Joint Detection, Tracking and Mapping by Semantic Bundle Adjustment

Nicola Fioraio Luigi Di Stefano
CVLab - Dept. of Computer Science and Engineering, University of Bologna
Viale Risorgimento, 2 - 40135 Bologna, Italy
{nicola.fioraio,luigi.distefano}@unibo.it

Abstract

In this paper we propose a novel Semantic Bundle Adjustment framework whereby known rigid stationary objects are detected while tracking the camera and mapping the environment. The system builds on established tracking and mapping techniques to exploit incremental 3D reconstruction in order to validate hypotheses on the presence and pose of sought objects. Then, detected objects are explicitly taken into account for a global semantic optimization *of both camera and object poses. Thus, unlike all systems proposed so far, our approach allows for solving jointly the* detection *and* SLAM *problems, so as to achieve object detection together with improved SLAM accuracy.*

1. Introduction

The visual SLAM problem concerns the ability to incrementally reconstruct the world and simultaneously localize the sensing device by means of visual cues only. In the last decade, the field has witnessed impressive advances, with effective tools available for applications such as AR [17, 23] or robot navigation and mapping [10, 24]. The classical approach builds on filtering techniques, such as the Extended Kalman Filter (EKF) [10, 11, 8]: visual features are tracked through frames and their 3D positions estimated along with the unknown camera pose. As only a small subset of the image pixels is tracked, such methods usually produce sparse maps. Alternatively, the visual SLAM problem has been tackled by a Bundle Adjustment (BA) style optimization [27] carried out on a selected subset of frames, usually referred to as *keyframes* [17, 24].

One of the most successful BA-style system is PTAM [17]. The authors propose to split the SLAM problem into two different tasks associated with parallel threads: one tracks the camera with respect to current estimates of landmark locations, the other is in charge of the global optimization over selected keyframes. As complexity grows rapidly with the number of features extracted from the environment, PTAM can be used effectively only within small workspaces. To overcome this limitation, Strasdat *et al*. [24] propose not to consider the whole set of past keyframes while tracking, but only a small subset of them, thereby achieving constant time complexity.

Recently, the advent of the Kinect, a 30Hz, low-cost, low-range RGB-D sensor, has fostered the SLAM community towards novel approaches that would optimally exploit the device. Initial proposals were mainly based on feature matching and incremental optimization [14, 13], but soon Newcombe *et al*. developed *Kinect Fusion* [23], a new dense algorithm providing outstanding performance both in terms of extremely low drift as well as smooth surface reconstruction.

While the tracking and mapping task has thus reached a certain degree of maturity, none of previous methods can seamlessly handle or derive semantic information within the visual SLAM process. Indeed, nowadays the scientific community is showing ever increasing interest for the novel field of semantic perception and mapping [12, 9, 19]. The intuition is that a partial reconstruction of the environment can improve the object detection task, so as to better handle nuisances such as occlusions, clutter and viewpoint changes, while the knowledge of object poses provides useful constraints to improve the mapping and tracking tasks. Though some interesting steps grounded on the above intuition have been made [9, 7, 4, 3], we claim that none of current proposals has really closed the *detection-SLAM* loop (see Sec. 2). Accordingly, in this paper we propose a novel framework for fully integrated SLAM and object detection, which we dub Semantic Bundle Adjustment. It features the following peculiar traits:

- it can work with both 2D and 3D sensors;
- object detection is cast as a BA-style optimization that can be integrated seamlessly into any BA-based SLAM system;
- SLAM constraints are deployed to robustify object detection, object detection constraints to improve SLAM;
- joint 6DOF object and camera poses estimation is

achieved through a novel *semantic* global optimization.

The paper is organized as follows. Next section summarizes similar works and highlights the novelty of our proposal. Then, the method is described in Sec. 3. Quantitative and qualitative results are provided in Sec. 4. Finally, Sec. 5 reports some concluding remarks.

