Article

The International Journal of Robotics Research
2015, Vol. 34(4-5) 476–492

DOI: 10.1177/0278364914561101
ijr.sagepub.com
SAGE

Real-time monocular image-based 6-DoF localization

Hyon Lim[1], Sudipta N. Sinha[2], Michael F. Cohen[2], Matt Uyttendaele[2] and H. Jin Kim[1]

Abstract

In this paper we present a new real-time image-based localization method for scenes that have been reconstructed offline using structure from motion. From input video, our method continuously computes six-degree-of-freedom camera pose estimates by efficiently tracking natural features and matching them to 3D points reconstructed by structure from motion. Our main contribution lies in efficiently interleaving a fast keypoint tracker that uses inexpensive binary feature descriptors with a new approach for direct 2D-to-3D matching. Our 2D-to-3D matching scheme avoids the need for online extraction of scale-invariant features. Instead, offline we construct an indexed database containing multiple DAISY descriptors per 3D point extracted at multiple scales. The key to the efficiency of our method is invoking DAISY descriptor extraction and matching sparingly during localization, and in distributing this computation over a temporal window of successive frames. This enables the system to run in real-time and achieve low per-frame latency over long durations. Our algorithm runs at over 30 Hz on a laptop and at 12 Hz on a low-power computer suitable for onboard computation on a mobile robot such as a micro-aerial vehicle. We have evaluated our method using ground truth and present results on several challenging indoor and outdoor sequences.

Keywords

Image-based localization, 2D-to-3D feature matching, structure from motion, pose estimation

1. Introduction

The problem of computing the position and orientation of a camera with respect to a geometric representation of the scene, which is referred to as *image-based localization*, is well studied in the computer vision and robotics communities. It has important applications in autonomous robot navigation (Royer et al., 2007; Achtelik et al., 2011; Meier et al., 2012), place recognition (Robertson and Cipolla, 2004; Schindler et al., 2007; Cummins and Newman, 2008; Milford, 2013) and augmented reality (Klein and Murray, 2007; Dong et al., 2009; Wagner et al., 2010). Broadly speaking, there are two types of approaches to image-based localization. The first set of methods addresses the problem of simultaneous localization and mapping (SLAM), where the camera is localized within an unknown scene. In contrast, approaches in the second category exploit prior knowledge of a map or 3D scene model that is constructed offline. Several recent methods fall within the second category (Irschara et al., 2009; Li et al., 2010, 2012; Sattler et al., 2011, 2012; Wendel et al., 2011, 2012), and this renewed interest has been sparked by progress in structure from motion (SfM) (Snavely et al., 2008; Jeong et al., 2012), which nowadays makes it possible to easily reconstruct large scenes in great detail.

Despite the fact that some of the recent approaches are scalable, real-time image-based localization in large environments remains a challenging problem. As the scene becomes larger, recognizing unique identifiable landmarks becomes much more challenging. Irschara et al. (2009), Li et al. (2010) and Sattler et al. (2011) overcame this difficulty by using scale-invariant DoG keypoint features described with scale-invariant feature transform (SIFT) descriptors (Lowe, 2004). However, these features are too expensive to extract in real-time. On the other hand, several real-time visual SLAM (Klein and Murray, 2007; Castle et al., 2011; Castle and Murray, 2011; Davison et al., 2007;

[1]Department of Mechanical and Aerospace Engineering, Seoul National University, Seoul, Korea
[2]Microsoft Research, Redmond, WA, USA

Corresponding author:
H. Jin Kim, Department of Mechanical and Aerospace Engineering, Seoul National University, 599 Gwanangno, Seoul 151-742, Korea.
Email: hjinkim@snu.ac.kr

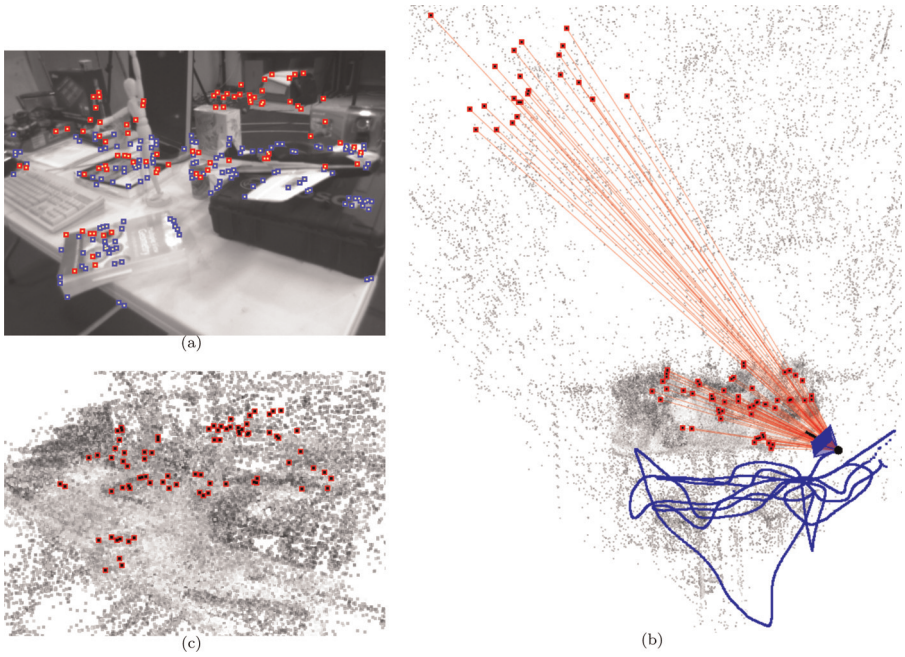


Fig. 1. Real-time localization of a quadrotor MAV in flight. (a) An input frame shown overlaid with tracked 2D points (in blue) and tracked 3D points in the map (in red). (b) The estimated flight path of the camera on the MAV. (c) The synthesized point cloud: view rendered using the estimated pose. Matched 3D points are shown (in red).

Williams et al., 2007) systems have been proposed. However, their performance degrades in larger scenes, where map maintenance becomes progressively expensive. These techniques are also less robust to fast and sudden camera motion, which makes them less attractive for situations where a precise camera pose is required persistently over long durations.

The topic of image-based localization has recently gained importance for autonomous aerial navigation, especially for GPS-denied areas (Achtelik et al., 2011; Weiss et al., 2011). It is particularly attractive for micro-aerial vehicles (MAV), such as quadrotors which are equipped with a camera and an onboard computer (Lim et al., 2012a; Meier et al., 2012). However, previous approaches (Irschara et al., 2009; Li et al., 2010; Sattler et al., 2011; Wendel et al., 2011; Li et al., 2012; Sattler et al., 2012; Wendel et al., 2012) are not fast enough for autonomous navigation on such platforms.

In this paper, we propose a new approach for continuously localizing a camera within a large environments, which have already been reconstructed using SfM. Our algorithm is real-time and runs over long periods with no fluctuation in the frame-rate. At its core lies a fast 2D keypoint tracker. Keypoints (Harris corners) from one frame are tracked in the following frame by matching them to candidate keypoints within a local search neighborhood (Ta et al., 2009) in the next frame. Binary feature descriptors (BRIEF) that are inexpensive to compute (Calonder et al., 2010) are then used to find the best frame-to-frame match. This fast tracker is interleaved with a new, efficient approach to find corresponding 3D points in the SfM reconstruction to the tracked 2D keypoints. These 2D–3D correspondences are then used to robustly estimate the camera pose in each frame. For recovering feature correspondences, we use DAISY feature descriptors (Tola et al., 2008; Winder et al., 2009) that are more accurate than binary descriptors but somewhat expensive to compute, and a kd-tree index built on these descriptors. This approach is related to recent work on *direct 2D-to-3D matching* (Sattler et al., 2011). However, in contrast to their work which focuses on localizing single images, we address the problem of continuous localization from a video stream over long durations and propose ideas to exploit spatio-temporal coherence.

We are able to achieve real-time performance without extracting scale-invariant keypoints at runtime. This is a key distinction from prior approaches (Dong et al., 2009; Irschara et al., 2009; Li et al., 2010; Sattler et al., 2011), which rely on the scale-invariance of SIFT features (Lowe, 2004). However, matching features across different scales is important for reliable 2D-to-3D matching. We address this requirement by computing redundant descriptors offline. For each 3D point in the map, we extract descriptors at multiple scales from multiple images and store indices with the descriptors to indicate the cameras in the map they were extracted from. This enables us to efficiently perform *place recognition*, where we prune false 2D–3D matches prior to running geometric verification during pose estimation. Other approaches typically achieve this using a vocabulary tree (Nister and Stewenius, 2006). Similar images are first retrieved (Dong et al., 2009; Irschara et al., 2009), after which 2D–3D matches are recovered indirectly from

pairwise feature matches. However, this has an overhead of quantizing descriptors and matching image pairs which we avoid in our approach.

Our feature matcher can also be used for localizing a single image from scratch. This is vital for localizing the camera in the first frame and for efficient relocalization, when the camera is lost. However, at other times when keypoint tracking is successful, we adopt a much more efficient *guided matching* approach for 2D-to-3D matching, related to strategies used in SLAM (Davison et al., 2007; Klein and Murray, 2007) and active matching (Handa et al., 2010). Unlike traditional robust feature matching (Lowe, 2004; Skrypnyk and Lowe, 2004), where ambiguous matches are usually pruned using a ratio test (Lowe, 2004; Irschara et al., 2009; Sattler et al., 2011), we recover multiple (one-to-many) 2D–3D match hypotheses for each tracked keypoint, and prune outliers later, during robust pose estimation. We also propose modifications to guided matching to distribute the computation over temporal windows of successive frames. By avoiding to compute many descriptors and kd-tree queries all at once, large fluctuations in the per-frame processing time are avoided. Lower per-frame latency allows keypoints with known 3D point correspondences to be tracked longer. Higher efficiency in tracking thus amortizes the cost of the relatively more expensive feature matching task, by requiring the matcher to be invoked less frequently during localization.

