# Context-Coherent Scenes of Objects for Camera Pose Estimation

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

***Abstract*— We propose an approach to vision-based pose estimation using object recognition and identity. Whereas feature based scene recognition and pose estimation methods are well established as effective means for estimating motion and recognizing locations, feature-based methods depend critically on the detection of common local features from one view of a scene to another. We focus on place recognition and pose change estimation in the context of large changes in viewing position, even to the extent that no common surfaces are seen between the two views. Our approach is based on using object identities and their inter-relationship to compute pose change. An important secondary outcome of our method is that it simultaneously infers the 3D poses of objects in the scene that are used as features. Such an object-based approach is inspired by a vast literature on human perception and has the potential for great robustness, albeit at the expense of accuracy. We propose a formulation of the problem using pairwise contextual constraints and develop an efficient algorithmic solution. We validate the approach and quantify its performance using the publicly available TUM SLAM dataset [1].**

## I. Introduction

This paper describes a new approach for camera pose estimation that is based upon inference of the 3D poses of objects in the observed scene. Reasoning about object poses introduces strong opportunities to use prior knowledge, such as the expected size of each object and contextual constraints that govern the relative layout of collections of nearby objects. As seen in the example provided by Figure 1, our method utilizes only RGB images captured from distinct locations, and based on object detections from a learned appearance model, we estimate both the 3D pose of each object in the scene as well as the relative 3D pose of the cameras.

We focus on wide-baseline camera pose estimation, as it is particularly well-suited to benefit from object-level reasoning. Unlike local features such as SIFT [2], where finding correspondences between images depends on stable local appearance, whole-object recognizers trained using large modern datasets are often able to recognize objects across a wide range of viewpoints [3], [4]. Whether we see the hood or the trunk, we still recognize the presence of a car. The wide-baseline pose estimation problem is one of the most difficult remaining aspects of visual SLAM. It is also a key element of global visual localization, collaborative multi-robot mapping and camera sensor networks.

As is common in visual navigation, finding correct correspondences between images is pre-requisite to effective pose estimation. False correspondences may arise when multiple objects of the same category are detected in an image, or when false positives are present. Thus, it is crucial that we select correspondences that are consistent with the scene geometry, which in our case consists of camera and object poses. The core element that enables our method to estimate 3D object poses from 2D image evidence (in the form of rectangular bounding boxes), is prior knowledge about the geometric relationship between objects, commonly referred to as object-to-object context. Here, we utilize a detailed geometric context model that we have previously proposed [5], which helps us to predict the 3D pose of objects from different camera views, and in turn helps to reject inconsistent correspondences. Our results demonstrate that this approach is highly effective in selecting correct correspondences for the benchmarks considered.

Beyond being a useful constraint to estimate camera motion, object pose in 3D is a highly desireable map representation. Recovering this level of knowledge has been the goal of previous efforts integrating many sensing modalities [6], but here we recover object maps using only a pair of RGB images. The remainder of this paper will describe the

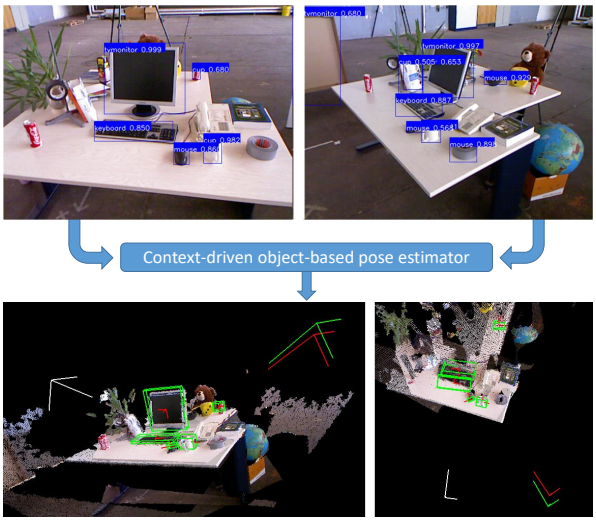



Fig. 1: Our method uses 2D bounding box detections of objects from two far-apart viewpoints to predict the 6-dof camera transformation between views. Top row shows the objects detected from two views. Bottom row shows the reference camera (white) the ground truth camera transformation (green), and the estimated camera transformation (red). Our algorithm also estimates 3D bounding cuboids for each detected object.

