Semantic Mapping for View-Invariant Relocalization

Jimmy Li[1], David Meger[1] and Gregory Dudek[1]

***Abstract*— We propose a system for visual simultaneous localization and mapping (SLAM) that combines traditional local appearance-based features with semantically meaningful object landmarks to achieve both accurate local tracking and highly view-invariant object-driven relocalization. Our mapping process uses a sampling-based approach to efficiently infer the 3D pose of object landmarks from 2D bounding box object detections. These 3D landmarks then serve as a view-invariant representation which we leverage to achieve camera relocalization even when the viewing angle changes by more than 125 degrees. This level of view-invariance cannot be attained by local appearance-based features (e.g. SIFT) since the same set of surfaces are not even visible when the viewpoint changes significantly. Our experiments show that even when existing methods fail completely for viewpoint changes of more than 70 degrees, our method continues to achieve a relocalization rate of around 90%, with a mean rotational error of around 8 degrees.**

I. Introduction

Visual simultaneous localization and mapping (SLAM) has traditionally relied on matching image intensities or local appearance-based features (e.g. SIFT [1], ORB [2]) between image frames to localize the camera and to reconstruct a point cloud map of the environment. While this approach has been shown to yield excellent tracking accuracy and computational efficiency [3], [4], the low-level map representation does not lend itself well to the relocalization task (global localization) when the viewing angle changes significantly, since the same surfaces are no longer visible.

In this paper we propose a SLAM system that builds a semantic map of the environment consisting of objects represented as 3-dimensional cuboids, a classical representation that has recently returned to favor for computer vision and machine learning [5]–[9]. These cuboids are inferred from 2D observations of objects in the form of detected bounding boxes. Since the presence of objects can be detected regardless of viewpoint, our object-based maps are inherently viewpoint invariant, and therefore well-equipped to support view-invariant relocalization. Figure 1 shows an example of the system operating on two video sequences of the same scene taken from drastically different viewing directions. The system is able to localize both camera trajectories in a common coordinate frame by spatially aligning the objects that are commonly visible in both sequences. Our method achieves this using only RGB images as input, without any depth measurements or inertial sensing.

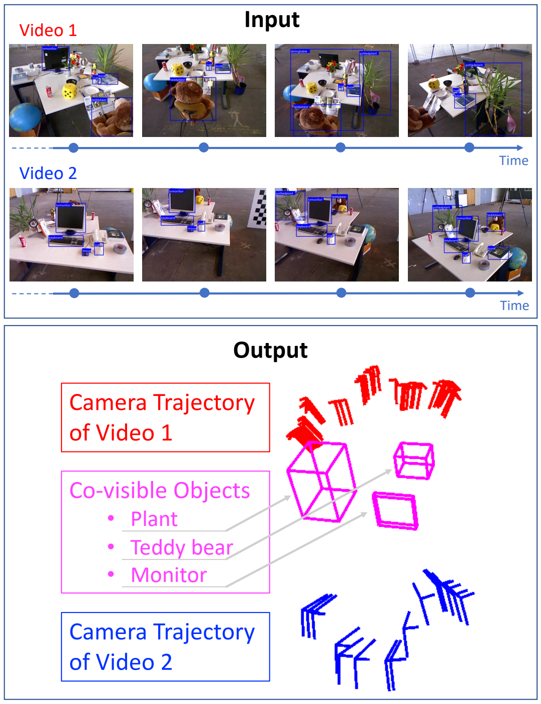

Fig. 1. View-invariant relocalization: Given two RGB video sequences of the same scene, our method first builds an object-based map using each sequence separately. It then uses the 3D layout of co-visible object landmarks to localize the camera poses of both trajectories in a common frame of reference.

The use of natural objects can provide great robustness across viewpoints, but the use of objects alone has several disadvantages. Due to their sparsity and the limited field of view of cameras, often very few objects are wholly visible in any given image. In addition, their projections onto the image plane are difficult to localize precisely – modern object detectors can only produce bounding boxes that roughly enclose the object in the image plane. Despite some successes [10], [11], these characteristics often make objects unsuitable for map initialization and local tracking in the absence of other cues, especially in the initial phase when only a few image frames are available.

Thus, instead of relying solely on object landmarks, we propose a hybrid approach: we use local appearance-based features to track the camera as it moves locally, and leverage the estimated camera trajectory to simplify the estimation of 3D object landmarks during tracking. By having multiple observations of each object during this local mapping process, we overcome the coarseness of bounding box measurements and reduce the chance of poor landmark estimates due to

*This work was supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada, and the NSERC Canadian Robotics Network (NCRN)

[1] The authors are affiliated with the Mobile Robotics Lab, the Center for Intelligent Machines (CIM), and the School of Computer Science at McGill University in Montréal, Canada. jimmyli, dmeger, dudek@cim.mcgill.ca

the camera's narrow field of view. Then, during subsequent localization where local appearance-based features becomes less helpful due to the lack of co-visible surfaces, object landmarks can facilitate higher-level reasoning and help to establish data associations across very large viewpoint changes. This synergistic integration between traditional visual SLAM and semantic landmarks is the key contribution of this work.

