Sirazitdinova *et al. EURASIP Journal on Image and Video Processing* (2016) 2016:43
DOI 10.1186/s13640-016-0146-1

EURASIP Journal on Image and Video Processing

RESEARCH Open Access

Towards efficient mobile image-guided navigation through removal of outliers

Ekaterina Sirazitdinova[1]*, Stephan M. Jonas[1], Jan Lensen[2], Deyvid Kochanov[1], Richard Houben[2], Hans Slijp[2] and Thomas M. Deserno[1]

Abstract

A novel approach for positioning using smartphones and image processing techniques is developed. Using structure from motion, 3D reconstructions of given tracks are created and stored as sparse point clouds. Query images are matched later to these 3D models. High computational costs of image matching and limited storage require compressing point clouds without loss of positioning performance. In this work, localization is improved and memory and storage requirements are minimized. We assumed that the computational speed and, at the same time, storage requirements benefit from reducing the number of points with appropriate outlier detection. In particular, our hypothesis was that positioning accuracy is maintained while reducing outliers in a reconstructed model. To evaluate the hypothesis, three methods were compared: (i) density-based (Sotoodeh, International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences XXXVI-5, 2006), (ii) connectivity-based (Wang et al. Comput Graph Forum 32(5):207–10, 2013), and (iii) our distance-based approach. In tenfold cross-validation, applied to a pre-reconstructed reference 3D model, localization accuracy was measured. In each new model, the positions of test images were identified and compared to the according positions in the reference model. We observed that outlier removal has a positive impact on matching run-time and storage requirements, while there are no significant differences in the localization error within the methods. That confirmed our initial hypothesis and allows mobile application of image-based positioning.

Keywords: Image-based localization, Mobile navigation, Structure from motion, 3D point clouds, Outlier removal

1 Introduction

Due to the rapid growth of technologies, pedestrian navigation has become widely accessible in the recent years. In the developed countries, smartphones are no longer considered as luxury items and are owned by the majority of population. The sensors installed in modern mobile devices, such as global positioning system (GPS) receiver, accelerometer, compass, gyroscope, and camera, provide a broad field of methods that can be applied for mobile navigation.

Satellite-based GPS is widely used in various navigational devices and applications. Being available on most modern smartphones, with a help of additional context information (e.g., map-based graphical representation of a city area), GPS can provide assistance in navigation. However, in large cities, where tall buildings block or reflect satellite signals, the positioning error of GPS is measured in meters [1]. Such an error does not allow using GPS to navigate blind or visually impaired people, since stepping off the sidewalk to the car lanes is dangerous and can be harmful. Even using differential correction[1], GPS alone is not sufficiently accurate to guide pedestrians in urban environments, because there are no distinct roads but narrow to broad paths to walk on.

Some other approaches use the number of steps detected by an accelerometer, reference points, and a mobile compass for navigation assistance. Fallah et al. [2] presented a successful example of this method. However, their system is designed for indoor environments, where maps are very accurate and clear landmarks (e.g., corners and doors) are available.

*Correspondence: ekaterina.sirazitdinova@rwth-aachen.de
[1]Department of Medical Informatics, Uniklinik RWTH Aachen, Pauwelsstr. 30, 52057 Aachen, Germany
Full list of author information is available at the end of the article

More recently, radio frequency identification (RFID) technology has found its use in the research area of navigation. One of the latest systems applied to navigation of visually impaired was proposed by Varpe and Wankhade [3]. On the user side, they have applied a mobile RFID reader, a transceiver for transmitting the tag's information, and an audio device to provide feedback to the user. To identify walking routes, RFID passive tag network has been employed on the path. Although the accuracy of such systems yields a precision of 1-m scale when a dense RFID tag configuration is used, it requires additional objects (i.e., RFID tags), which makes this technology costly and not easily adoptable for new environments [4].

An alternative method of user navigation with the help of smartphones is currently being developed [5]. This method makes use of image processing for navigation. This technique has been utilized mostly in robotics [6], but there are some other adaptations of it for indoor and outdoor pedestrian navigation as well [7–10]. Firstly, given tracks are reconstructed as sparse 3D point clouds using structure from motion (SfM) [11] and stored in a database. Secondly, with an interactive app running on a smartphone, query images are acquired in order to retrieve the location and direction of the camera (i.e., the pedestrian), which is a required component for navigation.

Scale invariant feature transform (SIFT) [12] features are extracted from the images taken with the client application. The features are reconstructed in the form of points in the 3D cloud. Comparing thousands of points from the model with the current query photograph is computationally expensive. Together with storage limitation, this causes the necessity of removing outliers from 3D data without affecting the positioning accuracy.

In this paper, we analyze outlier removal in generated 3D point clouds for pedestrian navigation. Our hypothesis states that it is possible to maintain positioning accuracy while reducing the number of outliers in a reconstructed 3D model. These developments are part of the smartphone-based system designed for navigation of visually impaired people [5].

2 State of the art

According to the definition of Grubbs [13], an outlying observation, or outlier, is "one that appears to deviate markedly from other members of the sample in which it occurs." Outliers in a 3D point cloud may be of different nature. Firstly, they may result from errors occurring during the reconstruction process, such as inherent inaccuracies in feature detection, false matching, and errors in estimation of fundamental and projection matrices. Second, non-static environment objects (e.g., cars, chairs and tables of street cafes, advertising and market stalls) add noise to the reconstruction.

In SfM 3D point clouds, outlier removal is possible in two stages. First, within the bundle adjustment, erroneous matches are usually discarded by the random sample consensus (RANSAC) [14] or its extensions, progressive sample consensus (PROSAC) [15], and preemptive RANSAC [16]. In order to do a robust estimation of parameters in terms of reconstruction, the following steps are repeated iteratively: (i) a seed group of matches is randomly selected; (ii) transformation from the seed group is computed; (iii) inliers to this transformation are found; (iv) if the number of inliers is sufficiently large, the least-squares estimate of the transformation on all of the inliers is recomputed. The transformation with the largest number of inliers is kept. With a sufficient number of inliers (more than 50%) and correctly chosen parameters, this method gives a good estimation of matches.

