Semantic Scene Models for Visual Localization Under Large Viewpoint Changes

Jimmy Li, Zhaoqi Xu, David Meger, and Gregory Dudek
School of Computer Science, Centre for Intelligent Machines
McGill University
Montreal, Canada
{*jimmyli,zhaoqixu,dmeger,dudek*}*@cim.mcgill.ca*

***Abstract*—We propose an approach for camera pose estimation under large viewpoint changes using only 2D RGB images. This enables a mobile robot to relocalize itself with respect to a previously-visited scene when seeing it again from a completely new vantage point. In order to overcome large appearance changes, we integrate a variety of cues, including object detections, vanishing points, structure from motion, and object-to-object context in order to constrain the camera geometry, while simultaneously estimating the 3D pose of covisible objects represented as bounding cuboids. We propose an efficient sampling-based approach that quickly cuts down the high-dimensional search space, and a robust correspondence algorithm that matches covisible objects via inter-object spatial relationships. We validate our approach using the publicly available Sun3D dataset [1], in which we demonstrate the ability to handle camera translations of up to 5.9 meters and camera rotations of up to 110 degrees.**

***Keywords*-semantic scene understanding; camera pose estimation; SLAM;**

I. Introduction

Loop closure is a key aspect of simultaneous localization and mapping (SLAM), since it allows the system to correct for drift and build a globally consistent map. Upon detecting a loop, constraints are imposed by data associations between the current observations and mapped landmarks that have been observed earlier in the trajectory. Existing monocular SLAM pipelines typically perform data associations using local appearance, either by directly matching pixel intensity values [2], or by matching features (i.e. SIFT [3], ORB [4]) computed on salient image patches [5]. A limitation of these existing methods is that they are not able to cope with large viewpoint changes, under which the scene appearance changes significantly. An example of such a scenario is shown in Figure 1, where we show two images of the same classroom taken from drastically different views.

Being able to cope with large viewpoint changes has important implications for robots navigating in large-scale environments. For example, suppose an agent visits a large shopping mall, and later enter the same mall via a different entrance. It should be able to recognize the shops it has previously visited even if it sees them from a completely different viewpoint. As robots operate in larger and more complex environments, view-invariant data association is becoming increasingly important for efficient navigation.

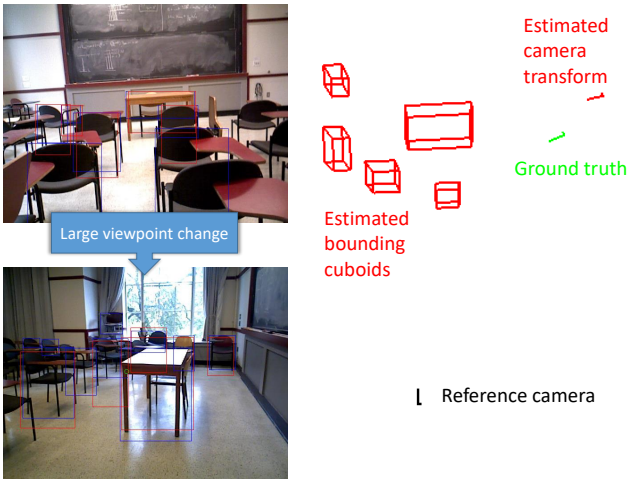


Figure 1. Two views of the same room that our system is able to match. Our method uses RGB images taken from two far-apart views (left) to simultaneously estimate the 6-DOF camera transformation as well as the 6-DOF pose and scale (length, width, height) of covisible objects (right). 2D object detections are used as robust view-invariant features for this task. In this example, our algorithm robustly handles 5.9 meters of camera translation and 110 degrees of camera rotation.

This paper presents an approach that uses objects as high-level features to overcome the appearance discrepancy between disparate views in order to achieve wide-baseline pose estimation using only RGB images. Modern object detectors have been trained on large datasets to recognize semantic identity regardless of viewing angle and partial occlusions. We use the popular Faster-RCNN [6] to produce 2D bounding box detections of objects, which we use as input to our algorithm.

Our algorithm simultaneously recovers the relative pose of two cameras and the 3D location of covisible objects, since these two problems are mutually constraining. We represent objects using 3D bounding cuboids, which encode the length, width, and height of objects in addition to their six degrees of freedom (6-DOF) pose. One challenging aspect of this approach is that each bounding box imposes fairly weak constraints on the object's geometry in 3D: each bounding box can be explained by a large number of cuboids that vary in size and distance from the camera. Therefore, careful inference is required. The automatic inference of 3D

object geometry from RGB images is an attractive aspect of our method, as it additionally enables semantic object recognition and scene understanding. Higher-level tasks such as natural language understanding and manipulation that require a semantic map of the environment could benefit from our approach.