Although this paper is focused on measuring the utility of object landmarks for the relocalization task, having a better semantic understanding of the world has many other important applications in robotics, such as manipulation and natural language understanding. Our cuboidal representation of objects describes occupied regions in 3D space, which is often more useful than point clouds for path planning and obstacle avoidance.

II. Related Work

Recently there has been considerable interest in improving the robustness of place recognition and visual localization tasks. Earlier work such as FAB-MAP [12], [13] that is based purely on matching local appearance-based features have been shown to be susceptible to changes in condition (e.g. changes in illumination, weather, season, time of day) [14]. Consequently a variety of methods have been proposed for condition-invariance, including matching image sequences [15]–[17], training location-specific image detectors [18]–[20], predicting environmental changes in images [21]–[23], and transforming images to become more illumination invariant [24]–[26]. Features extracted using convolutional neural networks (ConvNet) trained for image classification have also been shown to be robust to condition and drastic scale changes [27], [28]. Compared to these prior works, our method aims to handle a much greater degree of viewpoint invariance (over 125 degrees of viewpoint change).

The use of objects as landmarks for visual localization tasks has also been studied since they can be detected regardless of viewpoint and environmental condition, and are therefore highly useful for robust data association [10], [11], [29]–[38]. The adoption of this approach is in part due to the recent advancement of ConvNet-based object detectors [39]–[41]. While depth cameras have been leveraged to simplify the object detection task [33]–[35], in our approach we opt to use only RGB images as input, which makes our algorithm applicable to a wider range of hardware platforms due to the ubiquity of RGB sensors.

We are particularly inspired by the work of Bao et al., which also infers 3D cuboids from 2D bounding boxes under the structure from motion setting [30]. Our object mapping method is highly related the semantic SLAM system of Bowman et al., which also uses an expectation maximization scheme for iteratively solving data association and object pose update [37]. Our work puts greater emphasis on the relocalization problem and we directly measure relocalization performance under large viewpoint changes. In our own prior work, we have shown wide-baseline camera pose estimation using objects detected in two far-apart images [10], [11]. In this paper we extend this line of work and propose a full SLAM pipeline that is built on top of the popular ORB-SLAM framework.

III. Method

A. Problem Statement

Our method consists of two components which we will discuss in turn: 1) a semantic mapping algorithm that tracks the 3D pose of objects from frame to frame and produces a metric map containing objects; and 2) a relocalization algorithm, which, given two semantic maps of the same scene, aligns the two maps together and in doing so also produces the relative camera transformation.

B. Semantic Mapping

Given an RGB video sequence, we aim to compute the 3D poses of visible objects represented as bounding cuboids. A bounding cuboid is expressed as a 9-dimensional vector containing its position (x, y, z), orientation (roll, pitch, yaw), and scale (length, width, height).

Our mapping process, shown in Algorithm 1, processes image frames sequentially, and incrementally builds a semantic map as images stream in. Each image frame contains 2D bounding box detections, which we use to infer the 3D pose of objects. To facilitate object pose estimation, we use ORB-SLAM 2 [3], an existing visual SLAM technique, to track the camera pose during mapping. ORB-SLAM relies on matching local appearance-based features (ORB features [2]) between image frames to compute camera extrinsic parameters, and performs reliably given continuous video input. Having obtained known camera poses from ORB-SLAM, observations of objects in 2D can be triangulated probabilistically to allow lifting to full 3D object cuboids, through our sampling-based inference procedure.

We use Faster-RCNN [41] trained on the COCO dataset [42] to compute object observations in the form of 2D bounding box detections. To simplify the inference of 3D object geometry, we assume that objects are aligned with the scene layout. The scene layout consists of three orthogonal major axes, which can typically be reliably detected in urban environments. We use the method of Lee et al. [43] in our pipeline to compute the scene major axes, and restrict our inferred 3D object bounding cuboids to align with them.

To continuously refine the estimated pose of object landmarks as more observations become available, we use an expectation maximization scheme involving data association followed by object pose update in each iteration of the algorithm. Data association involves matching 3D landmarks with 2D object detections. During pose update we then use the matched detections to refine the 3D pose of landmarks, while also ensuring that the landmarks conform to object-to-object contextual relationships. Below we discuss each algorithmic component in turn. Corresponding line numbers of Algorithm 1 are shown in parentheses following each section header.

