Machine Vision and Applications (2017) 28:61–74
DOI 10.1007/s00138-016-0802-6

ORIGINAL PAPER

Toward evaluation of visual navigation algorithms on RGB-D data from the first- and second-generation Kinect

Marek Kraft[1] · **Michał Nowicki**[1] · **Adam Schmidt**[1] · **Michał Fularz**[1] · **Piotr Skrzypczyński**[1]

Received: 27 December 2015 / Revised: 12 July 2016 / Accepted: 14 July 2016 / Published online: 30 July 2016

Abstract Although the introduction of commercial RGB-D sensors has enabled significant progress in the visual navigation methods for mobile robots, the structured-light-based sensors, like Microsoft Kinect and Asus Xtion Pro Live, have some important limitations with respect to their range, field of view, and depth measurements accuracy. The recent introduction of the second- generation Kinect, which is based on the time-of-flight measurement principle, brought to the robotics and computer vision researchers a sensor that overcomes some of these limitations. However, as the new Kinect is, just like the older one, intended for computer games and human motion capture rather than for navigation, it is unclear how much the navigation methods, such as visual odometry and SLAM, can benefit from the improved parameters. While there are many publicly available RGB-D data sets, only few of them provide ground truth information necessary for evaluating navigation methods, and to the best of our knowledge, none of them contains sequences registered with thenew version of Kinect. Therefore, this paper describes a new RGB-D data set, which is a first attempt to systematically evaluate the indoor navigation algorithms on data from two different sensors in the same environment and along the same trajectories. This data set contains synchronized RGB-D frames from both sensors and the appropriate ground truth from an external motion capture system based on distributed cameras. We describe in details the data registration procedure and then evaluate our RGB-D visual odometry algorithm on the obtained sequences, investigating how the specific properties and limitations of both sensors influence the performance of this navigation method.

Keywords Data set · Evaluation · Visual navigation · RGB-D sensor

This work was supported by the grant from National Science Centre, Poland, funded according to the decision DEC-2013/09/B/ST7/01583.

✉ Marek Kraft
Kraft.Marek@put.poznan.pl

Michał Nowicki
Nowicki.Michal@put.poznan.pl

Adam Schmidt
Schmidt.Adam@put.poznan.pl

Michał Fularz
Fularz.Michal@put.poznan.pl

Piotr Skrzypczyński
Skrzypczynski.Piotr@put.poznan.pl

[1] Institute of Control and Information Engineering, Poznań University of Technology, ul. Piotrowo 3A, 60-965 Poznan, Poland

1 Introduction

Publicly available data for algorithm evaluation are instrumental to achieving scientific progress in many disciplines. Providing a common ground for objective and reliable bench marking improves research transparency and reproducibility [21]. The fields of computer vision and robotics are no exception. One relevant problem in robotics—the vision-based mobile robot navigation—is situated on the crossroads of these fields. Moreover, recent launch of compact, inexpensive RGB-D sensors based on structured-light [18] or time-of-flight cameras [19] provided the robotics community with an attractive sensing solution, enabling the integration of the depth and vision data within the robot navigation processing pipeline for increased accuracy.

The goal of vision-based navigation is the recovery of camera trajectory as a series of sensor poses based on the sensor data (i.e., visual odometry (VO) problem [13,28]).

Another approach to visual trajectory reconstruction is simultaneous localization and mapping (SLAM) [7,8]. The VO approach is usually key frame based. In most cases, the transformation function for coordinates of sparse sets of points matched across a number of frames is known [17,35]. This enables the use of various frameworks for inter-frame transformation error minimization for establishing a series of consecutive sensor displacements and poses [34]. The canonical SLAM approach on the other hand is based on filtering and aims at accumulating knowledge from past measurements to update the structure and motion information in the form of a possibly accurate probability distribution. The two concepts are combined within the recently popular graph-based SLAM. In this approach, the nodes of the graph correspond to poses of the robot and the edges in the graph express the spatial constraints between the nodes. The constraints are introduced based on successive measurements. The spatial configuration of the nodes that are maximally consistent with the measurements is found by solving an error minimization problem [15].

In this paper, a new RGB-D data set containing the measurements performed using both the first- and the second-generation Kinect (hereinafter Kinect v1 and Kinect v2, respectively) is described. From here on out, we will refer to the data set as PUTK^2—PUT Kinect 1 and Kinect 2 data set. The data set consists of eight sequences, registered in an office-like environment. The sequences are composed of color images and depth data frames acquired by both the sensors and are supplemented with accurate ground truth, enabling full reconstruction of relative displacement and pose changes across all sensor positions. The ground truth was established using an overhead system of mutually calibrated cameras. Moreover, great care was taken to ensure proper synchronization of the acquisition by both sensors and the overhead camera system, which further contributes to the overall accuracy of the trajectory reconstruction. The availability of precise ground truth data facilitates reliable navigation algorithm benchmarking using both color and depth images. The successful use of typical CCD cameras, instead of a commercial IR-based tracking system, like Vicon [36] or OptiTrack [23], to generate ground truth for a navigation-related data set demonstrates that with some extra effort on the programming and calibration side such a data set can be obtained at a fraction of the costs that are necessary to experiment with any of the commercial motion capture systems. Furthermore, the use of both versions of the Kinect sensors enables accurate side by side comparisons of characteristics of both devices.

To demonstrate the usefulness of the acquired data, results of the evaluation of a VO algorithm using the obtained sequences are presented and discussed. The evaluation includes the investigation of the influence of properties of both Kinect sensors on visual navigation performance. VO was chosen over SLAM for demonstration purposes, as for VO only the local consistency of the trajectory and the local map (sensor displacement model) is used to obtain the local trajectory estimate, whereas SLAM aims at assuring the consistency of the global map [37]. By its very nature, VO is therefore burdened by the unbounded local drift, but depends more directly on the quality of the input data. The complete benchmark data are available for download at: http://lrm.put.poznan.pl/putkk/ Aside from the images and trajectories, the Web site contains an additional, detailed description of the data formats and sample videos of the registered sequences.

This paper is organized as follows. Section 2 describes related work—other available vision-based navigation benchmark data and work on the comparison of Kinect sensors. Section 3 gives a brief description on the multi-camera acquisition setup for trajectory reconstruction, its configuration and calibration. The registered trajectories and their properties are presented in Sect. 4. Section 5 presents the VO algorithm using the registered data for accuracy benchmark, and the detailed results of the tests are presented in Sect. 6. Finally, Sect. 7 contains conclusions and summarizes the paper.

2 Related work

The need for common ground for reliable and objective robot navigation algorithm evaluation is long acknowledged in the research community. As algorithms improved and more sensors applied to this task, databases for laser scanner and vision-based navigation emerged [5,31]. Unfortunately, these early benchmarks do not include the depth data used by many of the state-of-the-art navigation algorithms. Moreover, as shown in [29], the data sets may contain improperly labeled data, and the acquisition process is usually not synchronized, which often limits their usefulness. Introduction of inexpensive RGB-D sensors spurred interest in their applications in computer vision and robotics. This led to the emergence of a range of data sets containing RGB-D data. While most of the publicly available benchmark data are meant for applications such as object segmentation and recognition [25,30,38], data sets for navigation evaluation have also been published.

The benchmark first presented in [33] and expanded in [32] contains 39 corresponding RGB and D image sequences gathered using the Asus Xtion (based on Kinect 1) sensor with ground truth registered using a motion capture system. Regrettably, for lengthy parts of the sequences taken by the sensor mounted on the robot, no physical objects are visible within the depth sensor's working range. Under such conditions, the inter-frame displacement and pose change cannot be established without resorting to a vision-only approach based on RGB images. Moreover, the data from the motion capture system and the data from the Kinect sensor are not perfectly synchronized because of the different sampling frequencies and potentially missing data, so that, an additional data association and interpolation step is required. A very large scale data set, containing data acquired along a 42 km indoor trajectory using stereo camera, laser scanners, IMU and the Kinect v1 sensor, is described in [11]. The ground truth for the data set was collected by aligning the laser scan results with the highly accurate construction plans of the building. The authors do not, however, describe how the sensor data are synchronized with the acquired ground truth trajectory. As stated in the article, the accuracy of the system is believed to be about 2–3 cm, but no detailed analysis or evaluation of this claim is given. While the system is undoubtedly very useful for evaluation of long-term autonomy, the solution presented in this paper is a better choice for navigation accuracy testing due to more precise ground truth and near-perfect synchronization. Another interesting approach to RGB-D-based navigation bench-marking is presented in [16]. The data set consists of images obtained from camera trajectories in ray-traced 3D models. As the data set is fully synthetic, perfect ground truth for trajectories and structure is directly available. However, such benchmarks, although useful, cannot fully replace the evaluation in real-life conditions. As shown in [12] and [18], the Kinect sensors exhibit distinctive characteristics, which influence the acquisition process significantly. To the knowledge of the authors, none of the currently available navigation benchmark contain the data registered using both types of Kinect sensors. The articles containing comparison of the accuracy of both types of Kinect sensor focus mostly on applications outside of robotics [1,39]. The rare exceptions deal with the properties of the sensors themselves, evaluated in the static, laboratory environment [20] or scenarios other than navigation such as recent work [14], which considers the tasks of 3D reconstruction and object recognition when comparing Kinect v2 with the RGB-D sensors based on structured light. The quantitative comparison with respect to ground truth obtained with a metrological laser scanner presented in [14] revealed that Kinect v2 provides less error in the mapping between the RGB and depth frames, and the obtained depth values are more constant with distance variations.

