ExMaps: Long-Term Localization in Dynamic Scenes using Exponential Decay

Alexandros Rotsidis Christof Lutteroth Peter Hall Christian Richardt

Centre for Digital Entertainment, University of Bath, United Kingdom

{A.Rotsidis, C.Lutteroth, P.M.Hall}@bath.ac.uk, christian@richardt.name

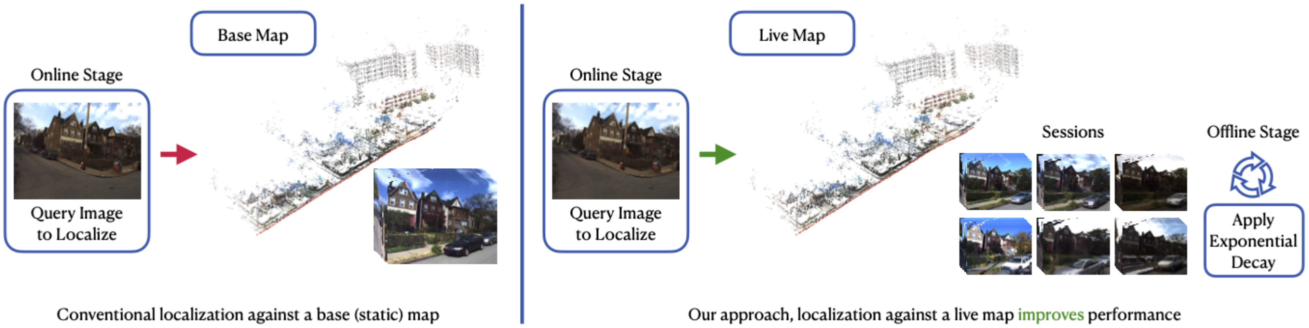

Figure 1. We propose a new approach for long-term localization in dynamic environments that updates a *base map* (left) over time as new data arrives to maintain an up-to-date *live map* (right). We introduce new stability scores for each world point that take into account both visibility and recency of observation (using an exponential decay) to distinguish reliable static and unreliable dynamic world points. We show that this enables localization of unseen query images in the live map with higher accuracy and speed compared to using the initial base map.

Abstract

Visual camera localization using offline maps is widespread in robotics and mobile applications. Most state-of-the-art localization approaches assume static scenes, so maps are often reconstructed once and then kept constant. However, many scenes are dynamic and as changes in the scene happen, future localization attempts may struggle or fail entirely. Therefore, it is important for successful long-term localization to update and maintain maps as new observations of the scene, and changes in it, arrive. We propose a novel method for automatically discovering which points in a map remain stable over time, and which are due to transient changes. To this end, we calculate a stability store for each point based on its visibility over time, weighted by an exponential decay over time. This allows us to consider the impact of time when scoring points, and to distinguish which points are useful for long-term localization. We evaluate our method on the CMU Extended Seasons dataset (outdoors) and a new indoor dataset of a retail shop, and show the benefit of maintaining a 'live map' that integrates updates over time using our exponential decay based method over a static 'base map'.

Acknowledgments *This research was supported by Dcactiv, the EPSRC CDE (EP/L016540/1), and an EPSRC-UKRI Innovation Fellowship (EP/S001050/1).*

1. Introduction

Pose estimation is a fundamental problem in 3D computer vision: determining a camera's position and orientation relative to a map from just a single image is crucial for many applications, including robotics and augmented reality (AR). Many traditional approaches assume that the camera moves through a static world, such that a map can be reconstructed once and then reused afterwards. However, this is not the case for many real-world environments, which are full of dynamics, including moving people, cars or trees, and changing weather, daylight and seasons. Being able to localize in an environment despite these dynamic changes, at future points in time, is referred to as *long-term localization* [2, 9, 37].

In this work, we focus on long-term localization within retail environments, which have to date received little attention compared to other environments, such as outdoor street scenes [45]. The main challenge with retail shops is their highly dynamic nature, as noted by Spera et al. [41]. Shoppers can pick up products and put them back, stores can restock as often as twice a day, temporary promotions might lead to products being rearranged, advertising posters are rotated regularly, and people may pass in front of the camera and occlude the scene. See the real-world example in Figure 2. This makes long-term localization in retail environments particularly challenging.

Long-term localization in retail environments enables a range of new applications, particularly for AR on consumer smartphones, which have the potential to revolutionize the sector. Many retail shops are keen to target specific shoppers through personalization. For example, a shopper might be shown additional information for a product, a comparison between similar products nearby, or a map of the shop could be adapted to show promotions based on a shopper's preferences. In large shops, localization can be used for in-shop navigation or for finding a specific product. A development of interest is the monitoring of the paths of customers therefore enabling analysis of their behavior [32, 41]. In future, localization could also be useful for robots that automatically restock products. From a business point of view, variations of product arrangements can be visualized on top of an aisle for preview and planning purposes, and maps can be rendered in AR for training new personnel using AR headsets such as the HoloLens.

We propose a new long-term localization approach that is specifically tailored for fast-paced and highly dynamic retail environments. We build and maintain a *live map* that is regularly updated based on new observations made during shoppers' AR sessions. The key to maintaining our live maps is to deduce which 3D scene points remain stable over time, and which are changing and thus unreliable for localization, based on their visibility over time. Specifically, we calculate a stability store for each point that decays exponentially over time if a point is not observed regularly. In combination with progressive sampling, this reduces the influence of unreliable points during pose estimation, and thus makes localization more robust in the long term. Our main contributions are:

1. A new map maintenance method that enables long-term localization in dynamic scenes by prioritizing stable 3D points over time.

2. Two novel time-varying point stability score for use with progressive samplers like PROSAC [8].

3. A new dataset for benchmarking and evaluating long-term localization in dynamic indoor environments.

2. Related Work

The review of related work will focus on camera pose estimation, primarily on structure-based methods, offline map maintenance for long-term localization [37], and indoor localization, specifically for retail environments. Camera pose estimation can be divided into three categories:

1. Image-based retrieval methods, match a query image against a database to find nearby images. This provides an initial estimate of the query image pose by retrieving similar images from a database [1, 6, 34, 47, 50, 51]. The camera pose can then be refined by looking for matches between the query image and the top-k retrieved images, or by subsequent matching to 3D pre-built map points [5].