[1]Mobile Robotics Lab, McGill University, Montreal, Canada {jimmyli,dmeger,dudek}@cim.mcgill.ca.

This work was supported by the Natural Sciences and Engineering Research Council (NSERC) through the NSERC Canadian Field Robotics Network (NCFRN).

TABLE I: Contextual model for monitor-keyboard

| FGD dimensions | value | weights |
|---|---|---|
| left-left-x | | 0 |
| left-right-x | -0.04 m | 1 |
| right-left-x | 0.04 m | 1 |
| right-right-x | | 0 |
| top-top-y | | 0 |
| top-bottom-y | | 0 |
| bottom-top-y | | 0 |
| bottom-bottom-y | 0 | 5 |
| front-front-z | 0.06 m | 3 |
| front-back-z | | 0 |
| back-front-z | | 0 |
| back-back-z | | 0 |
| relative-yaw | $\pi$ rad | 2 |

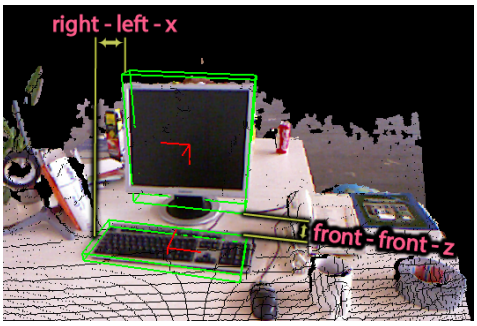



Fig. 2: Illustration of two dimensions of a face-centric geometric descriptor (FGD), shown by the green bounding box and associated red coordinate frame.

extensive related literature, outline our method and describe experiments and results on sequences of images from a publicly available dataset of indoor office images.

## II. Related Work

Many authors have addressed object recognition, pose estimation and the problem of relating object identity to the underlying scene. This ensemble of interlinked problems represents one of the biggest research topics in both computer vision and computational perception, as well as robotics, and has been considered in the literature in both robotics and computer vision [7], [8]. 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]. The relationship between object identity and scene structure was exploited for the purposes of image segmentation by Gould et al. 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 that there should be an exploitable connection between different levels of abstraction and different computational modalities goes back to some of the earliest results in computational vision [7].

The most closely related work to our own are methods that perform pose estimation using objects as a feature representation [12], [13], [14], [15], [16], [17]. We have been inspired by several of the geometric constructions, Bao's relation between image bounding boxes to 3D cuboids [12], as well as the focus on data association as a core inference step in Mu *et a.* [17]. We are not aware of any previous method that has used detailed pairwise spatial context models within the pose estimation reasoning, which is our central focus.

## III. Relevance-aware Context

### A. Face-centric Geometric Descriptors (FGDs)

In this work we make heavy use of context to regularize our estimates of 3D object poses. To characterize the 3D spatial relationship between two objects, we use face-centric geometric descriptors (FGDs) which we have introduced previously [5]. An object $A$ is represented by its oriented bounding cuboid, where the orientation is determined by its front face. We extract 6 points to summarize the cuboid's geometry, with one point on the center of each face. We denote these points as $A$-left, $A$-right, $A$-front, $A$-back, $A$-top, and $A$-bottom. To represent the spatial relationship between $A$ and a second object $B$, we express both objects in the reference frame of $A$. Here $A$ is the "anchor" and $B$ is the "follower". We then construct feature dimensions of the FGD of $A$-$B$ by subtracting corresponding positional elements between points on $A$ from points on $B$. For instance, we can construct the dimension right-left-x by subtracting the x element of A-right from that of B-left. Figure 2 illustrates two of the dimensions of an FGD for the monitor-keyboard relationship. To describe relative orientation, we add relative yaw to the FGD. We do not model roll and pitch since objects in indoor scenes are typically aligned with the gravity vector. The full list of FGD feature dimensions are shown in the first column of Table I.