To better constrain scene geometry, we apply several techniques from geometric computer vision that are known independently, but fused here into a novel approach. First, we infer vanishing points from each image, which give a rough sense for the vertical and horizontal alignment of each camera, narrowing the space of possible inter-camera poses to lie within the 4 canonical compass directions. Second, we take a short video, 25 consecutive frames from each viewpoint, allowing relative depth (accurate up to an unknown scale factor) to be recovered for keypoint pixels using structure from motion (SFM). This helps to establish a relative layout of the detected objects. Note that the video frames do not connect the disparate viewpoints, so continuous tracking between them is not possible. Finally, the specific geometry of each object, such as its length, width and height, and inter-object geometry, such as the propensity of objects to lie on a shared support surface, is used to anchor the scene onto correct metric space. In combination, these factors allow us to set up a consistent estimation problem, where image observations and the listed priors are combined. A satisfying solution, when found, indicates a successful loop closure, and produces accurate camera poses and 3D object geometry.

II. Related Work

A core task for a mobile robot is to understand the scene surrounding its workspace. This has sometimes been known as semantic analysis [7], [8], and has been studied by numerous authors across computer vision and robotics. For example, Torralba *et al.* developed an early and highly influential context-based vision system for place and object recognition that was able to identify familiar locations using "gist" appearance models [9]. Beyond single objects, several authors have considered the utility in using scene structure within semantic reasoning. For example, Gould *et al.* performed image segmentation wherein they used region boundaries as a step toward full object identification [10]. At the full object level, geometric context such as the notion of a common support plane have been studied by many authors (e.g., Bao *et al.* [11]). The awareness of the connection between different levels of abstraction and different computational modalities goes back to some of the earliest results in computational vision [7].

A number of methods have previously considered the problem of pose estimation using objects as a feature representation [12]–[18]. We have been inspired by several of the geometric constructions, especially Bao's relation between image bounding boxes to 3D cuboids [12]. One of our major contributions in this paper is an approach to recover accurate data association between corresponding objects over wide baseline camera motions. We were inspired by the work of [17]–[19] among others, as correspondence search is an unavoidable problem. In our own previous work, [20], we have considered the use of 3D object cuboids and inter-object context to perform camera pose reconstruction. That previous work did not consider the same unstructured wide baseline scenes that are the focus of this paper. In particular, our previous work assumed that the scale of detected objects is known and that the camera's viewing angle with respect to the ground plane is fixed. By removing these assumption in this work, the search space of the scene geometry is drastically increased, making it a much harder problem. We have added the use of SFM on nearby video frames, the use of a room-layout detector to find the major axes of the scene and a much more sophisticated correspondence matching approach.

III. Method

A. Overview

Let I_i and I_j be two RGB images that capture a scene from two different viewpoints, where viewpoints i and j can be arbitrarily far apart. We aim to find the 6-DOF camera pose C_j at view j relative to the camera pose at view i. For both I_i and I_j we extract three quantities that will be used as inputs to our pose estimation approach:

- Object detections $D_i = \{d_{i,1}, ...d_{i,n}\}$, where each $d_{i,k}$ consists of a 2D bounding box and a label. n is the total number of detected objects.
- Three orthogonal major axes A_i describing the layout of the scene. These exist in most man-made environments, especially in indoor scenes where objects are usually aligned with the walls.
- A sparse point cloud F_i that is the output of a conventional structure from motion (SFM) algorithm, computed on a short video sequence that contains I_i as part of the sequence. Since the point cloud is produced from RGB images, the reconstruction is accurate up to an unknown scale factor.

We propose to use the object detections computed from both images, D_i and D_j, as robust view-independent features. Let $O_i = \{o_{i,1}...o_{i,n}\}$ be the 3D bounding cuboids that correspond to the 2D detections in D_i. These are not directly observed, but rather must be inferred by our method. Each $o_{i,k} = \{R_{i,k}, t_{i,k}, s_{i,k}\}$ where $R_{i,k}$, $t_{i,k}$ and $s_{i,k}$ are the rotation (roll, pitch yaw), translation (x, y, z), and scale (length, width, height) of object $o_{i,k}$ respectively. $R_{i,k}$ and $t_{i,k}$ are in the camera's frame of reference. By inferring O_i and O_j in each view, and then determining object correspondences between the two views, we can recover the camera transformation C_j.