Nonetheless, the overall outcome model sometimes is not "clean" due to the inherent inaccuracies in feature detection, false matching, and errors in estimation of fundamental and projection matrices. This leads to the necessity of performing an additional step of outlier detection in the reconstructed 3D point cloud. In most vision-based city reconstruction approaches, outliers are removed only within the reconstruction process, and no "cleaning" techniques are applied to the generated point clouds [17–20]. That is explained by the visualization purpose of their reconstruction.

Taglioretti et al. [21] evaluated the performance of localization depending on the selected outlier removal method during the bundle adjustment. The forward search method [22] proved to be superior. However, the problem of additional outlier removal in SfM 3D point clouds has not been evaluated from the perspective of localization task before. In order to identify possible applicable techniques, we observed existing outlier detection approaches.

Based on Hodge and Austin [23], outlier detection approaches are categorized as

- distribution-based,
- depth-based,
- clustering-based,
- distance-based,
- density-based, and
- connectivity-based.

In *distribution-based* methods, the bulk of observations is estimated robustly by a suitable model distribution. Outliers are then defined as observations, which are unlikely to be generated by the distribution [24].

In *depth-based* approaches, data objects are organized in layers in the data space, with the expectation that shallow layers are more likely to contain outlying data objects than the deep layers [25].

In *clustering-based* techniques, a cluster of small size can be considered as clustered outliers [26].

In the approach by Knorr and Ng [27], an object in a dataset is a *distance-based* outlier if at least a given fraction of the other objects in the dataset lies at a distance greater than some given threshold. This approach does not make any assumptions about the data distribution and has better computational efficiency than depth-based methods, especially in large datasets.

In *density-based* methods, the relative density of a point compared to its neighbors is computed as outlier score. Using this approach, one can effectively identify local outliers in datasets with diverse clusters [28]. Breunig et al. [29] proposed a density-based approach relying on the local outlier factor (LOF) of each object, which is depending on the local density of its neighborhood. The neighborhood is defined by the distance to the $M(p)$th nearest neighbor. The value $M(p)$ is predefined. It corresponds to the minimum number of points used in the calculation of density.

Approaches from classification [23] sometimes can be combined into more complex methods. Thus, a mixture of density- and clustering-based approaches, in this paper, is referred to as *connectivity-based* approach.

Outlier removal outside the bundle adjustment in completely built SfM point clouds has not been addressed explicitly before. However, there are some approaches designed for laser-scanned point clouds. Such clouds are usually more accurate and consist of a higher number of points. We believe, nevertheless, that the principles of outlier removal in laser-scanned point clouds also work for SfM point clouds and, therefore, review here some approaches designed for laser-scanned point clouds.

In 2006, Sotoodeh [30] presented a LOF-based algorithm for outlier detection in laser-scanned point clouds. The author justifies the selection of a density-based algorithm due to its unconstrained behavior to the preliminary knowledge of the scanned scene and its independence from the varying density of the points. The method was able to detect most of the expected outliers in the scene; however, it was not robust against clusters of outliers. For that reason, in 2007, the author proposed a modified version of his algorithm based on hierarchical clustering [31]. The modified algorithm runs in two phases: in the first stage, it removes relatively large-scale erroneous measurements based on Euclidean minimum spanning tree edges. In the second phase, it detects and removes the outliers that might not be as obvious as the first ones but according to the scanned object surfaces, they are considered as wrong measurements. The algorithm was tested on terrestrial point clouds and returned a satisfying result: both, single and clustered outliers were removed. However, in some cases, user interaction was still required to determine whether a cluster is an outlier or an object. An additional drawback is a run-time complexity of $O(n^3)$, which makes the method inefficient for working with datasets containing thousands of points.

Luo and Liao [32] proposed outlier detection in laser point clouds extending distance- and density-based approaches. Their algorithm changes 3D data to 2D by slicing and projection and employs a KD tree to index the projected points. The authors use the local distance-based outlier factor (LDBOF) defined by Zhang et al. [33] as the outlier judgment criterion. LDBOF uses the relative location of an object to its neighbors to determine the degree to which the object deviates from its neighborhood. The authors claimed higher efficiency compared to the algorithms of Sotoodeh [30, 31]. However, they also mention the necessity of finding more robust parameters [32].

Recently, Wang et al. [34] designed a connectivity-based pipeline for outlier filtering and noise smoothing in low-quality point clouds from outdoor scenes. They first detect sparse outliers applying a scheme based on the relative density deviation of the local neighborhood and the average local neighborhood, providing a scoring strategy that includes a normalization to become independent from the specific data distribution. In order to remove further small dense outliers, a clustering method is used. According to the authors, detection is capable of removing all types of outliers without any user interactions.

3 Outlier removal applied to 3D point clouds

City-scale 3D point clouds are large, arbitrary datasets, and, therefore, the methods claiming computational efficiency were preferred over others. Another important criterion for selection of an outlier removal method was the ability of a method to be performed without any additional user interaction. Thus, the first approach of Sotoodeh [30] and the pipeline of Wang et al. [34] were implemented and applied to our datasets with some parameter adjustments.

While the density-based method runs in a linear time, the second part of the connectivity-based approach, performed by agglomerative hierarchical clustering, has the run-time complexity of $O(n^3)$. To assess the potential of computational speedup, an original distance-based method of outlier detection in 3D point clouds is proposed.