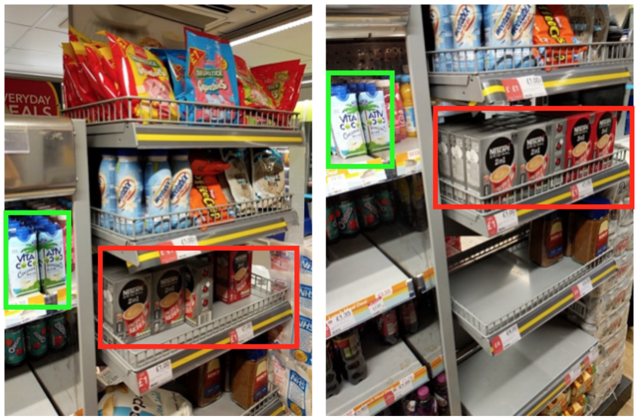

Figure 2. A highly dynamic retail environment, eight hours apart. Only some features remain reliable for localization (in green), while others (in red) would lead to mismatches and thus errors.

2. Learning-based methods are more recent and may directly regress camera pose from an input image [19, 20, 49] by representing the scene as the weights of a neural network. Learning-based methods do not yet outperform structure-based methods [36, 38] Other learning methods try to learn the feature matching function, i.e. given a small image patch they predict a 3D coordinate [29]. While some learning based methods as Brachmann and Rother [4] can achieve a higher pose accuracy than feature-based methods, but they might fail training in large outdoor scenes.

3. Structure-based methods are based on pre-built structure-from-motion maps, usually 3D point clouds. These methods tend to provide more accurate poses than image retrieval methods [36]. They typically assign one or more feature descriptors, such as SIFT [25], to each 3D point. For a given query image and its keypoint descriptors, 2D–3D correspondences are established using descriptor matching. Once these correspondences are obtained, solving a perspective-n-point (PnP) problem returns the camera pose [14, 18, 22, 26, 33, 35]. Adequate correct matches must be found to be able to estimate a pose, but also to verify that pose via inlier counting. This number of correct matches is typically 12 or more [21, 22]. Sattler et al. [37] showed that this condition might not be sufficient if the query image is taken under significantly different conditions. For a retail shop environment, this could, for example, be a query image of an aisle taken after restocking the previous day.

Structure from motion can be classified based on how well complex scenes are handled [46]. Prioritizing matching [7, 21, 35] improves efficiency by terminating once a set number of matches is found. Other approaches restrict the search space to improve speed [18, 22, 24], either using image retrieval [18] or co-visibility information [22, 24]. The quality of correspondences in changing conditions can be improved by scoring matches based on image semantics [43, 46]. Changing conditions can be related to the weather, as clouds change the lighting conditions, or seasons, e.g. different colour of tree leaves in autumn and spring, or snow during winter. Toft et al. [46] assign semantic 3D information beforehand and segment new query images in categories such as vegetation, road or building. If the category label in the query image matches the label of the 3D point, then it is considered a good candidate.

All the above methods aim to improve 2D–3D matching within a static point cloud. In contrast, we aim to improve the subsequent pose estimation stage, specifically the filtering and rejection of matches, to robustly handle dynamically changing scenes. We hence introduce the notion of time as new data arrives and update the point cloud to generate a time-dependent stability score for each 3D feature point.

Map maintenance for long-term localization is the process of updating a point cloud such that localization can still be achieved after some time, for instance by exploiting additional images from new user sessions. Utilizing information from multiple traversals at different points in time has been explored before [11, 12, 30]. Keeping a static map in memory, and localizing against it, is insufficient for long-term localization given that the scene's appearance changes frequently [11]. For example, a retail shop will look different after customers have emptied the shelves at the end of the day.

Dymczyk et al. [11] maintain a point cloud of fixed size by determining which points from new sessions to keep and which ones to discard from the old map. Similar to Mühlfellner et al. [30], they use scoring and sampling functions to determine which points are worth keeping. One potential scoring function is the number of times a point has been observed. The more often a point was seen, the more likely it is to be stable and thus a good match for localization. Another proposed sampling function prefers points with similar descriptors that show little variance.

Churchill and Newman [9] save complete past sessions and use multiple localizers to associate session data to previous sessions. If a low number of localizers succeeded, then the current session is added as new, and it is used for future localizations. However, in areas with many daily changes, such as retail shops, the number of stored sessions increases quickly, resulting in large memory consumption. We differ from this approach as we only append descriptors on 3D points, and compute a time-variant score per point. FAB-MAP [10] has been one of the most used place recognition algorithms [11]. It is appearance-based, uses a *bag of words* representation of sensor data, and scenes are represented as words from a vocabulary. Incoming data is then matched to the closest word. While it provides a level of robustness, its matching still fails over large changes in environmental appearance [27].

Indoor localization [23, 40, 48] has not received a lot of attention compared to outdoor localization [16, 18, 21, 22, 26, 46, 50]. Localization in retail shops has received even less attention [41]. Indoor localization is particularly challenging because of symmetric and repetitive elements, large image changes as furniture/items can move around, and the lack of features in texture-less areas [45]. Retail shops are different as they are full of various products and thus tend to be more textured. The most related work to ours is by Spera et al. [41, 42], who have benchmarked images-based retrieval techniques and methods based on regression against a dataset of images from a retail shop. Spera et al. [41] aimed for the 3DoF pose of a shopping trolley, while we aim for full 6DoF poses to support handheld mobile AR. They did not investigate 3D structure-based localization methods due to hardware limitations and complexity. Since we seek accurate camera poses and would like to avoid retraining for each new shop, we extend 3D structure-based methods to support long-term localization in dynamic environments. We refer to work by Dymczyk et al. [11] and Sattler et al. [37] for a more complete picture of the field.