2. Related Work

Many recent works share the idea of combining semantic knowledge and geometrical constraints for scene understanding. Unlike our proposal, most of them [12, 15, 20] perform the object detection task on a single view and hence do not enforce multi-view consistency. In particular, Ekvall *et al*. [12] do not try to estimate the exact position of detected objects, even though detection is tight to a SLAM framework. Others exploit geometric information for consistent object detection, but without deploying previously detected objects to constrain the mapping task, as instead we actually do. Vasudevan *et al*. [28] detect objects using a standard feature-based pipeline [21] and use the estimated relative poses for place representation. Meger *et al*. [22] find known objects in a map built from laser and odometry data, but again the object and camera poses are not estimated jointly.

Interesting results on joint detection and reconstruction have been reported in [9, 19]. The former introduces the notion of "cognitive loop", but detection is limited to cars and pedestrians and, moreover, strong assumptions are made about the environment and camera motion. The latter proposes joint pixel labeling and dense stereo reconstruction. They show that ambiguities in real word data can be solved by a unified approach, but the method requires calibrated cameras with small baselines. Castle *et al*. [6] detect planar objects by means of SIFT features [21] and insert them in a sparse map where feature points are tracked by a standard EKF SLAM [10]. A similar work, though extended to non-planar objects, is described in [7]. However, the detection problem is still not fully integrated into the SLAM framework, so decisions about object presence are taken using features coming from a single image. Recently, Bao *et al*. [4, 3] have proposed a *semantic* structure from motion technique which addresses the problem of estimating the camera poses and recognizing object categories from an image sequence. Though it shares similar intuitions with this work, our proposal is inherently different. First, we deal with an incremental SLAM scenario, while [4] extends the SFM paradigm and thus processes all frames at once. Also, an *external* object detector is needed in [4] to cast hypotheses and measurements, while we operate at a much lower level of abstraction: our measurements are feature correspondences established between incoming frames and a feature database. More importantly, in our proposal the object detection pipeline is not external but instead fully integrated into the SLAM framework so that object existence is inferred through a novel semantic bundle adjustment framework. Finally, our method is aimed at detection and full 6DOF pose estimation of object instances rather than category level recognition and image plane localization.

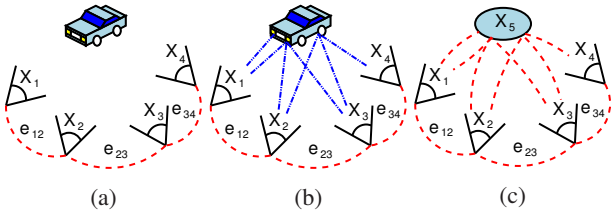


Figure 1: A standard pose graph (a) ignores any semantic information. Including into the optimization matches of object's features (b) as graph edges and the object pose as a vertex (c) to achieve object detection and improve SLAM.

3. Semantic Bundle Adjustment

Bundle adjustment is the problem of the joint estimation of a set of geometric parameters that are simultaneously optimized with respect to a cost function quantifying the model fitting error [27]. In a typical SLAM application, where the geometric unknowns are camera poses, a BA formulation allows for both tracking the sensor movement and reconstructing the environment incrementally. The cost function to be optimized is a sum of errors and can be written in the form [18]:

$$\mathbf{F}(\mathbf{x}) = \sum_{\langle i,j \rangle \in \mathcal{C}} \mathbf{e}(\mathbf{x}_i, \mathbf{x}_j, \mathbf{z}_{ij})^{\intercal} \, \Omega_{ij} \mathbf{e}(\mathbf{x}_i, \mathbf{x}_j, \mathbf{z}_{ij}) \qquad (1)$$

where $\mathbf{x} = (\mathbf{x}_1^{\intercal}, \ldots, \mathbf{x}_n^{\intercal})^{\intercal}$ is the vector of unknown parameters and $\mathcal{C}$ a set of constraints. Each index pair $\langle i, j \rangle \in \mathcal{C}$ refers to a constraint between parameter blocks $\mathbf{x}_i$ and $\mathbf{x}_j$ expressed as an error vector $\mathbf{e}_{ij} \overset{def}{=} \mathbf{e}(\mathbf{x}_i, \mathbf{x}_j, \mathbf{z}_{ij})$ with measured mean $\mathbf{z}_{ij}$ and information matrix Ω_{ij}. For instance, in a monocular SLAM scenario we might wish to track 2D features between subsequent frames, so that we are led to include in $\mathbf{x}$ both the 6DOF camera poses and the 3D location of matched features. Then, given an estimate for feature $\mathbf{x}_i$ and camera pose $\mathbf{x}_j$, the constraint $\mathbf{e}_{ij}^{\intercal} \Omega_{ij} \mathbf{e}_{ij}$ is the re-projection error of the 3D point onto the image plane with respect to the measured 2D feature point $\mathbf{z}_{ij}$.