3.1 The novel distance-based approach

We adopt the notion of distance-based outliers proposed by Knorr and Ng [27] for data-mining applications: "An object in a dataset is an outlier if at least a fraction of the objects in this dataset lies in a larger distance from this object." Our approach is based on the assumption that points belonging to building wall structures have normal distribution. Thus, we apply a double-threshold scheme: firstly, we reduce the impact of infrequent points in the model, the relative distances from which to the other points in the model are comparatively large. After eliminating such points, we estimate the second filtering factor based on the global mean over mean distances of each point's neighborhood.

Given a point set $P = \{p_1, \ldots, p_n\}$ (n is the number of points), outlier elimination is performed as follows:

1. At the beginning, for each point p_i, $(i = 1, \ldots, n)$, the k-nearest neighbors $N(p_i) = \{q_1, \ldots, q_k\} \subseteq P$ are determined. The value of $k = 32$ was selected for our approach by visual inspection as described in the following subsection.
   The function returns the set of indexes of a point's k-nearest neighbors and their distances to the point.
2. For each point p_i, the so called k-distance, denoted as $D_k(p_i)$, is defined as the distance $d(p_i, p_j)$, where $p_j \in N(p_i)$ is the neighbor farthest away in p_i's k-neighborhood—in other words, the longest distance among the distances from p_i to its k-nearest neighbors.
3. Then, for each point p_i, the average distance of its neighborhood $\overline{d}(p_i)$ is calculated as

$$\overline{d}(p_i) = \frac{\sum\limits_{q_j \in N(p_i)} d(p_i, q_j)}{k} \tag{1}$$

   Then, the standard deviation of the neighborhood distances is estimated as

$$\sigma = \sqrt{\frac{\sum\limits_{i=1}^{n} (\overline{d}(p_i) - \overline{D})^2}{n}} \tag{2}$$

   where $\overline{D}$ is the mean value of all $\overline{d}(p_i)$.
4. Subsequently, the point cloud is filtered so that all points that meet the condition $\overline{d}(p_i) \geq 10\sigma$ are eliminated. Having the point cloud filtered initially, the average distance $\overline{D}$ is recalculated with respect to the points left in the model. Then, the refined value $\overline{D}$ together with $D_k(p_i)$ is used for the final filtering phase: the points, for which the condition $D_k(p_i) \geq 3\overline{D}$ holds, are removed from the model. The remaining points are considered inliers.

All parameters were empirically derived, considering a set of constraints described further.

3.2 Constraints for parameters used in the proposed method

The first qualitative characteristic of outlier removal is a level of noise preserved in the model afterwards. The noise level stands for the relative number of points or point clusters remaining in the model after outlier detection, although they should have been removed. This characteristic is particularly important for aligning models and maps, which is a part of our image-based navigation system merging separate model fragments in the same coordinate space. A high level of noise can affect the alignment, as the outlying points can drag a model towards the wrong walls.

Removing as many outliers as possible, the main constraint for parameters adjustment (e.g., number of nearest neighbors, filtering coefficients) was retainability of model's structure, or, in other words, presence of all significant walls in the model after outlier removal.

This constraint is important for navigation, because we are interested in covering large area. At the same time, the correctness of the model's alignment, again, highly depends on the footprint structure, so that, in some cases, even an additional small wall can resolve ambiguity of scaling parameters and thus the right model placement. Therefore, it is rather important to have the majority of walls preserved after the outlier removal step.

For each outlier removal method, we achieved a tradeoff between the level of noise and model's retainability by adjusting the parameters. The parameter adjustment was performed on point clouds of different density through iterative testing using different combinations of parameters: in each test case, we compared the number of point clusters outside the facade (due to the small model size, it was possible to count them manually) and evaluated the completeness of facades. For our models, the local optimum was achieved with the described set of parameters. However, it may happen that further parameter adjustment might be required for the models of different density.

4 Experimental setup

4.1 Dataset

Evaluation was performed on a dataset recorded at the downtown of Maastricht, the Netherlands. The dataset results from 7 walks with a recording device (iPhone 5 (Apple Inc., USA) with acquisition application running on it) attached with a chest mount utility to the body of the person acquiring images (Fig. 1). Within a walk, images were acquired sequentially every second. A total of 3291 images were recorded. All recordings differ in date, time, and weather condition.

The route passes by several landmarks in the center of Maastricht. The main characteristics of the location are a large number of pedestrians, high vehicle traffic, and narrow streets and houses located close to the road. Additionally, the route's appearance changes most during spring and summer, as street cafes are active and numerous shops and stores are constantly changing decorations in and around showroom windows.

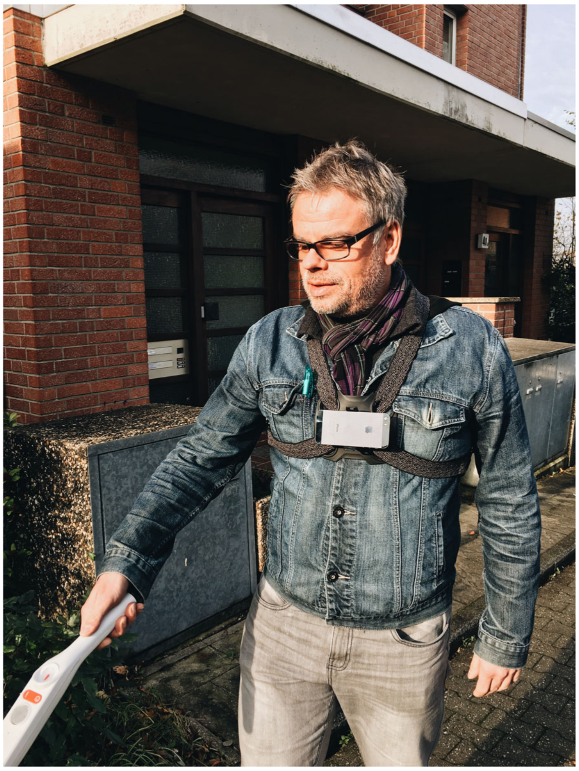

Fig. 1 Data acquisition and navigation. A smartphone is attached with a chest-mount to the user (*on the left*). For positioning, the user holds an interactive cane, connected to the system via Bluetooth interface and providing navigational clues in a form of a haptic feedback. Data is transferred to a computer using wireless network connection. Consent to use the photograph was obtained

Processing with VisualSFM [11] resulted in a dataset of 17 models. Each model represents a reconstructed set of building walls as a sparse 3D point cloud. The models contain from 200 to 12,792 points.