3. Methodology

Long-term localization is a hard problem due to the challenge of establishing reliable correspondences between a query image and the world map. Our proposed method assumes that the same dynamic place is visited frequently, e.g. via a warehouse robot or a self-driving car, or by shoppers in a shop. Our goal is to maintain the number of map points and to augment them with metadata that reflects the changes over time, such that future localizations are more likely to succeed. We draw inspiration from Stylianou et al. [44], that keeping descriptors over time can improve feature matching. We follow a standard feature-based pose estimation pipeline, based on prebuilt structure-from-motion maps. In each *session*, a set of new timestamped images arrives, which are used to update the *live map* by appending new descriptors to the existing 3D points and recomputing point stability scores using exponential decay. In this paper, we update the live map at the end of each session. Features are extracted from a query image, then using nearest-neighbor matching, we match them with to the mean descriptors of 3D points. Lowe's ratio is applied for outlier filtering before estimating camera poses using PROSAC [8] or RANSAC [13]. Finally, we refine the pose matrix by refitting it to all inliers.

The visibility matrix $\mathbf{V}$ represents which points (columns) are visible in which image (rows). An element $v_{i,p}$ in this matrix is 1 if and only if point p is visible in image i, and it is zero otherwise. This is easily obtainable from SfM software. After every session, points will have a new stability score we define as σ, inferred by applying exponential decay. A point with a high score is more likely to be observed in future sessions. Using these scores, we focus on the subsequent pose

Table 1. Toy example that illustrates two types of stability scores, per session and per image, applied to an example visibility matrix of 4 points (columns) and 6 images (rows), captured in 3 sessions (bottom entries are the most recent images). The columns under 'session' and 'image' illustrate the corresponding weights given to each visibility value in the row. The *per-session* stability score (Equation 2) treats all images in a session the same and thus cannot distinguish between points 2 and 3. The *per-image* stability score (Equation 3) favors more recent observations, even in the same session.

Session		**Image**		**Visibility matrix**			
#	weight	#	weight	P1	P2	P3	P4
1	0.13	1	0.13	1	0	0	1
1	0.13	2	0.18	1	0	0	1
2	0.25	3	0.25	0	1	0	0
2	0.25	4	0.35	0	0	1	0
3	0.50	5	0.50	0	0	0	0
3	0.50	6	0.71	1	0	0	0
stability per session – σ_p^{s}				0.76	0.25	0.25	0.25
stability per image – σ_p^{i}				1.02	0.25	0.35	0.31

estimation stage, that is improving the matching filtering stage, by including time data, in RANSAC and PROSAC as a score as we explain further in this section. We call this updated map the *live map*.

For evaluation and comparison purposes, we also prepare a *base map*, i.e. a map constructed using data only from a single session without any subsequent updates.

3.1. Exponential Decay of Visibility

Our key insight is that once a point has been observed in an image, the value of this visibility information decreases over time, becoming increasingly unreliable as time passes due to the dynamic nature of the world. We model this decrease over time as an exponential decay:

$$N(t) = 2^{-\lambda t}, \tag{1}$$

where λ is the decay constant and t is the time that has passed. This is inspired by nuclear physics, where exponential decay models how radioactive atoms decay over time, i.e. how they lose energy by emitting ionizing particles. In that scenario, λ^{-1} is called the *half-life* and measures the time required for a quantity to decay to half its initial value. For example, for a decay of $\lambda = 1$, each time step results in decay by half, i.e. $N(0) = 1$, $N(1) = 0.5$, $N(2) = 0.25$, and so on.

Table 1 shows an illustrative toy example with a visibility matrix of 4 points and 6 images. We show two cases where the exponential decay has been applied on a per-session or per-image basis, which we explain in more detail next. For example, point P1 in Table 1 was seen in the most recent image (#6) and session (#3), and is thus weighted highly. We consider images in the visibility matrix already decayed because we treat all images in $\mathbf{V}$ as past images.

Per-Session Stability Score Each new session of localized images from SfM is appended to the bottom of the visibility matrix $\mathbf{V}$ as a row. The values of the visibility matrix are then weighted *per session*, as illustrated in Table 1. Specifically, for session number $s \in [1, S]$, the weight is $N(S - s + 1)$, for a decay of $\lambda = 1$, which corresponds to a half-life of one session. We sum the weighted visibility for each point p across all images to obtain the per-session stability score:

$$\sigma_p^{\text{s}} = \sum_{i=1}^{I} N(S - s_i + 1) \cdot v_{i,p}, \tag{2}$$

where I is the number of images (number of rows of visibility matrix $\mathbf{V}$), s_i is the session number of image i, and $v_{i,p}$ is the visibility of point p in image i.

Per-Image Stability Score Our second stability score provides more granularity by weighting points on a per-image basis, giving more recent observations a higher weight:

$$\sigma_p^{\text{i}} = \sum_{i=1}^{I} N(I - i + 1) \cdot v_{i,p}. \tag{3}$$

To achieve a similar speed of decay as the per-session stability score σ_p^{s}, we set the decay constant to $\lambda = S/I$, such that the half-life $\lambda^{-1} = I/S$ corresponds to the average number of images per session. This is why images 1, 3 and 5 in Table 1 have the same weight for both stability scores. Both methods provide coarse and fine accuracy accordingly. In Table 1, point 2 and point 3 have the same s_s value. The s_i distribution though differentiates between the two points, and assigns to point 3 a higher value.

Both stability scores summarize how recently and frequently points have been observed over time. A high score likely identifies a point that was visible in the last few sessions. Points on permanent structures, such as shop shelves, furniture or long-term decorations will have the highest scores. This is because points on stable structures will be seen more frequently and they thus accumulate more and higher weights in their stability scores. Another point, even if seen multiple times in the past, might still receive a lower score compared to the former. This accurately models changes in a dynamic scene such as a retail store or outdoor environment. Products that are on a shelf and seen by a recent customer will tend to be better candidates for localization, than products that were not seen either because of occlusion or having been bought or removed. If products are bought or removed, the stability score of associated points will continue to decay over time. Once products are restocked or put back, the stability score of points on the product will increase again. In most retail shops, products tend to be at the same spot when restocked. Images with scores below a user-defined threshold could be culled to save memory and computational resources.

