Evaluating Egomotion and Structure-from-Motion Approaches Using the TUM RGB-D Benchmark

Jürgen Sturm[1], Wolfram Burgard[2] and Daniel Cremers[1]

***Abstract*— In this paper, we present the TUM RGB-D benchmark for visual odometry and SLAM evaluation and report on the first use-cases and users of it outside our own group. The benchmark contains a large set of image sequences recorded from a Microsoft Kinect associated with highly accurate and time-synchronized ground truth camera poses from an external motion capture system. The dataset consists in total of 39 sequences that were recorded in different environments and cover a large variety of scenes and camera motions. In this work, we discuss and briefly summarize the evaluation results of the first users from outside our group. Our goal with this analysis is to better understand (1) how other researcher use our dataset to date and (2) how to improve it further in the future.**

I. Introduction

Public datasets and benchmarks greatly support the scientific evaluation and objective comparison of algorithms. Several examples of successful benchmarks in the area computer vision have demonstrated that common datasets and clear evaluation metrics can significantly help to push the state-of-the-art forward. One highly relevant problem in robotics is the so-called simultaneous localization (SLAM) problem where the goal is to both recover the camera trajectory and the map from sensor data. The SLAM problem has been investigated in great detail for sensors such as sonar, laser, cameras, and time-of-flight sensors. Recently, novel low-cost RGB-D sensors such as the Kinect became available, and the first SLAM systems using these sensors have already appeared [1]–[3]. Other algorithms focus on fusing depth maps to a coherent 3D model [4]. Yet, the accuracy of the computed 3D model heavily depends on how accurate one can determine the individual camera poses.

With this dataset, we provide a complete benchmark that can be used to evaluate visual SLAM and odometry systems on RGB-D data. To stimulate comparison, we propose two evaluation metrics and provide automatic evaluation tools.

The TUM RGB-D benchmark [5] consists of 39 sequences that we recorded in two different indoor environments. Each sequence contains the color and depth images, as well as the ground truth trajectory from the motion capture system. We carefully calibrated and time-synchronized the Kinect sensor to the motion capture system. After calibration, we measured the accuracy of the motion capture system to validate the calibration.

[1] Jürgen Sturm and Daniel Cremers are with the Computer Vision Group, Computer Science Department, Technical University of Munich, Germany. {sturmju,cremers}@in.tum.de

[2] Wolfram Burgard is with the Autonomous Intelligent Systems Lab, Computer Science Department, University of Freiburg, Germany. burgard@informatik.uni-freiburg.de

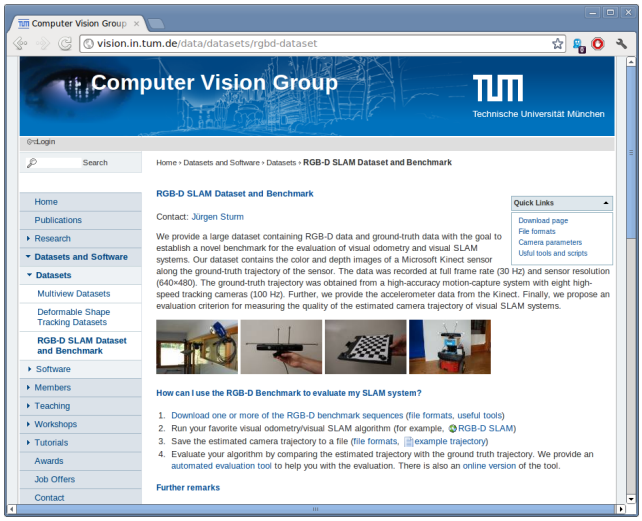


Fig. 1. Since July 2011, we offer a large dataset for the quantitative and objective evaluation of RGB-D SLAM systems. The benchmark website contains the dataset, evaluation tools and additional information.

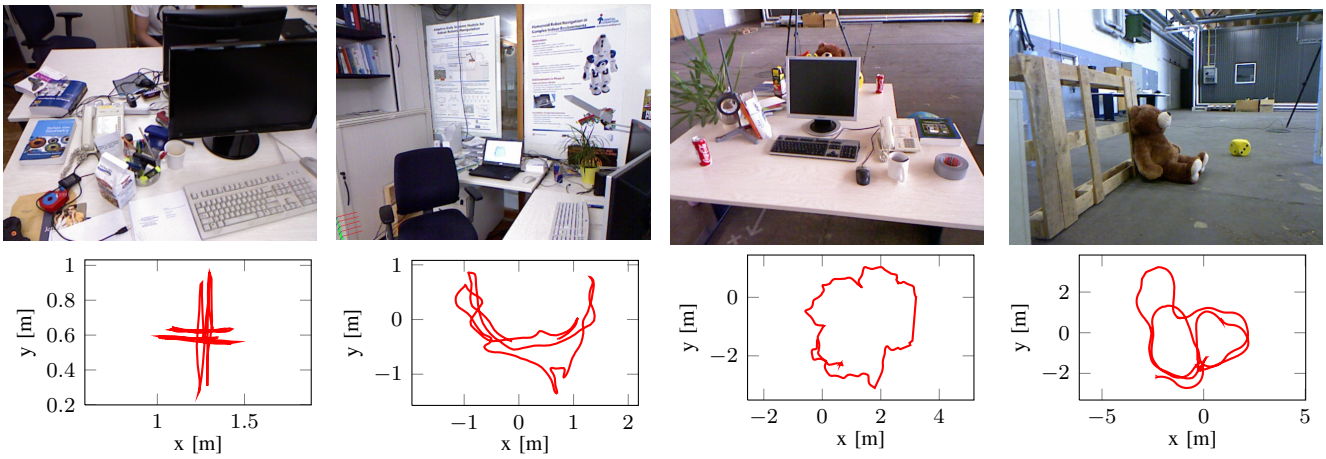
The benchmark website contains—next to additional information about the data formats, calibration data, and example code—videos for simple visual inspection of the dataset. All data is available online under the Creative Commons Attribution license (CC-BY 3.0) at