The paper is organized as follows. We discuss related work in Section 2 and introduce the key elements of our technique in Section 3. The offline and online stages of our method are described in Sections 4 and 5, respectively. Experimental evaluation is presented in Section 6, followed by concluding remarks.

This paper is an extended version of our previous work (Lim et al., 2012b) and contains an extensive set of new experiments that involve ground truth validation obtained using a Vicon motion capture system. These new results include:

- a quantitative evaluation of our method's accuracy on six new flight sequences captured from a quadrotor MAV;
- accuracy comparisons between our method and a modern SfM pipeline, the best known visual localization technique (but non real-time);
- rigorous evaluation of our method's robustness to various degrees of change in the scene geometry;
- extension of our framework for semantic localization tasks.

2. Related work

A number of existing works in image-based localization have adopted an image-based retrieval approach to the problem, and used it for urban scene navigation (Robertson and Cipolla, 2004) and city-scale location recognition (Schindler et al., 2007). However, these approaches often recover a coarse location estimate or may not even compute a complete six-degree-of-freedom (6-DoF) pose. Existing work on markerless augmented reality (Koch et al., 2005; Comport et al., 2006) addresses real-time 3D camera tracking but typically only with respect to specific objects, and that often requires a CAD model of the object.

Approaches for 3D camera tracking using visual landmark recognition (Skrypnyk and Lowe, 2004) based on SIFT features (Lowe, 2004) was proposed for *global localization* and used for robot navigation (Se et al., 2005). However, the need for repeated pairwise image matching in these approaches makes them too slow for real-time systems. Efficient keypoint recognition with randomized trees (Lepetit and Fua, 2006) and random ferns (Ozuysal et al., 2010) have enabled real-time camera tracking, but their significant memory requirements have limited their use beyond relatively small scenes.

SIFT features (Lowe, 2004) are also used in recent work (Irschara et al., 2009; Li et al., 2010; Sattler et al., 2011) on location recognition, where SfM is used to estimate 3D coordinates for the landmarks. The approaches scale well and some variants use the GPU (Irschara et al., 2009). However, these methods address the single-image localization problem, and are not fast enough for real-time localization from video.

Recently, a method for continuous localization was proposed for scenes reconstructed using SfM (Dong et al., 2009). It uses keyframe recognition repeatedly on video frames to indirectly recover 2D–3D matches. SIFT feature extraction is also the bottleneck in their method, which runs at 6 fps on a single thread and at 20 fps using parallel threads on four cores. Although our approach is related, it differs in the following ways: we explicitly track keypoints using binary descriptors (Calonder et al., 2010), to amortize the cost of feature matching over time, which is performed only as needed. Instead of keyframe-based matching, we use 2D-to-3D matching interleaved with tracking. This enables our system to exploit spatio-temporal coherence and reduce latency.

Real-time localization approaches for augmented reality on mobile devices have also been recently proposed (Arth et al., 2009; Wagner et al., 2010). However, these approaches derive their speedup from tracking relatively fewer features, making them less suitable for continuous 6-DoF localization in larger scenes or over longer durations. An efficient approach for tracking scale-invariant features in video was proposed by Ta et al. (2009), but it was used for object recognition, not real-time localization.

Visual SLAM systems have recently been used for real-time augmented reality (Davison et al., 2007; Castle and Murray, 2011), by utilizing parallel threads for tracking and mapping (PTAM) (Klein and Murray, 2007), using multiple local maps (Castle et al., 2011) and performing fast relocalization using random ferns (Williams et al., 2007). However, these approaches are susceptible to the problem of drift and error accumulation in the pose estimate, and

existing solutions for fast relocalization do not scale to larger scenes. PTAM (Klein and Murray, 2007) was recently used onboard a MAV for vision-based *position control*, i.e. hovering at a pre-specified location (Achtelik et al., 2011), and autonomous navigation was demonstrated for scenes with salient visual features (Blosch et al., 2010; Meier et al., 2012). However, these approaches focus more on the challenges of autonomous flight control, and so far have demonstrated vision-based localization either in small areas or within controlled scenes.

RGB-D camera-based visual SLAM has been proposed recently (Huang et al., 2011; Endres et al., 2012; Henry et al., 2012). Using depth from sensors such as Kinect, these approaches enable accurate mapping of indoor scenes. However, accurate localization in large scenes and outdoors with map built by SLAM, is not fully addressed.

3. Key elements of the proposed approach

We now discuss our scene representation and the building blocks of the new 2D-to-3D matching approach. The offline and online stages of our algorithm are described in Sections 3.6 and 4, respectively. Real-time localization requires efficient 2D-to-3D matching in two specific scenarios. First, for initializing localization or relocalization, the camera pose must be efficiently computed from a single image from scratch (Dong et al., 2009; Irschara et al., 2009; Li et al., 2010; Sattler et al., 2011). We call this *global matching*, which is challenging to achieve in real-time because the complete map must be searched. However, for intermediate video frames, a pose estimate computed from tracked 3D points in the previous frame is used to accelerate the search for 2D-to-3D matches in the current frame; we call this *guided matching*.

3.1. Scene representation

Our representation consists of a 3D scene reconstruction in a global coordinate system, which is computed using incremental SfM (Snavely et al., 2008) and bundle adjustment (Jeong et al., 2012) on an input sequence. This consists of the calibrated images, a 3D point cloud and a set of 2D–3D matches that encode the viewpoints a particular 3D point was triangulated from, during SfM. The calibrated images are used to build a database of feature descriptors for the 3D points, and a kd-tree index is constructed for the descriptors to support efficient approximate nearest neighbor (ANN) queries during feature matching. We extract keypoints using the Harris corner detector at multiple scales and compute DAISY descriptors (Tola et al., 2008), in particular T2-8a-2r6s descriptors (Winder et al., 2009) for each keypoint. Using principal component analysis (PCA) the descriptor dimension is reduced to 32. Descriptors in the database are labeled with their image indices, and the mapping between descriptors and corresponding 3D points is saved in a lookup table. This makes retrieving 3D points corresponding to the descriptors in the database very efficient. We further optimize the point retrieval by grouping cameras into overlapping clusters and use them for *place recognition*, as described later in Section 3.3.

3.2. Multi-scale features

To avoid extracting scale invariant keypoints (e.g. DoG (Lowe, 2004)) during online localization, offline we store in a database multiple descriptors for each 3D point corresponding to keypoints detected at multiple scales. First, multi-scale Gaussian image pyramids are computed and Harris corners are extracted in all levels of the pyramid. Orientation of each keypoint is computed by finding a peak in the gradient orientation histogram (Lowe, 2004), and a rotation invariant T2-8a-2r6s-32d DAISY descriptor (Winder et al., 2009) is computed from a resampled patch.

The 3D points in the map are then projected into the images they were triangulated from, during SfM. For each keypoint in a particular pyramid level of an image, the closest point amongst all 2D projections of the 3D points corresponding to that image is computed. If the closest point is within a threshold of τ pixels[1] (we set $\tau = 2$ pixels), that keypoint and its descriptor are assigned to the corresponding 3D point. This computation is repeated for each image pyramid to generate all the descriptors for every 3D point in the map.

Having multiple descriptors, as described above, has an associated overhead in storage. However, the redundancy in this representation allows keypoints extracted at a fixed scale during online localization to be matched to a set of descriptors in the database, as long as one of the descriptors in this set was extracted at a similar scale. Using multiple descriptors per 3D point is advantageous for ANN queries during feature matching for reasons pointed out by Boiman et al. (2008), where using multiple descriptors boosted the accuracy of a simple nearest neighbor classifier for image classification.

3.3. Place recognition

In large scenes, accurate global matching is often more difficult due to greater ambiguity in matching feature descriptors. A query descriptor in an image could match descriptors for several different 3D points, which have similar appearance. To address this, we perform coarse location recognition to filter as many incorrect 2D–3D matches as possible before running any geometric verification. As a result, fewer RANSAC (Fischler and Bolles, 1981) hypotheses are required during robust pose estimation, making that step computationally efficient.

For place recognition, we cluster nearby cameras during the offline stage into *location classes*, which are identified by solving an *overlapping view clustering* problem (Furukawa et al., 2010), where cameras with many SfM points in common are grouped into the same cluster (see Figure 2). We use an approach similar to that proposed by Furukawa et al. (2010), for clustering Internet image

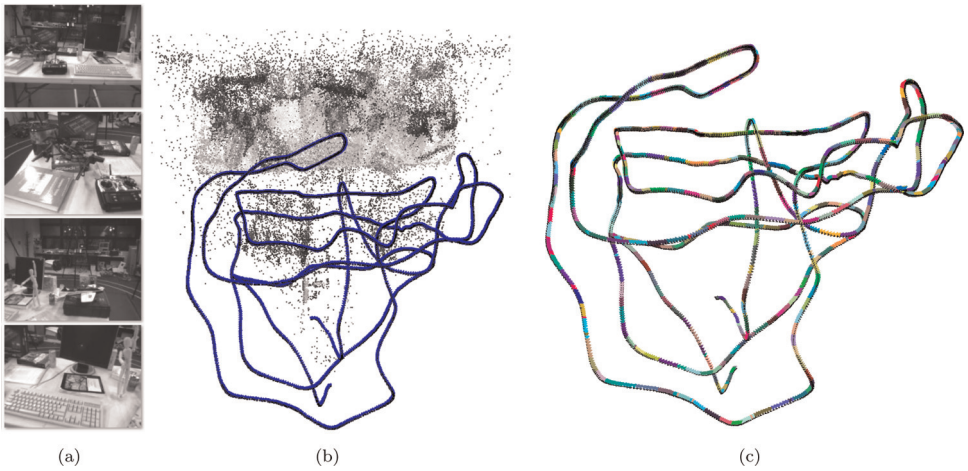


Fig. 2. (a) Four out of 2835 input frames sampled from a 30 Hz video sequence used to construct the map for the Vicon-Lab scene. (b) The SfM reconstruction and camera trajectory is shown (2835 cameras and 96,833 3D points). (c) The reconstructed cameras were grouped into 364 clusters that are shown in different colors in this visualization.

collections, which alternates between finding a disjoint partition of the cameras by analyzing the match graph from SfM, and growing the clusters locally by including cameras from neighboring clusters to improve the coverage of 3D points (Furukawa et al., 2010; Li et al., 2010). When localizing an image, the most likely location class is selected using a simple voting scheme over the set of matching descriptors returned by the ANN query on the image descriptors. Matched descriptors that correspond to 3D points that do not belong to the selected location are removed. This approach is non-parametric in contrast to some approaches that cluster features in pose space (Lowe, 2004), to achieve a similar goal. The global and guided matching methods are next described in more detail.

