Characterizing Visual Localization and Mapping Datasets

Sajad Saeedi*, Eduardo D C Carvalho*, Wenbin Li*,§, Dimos Tzoumanikas*,
Stefan Leutenegger*, Paul H J Kelly*, Andrew J. Davison*

***Abstract*— Benchmarking mapping and motion estimation algorithms is established practice in robotics and computer vision. As the diversity of datasets increases, in terms of the trajectories, models, and scenes, it becomes a challenge to select datasets for a given benchmarking purpose. Inspired by the Wasserstein distance, this paper addresses this concern by developing novel metrics to evaluate trajectories and the environments without relying on any SLAM or motion estimation algorithm. The metrics, which so far have been missing in the research community, can be applied to the plethora of datasets that exist. Additionally, to improve the robotics SLAM benchmarking, the paper presents a new dataset for visual localization and mapping algorithms. A broad range of real-world trajectories is used in very high-quality scenes and a rendering framework to create a set of synthetic datasets with ground-truth trajectory and dense map which are representative of key SLAM applications such as virtual reality (VR), micro aerial vehicle (MAV) flight, and ground robotics.**

I. Introduction

In pose estimation and real-time scene understanding, benchmarking and comparison between algorithms and datasets is important in the experimental evaluation of new proposed methods. Papers often report performance scores for SLAM systems; most often pose estimation accuracy, but also increasingly the execution time, the performance of dense scene reconstruction, and other measures such as power consumption. However, as the diversity of the datasets is growing, it becomes a challenging issue to decide which and how many datasets should be used to compare results. Furthermore, the performance reported on a particular dataset, with a certain scene and type of camera motion, may not be representative of how well an algorithm will work in a particular application of a practical interest. To address this concern, we believe that datasets should be characterized more systematically according to their complexity in terms of both trajectory and environment, and to relate test data to real scenarios.

The SLAMBench framework presented initial work on looking at the performance of a whole robot vision system [1]. In SLAMBench, a SLAM algorithm (specifically KinectFusion [2]) is measured in terms of both accuracy and computational cost across a range of processor platforms and using different language implementations. In SLAMBench2.0 [3] and SLAMBench3.0 [4], more SLAM algorithms are supported by a SLAM API and an I/O system. Similar benchmarking works have been performed in [5] and [6]. As the application of SLAM algorithms in robotics and computer vision is growing, it is becoming apparent that a more sophisticated approach to benchmarking is needed [7]. In this paper we develop ideas from statistics to propose novel metrics to label datasets in terms of the motion, structure, and appearance qualities which are important to SLAM performance. The proposed metrics are general and easy to compute, and can be used in various other robotic tasks. Moreover, since for complete benchmarking, having both trajectory and model ground-truth is necessary, we present a set of new synthetic visual SLAM datasets. Unlike other synthetic datasets such as ICL-NUIM [8], the new datasets are based on real-world motion-captured trajectories representative of real applications, and cover a broad range of motions including human walking/running, VR/AR, MAV flight, and ground robotics. These trajectories are used in probably the most highly detailed and professional models so far used in a SLAM dataset, with high-quality rendering to create RGB, depth, flow, and inertial measurements with full ground-truth. We apply our new metrics both to an existing well-known dataset and to our new data and highlight their advantageous properties.

In summary, the contributions of this paper are

- novel statistical metrics to characterize, quantitatively, the inherent tracking difficulty presented by different trajectories and environments, which nonetheless does not rely on any SLAM or motion estimation algorithm.
- 16 new datasets each with real-world trajectories, synthetic RGB and depth images, optic flow, and inertial measurements. The trajectories include virtual reality, MAV, ground robot, walking, and running motions.

For videos, datasets, and more details, see the project's website [9]. The rest of the paper is organized as follows: Section II presents metrics to evaluate difficulty of datasets with experimental results. Section III summarizes the paper. The Appendix describes the new synthetic dataset with real-world trajectories.

II. Visual SLAM Dataset Characterization

In this section first we introduce the Wasserstein distance and explain theoretically why it is a suitable metric for characterizing localization and mapping datasets. Then we present an example with the extended Kalman filter SLAM and particle filter SLAM applied to different trajectories. Then we explain the application of the metric for visual SLAM with several examples.

A. Wasserstein Distance

Wasserstein distance, or metric, is a quantity which measures the distance between two probability distributions. The k^{th} order Wasserstein distance, $k \geq 1$, between two n-dimensional probability densities p and q defined on $\Omega \subseteq \mathbb{R}^n$ is:

$$W_k(p,q)^k = \inf_{\alpha \in \Theta(p,q)} \int_{\Omega \text{x} \Omega} ||x-y||^k \alpha(x,y) dx dy, \quad (1)$$