`http://vision.in.tum.de/data/datasets/rgbd-dataset`

Since the launch of the benchmark website in July 2011 [6], six peer-reviewed scientific publications (plus one master's thesis) have appeared in which our benchmark has been used to evaluate egomotion estimation and SLAM approaches [7]–[13]. Therefore, we want to take the opportunity to discuss the results and to relate the evaluations to one another where applicable. Our goal is to learn from this analysis how our benchmark is received by the community and how to refine it in the near future.

II. Related Work

The simultaneous localization and mapping (or structure-from-motion) problem has a long history both in robotics [14]–[20] and in computer vision [17], [21]–[24]. Different sensor modalities have been explored in the past, including 2D laser scanners [25], [26], 3D scanners [27]–[29], monocular cameras [17], [22]–[24], [30]–[32], stereo systems [33], [34] and recently RGB-D sensors such as the Microsoft Kinect [1]–[3].



(a) fr1/xyz (b) fr1/room (c) fr2/desk (d) fr2/slam

Fig. 2. Four examples of sequences contained in our dataset. Whereas the top row shows an example image, the bottom row shows the ground truth trajectory. The fr1/xyz sequence contains isolated motions along the coordinate axes, fr1/room and fr2/desk are sequences with several loop closures in two different office scenes, and fr2/slam was recorded from a Kinect mounted on a Pioneer 3 robot in a search-and-rescue scenario.

For laser- and camera-based SLAM systems, there are several well-known datasets such as the Freiburg, Intel, Rawseeds and Newcollege datasets [35]–[37]. Geiger et al. [38] recently presented a benchmark for visual odometry from stereo images with ground truth poses. However the depth maps are not provided so that an additional pre-processing step is required. Also the first Kinect datasets have appeared. Pomerleau et al. [39] recorded a dataset with untextured point clouds from a Kinect in a motion capture studio. Also related is the work of Bao et al. [40] who aimed at the evaluation of semantic mapping and localization methods. However, in their dataset the camera poses were estimated from the color images of the Kinect, so that the ground truth is not accurate enough for our purpose. To the best of our knowledge, our dataset is therefore the first RGB-D dataset suitable for the evaluation of visual SLAM (and visual odometry) systems, as it contains both color and depth images and associated ground truth camera poses. An earlier version of our benchmark was presented recently [6]. Inspired from the feedback we received, we extended the original dataset with dynamic sequences, longer trajectories, and sequences recorded from a Kinect mounted on a mobile robot.

Next to the data itself, a suitable evaluation metric is required for the benchmarking of SLAM solutions. We advocate—similar to Olson et al. [41]—to evaluate the end-to-end performance of the whole system by comparing its output (map or trajectory) with the ground truth. The map can for example be evaluated by overlaying it onto the floor plan and searching for differences. Although, in principle, difference images between the two maps can be computed automatically [42], the performance is often only judged visually by searching for thin structures, kinks or ghosts like double walls. The alternative to map comparison is to evaluate a SLAM system by comparing the estimated camera motion against the true trajectory. Two frequently employed methods are the relative pose error (RPE) and the absolute trajectory error (ATE). The RPE measures the difference between the estimated motion and the true motion. It can either be used to evaluate the drift of a visual odometry system [43], [44] or the accuracy at loop closures of SLAM systems [45], [46] which is especially useful if only sparse, relative relations are available as ground truth. Instead of evaluating relative poses differences, the ATE first aligns the two trajectories and then evaluates directly the absolute pose differences. This method is well suited for the evaluation of visual SLAM systems [41], [47] but requires that absolute ground truth poses are available. As we provide dense and absolute ground truth trajectories, both metrics are applicable. For both measures, we provide a reference implementation that computes the respective error given the estimated and the ground truth trajectory.

In this paper, we first describe the TUM RGB-D benchmark and how it can be used to evaluate visual SLAM and odometry systems as presented recently [5]. In extension to this, we additionally discuss recent results other researchers have obtained by using our benchmark for evaluation and draw conclusions on how to improve further it.

