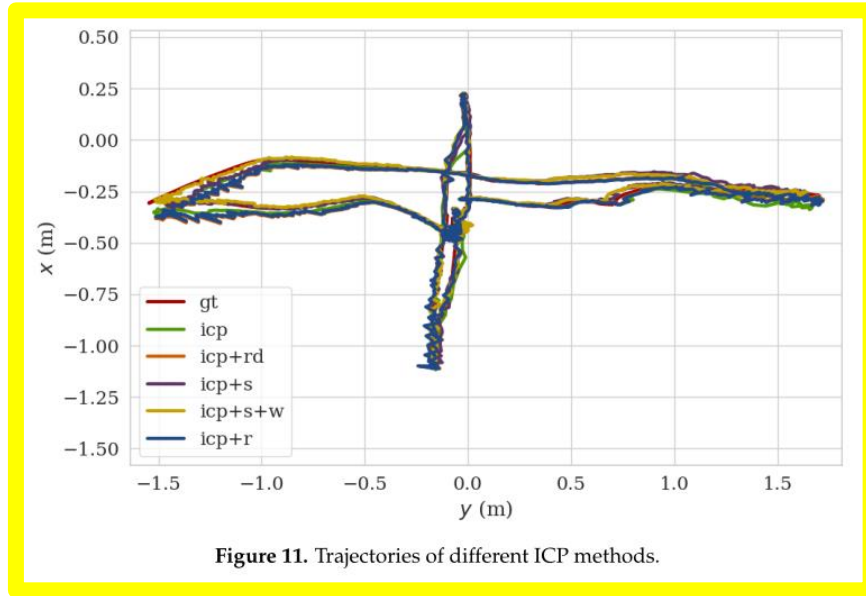


Table 5. Accuracy and processing time comparison of different methods

Method	Translation error (RMSE)				Average Processing Time(ms)	Relevant Processing time $\frac{t(x)-t(icp)}{t(icp)}$
	ATE (m)	improvement $\frac{ATE(x)-ATE(icp)}{ATE(icp)}$	RPE (m/s)	improvement $\frac{ATE(x)-ATE(icp)}{ATE(icp)}$		
icp	0.048	0	0.023	0	148	100%
icp+s	0.051	-6%	0.019	17%	28.0	18%
icp+s+w	0.047	2%	0.017	26%	36.2	25%
icp+rd	0.063	-31%	0.021	9%	20.4	14%
icp+r	0.070	-45%	0.015	28%	83.8	56%

Furthermore, we visualize the salient points in Figure 12 (a). Our selection algorithm selects the feature and the edge of the object from the raw input point cloud. The number of salient points is consistent (Figure 12(b)) even if the size of input cloud varies. Together with the good accuracy of icp+s and icp+s+w methods, it is believed that the salient point selection criteria is efficient and robust.

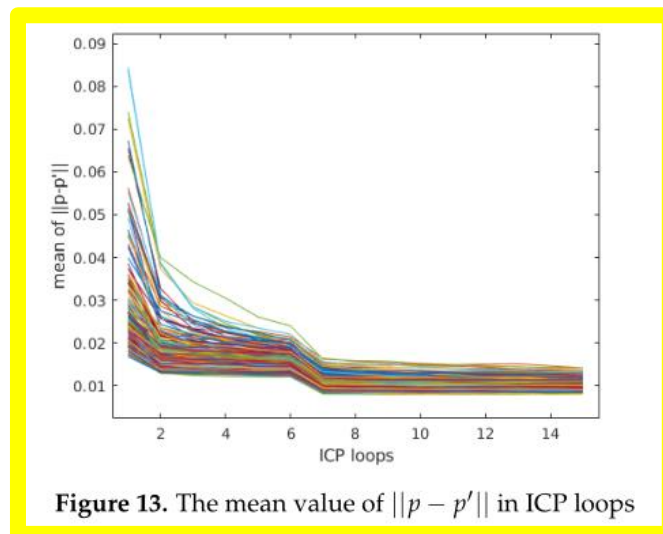
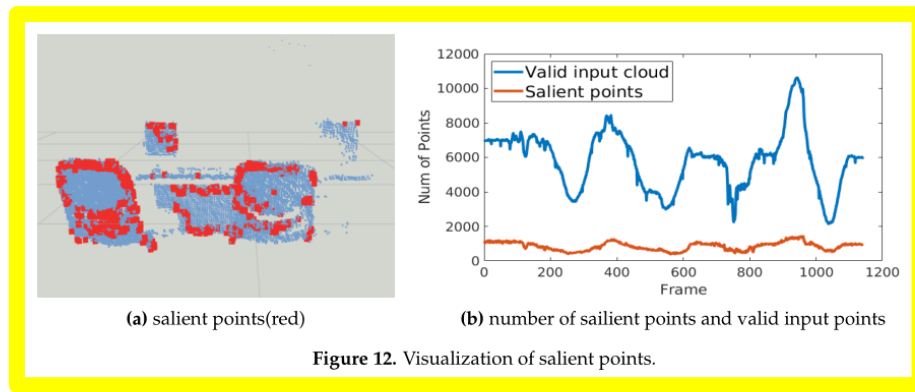