A key challenge of this problem is that the combination of camera pose, object correspondence, 3D object pose and

object scale results in a very large search space. To address this, we propose an inference strategy that works as follows. We begin by performing single-view inference on each view separately, in which we draw samples for O_i and O_j. The score of each sample depends on several factors: the scale and pose of each object should be such that the reprojection of its 3D bounding cuboid onto the image plane aligns well with the detection bounding box; the orientation of each object should be closely aligned with the major axes; the scale of each object should be consistent with the known scale distribution for its object category; objects should be contextually coherent (i.e. tables and chairs tend to lie on the same plane); the relative depth of objects should be consistent with the sparse point cloud. In order to measure consistency with the point cloud, we sample a latent scaling factor ψ, which we use to scale the points. Under good alignment, points in the point cloud should fall inside the objects' bounding cuboids. In order to sample efficiently, we draw cuboids along the ray extending from the camera center through the top center point of the detected bounding box. Justification for using the top center point will be given later. We also restrict the cuboids' orientations to those that are aligned with the major axes.

Having sampled two sets of 3D objects using the information from each view independently, we next reason about object correspondences $\Phi = \{\phi_{u,w}\}$ where $\phi_{u,w}$ indicates whether object $o_{i,u}$ corresponds to $o_{j,w}$. For each object $o_{i,k}$, we compose a spatial descriptor that encodes the identity of $o_{i,k}$ based on the relative pose of other objects in O_i. These descriptors allow us to gauge the spatial similarity between $o_{i,u}$ and $o_{j,w}$ which then allows us to give a high score to a set of correspondences Φ in which the corresponding members have high spatial similarity.

Given a single pair of corresponding objects $\phi_{u,w}$, and their pose relative to the camera, we can fully constrain the camera transformation. Estimating C_j in this way goes a long way towards narrowing down the possible value of C_j, but there is much room for improvement. Until now, the geometry of objects O_i and O_j are estimated from a single view. Using established correspondences between objects in the two views, we apply a final inference step that jointly searches over C_j and objects O_i with the goal of minimizing the object reprojection error in both views.

B. Problem Statement

For a single view i, we use the energy function $g(\psi_i, O_i, D_i, A_i, F_i)$ to capture the overall coherence of the latent cuboids O_i, the latent scaling factor ψ_i of the sparse point cloud F_i, and the detected quantities: 2D object detections D_i, major axes A_i, and the point cloud F_i. The correspondence problem is then defined as

$$\Phi^* = \underset{\Phi}{\operatorname{argmax}}\ g(\psi_i, O_i, D_i, A_i, F_i) \\ g(\psi_j, O_j, D_j, A_j, F_j) h(\Phi, O_i, O_j) \tag{1}$$

where $h(\Phi, O_i, O_j)$ scores the correspondence Φ between the two sets of bounding cuboids O_i and O_j. Notably, Φ is independent of object detections, major axes, and point clouds once the cuboids are given.

C. Single View Inference

At the single view inference stage our goal is to produce samples of O_i and O_j for both views i and j. A sample O_i is scored using the function g, which is factorized as follows

$$g(\psi_i, O_i, D_i, A_i, F_i) = \left(\prod_{k,l} g_X(o_{i,k}, o_{i,l}) \right) \\ \prod_k g_S(s_{i,k}) g_D(o_{i,k}, d_{i,k}) g_A(R_{i,k}, A_i) g_F(\psi_i, o_{i,k}, d_{i,k}, F_i) \tag{2}$$

$g_S(s_{i,k})$ measures whether the scale $s_{i,k}$ of the cuboid $o_{i,k}$ conforms to a known scale distribution. For simplicity, we check if the length, width, and height of the cuboid fall within an acceptable range for the known object category. If they do, then g_S is 1; otherwise, g_S is 0. To better exploit the covariance between scale dimensions, we could alternatively model the known scale distribution as a Gaussian or a mixture of Gaussians, and then use the likelihood of $s_{i,k}$ as the value of g_S.

$g_D(o_{i,k}, d_{i,k})$ measures alignment between a cuboid's projection in the image plane and its detected bounding box. We define it as

$$g_D(o_{i,k}, d_{i,k}) = \frac{1}{\varepsilon\left(r(o_{i,k}), d_{i,k}\right) + 1} \tag{3}$$

where ε is a novel object reprojection error formulated as

$$\begin{aligned} \varepsilon\left(r(o_{i,k}), d_{i,k}\right) = &|left\left(r(o_{i,k})\right) - left(d_{i,k})| + \\ &|top\left(r(o_{i,k})\right) - top(d_{i,k})| + \\ &|right\left(r(o_{i,k})\right) - right(d_{i,k})| \end{aligned} \tag{4}$$

Here $r(o_{i,k})$ is the projected bounding box of $o_{i,k}$; $left$, top, and $right$ give the left, top, and right sides of the bounding box respectively. Notably, we do not use the bottom boundary here, since the bottom portions of objects are often occluded, and object detectors are typically trained to only identify the visible portion of objects. Since the top of objects tend to be non-occluded, when sampling $o_{i,k}$ we draw cuboids whose top face is along the ray extending from the camera center through the top-center point of $d_{i,k}$. The depth of the cuboid can be estimated by sampling the object's width and length from a known scale distribution, and choosing a depth such that the reprojection error is minimized.