4.2 Preparation of test models

Inspired by the approaches of Strecha et al. [35] and Untzelmann et al. [36], we aligned all models to the OpenStreetMap [37]. To evaluate our initial hypothesis that positioning accuracy is maintained while reducing outliers in a reconstructed model, we selected from our dataset a reference model that allows the best automatic alignment to the real world coordinates (Fig. 2).

The selected model contains 11,650 points and 374 cameras. This model was then reconstructed again by tenfold cross-validation: all images used in the reference model were randomly partitioned into 10 sub-samples of equal size. For each new reconstruction, a newly selected single sub-sample containing 10% of original images was used as test data; the remaining 90% of images were used to reconstruct a model.

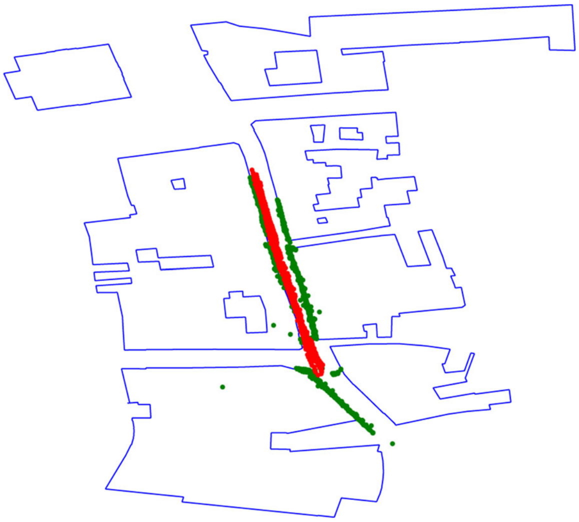

Fig. 2 Alignment of a model to the OpenStreetMap outline. *Green points* belong to wall structures; *red line* is a camera path

4.3 Testing process

To test the hypothesis, the following sequence of steps was applied to eight test reconstructions consisting of the largest amount of points:

1. Align each model to the map to estimate their scaling factors relatively to the real-world coordinate system.
2. Align the test reconstruction to the reference reconstruction. For that, we apply the estimated scaling parameters to the test and the reference models.
3. Estimate the translation between the models by calculating the difference between the models' centroids.
4. Refine translation and rotation by applying the iterative closest point (ICP) algorithm [38].
5. Estimate a position of each image not used for the reconstruction and record the matching time. To estimate the location of an image, SIFT features are extracted from it. Correspondences between the features and points in a 3D point cloud are determined. Since some of the found correspondences are matching outliers, the pose estimation procedure is wrapped in a RANSAC loop. RANSAC picks a random subset of matches and uses them to generate a hypothesis about the pose. It then tests the hypothesis against the full set. If the number of matches is large enough, RANSAC terminates returning the set of inliers and a pose estimated from them.
6. Use the corresponding positions of the reconstructed images from the reference model to estimate the localization error of each image. The error is

calculated as the distance between the estimated position and the reference position in 2D (as we localize the user in 2D, the z-component is omitted).

7. Apply the three outlier removal methods to the aligned test reconstruction. Repeat 5 and 6 with the resulting models.

To measure the matching time, we conducted the localization experiment 10 times on each of the test cases. All tests were computed on a single core of a PC equipped with the Intel Core i7 CPU running at 2.00 GHz.

4.4 Performance measures

Firstly, we observe the performance of outlier removal methods themselves according to the percent of points removed P_r by each method and in terms of time T_o required for a method to remove outliers.

Secondly, we evaluate the performance of localization process. For that, we distinguish between efficiency and quality indicators. Our goal is to achieve a trade-off between those two groups.

Efficiency indicators refer to performance in terms of processing time and memory requirements and estimate matching time T_m (in seconds) and model's size S_m (in KB) accordingly. In order to show the changes in performance caused by the application of a certain outlier removal method, we introduce the parameters for changes in matching time ΔT_{m0j} and space requirements ΔS_{m0j}, defined as

$$\Delta T_{m0j} = \frac{T_{m0} - T_{mj}}{T_{m0}} \times 100\% \tag{3}$$

$$\Delta S_{m0j} = \frac{S_{m0} - S_{mj}}{S_{m0}} \times 100\% \tag{4}$$

where $j = 1, \ldots, 4$ corresponds to a model in a test case. A *test case* contains four models: one model before outlier removal and three after different outlier removal methods applied.

Quality indicators describe localization performance associated with a certain model.

Let n be a total number of test images associated with a certain tested model. Given a test image contained in the reference model, an image is considered as *matched* if it is possible to reconstruct its position p in the tested model. Accordingly, n_m is the total number of matched images in the model. A match is considered as *correct* if the positioning error, estimated as a distance between a reconstructed position p and its corresponding position p_0 in the reference model, is less than a threshold τ

$$\left\| p_0 - p \right\| < \tau \tag{5}$$

We set $\tau = 1.6$ m (2–3 human steps).

The number of correct matches n_c is estimated as

$$n_c = \sum_{i=1}^{n_m} \left[\left\| p_{0i} - p_i \right\| < \tau \right] \tag{6}$$

Then, the matching rate R is calculated as the ratio of the number of correct matches n_c and the total number of images n

$$R = \frac{n_c}{n} \times 100\% \tag{7}$$

The matching error E is the average value of all positioning errors of the correct matches:

$$E = \frac{\sum_{i=1}^{n_m} \left\| p_{0i} - p_i \right\| \left(\left\| p_{0i} - p_i \right\| < \tau \right)}{n_c} \tag{8}$$

Based on these two indicators, we estimate weighted matching error E_w, which is used as an ultimate indicator for the quality of localization

$$E_w = wE \tag{9}$$

where w is a weighting coefficient of a certain model.