Figure 13 demonstrates that the error of mean $\|p - p'\|$ in each frame (different lines) decreases with the increase of ICP loops. In general, 3 (no motion) to 15 ICP loops were performed and the motion prediction of IMU provides the initial guess of the ICP algorithm. After about 10 loops, the mean $\|p - p'\|$ almost reach the minimum value. The refinement process continues until the corresponding transformation change (ΔT) is relatively small or the maximum loops (15 in our experiment) are reached.

4. What are the lengths and final position errors for the presented sequences? With these values, you could compute error as a percentage of the distance and thus indirectly compare to other systems.

Reply: The length of the trajectory and the final position error are list in the following table:

Dataset	Trajectory Length	Max ATE Error
Handheld Test	12.86m	0.13 m (compared with ground truth) (1.01%=0.13/12.86)
UAV Test in Lab	13.02m	0.11 m (compared with ground truth) (0.84%=0.11/13.02)
Indoor Test with a moving platform Link: https://youtu.be/lsl-8b6PmMI (added to video abstract link)	25.67m	0.82 (Compared with realsense T265 sensor) (3.19%=0.82/25.67)

The authors have added the detailed comparison of our method with the conventional ICP method and other improved methods using the handheld test dataset (see the reply to Reviewer's comment No. 3). The accuracy is present in ATE and RPE forms. Noted that in this work, the relevant keyframe function is incorporated into the frame-to-frame comparison in the Handheld test and the UAV test (please see the reply to the Reviewer's comment No. 6 below for details). The keyframe queues obtained in the repeated motion during the tests served as a map in the lab. Therefore, the accumulated errors are reduced, as shown in the above table. The error as a percentage of the distance is ~1%. On the other hand, in the indoor test with a moving platform, the trajectory is not repeated for a longer distance. Because no relevant keyframe function is applicable, the error increases. The error as a percentage of the distance is ~3%.

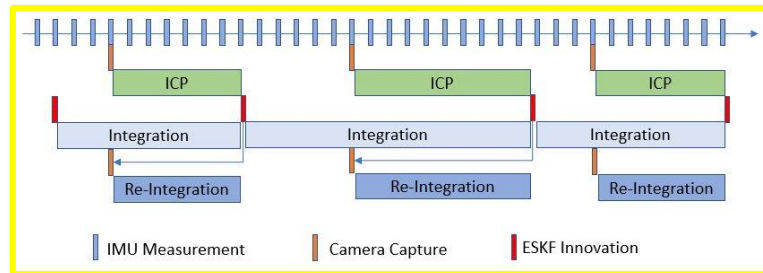
5. I believe "pre-integration" means something different to the community than what is shown in the article. I would say that you just perform "integration" and refer to the "pre-integration" when the linearization points are fixed and the new estimates are corrected based on the linearized terms as proposed by Forster in "IMU Preintegration on Manifold for Efficient Visual-Inertial Maximum-a-Posteriori Estimation". If you agree please change it especially in Fig. 7.

Reply: The Reviewer's comment is well taken with thanks. The authors agree with the Reviewer that the use of the "IMU Pre-integration" is not accurate in the original manuscript. Upon the Reviewer's suggestion, the authors have changed it to IMU integration.

