Towards Real-Time Semantic Localization

Hyon Lim
Seoul National University, Korea
hyonlim@snu.ac.kr

Sudipta N. Sinha
Microsoft Research, Redmond, USA
sudipsin@microsoft.com

***Abstract*— We describe an efficient image-based localization system which can be used for real-time, continuous semantic localization within a known environment. Our system can precisely localize a camera in real-time from a video stream within a fairly large scene that has been reconstructed offline using structure from motion (SfM). This is achieved by interleaving a fast keypoint tracker that uses BRIEF descriptors, with a direct 2D-to-3D matching approach for recognizing 3D points in the map. Our approach does not require the construction of an explicit semantic map. Rather semantic information can be associated with the 3D points in the SfM reconstruction and can be retrieved via recognition during online localization.**

I. Introduction

In robotics, *semantic localization* 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 [1], coarse location estimates could be recovered using only three states – nearby, near and far with respect to semantic landmarks in the scene. In contrast, our system aims for precise, semantic localization based on real-time 6-DOF image-based localization [2].

We represent the map with a 3D point cloud reconstruction computed using SfM, which also contains multiple DAISY feature descriptors [3,4] associated with the 3D points. By tracking keypoints in video and matching them to the 3D points, our system continuously estimates a precise pose estimate in real-time. The main idea involves interleaving a fast keypoint tracker that uses BRIEF features [5] with an efficient approach for direct 2D-to-3D matching. The 2D-to-3D matching 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 efficiency lies in invoking DAISY descriptor extraction and matching sparingly during online localization, and in distributing this computation over a window of successive frames. Fig. 1 shows the trajectory of a camera mounted on a quadrotor micro-aerial vehicle (MAV), computed using our real-time localization system, as the MAV is flown manually[1].

Unlike our work, visual SLAM (VSLAM) systems have the flexibility of being able to localize a camera within an unknown scene [6,7]. However, semantic localization in an unknown scene can be extremely challenging. Objects must be recognized by their categories, which is very difficult to achieve even without real-time constraints [8]. Although, prebuilt maps are necessary in our method, this also provides the underlying framework for storing detailed semantic information along with 3D points in the scene. During online localization, semantic information can be retrieved via visual recognition of 3D points in the map which are subsequently tracked in video. Our system can handle maps with an order of magnitude more 3D points than typically handled by VSLAM systems. This makes our system robust and enables both continuous localization over long durations within large scenes as well as fast relocalization whenever needed.

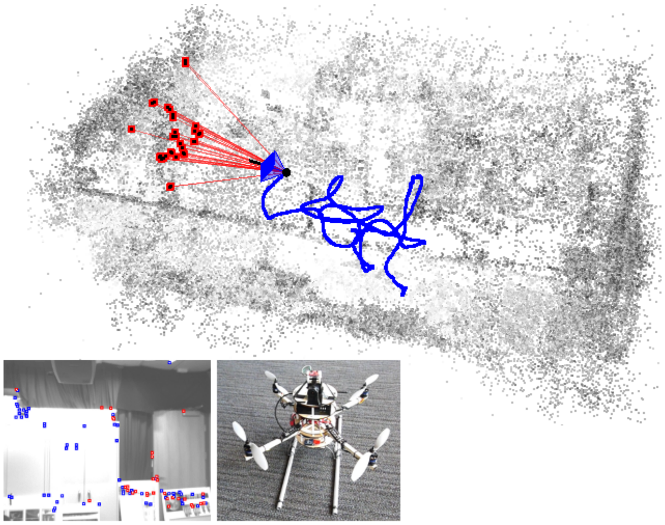

Fig. 1. The trajectory of a quadrotor micro aerial vehicle (MAV) within a 8m × 5m room computed using our method. The SfM reconstruction has 76K points. A video with the recognized landmarks are shown in red. The corresponding 3D points are shown on the map.

II. Our Method

We represent the scene with a 3D reconstruction in a global coordinate frame, which is computed using SfM from an image sequence. The calibrated images are used to build a database of DAISY feature descriptors associated with the 3D points. A kd-tree index is constructed over the descriptors to support efficient approximate nearest neighbor (ANN) queries during online feature matching. Fig. 2 shows an overview of the various steps.