III. Dataset

The Kinect sensor consists of an near-infrared laser that projects a refraction pattern on the scene, an infrared camera that observes this pattern, and a color camera in between. As the projected pattern is known, it is possible to compute the disparity using block matching techniques. Note that image rectification and block matching is implemented in hardware and happens internally in the sensor.

We acquired a large set of data sequences containing both RGB-D data from the Kinect and ground truth pose estimates from the motion capture system. We recorded these trajectories both in a typical office environment ("fr1", $6 \times 6\text{m}^2$) and in a large industrial hall ("fr2", $10 \times 12\text{m}^2$)

as depicted in Fig. 1. In most of these recordings, we used a handheld Kinect to browse through the scene. Furthermore, we recorded additional sequences with a Kinect mounted on a wheeled robot. Table I summarizes statistics over the 19 training sequences, and Fig. 2 shows images of four of them along with the corresponding camera trajectory. On average, the camera speeds of the fr1 sequences are higher than those of fr2. Except otherwise noted, we ensured that each sequence contains several loop closures to allow SLAM systems to recognize previously visited areas and use this to reduce camera drift. Inspired from successful benchmarks in computer vision such as the Middlebury optical flow dataset [48] and the KITTI vision benchmark suite [38], we split out dataset into a training and a testing part. While the training sequences are fully available, the testing sequences can only be evaluated on the benchmark website [49] to avoid over-fitting.

We grouped the recorded sequences into the categories "Calibration", "Testing and Debugging", "Handheld SLAM", and "Robot SLAM". In the following, we briefly summarize the recorded sequences according to these categories.

a) Calibration: For the calibration of intrinsic and extrinsic parameters of the Kinect and the motion capture system, we recorded for each Kinect

- one sequence with color and depth images of a handheld 8×6 checkerboard with 20 mm square size recorded by a stationary Kinect,
- one sequence with infrared images of a handheld 8×6 checkerboard with 20 mm square size recorded by a stationary Kinect,
- one sequence with color and depth images of a stationary 8×7 checkerboard with 108 mm square size recorded by a handheld Kinect.

b) Testing and Debugging: These sequences are intended to facilitate the development of novel algorithms with separated motions along and around the principal axes of the Kinect. In the "xyz" sequences, the camera was moved approximately along the X-, Y- and Z-axis (left/right, up/down, forward/backward) with little rotational components (see also Fig. 2a). Similarly, in the two "rpy" (roll-pitch-yaw) sequences, the camera was mostly only rotated around the principal axes with little translational motions.

c) Handheld SLAM: We recorded 11 sequences with a handheld Kinect, i.e., 6-DOF camera motions. For the "fr1/360" sequence, we covered the whole office room by panning the Kinect in the center of the room. The "fr1/floor" sequence contains a camera sweep over the wooden floor. The "fr1/desk", "fr1/desk2" and "fr1/room" sequences cover two tables, four tables, and the whole room, respectively (see Fig. 2b). In the "fr2/360_hemisphere" sequence, we rotated the Kinect on the spot and pointed it at the walls and the ceiling of the industrial hall. In the "fr2/360_kidnap" sequence, we briefly covered the sensor with the hand for a few seconds to test the ability of SLAM systems to recover from sensor outages. For the "fr2/desk" sequence, we set up an office environment in the middle of the motion capture area consisting of two tables with various accessoires like a monitor, a keyboard, books, see Fig. 2c. Additionally, during the recording of the "fr2/desk_with_person" sequence a person was sitting at one of the desks and continuously moved several objects around.

(a) Motion capture system

(b) Kinect sensor with reflective motion capture markers

Fig. 3. We use an external motion capture system from MotionAnalysis to track the camera pose of the Kinect.

Furthermore, we recorded two large tours through the industrial hall, partially with poor illumination and few visual features. In the "fr2/large_no_loop" sequence, special care was taken that no visual overlap exists in the trajectory. Our intention behind this was to provide a sequence for measuring the long-term drift of (otherwise loop closing) SLAM systems. In contrast, the "fr2/large_with_loop" sequence has a large overlap between the beginning and the end of the sequence, so that a large loop exists. It should be noted that these tours were so large that we had to leave the motion capture area in the middle of the industrial hall. As a result, ground truth pose information only exists in the beginning and in the end of the sequence.

d) Robot SLAM: We also recorded four sequences with a Kinect that was mounted on an ActivMedia Pioneer 3 robot. With these sequences, it becomes possible to demonstrate the applicability of SLAM systems to wheeled robots. We aligned the Kinect horizontally, looking forward into the driving direction of the robot, so that the horizon was roughly located in the center of the image. Note that the motion of the Kinect is not strictly restricted to a plane because occasional tremors (as a result of bumps and wires on the floor) deflected the orientation of the Kinect. During recording, we joysticked the robot manually through the scene.