### B. Context Modelling

Our context model describes the geometric spatial relationship between two objects that are represented by their oriented bounding cuboids. A context model for an anchor object $A$ and follower object $B$, denoted as $\Gamma^{AB}$ has two components:

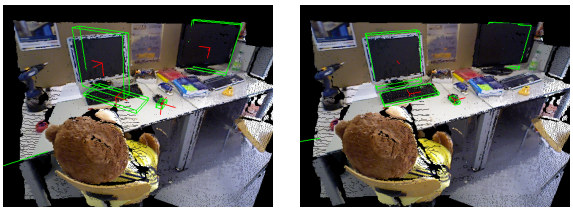
- An FGD vector $\Gamma^{AB}_{fgd}$ describing the typical relative position of the two objects
- A weight vector $\Gamma^{AB}_{w}$ of the same dimension as the FGD

Then, given an FGD vector $X_{c_i,c_j}$ representing an observed relationship between a cuboid $c_i$ and a cuboid $c_j$, we can evaluate how well it conforms to $\Gamma^{AB}$ by computing

$$\Psi_{cont}(\Gamma^{AB}, X_{c_i,c_j}) = \frac{1}{\Gamma^{AB}_{w} \cdot |\Gamma^{AB}_{fgd} - X_{c_i,c_j}| + 1} \qquad (1)$$

A higher value of $\Psi_{cont}(\Gamma^{AB}, X_{c_i,c_j})$ indicates better conformity to the context model.

Table I provides an example of a hand-coded context model describing the monitor-keyboard relationship, where monitor is the anchor. Column 2 of Table I shows the values for each dimensions, and column 3 shows the weights. Some



(a) No relevance (b) With relevance

Fig. 3: Without modelling contextual relevance (a), the keyboard and mouse on the desk to the left incorrectly try to align with the monitor on the desk to the right. Our relevance-aware context modelling (b) resolves this problem and leads to a more accurate 3D reconstruction.

of the value entries are empty since their corresponding weight is zero and therefore has no impact on the contextual score. A higher weight means discrepancies in that dimension will affect the contextual score more severely. The highest weight is given to bottom-bottom-y which means above all else, we insist that the bottom of the two objects are co-planar (sitting on the same surface).

While we have shown that these context models can be automatically learned from data [5], in this work we find it sufficient to manually construct them. FGDs offer a natural way for humans to specify relationships in terms of object boundaries.

### C. Contextual Relevance

One difficulty with using inter-object relationships is that many contextual cues are conditionally applicable. Consider the scene presented in Figure 3a. The keyboard and mouse belonging with the monitor on the left try to align themselves to face the monitor on the other desk to the right. This in turn causes the monitor on the left desk to turn to better accommodate the mouse and keyboard in front of it. As a result, we have a set of poorly fitted bounding cuboids.

To gain better robustness to far away objects, we introduce the notion of contextual relevance. Let

$$\psi = \Psi_{cont}(\Gamma^{AB}, X_{c_i,c_j})$$

Then a relevance-aware contextual score is defined as

$$\Psi_{rel}(\Gamma^{AB}, X_{c_i,c_j}) = \begin{cases} \psi & \text{if } \psi > \Gamma^{AB}_{\tau} \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

where $\Gamma^{AB}_{\tau}$ is the relevance threshold. Under the relevance-aware scoring scheme, the monitor-keyboard in our example will receive a low value of $\psi$ due to the spatial constraints imposed by their 2D bounding boxes. Thus, their relevance-aware score $\Psi_{rel}$ will be zero, effectively turning off the contextual relationship between them. This prevents our object pose estimator from favoring the result in 3a, and achieves a more accurate result in 3b.

## IV. Method

Given a pair of RGB images, $I_r$ and $I_t$, taken from a reference camera pose $P_r$ and a transformed camera pose $P_t$ respectively, we aim to estimate the relative 6-dof transformation $T$ from $P_r$ to $P_t$. We use Faster-RCNN [3] to detect objects in both images, and for each detected bounding box, we use the known scale of the object to sample a collection of bounding cuboids in 3D space whose projected bounding box aligns well with the detection. We define a scene as a collection of bounding cuboids, where each cuboid is generated from a different bounding box. Since we have multiple cuboid hypotheses for each bounding box, we end up with many scene hypotheses.

Having a set of scene hypotheses $S_r$ for $I_r$ and $S_t$ for $I_t$. We search for the pair of scenes $s_r \in S_r$ and $s_t \in S_t$ whose bounding cuboid layouts are maximally spatially consistent. By corresponding the cuboids in $s_r$ and $s_t$, we can use their relative transformations to estimate the camera transformation $T$. The remainder of this section will describe each step in detail.