Cost functions in the form of Eq. (1) are effectively represented by a graph: parameter block $\mathbf{x}_i$ maps to vertex i, while constraint $\mathbf{e}_{jk}$ to an edge connecting vertexes j and k. Note that two parameter blocks can be constrained by more than one measurement, so we could have more than one edge linking the same vertex pair. An example

of such representation is given in Fig. 1a. As the standard SLAM process deals with the exploration of a previously unseen environment, little assumptions can be done and the presence of specific objects seen through different frames, *e.g*. the car in Fig. 1a, cannot help the optimization task. On the other hand, given a database of features belonging to known objects, we would want to take advantage from matches across frames, as those in Fig. 1b, to achieve object detection and also improve SLAM. Purposely, the proposed solution, shown in Fig. 1c, explicitly includes into the graph the unknown object pose as a vertex constrained to the matching frames. In the following subsections we will show how this intuition can be used to develop a novel integrated SLAM and object detection pipeline that, unlike previous techniques [3, 7], does not rely on external detection routines, but validates each hypothesis within a unified *semantic* bundle adjustment framework. As our proposal is agnostic with respect to a specific BA-based SLAM approach, in the following we will simply refer to generic frame-to-frame constraints coming from a SLAM engine.

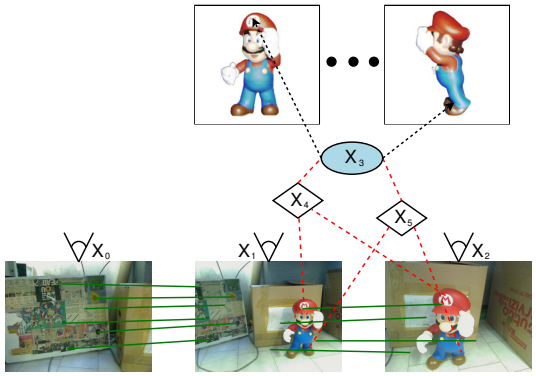
3.1. The Model Database

For each object instance to be detected we need a set of features. If a full 3D model is available, 3D keypoint detectors (*e.g*. [29]) and descriptors (*e.g*. [16, 26]) can be used; otherwise, the model can also be learned from a set of calibrated images, 2D keypoint detectors and descriptors such as [21] providing the required features in this scenario. Feature descriptors are saved for future matching. Then, as for feature positions, in the former embodiment we store 3D coordinates, in the latter 2D image coordinates together with their corresponding view poses.

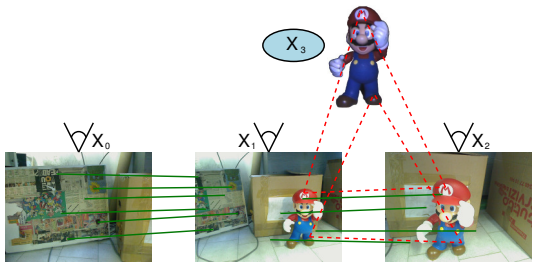
3.2. The Object Detection Pipeline

State-of-the-art feature-based 2D/3D object detection pipelines, such as *e.g*. [21, 1], typically rely on a single view for casting and validating hypotheses. Such an approach, though, is inherently hindered by viewpoint changes, clutter and occlusion. On the other hand, should several snapshots of the scene be available, it would not be straightforward to deploy such far richer information as there is no established machinery to carry out detection from cues gathered from different uncalibrated views. Our novel BA-formulation of the object detection problem effectively overcomes the above limitation.

The Validation Graph As soon as a new frame becomes available, features are extracted, described and matched to the model database (see Sec. 3.1). Here, outliers are allowed, since the proposed technique can discard wrong correspondences. Then, for every set of correspondences related to a given object we build a *validation* graph as shown in Fig. 2. If 2D features have been matched, we create a new



(a) Dashed red: semantic edges representing 2D feature reprojection errors. Landmark vertexes have diamond-like shapes to highlight their different dimensionality. Black dashed arrows represent the known transformations from the object's to its views' reference frames.