Algorithm 1 Semantic Mapping

```
1:  landmarks ← empty list
2:  while True do
3:    I ← next image from camera
4:    ORB-SLAM.track_monocular(I)
5:    if ORB-SLAM has not initialized then
6:      continue
7:    end if
8:    keyframes ← ORB-SLAM.get_keyframes()
9:    for k in keyframes do
10:     detections ← DetectObjects(k)
11:     Match(k, landmarks, detections)
12:   end for
13:   for o in landmarks do
14:     $K_o$ ← keyframes in which o is detected
15:     $D_o$ ← detections of o in keyframes $K_o$
16:     $H_o$ ← GenHypotheses($K_o$, $D_o$)
17:     Add o to $H_o$
18:   end for
19:   for o in landmarks do
20:     o ← $\text{argmin}_{h \in H_o}$ Score(h, landmarks)
21:   end for
22:   new_k ← newest keyframe
23:   new_detections ← DetectObjects(new_k)
24:   for d in new_detections do
25:     if d not matched with any o ∈ landmark then
26:       o ← InitLandmark(d, layout)
27:       Add o to landmarks
28:     end if
29:   end for
30: end while
```

1) Initialization (lines 3-8): During each iteration of our algorithm, we obtain a new image and ask ORB-SLAM to track the image. We wait until ORB-SLAM initializes, upon which estimated camera poses for keyframes become available. ORB-SLAM is designed to run in real time, and as such, it does not maintain the camera pose at every frame. Instead, bundle adjustment operates on a sparser set of keyframes, which reduces the computational load.

2) Data Association (lines 9-12): Once ORB-SLAM has initialized, we update the 3D pose of our object landmarks during each iteration. We start by projecting the landmarks (3D bounding cuboids) into each keyframe image as bounding boxes, and matching these projections with the detections. Matching is performed by using the Hungarian algorithm [44] which finds the optimal solution in $O(n^3)$ time given the cost of matching each projection with each bounding box, where n is the number of landmarks to be matched. The cost $c(p, d)$ of matching a projected bounding box p with a detected bounding box d is

$$c(p,d) = \frac{|p_l - d_l| + |p_t - d_t| + |p_r - d_r| + |p_b - d_b|}{d_r - d_l} \quad (1)$$

where the subscripts l, t, r, b denote the left, top, right, and bottom sides of the bounding box in pixel coordinates. The denominator normalizes the cost by the width of the detected bounding box to prevent larger bounding boxes from dominating the overall matching cost.

3) Object Pose Update (lines 13-21): The object pose update step addresses the most challenging aspects of our problem, which involves optimizing each object's pose and scale over a highly non-convex search space. The non-convexity arises from the complex 3D geometry of cuboids and their ambiguous projection into the image plane. A naive application of Markov chain Monte Carlo (MCMC) sampling techniques will result in very long run times as the sampler will need to traverse numerous local minima.

Our proposed strategy is to first efficiently generate multiple object hypotheses in a reduced search space, and then leverage them to quickly explore multiple local minima in the full search space. To this end, we begin by representing an object landmark as a single 3D point (x, y, z). From the previous step, we know its correspondence to detections in multiple keyframes. Since we know the keyframe camera poses from ORB-SLAM, we can triangulate the 3D point location of the landmark by intersecting rays extending from the camera centers through the detected bounding boxes. A key question is how to select a point in the bounding box to extend the ray through. A natural choice is to use the center of the bounding box, but we find that this often is not the best choice since the bottom of objects often become occluded when the camera is held at eye-level. Instead, we use the top-center point of the bounding box, which means that the triangulated point should be near the top surface of our object landmark. For simplicity, we will think of this triangulated point as approximating the top center point of a 3D landmark.

We follow the approach described by Hartley and Zisserman [45] and model the probability of the top center point X of a 3D landmark being projected onto the image point x_k in keyframe k as a normal distribution

$$p(x_k|X) = \frac{\exp\left(-\frac{1}{2}(f_k(X) - x_k)^T \Sigma^{-1} (f_k(X) - x_k)\right)}{\sqrt{(2\pi)^2 \det \Sigma}} \quad (2)$$

where f_k projects X into the image associated with keyframe k, and Σ is the 2×2 covariance matrix of x in image space. We wish to compute the *a posteriori* distribution $p(X|x_1, .., x_n)$, and assuming a uniform $p(X)$ and independent observations between views, we have

$$\begin{aligned} p(X|x_1, ...x_n) &= p(x_1, ...x_n|X)p(X)/p(x_1, ..., x_n) \\ &\sim p(x_1, ..., x_n|X) \\ &= p(x_1|X)...p(x_n|X). \end{aligned} \quad (3)$$

Thus, to obtain point-location hypotheses of our landmark, we can draw samples from $p(X|x_1, ...x_n)$, and the above equation allows us to compute the unnormalized probability of a sample. Several sampling techniques can be applied here. If a random walk Monte Carlo sampling method is used, it is possible to first triangulate the landmark using a method such as direct linear transform (DLT) [45] and then use the triangulated value as the initial sample to bootstrap the random walk. In order to achieve more efficient sampling, in our implementation we use importance sampling, and only draw samples that lie along the rays extending from the camera centers.