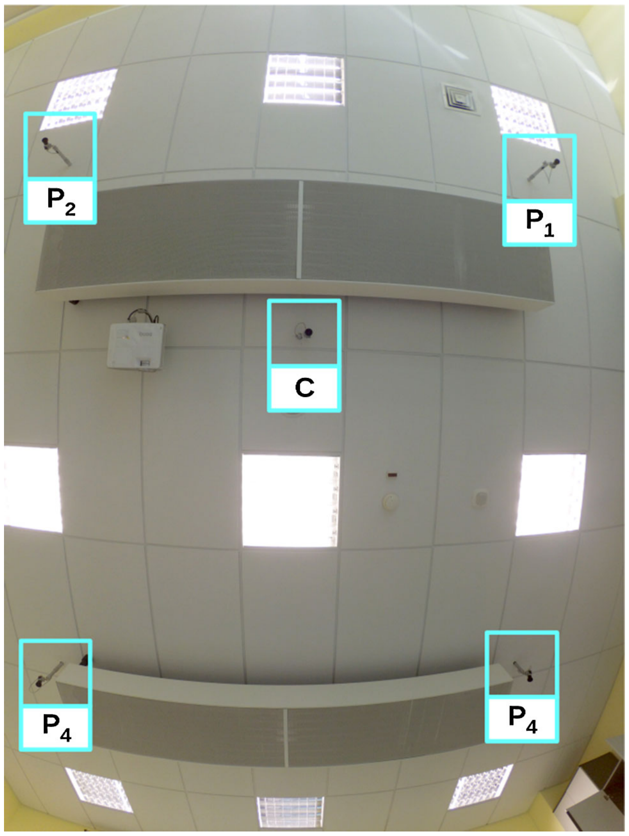

Fig. 1 Arrangement of the cameras in the multi-camera system

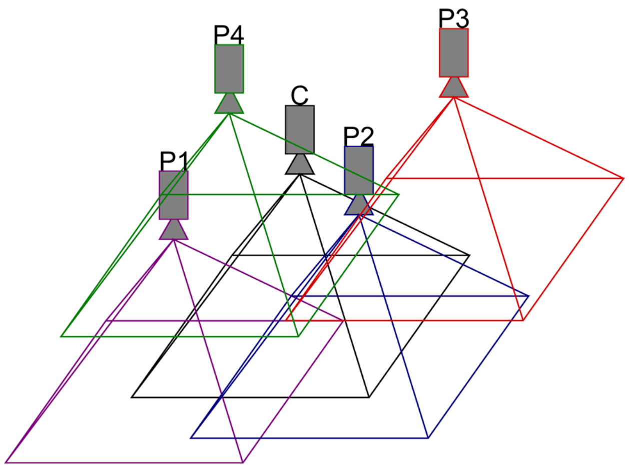

Fig. 2 Overlapping FOVs of the multi-camera system

3 Vision-based motion registration

3.1 Structure of the multi-camera vision system

The multi-camera vision system used for the registration of the PUTK^2 data set consists of five high-resolution Basler acA1600 cameras equipped with low-distortion, aspherical 3.5-mm lenses. The cameras are installed in an X-like pattern under the ceiling of the laboratory. Due to this particular arrangement, the field of view (FOV) of each of the peripheral cameras partially overlaps with FOVs of the two neighboring peripheral cameras and the central one. The cameras' arrangement is shown in Fig. 1, whereas Fig. 2 presents the idea of the overlapping FOVs. As a result, the central part of the room, in which most of the movement occurs, is observed by at least two cameras (usually three). This increases the accuracy of the reconstructed ground truth trajectory and facilitates the calibration of the multi-camera system.

Table 1 The intrinsic parameters of the multi-camera system

Cam.	Error	f_x	f_y	c_x	c_y	k_1	k_2	k_3	k_4	k_5	k_6	p_1	p_2
C	0.275	824.51	823.26	798.01	592.85	0.031	−0.020	−0.001	0.001	0.132	0.073	−0.111	0.175
P_1	0.244	828.94	827.94	804.31	625.73	0.056	−0.049	0.000	0.001	0.160	0.098	−0.140	0.203
P_2	0.300	826.91	825.86	798.34	594.55	−0.008	0.062	0.001	0.001	−0.020	0.039	−0.041	0.031
P_3	0.228	825.26	824.01	803.99	617.77	1.016	0.070	0.001	0.000	0.025	1.081	−0.055	0.078
P_4	0.301	828.58	827.40	789.80	609.33	0.001	0.058	0.001	0.000	0.011	0.046	−0.037	0.058

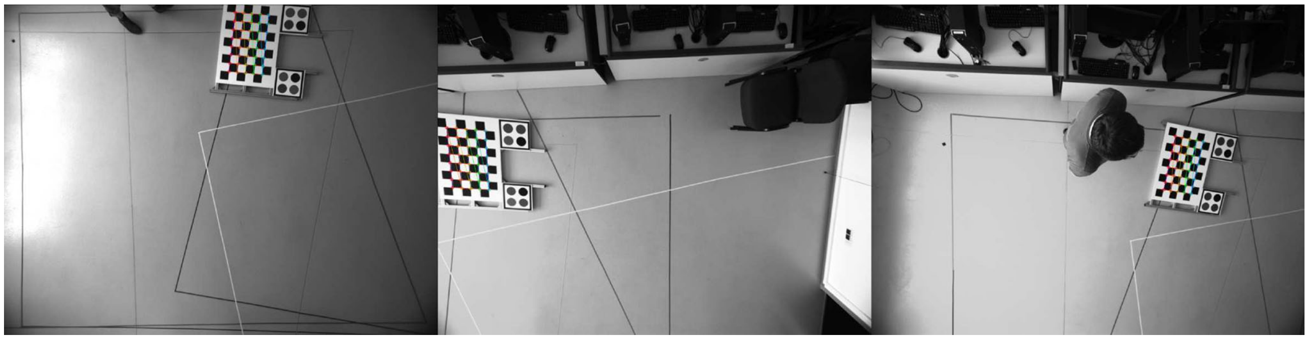

Fig. 3 Calibration marker observed simultaneously by three cameras

Each of the cameras was calibrated according to the rational lens model proposed by Claus and Fitzgibbon [6], assuming that the projection of a 3D point $p = \begin{bmatrix} x & y & z \end{bmatrix}^T$ onto image coordinates $q = \begin{bmatrix} u & v \end{bmatrix}^T$ is calculated as:

$$x' = \frac{x}{z} \tag{1}$$

$$y' = \frac{y}{z} \tag{2}$$

$$r^2 = x'^2 + y'^2 \tag{3}$$

$$R = \frac{1 + k_1 r^2 + k_2 r^4 + k_3 r^6}{1 + k_4 r^2 + k_5 r^4 + k_6 r^6} \tag{4}$$

$$x'' = x'R + 2p_1 x'y' + p_2(r^2 + 2x'^2) \tag{5}$$

$$y'' = y'R + 2p_1(r^2 + 2y'^2) + p_2 x'y' \tag{6}$$

$$u = f_x x'' + c_x \tag{7}$$

$$v = f_y y'' + c_y \tag{8}$$

where f_x, f_y, c_x and c_y are the focal lengths and the principal point, while k_i an p_i stand for the ith radial and tangential distortion coefficients, respectively. The obtained intrinsic parameters and the corresponding average reprojection errors are given in Table 1.

The precise reconstruction of the robot's trajectories requires knowing the exact poses of the cameras. The global multi-camera system calibration method proposed by Schmidt et al. [29] was used. Observations of a calibration marker placed in different poses, in which it is visible from at least two cameras (Fig. 3) are used to simultaneously minimize the reprojection error on all the images registered by all the cameras using the Levenberg–Marquardt algorithm.