3.4. Global matching

Given 2D keypoints in an image and their corresponding DAISY descriptors denoted as $Q = \{q_i\}$, we seek to retrieve a set of 3D point correspondences for them. For each descriptor q_i, we perform a k-ANN query based on priority search using a kd-tree (Arya and Mount, 1993; Lowe, 2004) which retrieves nearest neighbors $D_i = \{d_{ij}\}$ sorted in increasing order of distance $\{s_{ij}\}$, $j = 0 \ldots k$, from q_i. For each neighbor d_{ij}, where $s_{ij} < \sigma s_{i0}$, the corresponding 3D point X_{ij} is retrieved, and every cluster that X_{ij} belongs to, receives a vote equal to its strength s_{i0}^2/s_{ij}. We find the highest score $\tilde{s}$ amongst the clusters, and select the clusters that have a score of at least $\beta\tilde{s}$. The set of images in the selected clusters is denoted as S. We set parameters $k = 50$, $\sigma = 2.0$ and $\beta = 0.8$.

The set of retrieved descriptors D_i is filtered by retaining descriptors corresponding to the selected database images in S. Next, for each query q_i, we compute a set of retrieved 3D points, where a matching strength for each 3D point is obtained by summing the strengths of its corresponding descriptors d_{ij}, which were computed earlier. Finally, two sets of matches are constructed. The first set contains the best 3D point match for each keypoint, where the best two matches based on matching strength, passed a ratio test with a threshold of 0.75 (Lowe, 2004). The ratio is expressed by $r = d_1/d_2$ where d_1 and d_2 denote the Euclidean distance between the query descriptor and the best and the second best matching descriptors respectively. The second set contains one-to-many 3D point matches; for each keypoint all of the matches with ratios greater than 0.75 are included. The two sets of matches are used for pose estimation. The first set is used for generating RANSAC hypotheses whereas the second set is used in the verification step.

3.5. Guided matching

During guided matching, other than the usual set of keypoints and query descriptors, we are also given an additional set of keypoints with known 3D point correspondences. This knowledge will be exploited to efficiently retrieve 2D–3D matches for the query set. Unlike global matching, where a voting scheme was used to narrow down the search to a few images, here, the scope is computed by inspecting the known 2D–3D correspondences. Concretely, we count the number of 3D points (from the known matches) visible in each image and then select the top 30 database images where some 3D points were visible. The k -ANN search for the query descriptors is now constrained to retrieve descriptors that belong to one of the selected images.

Although this check could have been enforced after the nearest neighbor search step, significant speedup is obtained by avoiding unnecessary distance computations during the backtracking stage of the kd-tree search. Thus, by checking the descriptor's image label, those that are out-of-scope can be rejected early. We take the descriptors returned by the nearest neighbor query and obtain 3D point

matches from them using the steps described in Section 3.4. The final matches are obtained after geometric verification is performed on the set of one-to-many 2D–3D match candidates using the camera pose estimate computed from the known matches.

3.6. Offline preprocessing

The offline steps of our algorithm are now summarized.

- The input images are processed using SfM as shown in Figure 2(a) and (b).
- The cameras are grouped into overlapping clusters as shown in Figure 2(c).
- 2D keypoints are extracted from Gaussian image pyramids and multiple DAISY descriptors are computed. In our implementation, we use pyramids with two octaves and four sub-octaves.
- A kd-tree is built for all the descriptors with image labels, and appropriate lookup tables are constructed.

During online localization, we currently assume that the feature database, the kd-tree index and the map is small enough to fit into main memory. However, for larger scenes, an out-of-core approach would be required. In such a case, it is possible to partition the map into multiple, overlapping sub-maps, such that each sub-map has its own descriptor database and kd-tree index. Online localization would then require only a small number of relevant sub-maps to be stored in memory at any time.

4. Real-time localization

In this section, we first describe our approach for 2D keypoint tracking in a video stream. We then discuss how guided matching is interleaved with keypoint tracking and finally describe pose estimation and filtering. Algorithm 1 summarizes the online algorithm for localizing frame f, given the map M, and a track table T, which is updated every frame. The table T stores the features tracks, feature descriptors, multiple 3D point match hypotheses and other attributes.

4.1. Keypoint tracking

To track 2D keypoints in video, we extract Harris corners in every frame. Next, for a $\mu \times \mu$ square patch around each keypoint, a 256-bit BRIEF descriptor (Calonder et al., 2010) is computed. The keypoints tracked in the prior frame are compared to all of the keypoint candidates in the current frame, within a $\rho \times \rho$ search window around their respective positions in the prior frame. BRIEF descriptors are compared using Hamming distance,[3] and the best candidate is accepted as a match, when the ratio between the distances to the best and second-best match is less than ψ. In our implementation, we set parameters as follows: $\mu = 32$, $\rho = 48$ and $\psi = 0.8$. In Algorithm 1, TRACK-2D performs keypoint tracking. When the feature count drops below κ_1 (= 25), new candidates are inserted to the track table using (ADD-GOOD-FEATURES) in regions of the image where 3D points are not being tracked in the current frame.

Algorithm 1. $P \leftarrow$ LOCALIZE-FRAME (f, T, M).

```
KALMAN-FILTER-PREDICT ()
P←∅
K←EXTRACT-KEYPOINTS (f)
η←TRACK-2D (f, K, T)
if η < κ1 then
   ADD-GOOD-FEATURES (f, K, T)
end if
C1←FETCH-2D-3D-MATCHES-FROM-TABLE (T)
if |C1| > κ2 then
   P←ESTIMATE-POSE (C1)
   if MATCHES-PENDING (T) then
      GUIDED-MATCHING (f, T, P, M)
   end if
else
   GLOBAL-MATCHING (f, T, M)
end if
C2←FETCH-2D-3D-MATCHES-FROM-TABLE (T)
if |C2| > κ2∧C1≠C2 then
   P←ESTIMATE-POSE (C2)
end if
KALMAN-FILTER-UPDATE (P)
return P
```

BRIEF descriptors lack rotational and scale invariance, but can be computed very fast. Using BRIEF, our method can track many more features than the Kanade–Lucas–Tomasi feature tracker (KLT) (Tomasi and Kanade, 1991), for a given computational budget. Keypoint extraction is the main bottleneck in our tracker. We have tried using FAST corners, but found Harris corners to be more repeatable. We do not prune the detected Harris corners using a non maximal suppression step, but instead, select all keypoint candidates that have a cornerness value greater than an adaptive threshold. The contrast-sensitive threshold is set to $\gamma\tilde{r}$ where $\tilde{r}$ is the maximum cornerness of keypoint candidates in the previous frame and $\gamma = 0.001$. We do not perform any geometric verification during keypoint tracking, but let the subsequent random sample consensus (RANSAC)-based pose estimation step deal with outliers.

4.2. Distributing matching computation

When new keypoints are added by calling the function ADD-GOOD-FEATURES, computing their DAISY descriptors and querying the kd-tree immediately will increase the computational overhead in those frames. However, these matches are not needed right away. Therefore, we distribute this computation over several successive frames, performing guided matching only on a small configurable batch of keypoints at a time (usually 100–150), until all pending keypoints have been processed (i.e. MATCHES-PENDING returns false). Our lazy evaluation strategy also reduces the

overall number of descriptors and queries computed. This is because the keypoint tracker often loses track of many features in the frame after new keypoints are inserted into the track table. Thus, by introducing a delay into feature descriptor extraction and matching tasks, we avoid redundant computation on several keypoints that have a short life-cycle at the 2D tracking stage. Once a 3D point match is found, it is saved in the track table and used as long as the keypoint is accurately tracked. When fewer than κ_2 (= 10) 2D–3D matches are available to the tracker, it relocalizes by calling GLOBAL-MATCHING.

4.3. Pose estimation and filtering

We now describe how the 6-DoF camera pose is robustly computed from the given 2D-to-3D matches. A standard RANSAC procedure is employed, based on hypotheses generated using the three-point pose estimation (Fischler and Bolles, 1981) method. The best estimate is retained along with the set of inliers, after which the pose parameters are refined using nonlinear least-squares optimization. A minimum inlier count threshold of 15 inliers is used to reject incorrect or uncertain pose estimates. Note that, in frames where only keypoint tracking is performed, a smaller number of RANSAC hypotheses is sufficient, since outliers can only be introduced during keypoint tracking and the percentage of outliers is typically quite small since the ratio test (involving BRIEF descriptors) directly eliminates most erroneous 2D–2D matches in consecutive frames.

In our system, discrete linear Kalman filters (Kalman, 1960) are employed to estimate a smooth camera trajectory based on 6-DoF pose estimates computed as described above. Similar to Meier et al. (2012), we use two discrete filters that have position and velocity, and orientation and angular velocity respectively as states. These two filters take position and orientation of the camera respectively as inputs while assuming a constant velocity and constant angular velocity model. We assume zero acceleration between successive frames of video (Davison et al., 2007). The estimated velocity and angular velocity can be used to predict the camera pose in future frames. Decoupling the translational and rotational dynamics is a reasonable choice due to the design of quadrotor MAVs as discussed by Meier et al. (2012). The simplified approach worked well in our evaluations, but for handling general motion involving hand-held cameras in arbitrary configurations, the method proposed by Davison et al. (2007) is more appropriate.