### A. Generating Cuboids

To generate 3D cuboid hypotheses from a detected 2D bounding box, we search along the ray extending from the camera center through the center of the bounding box. We assume that the direction of the gravity vector is known, and that objects lie on a flat surface. So the only free parameters for a cuboid hypothesis are its distance to the camera along the ray and its yaw.

Let $c = (b, p, d, y)$ be a cuboid whose distance to the camera is $d$ and whose yaw is $y$. $b$ is its corresponding bounding box, as given by the object detector, and $p$ is the bounding box of its projection into the image plane. The semantic label of $c$ is denoted by $lab(c)$ which is the same as the detected semantic label of the bounding box, denoted as $lab(b)$. For any bounding box $x$, we denote its position in the image as $left(x)$, $top(x)$, $right(x)$ and $bottom(x)$.

We begin by generating many cuboids at discrete distance and yaw intervals. We denote this initial set of cuboids for bounding box $b$ as $C_b$. Then, for each $c$ in $C_b$, we perform coordinate descent on $d$ and $y$ to maximize the following objective:

$$argmin_{d,y} |left(b) - left(p)| + |top(b) - top(p)| + |right(b) - right(p)| + |bottom(b) - bottom(p)|$$

The many initial cuboids allow our method to produce hypotheses that cover various equally valid explanations of the 3D object, such as close-up views of a narrow side versus farther views of a broad side, and symmetries such as the left or right side of a car. After optimizing every initial cuboid until it maximizes the projection agreement locally, we prune away redundant cuboids whose distance and yaw are similar up to a threshold. This leaves us with a refined set of cuboids $C'_b$, which we use in the next step.

### B. Generating Scenes

Let $B = \{b\}$ be the set of all detected bounding boxes in an image. We now generate a set of scene hypotheses where

each scene is $s = \{c_1, c_2, ...c_n\}$ such that each bounding box $b \in B$ is explained by one cuboid $c \in s$.

To efficiently generate likely scenes, we sequentially select objects based on contextual constraints. Let $c_i$, called the seed, be a cuboid hypothesis for bounding box $b_i$. Then for every other bounding box $b_o \in B \setminus \{b_i\}$, we select a cuboid $c_o \in C'_{b_o}$ such that the contextual score between $c_i$ and $c_o$ is maximized. To obtain the contextual score, we compose the FGD descriptor between $c_i$ and $c_o$, denoted as $X_{c_i,c_o}$, and use equation 2 to compute $\Psi_{rel}(\Gamma^{lab(c_i),lab(c_o)}, X_{c_i,c_o})$.

We generate a scene using every cuboid hypothesis as the seed. This gives us a large set of scenes $S = \{s_1, s_2, ..., s_m\}$. We filter out redundant scenes in $S$, keeping only ones that are sufficiently different from each other.

### C. Scene Refinement

We find it helpful to refine each scene $s \in S$ by using coordinate descent to fine-tune the pose of each cuboid $c \in s$ such that the scene is contextually coherent overall. To this end, we perform the following optimization

$$argmax_{\{\forall c: c \in s\}} \sum_{i,j \in s} \mathbb{1}_{i \neq j} \Psi_{rel}(\Gamma^{lab(c_i),lab(c_j)}, X_{c_i,c_j}) \quad (3)$$

Recall that the free parameters of a cuboid $c$ are its distance to the camera $d$ and its yaw $y$. By searching only over these two parameters, we incorporate the constraints imposed by the bounding box detection, without having to explicitly insert an additional term into the optimization.

At this point, we keep only the $k$ top scoring hypotheses based on the criteria in equation 3. This is useful for reducing the search space for the subsequent step, in which we exhaustively search over all pairs of scenes extracted from the two images.

### D. Finding Scene Correspondence

Let $S_r$ and $S_t$ be the scenes generated from images $I_r$ and $I_t$ respectively. We now seek a scene $s_r \in S_r$ and a scene $s_t \in S_t$ such that the cuboids in $s_r$ and $s_t$ are consistent both in terms of semantic labels and 3D spatial positions.