Having sampled a set of object landmark hypotheses as 3D points, we instantiate a 3D bounding cuboid for each point, anchoring the center of the top face of the bounding cuboid at the 3D point. We align the cuboid with the major axes of the room layout to prevent having to search over the full space of orientations. To initialize the scale of an object, we use the average length, width, and height of that object type, and then apply an isotropic scaling to all three scale dimensions such that the projection of the cuboid aligns as well as possible with the detected bounding boxes in the relevant keyframes. Note that this gives an imprecise initial cuboid estimate since the top center face of a cuboid typically does not project onto the top center point of a bounding box, and the scale dimensions of an object instance may be very different from the average. Nevertheless, this process gets us close to a local minimum, which allows us to then refine the cuboid hypothesis with Metropolis-Hastings MCMC [46] over the pose (x,y,z) and scale (length, width, height) jointly. This process of generating a list of cuboid hypotheses corresponds to GenHypotheses in the algorithm listing.

We now score the cuboid hypotheses we have generated. Let L be the set of landmarks, and H_o be the hypotheses of $o \in L$. The score $S(h)$ of an object hypothesis h for landmark o is

$$S(h) = \frac{\alpha}{|K_o|} \sum_{k \in K_o} c(f_k(h), \delta_k) + \sum_{\substack{o' \in L \\ h \notin H_{o'}}} \min_{h' \in H_{o'}} \Gamma(h, h'). \tag{4}$$

Here, K_o contains the keyframes in which o has an associated detection, δ_k is the associated detection in k, and f_k projects h into keyframe k. The cost function c is as describe in equation 1. α is a tuning factor used to scale the first sum. Γ measures the coherence of object cuboids h and h' based on their typical contextual relationship (e.g. mouse and keyboard tend to lie on the same surface). More details of contextual modelling will be given in the next section. Here, a lower score indicates a better hypothesis.

Finally, after having scored every hypothesis for all the landmarks, we update each landmark with the best-scoring hypothesis. Note that this is only done if the best-scoring hypothesis achieves a better score than the score of the existing estimate.

4) Contextual Coherence (line 20): We use contextual constraints to regularize the estimation of object pose by encouraging collections of object landmarks to conform to typical spatial relationships. Our prior work on object-to-object context modelling [10], [11], [47] show that coplanarity between objects (e.g. tables and chairs tend to lie on the same surface) is a highly reliable constraint for object pose estimation [10], and so for this work we define Γ as

$$\Gamma(h, h') = \mathbb{1}_{\textsc{Coplanar}(h,h')} \textsc{BottomDist}(h, h') \tag{5}$$

where BottomDist(h,h') gives the distance between the bottom surfaces of two cuboids and Coplanar indicates whether two objects typically lie on the same surface. For example, keyboards and monitors tend to lie on the same surface where as keyboards and chairs do not.

5) Landmark Initialization (lines 22-29): Thus far, we have discussed how existing landmark estimates are refined in each iteration of our algorithm. At each iteration, if a detected bounding box is not matched to any existing landmark during the data association step, then a new object landmark is instantiated for this detection. To do this, we simply generate a cuboid whose projection aligns well with the detection. The accuracy of the cuboid matters little, since the camera will not have moved very much in the subsequent keyframes, and therefore the cuboid's projection in subsequent keyframes will be good enough to establish data association. Once multiple views of the landmark are available, our algorithm will be able to quickly refine the landmark's pose and scale.

In summary, our semantic mapping process creates a map containing 3D object landmarks. Having two semantic maps taken from the same environment, we can align the landmarks in the two maps to achieve view-invariant relocalization. We discuss this in the next section.

C. Relocalization

Given two semantic maps L_1 and L_2 both consisting of a set of object landmarks, relocalization can be summed up with the following maximization problem

$$s^*, o_1^*, o_2^* = \underset{s, o_1 \in L_1, o_2 \in L_2}{\operatorname{argmax}} \; \Omega\Big(\theta(o_1, L_1), \psi\big(s, \theta(o_2, L_2)\big)\Big) \tag{6}$$

where $\theta(o, L)$ returns a new set of landmarks in which the poses of objects in L are expressed in the coordinate frame of o, and $\psi(s, L)$ returns a set of landmarks where L is scaled by s.

The function $\Omega(L_1, L_2)$ performs two operations. The first operation is running the Hungarian algorithm [44] to compute an optimal matching between the two sets of object landmarks. The Hungarian algorithm requires the cost of matching a pair of objects $o_1 \in L_1$ and $o_2 \in L_2$, which in this case is the Euclidean distance between the centroids of o_1 and o_2. We are able to directly compare their poses because the transformations performed by θ and ψ ensure that the two sets of landmarks are expressed in a common frame of reference. To ensure objects with different detected labels do not match, we simply add a constant to the matching cost if their labels are not the same.

The second operation is to identify inliers to the matching. The Hungarian algorithm will try to match as many objects as possible, which entails that an object only visible in the first trajectory can be matched to an object that is only visible in the second trajectory. Correctly matched objects should have close proximity in 3D space so we filter out matches where the inter-object Euclidean distance is greater than a threshold. The remaining matches are called the *inliers* of this match. The function $\Omega(L_1, L_2)$ returns the number of inliers.