Table 2 The rotation (r_x, r_y, r_z) and translation (t_x, t_y, t_z) vectors of the peripheral cameras with respect to the Marker's coordinate system

Cam.	r_x	r_y	r_z	$t_x[m]$	$t_y[m]$	$t_z[m]$
P_1	−0.16	0.63	−3.05	−2.31	1.79	0.17
P_2	−0.26	−0.24	−0.15	1.33	−1.63	0.13
P_3	−0.43	−0.11	−0.11	−2.29	−1.65	0.09
P_4	0.03	−0.58	3.06	1.30	1.76	0.13

The model parameters vector β_{MCS} consists of the poses of the peripheral cameras and the poses of the calibration marker during consequent observations:

$$\beta_{\text{MCS}} = \begin{bmatrix} r_{P_1} & t_{P_1} & \cdots & r_{P_4} & t_{P_4} & r_{M_1} & t_{M_1} & \cdots & r_{M_N} & t_{M_N} \end{bmatrix} \tag{9}$$

where r_{P_i} and t_{P_i} stand for the orientation described using the Rodrigues' rotation formula and translation vectors of the ith peripheral camera w.r.t. central camera's coordinate system. Similarly, r_{M_i} and t_{M_i} represent the ith pose of the calibration. The results of the calibration procedure are presented in Table 2 and in Fig. 4.

3.2 Experimental setup with the Kinect sensors

The mobile robot used in the experiments was equipped with two sensors: Kinect v1 and Kinect v2 facing forward, and

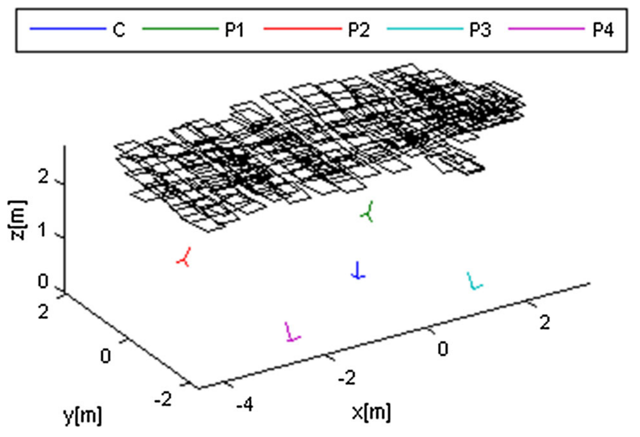

Fig. 4 Estimated poses of the peripheral cameras (P1—*green*, P2—*red*, P3—*cyan* and P4—*magenta*) and poses of the calibration marker (*black*) with regard to the central camera coordinate system (*blue*) (color figure online)

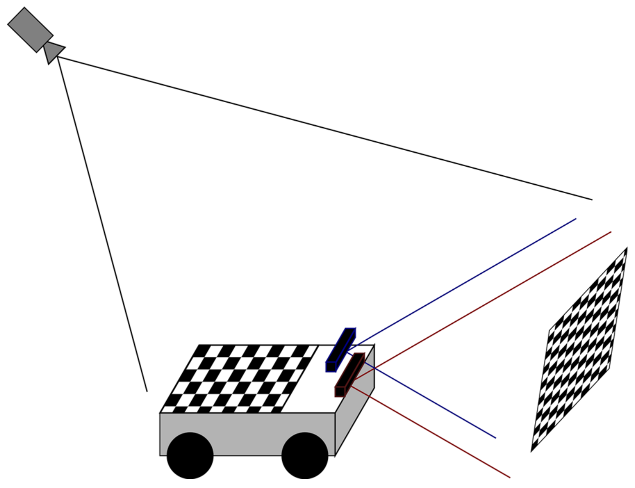

Fig. 5 Calibration of the Kinect sensors w.r.t. the Marker's coordinate system

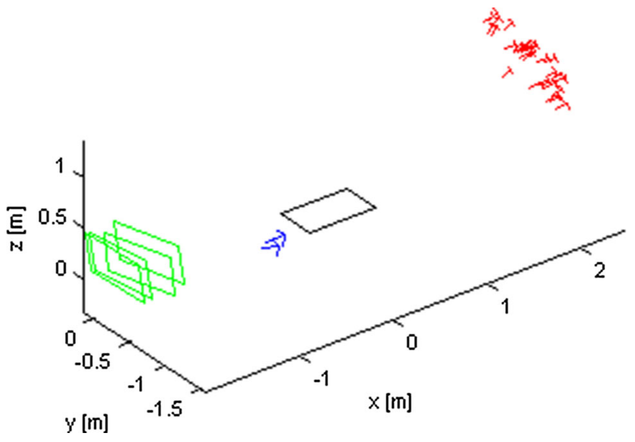

Fig. 6 Estimated poses of the two Kinect sensors (*blue*), the external camera (*red*) and the external marker (*green*) w.r.t. Robot Marker's coordinates (*black*) (color figure online)

Table 3 Rotation (r_x, r_y, r_z) and translation (t_x, t_y, t_z) vectors of the Kinect sensors

Sensor	r_x	r_y	r_z	t_x (m)	t_y (m)	t_z (m)
K_1	−1.15	−1.18	1.23	−0.11	−0.18	−0.07
K_2	−1.03	−1.03	1.29	−0.10	−0.11	−0.21

a firmly attached chessboard marker facing up. The marker was used to calculate the ground truth trajectory of the robot.

As the visual navigation algorithms track the pose of the moving camera, it was necessary to estimate the transformations between the robot's sensors and the chessboard marker. The algorithm described in [29] was used for that purpose. The method uses images of an external calibration marker observed by the robot's sensors and images registered by an external camera observing both the external and the robot's marker (Fig. 5).

Before the procedure, both the Kinect sensors (K_1 and K_2) and the external camera (E) were calibrated using the rational model. Afterward, the poses of the Kinect sensors w.r.t. the Robot Marker were determined by simultaneous minimization of the reprojection error of the chessboard patterns' (both the external and the robot's) corners on all the images registered by the Kinect sensors and the external camera. Figure 6 shows the visualization of the estimated sensor's positions, while Table 3 contains the numerical data.

The registration system schematic is presented in Fig. 7. It consisted of five overhead cameras, two Kinect sensors, three computers and some additional network equipment. Such distributed solution was necessary as the stream of data from sensors was to intensive to handle for a single device. Thus, one of the computers was responsible for capturing video streams from the overhead cameras. Dedicated network adapters were used to facilitate this task, providing separate connection for each camera. The same machine was also used as a server that synchronized the data acquisition procedure. Two low-profile notebooks were mounted on the robot for grabbing frames from both Kinect sensors. To provide the throughput necessary to save all the acquired data, each computer was equipped with a fast SSD disk.

Registration system elements were connected by Ethernet cable, and the TCP/IP-based communication was used. This solution was preferred over wireless, using either WiFi or proprietary modules, as it offered the lowest and more importantly the most stable delay. The delay was monitored during the whole experiment and was below 1 ms.

The server awaits for all the elements of the system to be in ready state, and when that happens, the trigger signal is sent. That way all the acquired images are synchronized, noting that the maximum frequency of operation was only about 11 frames per second, as the system works as fast as the slowest component.

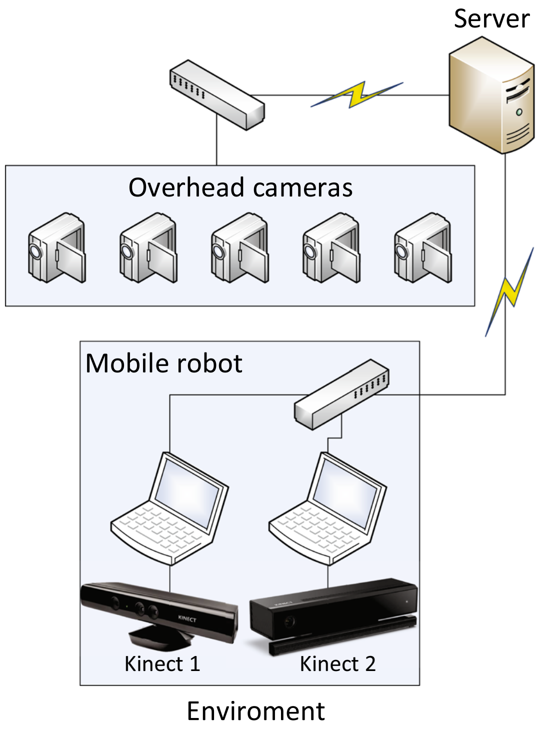

Fig. 7 Functional schematics of the registration system

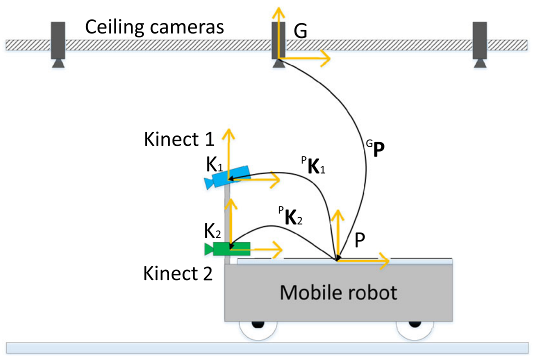

Fig. 8 Coordinate systems of Kinect v1 (K_1), Kinect v2 (K_2), ceiling cameras/global (G) and the pattern on a mobile robot (P)

4 Gathering the RGB-D data

The robotic setup used for our experiments with Kinect v1 and Kinect v2 sensors and vision-based ground truth system is presented in Fig. 8.