$g_A(R_{i,k}, A_i)$ measures the alignment between the rotation $R_{i,k}$ of object $o_{i,k}$ and the major axes A_i. We follow [21] to extract three orthogonal vanishing points from the image, and recover three orthogonal unit vectors which are the major axes of the room layout. We represent A_i as a rotation

between the standard axes in the camera's reference frame and the detected major axes. $g_A(R_{i,k}, A_i)$ is then defined as

$$g_A(R_{i,k}, A_i) = \frac{1}{angle(R_{i,k}, A_i) + 1} \tag{5}$$

where $angle$ gives the angle between the object's rotation and the major axes.

$g_F(\psi_i, o_{i,k}, d_{i,k}, F_i)$ evaluates the agreement between the cuboid $o_{i,k}$ and a scaled version of the sparse point cloud F_i, where the scaling factor is ψ_i. We denote the scaled point cloud as $\psi_i F_i$. Then, we have

$$g_F = \begin{cases} \omega & \text{if } \exists q \in \psi_i F_i : q \text{ is inside } o_{i,k} \text{ and } d_{i,k} \\ 1 & \text{otherwise} \end{cases} \tag{6}$$

where $\omega > 1$ is a constant. Intuitively, g_F is higher if the cuboid $o_{i,k}$ contains at least one point $q \in \psi F_i$ such that the projection of q in the image plane is inside the detected bounding box $d_{i,k}$. We find that it is incorrect to try to maximize the number of points contained inside each cuboid since many points that fall inside the bounding box $d_{i,k}$ do not actually lie on the object, but rather on the floor or on surfaces behind the object.

$g_X(o_{i,k}, o_{i,l})$ measures the contextual coherence of two objects in a scene. We define it as

$$g_X(o_{i,k}, o_{i,l}) = \frac{1}{\mathbb{1}_{copl(o_{i,k}, o_{i,l})} bot(o_{i,k}, o_{i,l}) + 1} \cdot \frac{1}{\mathbb{1}_{sep(o_{i,k}, o_{i,l})} overlp(o_{i,k}, o_{i,l}) + 1} \tag{7}$$

Here, $copl$ is true if two given objects are usually coplanar (lie on the same surface), and sep is true if two objects are usually spatially separated. For example, tables and chairs tend to be coplanar, whereas bottles and chairs are usually not. Monitors and keyboards are usually spatially separated, whereas tables and chairs are often not (i.e. when a chair is tucked under a table, their bounding cuboids overlap). bot measures the distance between the bottom faces of two bounding cuboids, and $overlp$ measures the amount of overlap between two cuboids.

D. Correspondence Inference

Given cuboid samples O_i and O_j, our goal now is to compute a set of possible correspondences $\Phi = \{\phi_{u,w}\}$ where each $\phi_{u,w}$ is 1 if $o_{i,u} \in O_i$ corresponds with $o_{j,w} \in O_j$, and 0 if they do not correspond.

Let $O_i^u = \{o_{i,1}^u ... o_{i,n}^u\}$ represent the pose of cuboids O_i in the reference frame of $o_{i,u}$. We can then compare O_i^u and O_j^w to measure the spatial difference $\Lambda_{u,w}$ between $o_{i,u}$ and $o_{j,w}$. A higher spatial difference means two objects are more spatially dissimilar in terms of their relation to other objects in the scene. One caveat is that for any $o_{i,u}$ and $o_{j,w}$, their orientations with respect to other objects may not be consistent, even if they do in fact correspond. While we could rely on sampling to eventually draw a consistent orientation for both objects, this would drastically increase the number of samples needed. Instead, we search over all plausible axis-aligned orientations of $o_{j,w}$ and use the one that minimizes $\Lambda_{u,w}$.

Comparing O_i^u and O_j^w requires us to form correspondences between their elements. To this end, we use the Hungarian algorithm [22], which solves the assignment problem in polynomial time. It requires as input a cost of assignment between each pair of objects. We use the distance between each $o_{i,k}^u$ and $o_{j,l}^w$ as the assignment cost. We increase the cost if the detection labels of $o_{i,k}^u$ and $o_{j,l}^w$ do not agree to discourage them from corresponding. The Hungarian algorithm returns the total cost of making a set of correspondences. We can simply use the total cost as $\Lambda_{u,w}$.