Intuitively, equation 6 seeks a pair of cuboids o_1^* and o_2^*, one from each semantic map, such that by expressing the maps in their reference frames and applying a scaling s^* to the second map, we get good spatial alignment between the two sets of object landmarks. Sometimes multiple pairs of cuboids will tie for the maximal number of inliers. In this case, we choose the set of inliers that contain less frequently-occurring object categories since these are less susceptible to ambiguous layouts.

Note that by having identified o_1^* and o_2^* and knowing the relative transformation between their respective frames of reference, we have implicitly also obtained the relative transformation between the two camera trajectories. This is because the rigid transformation between camera poses and landmark objects are known.

An alternative approach to handling scale differences in the two maps is to first leverage the typical known size of objects to scale both semantic maps to be at absolute scale. However, note that many objects have large variances in size (e.g. bottles). Therefore, if two semantic maps do not contain the exact same set of objects, as is often the case under large viewpoint changes, the two maps are likely to end up being scaled differently anyway.

IV. Dataset

Our dataset consists of 14 RGB video sequences: 6 are videos from the UWv2 dataset [48]; one video is the freiburg2_desk sequence of the TUM dataset [49]; and 7 are collected by us using the camera of a commodity smartphone (Huawei Honor 6X) in realistic indoor environments (apartment, computer lab, and lounge). Our goal here is to collect video sequences that capture a set of static objects over a large span of viewing angles.

To prepare each sequence, we use ORB-SLAM to compute the camera pose at every single image frame, which we will use as ground truth. This is done by first running ORB-SLAM in mapping mode, and then processing all image frames in ORB-SLAM's relocalization mode. In other words, we will be comparing our relocalization performance against ORB-SLAM's ability to localize given a continuous video when there are no gaps in viewpoint. ORB-SLAM has been shown to achieve centimeter accuracy for short video sequences and is sufficient for our evaluation [3].

Next, we decompose each sequence into even segments $Q = (s_1, s_2, s_3, ...s_n)$, and keep every other segment $Q^* = (q_1, q_3, q_5...q_n)$. We can then pick any two segments $(q_i, q_j) \in Q^*$ to evaluate relocalization since (q_i, q_j) are now discontinuous (e.g. the views connecting (q_i, q_j) are not given to the system). The system first builds a map using the video segment q_i, then builds a map using q_j, and finally attempts to relocalize using the two maps. In total we evaluate our method on a dataset that contains 52 pairs of segments.

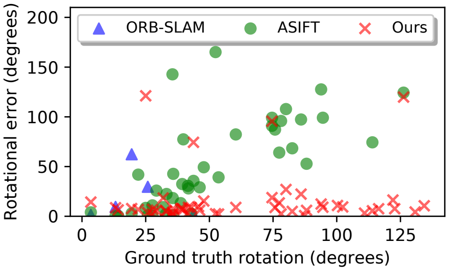

Fig. 2. Rotational error in camera pose estimation during relocalization, as a function of the amount of actual viewpoint change. For our method, we do not observe a large increase in error even as the viewpoint changes by more than 125 degrees, which demonstrates the view-invariant nature of our method. ORB-SLAM does not produce answers beyond 30 degrees due to the lack of feature matches. The proportion of false matches produced by ASIFT feature matching increases as viewpoint change increases, which leads to increased errors.

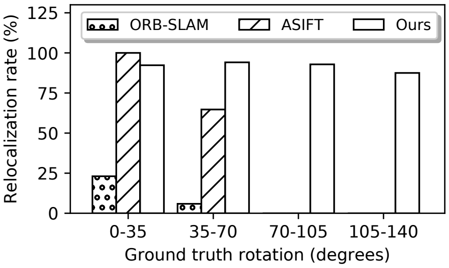

Fig. 3. Relocalization success rate of various methods. Traditional methods based on local appearance-based features (ORB-SLAM, ASIFT) stop working when viewpoint changes by more than 70 degrees, since the same set of surfaces are no longer visible. Our object-based method is robust to much larger viewpoint changes.

V. Experimental Results

A. Pose Estimation

To evaluate the accuracy of relocalizing segments (q_i, q_j), we measure the rotational error of the estimated camera transformation between the first keyframe obtained when mapping q_1 and the first keyframe obtained when mapping q_2. We do not measure translational error since we compute our ground truth using monocular ORB-SLAM, which does not produce absolute scale. However, we find that translational error and rotational error are highly correlated.

Figure 2 shows the rotational error of our method in degrees for all 52 relocalization attempts. The horizontal axis shows the amount of ground truth rotation between the two frames in q_1 and q_2 that have the most similar viewpoint. From the figure we see that our method robustly handles over 125 degrees of camera rotation, which demonstrates our method's robustness to very large viewpoint changes. Figure 4 shows successful relocalization in six different scenes.