In this setup, the ceiling-mounted cameras provide information about the motion of the image calibration pattern attached to the robot (coordinate system P) in the coordinate system of the ceiling cameras, considered as the global coordinate system (G), which can be written as ${}^{G}\mathbf{P}$. Transformations between the coordinate systems of Kinects (K_1 and K_2) and the coordinate system of the pattern mounted on the robot P were found prior to operation with additional external calibration (Sect. 3.2) and are denoted as ${}^{P}\mathbf{K}_1$ and ${}^{P}\mathbf{K}_2$ for Kinect v1 and Kinect v2, respectively. To ease the use of the data, we provide the ground truth trajectory of the Kinect v1 and the Kinect v2 with respect to the global coordinate system G, which was computed with the following equation:

$$ {}^{G}\mathbf{P}\,{}^{P}\mathbf{K}_i = {}^{G}\mathbf{K}_i, \qquad (10) $$

where ${}^{G}\mathbf{K}_i$ is the current Kinect ground truth pose estimate in the global coordinate system G and i stands for 1 or 2 for the Kinect v1 or the Kinect v2, respectively. Finally, ground truth sensor trajectories for each of the Kinects and for each experiment in the data set are made publicly available, accompanying the respective sequence of RGB and depth images.

Eight robot trajectories were registered during the experiments. The first four of them simulated the indoor exploration scenario with its characteristic features:

– long, approximately straight sections,
– sudden turns,
– multiple loops,
– moving backwards.

The main purpose of those trajectories is to provide benchmarking material for the comparison of the visual navigation algorithms using either the Kinect v1 or the Kinect v2 sensors in a typical conditions. Figure 9 contains the reconstructions of those trajectories.

During the remaining four trajectories, the robot moved approximately along a path with speeds varying between the trials. The purpose of this part of the data set is to provide data suitable for evaluation of both sensors' robustness to different motion speeds. The overlaid reconstructed trajectories are presented in Fig. 10.

The trajectories were reconstructed offline. The consecutive poses of the robot were calculated independently according to the observations from the motion registration system. Due to the overlapping FOVs of the cameras (cf. Fig. 2), the robot was usually observed by more than one camera. The Levenberg–Marquardt algorithm was used to find the robot pose in the coordinate system of the central camera by minimizing the average reprojection error of the chessboard marker corners on all the images captured for this pose. Table 4 contains the number of frames, the length and the average robot's velocity for the registered trajectories. All the RGB and depth images are stored in a loss-less PNG format to ensure that image compression will not influence the results of any evaluation using the images as source data. The depth data are stored in a 16-bit grayscale image, in which a single bit corresponds to a depth of 1 mm. The supplementary ground truth position data (position and orientation in quaternion format) have the same structure and

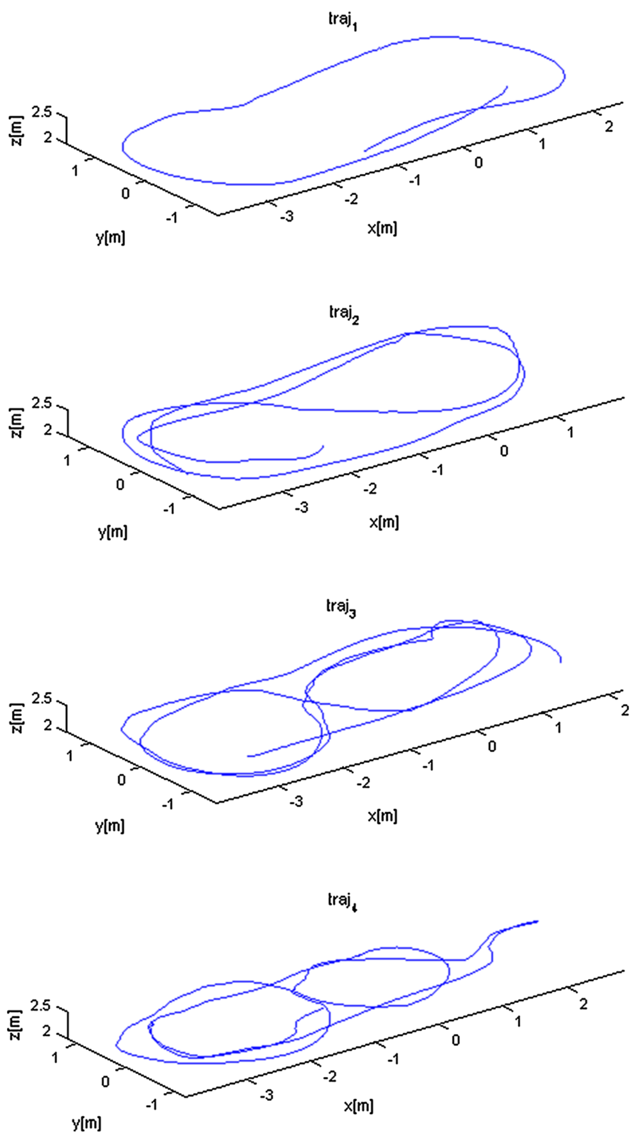

Fig. 9 The first four trajectories of the robot w.r.t. coordinate system of the central camera

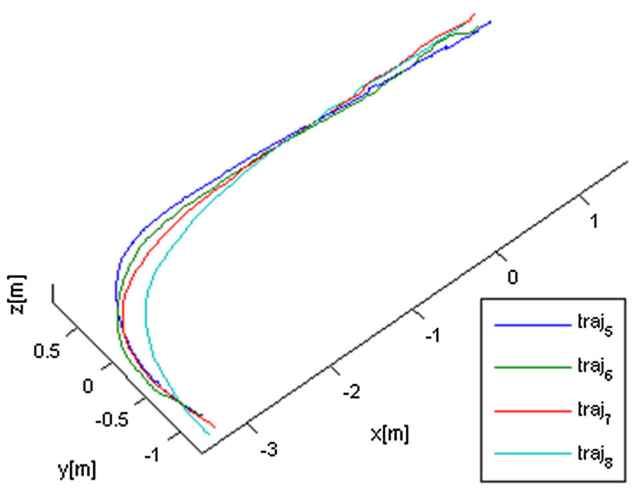

Fig. 10 The last four trajectories of the robot

Table 4 The parameters of the registered trajectories

Traj.	No. of frames	Length (m)	mean vel. (m/s)
$traj_1$	1539	18.7064	0.1215
$traj_2$	2563	31.6145	0.1233
$traj_3$	2754	39.2709	0.1426
$traj_4$	2854	33.3703	0.1169
$traj_5$	410	6.5381	0.1595
$traj_6$	169	6.3842	0.3778
$traj_7$	86	6.4865	0.7542
$traj_8$	59	6.1143	1.0363

format as the ones provided by the data set described in [32]. Camera calibration data are also included, along with sample code snippets for Kinect image distortion correction.

5 RGB-D visual odometry as the test application

5.1 The PUT RGB-D visual odometry system

The VO algorithm enables to determine the motion of the sensor using only a sequence of images, without creating a map of the environment [28]. A VO algorithm can also serve as a base for a solution to the SLAM problem formulated as the pose-based SLAM using graph optimization [4]. The use of RGB-D frames that contain direct depth measurements enables the use of 3D-to-3D feature correspondences for frame-to-frame motion estimation, instead of the 2D-to-2D correspondences in monocular VO [13].

The goal is to estimate the motion between the first frame $I_{v(k)}$ and the last frame $I_{v(k+n)}$ in a sequence of n RGB-D frames. This can be accomplished by matching the salient visual features from the first and the last frame, followed by the estimation of the geometric transformation between two sets of 3D points. However, the matching-based approach commonly used in RGB-D VO and SLAM is computationally demanding and often requires hardware acceleration [9]. Therefore, we have proposed an alternative approach, based on sparse optical flow tracking of the visual features [22]. With this approach, we detect the point features in the $I_{v(k)}$ frame and then track these points through the n images. The tracked features define the correspondences between $I_{v(k)}$ and $I_{v(k+n)}$. Then, the depth data are associated with the visual key points resulting in two sets of matched 3D points. The structure of the used VO pipeline is presented in Fig. 11.

The data processing starts with the detection of a set of point features (key points) that should be localized precisely in the image and be repeatable in a sequence of images showing the same scene. In our previous work [4], we have

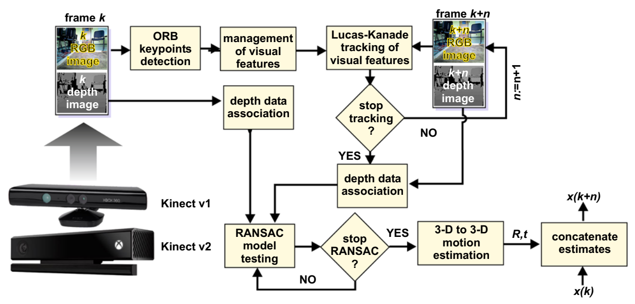

Fig. 11 Block scheme of the PUT RGB-D VO system

investigated three point feature detectors: FAST [26], ORB [27] and SURF [3], and we have found the ORB features to be the most suitable, due to their multi-scale corner-like detector, which yields highly repeatable features at a reasonable computation effort.