3.2. Pipeline

In this section, we present our long-term localization pipeline in more detail. The first stage is to create a base map from the initial session. For this, we use COLMAP [39] with default settings and extract up to 2000 features for reconstruction. For incoming new query images, we extract 800 SIFT features and use brute-force matching between the descriptors of the 2D keypoints and the mean descriptor for each 3D point in the map. We use 800 features to match modern SLAM frameworks, such as ORB-SLAM2 [31], which use numbers of around 1,000. Every time a new image is added to the map, we append the newly extracted descriptors to the corresponding 3D points' list of descriptors to update the mean descriptors of each 3D point.

We fetch the two nearest neighbors for a query descriptor, and we pick the one with the lower Euclidean distance from the query descriptor. We use a value of 0.9 for Lowe's ratio test [25], similar to Toft et al. [46], to avoid rejecting correct matches. As a baseline, we run RANSAC [13], as well as a weighted variation of RANSAC ("with σ^{s}") similar to Toft et al. [46]. This variation does not pick random matches uniformly, but according to a predefined probability density, which we obtain by normalizing the per-session stability scores σ^{s} of all identified matches. For example, if there are 50 matches between a query image's descriptors and the corresponding 3D points in the live map, then 50 σ^{s} scores will be normalized and used as a distribution to sample from. This way, we promote that more stable points are selected preferentially. We use a maximum of 3,000 iterations for both RANSAC versions.

To optimally utilize our new stability scores, we use *PROSAC* [8], which stands for progressive sample consensus. Unlike RANSAC, which treats all matches the same by drawing random samples uniformly, PROSAC adds a notion of quality or priority to matches. When using PROSAC, matches must be sorted in descending order based on some score, the higher the score, the better the match. By default, PROSAC uses the inverse of Lowe's ratio [8]. In the next section, we evaluate and compare a variety of PROSAC versions using different combinations of our stability scores and Lowe's ratio.

For both RANSAC and PROSAC, we use an inlier threshold of 5 pixels for the reprojection error, like Sattler et al. [37]. Once we have the correspondences from RANSAC or PROSAC, we estimate the camera pose using P3P [14] based on all inlier matches.

4. Experiments and Evaluation

In this section, we evaluate and compare our proposed approach to baseline methods on two datasets. We next introduce these datasets and baselines, as well as the metrics we use for evaluation and comparison.

4.1. Datasets

Most long-term localization datasets, such as *CMU Extended Seasons* [37] and *RobotCar Oxford* [28], are focused on outdoor environments, which mostly exhibit cyclical changes, such as day/night, seasonal and weather changes. Indoor scenes, especially retail shops, arguably exhibit less cyclical and more random changes, mostly caused by human interactions, such as moving objects around. Current outdoor datasets for long-term localization are therefore not ideal for evaluating methods like ours that are tailored to indoor localization. Since there are no available datasets for long-term localization in retail shops, we created a new dataset for this specific purpose.

CMU Extended Seasons [3, 37] This dataset contains 100,000 images taken over a period of 12 months in Pittsburgh, PA, USA. Images were captured with a stereo rig mounted on an SUV. We only used the left camera view for our benchmarks to simulate a standard monocular session. The dataset contains 25 *slices*, each representing a traversal of a specific sub-area of the complete route. We use 5 slices for benchmarking, specifically slices 3, 4, 6, 10 and 11. Each slice contains images taken at different times, months apart, in 12 different conditions. These conditions are our sessions.

We reconstruct the base map from the 'sunny/no foliage' condition, like Sattler et al. [37]. This base map is visualized in Figure 3. We set aside one session to use as a query session. We use the remaining sessions as additional sessions to create the live map on top of the base map. To create the base map and obtain ground-truth poses for the query session images, we use COLMAP [39]. We obtain the query images' ground-truth poses by localizing them with COLMAP in the live map after localizing all other remaining sessions, but do not keep the poses in that map. In a real-life scenario, the data will be continuously and perpetually added as it is received. For now, we only have a fixed number of sessions in a particular dataset. Not all query images localize, as expected. For those that do localize, we save their pose as ground truth for later evaluation. To validate that the query images localized accurately, we visually inspected the latest live map's 3D points projected on each query image, using COLMAP camera poses and intrinsics, and found the results were satisfactory as points were projected correctly on the images.

Retail Shop Dataset Since there are no available datasets for long-term localization in retail shops, we created our own. We collected the dataset over the span of 5 days, in which we

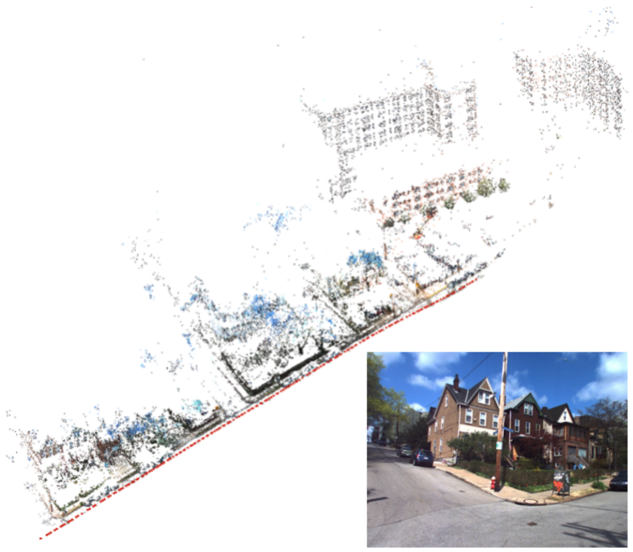

Figure 3. The base map for the CMU dataset is constructed using only the left camera images in the ‘sunny/no foliage’ condition. The starting point is shown in the bottom right.

visited a local shop twice a day, once at opening time in the morning and once later in the evening before closing time (see example in Figure 2). This allowed for sufficient time in between for customers to interact with the scene by picking up products, and for the shop to be restocked twice a day. Each session contains about 450 RGB images of 640×480 resolution obtained from ARCore [15], and covered both sides of a small shopping aisle. We used ARCore as a foundation for a larger augmented reality application. We chose a low image resolution to reduce network bandwidth and transfer time when sending images to a server.