For each jth model in a test case, where $j = 1, \ldots, 4$, the coefficient w_j is calculated as follows

$$w_j = 1 - \frac{R_j - \min\{R_1, \ldots, R_4\}}{100\%} \tag{10}$$

In fact, the ICP alignment of a test model to the reference model might contain an error up to 1 m. Thus, the absolute values of localization measurements might not be precise. However, as we use always the same alignment, the positioning errors are estimated in the same coordinate system within a test case; hence, the correct estimate of relative errors is possible. As we are interested in comparing the quality of localization, our final quality indicator is

$$\Delta E_{w0j} = E_{w0} - E_{wj} \tag{11}$$

where E_{w0} is the weighted localization error associated with the reference model, and E_{wj} ($j = 1, \ldots, 3$) are the corresponding weighted errors in localization using the models after the outlier removal methods applied.

We run one-way analysis of variance (ANOVA) on the entire sample of positioning errors to see whether the changes in positioning performance are significant or not, depending on the outlier method applied.

5 Results

5.1 Outlier removal

According to visual inspection, each of the approaches is able to reduce noise while preserving the model structure (Fig. 3). Comparing to the original models containing sparse outliers, the outcomes of all outlier removal methods look clean. Some wall fragments containing relatively

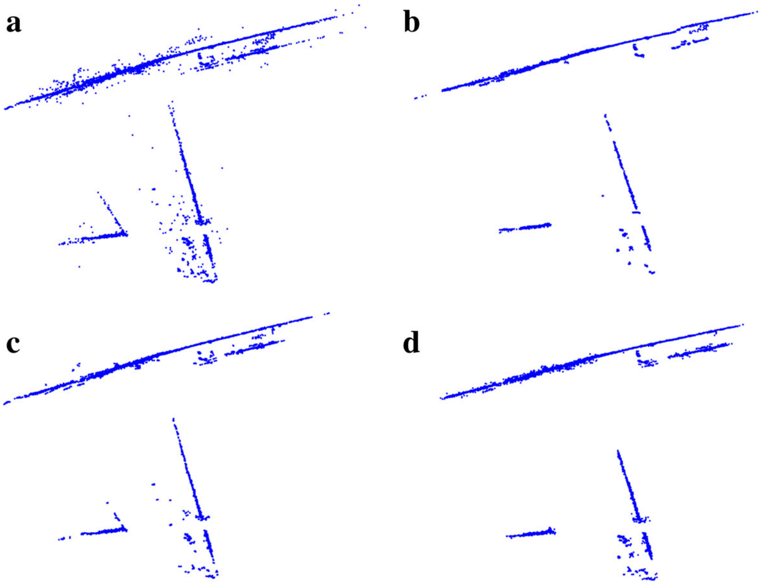

Fig. 3 Comparison of outlier removal effect on 3D model. **a** Original 3D model footprint and 3D footprints after **b** density-based, **c** connectivity-based, and **d** distance-based outlier removal methods applied. For a better visibility, the models are rotated upright according to the pre-computed model's gravity vector

fewer feature points than other parts might be missing; however, the basic structures are always preserved.

On average, the density-based method classified the biggest number of points (33.3% of the initial number) as outliers, while the smallest result was obtained by the distance-based method (10.2%) (Table 1).

Regarding the outlier removal time, on average, our distance-based approach (Fig. 4, blue) outperforms the density-based approach (Fig. 4, red) to around 45% for all models regardless of the number of points they contain. The computational time of connectivity-based approach (Fig. 4, green) grows in a polynomial way with increase of model's size. Hence, for a model consisting of about 10,000 points, outlier removal will take approximately 7 s.

5.2 Computation and storage requirements

The experiment has shown that in all cases, the reduction of outliers leads to the noticeable improvement in matching time T_m (Fig. 5, top-left panel) and has a positive impact on model's size S_m (Fig. 5, top-right panel), comparing to the performance associated with a model before outlier removal.

The benefits in matching time ΔT_{m0j} and storage requirements ΔS_{m0j} are proportional to the number of points P_r removed from the model (Table 1).

5.3 Quality of localization

For the extreme case (the density-based approach, removing 33.3% of points from the model), the probability to locate an image with a precision up to 1.6 m was 70%. Using this threshold, the absolute error values were below 0.56 m for all of the cases (Fig. 5, bottom-left panel).

The average localization error resulted as the lowest (0.51 m) for our outlier removal method (Fig. 5, blue bar on the bottom-left panel). At the same time, taking into account the matching rate, the relative weighted localization error tended to increase for the methods classifying a greater number of points as outliers compared to the reference model (Table 1). The ANOVA test with 3 degrees of freedom applied on the entire set of positioning

Table 1 Evaluation results. ΔT_{m0j}, ΔS_{m0j}, and ΔE_{w0j} are calculated with the Eqs. 3, 4, and 11, respectively

	Outliers removed P_r (%)	Benefit in computational time ΔT_{m0j} (%)	Benefit in storage requirements ΔS_{m0j} (%)	Loss in the accuracy of localization ΔE_{w0j} (cm)
Density-based	*33.3*	*31.1*	*28.97*	8
Connectivity-based	20.4	17.6	19.24	4
Distance-based	10.2	8.8	10.1	*1*
Before outlier removal	0	0	0	0

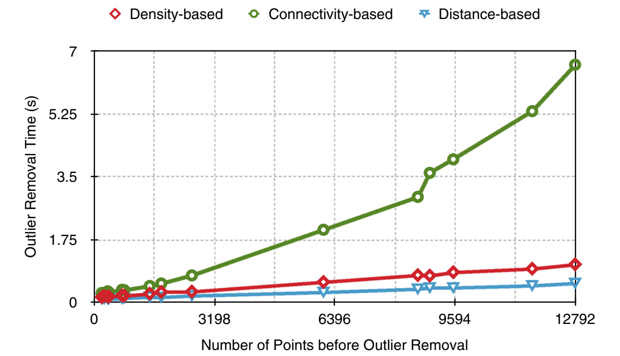

Fig. 4 Runtime performance T_o of outlier removal methods. Outlier removal was applied to 15 models with different sizes

errors resulting in the F value of 0.32 and P value of 0.8 has shown that there is no evidence in difference in the mean values of positioning errors depending on the outlier removal method.