(b) Dashed red: frame-to-object edges from 3D feature matching and the corresponding frame-to-frame edges derived from that matching.

Figure 2: A validation graph is built from frame-to-object correspondences as well as frame-to-frame constraints (solid green) provided by the SLAM engine; both 2D (a) and 3D (b) features can be used.

vertex representing the unknown 3D position of the landmark and a set of edges to include its reprojection errors into the cost function. For instance, consider vertex $\mathbf{x}_4$ in Fig. 2a, which is, let's say, the landmark position for the n^{th} object feature. The associated cost terms are:

$$\left\|\mathbf{q}_3^n - V_3\left[\mathbf{x}_4\right]\right\|^2 + s_1^{3,n}\left\|\mathbf{p}_1^{3,n} - V_1\left[\mathbf{x}_4\right]\right\|^2 + s_2^{3,n}\left\|\mathbf{p}_2^{3,n} - V_2\left[\mathbf{x}_4\right]\right\|^2 \quad (2)$$

where $\mathbf{q}_o^n$ denotes the n^{th} 2D feature point learned for the o^{th} object, $\mathbf{p}_i^{o,n}$ is the 2D feature point of the i^{th} frame that matches $\mathbf{q}_o^n$ with probability $s_i^{o,n}$, $V_i\left[\mathbf{r}\right]$ rotates and translates $\mathbf{r} \in \mathbb{R}^3$ so as to apply the current pose estimate associated with the i^{th} vertex, *i.e*. $\mathbf{x}_i^{-1}$, and then projects the point onto the image plane. When V_i concerns an object pose, such image plane is one of the calibrated views of the object acquired at training time, so the reprojection is

chained with the known rigid transformation between the object reference frame and the view reference frame (the black dashed arrows in Fig. 2a). Note that the terms in Eq. (2) can be easily written according to the more general form of Eq. (1), *i.e*.

$$\left\|\mathbf{q}_3^n - V_3\left[\mathbf{x}_4\right]\right\|^2 = \left(\mathbf{q}_3^n - V_3\left[\mathbf{x}_4\right]\right)^\intercal \mathrm{I}_{2\times 2}\left(\mathbf{q}_3^n - V_3\left[\mathbf{x}_4\right]\right) \quad (3)$$

and

$$s_1^{3,n}\left\|\mathbf{p}_1^{3,n} - V_1\left[\mathbf{x}_4\right]\right\|^2 = \left(\mathbf{p}_1^{3,n} - V_1\left[\mathbf{x}_4\right]\right)^\intercal \left(s_1^{3,n}\mathrm{I}_{2\times 2}\right)\left(\mathbf{p}_1^{3,n} - V_1\left[\mathbf{x}_4\right]\right). \quad (4)$$

A similar approach is employed when 3D features are available. In this scenario 3D coordinates are known in every vertex reference frame, so no landmark vertex has to be created but instead we can directly link the camera frames to the object (see Fig. 2b). Moreover, we can constrain together the frames in which matches related to the same object features are found by an extra edge representing a *virtual* match. Accordingly, if m_{ij} is the event *"feature i matches feature j"* and we know that $\Pr(m_{ik}) = s_{ik}$ and $\Pr(m_{jk}) = s_{jk}$, we wish to know $\Pr(m_{ij})$. Assuming that m_{ik} and m_{jk} are independent and

$$\Pr\left(m_{ij}\,|m_{ik}, m_{jk}\right) = \begin{cases} 1, & \text{if } m_{ik} = \text{TRUE} \wedge \\ & \quad m_{jk} = \text{TRUE} \\ 0, & otherwise \end{cases} \quad (5)$$

then $\Pr(m_{ij}) = s_{ik}s_{jk}$. Thus, recalling that now $\mathbf{q}_o^n$ and $\mathbf{p}_i^{o,n}$ represent 3D features, the *semantic* edges for the n^{th} object feature in Fig. 2b can be written as follows:

$$\begin{aligned} & s_1^{3,n}\left\|\mathbf{q}_3^n - \mathbf{x}_3^{-1}\mathbf{x}_1\left[\mathbf{p}_1^{3,n}\right]\right\|^2 \\ & + s_2^{3,n}\left\|\mathbf{q}_3^n - \mathbf{x}_3^{-1}\mathbf{x}_2\left[\mathbf{p}_2^{3,n}\right]\right\|^2 \\ & + s_1^{3,n}s_2^{3,n}\left\|\mathbf{p}_2^{3,n} - \mathbf{x}_2^{-1}\mathbf{x}_1\left[\mathbf{p}_1^{3,n}\right]\right\|^2 \end{aligned} \quad (6)$$

Finally, we add frame-to-frame constraints from the SLAM engine in order to robustify detection and get consistent optimization results. If the current frame is the first matching that object, we also expand the validation graph with frame-to-frame constraints to the previous frame (*c.f*. frame $\mathbf{x}_0$ in Fig. 2).

Hypotheses Validation and Pose Refinement The validation graph is optimized minimizing the cost function in Eq. (1), which includes both frame-to-frame as well as frame-to-object constraints. Then, to retain or discard edges, we rely on the global weighted mean residual from the last global optimization, $\overline{\varrho}$, as defined in Sec. 3.3. This residual can be interpreted as the expected error provided by correct correspondences. Therefore, we compare this value to the residual of each match coming from the last processed frame and, if this is above some threshold, the edge is removed. More precisely, in case of 2D feature matches we remove the edge from the frame vertex to the landmark vertex if

$$\left\|\mathbf{p}_i^{o,n} - V_i\left[\mathbf{x}_{h(o,n)}\right]\right\|^2 \geq \alpha\overline{\varrho} \quad (7)$$

where $h(o,n)$ returns the index of the landmark vertex associated with the n^{th} feature point on the o^{th} object and α is a given parameter. Of course, if the removal of the frame-to-landmark edge leaves the landmark vertex attached only to the object, we delete the object-to-landmark edge too. As for the 3D feature scenario instead, we compare every frame-to-object edge to the threshold, so that if

$$\left\|\mathbf{q}_o^n - \mathbf{x}_o^{-1}\mathbf{x}_i\left[\mathbf{p}_i^{o,n}\right]\right\|^2 \geq \alpha\overline{\varrho} \quad (8)$$

the edge is removed. Under the hypotheses given in the previous paragraph about feature matching, the edges representing virtual matches created toward other frame vertexes are deleted as well.

After this cleaning procedure, the semantic edges, *i.e*. the frame-to-landmark or the frame-to-object edges, connected to the last processed frame are counted. If above a minimum number, the remaining constraints are validated, otherwise we treat them as noise and remove them from the validation graph. Such a threshold is not critical and we set it to 3 in our experiments.

If the validation process is successful, an other optimization is run on the remaining edges to refine the estimate. Then, a final cleaning is performed on the whole validation graph with a threshold $\overline{\varrho}$ computed just as before. This procedure can leave the graph in three different states, defined from the comparison between the final number of semantic edges N_{se} and two thresholds η_f and η_t, with $\eta_f < \eta_t$:

$N_{\text{se}} < \eta_f$ the object is classified as a false detection, the validation graph is destroyed and the object is removed from the global graph, if present (*c.f*. Sec. 3.3);

$\eta_f \leq N_{\text{se}} < \eta_t$ the detection is ambiguous, the validation graph is saved waiting for more visual cues, but the object is removed from the global graph, if present (*c.f*. Sec. 3.3);

$N_{\text{se}} \geq \eta_t$ the object is detected and added to, if not present, or updated in, if already present, the global graph (*c.f*. Sec. 3.3).

Again, the two thresholds are not critical, as at every frame the validation of new hypotheses benefits from previously refined feature matches and, moreover, the final cleaning retains only the best edges among all, *i.e*. those most coherent with the current solution. In our experiments we set $\eta_f \triangleq 3$

and $\eta_t \triangleq 10$. A higher η_f could skip heavily occluded objects detectable only with a few matches from several views, while the difference $\delta = \eta_t - \eta_f$ is related to the robustness of the validation routine: the higher δ is, the more correspondences, possibly from different frames, are needed for a detection, but also the less false positive are propagated to the global graph and possibly deleted in the following frames.