We use images from the first day to construct the base map. For the live map, we use session images taken subsequently and hold out one session for querying, similarly for the CMU Extended Seasons dataset. Considering that most sessions in the retail shop were captured close in time, most images were localized successfully.

4.2. Baselines and Comparisons

We quantitatively compare several versions of our proposed method to vanilla RANSAC [13] and vanilla PROSAC [8], on both the base and live map. PROSAC uses the inverse Lowe’s distance ratio of d_2/d_1, where d_1 is the distance of the closest 3D point’s descriptor, and d_2 the distance of the second-closest match. We also compare to our modified RANSAC version, using the per-session stability score σ^{s} (“with σ^{s}”) against the same version of RANSAC using $v_p = \sum_{i=1}^{I} v_{i,p}$, which is simply the sum of each column in the visibility matrix (“with v_1”). This equates to the number of cameras a point was viewed from, which is one scoring function proposed by Dymczyk et al. [11]. In both cases,

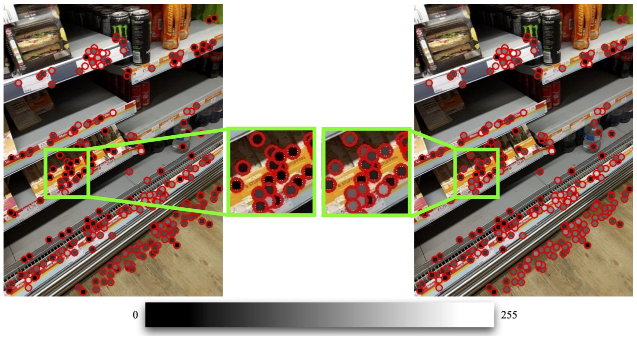

Figure 4. To illustrate the difference between our proposed time-infused score and the plain visibility score (i.e. the number of cameras a point was viewed from), we selected a random picture from the live map and project the map onto it. Each point here is associated with a stability score σ^{i} (left image) and a visibility score v_1 (right image). The higher the value, the more stable the point is. Most of the products’ points from the left cropped-image are **darker** from the ones in the right. This confirms our hypothesis, that those points on the products (i.e. dynamic objects) and are less likely to be used for pose estimation using our score σ^{i}.

we normalize the scores across all matches to constructing a probability distribution to sample from. This enables us to preferentially sample stable matches during the inner RANSAC loop. Finally, we estimate two poses for each query image, one in the base map and one in the live map, using our matching function. We compare both poses to the ground-truth poses obtained from COLMAP as described in Section 4.1 and measure the error between them.

We explain the score used for RANSAC and its variations, and the PROSAC scores. A 3D point from a live map is associated with two scores, σ^{s} and σ^{i}. From the feature matching stage, an image’s query feature will be associated with the two nearest neighbors, i.e. 3D points. Each 3D point from the live map has a σ^{s} and σ^{i}. An image’s query feature after matching will be associated with 6 scores, used in evaluation. Two for its nearest neighbor σ^{s}_1, σ^{i}_1 and two for its second nearest neighbor σ^{s}_2, σ^{i}_2. Finally two v_1 and v_2. In Figure 4 we visualize the difference between the stability score σ^{i} and the simple score of v. Darker points indicates more dynamic points, thus will have a lower probability being used in pose estimation, which is the case using score σ^{i}. On the contrary using the score v, products (dynamic objects) have a higher value, lighter color, which is undesirable. In Table 2 we show the scores we used for PROSAC’s scoring list and our RANSAC’s distribution.

Lowe [25] proposed a simple method for filtering keypoint matches by removing matches when the second-best match is almost as good as the first one. The rationale behind the inverse of *Lowe’s ratio* is that the higher the ratio d_2/d_1, the more informative the match is [8]. Using the same logic we assume a match is more informative when the ratio between the stability score per image values is higher, $\sigma^{\mathrm{i}}_1/\sigma^{\mathrm{i}}_2$.

Table 2. Comparison of the convergence behavior of different RANSAC and PROSAC versions with various scoring functions when applied on the base and live maps. Columns show the average number of inlier and outlier matches, the number of iterations, run time in seconds, and the average pose and orientation errors in meters and degrees, respectively. Each number is the average value over 15 independent runs. Notice that the translation errors for the retail shop are lower than for CMU due to the smaller size of the scene (~6 m compared to hundreds).

	5 CMU Slices						**Retail Shop**					
	Inl.	Outl.	Iters	Secs	Er.[m]	Er.[°]	Inl.	Outl.	Iters	Secs	Er.[m]	Er.[°]
Base Map												
RANSAC [13]	47	105	749	1.52	2.00	5.46	86	123	320	0.15	0.07	5.46
PROSAC [8] (d_2/d_1)	38	113	30	0.30	1.58	4.77	72	137	26	0.05	0.07	7.41
Live Map												
RANSAC [13]	57	108	519	1.00	0.46	2.06	104	124	246	0.12	0.02	2.18
RANSAC with σ_1^{s}	58	107	364	1.04	0.38	1.71	106	122	163	0.14	0.02	1.88
RANSAC with v_1	58	107	386	0.95	0.40	1.98	106	122	166	0.14	0.02	1.53
PROSAC [8] (d_2/d_1)	46	119	8	0.13	0.92	3.44	83	145	10	0.04	0.07	6.65
PROSAC with $\max(\sigma_1^{\text{s}}, \sigma_2^{\text{s}})$	41	125	75	0.22	1.74	3.75	84	145	44	0.05	0.05	4.94
PROSAC with $\max(\sigma_1^{\text{i}}, \sigma_2^{\text{i}})$	40	126	78	0.26	1.51	3.21	82	146	42	0.06	0.05	5.09
PROSAC with $\max(v_1, v_2)$	40	126	77	0.25	1.88	4.06	83	145	40	0.05	0.06	5.27
PROSAC with $\sigma^{\text{i-LR}}$	47	119	65	0.20	0.76	3.13	89	140	53	0.05	0.04	3.79

Similarly for the stability score per session, $\sigma_1^{\text{s}}/\sigma_2^{\text{s}}$. A high value for σ_1^{s} means that this is a more static point compared to a low value of σ_2^{s}, the value of the second nearest neighbor. There needs to be enough difference between the best and second-best matches [25]. Since a higher value of the inverse Lowe's ratio, d_2/d_1 and $\sigma_1^{\text{s}}/\sigma_2^{\text{s}}$ indicate a more discriminative and stable point respectively, we use the product of both as a score, denoted as $\sigma^{\text{i-LR}}$.