Once we have computed $\Lambda_{u,w}$ between every pair of objects in the two views. We can use them to find a set of correspondences Φ such that $\Lambda_{u,w}$ is low for every $\phi_{u,w} \in \Phi$. This is achieved by running the Hungarian algorithm again, this time using $\Lambda_{u,w}$ as the cost for assigning $o_{i,u}$ to $o_{j,w}$. Note that the correspondences obtained in this way are not our final answer, since the Hungarian algorithm will try to make as many correspondences as possible. Given the large viewpoint change between views, it is very likely for some objects to only appear in one of the views. In such cases, non-covisible objects should not be allowed to correspond. To this end, we filter the correspondences Φ as follows.

We define u' and w' such that $\phi_{u',w'} \in \Phi$ is 1 and $\Lambda_{u',w'}$ is minimized. This is our best scoring correspondence with the least spatial difference. We then form $O_i^{u'}$ and $O_j^{w'}$ and set $\phi_{u,w} = 0$ if $\|o_{i,u}^{u'} - o_{j,w}^{w'}\|_2 < \tau$, where τ is a threshold. This leaves us with our final correspondence estimate Φ.

After the above filtering step, the remaining correspondences are referred to as inliers. To score our correspondence estimate, we define the function $h(\Phi, O_i, O_j)$ in equation 1 to be equal to the number of inliers. The proportion of detected objects that are inliers can be leveraged to determine whether a loop closure exists, which is useful when incorporating our method into a full SLAM pipeline. However, in this work we assume that the given viewpoints overlap, allowing us to focus solely on the pose estimation aspect of the problem. We leave the development and evaluation of the full loop closure and relocalization tasks to future work.

In our implementation, we run single view inference a fixed number of times. For each pair of O_i and O_j we sample, we compute Φ, resulting in many samples of (O_i, O_j, Φ). We denote the highest-scoring correspondence as Φ^*, which we use for the subsequent step.

E. Camera Pose Refinement

Let ϕ_{u^*,w^*}^* be the inlier of Φ^* with the lowest spatial difference. We can then use o_{i,u^*} and o_{j,w^*} to fully constrain a 6-DOF camera transformation C_j. There is however much

improvement to be made to the estimate. So far, the geometry of O_i and O_j are both inferred from a single viewpoint. We now have the opportunity to use the correspondence inliers to impose multi-view constraints and further improve our estimate. We use Metropolis-Hastings MCMC sampling [23] to jointly search over the space of C_j and O_i, with the objective of minimizing the following reprojection error for all objects $o_{i,u}$ that have a correspondence.

$$\epsilon = \varepsilon(r(o_{i,u}), d_{i,u}) + \varepsilon(r(o_{i,u}, C_j), d_{j,w}) \tag{8}$$

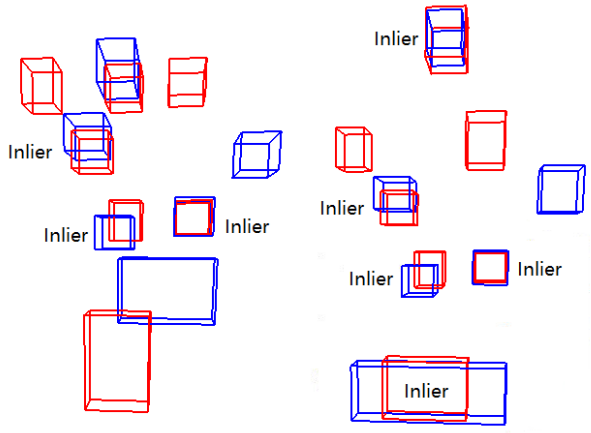
where $r(o_{i,u})$ is the projected bounding box of $o_{i,u}$ in the reference image I_i, and $r(o_{i,u}, C_j)$ is the projection of $o_{i,u}$ in image I_j of the second viewpoint.

F. Inlier Optimization

Due to the size of the sample space, we find that our implementation often does not draw enough samples to sufficiently cover the space of O_i and O_j. The consequence is that there may be very few inliers due to the samples O_i and O_j being overly dissimilar. Figure 2(a) illustrates this problem. The blue cuboids are O_i and the red cuboids are O_j. The bottom-most object should be an inlier, but they are barely overlapping. Scenarios like this are common, since the cues we use only weakly constrain the scene geometry. In other words, many different layouts can explain the detected bounding boxes and the sparse point cloud. A lack of inliers critically hinders the ability of our algorithm to judge the correctness of a sample, since our scoring function in equation 1 depends directly on the number of inliers. Furthermore, a small inlier set means that less constraints are available in the subsequent refinement step, which leads to poorer final estimates of C_j. To combat this problem, we propose the following approach.