3.3. Semantic SLAM

The validation graphs built in the previous section for every matched object covers only a subset of the whole map, *i.e*. those frames matching with that object. Hence, we carry out a global semantic optimization in order to jointly optimize all camera poses as well as all object poses. This global graph comprises:

- all the camera pose vertexes with frame-to-frame constraints coming from the SLAM engine;
- all the pose vertexes of those objects for which the validation procedure turned out successful (*c.f*. Sec. 3.2);
- all frame-to-landmark and object-to-landmark constraints, in case of 2D feature matching, or frame-to-object and virtual frame-to-frame constraints, in case of 3D feature matching, coming from detected objects' validation graphs.

Optimizing such a graph spreads the error over all the estimates and gives a consistent global solution. Finally, we compute the weighted mean residual as

$$\overline{\varrho} = \frac{\sum w_{ij} \mathbf{e}_{ij}^{\intercal} \mathbf{e}_{ij}}{\sum w_{ij}} \tag{9}$$

where we have assumed, for simplicity, that an expression for the weight w_{ij} can be derived for all the edges, *e.g*. the parameter $s_i^{o,n}$ for our semantic edges. This residual represents the mean expected error for an edge consistent with the current solution. As such, it can be used as validation threshold during the next object detection routine, as already discussed in Sec. 3.2.

4. Evaluation

We implemented the proposed framework for evaluation purposes using the 3D formulation within the G2O graph optimizer [18]. For both models and frames we extract 3D keypoints by means of the Intrinsic Shape Signature detector [29] at three different scales and then describe such keypoints by Spin Images [16]. For each scale, an index is created including all models' descriptors at that scale and for each descriptor in an incoming frame, a k-nearest neighbor search at its scale based on the Euclidean distance is performed. Matches are validated according to the distance-ratio criterion suggested in [21]: denoting as d_1 the nearest distance and d_2 the nearest belonging to a different model, the match is accepted if $d_1/d_2 < \beta$, with $\beta = 0.9$.

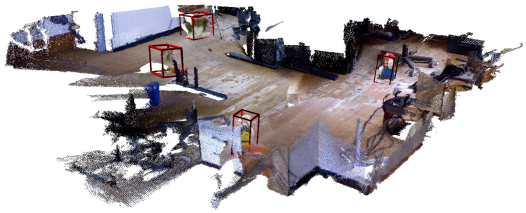


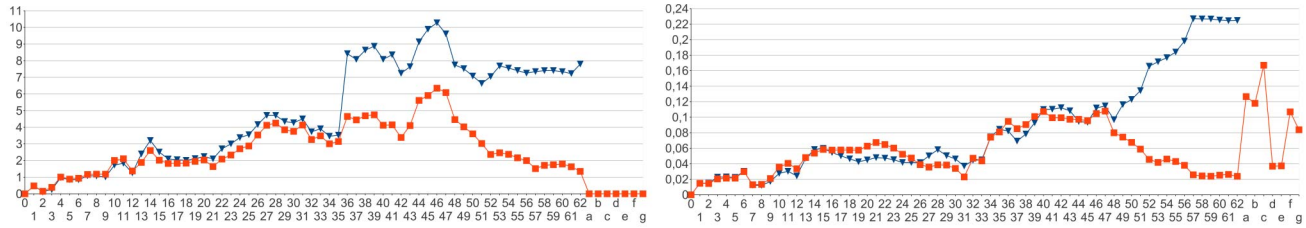
Figure 3: *4-objects* sequence: final reconstruction with aligned bounding boxes around detected objects.

The adopted strategy returns a large number of matches, so a reduced set is picked out by a RANSAC-based 6DOF object pose estimation [2]. We set the inlier threshold to 0.05m, which in our experiments is roughly 20 times the mesh resolution. The final set still includes outliers, but our semantic bundle adjustment algorithm can cope with them effectively. With reference to Eq. (7) (8), we set $\alpha = 7$, though we found equally good or sometimes even slightly better results with values in range $[5, 8]$ too.