In our RANSAC versions for the live map, we used σ_1^{s} and v_1. For each set of matches, we normalized the values and use them to sample potential inliers, similarly to Toft et al. [46], instead of uniformly drawing as in vanilla RANSAC.

4.3. Metrics

Here we explain the metrics we use to measure and compare the performance and accuracy of methods in Table 2. All reported metrics are averaged over 15 independent runs.

We measure performance using the number of inliers, outliers, RANSAC/PROSAC iterations and run time. Generally, the higher the number of inliers and the lower the number of outliers, the more robust the estimated camera pose is. A higher percentage of inliers also tends to decrease the maximum number of iterations used by the RANSAC/PROSAC model fitting, and thus the required run time. Run times are measured on a computer with a quad-core 4 GHz CPU and 16 GB RAM. All methods are implemented in Python.

We measure the accuracy of poses similar to Sattler et al. [37]. The position error is computed as the Euclidean distance between the estimated and ground-truth camera centers, $\|C_{\text{est}} - C_{\text{gt}}\|_2$, while the absolute orientation error is calculated from the estimated rotation matrix $\mathbf{R}_{\text{est}}$ and the ground-truth rotation matrix $\mathbf{R}_{\text{gt}}$ [17].

Table 3. Average number of matches per image for each slice of the CMU dataset and the Retail Shop. Our live map consistently produces more matches than the base map, on average 10% more.

	Base Map Matches	Live Map Matches
CMU Slice #3	160	**175**
CMU Slice #4	218	**222**
CMU Slice #6	140	**165**
CMU Slice #10	126	**136**
CMU Slice #11	113	**130**
Retail Shop	209	**229**

4.4. Results

Table 2 shows the mean number of inliers, outliers and iterations, run time in seconds, as well as translation and rotation errors of all the query images for all five slices of the CMU dataset and the retail shop dataset, for both the base and live map. The results show that the live map consistently increases the number of inliers compared to the base map, for all RANSAC and PROSAC versions. This clear increase allows some query images that failed to localize in the base map to be localized in the live map. The live map also consistently reduces run time *and* pose errors compared to using the base map. The most accurate poses are obtained using RANSAC with our per-session stability score σ_1^{s} on the CMU dataset, and using RANSAC with summed visibility on the retail shop dataset. Among all versions of PROSAC, our score $\sigma^{\text{i-LR}}$ consistently returned more inliers and lower pose errors than all other PROSAC versions on both datasets. We suggest that this version of PROSAC is a good alternative

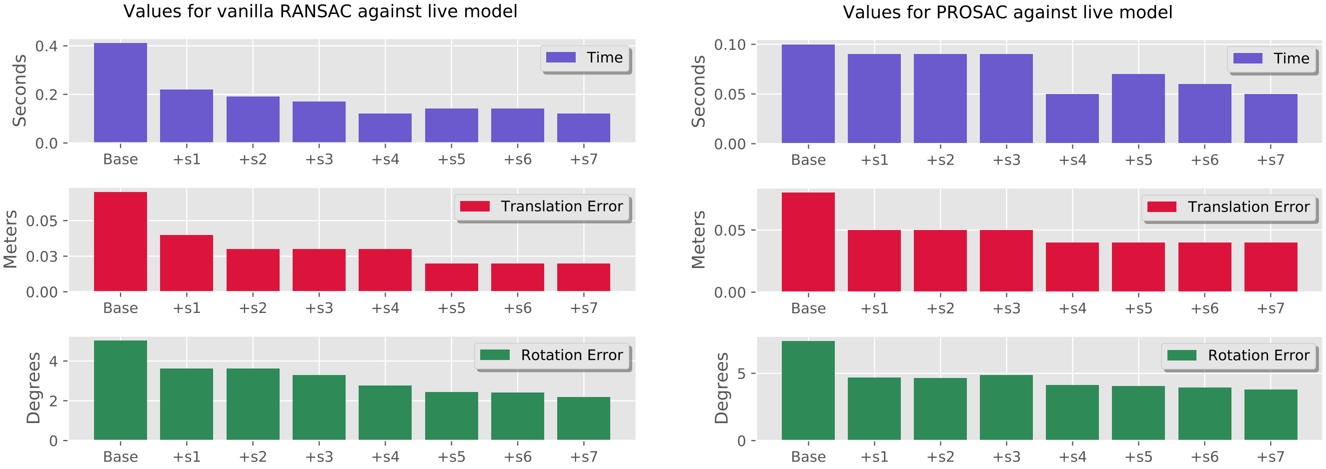

Figure 5. Decrease in run time, translation and rotation errors as more sessions are added to the live map of the retail store dataset. On the left, we show vanilla RANSAC [13], and on the right, we show our PROSAC version using the $\sigma^{\text{i-LR}}$ score.

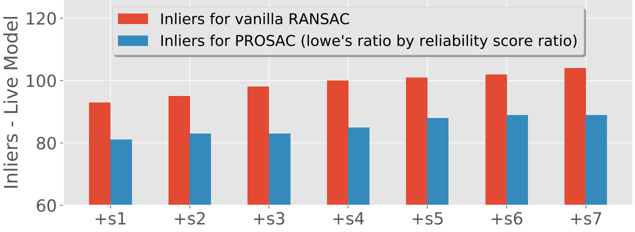

Figure 6. Number of inliers as 7 sessions are added over time for the retail shop live map, for vanilla RANSAC (red) and our PROSAC version with stability score $\sigma^{\text{i-LR}}$ (blue). In both cases, there is a steady increase of inliers as more sessions are added to the live map, indicating more robust long-term localization.

to RANSAC on the live map if speed is a priority and not accuracy. All PROSAC versions were 2–8 times faster than RANSAC, with vanilla PROSAC being the fastest.