* Department of Computing, Imperial College London, UK
§ Department of Computer Science, University of Bath, UK

where $\Theta(p,q)$ denotes the set of all joint probability densities on $\Omega \times \Omega$ whose respective marginals correspond to q and p. The Wasserstein distance is a particular case of Kantorovich's formulation of the optimal transport problem [10], and is a metric over the set of all probability distributions on Ω. The joint probability density $\alpha(x,y)$ which appears in the integrand of Eq. (1) can be interpreted as the amount of probability mass needed to transform p into q. Taking the infimum over this joint distribution yields the optimal transport plan between p and q, where the integrand is weighted by the metric $d(x,y) := ||x-y||^k$, usually denoted as the cost function. Note that the Wasserstein distance needs not to be restricted over probability densities, i.e. the Radon-Nikodym derivative of probability measures which are absolutely continuous w.r.t to a reference measure Q (where Q is the Lebesgue measure in the continuous case), but can generally be defined over probability measures on the same Polish metric space [10]. Given the scope of this paper, all distributions considered, in particular the Gaussian, have a probability density representation and hence the discussion is simplified accordingly. For two Gaussian distributions, $p := N(\mu_p, \Sigma_p)$ and $q := N(\mu_q, \Sigma_q)$ with the same dimension n, the second order Wasserstein metric has a closed-form solution [11]:

$$W_2(p,q)^2 = ||\mu_p - \mu_q||^2 + \mathrm{tr}\left(\Sigma_p + \Sigma_q - 2(\Sigma_q^{\frac{1}{2}}\Sigma_p\Sigma_q^{\frac{1}{2}})^{\frac{1}{2}}\right) \tag{2}$$

where $||.||^2$ denotes the squared Euclidean distance and $\mathrm{tr}(.)$ the trace operator. In general, $W_k(p,q)$ is intractable due to the need of taking the infimum over a family of joint probability measures, and usually the most practically useful cases are $W_1(p,q)$ and $W_2(p,q)$, i.e. first and second order Wasserstein metrics. Note that the first order Wasserstein metric is also known as the earth mover's distance (EMD) [12]. Applying Hölder's inequality shows that if $k_1 \leq k_2$, then $W_{k_1}(p,q) \leq W_{k_2}(p,q)$ [10] (CH6, Eq. 6.4). This is related to the fact that results for $W_2(p,q)$ are stronger but harder to establish when compared to $W_1(p,q)$. Furthermore, due to being a metric, it also has the desirable properties of being symmetric, non-negative, obeying the triangle inequality and being 0 if and only if p and q are the same.

We now focus on discussing the relevance of the Wasserstein distance for characterizing motion in visual SLAM. The Wasserstein distance takes into account both geometrical properties, as encoded in $d(x,y)$, and also the probabilistic structure by integration over $\Theta(p,q)$. Taking the infimum yields a single non-negative number which quantifies the optimal transport plan between two distributions p and q. In the context of visual SLAM, one could think of p and q as two probability distributions over desired quantities, such as poses, measurements and/or landmarks, which are consecutive in time. Hence the argument is that, for two such consecutive distributions p and q, higher value of Wasserstein distance would be associated with a higher discrepancy between p and q, and hence harder to characterize motion for such scenario. There are plenty of options when it comes to choosing divergences and metrics between two probability measures, whose appropriateness is often dictated by the application context, computational tractability and other mathematical properties. One such example is the Kullback-Leibler (KL) divergence, which is popular in the fields of Statistics, Machine Learning and Information Theory, and has previously been considered in [13] in the context of motion characterization. Even though both Wasserstein metric and KL divergence have convenient closed-form solutions under Gaussian distributions, we note that the KL divergence is not symmetric unlike the Wasserstein distance, and hence does not translate to plausible physical meaning given the symmetric nature of motion in time. Our initial experiments have also considered the Jensen-Shannon (JS) divergence, due to being symmetric, but since this quantity is bounded above by a constant value, it was not appropriate for comparing motion change across different trajectories and environments. The reader is referred to [14], Example 1 from Section 2, for a non-Gaussian example where the Wasserstein distance provides a reasonable answer, in contrast with the KL divergence and JS divergence.

B. Applying Wasserstein Metric to SLAM Datasets

In this section, it is explained how the Wasserstein metric is applied to the visual odometry and SLAM problems, when ground truth information is available. Assuming $x_{1:t}$, $m_{1:t}$, $z_{1:t}$, and $u_{1:t}$ are the pose, map, measurements, and control signals for times 1 to t, the full SLAM problem is defined as maximizing the following probability distribution [15] (Ch. 11, Eq. (11.9)):

$$\begin{aligned} &p(x_{1:t}, m_{1:t} | z_{1:t}, u_{1:t}) \\ &= \eta p(x_0, m_0) \prod_t p(x_t | x_{t-1}, u_t) \prod_t p(z_t | x_t, m_t), \end{aligned} \tag{3}$$

Where η is the normalization constant, and x_0, m_0 are the initial values of pose and the map, respectively. Having ground truth information available for $\{x_{1:t}, m_{1:t}\}$ means that there is no need to estimate the posterior distribution described in Eq. (3), since there is no uncertainty to be expressed over these quantities. Hence the full SLAM problem simplifies to modelling the likelihood term $\prod_t p(z_t|\hat{x}_t, \hat{m}_t)$, where we now write $\{\hat{x}_{1:t}, \hat{m}_{1:t}\}$ for the full set of ground truth poses and map at time t. Similarly, for the online SLAM problem, which is the focus of the paper, corresponding to known ground truth and map, the distribution of interest is the measurement likelihood at each time t: $p(z_t|\hat{x}_t, \hat{m}_t)$. We note that $\hat{m}_t$ is a vector containing K_t landmark locations, where K_t denotes the number of observed landmarks at time t. Given the knowledge of map and pose at time t, we assume conditional independence of measurements and hence re-write the measurement likelihood as follows:

$$p(z_t|\hat{x}_t, \hat{m}_t) = \prod_{k=1}^{K_t} p(z_{t,k}|\hat{x}_t, \hat{m}_t) \tag{4}$$