In the "fr2/pioneer_360" sequence, we drove the robot in a loop around the center of the (mostly) empty hall. Due to the large dimensions of the hall, the Kinect could not observe the depth of the distant walls for parts of the sequence. Furthermore, we set up a search-and-rescue scenario in the hall consisting of several office containers, boxes, and other feature-poor objects, see Fig. 2d. As a consequence, these sequences have depth, but are highly challenging for methods that rely on distinctive keypoints. In total, we recorded three sequences "fr2/pioneer_slam", "fr2/pioneer_slam2", and "fr2/pioneer_slam3" that differ in the actual trajectories but all contain several loop closures.

IV. Data acquisition

All data was recorded at full resolution (640×480) and full frame rate (30 Hz) of the Microsoft Xbox Kinect sensor on a

Linux laptop running Ubuntu 10.10 and ROS Diamondback. For recording the RGB-D data, we used two different off-the-shelf Microsoft Kinect sensors (one for the "fr1" sequences, and a different sensor for the "fr2"). To access the color and depth images, we used the openni_camera package in ROS which internally wraps PrimeSense's OpenNI-driver [50]. As the depth image and the color image are observed from two different cameras, the observed (raw) images are initially not aligned. To this aim, the OpenNI-driver has an option to register the depth image to the color image using a Z-buffer automatically. This is implemented by projecting the depth image to 3D and subsequently back-projecting it into the view of the color camera. The OpenNI-driver uses for this registration the factory calibration stored on the internal memory. Additionally, we used the kinect_aux driver to record the accelerometer data from the Kinect at 500 Hz.

To obtain the camera pose of the Kinect sensor, we used an external motion capture system from MotionAnalysis [51]. Our setup consists of eight Raptor-E cameras with a camera resolution of 1280×1024 pixels at up to 300 Hz (see Fig. 3a). The motion capture system tracks the 3D position of passive markers by triangulation. To enhance the contrast of these markers, the motion capture cameras are equipped with infrared LEDs to illuminate the scene. We verified that the Kinect and the motion capture system do not interfere: The motion capture LEDs appear as dim lamps in the Kinect infrared image with no influence on the produced depth maps, while the projector of the Kinect is not detected at all by the motion capture cameras.

Finally, we also video-taped all experiments with an external video camera to capture the camera motion and the scene from a different view point. All sequences and movies are available on our website [49].

All components of our setup have been carefully calibrated, in particular the intrinsic and extrinsic calibration of the color camera, depth sensor and motion capture system. Furthermore, we synchronized the timings between all components due to time delays in pre-processing, buffering, and data transmission of the individual sensors. For more details, and a description of the file formats etc., we refer to the original benchmark paper [5].

Sequence Name	Duration [s]	Avg. Trans. Vel. [m/s]	Avg. Rot. Vel. [deg/s]
Testing and Debugging			
fr1/xyz	30	0.24	8.92
fr1/rpy	28	0.06	50.15
fr2/xyz	123	0.06	1.72
fr2/rpy	110	0.01	5.77
Handheld SLAM			
fr1/360	29	0.21	41.60
fr1/floor	50	0.26	15.07
fr1/desk	23	0.41	23.33
fr1/desk2	25	0.43	29.31
fr1/room	49	0.33	29.88
fr2/360_hemisphere	91	0.16	20.57
fr2/360_kidnap	48	0.30	13.43
fr2/desk	99	0.19	6.34
fr2/desk_with_person	142	0.12	5.34
fr2/large_no_loop	112	0.24	15.09
fr2/large_with_loop	173	0.23	17.21
Robot SLAM			
fr2/pioneer_360	73	0.23	12.05
fr2/pioneer_slam	156	0.26	13.38
fr2/pioneer_slam2	116	0.19	12.21
fr2/pioneer_slam3	112	0.16	12.34

TABLE I
LIST OF AVAILABLE RGB-D SEQUENCES

V. EVALUATION METRICS

A SLAM system generally outputs the estimated camera trajectory along with an estimate of the resulting map. While it is in principle possible to evaluate the quality of the resulting map, accurate ground truth maps are difficult to obtain. Therefore, we propose to evaluate the quality of the estimated trajectory from a given input sequence of RGB-D images. This approach simplifies the evaluation process greatly. Yet, it should be noted that a good trajectory does not necessarily imply a good map, as for example even a small error in the map could prevent the robot from working in the environment (obstacle in a doorway).

For the evaluation, we assume that we are given a sequence of poses from the estimated trajectory $\mathbf{P}_1, \ldots, \mathbf{P}_n \in \mathrm{SE}(3)$ and from the ground truth trajectory $\mathbf{Q}_1, \ldots, \mathbf{Q}_n \in \mathrm{SE}(3)$. For simplicity of notation, we assume that the sequences are time-synchronized, equally sampled, and both have length n. In practice, these two sequences have typically different sampling rates, lengths and potentially missing data, so that an additional data association and interpolation step is required.