6 Discussion

The problem of outlier removal in photogrammetric point clouds in the context of image-guided localization has not been studied exhaustively before. This study encourages using outlier removal in the applications, where matching time and storage requirements are important constraints for usage. Within our study, two approaches initially designed for point clouds generated with a laser scanner have been implemented and shown applicable for photogrammetric point clouds, too. Hence, we assume that our distance-based approach designed and tested with photogrammetric point clouds is also applicable for laser-scanned point clouds.

The average error of our localization is 0.56 m (Fig. 5, red bar on the bottom-left panel) including the loss in quality of 8 cm (Table 1) after outlier removal. Furthermore, this value additionally accumulates an error gained in the process of alignment to the reference model, which we are unable to extract from the final result. Comparing our results with the usual performance of GPS, when the positioning error can be up to several meters, we consider the loss in quality of 8 cm as reliable and acceptable. The ANOVA test confirms those losses as insignificant.

Together with the fact that the conducted experiment has shown obvious benefits of outlier removal in terms of matching time and space requirements, it makes us believe that our initial hypothesis holds.

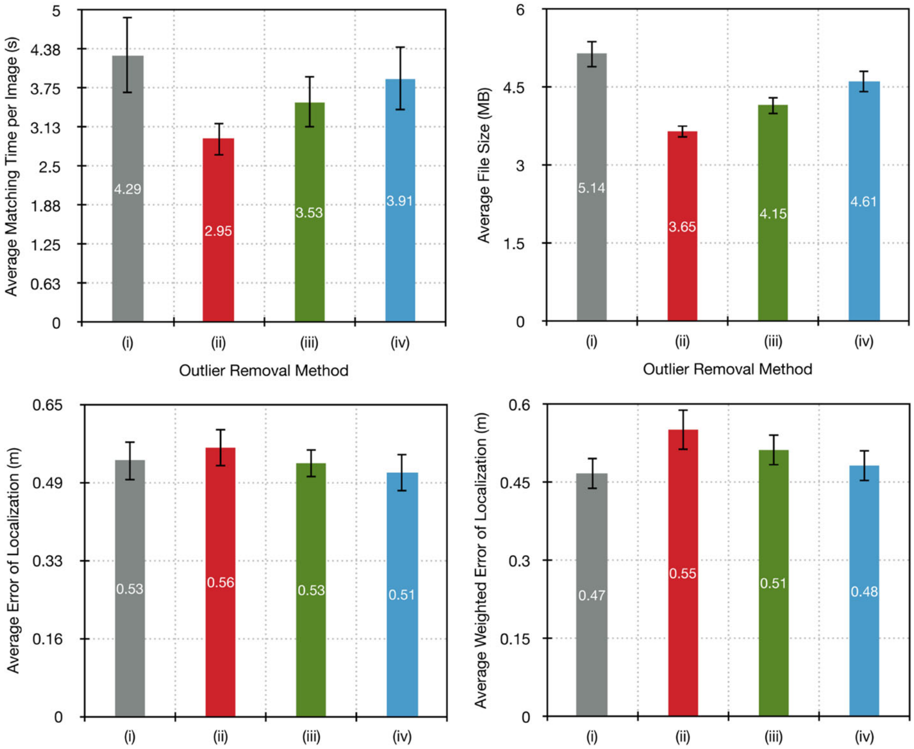

Fig. 5 Evaluation results associated with the models. (i) Original model before outlier removal and the models after (ii) density-based, (iii) connectivity-based, and (iv) distance-based methods applied. Average matching time is the average of T_{mj} returned by each jth test case; average file size—the average of all S_{mj}. Average error of localization is the average of all E_j defined by Eq. (8), and average weighted error is the average of all E_{wj} defined by Eq. (9)

Outlier removal can be applied to numerous tasks of image-based navigation, such as navigation of blind, navigation in the environments where GPS is unavailable (e.g., indoor) or unreliable (e.g., narrow streets with tall buildings), and recognition of landmarks and virtual tours. Not only user-oriented positioning tasks may benefit from outlier removal but also, for example, it may find its use in video-based tracking tasks in medical applications (e.g., colonoscopy, bronchoscopy, panendoscopy). Furthermore, outlier removal is good for applications requiring scene visualization.

In this work, we have evaluated three methods. However, it is not that easy to select the one suitable method for universal use. The superior method certainly is application-dependent. Thus, if a navigational system is equipped with supporting sensors (e.g., accelerometer, gyroscope) and algorithms (e.g., landmarks-based positioning correction) allowing for the adjustment of positioning results, then the fastest method shall be chosen (the density-based method). Otherwise, depending on the required precision, a robuster method would be preferable (our distance-based method). The connectivity-based approach returns also good results; however, due to its cubical algorithmic complexity, the approach is not suitable for the applications requiring iterative outlier removal in the point clouds containing hundreds of thousands of points.

The distance-based method leads to benefits in computational time and storage requirements of about 10%. In defense of the feasibility of using this method, we can say that using this 10% of time improvement, it is possible to match 10% more images, which will lead to a robuster positioning. However, for a better justification, a user study with a working prototype is required. It is necessary to investigate user reaction on the system's performance in terms of the tolerance for waiting time and positioning error. This will be addressed in the future work.