We exhaustively consider every pair $(s_r, s_t)$. For each pair, we find all possible correspondences between the cuboids based on their semantic label. A correspondence is denoted as $q = (c_r, c_t)$ such that $c_r \in s_r$ and $c_t \in s_t$ and $lab(c_r) = lab(c_t)$. Let $Q$ be the set of all possible correspondences.

Next, we use a sampling-based approach to search for a subset of correspondences in $Q$ that are most likely to be correct. A sample consists of 3 correspondences, drawn randomly from $Q$. This gives us 3 unique cuboids $\{c_r^1, c_r^2, c_r^3\} \in s_r$ and 3 cuboids $\{c_t^1, c_t^2, c_t^3\} \in s_t$.

Having 3 correspondences allows us to construct a reference frame $F_r$ based on $\{c_r^1, c_r^2, c_r^3\}$ and a reference frame $F_t$ based on $\{c_t^1, c_t^2, c_t^3\}$. If we have sampled correct correspondences, then the relative transformation between these frames will allow us to align the remaining cuboids in the scenes. Let $cent(c)$ denote the centroid of a cuboid $c$. Then using $cent(c_r^1)$, $cent(c_r^2)$, and $cent(c_r^3)$, it is possible to form a 3-dimensional orthonormal basis. We can form one basis vector $e_1$ by normalizing $cent(c_r^1) - cent(c_r^2)$, a second basis vector $e_2$ by finding the unit vector orthogonal to the plane formed by the 3 centroid points, and a third one $e_3$ by computing the cross product of $e_1$ and $e_2$. Performing this operation for both $\{c_r^1, c_r^2, c_r^3\}$ and $\{c_t^1, c_t^2, c_t^3\}$ gives us two bases: $E_r$ and $E_t$ respectively. We can then form a frame of reference $F_r$ using $E_r$ and the centroid of $\{c_r^1, c_r^2, c_r^3\}$, as well as $F_t$ using $E_t$ and the centroid of $\{c_t^1, c_t^2, c_t^3\}$. From here, it is straightforward to find change of frame transformations $T_r$ and $T_t$ that allow us to express the poses of all cuboids in $s_r$ in the frame of $F_r$ and all cuboids in $s_t$ in the frame of $F_t$.

Let $(c'_r, c'_t)$ be a correspondence that is not contained in the sample. This correspondence is considered an inlier if

$$||T_r cent(c'_r) - T_t cent(c'_t)||_2 < \tau$$

where $\tau$ is the inlier threshold. The score of the sample is defined as

$$\sum_{(c'_r,c'_t) \in Q} \frac{1}{||T_r cent(c'_r) - T_t cent(c'_t)||_2 + 1}$$

After generating many samples, we rank the samples first by the number of inliers, then by the score, both in descending order.

### E. Estimating Camera Transformation

At this point, we can simply compute the change of frame $T_{t-r}$ from $F_t$ to $F_r$ using our highest-ranking sample, and report it as the camera transformation between the views. This works fine in most cases, but there are several edge cases that can sometimes result in unreasonable estimates. We have identified two primary sources of error. Firstly, if there are many objects of the same semantic label (i.e., many bottles lined up across a shelf), then correspondences between objects can become highly ambiguous. Clutter often makes it easy for incorrect matches to accidentally align. Secondly, if all three objects in a sample lie roughly on a line, then very small noise in the objects' poses could cause huge errors in the camera transformation estimate.

To tackle the first problem, we assume that the camera stays roughly at eye level and reject any unlikely camera transformations (i.e. camera turns upside-down). To address the second problem, we make a modification earlier in the sampling stage. If the cuboids $\{c_r^1, c_r^2, c_r^3\}$ obtained from a sample are close to being co-linear, we immediately reject the sample.

## V. Data

We evaluate our approach using two handheld SLAM sequences (`fr2/desk` and `fr3/long_office_household`) in the publicly available TUM RGB-D SLAM Dataset [1]. These sequences consist of RGB-D videos of indoor scenes with 6-dof camera poses. Note that our pipeline does not use the depth image.