Revised: Line 275 Page 13 in the Revised Manuscript:

4.5 Integration and Re-Integration

In the VIO system, the IMU data are updated at a high rate (i.e., >200 Hz for a typical MEMS IMU) while the camera capture rate is relatively slow (i.e., <30 Hz). In addition, the ICP processing time is dependent on the numerical convergence rate and is not constant. Therefore, we need to maintain the state queue and update it properly using the integration/re-integration method.



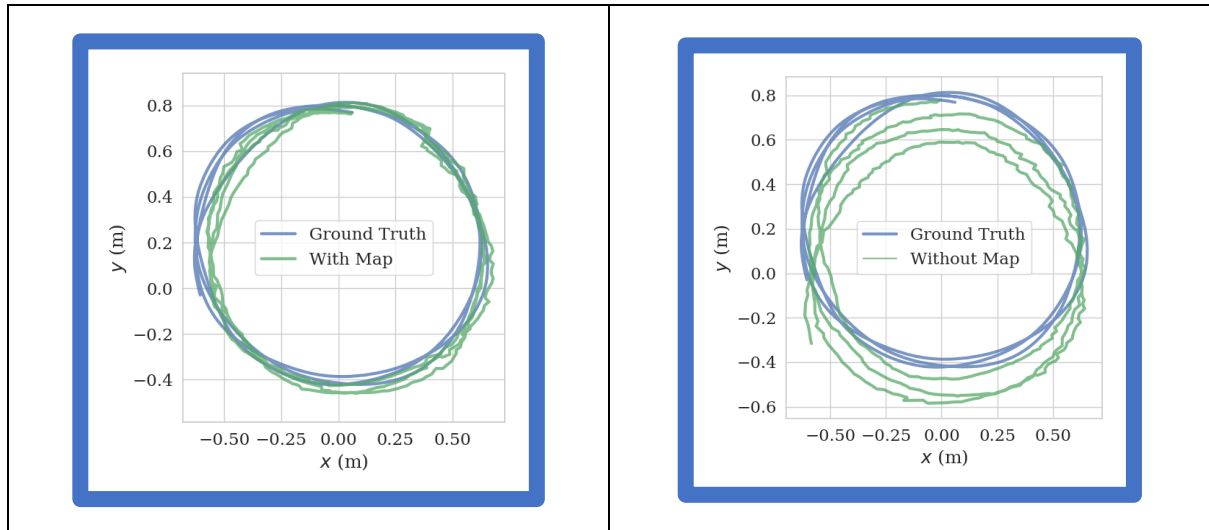
As shown in Figure 8, the above procedure results in a fixed-length state queue being maintained. The integration of the state is driven by the IMU measurement, i.e., whenever an IMU measurement is inputted, the state will be updated and stored in the queue.

6. The usual ICP solutions consist of frame-to-frame and frame-to-map pipelines with frame-to-map to increase accuracy and reduce drift over time. Did you consider experiments with longer sequences? What would be the impact of lacking frame-to-map ICP when you operate for longer periods in the same environment?

Reply: The Reviewer's comment is deeply appreciated. In the next step of this research, the authors will build a point cloud map which supports the loop detection. Currently, the authors use the frame-to-frame pipeline with a maintained keyframes queue. When a new frame comes, the algorithm will use the guessed position to find out a closest keyframe and align this new frame with the found keyframe. And when the estimated pose of a frame is away from its keyframe, it will be marked as a new keyframe and pushed into the keyframe queue.

Regarding the impact of lacking frame-to-map ICP when we operate for longer periods in the same environment, obviously, the drift of odometry will accumulate and the absolute trajectory error will continue increase accordingly, even in the same environment (please also see the Reply to the Reviewer's comment No. 4 above). The authors show the estimated trajectory with and without the keyframe feature in the following figure:

With maintained keyframes queue (as the map)	Without maintained keyframes queue
--	------------------------------------



7. Minor issues:

- abbreviations Equ is rather strange and I would recommend Eq.- there is an additional line in Table 3 that is not needed- there is once "Jacobin" instead of "Jacobian"

Reply: Thanks for pointing out these mistakes. The authors have changed all “Equ” to “Eq.”, removed the redundant line in the Table 3, and corrected the typo mistake of Jacobian in the line above Eq. (39).