5. Application: Semantic localization

Semantic localization usually refers to the task where the robot must report its location semantically with respect to objects or regions in the scene rather than reporting 6-DoF pose or position coordinates. In prior work on semantic localization using contextual maps (Yi et al., 2009), only coarse location estimates could be recovered with respect to scene landmarks. Recently, visual SLAM (Klein and Murray, 2007) has been extended towards recognizing keyframes in real-time (Castle and Murray, 2011). Recognizing object categories in real-time is nowadays feasible with RGB-D sensors (Salas-Moreno et al., 2013), when the number of categories is small.

We now describe how semantic localization can be performed with simple extensions to our proposed technique. During offline mapping, an additional interactive stage is now required to annotate 3D points in the map with semantic labels. Labeling 3D points directly can be difficult since SfM point clouds can be difficult to interpret at high levels of detail. Therefore, we use an image-based interface similar to that proposed by Snavely et al. (2008) for annotating photo collections. In our system, the user browses the images that were used to reconstruct the map. The 3D points triangulated from a specific viewpoint are reprojected into that image. The user draws a bounding box on this image around the object of interest and labels it. The triangulated 2D keypoints within the bounding box are then selected and the label is transferred to the 3D points associated with these 2D keypoints.

During online localization, semantic information can be efficiently retrieved by recognizing labeled 3D points in the map using our technique. For each 3D point that is being successfully tracked, the corresponding labels are retrieved and the set of object instances are ranked in decreasing order based on the overall number of votes they receive. A threshold on the minimum number of votes is used to trigger detections in real-time. Apart from recognizing object instances in this fashion, the camera pose estimated by our method can also enable indirect semantic reasoning about the scene. For instance, the relative position of objects in the scene to the camera or distance to objects that are not yet visible can also be predicted.

6. Experimental results

We have evaluated our system in five different environments using 16 evaluation sequences. Relevant details about these datasets are summarized in Table 1. The image sequences and ground truth data is being made available publicly through the project website.[4] In this section, we first describe the experimental setup used for evaluation. The experimental results are then presented in three sections. A description of the multimedia extensions showing videos of our real-time system in action can be found in the Appendix.

First, we analyzed the absolute accuracy of our real-time localization method in the Vicon-Lab indoor scene, where we have acquired image sequences with accurate ground truth pose data. In two out of the six Vicon-Lab sequences used for evaluation, scene objects were intentionally removed to test the robustness of our method to changes in the scene. In this set of experiments, we also compared our method with the offline SfM system, which we consider

Table 1. Datasets. The physical scene size, the number of evaluation sequences, the number of cameras and 3D points reconstructed in the map, the number of indexed DAISY descriptors, the number of clusters and the in-memory footprint for each dataset are listed. The OUTDOOR1 and OUTDOOR2 sequences were made publicly available by Dong et al. (2009).

Map	Scene size (m)	Number of evaluation sets	Number of cameras	Number of 3D points	Number of descriptors	Number of clusters	Memory usage (MB)
VICON-LAB	7 × 4	6	2835	96,833	1,817,114	364	168
LAB	8 × 5	4	2111	76,560	1,019,253	450	124
HALL	30 × 12	4	2749	88,248	1,377,785	253	111
OUTDOOR1	36 × 24	1	1448	120,313	1,241,045	188	117
OUTDOOR2	26 × 20	1	1011	26,484	1,282,227	126	107

(a) EVAL-b (b) EVAL-SC-a (c) EVAL-SC-b (d) EVAL-SC-c

Fig. 3. A cluttered table in the Vicon-Lab scene, seen in input frames from four evaluation sequences. Objects on the cluttered table were progressively removed as sequences EVAL-SC-a and EVAL-SC-b were acquired and only the monitor, keyboard and a couple of boxes remain when the sequence EVAL-SC-c was captured.

here as the state-of-the-art image-based localization method that is not real-time.

In the second set of experiments, we tested the performance of our system on eight sequences captured in larger indoor scenes, LAB and HALL, focusing on computational efficiency and robustness for large scenes and long sequences. Here, we also demonstrate the effectiveness of our approach for the application of semantic localization. Finally, we present results on the OUTDOORS1 and OUTDOORS2 sequences made publicly available by Dong et al. (2009), where we show that outdoor scenes are also well handled by our system which is similar to offline SfM in terms of accuracy but is computationally much more efficient.

6.1. Experimental setup

We first describe the data acquisition phase for the Vicon-Lab experiments. A Point Grey Firefly MV camera with a Computar T2616FICS lens was used to capture images at 640 × 480 pixel resolution at 30 Hz. The horizontal and vertical field of view of the lens is 99.6° and 75.4°, respectively. The camera was mounted front-facing on a quadrotor MAV, equipped with an off-the-shelf inertial measurement unit (PX4-IMU) and a small computer board (Compulab, 2011). Using our custom acquisition software, we ensured that all captured images were temporally synchronized with the inertial measurement unit (IMU) data. A Vicon motion capture system was used to track external markers on the camera at 100 Hz. Figure 4 shows a rough schematic diagram of the hardware components that were involved.

In all, seven Vicon-Lab sequences were captured with complete ground truth (see Table 2 for details). The first sequence was captured from the hand-carried MAV and was used to construct the map needed by our method. Subsequently, six sequences were captured in succession with the MAV in flight. While acquiring the first three sequences, denoted as EVAL-a, b and c in Table 2, the scene did not change much. However, in the three subsequent sequences, EVAL-SC-a, b and c, objects were intentionally removed to test the robustness of our method to changes in scene geometry. A few example frames are shown in Figure 3.

The eight evaluation sequences from the LAB and HALL scenes were also captured at 640 × 480 resolution using a Point Grey Firefly MV camera. Six of these were captured from a hand-carried camera and remaining two of these were captured using a quadrotor MAV in flight. The two outdoor sequences from Dong et al. (2009) were also captured with hand-held, consumer cameras. Table 3 provides the relevant details.

6.2. Experiments on Vicon-Lab

In order to perform ground truth evaluation, we first align our map to ground truth coordinate system by registering the camera poses in the map to the ground truth positions using the method of Horn (1987). This transformation is

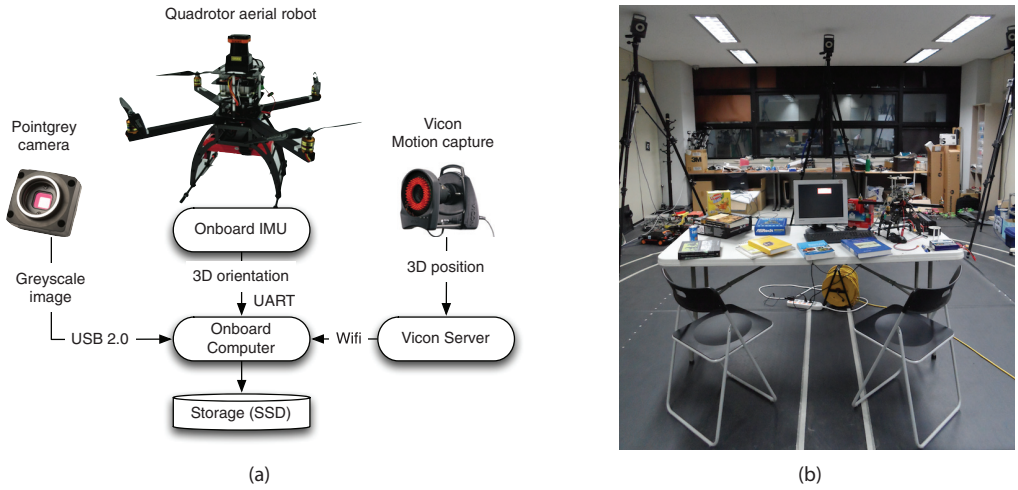


Fig. 4. Schematic diagram showing hardware components in our acquisition setup and an image of the Vicon-Lab scene: (a) dataset acquisition system; (b) example environment configuration.

Table 2. Ground truth evaluation (Vicon-Lab). For each of the six evaluation sequences, columns 2–5 list the number of frames (F), the number of frames successfully localized (F_{loc}), the percentage of frames successfully localized (P_{loc}) and the per-frame running timing (T_{loc}) (mean ± SD) of our method. Columns 6–8 report the number of frames in which feature matching computation was performed (F_{match}), the corresponding percentage (P_{match}) and the per-frame timing (T_{match}) for those selected frames. Finally, the position and orientation error (mean ± SD) computed using ground truth data is reported in the two rightmost columns. Refer to Section 6.2 for the details.

Sequence	Number of frames (F)	F_{loc}	P_{loc}	T_{loc} (ms)	F_{match}	P_{match}	T_{match} (ms)	Position error (cm)	Rotation error (degrees)
EVAL-a	608	579	95%	25 ± 7	108	18%	35 ± 10	10.8 ± 4.7	1.6 ± 0.7
EVAL-b	3532	3154	89%	26 ± 6	716	41%	34 ± 6	5.6 ± 6.2	2.7 ± 1.3
EVAL-c	1989	1868	94%	25 ± 7	445	22%	35 ± 9	9.4 ± 2.9	1.7 ± 0.8
EVAL-SC-a	2787	2425	87%	24 ± 5	713	29%	30 ± 8	7.6 ± 5.7	4.0 ± 3.9
EVAL-SC-b	1940	1649	85%	24 ± 7	472	24%	33 ± 8	7.6 ± 3.3	1.7 ± 1.0
EVAL-SC-c	1661	1143	69%	23 ± 4	623	38%	27 ± 5	9.4 ± 6.7	1.7 ± 0.7

Table 3. Localization statistics. For each of the evaluation sequences in the LAB and HALL scenes and for the outdoor sequences, columns 3–6 list the number of frames (F), the number of frames successfully localized (F_{loc}), the percentage of frames successfully localized (P_{loc}) and the per-frame running timing (T_{loc}) (mean ± SD) of our method. Columns 7–10 report the number of frames in which feature matching computation was performed (F_{match}), the corresponding percentage (P_{match}) and the per-frame timing (T_{match}) for those selected frames.