To make the whole feature extraction process more robust, we detect the features in sub-images and employ clustering of the resulting key points. The detection of features in separate, slightly overlapping sub-images helps to distribute the key points evenly on the image. The RGB image is divided into 16 equal, square sub-images, and the ORB detection is performed individually in each window with the adaptation of detector parameters to ensure a similar number of features in each square. After feature detection, the DBScan [10]—a fast, unsupervised clustering algorithm—is used to detect groups of key points and, then, one strongest key point from each group is selected. This way nearby features that were detected on a small area of the image are represented by one key point, which helps to avoid the aliasing of key points resulting in false feature correspondences.

The core part of the VO pipeline is motion estimation based on two sets of corresponding 3D point features, whose correspondences are determined by sparse optical flow tracking. The VO tracks features over a sequence of RGB images between the two key frames that are processed with depth images. Key points are detected at the key frame, and then the positions of these points in the new image in a sequence are determined by searching locally. To accomplish this, the pyramid implementation of the Lucas–Kanade algorithm [2] is applied. This algorithm is initialized with key points from the ORB detector. If the number of successfully tracked features falls below a given threshold, new key points are detected in the current image and inserted into the pool of tracked features. When the maximum number n of the RGB frames in a sequence is reached (the default value is $n = 4$), the rigid transformation between the key frames is computed. The transformation estimation procedure is embedded in the RANSAC scheme to make it robust to outliers resulting from imperfect tracking. In every iteration, the RANSAC randomly selects three pairs of points from the set of tracked key points and estimates the candidate transformation using the Umeyama algorithm [35]. A modern variant of the RANSAC algorithm is used, like in [24], that estimates the number of necessary iterations and allows an iterative correction of the final transformation by rejecting the inlier pairs that are the least probable within the estimated model.

5.2 Evaluating the PUT RGB-D VO on the PUTK^2 data set

The PUT RGB-D VO system was already extensively tested on publicly available data sets, and the results were published in [4]. The data sets that were used so far were the TUM RGB-D benchmark [32] and the ICL-NUIM data set [16]. The evaluation revealed that the tracking-based VO is fast and accurate, but only if it is fed by good quality images at a high frame rate [4]. Therefore, we consider the PUT RGB-D VO system a good candidate to evaluate our new Kinect v1 and Kinect v2 data set, as the varying performance of the VO should demonstrate the importance of such factors like image resolution, sensor field of view, the presence of motion blur and depth artifacts.

The operation of the PUT RGB-D VO system on the RGB and depth images from Kinect v1 and Kinect v2 is mainly the same, but some minor modifications were necessary to accommodate the Kinect v2 data. The main reason for these changes is the different resolution of the images provided by both sensors. The Kinect v1 yields RGB images of size 640×480 and depth images of size 320×240 that are rescaled to match the RGB images. However, the new Kinect v2 produces RGB images of size 1920×1080 and depth images of size 512×424 that again are rescaled to match the color images. Due to the significant difference in image sizes, all parameters or thresholds that are defined in pixel values needed to be properly adjusted. An exemplary parameter is the maximum distance between points belonging to the same cluster in the DBScan algorithm. However, we have decided to still divide RGB frames into the same number of sub-images in both cases. During the tests, the images from both sensors were processed in the VO pipeline, but not memorized, as due to the size of Kinect v2 images they would quickly fill whole RAM of a typical PC.

We demonstrate the trajectory reconstruction accuracy achieved on the Kinect v1 and Kinect v2 sequences. The quantitative results are obtained with respect to the ground truth trajectories, applying the relative pose error (RPE) and absolute trajectory error (ATE) metrics introduced in [32]. The RPE is well suited for measuring the drift of a VO system, like the one we apply in our evaluation of the Kinect v1 and Kinect v2 data. The ATE metric is appropriate rather for full SLAM systems that are able to correct the drift of the estimated trajectory [4]. Although for a VO system the absolute trajectory errors grow with time due to the unavoidable drift, we use this metric to demonstrate visually how far is the estimated trajectory from the ground truth one.

The RPE error corresponds to the local drift of the estimated trajectory. To compute RPE, the relative transformation between the neighboring points of the ground truth trajectory $\mathbf{T}^{\mathrm{GT}} = \mathbf{T}_1^{\mathrm{GT}}, \ldots, \mathbf{T}_k^{\mathrm{GT}}$ and the estimated trajectory $\mathbf{T}^{\mathrm{E}} = \mathbf{T}_1^{\mathrm{E}}, \ldots, \mathbf{T}_k^{\mathrm{E}}$ is computed, and the relative error at the time stamp i is given by:

$$\mathbf{E}_i = \left((\mathbf{T}_i^{\mathrm{GT}})^{-1}\mathbf{T}_{i+n}^{\mathrm{GT}}\right)^{-1} \left((\mathbf{T}_i^{\mathrm{E}})^{-1}\mathbf{T}_{i+n}^{\mathrm{E}}\right), \tag{11}$$

where n matches the length of the sequence of RGB-D frames tracked by the VO system (cf. Sect. 5.1). Taking the translational or rotational part of $\mathbf{E}_i$, we obtain the translational or rotational RPE, respectively.

The ATE error is based on the Euclidean distances between the estimated trajectory $\mathbf{T}^{\mathrm{E}}$ and the ground truth trajectory $\mathbf{T}^{\mathrm{GT}}$. At first, we map the estimated trajectory onto the ground truth trajectory by computing the transformation $\mathbf{T}^{\mathrm{S}}$ that is the least-square solution to the alignment problem [32]. Then, the error is computed as:

$$\mathbf{F}_i = (\mathbf{T}_i^{\mathrm{GT}})^{-1}\mathbf{T}^{\mathrm{S}}\mathbf{T}_i^{\mathrm{E}}, \tag{12}$$

for each ith trajectory node (time stamp). We extract the translational component of $\mathbf{F}_i$ and compute the root-mean-square error (RMSE), along with the standard deviation over all time indices. We use the evaluation tools provided with the TUM RGB-D benchmark [32] to compute the RPE and ATE metrics in our experiments.

6 Evaluation results

All experiments with the VO system were performed on a desktop PC with Intel Core i7-2600 3.4 GHz CPU and 16 GB RAM. The VO uses only a single core of the processor. On the tested sequences, our VO pipeline was running at about 50 frames per second (fps) for the Kinect v1 and at about 10 fps for the Kinect v2 data. As the data sets were recorded at 11 fps, the VO performance can be considered real time for both sensors. Details as to the processing times in the VO pipeline are as follows. The feature detection and management time in a single RGB frame averaged over the tested sequences was 4.25 ms for Kinect 1 and 10.77 ms for Kinect 2. However, considering that detection of the point features is performed only when the number of successfully tracked features falls below a threshold, the average detection time per frame was 1.63 ms for the Kinect v1 and 1.31 ms for the Kinect v2. The average tracking time between two consecutive RGB-D key frames in a sequence was 3.21 ms for the Kinect v1 and 18.11 ms for the Kinect v2. The transformation estimation (RANSAC) took less than 1 ms for both of Kinects—this was possible because only few iterations of RANSAC were necessary to find an acceptable transformation model, which is attributed to the good feature associations maintained by the Lucas–Kanade tracker. In all tests, the maximum number of RGB images for Lucas–Kanade tracking was set to $n = 4$, and the maximum number of tracked features was 500.

The impact of the used RGB-D data on the quality of the estimated trajectories has been investigated using the VO system, to avoid a situation when the results are altered by trajectory optimization in a SLAM back end. The RPE computed every n frames using (11) reveals relative errors in translation between the successive RGB-D key frames of the VO system. Figure 12 shows the relative translational errors for the VO system tested on the four sequences that resemble indoor exploration trajectories of a mobile robot. The qualitative difference between the results obtained using the data from Kinect v1 and Kinect v2 is clearly visible on all these plots. The trajectories recovered from the Kinect v2 data tend to have smaller translational RPE values through the whole sequence. For the data from both sensors, the RPE peaks occasionally to values a magnitude larger than the average. These peaks coincide in time with the more sharp turns of the ground truth trajectories. Apparently, when the robot makes a turn, the amount of point features that can be tracked over several frames decreases, contributing to a larger error in the computed transformation between the key frames. However, the RPE peaks for the Kinect v1 and Kinect v2 data rarely appear in exactly the same moments along the trajectories, which suggests that the point features found by the VO system in the images from both sensors are considerably different.

The ATE results, plotted in Fig. 13 for two example sequences, confirm our observations as to the better accuracy of the trajectories recovered from Kinect v2 data. The drift is apparent in all the tested sequences, but it is much more pronounced in the trajectories obtained using Kinect v1 data (Fig. 13a, c).