In Figure 2 we see a few clearly incorrect estimates with very high errors. We find that this is usually due to

TABLE I

Relocalization Mean Rotational Error

	Mean (degrees)	Stdev (degrees)
ORB-SLAM	10.6	11.5
ASIFT	16.8	14.8
Ours	8.3	5.4

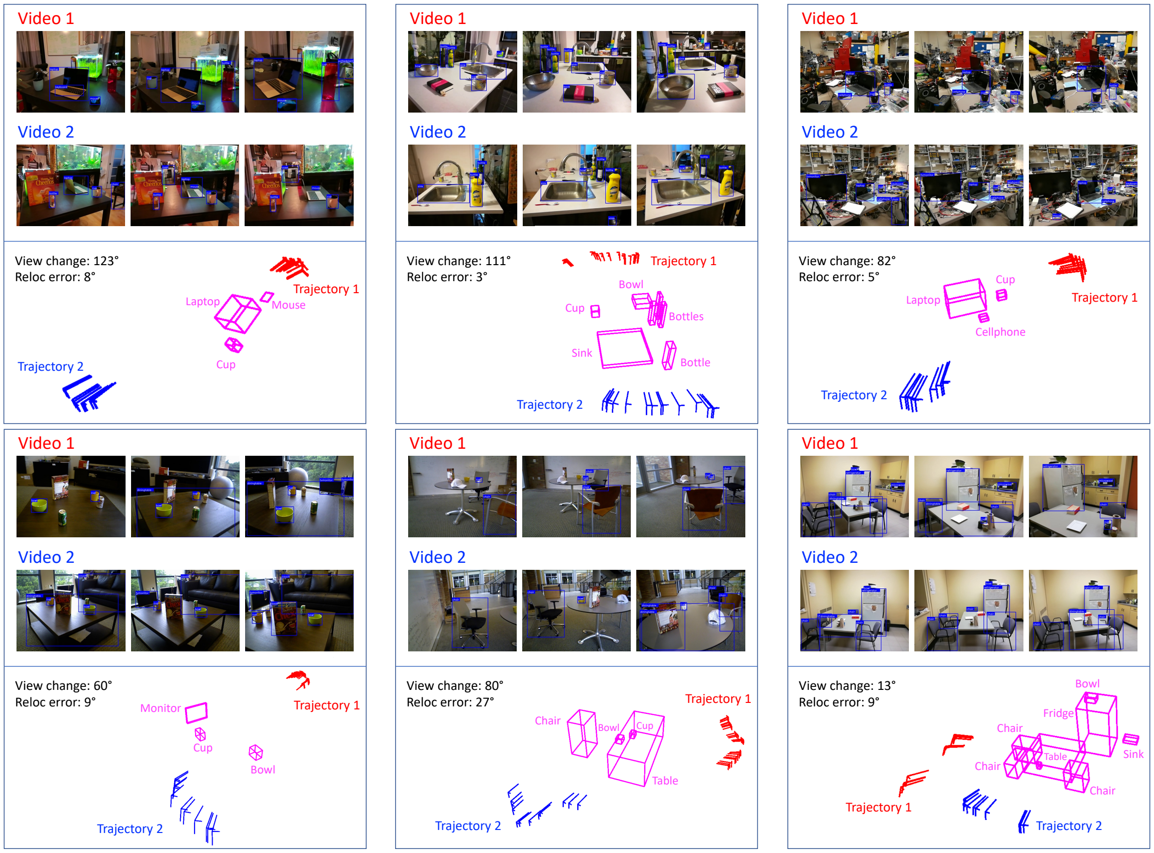

Fig. 4. Six examples of successful relocalization attempts. Our system takes two video sequences, and outputs the estimated camera trajectory associated with both videos in a common frame of reference. For each trajectory we plot the camera poses of keyframes. Co-visible objects that are used to relocalize (inliers) are shown in purple. Objects that are not used for relocalization are not shown here to reduce clutter. For each example we show the amount of viewpoint change between the two trajectories, as well as the relocalization error measured as the rotational error in the relative transformation between the first keyframes of the two trajectories.

highly cluttered scenes with lots of partial occlusion, which leads to poor map estimates and incorrect data associations accidentally aligning.

B. Comparison with ORB-SLAM and ASIFT

We compare with two existing methods that rely on local appearance-based features to solve for camera transformation between images. The first method is ORB-SLAM's relocalization module, which hinges on the matching of ORB features. During relocalization, candidate image frames are first retrieved based on the bag of words [50] representation, and then checked for geometric consistency by matching image features with the existing map using the PnP algorithm [51], which also produces a camera transformation. Given a pair of segments, we let ORB-SLAM map using the first segment, and attempt relocalization using all images of the second segment.

The second method we compare against is fundamental matrix estimation using ASIFT [52] feature matches. ASIFT is a state-of-the-art affine-invariant local image descriptor that is highly robust to rotation and tilt. Given a pair of segments, we take the 50 image pairs (each pair consists of one image from each segment) that are the most similar in viewpoint. We attempt to compute ASIFT matching on each pair, and if matching is successful we estimate a fundamental matrix using the 8-point algorithm [45].