A. Map Construction

The map is built offline using the following steps:

- The input images are processed using SfM [9].
- The cameras are grouped into overlapping clusters.

[1] See http://goo.gl/Vp6ps for a video of our real-time system.

- Keypoints and DAISY descriptors are extracted at multiple scales in the images and associated with the 3D points.
- A kd-tree is built for all the descriptors. Appropriate lookup tables are built to support efficient queries to find which image or 3D point a feature descriptor belongs to.

Semantic labels can be added to the 3D points by annotating the images with object names and bounding boxes [9]. Using the 2D-3D correspondences obtained from SfM, the labels can be easily mapped from pixels to subsets of 3D points.

B. *Object Labeling*

The 3D points are labeled with tags in interactive 3D viewer of point-cloud. User draws a bounding box in an image around object. All 2D measurements in that image within the bounding box are selected and the associated 3D points are selected. An user provides a tag for the selected 3D points. The tag is associated to the 3D points. This tag will be shown on image with position detected during real-time localization.

C. *Place Recognition*

In large scenes, global matching can be difficult due to greater ambiguity in feature descriptors. To address this, we perform coarse place recognition to filter erroneous 2D-3D matches before 6-DOF pose estimation step. As a result, fewer RANSAC hypotheses are required during robust pose estimation, making that step more efficient. For place recognition, we cluster nearby cameras based on SfM results by solving an *overlapping view clustering* problem where cameras with many SfM points in common are grouped into the same cluster [10]. When localizing an image, the most likely camera group is selected using a simple voting scheme over the set of matching descriptors returned by the ANN query on the descriptor group.

D. *Real-time Localization*

Our algorithm aims for real-time localization over long periods and at avoiding fluctuations in the frame-rate. At its core lies a fast keypoint tracker. Keypoints (Harris corners) from one frame are tracked in the following frame by matching to candidate keypoints within a local search window in the next frame [11]. Binary feature descriptors (BRIEF) [5] are used to find the best frame-to-frame matches. This fast tracker is interleaved with an efficient approach to find which 3D points in the map correspond to the tracked keypoints. The camera pose for each frame is robustly estimated from these 2D-3D matches. For determining these matches, DAISY descriptors [3,4] must be extracted. This can be computationally expensive depending on the number of descriptors extracted and queried in the kd-tree. Our system amortizes this cost by requiring that the feature matching be performed on demand and by spreading the computation over a window of successive frames.

For each 3D point currently being tracked by localizer, we lookup its corresponding tags and each 3D point gives 1 vote for each of its tags. To determine which tag is good for a 3D point, we check which tag has more than 2 votes.

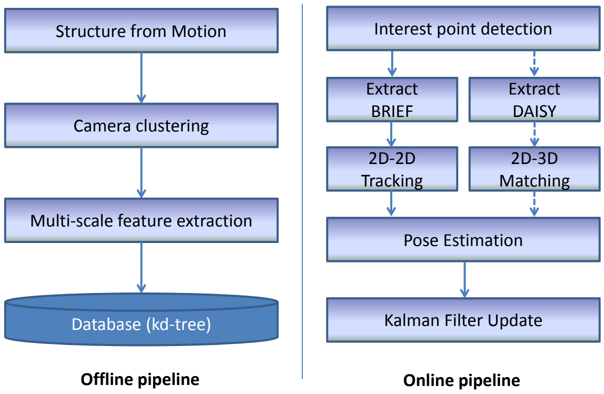

Fig. 2. Overview of the offline and online processing steps in our system. As mentioned in Section I, extraction of DAISY features and 2D-3D matching queries are not executed at every frame.