Another future task is incorporation of outlier removal into the bundle adjustment process. Iteratively applying outlier removal after each new nth image (e.g., $n = 100$) might decrease the number of erroneously reconstructed models.

Furthermore, from the perspective of increasing the efficiency of mobile image-based navigation, we believe that a right choice of descriptors (e.g., SIFT, SURF [39], ORB [40], BRISK [41]) may also reduce computational time and models' size. This is a subject of our additional study.

Another method for reducing the number of required matches, and thereby decreasing the time for localization, is pruning the search space. This will be achieved by reducing the points to an area within a certain range around the most likely position (e.g., based on prior position and trajectory). A careful evaluation will be needed to investigate the trade-off between positioning accuracy and matching time. An iterative approach with a growing region around the estimated position is also possible, as the most expensive calculation is the matching process. One can also use the direction from which a point is seen to further reduce the number of eligible points.

Image-based navigation has all chances to become available on a consumer level with a help of modern mobile devices. There are many ways of improving the technology and, with additional optimizations, the task of image-guided navigation has a chance to be performed in real time. Moreover, with the further hardware development, all computational complexity can be shifted to a mobile device, and the models can be stored in the device's memory, which will eliminate the bottleneck of wireless communications between the device and the server and will enable the technology usage when device is offline.

7 Conclusions

We managed to prove our hypothesis that outlier removal in 3D point clouds is beneficial for image-guided mobile navigation. Reduction of the number of points in the models yields to computational speedup and also enables to store more models on a single device, while the changes in positioning accuracy remain unchanged.

Endnote

[1] http://www.gdgps.net/

Acknowledgements
This work was co-funded by the German Federal Ministry of Education and Research (BMBF, Grant No. 16SV5846) and the European Commission's Ambient Assisted Living (AAL) Joint Programme ICT for aging well (EU, Grant No. 810302758160—IMAGO).

Authors' contributions
ES performed literature research on outlier removal, designed the study, developed and implemented the algorithms, performed the evaluation, and wrote the manuscript. SJ coordinated the research part of the IMAGO project and designed and developed the acquisition app and the framework for the server-side data processing. JL performed the data acquisitions and was responsible for the database implementation. DK implemented the 3D reconstruction pipeline and positioning functionality. RH developed the matching and navigational algorithms and prototypes. HS coordinated the overall IMAGO project and image acquisition as well as test runs. TD participated in the study design and coordination and revised the manuscript. All authors read and approved the final manuscript.

Competing interests
The authors declare that they have no competing interests.

Author details
[1]Department of Medical Informatics, Uniklinik RWTH Aachen, Pauwelsstr. 30, 52057 Aachen, Germany. [2]Applied Biomedical Systems, Oxfordlaan 55, 6229 Maastricht, The Netherlands.

Received: 25 April 2016 Accepted: 17 November 2016
Published online: 07 December 2016