4.1. Quantitative Results

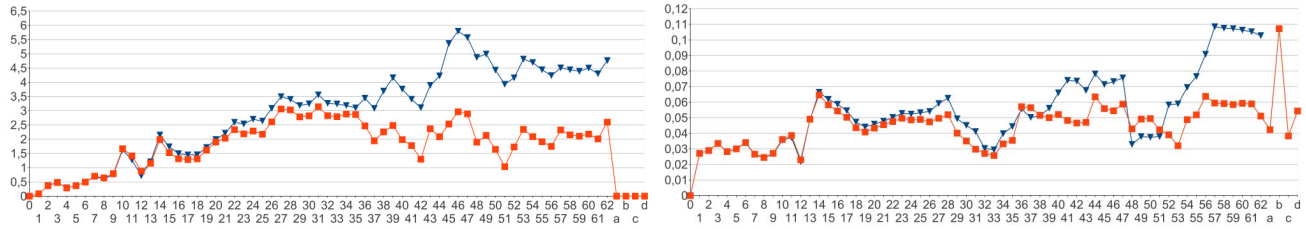
Our proposal is about the estimation of both camera and objects 6DOF poses, but to the best of our knowledge no publicly available dataset provides all such ground-truth information. Therefore, to achieve quantitative results, we decided to render object meshes into frames belonging to the publicly available RGB-D SLAM dataset [25], which is a collection of sequences acquired by a Kinect and available together with ground-truth camera poses. Moreover, before rendering objects, we performed a 2Hz time subsampling and an ICP-like [5] pose optimization in order to improve the accuracy of ground-truth camera poses. This non-trivial refinement was successfully performed on the "freiburg1_floor" sequence. Then, from the refined sequence, we created the *7-objects* and *4-objects* sequences by rendering into RGB-D frames 4 and 7 objects from our model database respectively. A database composed of all the 7 full 3D models has been used for both sequences, the presence of non-detectable object models in the *4-objects* sequence acting as a nuisance within the feature matching process.

Frame-to-frame correspondences are obtained by matching SIFT features [21], projecting them in 3D based on the depth map and running a RANSAC-based camera pose estimation [2], which is the basic approach taken by recent SLAM engines for RGB-D sensors such as [14, 13]. The sensor carries out a closed loop in a medium-size office, but



(a) (b)

Figure 5: 7-objects sequence: Final error in rotation (a), in degrees, and translation (b), in meters, for every processed frame (numbered) and detected objects (a: *Doll*, b: *Duck*, c: *Frog*, d: *Mario*, e: *Rabbit*, f: *Squirrel*, g: *Tortoise*). Blue triangles: plain SLAM; red squares: our technique.



(a) (b)

Figure 6: *4-objects* sequence: Final error in rotation (a), in degrees, and translation (b), in meters, for every processed frame (numbered) and the detected objects (a: *Doll*, b: *Duck*, c: *Frog*, d: *Mario*). Blue triangles: plain SLAM; red squares: our technique.

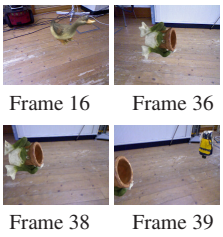
Figure 4: *7-objects* sequence: without explicit loop closure detection a plain SLAM engine (left) accumulates error over time, whilst the ability to detect *Mario* allows our technique to also constrain the last and first frames and thus to tear down the global error (right) .

no explicit loop closure detection scheme is implemented. The results in Fig. 5 and 6 show, together with object detection (also in Fig. 3), improved accuracy compared to the adopted plain SLAM engine due to the links between frames established through our semantic edges, such as in particular those due to detection of objects at the beginning and at the end of the sequences (see Fig. 4).

To illustrate the effectiveness of the proposed match validation technique, Tab. 1 reports the frame-to-object edges in the validation graph of the *Frog* model for some frames of the *7-objects* sequence. *Frog* is firstly matched in frame 16, with 11 feature correspondences surviving the RANSAC step (row "Frame 16" in Tab. 1). These matches are clearly false positives (*Frog* is the object displayed in frame 36) and our cleaning algorithm is able to recognize this inconsistency leaving only 3 edges into the graph. Thus, the object pose is not inserted into the global graph to be optimized, though the wrong edges are saved. Later, *Frog* is correctly matched, the correct correspondences pass all the validation tests and the object is added to the global graph, but the previous wrong edges raise the weighted mean residual and cause a large reconstruction error (rows from "Frame 34" to "Frame 37"). However, after a few frames, as more good correspondences are gathered, the 3 wrong edges are detected and erased, thus tearing down the error on the object's pose estimate (rows "Frame 38" and "Frame 39").