Where $z_{t,k}$ denotes the vector of measurements at time t with respect to landmark $\hat{m}_{t,k}$. Furthermore, not all likelihood

terms in Eq. (4) will be of interest, where instead one is interested in quantifying discrepancies only in between measurements corresponding to the same landmarks at times $t-1$ and t. This means that motion characterization will be restricted to corresponding measurements, and is also a mathematically crucial argument given it ensures that the measurement random vectors of interest will have the same dimension for all consecutive time steps. Let $\mathcal{M} := \hat{m}_{t-1} \cap \hat{m}_t$ be the set containing the $K_{t-1,t}$ common landmarks being observed/estimated at times $t-1$ and t. For two consecutive time-steps, $t-1$ and t, we write $\tilde{z}_{t-1}$, $\tilde{z}_t$ for the measurements random vectors with corresponding landmarks and $P_{t-1} := p(\tilde{z}_{t-1}|\hat{x}_{t-1}, \mathcal{M})$ and $P_t := p(\tilde{z}_t|\hat{x}_t, \mathcal{M})$ for the distributions of interest. The quantity of interest between times $t-1$ and t is the following Wasserstein distance:

$$w(t) = W_2(P_t, P_{t-1}). \tag{5}$$

Furthermore, we are interested in independently characterizing motion for the bearing ϕ and range r measurements, due to the fact that these vary on different scales and one would like to have a more explicit description of sources of motion change over time. This means that $(\tilde{z}_{t-1}, \tilde{z}_t)$ separates into $(\tilde{z}_{t-1,\phi}, \tilde{z}_{t,\phi})$ and $(\tilde{z}_{t-1,r}, \tilde{z}_{t,r})$, with corresponding distributions $(P_{t-1,\phi}, P_{t,\phi})$ and $(P_{t-1,r}, P_{t,r})$. The trajectory will then be characterized for each two consecutive time-steps by computing $w_\phi(t)$ and $w_r(t)$, which are the time consecutive Wasserstein distances corresponding to ϕ and r, respectively. In order to summarize the whole trajectory into an aggregated metric, for the purpose of comparison, it is useful to then take the sample median of both Wasserstein distances, or rather considering box-plots for more detailed sample-based overview. Assuming a measurement model with Gaussian noise as typically done in the SLAM literature [15], and given the fact that ground truth information is available, we get that $P_{t-1,\phi}, P_{t,\phi}$ and $P_{t-1,r}, P_{t,r}$ have Gaussian distributions:

$$\tilde{z}_{t-1,\phi}|\hat{x}_{t-1}, \mathcal{M} \sim \mathcal{N}(\tilde{z}_{t-1,\phi}|\mu_{t-1,\phi}, \Sigma_{t-1,\phi}), \tag{6}$$
$$\tilde{z}_{t,\phi}|\hat{x}_t, \mathcal{M} \sim \mathcal{N}(\tilde{z}_{t,\phi}|\mu_{t,\phi}, \Sigma_{t,\phi}), \tag{7}$$
$$\tilde{z}_{t-1,r}|\hat{x}_{t-1}, \mathcal{M} \sim \mathcal{N}(\tilde{z}_{t-1,r}|\mu_{t-1,r}, \Sigma_{t-1,r}), \tag{8}$$
$$\tilde{z}_{t,r}|\hat{x}_t, \mathcal{M} \sim \mathcal{N}(\tilde{z}_{t,r}|\mu_{t,r}, \Sigma_{t,r}), \tag{9}$$

where $\mu_{t-1,\phi}, \mu_{t,\phi}$ and $\mu_{t-1,r}, \mu_{t,r}$ are mean vectors, and $\Sigma_{t-1,\phi}, \Sigma_{t,\phi}$ and $\Sigma_{t-1,r}, \Sigma_{t,r}$ are diagonal covariance matrices due to the conditional independence assumption written in Eq. (4). Finally, under the assumed Gaussian model and by Eq. (2), we can write $w_\phi(t)$ and $w_r(t)$ in the following closed-form:

$$w_\phi^2(t) = ||\mu_{t,\phi} - \mu_{t-1,\phi}||^2 + ||\Sigma_{t,\phi}^{\frac{1}{2}} - \Sigma_{t-1,\phi}^{\frac{1}{2}}||_F^2, \tag{10}$$
$$w_r^2(t) = ||\mu_{t,r} - \mu_{t-1,r}||^2 + ||\Sigma_{t,r}^{\frac{1}{2}} - \Sigma_{t-1,r}^{\frac{1}{2}}||_F^2, \tag{11}$$

where $||.||_F^2$ denotes the squared Frobenius norm, which arises due to having diagonal covariance matrices.

C. Simulated 2D Feature-based SLAM

An experiment in a simulated environment is presented with four different trajectories of various difficulty levels.

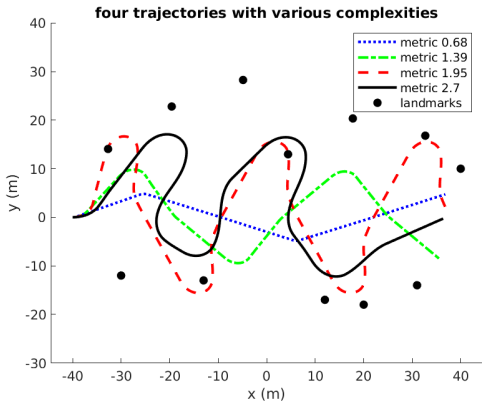

Fig. 1: Four trajectories with their difficulty scores based on the Wasserstein metric.