Map	Sequence name	Number of frames (F)	F_{loc}	P_{loc}	T_{loc} (ms)	F_{match}	P_{match}	T_{match} (ms)
LAB	WALK1	237	237	100%	19 ± 4	10	4%	27 ± 11
LAB	WALK2	3793	3790	99.9%	18 ± 3	210	6%	23 ± 6
LAB	FLIGHT1	1000	1000	100%	17 ± 2	34	3%	22 ± 5
LAB	FLIGHT2	1210	1204	99.5%	17 ± 3	47	4%	23 ± 7
HALL	WALK1	475	475	100%	17 ± 3	27	6%	20 ± 7
HALL	WALK2	713	712	100%	17 ± 2	30	4%	20 ± 4
HALL	WALK3	540	540	100%	16 ± 1	33	6%	18 ± 3
HALL	WALK4	201	201	100%	16 ± 8	4	2%	24 ± 7
OUTDOOR1		1033	1033	100%	21 ± 4	169	16%	25 ± 6
OUTDOOR2		605	603	100%	27 ± 10	115	19%	40 ± 15

applied to each evaluation sequence to compute position and orientation errors with respect to the ground truth. Here, errors are only computed on frames that were successfully localized by our method (F_{loc} in Table 2). In our evaluation, pose estimate is considered to be successful when the inlier count during 6-DoF pose estimation

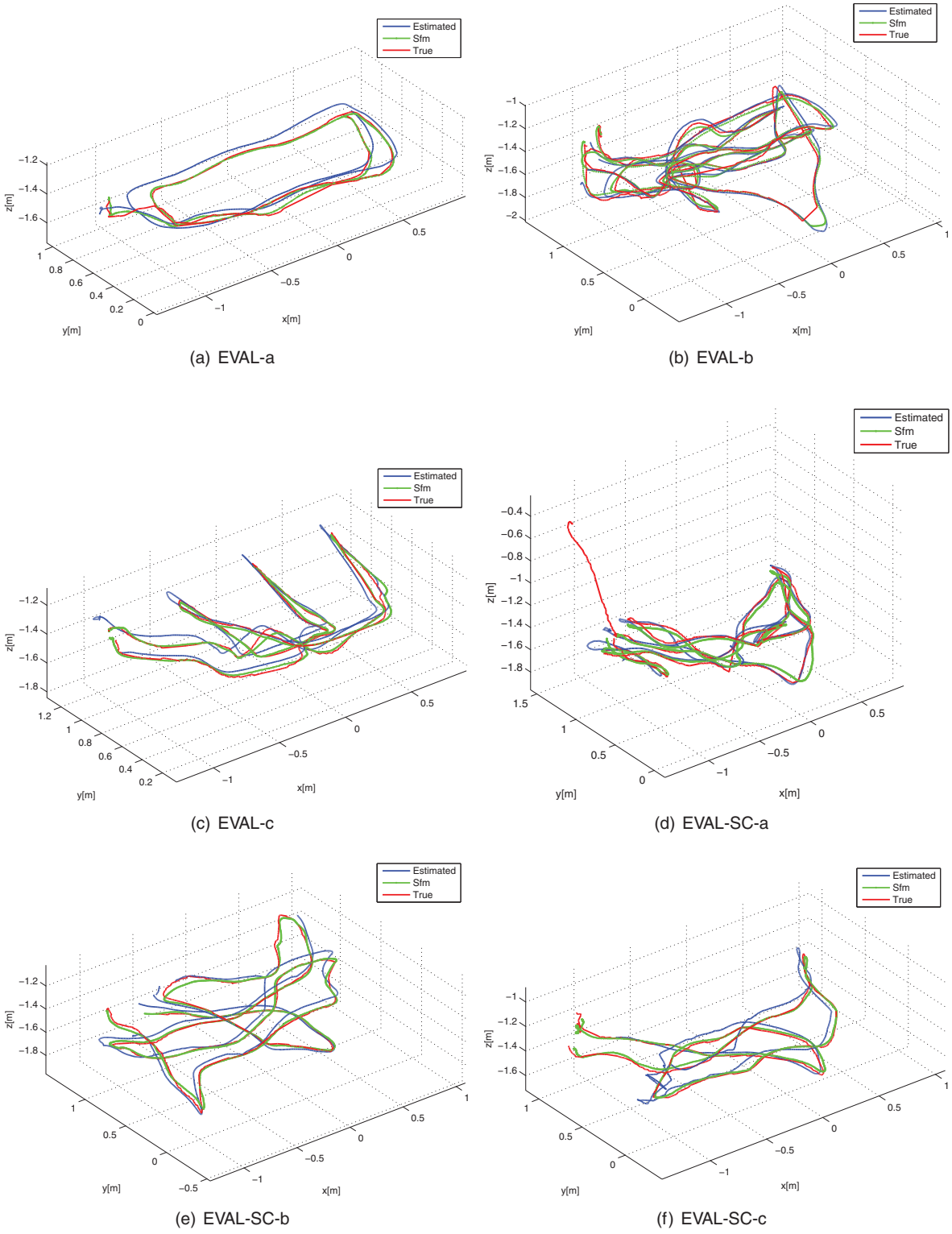


Fig. 5. Flight paths (Vicon-Lab sequences). For each sequence, the flight paths computed using proposed method and using offline SfM are visualized along with the ground truth trajectories. Please refer to Section 6.2 and Table 2 for the details.

exceeds the minimum inlier threshold which is set to 15 in all our experiments. We also processed each evaluation sequence independently with our offline SfM pipeline and aligned the estimated camera poses in these SfM reconstructions to ground truth camera poses. Figure 5 shows 3D visualizations of the flight paths recovered for the six sequences using our method as well as using SfM.

Table 2 summarizes the results of the ground truth evaluation on the Vicon-Lab sequences. The percentage of images for which a 6-DoF pose estimate was successfully computed ranged from 89% to 95% for the EVAL-a, b and c sequences respectively. The mean position and orientation errors ranged between 5.6 and 10.8 cm and 1.6° and 2.7° respectively. On the EVAL-SC-a, b and c sequences, where scene geometry was intentionally modified, the success rate of our method fell to 87%, 85% and 69%, respectively, but the mean position error and orientation errors were in the range of 7.6–9.4 cm and 1.7–4.0°, respectively, which is reasonably low. Figure 6 shows detailed plots for (EVAL-b), the longest evaluation sequence in this set. Specifically, Figure 6 (e,f) shows a comparison of the error histograms obtained using our method and offline SfM. The fact that these error distributions are quite similar and concentrated around zero reaffirms the overall high accuracy achieved using our method. See Multimedia Extension 1 for a video of our system in action.

6.3. Experiments on indoor datasets

We next evaluated our method in the LAB scene. In this case, the reconstructed map had a higher degree of completeness and even small objects were well represented in the 3D point cloud. The localization success rates for the four sequences that were tested- WALK1, WALK2, FLIGHT1 and FLIGHT2, were quite high (99.5% – 100%) and further details about these experiments are reported in Table 3. Extensions 2 and 3 show real-time screen capture of our system running on these sequences.

To analyze accuracy, we compared our pose estimates on LAB-WALK1 with the offline SfM reconstruction (obtained from the same images) by treating it as a substitute for ground truth. This is reasonable since our SfM pipeline was shown to be quite accurate in the ground truth evaluation (see Section 6.2). The map was scaled to real-world dimensions using known distances between scene landmarks. Figure 7 plots the position error for the LAB-WALK1 sequence along with other relevant statistics. The mean position and orientation errors were 5.1 cm and 1.7°, respectively.

The evaluation on the 2-minute-long WALK2 sequence highlights the ability of our system to continuously localize the camera in all sections of the room which is 8 m $\times$ 5 m in size. The effectiveness of our method can be seen in Extension 3 which also demonstrates the robustness of proposed method to large viewpoint and lighting change. We also used this sequence to demonstrate the application of semantic localization. For this task, 28 scene objects in the LAB scene were annotated offline as described in Section 5. The accuracy and robustness of the instance recognition is best seen in the video extension. Figure 8 shows a few screenshots of the real-time system and relevant statistics involving online localization are plotted with respect to time on the complete sequence.

We also tested our method on the larger HALL scene which contains a narrow corridor, multiple doors, several textureless walls and multiple chairs with similar appearance. The relatively fewer discriminative visual features in this scene makes 2D-to-3D matching more challenging than in the other indoor scenes. Figure 9(a) shows a challenging input frame where the camera is moving from one room into another (see also Extension 4). All frames from all the four test sequences were successfully localized using our approach at frame-rates exceeding 30 Hz (see Table 3). Figure 9(b) shows a visualization of the estimated camera trajectories overlaid in a top-view of the map.

6.4. Experiments on outdoor datasets

Figure 10 shows the camera trajectories estimated by our method and those obtained using offline SfM on the two OUTDOOR sequences (see also Extension 5). The high accuracy of our method can be qualitatively judged from the fact that the two trajectories appear to be well aligned in both cases.

Since these sequences do not have ground truth data, we analyze the accuracy of our pose estimates by quantitatively comparing them with the results obtained using offline SfM, after the coordinate systems were aligned using the same method mentioned in Section 6.2. Using coarse estimates of scene dimensions to approximately scale the camera trajectories to their true dimensions, we estimated the mean position error in both sequences to be less than 15 cm. The mean orientation error, which does not depend on the knowledge of scene dimensions was found to be less than 1.6°. It is worth noting that according to Dong et al. (2009), the system based on PTAM (Klein and Murray, 2007), failed on both these sequences due to fast camera motion, even when it was configured to run at 5 frames per second which is considerably slower than real-time.