A. Relative pose error (RPE)

The relative pose error measures the local accuracy of the trajectory over a fixed time interval Δ. Therefore, the relative pose error corresponds to the drift of the trajectory which is in particular useful for the evaluation of visual odometry systems. We define the relative pose error at time step i as

$$\mathbf{E}_i := \left(\mathbf{Q}_i^{-1}\mathbf{Q}_{i+\Delta}\right)^{-1}\left(\mathbf{P}_i^{-1}\mathbf{P}_{i+\Delta}\right). \tag{1}$$

From a sequence of n camera poses, we obtain in this way $m = n - \Delta$ individual relative pose errors along the sequence. From these errors, we propose to compute the root mean squared error (RMSE) over all time indicies of the translational component as

$$\mathrm{RMSE}(\mathbf{E}_{1:n}, \Delta) := \left(\frac{1}{m}\sum_{i=1}^{m} \|trans(\mathbf{E}_i)\|^2\right)^{1/2}, \tag{2}$$

where $trans(\mathbf{E}_i)$ refers to the translational components of the relative pose error $\mathbf{E}_i$. It should be noted that some researchers prefer to evaluate the mean error instead of the root mean squared error which gives less influence to outliers. Alternatively, it is also possible to compute the median instead of the mean, which attributes even less

influence to outliers. If desired, additionally the rotational error can be evaluated, but usually we found the comparison by translational errors to be sufficient (as rotational errors show up as translational errors when the camera is moved). Furthermore, the time parameter Δ needs to be chosen. For visual odometry systems that match consecutive frames, $\Delta = 1$ is an intuitive choice; $\mathrm{RMSE}(\mathbf{E}_{1:n})$ then gives the drift per frame. For systems that use more than one previous frame, larger values of Δ can also be appropriate, for example, for $\Delta = 30$ gives the drift per second on a sequence recorded at 30 Hz. It should be noted that a common (but poor) choice is to set $\Delta = n$ which means that the start point is directly compared to the end point. This metric can be misleading as it penalizes rotational errors in the beginning of a trajectory more than towards the end [43], [45]. For the evaluation of SLAM systems, it therefore makes sense to average over all possible time intervals Δ, i.e., to compute

$$\mathrm{RMSE}(\mathbf{E}_{1:n}) := \frac{1}{n}\sum_{\Delta=1}^{n}\mathrm{RMSE}(\mathbf{E}_{1:n},\Delta). \tag{3}$$

Note that the computational complexity of this expression is quadratic in the trajectory length. Therefore, we propose to approximate it by computing it from a fixed number of relative pose samples. Our automated evaluation script allows both the exact evaluation as well as the approximation for a given number of samples.

B. Absolute trajectory error (ATE)

For visual SLAM systems, additionally the global consistency of the estimated trajectory is an important quantity. The global consistency can be evaluated by comparing the absolute distances between the estimated and the ground truth trajectory. As both trajectories can be specified in arbitrary coordinate frames, they first need to be aligned. This can be achieved in closed form using the method of Horn [52], which finds the rigid-body transformation $\mathbf{S}$ corresponding to the least-squares solution that maps the estimated trajectory $\mathbf{P}_{1:n}$ onto the ground truth trajectory $\mathbf{Q}_{1:n}$. Given this transformation, the absolute trajectory error at time step i can be computed as

$$\mathbf{F}_i := \mathbf{Q}_i^{-1}\mathbf{S}\mathbf{P}_i. \tag{4}$$

Similar to the relative pose error, we propose to evaluate the root mean squared error over all time indices of the translational components, i.e.,

$$\mathrm{RMSE}(\mathbf{F}_{1:n}) := \left(\frac{1}{n}\sum_{i=1}^{n}\|\mathit{trans}(\mathbf{F}_i)\|^2\right)^{1/2}. \tag{5}$$

VI. Case Studies

We are interested in learning how our dataset and benchmark has been received by the research community. Furthermore, we want to find out how the benchmark is used (in particular, which sequences have been found to be most useful), which evaluation metric are applied, and above all, to check to what degree (if at all) the benchmark actually contributes to scientific progress in the field. Therefore, we searched for scientific publications that used our benchmark for evaluation purposes using Google Scholar. We found a total of six peer-reviewed conference papers and one master thesis that used our benchmark for evaluation.

Endres et al. [8] analyzed in collaboration with our group the properties of the open-source RGB-D SLAM system [2]. The system is based on feature matching and Graph SLAM. The evaluation was carried out on all nine "fr1" sequences using the RPE metric. This evaluation was already performed during the creation of the dataset, and we made all estimated trajectories publicly available as a reference. On average, we measured the average error of RGB-D SLAM to 9.7 cm and 3.95° over all sequences. Furthermore, we analyzed the influence of the chosen feature descriptor and matcher and found that SURF provides the best trade-off between computation time and accuracy.

Osteen et al. [10] describe an approach to recover the rotational egomotion by matching two depth images using Fourier analysis. The key insight is that a convolution in the position space is equivalent to single product in frequency space which saves one order of magnitude in terms of computational complexity. As error metric, they computed the rotational error on a frame-to-frame basis (RPE). The error evolution was analyzed in great detail on the "fr1/room" and "fr1/360" sequences and an exhaustive evaluation over all "fr1" sequences was presented. Osteen et al. compared two variants of their approach to (1) GICP, (2) GICP in combination with visual odometry from OpenCV, and (3) the RGB-D SLAM system.