The differences in the accuracy of the recovered trajectories are clearly demonstrated by the statistical data in Fig. 14. For all four sequences, the trajectories estimated using Kinect v2 data have smaller RPE RMSE, both translational and rotational, as well as smaller ATE RMSE values. In particular, differences in the rotational relative errors are significant (Fig. 13b). This suggests that the wider horizontal field of view in the Kinect v2 sensor enables to obtain more features that are common between the neighboring frames when the robot is turning, whereas in the Kinect v1 data the amount of common point features between the frames considered in the VO is often very small during sharp turns.

Whereas in the navigation task we lack such a precise ground truth, as used for example in [14], to obtain statistics for depth data on controlled scenes, careful inspection

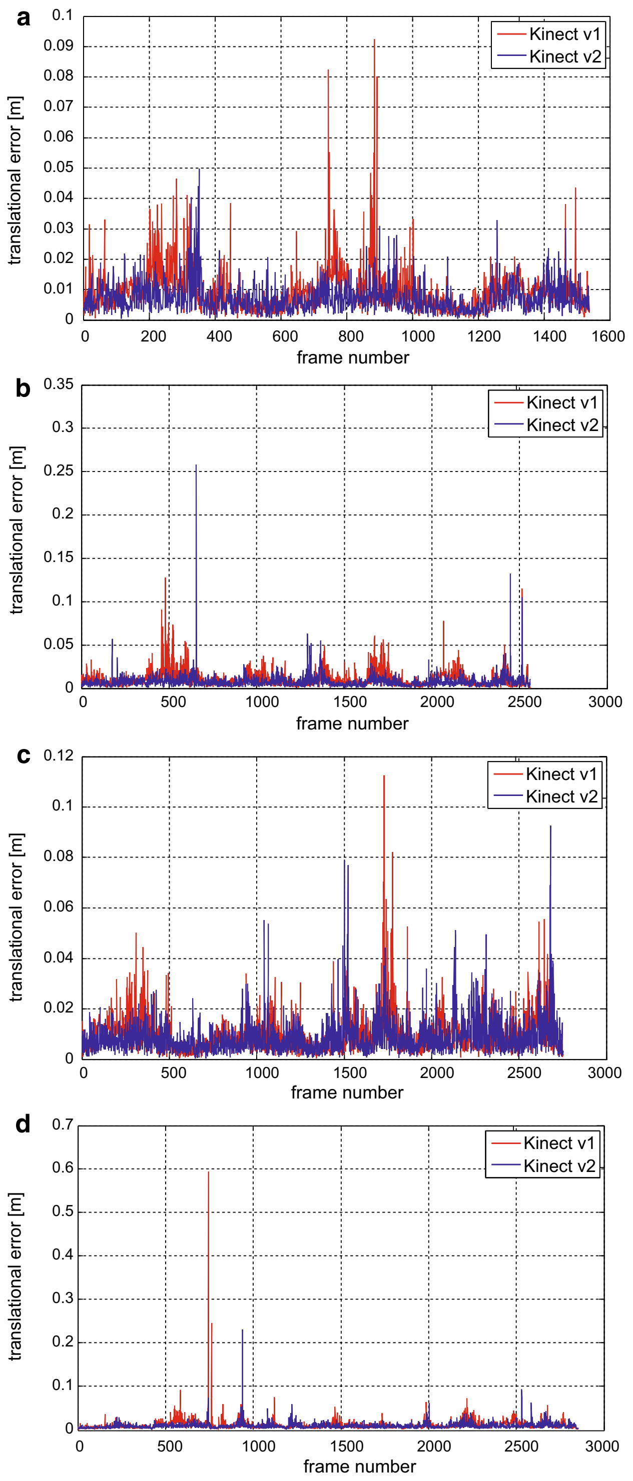

Fig. 12 Comparison of the trajectory drift for the Kinect v1 and Kinect v2 data with the positional RPE metric on four different sequences: no. 1 (**a**), no. 2 (**b**), no. 3 (**c**) and no. 4 (**d**)

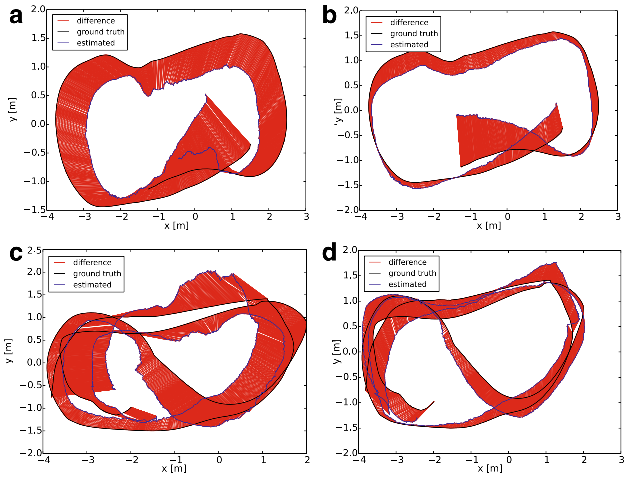

Fig. 13 Estimated and ground truth trajectories with the absolute trajectory errors (ATE) for the sequences no. 1 (**a**, **b**) and no. 2 (**c**, **d**), using the Kinect v1 (**a**, **c**) and Kinect v2 (**b**, **d**) data

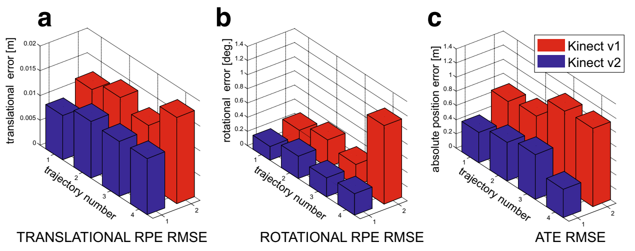

Fig. 14 Comparison of the trajectory estimation accuracy for the first four sequences simulating the indoor exploration scenario: translational RPE (**a**), rotational RPE (**b**), and ATE (**c**)

of the recorded frames reveals that there are important differences in the quality of the depth data from both sensors. Examples are shown in Fig. 15. The Kinect v1 depth frame (Fig. 15b) contains many areas of missing range information (shown in black), which results from the nature of the pattern of "speckles"—points projected by the IR emitter of the sensor on the scene, and the vulnerability to excessive ambient light [18]. The Kinect v2 depth frame of the same scene (Fig. 15d) has much fewer areas of missing data; particularly, there are no missing data neighboring to edges of objects, which is the case for Kinect v1. The missing depth data in Kinect v2 are rather isolated (Fig. 15h), whereas in Kinect v1 the missing range measurements form relatively large areas (Fig. 15f), preventing the VO from finding valid 3D features in a bigger part of the frame. These differences are allegedly related to the fact that Kinect v1 uses correlation to compare the observed pattern to some reference pattern, which creates dependencies in-between pixels on the depth image, while in Kinect v2 the depth is measured independently for each pixel [14]. The quality of the RGB frames is similar for both sensors, but the Kinect v2 has higher resolution of the RGB camera and a larger horizontal field of view, and thus, an image taken from the same position as for Kinect v1 captures a slightly larger part of the scene.

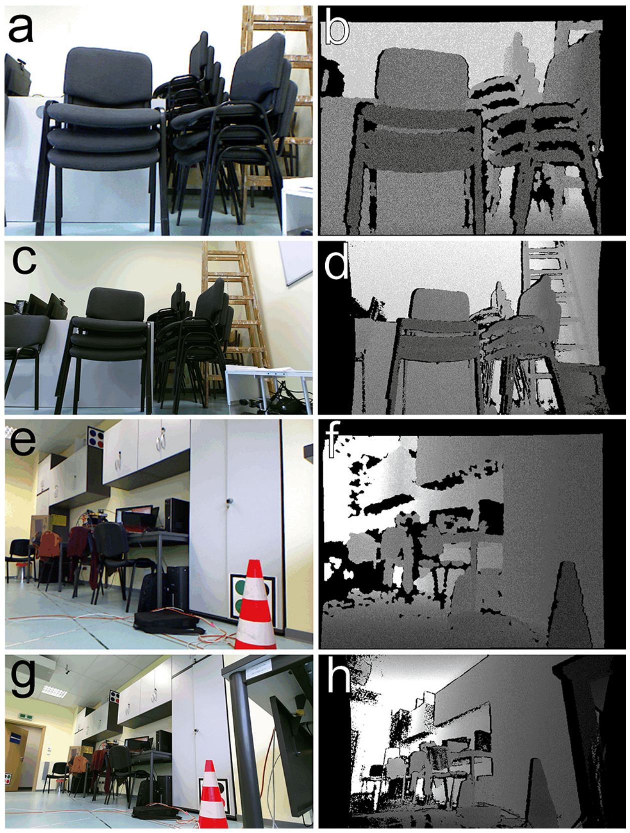

Fig. 15 Example RGB and depth frames from the trajectory no. 1 illustrating differences between the Kinect v1 (**a**, **b**, **e**, **f**) and Kinect v2 (**c**, **d**, **g**, **h**)