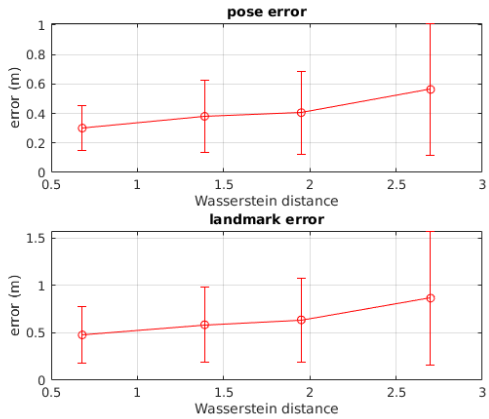

Fig. 2: 2D error bars of EKF SLAM applied to four trajectories. Each trajectory has been run 100 times, and mean and standard deviation of the both pose and landmark locations are shown. (a) pose error. (b) RMS landmark error. As each trajectory's difficulty metric is increasing, the variance of the error of the estimate pose and landmarks positions is also increasing.

The sensor measurements are range and bearing measurement of the features within the field of the view. The trajectories with the map of the features are shown in Fig 1. Each trajectory has a difficulty score, demonstrating the complexity of the trajectory. Since the ground-truth pose, measurements, and noise statistics are known, we use (11) to calculate the Wasserstein metric for each trajectory. Note that for simplicity, in this simulated experiment, we report the Wasserstein metric only on the range values. For each consecutive measurements, the metric is calculated, and for the trajectory, the median[1] Wasserstein metric is reported on Fig. 1, 2, and 3. Then each trajectory has been analyzed with the extended Kalman filter (EKF)-based SLAM [16] and also with the Rao-Blackwellized particle filter, known as FastSLAM [17]. The analyses show the relationship between the introduced difficulty metric and the performance measures.

Fig. 2 shows the error-bar plots for EKF SLAM. Each trajectory has been run 100 times, and the mean and standard deviation of both pose and landmark location error are shown. As the trajectories are getting more complex (increas-

[1]the median and mean coincide in Gaussian distributions.

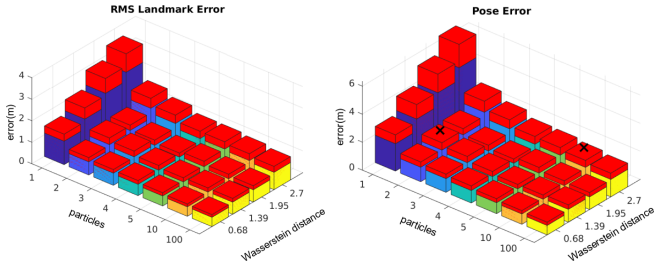

Fig. 3: 3D error bars of the Rao-Blackwellised particle filter SLAM applied to the four trajectories. Each trajectory has been run 100 times, with various number of particles. The mean of both pose and landmark locations are shown. The red bar on top of each bar shows the standard deviation. (right) pose error. (left) RMS landmark error. As each trajectory's metric is increasing, more particles are needed to achieve a low error variance on pose and landmark positions, and also error is increasing following the metric, across fixed number of particles.

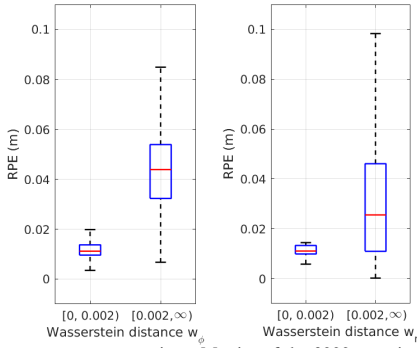

Fig. 4: On x-axes, two ranges are given. Metrics of the 3000 experimented frames fall within these ranges. Box-plots of RPEs corresponding to these frames are shown on the y-axes. Higher Wasserstein metric indicates higher median and variance in RPE.

ing Wasserstein distance on x-axis), it is harder to achieve a lower variance in pose and landmark error. Fig 3 shows the results for FastSLAM. For each trajectory, different number of particles are used. At each configuration, the algorithm was run 100 times, and the mean pose and RMS landmark error is shown. The standard deviation of each 3D error bar is shown in red. At easy trajectories, the algorithm is able to achieve a low variance in error with little number of particles; however, to achieve the same results with a difficult trajectory, more resources are needed, indicating the difficulty of the trajectory. For instance the trajectory with metric 1.39 is able to achieve the mean pose of 1.35 m and variance of 0.2 m with only 2 particles, but to achieve the same error with trajectory of metric 2.7, 10 particles are needed. (these two bars have been marked with a cross sign). See the code to reproduce the results [9]. An explanation for this behaviour is that for difficult trajectories, non-linearity is higher than simple ones. To make a better approximation of the nonlinear models, more particles are needed.