Thierfelder [11] investigated particle swarm optimization for egomotion estimation in her master's thesis. Particle swarm optimization can be seen as an extension of particle filtering where the particles interact with each other, i.e., are subject to attractive and repulsive forces. For the evaluation, the RPE was chosen as the error metric because for egomotion estimation the relative drift is the important quantity. A parameter study was conducted on two debugging sequences ("fr2/xyz" and "fr2/rpy"), followed by a quantitative comparison of their approach with the egomotion estimation of the RGB-D SLAM system on all "fr1"-sequences.

Andreasson and Stoyanov [9] convert the point clouds into a Gaussian mixture model using the normal distribution transform (NDT) in combination with visual features. The registration is carried out in the reduced space which is much more efficient than point-based ICP. In their evaluation, various variants of the algorithm are compared to each other on six benchmark sequences ("fr1/360", "fr1/desk", "fr1/desk2", "fr1/floor", "fr1/room", "fr2/desk") for which both the frame-to-frame RPE is specified.

Steinbrücker et al. [7] (from our group) developed and analyzed an energy-based approach to egomotion estimation based on image warping. In this approach, the transformation between two RGB-D frames is directly computed maximizing photo-consistency using non-linear minimization. In the evaluation, only the "fr1/desk" and "fr2/desk" sequence were analyzed. During debugging, "fr1/xyz" and "fr2/xyz" were used extensively. The accuracy and robustness of their

method was compared to GICP.

Stückler and Behnke [12], [13] extend the NDT approach for real-time applications. In particular, registration is performed from coarse-to-fine and the map is stored in an oct-tree to increase memory efficiency. The maximum level of the oct-tree depends on the distance to the sensor which better reflects the noise characteristics. As the (local) maps accumulate information from several frames, the robustness for larger camera motions is significantly increased in comparison to GICP and our warping method. The evaluation was performed on the "fr1/desk" and "fr2/desk" sequences and compared to the results of Steinbrücker et al. [7].

We also analyzed the log files of our webserver. Over the past 12 month, the benchmark website had more than 13.000 unique visitors. In total, we counted more than 290.000 file downloads (including the preview thumbnails, movies, and archives) with a total volume of almost 60 TB. Note that each sequence of the dataset has a size of 0.5–3 GB, while all files offered by us sum to 250 GB (with some redundancy, i.e., including both the tgz and bag files). Our online evaluation service was used more than 4.000 times.

VII. Discussion

In all before mentioned studies, the researchers evaluated the relative pose error (RPE) as proposed in our recent paper [5]. While this choice clearly facilitates comparisons of alternative approaches and relieves researchers from developing their own metrics (let alone, their re-implementation), the second proposed metric, the absolute trajectory error (ATE), has not been applied at all so far. To reduce ambiguities in the evaluation procedure, we therefore plan to retract the ATE and to encourage new users to solely evaluate the RPE. Our motivation for proposing the ATE for SLAM evaluation is that it can easily be visualized, but it is also strongly correlated to the RPE and thus redundant.

A second point for discussion is that in all of the published studies only a subset of the available sequences was used for benchmarking. Three of the five studies evaluated the approaches on all "fr1" sequences, while the other performed the evaluation on the "fr1/desk" and "fr2/desk" sequences. One reason for this might be that the "fr2" sequences were published three months later and are generally more challenging than the "fr1" sequences (larger room, robot-mounted Kinect, etc). Moreover, from the perspective of the individual researchers, it is also tedious to evaluate an approach on all available sequences. Therefore, we think that it might be helpful to pre-select a minimal subset of sequences for evaluation. This selection could either be recommended by us or could evolve naturally. For example, we believe that the "desk" sequences have often be chosen because they cover a large motion, end with a loop closure and contain both sufficient texture and structure and thus might be a good choice for one of the reference sequences.

The testing sequences, i.e., the sequences without public groundtruth, have not been used at all for evaluation so far. We believe that the reason for this is that the benchmark is still quite new and it makes at first more sense to evaluate one's approach on the training sequences. We hope that later, when more competitive and elaborate studies appear, these testing sequences will prove useful as an independent and unbiased performance measurement. To support this development, we plan to extend our website with automated submission and ranking tools. To add the ranking of a new algorithm to this website, it will then be necessary to evaluate the approach on a minimal set of (to be selected) essential sequences.

Lastly, we have been asked by several researchers whether we could provide additional log files with certain properties, for example, to record additional sequences with dynamic objects, a longer sequence in a household environment, sequences without texture and/or without structure, and data recorded with an Asus Xtion sensor. We started preparations to add these sequences. Given that we have already 39 sequences listed on the benchmark website, we plan to reduce the information overload for new benchmark users by re-organizing the list of sequences into meaningful groups of essential and additional sequences.