Table 5 ATE RMSE and RPE RMSE for the RGB-D VO system measured on all sequences

data sequence description	ATE [m]	ATE std dev. [m]	trans. RPE [m]	rot. RPE [°]
Kinect v1, $traj_1$	0.706	0.224	0.011	0.26
Kinect v2, $traj_1$	0.424	0.227	0.008	0.19
Kinect v1, $traj_2$	0.759	0.294	0.013	0.38
Kinect v2, $traj_2$	0.544	0.247	0.011	0.31
Kinect v1, $traj_3$	1.087	0.520	0.012	0.29
Kinect v2, $traj_3$	0.632	0.327	0.011	0.26
Kinect v1, $traj_4$	1.101	0.448	0.017	1.10
Kinect v2, $traj_4$	0.401	0.155	0.011	0.30
Kinect v1, $traj_5$	1.087	0.520	0.012	0.29
Kinect v2, $traj_5$	0.632	0.327	0.011	0.20
Kinect v1, $traj_6$	0.173	0.084	0.026	0.35
Kinect v2, $traj_6$	0.072	0.032	0.020	0.57
Kinect v1, $traj_7$	0.107	0.047	0.042	0.89
Kinect v2, $traj_7$	0.048	0.020	0.033	0.61
Kinect v1, $traj_8$	0.066	0.035	0.045	1.12
Kinect v2, $traj_8$	0.041	0.016	0.043	0.86

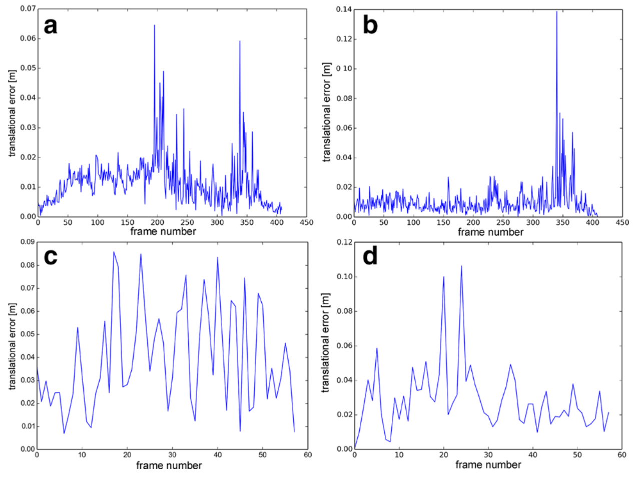

Fig. 16 Positional RPE showing the difference in drift on the slowest motion sequence no. 5 (**a**, **b**), and the fastest motion sequence no. 8 (**c**, **d**) for the Kinect v1 (**a**, **c**) and Kinect v2 (**b**, **d**) data

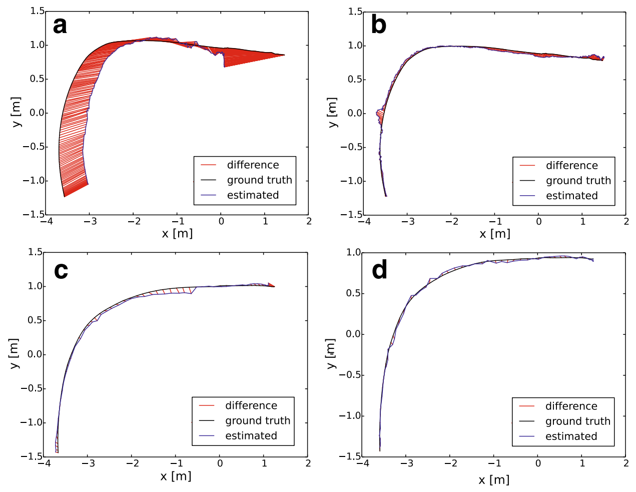

Fig. 17 ATE plots computed on the sequences no. 5 (**a**, **b**) and no. 8 (**c**, **d**), for the Kinect v1 (**a**, **c**) and Kinect v2 (**b**, **d**) data

The trajectories recovered by our VO system from the remaining four trajectories, when the robot moved with different speeds along approximately the same path demonstrated that both sensors are robust w.r.t. the increased motion speed, showing no significant motion blur in the RGB and depth data. Example translational RPE results are shown in Fig. 16. The translational relative errors have similar values for both sensors and different speeds. For the Kinect v2 data, the peak errors are slightly larger. This is attributed to the increased possibility of motion blur in the higher-resolution RGB images from the Kinect v2 sensor. However, the averaged translational and rotational RPE is still slightly higher for the Kinect v1 data for all the motion speeds, as shown in Table 5, which summarizes results for all eight sequences. The ATE results depicted in Fig. 17 show that the absolute trajectory errors are smaller for the sequence with larger speed.

Since the localization error in VO is cumulative, integration of individual frame-to-frame transformation errors across a larger number of frames results in a larger overall error. The trajectories 5–8 have similar length, so the trajectories with lower velocity are composed of more frames and the accumulated error is therefore larger.

The Kinect v2 has higher resolution of the images, and the depth data yielded by this sensor usually contain fewer areas of missing depth than the data obtained from Kinect v1. This results in a higher number of useful point features that are found in the RGB data and have associated valid depth values. Therefore, the number of correct matches (i.e., RANSAC inliers) between the sets of features in the neighboring key frames is usually higher for the Kinect v2 frames. The importance of the increased number of point features is particularly visible when the speed of the sensor increases (trajectories 5–8). When the sensor moves faster, distances between the key frames are larger, and thus, it is more difficult to track the key points using the Lucas–Kanade algorithm. More features detected at each key frame helps to ensure that the number of key points that survive tracking till the next key frame will be large enough to compute a correct transformation between the key frames.

7 Conclusions

The article introduces the PUTK^2 RGB-D data set for the evaluation of robot navigation algorithms. The RGB-D data were registered by the Kinect v1 and Kinect v2 sensors moving along eight different trajectories in an indoor environment. The sensor data consist of high-quality, uncompressed RGB and depth images and are supplemented with high-quality ground truth. The ground truth data were generated using a multi-camera system for sensor pose registration, with additional emphasis on careful synchronization during the acquisition process. As a result, the data consist of millimeter-accurate position data and facilitate full pose recovery. To the best knowledge of the authors, the presented data set is the first one to include the data registered by both Kinect sensor types in a side by side setup with such a reliable ground truth. The registration environment was specially arranged to ensure that physical objects are always visible in the working range of the Kinect sensors. Possible applications of the data set include vision- and depth-based navigation evaluation, 3D reconstruction and sensor characteristics comparison. Its usefulness is demonstrated in the paper with an example VO algorithm evaluation.

The evaluation of our RGB-D VO algorithm on the new data set allowed us to compare both versions of the Kinect sensor in the context of indoor navigation. As the tested VO system is feature based, the most important differences between the Kinect v1 and Kinect v2 concern the number of features that can be extracted from a frame and tracked reliably over several frames. The number of features is higher in the Kinect v2 due to the higher resolution of the RGB and depth images, and the smaller number of missing data areas in the depth frames of Kinect v2. Therefore, the Kinect v2 outperformed its predecessor on all the tested sequences with respect to both the relative pose errors and the drift of the whole trajectory. Moreover, the wider horizontal field of view of the Kinect v2 contributes to smaller relative pose errors when the sensor makes turns. The much higher resolution of RGB images makes the processing of Kinect v2 RGB-D data more computation demanding. However, the similar average features detection time per frame achieved by our VO on both the Kinect v1 and Kinect v2 data suggests that the point features are tracked more reliably, over a higher number of RGB images in the Kinect v2 data, as the Lucas–Kanade tracker benefits from the higher resolution of images.

The presented PUTK^2 data set is publicly available and can be freely downloaded and distributed, providing common ground for research and evaluation of both contemporary and future visual navigation algorithms.