6.5. Timings

A single-threaded C++ implementation of our algorithm runs at an average frame-rate exceeding 30 Hz on all our datasets, on a laptop with an Intel Core 2 Duo 2.66 GHz processor running Windows 7. The mean timings per frame varied from 23 to 37 ms on all our evaluation sequences as reported in Tables 2 and 3. For frames, where either global or guided matching computation was invoked, the mean timings varied between 27 and 40 ms. Figures 7 and 8 provide some insight into how our efficient 2D-to-3D matching approach lowers the processing overhead for frames in which guided matching computation must be performed. Specifically, Figure 8(i) shows that for the LAB-WALK2 sequence that was over 2 minutes long, our system took 75 ms in the first frame, 40–55 ms for relocalization when it lost track, and the average processing time was 18 ms.

In comparison, the method proposed by Dong et al. (2009) was reported to run at 6 Hz on a single core and

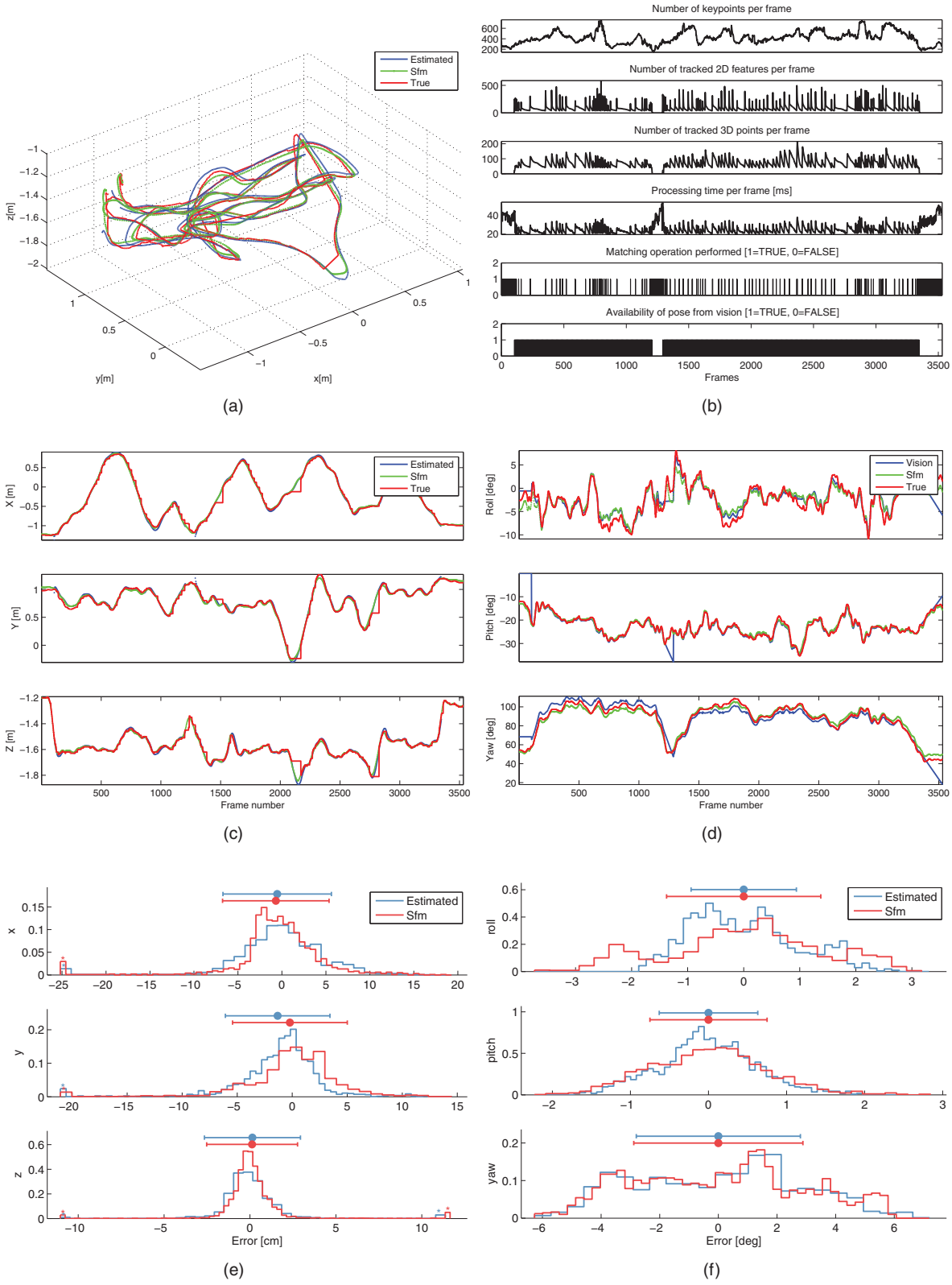


Fig. 6. Results on Vicon-Lab EVAL-b. (a) 3D visualization of camera trajectories (our method in blue, offline SfM in green and ground truth in red). (b) Plots (top to bottom) show various relevant statistics related to the performance of our localization approach on the full sequence. (c, d) Quantitative evaluation of position coordinates (x, y, z) and orientations (roll, pitch and yaw), respectively, estimated by our method and offline SfM. (e, f) Distribution of position and orientation error, respectively, from the two methods.

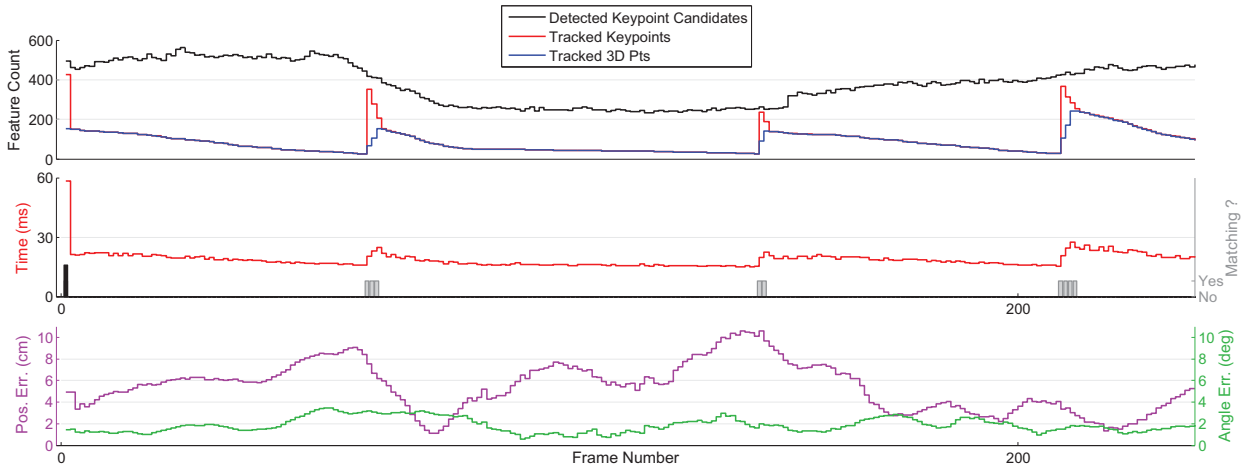


Fig. 7. LAB-WALK1 sequence (237 frames). Top: The number of detected keypoints, the number of tracked keypoints and the number of keypoints with 2D–3D matches are shown. Middle: The red curves shows the per-frame processing time. Frames where guided matching or global matching is computed are shown with gray and tall black bars (at the first frame) respectively. Bottom: The per-frame error in position (in centimeters) and orientation (in degrees). The evaluation methodology is explained in Section 6.3. The maximum position error (within an 8 m × 5 m room) occurs when relatively fewer 3D points are being tracked, but this error becomes smaller as the system starts tracking more 3D points.

20 Hz using four cores on a modern CPU. Based on our timings on the sequences they provide, our method appears to be at least fives times faster than their single-threaded implementation.

To test the feasibility of our method for onboard deployment on a quadrotor MAV, we ran our implementation on a Fit-PC2i (Compulab, 2011), an off-the-shelf compact computer that weighs less than 350 g and consumes only 10 W power at full load. It was equipped with an Intel Atom Z550 2 GHz CPU, 2 GB RAM and a 64 GB SSD drive. Nowadays, modern MAVs are often equipped with faster onboard computers (Meier et al., 2012) but the Fit-PC2i configuration seemed a good match for most modest onboard computers that are in use these days.

Our algorithm runs at about 12 Hz on the Fit-PC2i. It was tested on the LAB-FLIGHT1 and LAB-FLIGHT2 sequences. The success rate of localization on these two sequences was 100% and 98.5%, respectively. Although our experiments on the Fit-PC2i were simulated on pre-captured 30 Hz video, we simulated a 12 Hz input stream by dropped the requisite number of frames depending on the processing time of the current frame. The frame-rates we obtained are comparable with the 5–10 Hz onboard visual SLAM system (Klein and Murray, 2007) used for vision-based position control (Achtelik et al., 2011). These preliminary experiments indicate that if combined with an IMU, our method is viable for autonomous aerial navigation in areas larger and more complex than what has been tackled before (Blosch et al., 2010; Achtelik et al., 2011; Meier et al., 2012).

6.6. Failure cases

Although we have demonstrated that robust and accurate monocular image-based localization is feasible in real-time, our system currently cannot localize the camera when observing parts of the scene that are not well represented in the map. The low localization rate (69%) on the Vicon-Lab EVAL-SC-c sequence is primarily due to this reason. In this case, most of the objects on the cluttered table top have been removed and our system is often unable to find enough 2D-to-3D correspondences to reliably compute the camera pose. This limitation could be addressed by combining the system with an online mapping or visual SLAM module that will allow the system to better extrapolate beyond the region already represented well in the map or model regions that have changed significantly on the fly.

However, extending the map in real-time for large scenes creates several interesting challenges for online mapping. Specifically, one of the challenge will be computationally efficient online algorithms for indexing new feature descriptors. This will be required to enable accurate 2D-to-3D matching in regions of the map that are modeled on the fly. Extending the algorithm to model changes in the geometry and appearance of the scene is an important issue that must be tackled for practical deployment of image-based localization systems. These are clearly issues that need to be addressed in future work.