References

1. Dep-t of Defence, Global positioning system standard positioning service performance standard, 4-th edition (2008). http://www.gps.gov/technical/ps/2008-SPS-performance-standard.pdf Accessed 1 Oct 2016
2. N Fallah, I Apostolopoulos, K Bekris, E Folmer, in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. CHI '12*. The user as a sensor: navigating users with visual impairments in indoor spaces using tactile landmarks (ACM, New York, 2012), pp. 425–432
3. KM Varpe, MP Wankhade, Visually impaired assistive system. Int. J. Comput. Appl. **77**(16), 5–10 (2013)
4. N Li, B Becerik-Gerber, Performance-based evaluation of RFID-based indoor location sensing solutions for the built environment. Adv. Eng. Inform. **25**(3), 535–546 (2011)
5. SM Jonas, E Sirazitdinova, J Lensen, D Kochanov, H Mayzek, T de Heus, R Houben, H Slijp, TM Deserno, Imago: image-guided navigation for visually impaired people. JAISE. **7**(5), 679–692 (2015)
6. AJ Davison, ID Reid, ND Molton, O Stasse, Monoslam: Real-time single camera slam. Pattern Anal. Mach. Intell. IEEE Trans. **29**(6), 1052–1067 (2007)
7. H Hile, R Vedantham, G Cuellar, A Liu, N Gelfand, R Grzeszczuk, G Borriello, in *Proceedings of the 7th International Conference on Mobile and Ubiquitous Multimedia. MUM '08*. Landmark-based pedestrian navigation from collections of geotagged photos (ACM, New York, 2008), pp. 145–152
8. S Treuillet, E Royer, Outdoor/indoor vision based localization for blind pedestrian navigation assistance. Int. J. Image Graph. **10**(04), 481–496 (2010)
9. J Ventura, C Arth, G Reitmayr, D Schmalstieg, Global localization from monocular slam on a mobile phone. IEEE Trans. Vis. Comput. Graph. **20**(4), 531–539 (2014)
10. P Chippendale, V Tomaselli, V D'Alto, G Urlini, CM Modena, S Messelodi, SM Strano, G Alce, K Hermodsson, M Razafimahazo, T Michel, GM Farinella, in *Computer Vision - ECCV 2014 Workshops: Zurich, Switzerland, September 6-7 and 12, 2014, Proceedings, Part III*, ed. by L Agapito, MM Bronstein, and C Rother. Personal shopping assistance and navigator system for visually impaired people (Springer, Cham, 2015), pp. 375–390
11. C Wu, in *Proceedings of the 2013 International Conference on 3D Vision. 3DV '13*. Towards linear-time incremental structure from motion (IEEE Computer Society, Washington, 2013), pp. 127–134
12. DG Lowe, in *International Conference on Computer Vision, 1999*. Object recognition from local scale-invariant features (IEEE Computer Society, Washington, 1999), pp. 1150–1157
13. FE Grubbs, Procedures for detecting outlying observations in samples. Technometrics. **11**(1), 1–21 (1969)
14. MA Fischler, RC Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM. **24**(6), 381–15 (1981)
15. O Chum, J Matas, in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01. CVPR '05*. Matching with prosac—progressive sample consensus (IEEE Computer Society, Washington, 2005), pp. 220–226
16. D Nistér, in *Proceedings of the Ninth IEEE International Conference on Computer Vision - Volume 2. ICCV '03*. Preemptive RANSAC for live structure and motion estimation (IEEE Computer Society, Washington, 2003), pp. 199–207
17. N Snavely, SM Seitz, R Szeliski, in *ACM SIGGRAPH 2006 Papers. SIGGRAPH '06*. Photo tourism: exploring photo collections in 3D (ACM, New York, 2006), pp. 835–846
18. S Agarwal, N Snavely, I Simon, SM Seitz, R Szeliski, in *Proceedings of the 12th International Conference on Computer Vision. ICCV'09*. Building rome in a day (IEEE Computer Society, Washington, 2009), pp. 72–79
19. A Irschara, C Zach, M Klopschitz, H Bischof, Large-scale, dense city reconstruction from user-contributed photos. Comput. Vis. Image Underst. **116**(1), 2–14 (2012)
20. J-M Frahm, P Fite-Georgel, D Gallup, T Johnson, R Raguram, C Wu, Y-H Jen, E Dunn, B Clipp, S Lazebnik, M Pollefeys, in *Proceedings of the 11th European Conference on Computer Vision: Part IV. ECCV'10*. Building rome on a cloudless day (Springer, Berlin, Heidelberg, 2010), pp. 368–381
21. C Taglioretti, AM Manzino, T Bellone, I Colomina, On outlier detection in a photogrammetric mobile mapping dataset. ISPRS—International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. **XL-3/W2**, 227–233 (2015)
22. AC Atkinson, M Riani, A Cerioli, *Exploring Multivariate Data with the Forward Search. Springer series in statistics*. (Springer, New York, 2004)
23. V Hodge, J Austin, A survey of outlier detection methodologies. Artif. Intell. Rev. **22**(2), 85–42 (2004)
24. MPJVD Loo, *Distribution based outlier detection for univariate data*. (Technical Report 10003, Statistics Netherlands, The Hague, Netherlands, 2010)
25. T Johnson, I Kwok, RT Ng, in *Proceedings of the 4th Int Conf on Knowledge Discovery and Data Mining*, ed. by R Agrawal, PE Stolorz, and G Piatetsky-Shapiro. Fast computation of 2-dimensional depth contours (AAAI Press, New York, 1998), pp. 224–228
26. L Kaufman, PJ Rousseeuw, *Finding Groups in Data: an Introduction to Cluster Analysis*, 9th edn. (Wiley-Interscience, New York, 1990)
27. EM Knorr, RT Ng, in *Proceedings of the 24rd International Conference on Very Large Data Bases. VLDB '98*. Algorithms for mining distance-based outliers in large datasets (Morgan Kaufmann Publishers Inc., San Francisco, 1998), pp. 392–403
28. T Hu, SY Sung, Detecting pattern-based outliers. Pattern Recogn. Lett. **24**(16), 3059–10 (2003)
29. MM Breunig, H-P Kriegel, RT Ng, J Sander, in *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data. SIGMOD '00*. LOF: identifying density-based local outliers (ACM, New York, 2000), pp. 93–104
30. S Sotoodeh, in *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences XXXVI-5*. Outlier detection in laser scanner point clouds (Copernicus Publications, Göttingen, 2006), pp. 297–302
31. S Sotoodeh, in *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences XXXVI-3*. Hierarchical clustered outlier detection in laser scanner point clouds (Copernicus Publications, Göttingen, 2007), pp. 383–388
32. D Luo, L Liao, in *Proceedings of the 2010 International Conference on Artificial Intelligence and Education (ICAIE)*. Mining outliers from point cloud by data slice (IEEE Computer Society, Washington, 2010), pp. 663–666
33. K Zhang, M Hutter, H Jin, in *Proceedings of the 13th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining. PAKDD '09*. A new local distance-based outlier detection approach for scattered real-world data (Springer, Berlin, Heidelberg, 2009), pp. 813–822
34. J Wang, K Xu, L Liu, J Cao, S Liu, Z Yu, XD Gu, Consolidation of low-quality point clouds from outdoor scenes. Comput. Graph. Forum. **32**(5), 207–10 (2013)
35. C Strecha, T Pylvänäinen, P Fua, in *Proceedings of the 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Dynamic and scalable large scale image reconstruction (IEEE Computer Society, Washington, 2010), pp. 406–413
36. O Untzelmann, T Sattler, S Middelberg, L Kobbelt, in *Proceedings of the 2013 IEEE International Conference on Computer Vision Workshops (ICCVW)*. A scalable collaborative online system for city reconstruction (IEEE Computer Society, Washington, 2013), pp. 644–651
37. OpenStreetMap, OpenStreetMap contributors (2014). https://www.openstreetmap.org/. Accessed 25 Feb 2016
38. Z Zhang, Iterative point matching for registration of free-form curves and surfaces. Int. J. Comput. Vis. **13**(2), 119–152 (1994)
39. H Bay, A Ess, T Tuytelaars, LV Gool, Speeded-up robust features (surf). Comp. Vision Image Underst. **110**(3), 346–359 (2008)
40. E Rublee, V Rabaud, K Konolige, G Bradski, in *Computer Vision (ICCV), 2011 IEEE International Conference On*. Orb: an efficient alternative to sift or surf (IEEE Computer Society, Washington, 2011), pp. 2564–2571
41. S Leutenegger, M Chli, RY Siegwart, in *Computer Vision (ICCV), 2011 IEEE International Conference On*. Brisk: binary robust invariant scalable keypoints (IEEE Computer Society, Washington, 2011), pp. 2548–2555