Suppose we have drawn samples of O_i and O_j, and have used them to compute Φ. We have also found u', w' exactly as we have described previously. Further, suppose $\phi_{u,w} = 1$ for some u and w, but $\|o_{i,u}^{u'} - o_{j,w}^{w'}\|_2 > \tau$, meaning that $\phi_{u,w}$ should now be set to 0. However, this time, before setting $\phi_{u,w}$ to 0, we intervene and make an attempt to search for new values of $o_{i,u}$ and $o_{j,w}$ such that $\|o_{i,u}^{u'} - o_{j,w}^{w'}\|_2$ is reduced.

We use Metropolis-Hastings MCMC to search over the space of $o_{i,u}$ and $o_{j,w}$. At each iteration, we express their pose relative to $o_{i,u'}$ and $o_{j,w'}$ and use $\|o_{i,u}^{u'} - o_{j,w}^{w'}\|_2$ as the cost function, in an attempt to minimize it. If, at the end of a fixed number of iterations, $\|o_{i,u}^{u'} - o_{j,w}^{w'}\|_2$ is still greater than τ, then we can be more confident that it is indeed an outlier. Otherwise if the value becomes less than τ, we consider it as an inlier and allow it to be used for the camera pose refinement step. Figure 2(b) illustrates the cuboids after inlier optimization. We see that the bottom object is now an inlier. The procedure we have described here is used to re-assess all correspondences before they are filtered out.



(a) Before optimization (b) After optimization

Figure 2. The optimization step is crucial for increasing the number of correspondence inliers. This in turn allows our algorithm to more easily identify good scene hypotheses and to better constrain the camera pose transformation.

IV. Data

To evaluate our approach, we extract 47 image pairs from 9 different video sequences of the Sun3D dataset [1]. These exhibit a variety of scenes, including classrooms, office areas, lounges, meeting rooms, and corridors. The image pairs undergo translations of up to 6 meters and rotations of up to 127 degrees. We leverage the available camera trajectory to thin down the number of frames, ensuring that the camera has translated at least 10cm or has rotated by at least 10 degrees. We then use an automated system to extract pairs from the thinned frames, keeping only pairs for which the two views overlap based on the camera trajectory.

We pre-process the images we have extracted by computing object detection, major axes, and structure from motion (SFM). Detection is performed using Faster-RCNN [6], and major axes are detected using [21]. To compute SFM centered on an image I_i, we take a sequence of images $I_i - k, ...I_i, ...I_i + k$ where $k = 12$, and use them as input to Bundler [24].

To ensure proper evaluation of our method, we prune the image pairs using the following criteria. A pair must have at least four covisible objects that are correctly detected, and contain at least one SFM feature point within each detected bounding box. Objects whose bounding boxes are truncated along the left, top, or right edge of the image are disqualified. Recall that our method does not rely on the bottom edge of bounding boxes so truncation along the bottom is acceptable. Figure 3 shows the distribution of object categories in our dataset.

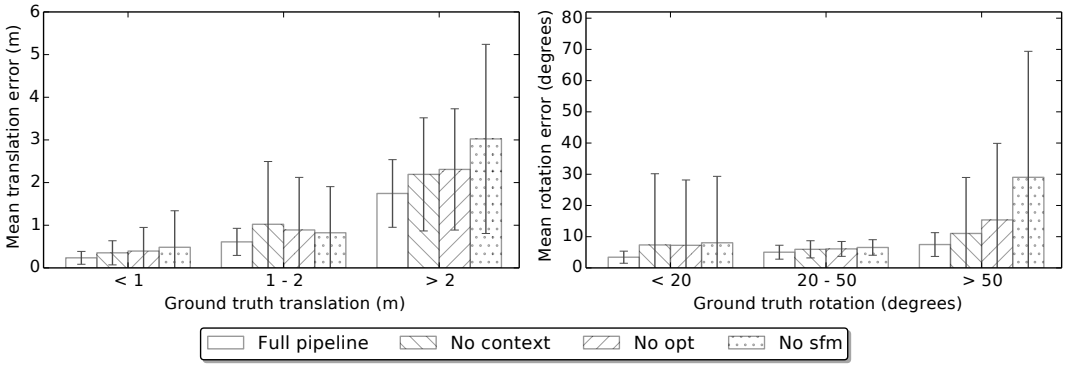


Figure 4. Translation (left) and rotation (right) error as a function of ground truth baseline between images for four variants of our method. Error bars indicate one standard deviation. In all cases our full pipeline is superior.