Table 3 shows the average number of matches per image for each CMU slice and the retail shop dataset. There is a consistent increase in the number of matches from the base map to the live map, on average around 10 percent more. One can argue that they could be noise missed by Lowe's ratio. This is not the case given the improvements in performance shown in Table 2.

Figure 5 and Figure 6 show that the live map keeps improving over time as more and more sessions of images are added. As shown in Figure 6, there is a consistent increase in the number of inlier matches during both RANSAC and our PROSAC version with $\sigma^{\text{i-LR}}$ score. At the same time, Figure 5 shows that both the run time and the translation and rotation errors are steadily reduced as additional sessions are integrated into the live map of the retail shop dataset.

These results demonstrate that including information about *time* for a 3D point leads to improved long-term localization results compared to relying only on appearance information, as represented by SIFT features.

5. Conclusion and Future Work

In this paper, we proposed a new method that exploits the value of time in structure-from-motion data to enable more robust long-term localization in dynamic environments. We show that by applying exponential decay on a visibility matrix extracted by SfM, we can score the stability of all 3D points over time on a per-session or even per-image basis. Our results demonstrate that live maps continuously improve over time as new data is integrated. However, this requires frequent data input from users, which might be a limitation in some circumstances. Also, new session data should be able to uniformly localize. If only segments of a session localize the live map then will contain parts that are not updated hence useful for future localizations. Collecting data over a year so all the seasonal cyclic changes are captured could be a possible "bootstrap" stage before deploying an *ExMap*. Our findings also confirm earlier statements by Mühlfellner et al. [30] and Stylianou et al. [44] that merging data from multiple sessions and keeping SIFT data over time helps improves long-term localization. This provides an improvement over static maps, which are not updated over time and thus quickly become outdated, leading to an increased failure rate of localization.

The experiments carried out have shown that along with appearance information such as SIFT descriptors when our time-related scores are combined we can improve the number of inliers returned by PROSAC/RANSAC and increase the pose accuracy, without increasing the number of points in a map. Over time we only keep the descriptors of previous localized images for each existing 3D point, thus our method is not memory-expensive. In future work, we would like to explore adding new points or compressing existing maps based on these scores. We plan to release the source code of our method and the retail shop dataset.

References

[1] Relja Arandjelović, Petr Gronát, Akihiko Torii, Tomás Pajdla, and Josef Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. In *CVPR*, 2016. 2

[2] Roberto Arroyo, Pablo Fernández Alcantarilla, Luis Miguel Bergasa, and Eduardo Romera. Are you able to perform a life-long visual topological localization? *Autonomous Robots*, 42:665–685, 2018. 1

[3] Hernan Badino, Daniel Huber, and Takeo Kanade. The CMU visual localization data set. http://3dvis.ri.cmu.edu/data-sets/localization, 2011. 5

[4] Eric Brachmann and Carsten Rother. Learning less is more – 6D camera localization via 3D surface regression. In *CVPR*, 2018. 2

[5] Federico Camposeco, Andrea Cohen, Marc Pollefeys, and Torsten Sattler. Hybrid camera pose estimation. In *CVPR*, 2018. 2

[6] David M. Chen, Georges Baatz, Kevin Köser, Sam S. Tsai, Ramakrishna Vedantham, Timo Pylvänäinen, Kimmo Roimela, Xin Chen, Jeff Bach, Marc Pollefeys, Bernd Girod, and Radek Grzeszczuk. City-scale landmark identification on mobile devices. In *CVPR*, 2011. 2

[7] Siddharth Choudhary and P. J. Narayanan. Visibility probability structure from SfM datasets and applications. In *ECCV*, 2012. 2

[8] Ondrej Chum and Jiri Matas. Matching with PROSAC – Progressive sample consensus. In *CVPR*, 2005. 2, 3, 5, 6, 7

[9] Winston Churchill and Paul Newman. Experience-based navigation for long-term localisation. *The International Journal of Robotics Research*, 32(14):1645–1661, 2013. 1, 3

[10] Mark Cummins and Paul Newman. FAB-MAP: Probabilistic localization and mapping in the space of appearance. *The International Journal of Robotics Research*, 27(6):647–665, 2008. 3

[11] Marcin Dymczyk, Simon Lynen, Titus Cieslewski, Michael Bosse, Roland Siegwart, and Paul Furgale. The gist of maps – summarizing experience for lifelong localization. In *ICRA*, 2015. 3, 6

[12] Marcin Dymczyk, Thomas Schneider, Igor Gilitschenski, Roland Siegwart, and Elena Stumm. Erasing bad memories: Agent-side summarization for long-term mapping. In *IROS*, 2016. 3

[13] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981. 3, 5, 6, 7, 8

[14] Xiao-Shan Gao, Xiao-Rong Hou, Jianliang Tang, and Hang-Fei Cheng. Complete solution classification for the perspective-three-point problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(8):930–943, 2003. 2, 5

[15] Google. ARCore SDK, 2020. https://developers.google.com/ar. 6

[16] Petr Gronát, Guillaume Obozinski, Josef Sivic, and Tomas Pajdla. Learning and calibrating per-location classifiers for visual place recognition. In *CVPR*, 2013. 3

[17] Richard I. Hartley, Jochen Trumpf, Yuchao Dai, and Hongdong Li. Rotation averaging. *International Journal of Computer Vision*, 103:267–305, 2012. 7

[18] Arnold Irschara, Christopher Zach, Jan-Michael Frahm, and Horst Bischof. From structure-from-motion point clouds to fast location recognition. In *CVPR*, 2009. 2, 3

[19] Alex Kendall and Roberto Cipolla. Modelling uncertainty in deep learning for camera relocalization. In *ICRA*, 2016. 2