References

1. Amon, C., Fuhrmann, F., Graf, F.: Evaluation of the spatial resolution accuracy of the face tracking system for Kinect for Windows v1 and v2. In: Proceedings of the 6th Congress of the Alps Adria Acoustics Association (2014)
2. Baker, S., Matthews, I.: Lucas–Kanade 20 years on: a unifying framework. Int. J. Comput. Vis. **56**(3), 221–255 (2004)
3. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: Speeded-up robust features (SURF). Compu. Vis. Image Underst. **110**(3), 346–359 (2008)
4. Belter, D., Nowicki, M., Skrzypczyński, P.: On the performance of pose-based RGB-D visual navigation systems. In: Cremers, D., Reid, I., Saito, H., Yang M.H. (eds.) Computer Vision—-ACCV 2014, Lecture Notes in Computer Science, vol. 9004, pp. 407–423. Springer International Publishing (2015)
5. Ceriani, S., Fontana, G., Giusti, A., Marzorati, D., Matteucci, M., Migliore, D., Rizzi, D., Sorrenti, D.G., Taddei, P.: Rawseeds ground truth collection systems for indoor self-localization and mapping. Auton. Robots **27**(4), 353–371 (2009)
6. Claus, D., Fitzgibbon, A.W.: A rational function lens distortion model for general cameras. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005. CVPR 2005, vol. 1, pp. 213–219. IEEE (2005)
7. Davison, A., Reid, I., Molton, N., Stasse, O.: MonoSLAM: real-time single camera SLAM. IEEE Trans. Pattern Anal. Mach. Intell. **29**(6), 1052–1067 (2007)

8. Durrant-Whyte, H., Bailey, T.: Simultaneous localization and mapping: part I. IEEE Robot. Autom. Mag. **13**(2), 99–110 (2006)
9. Endres, F., Hess, J., Sturm, J., Cremers, D., Burgard, W.: 3-D mapping with an RGB-D camera. IEEE Trans. Robot. **30**(1), 177–187 (2014)
10. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining, pp. 226–231 (1996)
11. Fallon, M., Johannsson, H., Kaess, M., Leonard, J.J.: The MIT Stata Center dataset. Int. J. Robot. Res. **32**(14), 1695–1699 (2013)
12. Fankhauser, P., Bloesch, M., Rodriguez, D., , Kaestner, R., Hutter, M., Siegwart, R.: Kinect v2 for mobile robot navigation: evaluation and modeling. In: IEEE International Conference on Advanced Robotics (ICAR) (2015)
13. Fraundorfer, F., Scaramuzza, D.: Visual odometry: part II: matching, robustness, optimization, and applications. IEEE Robot. Autom. Mag. **19**(2), 78–90 (2012)
14. Gesto Diaz, M., Tombari, F., Rodriguez-Gonzalvez, P., Gonzalez-Aguilera, D.: Analysis and evaluation between the first and the second generation of rgb-d sensors. IEEE Sens. J. **15**(11), 6507–6516 (2011)
15. Grisetti, G., Kümmerle, R., Stachniss, C., Burgard, W.: A tutorial on graph-based SLAM. IEEE Intell. Transp. Syst. Mag. **2**(4), 31–43 (2010)
16. Handa, A., Whelan, T., McDonald, J., Davison, A.J.: A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM. In: 2014 IEEE International Conference on Robotics and Automation (ICRA), pp. 1524–1531. IEEE (2014)
17. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, Cambridge (2003)
18. Khoshelham, K., Elberink, S.O.: Accuracy and resolution of Kinect depth data for indoor mapping applications. Sensors **12**(2), 1437–1454 (2012)
19. Lachat, E., Macher, H., Mittet, M.A., Landes, T., Grussenmeyer, P.: First experiences with Kinect v2 sensor for close range 3D modelling. ISPRS Int. Arch. Photogram. Remote Sens. Spat. Inf. Sci. XL-5/W4 93–100 (2015) doi:10.5194/isprsarchives-XL-5-W4-93-2015
20. Maykol Pinto, A., Costa, P., Moreira, A., Rocha, L., Veiga, G., Moreira, E.: Evaluation of depth sensors for robotic applications. In: 2015 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC), pp. 139–143 (2015)
21. Molloy, J.C.: The open knowledge foundation: open data means better science. PLoS Biol. **9**(12), e1001,195 (2011)
22. Nowicki, M., Skrzypczyński, P.: Combining photometric and depth data for lightweight and robust visual odometry. In: 2013 European Conference on Mobile Robots (ECMR), pp. 125–130. IEEE (2013)
23. Optitrack—Motion capture systems. https://www.optitrack.com/
24. Raguram, R., Chum, O., Pollefeys, M., Matas, J., Frahm, J.: USAC: a universal framework for random sample consensus. IEEE Trans. Pattern Anal. Mach. Intell. **35**(8), 2022–2038 (2013)
25. Richtsfeld, A., Morwald, T., Prankl, J., Zillich, M., Vincze, M.: Segmentation of unknown objects in indoor environments. In: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4791–4796 (2012)
26. Rosten, E., Drummond, T.: Machine learning for high-speed corner detection. In: Computer Vision–ECCV 2006, pp. 430–443. Springer (2006)
27. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: ORB: an efficient alternative to SIFT or SURF. In: 2011 IEEE International Conference on Computer Vision (ICCV), pp. 2564–2571. IEEE (2011)
28. Scaramuzza, D., Fraundorfer, F.: Visual odometry [tutorial]. IEEE Robot. Autom. Mag. **18**(4), 80–92 (2011)
29. Schmidt, A., Kasiński, A., Kraft, M., Fularz, M., Domagała, Z.: Calibration of the multi-camera registration system for visual navigation benchmarking. Int. J. Adv. Robot. Syst. **11**, 83 (2014)
30. Singh, A., Sha, J., Narayan, K., Achim, T., Abbeel, P.: BigBIRD: A large-scale 3D database of object instances. In: 2014 IEEE International Conference on Robotics and Automation (ICRA), pp. 509–516 (2014)
31. Smith, M., Baldwin, I., Churchill, W., Paul, R., Newman, P.: The New College vision and laser data set. Int. J. Robot. Res. **28**(5), 595–599 (2009)
32. Sturm, J., Engelhard, N., Endres, F., Burgard, W., Cremers, D.: A benchmark for the evaluation of RGB-D SLAM systems. In: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 573–580. IEEE (2012)
33. Sturm, J., Magnenat, S., Engelhard, N., Pomerleau, F., Colas, F., Cremers, D., Siegwart, R., Burgard, W.: Towards a benchmark for RGB-D SLAM evaluation. In: RGB-D Workshop on Advanced Reasoning with Depth Cameras at Robotics: Science and Systems Conference (RSS) (2011)
34. Triggs, B., McLauchlan, P.F., Hartley, R.I., Fitzgibbon, A.W.: Bundle adjustment—a modern synthesis. In: Triggs, B., Zisserman, A., Szeliski, R. (eds.) Vision Algorithms: Theory and Practice, pp. 298–372. Springer-Verlag, Berlin, Heidelberg (2000)
35. Umeyama, S.: Least-squares estimation of transformation parameters between two point patterns. IEEE Trans. Pattern Anal. Mach. Intell. **13**(4), 376–380 (1991)
36. Motion capture systems vicon. http://www.vicon.com/
37. Williams, B., Reid, I.: On combining visual SLAM and visual odometry. In: 2010 IEEE International Conference on Robotics and Automation (ICRA), pp. 3494–3500. IEEE (2010)
38. Xiao, J., Owens, A., Torralba, A.: SUN3D: a database of big spaces reconstructed using SfM and object labels. In: 2013 IEEE International Conference on Computer Vision (ICCV), pp. 1625–1632 (2013)
39. Zennaro, S., Munaro, M., Milani, S., Zanuttigh, P., Bernardi, A., Ghidoni, S., Menegatti, E.: Performance evaluation of the 1st and 2nd generation Kinect for multimedia applications. In: 2015 IEEE International Conference on Multimedia and Expo (ICME), pp. 1–6 (2015)

Marek Kraft graduated from Poznan University of Technology in 2005. He received the PhD from the same university in 2013. In the same year, he has been appointed as an assistant professor at the university's Institute of Control and Information Engineering. His research interests include computer vision, embedded systems, robotics, parallel processing and high-performance computing.

Michał Nowicki graduated from Poznan University of Technology in 2014 receiving BSc and MSc in Automatic Control and Robotics from the Poznan University of Technology, Poznan, in 2013 and 2014, respectively, and BSc in Computer Science from the same university in 2014. Currently, he is a research assistant and PhD student at the Institute of Control and Information Engineering. His research interests include computer vision, simultaneous localization and mapping, walking robots, and Android-based navigation.

Adam Schmidt received his MSc and PhD in Automatic Control and Robotics from Poznan University of Technology in 2007 and 2014, respectively. In 2014, he was appointed an assistant professor at the Institute of Control and Information Engineering of the Poznan University of Technology. His research interests include computer vision, simultaneous localization and mapping and machine learning.

Michał Fularz received his MSc degree in Control Engineering and Robotics in 2010 from Poznan University of Technology. Now he is a researcher pursuing the PhD in PUT vision team (www.vision.put.poznan.pl) at the same university. Michał's area of expertise is hardware acceleration of image processing algorithms for FPGAs and embedded systems. His current work also includes development of smart camera designed for large-scale surveillance systems.

Piotr Skrzypczyński graduated from the Poznan University of Technology (PUT) in 1993. He received PhD and DSc degrees in Robotics from the same university in 1997 and 2007, respectively. Since 2010, he is an associate professor at the Institute of Control and Information Engineering (ICIE) of PUT, and since 2012 head of the Automation and Robotics Division at ICIE. He leads the Mobile Robotics Laboratory at ICIE. Prof. Skrzypczyński is author or co-author of more than 150 technical papers in the areas of robotics and computer science. His current research interests include autonomous mobile robots, simultaneous localization and mapping, multisensor fusion, and computational intelligence in robotics.