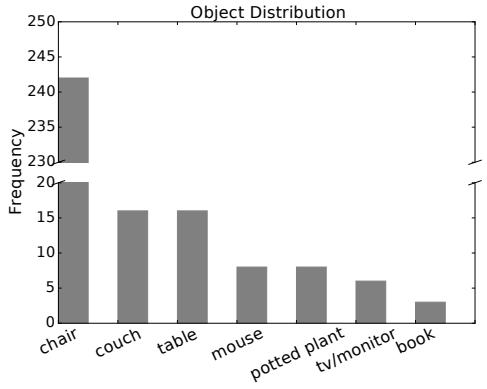


Figure 3. The distribution of object categories in our dataset.

Table I
DIRECTIONAL CORRECTNESS

Dimension	Accuracy
x	0.87
y	0.94
z	0.98

Table II
CORRESPONDENCE PERFORMANCE

Precision	Recall
0.87	0.75

V. RESULTS

A. Translational and Rotational Errors

Figure 4 shows the mean translational and rotational errors of the camera pose estimate compared to the ground truth. Here, we aim to measure the importance of context, the inlier optimization step, and the constraints imposed by the SFM point cloud. We run our full pipeline, as well as three additional runs where we disable one of these components.

From the figure, we see that our full pipeline is superior in all cases. Turning off context removes an important constraint on the scene geometry. Since the camera is often pitched slightly downward when viewing a set of objects, the sampled objects are often floating in air (too close) or below the ground (too far). Coplanarity is an important constraint that alleviates some of these problems. We also find that the non-overlap constraint is effective in rejecting implausible samples in clutters of close-by objects.

The optimization step is important since it has a big impact on the number of inliers. In some cases, it doubles or even triples the number of inliers. Without the optimization step, incorrect samples often have the same number of inliers as the correct samples, which hinders the scoring function's ability to reward the correct sample.

Like context, the SFM point cloud is an important regularizer that prevents highly unlikely scene layouts. It is interesting to note that using only context without SFM and vice versa does not lead to good performance. This indicates that both are fairly weak cues. The utility of the SFM is limited by the difficulty in determining whether a point actually lies on an object. Too often feature points inside an object's detected bounding box belongs to surfaces that are quite far from the object (i.e. on the wall behind the object, or on an occluding object that is very close to the camera).

B. Other Quantitative Measures

We present two additional quantitative metrics. The translation and rotational errors do not tell us whether the estimated camera transform is in the right general direction. Even if there are errors, it would be reassuring to know that the estimated camera pose is on the right track. Let $t_g = (x_g, y_g, z_g)$ be the ground truth translation and $t_e = (x_e, y_e, z_e)$ be the estimated translation. We compare the sign of each dimension (i.e. the sign of x_g is compared with the sign of x_e), and check for the agreement of the signs. Table I shows the accuracy of directional correctness for each dimension. The numbers show that our method

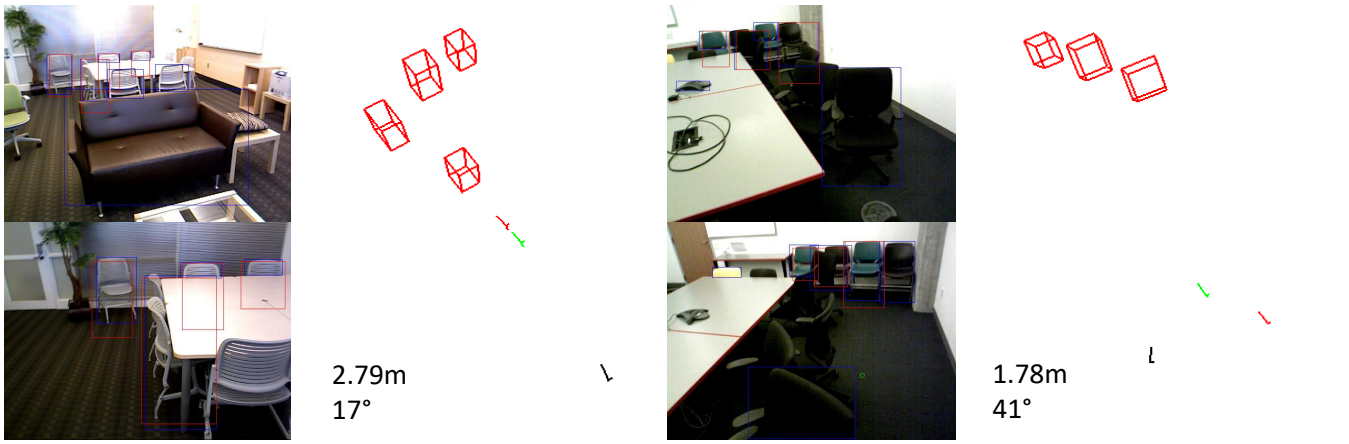


Figure 5. Visualization of our algorithm's output. The reference camera is shown in black, the ground truth camera transformation in green, and our estimate in red. Estimated bounding cuboids are drawn in red. Object detection and reprojection of cuboids are shown in blue and red respectively on the RGB image. We also indicate the ground truth translation and rotation. The camera estimate of the scene on the right has larger errors due to incorrect object correspondences.

almost always moves the camera estimate in the correct direction.