VIII. Conclusions

In general, the benefits of a benchmark are (1) that it is an effective and affordable way of evaluating such approaches and (2) that it comes with well-defined performance metrics. In this paper, we presented the TUM RGB-D benchmark and discussed its two error metrics. Furthermore, we reviewed the first scientific publications where it was used for evaluation. From this analysis, we conclude that the benchmark has evolved into a valuable tool for several researchers working on egomotion estimation and SLAM approaches. Based on the valuable discussions and feedback we obtained so far, we plan (1) to simplify the benchmark by reducing it to a few, relevant sequences and to a single error metric, (2) provide additional data for special purpose evaluations (which will however not become part of the regular benchmark), and (3) adding an automated submission and ranking system to the benchmark website to stimulate the comparison of alternative methods.

References

[1] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox, "RGB-D mapping: Using depth cameras for dense 3D modeling of indoor environments," in *Intl. Symp. on Experimental Robotics (ISER)*, 2010.

[2] N. Engelhard, F. Endres, J. Hess, J. Sturm, and W. Burgard, "Real-time 3D visual SLAM with a hand-held RGB-D camera," in *RGB-D Workshop on 3D Perception in Robotics at the European Robotics Forum*, 2011.

[3] C. Audras, A. Comport, M. Meilland, and P. Rives, "Real-time dense appearance-based SLAM for RGB-D sensors," in *Australasian Conf. on Robotics and Automation*, 2011.

[4] R. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon, "KinectFusion: Real-time dense surface mapping and tracking," in *Intl. Symposium on Mixed and Augmented Reality (ISMAR)*, 2011.

[5] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *Intl. Conf. on Intelligent Robot Systems (IROS)*, Oct. 2012.

[6] J. Sturm, S. Magnenat, N. Engelhard, F. Pomerleau, F. Colas, W. Burgard, D. Cremers, and R. Siegwart, "Towards a benchmark for RGB-D SLAM evaluation," in *RGB-D Workshop on Advanced Reasoning with Depth Cameras at RSS*, June 2011.

[7] F. Steinbrücker, J. Sturm, and D. Cremers, "Real-time visual odometry from dense RGB-D images," in *ICCV Workshop on Live Dense Reconstruction with Moving Cameras*, 2011.

[8] F. Endres, J. Hess, N. Engelhard, J. Sturm, D. Cremers, and W. Burgard, "An evaluation of the RGB-D SLAM system," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2012.

[9] T. Stoyanov, M. Magnusson, and A. Lilienthal, "Point set registration through minimization of the L2 distance between 3D-NDT models," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2012.

[10] P. Osteen, J. Owens, and C. Kessens, "Online egomotion estimation of RGB-D sensors using spherical harmonics," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2012.

[11] S. Thierfelder, "Modified particle swarm optimization for a 6 DOF local pose estimation algorithm by using a RGB-D camera," Master's thesis, 2012.

[12] J. Stückler and S. Behnke, "Model learning and real-time tracking using multi-resolution surfel maps," in *AAAI*, 2012.

[13] ——, "Robust real-time registration of rgb-d images using multi-resolution surfel representations," in *German Conference on Robotics (ROBOTIK)*, 2012.

[14] F. Lu and E. Milios, "Globally consistent range scan alignment for environment mapping," *Autonomous Robots*, vol. 4, no. 4, pp. 333–349, 1997.

[15] F. Dellaert, "Square root SAM," in *Proc. of Robotics: Science and Systems (RSS)*, Cambridge, MA, USA, 2005.

[16] E. Olson, J. Leonard, and S. Teller, "Fast iterative optimization of pose graphs with poor initial estimates," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2006.

[17] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *IEEE and ACM Intl. Symposium on Mixed and Augmented Reality (ISMAR)*, 2007.

[18] M. Kaess, A. Ranganathan, and F. Dellaert, "iSAM: Incremental smoothing and mapping," *IEEE Trans. on Robotics, TRO*, vol. 24, no. 6, pp. 1365–1378, Dec 2008.

[19] G. Grisetti, C. Stachniss, and W. Burgard, "Non-linear constraint network optimization for efficient map learning," *IEEE Transactions on Intelligent Transportation systems*, vol. 10, no. 3, pp. 428–439, 2009.

[20] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g2o: A general framework for graph optimization," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2011.

[21] H. Jin, P. Favaro, and S. Soatto, "Real-time 3-D motion and structure of point features: Front-end system for vision-based control and interaction," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2000.

[22] M. Pollefeys and L. Van Gool, "From images to 3D models," *Commun. ACM*, vol. 45, pp. 50–55, July 2002.

[23] D. Nistér, "Preemptive ransac for live structure and motion estimation," *Machine Vision and Applications*, vol. 16, pp. 321–329, 2005.

[24] J. Stühmer, S. Gumhold, and D. Cremers, "Real-time dense geometry from a handheld camera," in *DAGM Symposium on Pattern Recognition (DAGM)*, 2010.