[20] Alex Kendall, Matthew Grimes, and Roberto Cipolla. PoseNet: A convolutional network for real-time 6-DOF camera relocalization. In *ICCV*, 2015. 2

[21] Yunpeng Li, Noah Snavely, and Daniel P. Huttenlocher. Location recognition using prioritized feature matching. In *ECCV*, 2010. 2, 3

[22] Yunpeng Li, Noah Snavely, Dan Huttenlocher, and Pascal Fua. Worldwide pose estimation using 3D point clouds. In *ECCV*, 2012. 2, 3

[23] Hyon Lim, Sudipta N. Sinha, Michael F. Cohen, and Matthew Uyttendaele. Real-time image-based 6-DOF localization in large-scale environments. In *CVPR*, 2012. 3

[24] Liu Liu, Hongdong Li, and Yuchao Dai. Efficient global 2D–3D matching for camera localization in a large-scale 3D map. In *ICCV*, 2017. 2

[25] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. 2, 5, 6, 7

[26] Simon Lynen, Bernhard Zeisl, Dror Aiger, Michael Bosse, Joel Hesch, Marc Pollefeys, Roland Siegwart, and Torsten Sattler. Large-scale, real-time visual–inertial localization revisited. *The International Journal of Robotics Research*, 39 (9):1061–1084, 2020. 2, 3

[27] Will Maddern, Michael Milford, and Gordon Wyeth. Towards persistent localization and mapping with a continuous appearance-based topology. In *Robotics: Science and Systems*, 2013. 3

[28] Will Maddern, Geoff Pascoe, Chris Linegar, and Paul Newman. 1 Year, 1000km: The Oxford RobotCar Dataset. *The International Journal of Robotics Research*, 36(1):3–15, 2017. 5

[29] Daniela Massiceti, Alexander Krull, Eric Brachmann, Carsten Rother, and Philip H. S. Torr. Random forests versus neural networks — What's best for camera localization? In *ICRA*, 2017. 2

[30] Peter Mühlfellner, Mathias Bürki, Michael Bosse, Wojciech Derendarz, Roland Philippsen, and Paul Furgale. Summary maps for lifelong visual localization. *J. Field Robot.*, 33(5): 561–590, 2016. 3, 8

[31] Raúl Mur-Artal and Juan D. Tardós. ORB-SLAM2: an open-source SLAM system for monocular, stereo, and RGB-D cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017. 5

[32] Vito Santarcangelo, Giovanni Maria Farinella, and Sebastiano Battiato. Egocentric vision for visual market basket analysis. In *ECCV Workshops*, pages 518–531, 2016. 2

[33] Thorsten Sattler, Bastian Leibe, and Leif Kobbelt. Fast image-based localization using direct 2D-to-3D matching. In *ICCV*, 2011. 2

[34] Torsten Sattler, Tobias Weyand, Bastian Leibe, and Leif Kobbelt. Image retrieval for image-based localization revisited. In *BMVC*, 2012. 2

[35] Thorsten Sattler, Bastian Leibe, and Leif Kobbelt. Efficient effective prioritized matching for large-scale image-based localization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(9):1744–1756, 2017. 2

[36] Torsten Sattler, Akihiko Torii, Josef Sivic, Marc Pollefeys, Hajime Taira, Masatoshi Okutomi, and Tomas Pajdla. Are large-scale 3D models really necessary for accurate visual localization? In *CVPR*, 2017. 2

[37] Torsten Sattler, Will Maddern, Carl Toft, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, Fredrik Kahl, and Tomas Pajdla. Benchmarking 6DOF outdoor visual localization in changing conditions. In *CVPR*, 2018. 1, 2, 3, 5, 7

[38] Torsten Sattler, Qunjie Zhou, Marc Pollefeys, and Laura Leal-Taixe. Understanding the limitations of CNN-based absolute camera pose regression. In *CVPR*, 2019. 2

[39] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 5

[40] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in RGB-D images. In *CVPR*, 2013. 3

[41] Emiliano Spera, Antonino Furnari, Sebastiano Battiato, and Giovanni Maria Farinella. Egocentric shopping cart localization. In *ICPR*, 2018. 1, 2, 3

[42] Emiliano Spera, Antonino Furnari, Sebastiano Battiato, and Giovanni Maria Farinella. EgoCart: a benchmark dataset for large-scale indoor image-based localization in retail stores. *IEEE TCSVT*, 2020. 3

[43] Erik Stenborg, Carl Toft, and Lars Hammarstrand. Long-term visual localization using semantically segmented images. In *ICRA*, 2018. 3

[44] Abby Stylianou, Austin Abrams, and Robert Pless. Characterizing feature matching performance over long time periods. In *WACV*, pages 892–898, 2015. 3, 8

[45] Hajime Taira, Masatoshi Okutomi, Torsten Sattler, Mircea Cimpoi, Marc Pollefeys, Josef Sivic, Tomas Pajdla, and Akihiko Torii. InLoc: Indoor visual localization with dense matching and view synthesis. In *CVPR*, 2018. 1, 3

[46] Carl Toft, Erik Stenborg, Lars Hammarstrand, Lucas Brynte, Marc Pollefeys, Torsten Sattler, and Fredrik Kahl. Semantic match consistency for long-term visual localization. In *ECCV*, 2018. 2, 3, 5, 7

[47] Akihiko Torii, Relja Arandjelović, Josef Sivic, M. Okutomi, and Tomás Pajdla. 24/7 place recognition by view synthesis. In *CVPR*, 2015. 2

[48] Shenlong Wang, Sanja Fidler, and Raquel Urtasun. Lost shopping! Monocular localization in large indoor spaces. In *ICCV*, 2015. 3

[49] Jian Wu, Liwei Ma, and Xiaolin Hu. Delving deeper into convolutional neural networks for camera relocalization. In *ICRA*, 2017. 2

[50] Amir Roshan Zamir and Mubarak Shah. Accurate image localization based on Google maps street view. In *ECCV*, 2010. 2, 3

[51] Wei Zhang and Jana Kosecka. Image based localization in urban environments. In *3DPVT*, pages 33–40, 2006. 2