D. Dense RGBD Odometry

In this section, another experiment is performed to demonstrate the relation between the Wasserstein distance and the median and variance of the pose estimation error with a dense RGBD odometry algorithm [18]. We demonstrate that for each pair of frames, as the Wasserstein metric increases, the median and variance in the relative pose error (RPE) [19] also increases. RPE is an indication of the drift of the estimated pose from the ground-truth pose. This experiment provides another empirical evidence that the Wasserstein distance can be used as a metric to assess the difficulty of the frames for odometric pose estimation. A higher Wasserstein metric indicates that the median and variance in the estimated error is higher, which shows the sensitivity of the algorithm to be unstable. Note that the used dense RGBD odometry algorithm, like many other algorithms, is based on a Taylor series expansion around a reference point, and hence higher discrepancy in consecutive frames may result in higher Wasserstein distance and also in higher approximation error.

Eq. (10) and (11) are used to calculate the Wasserstein metric for each pair of consecutive RGB and depth images. Notice that unlike the previous simulated example, where variances used in these equations were known, it is practically intractable to generate statistics such as variances from images. In the following paragraph, it has been explained how these variances have been estimated empirically. 3000 frames from the ICL dataset (see Appendix) are used for pose estimation in this experiment. For each pair of frames, the ground-truth optic flow is used to determine the corresponding pixels. For each corresponding pair of pixels, from ground truth, range and angle ($\mu_{t,r}$, $\mu_{t-1,r}$, $\mu_{t,\phi}$, $\mu_{t-1,\phi}$) values are estimated in camera frame. The variance for range measurements ($\Sigma_{t,r}$, $\Sigma_{t-1,r}$) are estimated based on the method presented in [20]. For variance in angles of the corresponding pixels ($\Sigma_{t,\phi}$, $\Sigma_{t-1,\phi}$), photo consistency in a local patch is considered; for a pixel p in image I_t, if the corresponding pixel in image I_{t-1} is $q_{0,0}$, We use the following relation to calculate the uncertainty:

$$\Sigma_{t,\phi}(p) = \sum_{i,j \in \mathcal{P}} \tfrac{1}{\delta} |p - q_{i,j}|, \tag{12}$$

where $\mathcal{P}$ is a local patch of 3×3 pixels from I_{t-1} centered at $q_{0,0}$, and δ is a weight value, set experimentally to 0.01 to have comparable magnitudes in w_r and w_ϕ. $\Sigma_{t-1,\phi}$ is determined similarly. Fig. 4 depicts box-plot graphs that show the relation between the Wasserstein distances, calculated over ranges w_r and bearings w_ϕ, and the sample quantiles of the frame-by-frame RPEs. The intervals shown on the x-axes, $[0,\ 0.002)$ and $[0.002,\ \infty)$, are chosen such that approximately the same number of sample are available in each interval. These intervals are kept constant for the rest of the paper. The red line in the middle of each box is the median RPE. Clearly, the figure demonstrates that as the Wasserstein metrics increase, the sample median and variance of RPE values also increase. This shows that the proposed Wasserstein metrics, which are independent of the SLAM algorithm, can be used to estimate the expected variance in RPE, before images are processed by the SLAM algorithm.

E. Dataset Characterization

In the previous Sections, it was shown that the Wasserstein distance can be used to draw a relation between the variation

ICL Dataset						
Median:	w_ϕ	w_r	RPE	w_ϕ	w_r	RPE
Scence:	**Deer**			**Diamond**		
MAV-Fast	0.0029	0.0352	0.0536	0.0035	0.0715	0.0724
MAV-Slow	0.0016	0.0220	0.0487	0.0016	0.0306	0.0360
VR-Fast	0.0034	0.0500	0.0752	0.0028	0.0774	0.0749
VR-Slow	0.0022	0.0299	0.0487	0.0029	0.0468	0.0492
Running	0.0035	0.0276	0.0624	0.0040	0.0487	0.0701
Walking	0.0034	0.0275	0.0501	0.0029	0.0382	0.0435
Walking-Head	0.0027	0.0276	0.0423	0.0019	0.0403	0.0418
Ground Robot	0.0033	0.0058	0.0105	0.0015	0.0018	0.0108

TABLE I: The median Wasserstein distances and RPE (meters) for 16 ICL trajectories.

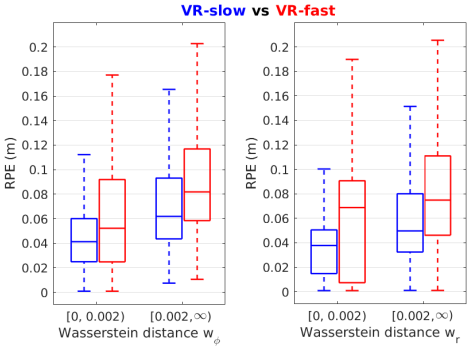

Fig. 5: Box-plots of RPEs vs Wasserstein metric for two example trajectories VR-slow (blue) and VR-fast (red) in both scenes, Deer and Diamond. Table I shows that VR-fast has higher metrics than VR-slow, hence higher RPE variance/median in all frames.

in error and the input measurements. Based on this relation, we apply the Wasserstein distance to characterize two datasets. ICL and TUM RGBD. Both datasets are based on real-world trajectories, captured by a motion capture system. Table I presents the proposed metrics for 16 trajectories in the ICL dataset. ICL dataset is a new dataset explained in the Appendix of the paper. In ICL dataset, ground truth optic flow and depth are available; thus we calculate the metrics as explained in the previous section. As the median metrics grow, the median RPE also grows. Fig. 5 shows the box-plot of the RPE values (using dense RGBD odometry algorithm [18]) vs Wasserstein distance for two example trajectories VR-slow (blue) and VR-fast (red). VR-fast is composed of rapid head motions, typical of VR application. VR-slow is similar but without rapid motions [9]. According to the figure, median and varince of RPE in VR-fast is higher than VR-slow. Consistently, Table I shows that VR-fast has higher metrics than VR-slow, hence higher variance in RPE error across all frames.