Figure 2 shows relocalization errors for both ORB-SLAM and ASIFT. Note that these methods do not always produce an answer since the feature matching step could simply fail to produce any matches. Additionally, we measure the relocalization rate of all methods, which is shown in Figure 3. To qualify as successful relocalization, an algorithm must 1) produce an answer, and 2) the estimated camera transformation must have a rotational error of less than 40 degrees. From the figure we see that ORB-SLAM and ASIFT fail beyond 70 degrees of viewpoint change, while our method continues to perform reliably beyond 125 degrees. Table I gauges the precision of the three methods by showing the mean and standard deviation of the rotational error for successful relocalization attempts. We see that although local appearance-based features are excellent for tracking, they become less reliable when relocalizing over large baselines. Our method demonstrates not only robustness to viewpoint, but also superior precision.

VI. Conclusion

In this paper we have demonstrated view-invariant relocalization using object landmarks. For future work we intend to experiment in outdoor scenes and larger scale environments, and use the geometrically-detailed semantic map produced by our method for other robotic tasks such as manipulation and natural language direction following.

References

[1] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, pp. 91–110, 2004.

[2] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *International Conference on Computer Vision (ICCV)*. IEEE, 2011, pp. 2564–2571.

[3] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM: a versatile and accurate monocular SLAM system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[4] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *European Conference on Computer Vision (ECCV)*, September 2014.

[5] S. Helmer, D. Meger, P. Viswanathan, S. McCann, M. Dockrey, P. Fazli, T. Southey, M. Muja, M. Joya, J. Little, D. Lowe, and A. Mackworth, "Semantic robot vision challenge: Current state and future directions," *arXiv preprint arXiv:0908.2656*, 2009.

[6] S. Song and J. Xiao, "Deep Sliding Shapes for amodal 3D object detection in RGB-D images," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[7] J. Ku, M. Mozifian, J. Lee, A. Harakeh, and S. Waslander, "Joint 3d proposal generation and object detection from view aggregation," *arXiv preprint arXiv:1712.02294*, 2017.

[8] W. Luo, B. Yang, and R. Urtasun, "Fast and furious: Real time end-to-end 3d detection, tracking and motion forecasting with a single convolutional net," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 3569–3577.

[9] H. Chu, W.-C. Ma, K. Kundu, R. Urtasun, and S. Fidler, "Surfconv: Bridging 3d and 2d convolution for rgbd images," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[10] J. Li, D. Meger, and G. Dudek, "Context-coherent scenes of objects for camera pose estimation," in *International Conference on Intelligent Robots and Systems (IROS)*, 2017.

[11] J. Li, Z. Xu, D. Meger, and G. Dudek, "Semantic scene models for visual localization under large viewpoint changes," in *Conference on Computer and Robot Vision (CRV)*, Toronto, Canada, May 2018.

[12] M. Cummins and P. Newman, "Fab-map: Probabilistic localization and mapping in the space of appearance," *The International Journal of Robotics Research*, vol. 27, no. 6, pp. 647–665, 2008.

[13] ——, "Appearance-only slam at large scale with fab-map 2.0," *The International Journal of Robotics Research*, vol. 30, no. 9, pp. 1100–1123, 2011.

[14] C. Valgren and A. J. Lilienthal, "Sift, surf and seasons: Long-term outdoor localization using local features," in *3rd European conference on mobile robots (ECMR)*, 2007, pp. 253–258.

[15] M. J. Milford and G. F. Wyeth, "Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights," in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2012, pp. 1643–1649.

[16] E. Johns and G.-Z. Yang, "Feature co-occurrence maps: Appearance-based localisation throughout the day," in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2013, pp. 3212–3218.

[17] T. Naseer, L. Spinello, W. Burgard, and C. Stachniss, "Robust visual robot localization across seasons using network flows." in *AAAI Conference on Artificial Intelligence*, 2014, pp. 2564–2570.

[18] C. Linegar, W. Churchill, and P. Newman, "Made to measure: Bespoke landmarks for 24-hour, all-weather localisation with a camera," in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 787–794.

[19] C. McManus, B. Upcroft, and P. Newmann, "Scene signatures : localised and point-less features for localisation," in *Robotics: Science and Systems (RSS)*, University of California, Berkeley, CA, July 2014. [Online]. Available: https://eprints.qut.edu.au/76158/

[20] J. Hawke, A. Bewley, and I. Posner, "What makes a place? building bespoke place dependent object detectors for robotics," in *International Conference on Intelligent Robots and Systems (IROS)*, Sept 2017, pp. 5100–5107.

[21] S. Lowry, M. Milford, and G. Wyeth, "Transforming morning to afternoon using linear regression techniques," in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 3950–3955.

[22] P. Neubert, N. Sunderhauf, and P. Protzel, "Appearance change prediction for long-term navigation across seasons," in *European Conference on obile Robots (ECMR)*. IEEE, 2013, pp. 198–203.

[23] P. Neubert, N. Sünderhauf, and P. Protzel, "Superpixel-based appearance change prediction for long-term navigation across seasons," *Robotics and Autonomous Systems*, vol. 69, pp. 15–27, 2015.