Since our aim is to evaluate our algorithm's ability to handle large changes in camera pose, we filter the sequences

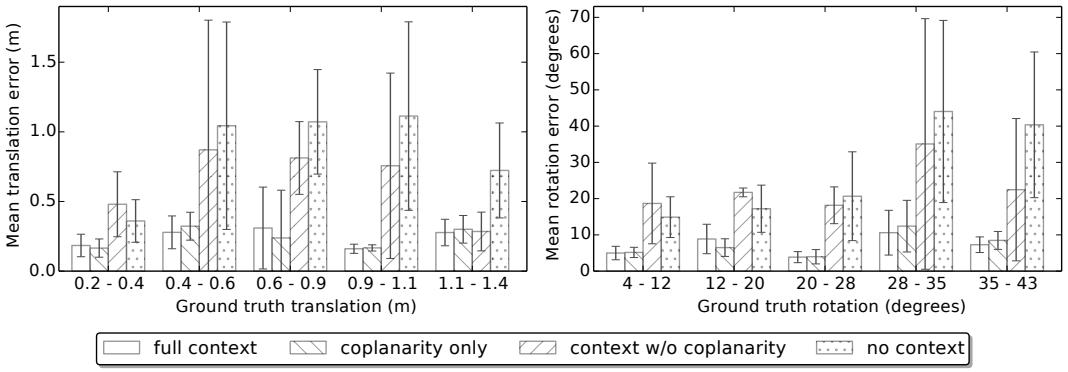



Fig. 4: Translation (left) and rotation (right) error as a function of ground truth baseline between images for four variants of our method. In all cases, error bars indicate one standard deviation. In most cases full context is superior.

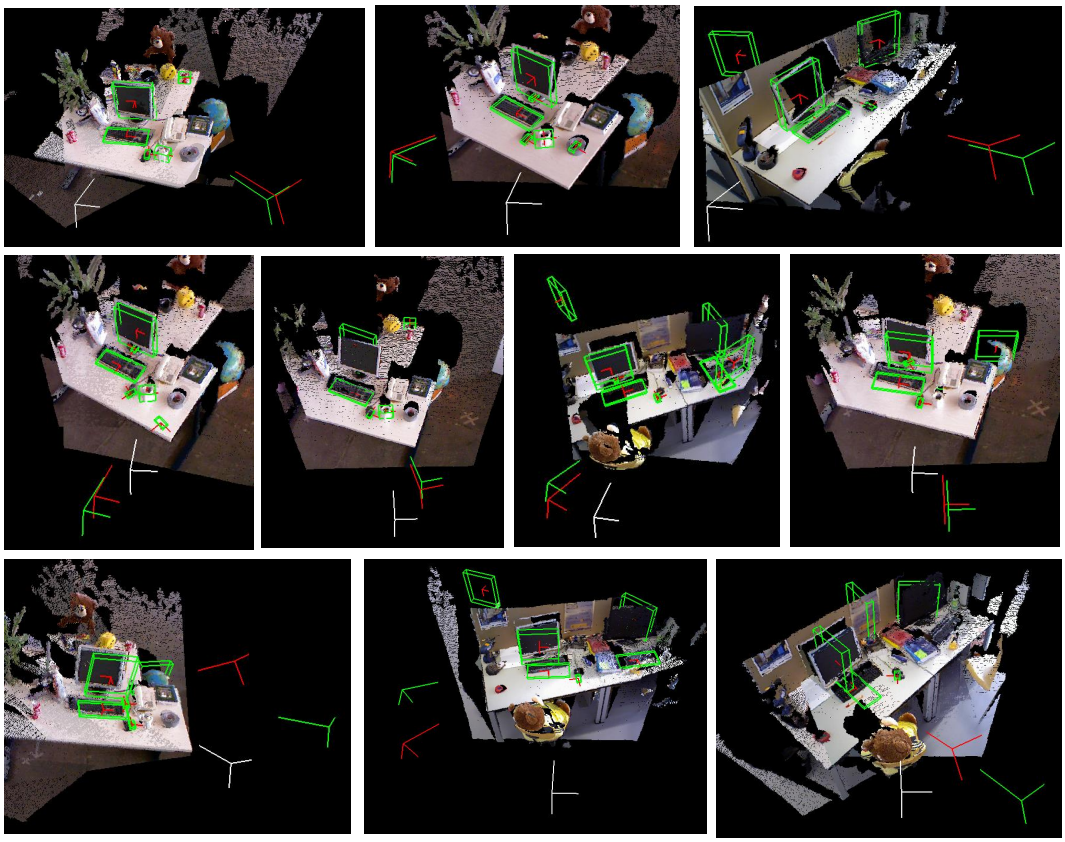


Fig. 5: Visualization of our pose estimates. The reference camera is shown in white, the ground truth camera transformation in green, and our estimate in red. The first two rows show low-error estimates for large baselines (top) and smaller baselines (middle). The third row shows incorrect estimates with large errors. Reconstructed bounding cuboids are shown in green.