For TUM RGBD, there is no ground-truth optic flow available; however, given the ground truth poses and measurements, by creating a local 3D map, we were able to determine the corresponding pixels, and determine the metrics as done for the ICL dataset. To deal with occlusions, we use photo consistency constrains by predicting images from the 3D model for known poses. If the predicted pixels at a known pose do not have similar intensities values as the actual image at that pose, it will be considered as occlusion, with no correspondence for that pixel. Table II shows the median metrics and RPEs for some of the trajectories in TUM dataset. Although these trajectories do not have the same length, consistently RPE follows the metrics. Fig. 6 compares the variance/median of RPE w.r.t Wasserstein metrics for the frames of *F3/struct_txt_f* and *walking-head* from ICL. From this figure, and the latter two Tables, it is noticeable that a higher metric is related to higher RPE variance, indicating that the metrics are representative of the difficulty of the frames for pose estimation.

TUM-RGDB Dataset [19]			
Median:	w_ϕ	w_r	RPE (m)
F3/nostrct_txt_f (465 frames)	0.0030	0.0117	0.0238
F3/nostrct_notxt_f (474 frames)	0.0029	0.0065	0.0131
F3/strct_txt_f (938 frames)	0.0019	0.0062	0.0131
F3/struct_notxt_f (814 frames)	0.0013	0.0034	0.0098

TABLE II: The median Wasserstein distances and RPE for four TUM trajectories.

III. Conclusions

This paper has presented metrics to characterize localization and mapping trajectories in different environments, quantitatively. The metrics use ground-truth information. Through three algorithms, EKF-SLAM, FastSLAM, and dense RGBD odometry applied to simulated features, synthetic datasets, and real-world datasets, we have demonstrated that the higher values of the metrics are related with higher RPE variance and median. Additionally, a new synthetic dataset with eight different trajectories in two different and highly detailed models has been generated. The trajectories were recorded with a motion capture system and include various realistic applications such as virtual reality, robot navigation, and human motion. One advantage of the metrics is that the trajectories can be characterized based on these metrics and SLAM developers and researchers can easily develop algorithms that are customized to a certain class of trajectories. Additionally, it is possible to use the metrics for active SLAM, i.e. to design a real-time motion planning algorithm with an objective function based on Wasserstein distance. In future, we plan to apply the metric to other SLAM datasets, so the that developers can selectively work with trajectories with known difficulty levels. We also plan to add the metric to SLAMBench3.0 [4], and extended the metric to support multi-robot trajectories [21].

Acknowledgment

This research is supported by Engineering and Physical Sciences Research Council (EPSRC), grant references EP/K008730/1 and EP/N018494/1.

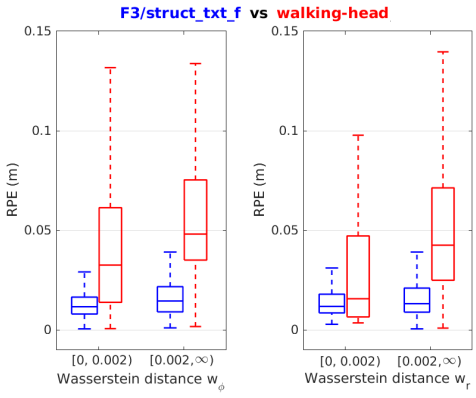

Fig. 6: Box-plot of RPE w.r.t Wasserstein distance for two trajectories from ICL and TUM datasets. Higher Wasserstein metric indicates higher variance/median in RPE.

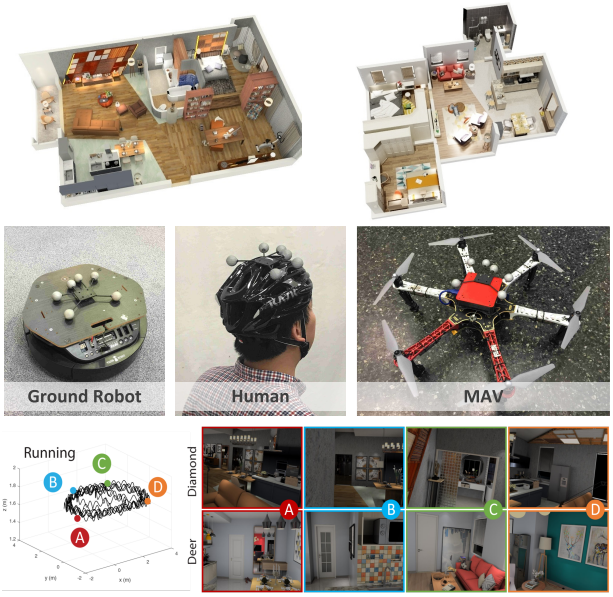

Fig. 7: (**top left**) Topview of Diamond scene. (**top right**) Topview of Deer scene. (**middle**) Vicon setup for real-world trajectory capture: (from left to right) ground robot, human helmet, and MAV. (**bottom**) A sample trajectory, Running, and camera views associated with the two synthetic scenes. Four selected camera poses are visualized within the scenes Diamond and Deer respectively.