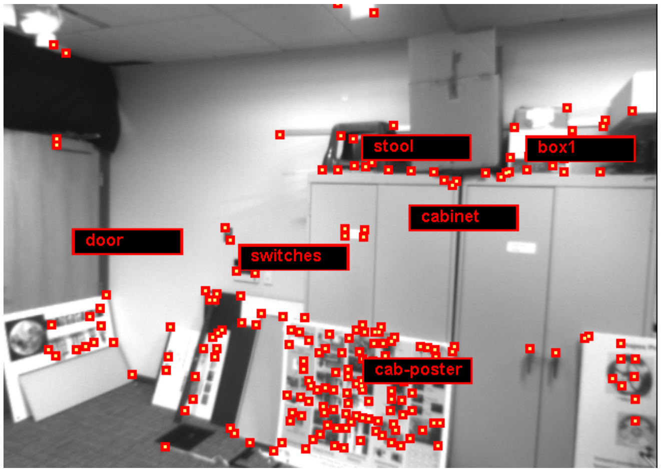

Fig. 3. Annotated descriptions are shown on objects.

The image location of each tag is computed on the fly as follows. By considering all the 2D tracked features in the current frame, which have been matched to the 3D points having that tag. The mean x and y positions of these 2D features is computed.

III. Results

A single-threaded C++ implementation of our system runs at an average frame-rate exceeding 30Hz on multiple datasets on a laptop with an Intel Core 2 Duo 2.66GHz processor running Windows 7. It is about fives times faster than the single-threaded implementation of [12], which runs at 6Hz (and at 20Hz using four cores). To test the feasibility of our method for onboard processing on a small MAV, we designed our own quadrotor vehicle mounted with the PointGrey FireflyMV camera and a FitPC2i[2] computer running Windows 7. Our algorithm runs at about 12Hz on the FitPC.

As described in Section II-B, labels are assigned to features by a user to associated physical objects. During the real-time localization, objects are simultaneously tracked. Object

[2] The FitPC, which has an Intel Atom Z550 2GHz CPU, 2GB RAM and a 64GB SSD drive, weighs less than 500gms. (incl. battery) and consumes only 10W at full load.

labels are displayed in position of objects in an image which is obtained by localizer. Fig. 3 shows that annotated texts are shown on the position of object in an image.

IV. Conclusions

Our real-time localization system [2] is extended for semantic localization, by augmenting the 3D points in the map with semantic labels. This is done by manually inserting annotation in the 2D images used for map construction (offline SfM) and automatically transferring the labels to the 3D points in the map.

During online localization, our system recognizes subsets of 3D points in the map using an efficient 2D-to-3D matching approach and then tracks the 3D points in video. Whatever semantic labels are stored with the tracked 3D points can be used to recognize objects or locations in the video. Additional semantic information can be inferred from the camera pose estimate and from the accurate 3D map where objects and semantic locations are precisely localized. For instance, the relative location or distances to nearby objects that are not yet visible in the camera can be predicted using this information.

References

[1] C. Yi, I. H. Suh, G. H. Lim, and B.-U. Choi, "Active-semantic localization with a single consumer-grade camera," in *SMC*, 2009.

[2] H. Lim, S. N. Sinha, M. Cohen, and M. Uyttendaele, "Real-time image-based 6-dof localization in large-scale environments," in *CVPR (to appear)*, June. 2012.

[3] E. Tola, V. Lepetit, and P. Fua, "A fast local descriptor for dense matching," in *CVPR*, 2008.

[4] S. A. J. Winder, G. Hua, and M. Brown, "Picking the best DAISY," in *CVPR*, 2009, pp. 178–185.

[5] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "BRIEF: Binary Robust Independent Elementary Features," in *ECCV*, 2010.

[6] B. Williams, G. Klein, and I. Reid, "Real-time SLAM relocalisation," in *ICCV*, 2007.

[7] G. Klein and D. Murray, "Parallel Tracking and Mapping for Small AR Workspaces," in *ISMAR*, November 2007.

[8] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *IJCV*, vol. 88, no. 2, pp. 303–338, June 2010.

[9] N. Snavely, S. M. Seitz, and R. Szeliski, "Modeling the World from Internet Photo Collections," in *IJCV*, 2008.

[10] Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski, "Towards internet-scale multi-view stereo," in *CVPR*, 2010.

[11] D. Ta, W.-C. Chen, N. Gelfand, and K. Pulli, "SURFTrac: Efficient Tracking and Continuous Object Recognition using Local Feature Descriptors," in *CVPR*, 2009.

[12] Z. Dong, G. F. Zhang, J. Y. Jia, and H. J. Bao, "Keyframe-based Real-time Camera Tracking," in *ICCV*, 2009.