7. Conclusions

In this paper, we have proposed a new approach for real-time, image-based 6-DoF localization using a single camera, in scenes reconstructed in advance using a SfM technique. Our algorithm efficiently combines 2D keypoint tracking in video with direct 2D-to-3D matching, without the computational burden of having to extract scale-invariant features during online localization. Our method

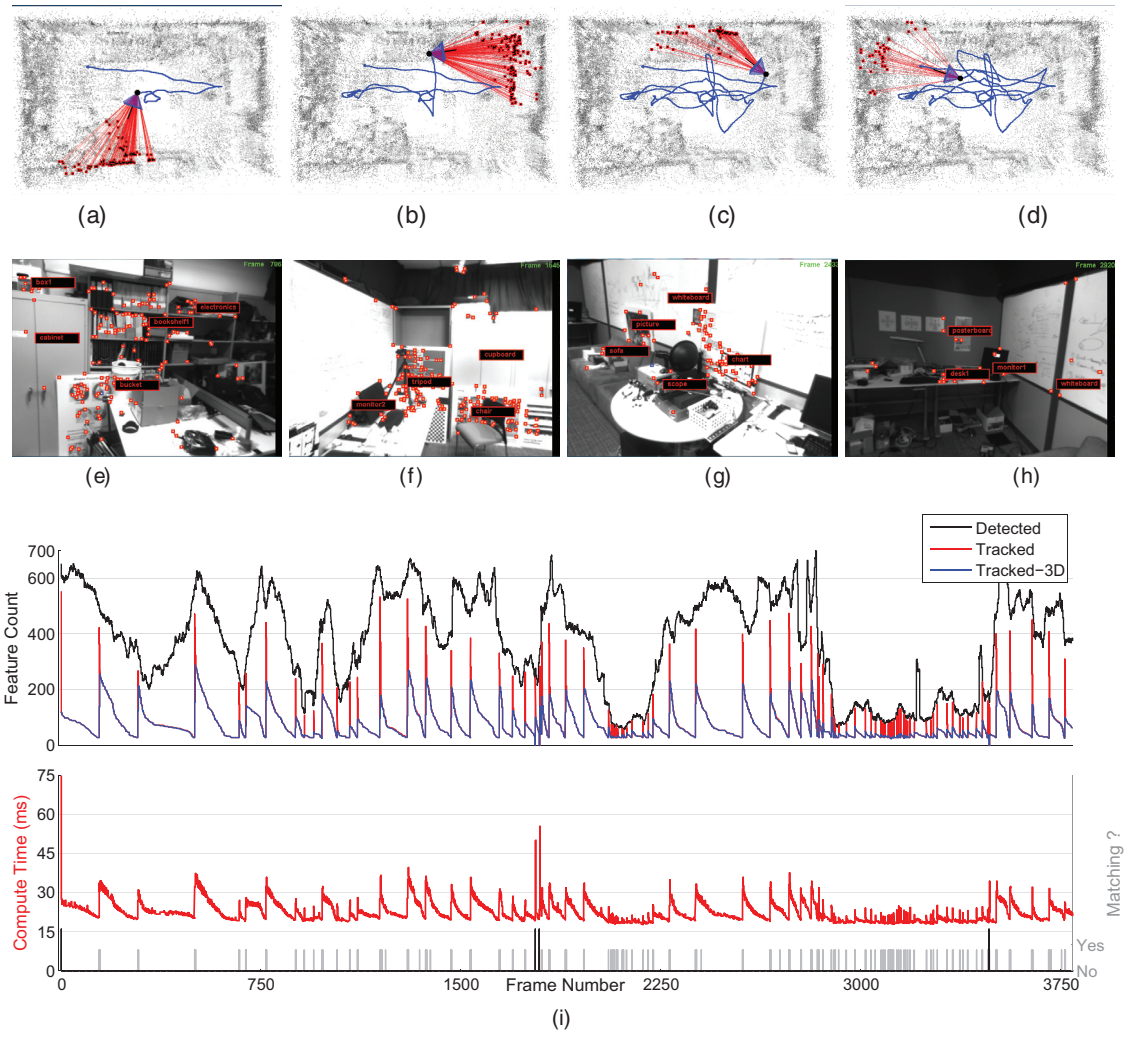


Fig. 8. Lab results (WALK2 sequence). (a–d) Camera pose estimated for four selected frames from a long hand-held sequence containing 3793 frames. The estimated trajectory (blue) is overlaid on a top-view rendering of the map. (e–h) The corresponding input frames with tracked features and annotations of object instances recognized in these frames. See multimedia Extension 3 for the system in operation. (i) The top plot shows the number of candidate keypoints per frame, the number of tracked 2D keypoints and the number of tracked 3D points in the map. The bottom plot displays per-frame timings (in milliseconds) and the short–gray and tall–black bars indicate the frames in which guided matching and global matching computation was performed. Tracking was lost twice around f_{1800} and once around f_{3500} requiring relocalization. In spite of the lower tracking efficiency between frames f_{2000}–f_{2200} and f_{2800}–f_{3500}, when relatively fewer keypoints were detected (in a dark corner of the room), the camera was consistently localized.

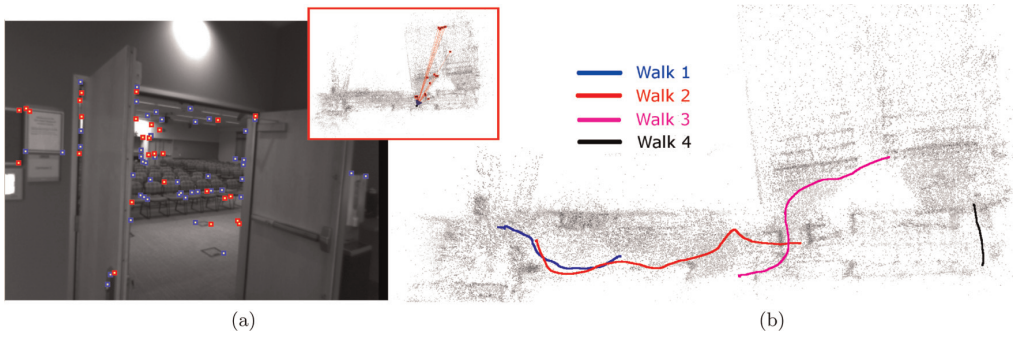
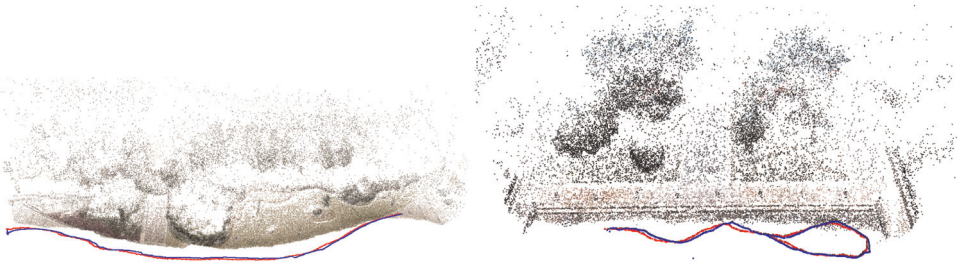


Fig. 9. HALL-WALK sequences. Left: A frame from the WALK3 sequence and the corresponding camera pose estimate shown in the top-view of the map (inset). Right: Computed trajectories of the camera in the corridor (WALK1–2), and entering the room through different doors (WALK3–4). Our approach worked well in this large-scale scene (approximately 30 m $\times$ 12 m) despite the presence of many ambiguous and confusing features. See Multimedia Extension 4 for further information.



(a) (b)

Fig. 10. Outdoor sequences from Dong et al. (2009): the trajectories estimated by our method (shown in blue) are qualitatively similar to those computed using offline SfM (shown in red). See Multimedia Extension 5 for further details.

exploits spatio-temporal coherence, invoking expensive feature matching computations infrequently and in a distributed fashion over a temporal window. Our implementation can process 640 × 480 video faster than video-rate on a modest laptop and an extensive quantitative evaluation suggests that the approach can be accurate in both indoor and outdoor settings. Our preliminary experiments have also shown that it can be feasible for onboard computation on a MAV and could have interesting applications in autonomous robot navigation and semantic localization.

Acknowledgements

Part of this work was undertaken while Hyon Lim was visiting Microsoft Research as a research intern.

Funding

This work was supported in part by the National Research Foundation of Korea (NRF) grant funded by the Korea Government (MEST) (grant number 2013013911).

Notes

1. The distance is computed in the appropriate pyramid level.
2. Each point could be present in multiple overlapping clusters.
3. Computed with bitwise XOR followed bit-counting using the parallel bit-count algorithm, see http://graphics.stanford.edu/~seander/bithacks.html.
4. See http://icsl.snu.ac.kr/~hyonlim/ijrr2014/

References

Achtelik M, Weiss S and Siegwart R (2011) Onboard IMU and monocular vision based control for MAVs in unknown in-and outdoor environments. In: *2011 IEEE international conference on robotics and automation (ICRA)*. IEEE, pp. 3056–3063.

Arth C, Wagner D, Klopschitz M, Irschara A and Schmalstieg D (2009) Wide area localization on mobile phones. In: *8th IEEE international symposium on mixed and augmented reality, 2009 (ISMAR 2009)*. IEEE, pp. 73–82.

Arya S and Mount DM (1993) Approximate nearest neighbor queries in fixed dimensions. In: *Proceedings of the fourth annual ACM-SIAM symposium on discrete algorithms*. Philadelphia, PA: Society for Industrial and Applied Mathematics, pp. 271–280.

Blosch M, Weiss S, Scaramuzza D and Siegwart R (2010) Vision based MAV navigation in unknown and unstructured environments. In: *2010 IEEE international conference on robotics and automation (ICRA)*. IEEE, pp. 21–28.