No	Scene	Area(m^2)	Description
1	Diamond	126.9	wide interior space, more than 6.95 M triangles
2	Deer	89.7	crowded living space, more than 13.62 M triangles

TABLE III: Highly detailed scenes used for rendering.

No	Trajectory	Time(s)	Length(m)
1	**MAV-Fast**: high FPS & stable	100.0	151.3
2	**MAV-Slow**: low FPS & jittery	204.9	92.9
3	**VR-Fast**: rapid head motion	61.4	58.1
4	**VR-Slow**: includes slow walking	65.4	38.6
5	**Running**: normal running	60.9	100.6
6	**Walking**: normal walking	64.1	50.2
7	**Walking-Head**: sudden motion	62.4	76.3
8	**Ground Robot**: close to floor	79.9	16.8

TABLE IV: Trajectories used to generate datasets in two different environments.

APPENDIX
ICL DATASET

Numerous datasets, real-world and synthetic, are available for various tasks in robotics and computer vision. Examples include SceneNet [22], InteriorNet [23], SUNCG [24], Sintel [25], New College [26], KITTI [27], UnrealCV [28], UnrealStereo [29], ICL-NUIM [8], TUM-RGBD [19], and EuRoC [30]. Here, we focus on indoor scene understanding for applications that require various types of motion. Examples of these motions are human walking/running, ground robot navigation, MAV navigation, and typical motions in virtual reality applications. Compared to other synthetic datasets such as ICL-NUIM [8], our dataset has more diversity. Our dataset is the only synthetic dataset with real-world trajectories recorded by motion capture. The trajectories have not been modified, scaled or altered. RGB, depth, optic flow, and inertial measurements for each trajectory are provided.

A. Rendering System

The photo-realistic renderer used in this work was built on top of Intel Embree [31], an open-source CPU based ray-tracer. Our renderer supports common functions such as global illumination and mirror materials. To simulate real-world artifacts, we implement additional features such as motion blur, random lighting color/strength, and specular/transparent materials. Note that we also considered other famous open-source alternatives for rendering, like NVidia Optix&photonmap and Blender. Although the former is GPU based and has been used in a recent dataset [32], it often requires a large amount of GPU memory to host a complex scene. The latter is less flexible in supporting the real-world artifacts we need. Additionally, POVRay and OpenGL are also CPU based renderers and widely used in the field. Although POVRay is capable of rendering high-quality images, it is often slower than Embree given the same CPU setup. It is hard for OpenGL to support real-world photometric effects.

For high-quality scene data, we use two high-resolution scenes (Diamond and Deer, Fig. 7 (top)) which were created by an award-winning professional using Foundry NukeX/Modo. Both scenes contain more than 6M triangles, more than 120 furniture models, and difficult materials such as mirror, transparency, and specular surfaces. Scene *Diamond* represents a wide interior space with low light conditions whilst *Deer* shows a crowded living room with multiple small objects occluded from each other. Hence, as summarized in Table III, we believe such proposed scenes are representative for most of real-world daily environment.

Note that we don't have a specific requirement on rendering speed but prefer good image quality. In the actual rendering, we output 640×480 images and each render with custom parameters setting takes on average 18 seconds on an i7 6800k CPU (3.4 GHz, 6 cores). This rendering time can also be reduced if image quality is sacrificed [31]. In addition to noisy and noise-free inertial measurements, both noisy and noise-free RGB and depth images, based on the noise models introduced in [8], along with other information e.g. camera parameters, frame rate, etc. are available [9].

B. Trajectories

Our datasets have been created from realistic trajectories, recorded using a motion capture system at 100 Hz. Fig. 7 (middle) shows the Vicon setup for the trajectories, which are recorded by a ground robot, an MAV, and a person performing specific tasks. Eight different trajectories were recorded with different types of motions. Two trajectories are for an MAV moving with fast and slow motions. Two other trajectories are for virtual reality with fast and slow head motion. Three trajectories are for a person walking, running, and also moving their head rapidly while walking. Finally, there is one trajectory recorded by a ground robot. The trajectories are carefully placed in the models to create realistic scenarios. Fig. 7 (bottom) shows a trajectory with sample images from the dataset. Table IV summarizes these trajectories. For evaluation results of the trajectories, see [9].

REFERENCES

[1] L. Nardi, B. Bodin, M. Z. Zia, J. Mawer, A. Nisbet, P. H. J. Kelly, A. J. Davison, M. Luján, M. F. P. O'Boyle, G. Riley, N. Topham, and S. Furber, "Introducing SLAMBench, a performance and accuracy benchmarking methodology for SLAM," in *ICRA*, 2015, pp. 5783–5790.

[2] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon, "KinectFusion: Real-time dense surface mapping and tracking," in *ISMAR*, 2011, pp. 127–136.

[3] B. Bodin, H. Wagstaff, S. Saeedi, L. Nardi, E. Vespa, J. Mawer, A. Nisbet, M. Luján, S. Furber, A. J. Davison, P. H. J. Kelly, and M. F. P. O'Boyle, "SLAMBench2: Multi-Objective Head-to-Head Benchmarking for Visual SLAM," in *ICRA*, 2018, pp. 3637–3644.