such that between each consecutive frame, the camera translation is at least 30 cm or the rotation is at least 10 degrees. We take the remaining frames $f_1, ...f_n$, and perform the following operation to generate pairs of frames: for each $f_i$, we form pairs $(f_i, f_{i+1})...(f_i, f_{i+8})$. This procedure gives us a set of frames whose relative translation ranges between 0.2 and 1.4 meters, and whose relative rotation ranges from 4 to 43 degrees. For each pair $(f_i, f_j)$, we also keep the reversed pair $(f_j, f_i)$. After performing object detection using Faster-RCNN [3], we remove each pair $(f_i, f_j)$ if less than 4 objects are correctly detected in both $f_i$ and $f_j$. While this requirement of overlapping objects may seem to hinder generality, we argue that a robotics system that uses our approach could actively seek advantageous viewing angles that maximize the number of objects in view. The objects we consider in our implementation are monitor, keyboard, mouse, and cup.

## VI. Results

We have evaluated the pose estimation accuracy of our approach against the ground truth camera poses available with the dataset. Figure 4 holds results for all method variants. We experiment with keeping only co-planar contextual constraints between objects (coplanar only), removing coplanar constraints (context w/o coplanarity), and removing contextual cues altogether (no context). To analyze the error of each method as a function of the real baseline between images we group pairs into five bins for translation and five for rotation.

We would normally expect the error to increase as baseline grows, since larger baselines are more likely to lead to occlusion and different parts of the scene being visible in the in the two images. However, we see that for "full context" and "coplanarity only", no such trend is observed. This is evidence that our object-based approach is robust to large changes in viewpoint.

In general, utilizing all available contextual cues is effective, although the coplanarity aspect of the context model has proven to typically make a larger contribution to performance than the object-specific cues such as yaw alignment.

In Figure 5 we show visualizations of our pose estimation results. The reference camera is drawn in white, the ground truth camera transformation in green, and our estimated camera transformation in red. The first two rows show estimates with low errors, and the third row shows incorrect estimates with large errors. Reconstructed bounding cuboids are also shown in green.

## VII. Conclusions

We have described a method to estimate the pose of a moving camera as well as that of each of the objects in the scene based upon 2D object recognitions and several types of prior contextual knowledge. By inferring depths and angles for each object relative to each camera, we create "scenes", which are groups of objects that fit expected spatial layouts. These scene samples in each image provide sufficient geometric information to solve for the camera pose, as long as sufficiently many corresponding objects can be found. Our results demonstrate that context is an essential element in this process, as scene samples that only use independent-object geometric constraints provide much worse pose estimation. With context, we are able to estimate camera pose across significant ground truth translation and rotations, using solely object information.

Our approach is intended to interface with higher-level reasoning such as object manipulation, natural language instructions or multi-robot localization. For these purposes, the outputs of our module will seed a sensor-feedback behavior that will servo to one of our detected objects or guide the camera to gain new visual information. The 3D object information used as our map representation is ideal for this interaction. We also hope to embed our object and context constraints within a visual odometry method that utilizes both local visual features and object information. We believe that objects will prove to be useful when drastic changes such as blur, lighting or weather destroys the high resolution required for local feature matching, leading to an accurate and robust approach.

## References


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

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

[3] 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.

[4] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 779–788.

[5] J. Li, D. Meger, and G. Dudek, "Learning to generalize 3d spatial relationships," in *Proceedings of the International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 5744 – 5749.

[6] D. Meger, P.-E. Forssén, K. Lai, S. Helmer, S. McCann, T. Southey, M. Baumann, J. J. Little, and D. G. Lowe, "Curious george: An attentive semantic robot," *Robotics and Autonomous Systems*, vol. 56, no. 6, pp. 503–511, 2008.

[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, M. A. Rubin, *et al.*, "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. Glvez-Lpez, L. Riazuelo, J. D. Tards, 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.