Boiman O, Shechtman E and Irani M (2008) In defense of nearest-neighbor based image classification. In: *IEEE conference on computer vision and pattern recognition, 2008 (CVPR 2008)*, pp. 1–8.

Calonder M, Lepetit V, Strecha C and Fua P (2010) Brief: Binary robust independent elementary features. In: *Computer Vision–ECCV* 2010. New York: Springer, pp. 778–792.

Castle R and Murray D (2011) Keyframe-based recognition and localization during video-rate parallel tracking and mapping. *Image and Vision Computing* 29(8): 524–532.

Castle RO, Klein G and Murray DW (2011) Wide-area augmented reality using camera tracking and mapping in multiple regions. *Computer Vision and Image Understanding* 115(6): 854–867.

Comport AI, Marchand E, Pressigout M and Chaumette F (2006) Real-time markerless tracking for augmented reality: the virtual visual servoing framework. *IEEE Transactions on Visualization and Computer Graphics* 12(4): 615–628.

Compulab (2011) fit-PC2i. http://www.fit-pc.com/web/fit-pc/fit-pc2-i/ (Accessed 7 August 2013).

Cummins M and Newman P (2008) FAB-MAP: Probabilistic localization and mapping in the space of appearance. *The International Journal of Robotics Research* 27(6): 647–665.

Davison AJ, Reid ID, Molton ND and Stasse O (2007) MonoSLAM: Real-time single camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29(6): 1052–1067.

Dong Z, Zhang G, Jia J and Bao H (2009) Keyframe-based real-time camera tracking. In: *2009 IEEE 12th international conference on computer vision*. IEEE, pp. 1538–1545.

Endres F, Hess J, Engelhard N, Sturm J, Cremers D and Burgard W (2012) An evaluation of the RGB-D slam system. In: *2012 IEEE international conference on robotics and automation (ICRA)*. IEEE, pp. 1691–1696.

Fischler MA and Bolles RC (1981) Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM* 24(6): 381–395.

Furukawa Y, Curless B, Seitz SM and Szeliski R (2010) Towards internet-scale multi-view stereo. In: *2010 IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 1434–1441.

Handa A, Chli M, Strasdat H and Davison AJ (2010) Scalable active matching. In: *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*.

Henry P, Krainin M, Herbst E, Ren X and Fox D (2012) RGB-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor environments. *The International Journal of Robotics Research* 31(5): 647–663.

Horn BKP (1987) Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A* 4(4): 629–642.

Huang AS, Bachrach A, Henry P, et al. (2011) Visual odometry and mapping for autonomous flight using an RGB-D camera. In: *International symposium on robotics research (ISRR)*, Flagstaff, AZ.

Irschara A, Zach C, Frahm JM and Bischof H (2009) From structure-from-motion point clouds to fast location recognition. In: *IEEE conference on computer vision and pattern recognition, 2009 (CVPR 2009)*. IEEE, pp. 2599–2606.

Jeong Y, Nister D, Steedly D, Szeliski R and Kweon IS (2012) Pushing the envelope of modern methods for bundle adjustment. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34(8): 1605–1617.

Kalman RE (1960) A new approach to linear filtering and prediction problems. *Transactions of the ASME—Journal of Basic Engineering* 82(Series D): 35–45.

Klein G and Murray D (2007) Parallel tracking and mapping for small AR workspaces. In: *6th IEEE and ACM international symposium on mixed and augmented reality, 2007 (ISMAR 2007)*. IEEE, pp. 225–234.

Koch R, Koeser K, Streckel B and Evers-Senne JF (2005) Markerless image-based 3D tracking for real-time augmented reality applications. In: *The 7th international workshop on image analysis for multimedia interactive services*, Montreux, Switzerland.

Lepetit V and Fua P (2006) Keypoint recognition using randomized trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28(9): 1465–1479.

Li Y, Snavely N, Huttenlocher D and Fua P (2012) Worldwide pose estimation using 3D point clouds. In: A Fitzgibbon, S Lazebnik, P Perona, Y Sato and C Schmid (eds.) *Computer Vision – ECCV* 2012 (*Lecture Notes in Computer Science*, *vol. 7572*). New York: Springer, pp. 15–29.

Li Y, Snavely N and Huttenlocher DP (2010) Location recognition using prioritized feature matching. In: *ECCV*.

Lim H, Park J, Lee D and Kim HJ (2012a) Build your own quadrotor: Open-source projects on unmanned aerial vehicles. *IEEE Robotics & Automation Magazine* 19(3): 33–45.

Lim H, Sinha SN, Cohen MF and Uyttendaele M (2012b) Real-time image-based 6-DOF localization in large-scale environments. In: *2012 IEEE conference on computer vision and pattern recognition (CVPR)*. IEEE, pp. 1043–1050.

Lowe DG (2004) Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* 60(2): 91–110.

Meier L, Tanskanen P, Heng L, Lee GH, Fraundorfer F and Pollefeys M (2012) Pixhawk: A micro aerial vehicle design for autonomous flight using onboard computer vision. *Autonomous Robots*33(1–2): 21–39.

Milford M (2013) Vision-based place recognition: How low can you go? *The International Journal of Robotics Research* 32(7): 766–789.

Nister D and Stewenius H (2006) Scalable recognition with a vocabulary tree. In: *2006 IEEE computer society conference on computer vision and pattern recognition*, vol. 2. IEEE, pp. 2161–2168.

Ozuysal M, Calonder M, Lepetit V and Fua P (2010) Fast keypoint recognition using random ferns. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32(3): 448–461.

Robertson DP and Cipolla R (2004) An image-based system for urban navigation. In: *BMVC*, pp. 1–10.

Royer E, Lhuillier M, Dhome M and Lavest JM (2007) Monocular vision for mobile robot localization and autonomous navigation. *International Journal of Computer Vision* 74(3): 237–260.

Salas-Moreno RF, Newcombe RA, Strasdat H, Kelly PHJ and Davison AJ (2013) Slam++ : Simultaneous localisation and mapping at the level of objects. In: *Computer vision and pattern recognition (CVPR)*.

Sattler T, Leibe B and Kobbelt L (2011) Fast image-based localization using direct 2D-to-3D matching. In: *2011 IEEE international conference on computer vision (ICCV)*. IEEE, pp. 667–674.

Sattler T, Leibe B and Kobbelt L (2012) Improving image-based localization by active correspondence search. In: *Computer Vision–ECCV* 2012. New York: Springer, pp. 752–765.

Schindler G, Brown M and Szeliski R (2007) City-scale location recognition. In: *IEEE conference on computer vision and pattern recognition, 2007 (CVPR'07)*. IEEE, pp. 1–7.

Se S, Lowe DG and Little JJ (2005) Vision-based global localization and mapping for mobile robots. *IEEE Transactions on Robotics* 21(3): 364–375.

Skrypnyk I and Lowe DG (2004) Scene modelling, recognition and tracking with invariant image features. In: *Third IEEE and ACM international symposium on mixed and augmented reality, 2004 (ISMAR 2004)*, pp. 110–119.

Snavely N, Seitz SM and Szeliski R (2008) Modeling the world from internet photo collections. *International Journal of Computer Vision* 80(2): 189–210.

Ta DN, Chen WC, Gelfand N and Pulli K (2009) Surftrac: Efficient tracking and continuous object recognition using local feature descriptors. In: *IEEE conference on computer vision and pattern recognition, 2009 (CVPR 2009)*. IEEE, pp. 2937–2944.

Tola E, Lepetit V and Fua P (2008) A fast local descriptor for dense matching. In: *IEEE conference on computer vision and pattern recognition, 2008 (CVPR 2008)*. IEEE, pp. 1–8.

Tomasi C and Kanade T (1991) *Detection and Tracking of Point Features*. Technical Report CMU-CS-91-132, Carnegie Mellon University.

Wagner D, Reitmayr G, Mulloni A, Drummond T and Schmalstieg D (2010) Real-time detection and tracking for augmented reality on mobile phones. *IEEE Transactions on Visualization and Computer Graphics* 16(3): 355–368.

Weiss S, Scaramuzza D and Siegwart R (2011) Monocular-SLAM-based navigation for autonomous micro helicopters in GPS-denied environments. *Journal of Field Robotics* 28(6): 854–874.

Wendel A, Irschara A and Bischof H (2011) Natural landmark-based monocular localization for MAVs. In: *IEEE international conference on robotics and automation (ICRA)*. IEEE, pp. 5792–5799.

Wendel A, Maurer M and Bischof H (2012) Visual landmark-based localization for MAVs using incremental feature updates. In: *2012 second international conference on 3D imaging, modeling, processing, visualization and transmission (3DIMPVT)*. IEEE, pp. 278–285.

Williams B, Klein G and Reid I (2007) Real-time SLAM relocalisation. In: *IEEE 11th international conference on computer vision, 2007 (ICCV 2007)*. IEEE, pp. 1–8.

Winder S, Hua G and Brown M (2009) Picking the best DAISY. In: *IEEE conference on computer vision and pattern recognition, 2009 (CVPR 2009)*. IEEE, pp. 178–185.

Yi C, Suh IH, Lim GH and Choi BU (2009) Active-semantic localization with a single consumer-grade camera. In: *IEEE international conference on systems, man and cybernetics, 2009 (SMC 2009)*. IEEE, pp. 2161–2166.

Appendix: Index to Multimedia Extensions

Archives of IJRR multimedia extensions published prior to 2014 can be found at http://www.ijrr.org, after 2014 all videos are available on the IJRR YouTube channel at http://www.youtube.com/user/ijrrmultimedia

Table of Multimedia Extensions

Extension	Media type	Description
1	Video	Video of evaluation sequences in the Vicon-Lab environment.
2	Video	Video of localization on the FLIGHT2 sequence in the LAB environment.
3	Video	Video of semantic localization on WALK2 sequence in the LAB environment.
4	Video	Video of localization results from four sequences in the HALL environment.
5	Video	Video of localization on outdoor sequences captured by Dong et al. (2009).