We also evaluate the correspondence precision and recall, which we show in Table II. High precision is desirable since incorrect correspondences impose incorrect constrains on camera geometry. Recall indicates what proportion of covisible objects are identified. A lower recall means that the algorithm is not taking full advantage of all the covisible objects when estimating camera pose.

C. Qualitative Assessment

Figures 1 and 5 illustrate our algorithm's reconstruction for a variety of scenes. In Figure 1 we see our system handling 5.9 meters of translation and 110 degrees of rotation. In Figure 5 on the left, we show our method handling a very large forward translation of 2.79 meters. On the right, we show an estimate with larger errors due to the lack of recognized inliers and one of the object correspondences being incorrect.

VI. Conclusions

We have described a method for robustly estimating camera pose under large viewpoint changes using RGB images. By combining multiple cues, including object detection, vanishing points, structure from motion (SFM), and object-to-object context, we are able to formulate an efficient sampling procedure to rapidly cut down a high-dimensional search space.

In future work, we intend to integrate our object-based relocalization system with a modern appearance based visual odometry system, allowing consistent mapping of both detailed geometry and semantic objects over large spaces. While semantic objects provide a strong viewpoint invariance, they are found more sparsely than local texture patches. Consequently, an object-based approach is prone to low numbers of observed objects, which makes disambiguation of places challenging. This motivates a hybrid approach that integrates both local appearance features and objects, as well as active approaches that direct the robots gaze towards object-rich regions.

We intend to evaluate our system on more indoor and outdoor urban environments. In particular we would like to demonstrate its ability to be invariant to lighting conditions, weather, and other seasonal changes since object detectors are robust to these variabilities. Our method could also be integrated into higher-level tasks such as manipulation and natural language understanding, which often depend on having semantically meaningful scene reconstructions.

References

[1] J. Xiao, A. Owens, and A. Torralba, "Sun3d: A database of big spaces reconstructed using sfm and object labels," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 1625–1632.

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

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

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

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

[6] 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.

[7] H. G. Barrow and J. M. Tenenbaum, "Computational vision," *Proceedings of the IEEE*, vol. 69, no. 5, pp. 572–595, 1981.

[8] G. Dudek and M. Jenkin, *Computational principles of mobile robotics*. Cambridge university press, 2010.

[9] A. Torralba, K. P. Murphy, W. T. Freeman, and M. A. Rubin, "Context-based vision system for place and object recognition." in *ICCV*, vol. 3, 2003, pp. 273–280.

[10] S. Gould, T. Gao, and D. Koller, "Region-based segmentation and object detection," in *Advances in neural information processing systems*, 2009, pp. 655–663.

[11] S. Y. Z. Bao, M. Sun, and S. Savarese, "Toward coherent object detection and scene layout understanding," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010, pp. 65–72.

[12] S. Y. Bao, M. Bagra, Y.-W. Chao, and S. Savarese, "Semantic structure from motion with points, regions, and objects," in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 2012.

[13] N. Atanasov, M. Zhu, K. Daniilidis, and G. Pappas, "Localization from semantic observations via the matrix permanent," *International Journal of Robotics Research*, vol. 35, pp. 73–99, 2016.

[14] 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 *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, 2013.

[15] J. Civera, D. Gálvez-López, L. Riazuelo, J. D. Tardös, and J. M. M. Montiel, "Towards semantic slam using a monocular camera," in *Proceedings of International Conference on Intelligent Robots and Systems (IROS)*, 2011, pp. 1277–1284.

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

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

[18] S. Bowman, N. Atanasov, K. Daniilidis, and G. J. Pappas, "Probabilistic data association for semantic slam," in *Proceedings of the International Conference on Robotics and Automation*, 2017.

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

[20] J. Li, D. Meger, and G. Dudek, "Context-coherent scenes of objects for camera pose estimation," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2017.

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

[22] 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.

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

[24] N. Snavely, S. Seitz, and R. Szeliski, "Photo tourism: Exploring image collections in 3d. acm transactions on graphics," *ACM Transactions on Graphics*, 2006.