[25] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM: A factored solution to the simultaneous localization and mapping problem," in *Prof. of the National Conf. on Artificial Intelligence (AAAI)*, 2002.

[26] G. Grisetti, C. Stachniss, and W. Burgard, "Improved techniques for grid mapping with rao-blackwellized particle filters," *IEEE Transactions on Robotics (T-RO)*, vol. 23, pp. 34–46, 2007.

[27] A. Nüchter, K. Lingemann, J. Hertzberg, and H. Surmann, "6D SLAM – 3D mapping outdoor environments: Research articles," *J. Field Robot.*, vol. 24, pp. 699–722, August 2007.

[28] M. Magnusson, H. Andreasson, A. Nüchter, and A. Lilienthal, "Automatic appearance-based loop detection from 3D laser data using the normal distributions transform," *Journal of Field Robotics*, vol. 26, no. 11–12, pp. 892–914, 2009.

[29] A. Segal, D. Haehnel, and S. Thrun, "Generalized-icp," in *Robotics: Science and Systems (RSS)*, 2009.

[30] K. Koeser, B. Bartczak, and R. Koch, "An analysis-by-synthesis camera tracking approach based on free-form surfaces," in *German Conf. on Pattern Recognition (DAGM)*, 2007.

[31] K. Konolige and J. Bowman, "Towards lifelong visual maps," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2009.

[32] H. Strasdat, J. Montiel, and A. Davison, "Scale drift-aware large scale monocular SLAM," in *Proc. of Robotics: Science and Systems (RSS)*, 2010.

[33] K. Konolige, M. Agrawal, R. Bolles, C. Cowan, M. Fischler, and B. Gerkey, "Outdoor mapping and navigation using stereo vision," in *Intl. Symp. on Experimental Robotics (ISER)*, 2007.

[34] A. Comport, E. Malis, and P. Rives, "Real-time quadrifocal visual odometry," *Intl. Journal of Robotics Research (IJRR)*, vol. 29, pp. 245–266, 2010.

[35] C. Stachniss, P. Beeson, D. Hähnel, M. Bosse, J. Leonard, B. Steder, R. Kümmerle, C. Dornhege, M. Ruhnke, G. Grisetti, and A. Kleiner, "Laser-based SLAM datasets." [Online]. Available: http://OpenSLAM.org

[36] "The Rawseeds project," http://www.rawseeds.org/rs/datasets/.

[37] M. Smith, I. Baldwin, W. Churchill, R. Paul, and P. Newman, "The new college vision and laser data set," *Intl. Journal of Robotics Research (IJRR)*, vol. 28, no. 5, pp. 595–599, 2009.

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

[39] F. Pomerleau, S. Magnenat, F. Colas, M. Liu, and R. Siegwart, "Tracking a depth camera: Parameter exploration for fast ICP," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2011.

[40] S. Bao and S. Savarese, "Semantic structure from motion," in *IEEE Intl. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[41] E. Olson and M. Kaess, "Evaluating the performance of map optimization algorithms," in *RSS Workshop on Good Experimental Methodology in Robotics*, 2009.

[42] R. Vincent, B. Limketkai, M. Eriksen, and T. De Candia, "SLAM in real applications," in *RSS Workshop on Automated SLAM Evaluation*, 2011.

[43] A. Kelly, "Linearized error propagation in odometry," *Intl. Journal of Robotics Research (IJRR)*, vol. 23, no. 2, 2004.

[44] K. Konolige, M. Agrawal, and J. Solà, "Large scale visual odometry for rough terrain," in *Intl. Symposium on Robotics Research (ISER)*, 2007.

[45] R. Kümmerle, B. Steder, C. Dornhege, M. Ruhnke, G. Grisetti, C. Stachniss, and A. Kleiner, "On measuring the accuracy of SLAM algorithms," *Autonomous Robots*, vol. 27, pp. 387–407, 2009.

[46] W. Burgard, C. Stachniss, G. Grisetti, B. Steder, R. Kümmerle, C. Dornhege, M. Ruhnke, A. Kleiner, and J. Tardós, "A comparison of SLAM algorithms based on a graph of relations," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2009.

[47] W. Wulf, A. Nüchter, J. Hertzberg, and B. Wagner, "Ground truth evaluation of large urban 6D SLAM," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2007.

[48] S. Baker, D. Scharstein, J. Lewis, S. Roth, M. Black, and R. Szeliski, "A database and evaluation methodology for optical flow," *Intl. Journal of Computer Vision (IJCV)*, vol. 92, no. 1, 2011.

[49] http://vision.in.tum.de/data/datasets/rgbd-dataset.

[50] PrimeSense, Willow Garage, SideKick and Asus, "Introducing OpenNI," http://http://www.openni.org.

[51] MotionAnalysis, "Raptor-E Digital RealTime System," http://www.motionanalysis.com/html/industrial/raptore.html.

[52] B. Horn, "Closed-form solution of absolute orientation using unit quaternions," *Journal of the Optical Society of America A*, vol. 4, pp. 629–642, 1987.