[4] M. Bujanca, P. Gafton, S. Saeedi, A. Nisbet, B. Bodin, M. F. OBoyle, A. J. Davison, P. H. Kelly, G. Riley, B. Lennox, M. Luján, and S. Furber, "SLAMBench 3.0: Systematic automated reproducible evaluation of SLAM systems for robot vision challenges and scene understanding," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2019.

[5] D. Jeffrey and S. Davide, "A benchmark comparison of monocular visual-inertial odometry algorithms for flying robot," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 2502–2509.

[6] M. Abouzahir, A. Elouardi, R. Latif, S. Bouaziz, and A. Tajer, "Embedding SLAM algorithms: Has it come of age?" *Robotics and Autonomous Systems*, vol. 100, pp. 14 – 26, 2018.

[7] S. Saeedi, B. Bodin, H. Wagstaff, A. Nisbet, L. Nardi, J. Mawer, N. Melot, O. Palomar, E. Vespa, T. Spink, C. Gorgovan, A. Webb, J. Clarkson, E. Tomusk, T. Debrunner, K. Kaszyk, P. Gonzalez-De-Aledo, A. Rodchenko, G. Riley, C. Kotselidis, B. Franke, M. F. P. O'Boyle, A. J. Davison, P. H. J. Kelly, M. Lujn, and S. Furber, "Navigating the landscape for real-time localization and mapping for robotics and virtual and augmented reality," *Proceedings of the IEEE*, vol. 106, no. 11, pp. 2020–2039, 2018.

[8] A. Handa, T. Whelan, J. McDonald, and A. Davison, "A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM," in *ICRA*, 2014, pp. 1524–1531.

[9] "ICL Dataset," http://vinben.github.io/lmdata.

[10] C. Villani, *Optimal Transport: Old and New*, ser. Grundlehren der mathematischen Wissenschaften. Springer, 2008.

[11] "The Frechet distance between multivariate normal distributions," *Journal of Multivariate Analysis*, vol. 12, no. 3, pp. 450 – 455, 1982.

[12] Y. Rubner, C. Tomasi, and L. J. Guibas, "The earth mover's distance as a metric for image retrieval," *International Journal of Computer Vision*, vol. 40, no. 2, pp. 99–121, 2000.

[13] S. Saeedi, L. Nardi, E. Johns, B. Bodin, P. H. Kelly, and A. J. Davison, "Application-oriented design space exploration for SLAM algorithms," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 5716–5723.

[14] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *International Conference on Machine Learning*, 2017, pp. 214–223.

[15] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, MA, USA: The MIT press, 2005.

[16] M. W. M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localization and map building (SLAM) problem," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 3, pp. 229–241, 2001.

[17] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM: A factored solution to the simultaneouls localization and mapping problem," in *Proceedings of the AAAI National Conference on Artificial Intelligence*, 2002.

[18] F. Steinbrucker, J. Sturm, and D. Cremers, "Real-time visual odometry from dense RGB-D images," in *IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, 2011, pp. 719–722.

[19] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *IROS*, 2012, pp. 573–580.

[20] J. T. Barron and J. Malik, "Intrinsic scene properties from a single RGB-D image," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 17–24.

[21] S. Saeedi, M. Trentini, M. Seto, and H. Li, "Multiple-robot simultaneous localization and mapping: A review," *Journal of Field Robotics*, vol. 33, no. 1, pp. 3–46, 2016.

[22] A. Handa, V. Patraucean, S. Stent, and R. Cipolla, "SceneNet: An annotated model generator for indoor scene understanding," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 5737–5743.

[23] W. Li, S. Saeedi, J. McCormac, R. Clark, D. Tzoumanikas, Q. Ye, Y. Huang, R. Tang, and S. Leutenegger, "InteriorNet: Mega-scale multi-sensor photo-realistic indoor scenes dataset," in *British Machine Vision Conference (BMVC)*, 2018.

[24] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser, "Semantic scene completion from a single depth image," *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[25] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black, "A naturalistic open source movie for optical flow evaluation," in *ECCV*. Springer Berlin Heidelberg, 2012, pp. 611–625.

[26] M. Smith, I. Baldwin, W. Churchill, R. Paul, and P. Newman, "The New College vision and laser data set," *The International Journal of Robotics Research*, vol. 28, no. 5, pp. 595–599, 2009.

[27] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the KITTI vision benchmark suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[28] W. Qiu and A. Yuille, "UnrealCV: Connecting computer vision to Unreal Engine," *arXiv preprint arXiv:1609.01326*, 2016.

[29] Y. Zhang, W. Qiu, Q. Chen, X. Hu, and A. Yuille, "Unrealstereo: A synthetic dataset for analyzing stereo vision," *arXiv preprint arXiv:1612.04647*, 2016.

[30] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, "The EuRoC micro aerial vehicle datasets," *IJRR*, vol. 35, no. 10, pp. 1157–1163, 2016.

[31] I. Wald, S. Woop, C. Benthin, G. S. Johnson, and M. Ernst, "Embree: a kernel framework for efficient CPU ray tracing," *ACM Transactions on Graphics*.

[32] J. McCormac, A. Handa, S. Leutenegger, and A. J. Davison, "SceneNet RGB-D: 5M photorealistic images of synthetic indoor trajectories with ground truth," 2016.