[24] P. Corke, R. Paul, W. Churchill, and P. Newman, "Dealing with shadows: Capturing intrinsic scene appearance for image-based outdoor localisation," in *International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2013, pp. 2085–2092.

[25] B. Upcroft, C. McManus, W. Churchill, W. Maddern, and P. Newman, "Lighting invariant urban street classification," in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 1712–1718.

[26] W. Maddern, A. Stewart, C. McManus, B. Upcroft, W. Churchill, and P. Newman, "Illumination invariant imaging: Applications in robust vision-based localisation, mapping and classification for autonomous vehicles," in *Proceedings of the Visual Place Recognition in Changing Environments Workshop, IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China*, vol. 2, 2014, p. 3.

[27] N. Sünderhauf, S. Shirazi, A. Jacobson, F. Dayoub, E. Pepperell, B. Upcroft, and M. Milford, "Place recognition with convnet landmarks: Viewpoint-robust, condition-robust, training-free," in *Robotics: Science and Systems (RSS)*, 2015.

[28] A. Holliday and G. Dudek, "Scale-robust localization using general object landmarks," in *International Conference on Intelligent Robots and Systems (IROS)*, 2018.

[29] P. Espinace, T. Kollar, A. Soto, and N. Roy, "Indoor scene recognition through object detection," in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2010, pp. 1406–1413.

[30] S. Y. Bao, M. Bagra, Y.-W. Chao, and S. Savarese, "Semantic structure from motion with points, regions, and objects," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[31] D. Meger, C. Wojek, J. J. Little, and B. Schiele, "Explicit occlusion reasoning for 3d object detection." in *British Machine Vision Conference (BMVC)*. Citeseer, 2011, pp. 1–11.

[32] R. Frampton and A. Calway, "Place recognition from disparate views," in *British Machine Vision Conference (BMVC)*. BMVA, 2013.

[33] R. F. Salas-Moreno, R. A. Newcombe, H. Strasdat, P. H. J. Kelly, and A. J. Davison, "Slam++: Simultaneous localisation and mapping at the level of objects," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[34] B. Mu, J. L. Shih-Yuan Liu, Liam Paull, and J. P. How, "Slam with objects using a nonparametric pose graph," in *International Conference on Intelligent Robots and Systems (IROS)*, 2016.

[35] Y. Xiang and D. Fox, "Da-rnn: Semantic mapping with data associated recurrent neural networks," in *Robotics: Science and Systems (RSS)*, 2017.

[36] F. Chayya, D. Reddy, S. Upadhyay, V. Chari, M. Zia, and K. Krishna, "Monocular reconstruction of vehicles: Combining slam with shape priors," in *International Conference on Robotics and Automation (ICRA)*, 2016.

[37] S. Bowman, N. Atanasov, K. Daniilidis, and G. J. Pappas, "Probabilistic data association for semantic slam," in *International Conference on Robotics and Automation (ICRA)*, 2017.

[38] A. G. Toudeshki, F. Shamshirdar, and R. Vaughan, "Robust uav visual teach and repeat using only sparse semantic object features," in *Conference on Computer and Robot Vision (CRV)*, Toronto, ON, Canada, May 2018.

[39] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European Conference on Computer Vision (ECCV)*. Springer, 2016, pp. 21–37.

[40] J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger," in *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017, pp. 6517–6525.

[41] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems (NIPS)*, 2015.

[42] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European Conference on Computer Vision (ECCV)*. Springer, 2014, pp. 740–755.

[43] D. C. Lee, M. Hebert, and T. Kanade, "Geometric reasoning for single image structure recovery," in *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2009, pp. 2136–2143.

[44] J. Munkres, "Algorithms for the assignment and transportation problems," *Journal of the society for industrial and applied mathematics*, vol. 5, no. 1, pp. 32–38, 1957.

[45] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[46] W. K. Hastings, “Monte carlo sampling methods using markov chains and their applications,” *Biometrika*, vol. 57, no. 1, pp. 97–109, 1970.

[47] J. Li, D. Meger, and G. Dudek, “Learning to Generalize 3D Spatial Relationships,” in *International Conference on Robotics and Automation (ICRA)*, Stockholm, Sweden, May 2016.

[48] P. Henry, D. Fox, A. Bhowmik, and R. Mongia, “Patch volumes: Segmentation-based consistent mapping with rgb-d cameras,” in *2013 International Conference on 3D Vision (3DV)*. IEEE, 2013, pp. 398–405.

[49] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, “A benchmark for the evaluation of rgb-d slam systems,” in *International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2012.

[50] D. Gálvez-López and J. D. Tardos, “Bags of binary words for fast place recognition in image sequences,” *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1188–1197, 2012.

[51] V. Lepetit, F. Moreno-Noguer, and P. Fua, “Epnp: An accurate o (n) solution to the pnp problem,” *International Journal of Computer Vision*, vol. 81, no. 2, p. 155, 2009.

[52] G. Yu and J.-M. Morel, “Asift: An algorithm for fully affine invariant comparison,” *Image Processing On Line*, vol. 1, pp. 11–38, 2011.