4.2. Qualitative Results

In this experiment we make a step further and, though qualitatively, validate our proposal on a truly real Kinect sequence taken in our Lab. Accordingly, the feature matching task gets more difficult, and we found it more appropriate to rely here on a more advanced descriptor deploying both shape and color information such as [26]. As depicted in Fig. 7, we performed a complete loop capturing the object *Doll* at the beginning and at the end of the sequence. Then, we ran both the plain SLAM engine and our novel semantic pipeline: while a simple tracking scheme eventually drifts (see Fig. 7a), our approach correctly validates object presence and implicitly closes the loop by deploying object de-



	$\mathbf{x}_{16}$	$\mathbf{x}_{34}$	$\mathbf{x}_{35}$	$\mathbf{x}_{36}$	$\mathbf{x}_{37}$	$\mathbf{x}_{38}$	$\mathbf{x}_{39}$	Pose Error (R / t)
Frame 16	3 (11)	—	—	—	—	—	—	n/a
Frame 34	3	31 (38)	—	—	—	—	—	114.6° / 1.053m
Frame 35	3	31	55 (55)	—	—	—	—	88.5° / 1.190m
Frame 36	3	31	55	122 (122)	—	—	—	86.3° / 1.237m
Frame 37	3	31	55	122	127 (127)	—	—	86.8° / 1.273m
Frame 38	0	31	55	122	127	123 (123)	—	0° / 0.180m
Frame 39	0	31	55	122	127	123	0 (47)	0° / 0.200m

Table 1: *7-objects* sequence: an excerpt from the validation graph of model *Frog*. Rows reports the number of frame-to-object edges for the vertexes in the graph at the end of the validation procedure for the frame in first column. Also, the number of matches before edge cleaning is shown in brackets. Last column reports the pose error for *Frog* in the global graph.

(a) (b)

Figure 7: A plain SLAM engine (a) accumulates errors and eventually cannot close the loop; integrated object detection and semantic optimization (b) allows for implicit loop closure and improved reconstruction.

Figure 8: AR with 3D occlusion handling: once the *Doll* is detected (top right) a red umbrella can be rendered, even when the object is later occluded (bottom right); the final augmented 3D reconstruction is shown on the left.

tection information (see Fig. 7b).

Finally, we demonstrate the ability to carry out Augmented Reality with full 3D occlusion handling. Indeed, unlike previous work which enables to augment only the whole scene reconstruction due to the lack of semantic information related to individual objects [17, 23], our framework brings in seamlessly 3D mapping, camera tracking and object detection/localization, *i.e.* all the information needed to pick up visual content specifically related to each object and render it coherently with respect to the 3D structure of the scene, in particular so as to handle occlusions effectively.

5. Concluding Remarks

We have proposed a novel Semantical Bundle Adjustment framework which allows for solving jointly the object detection and SLAM problems. From a probabilistic point of view, the underline density is multi-modal due to the mixing of probabilities related to object existence and object/camera poses. To address the problem through bundle adjustment, which is inherently unimodal, we follow two steps: first the validation graphs establishes upon objects' existence, then the global semantic graph jointly solves for camera and detected object poses.

The current implementation of the framework is mainly aimed at demonstrating the underlying theory and relies on its 3D formulation. As such, it is not conceived as a real-time application, the bottleneck being detection, description and matching of 3D features, which requires several seconds, although the semantic optimization can run in the order of tens of milliseconds. We plan to come-up soon with a real-time application grounded on the 2D formulation of the theory, leveraging on a fast SIFT GPU implementation and still providing a dense 3D reconstruction through RGB-D sensing. From a theoretical perspective, we plan to extend our work towards two main directions: detection of multiple object instances, which is not supported in the current formulation but quite easily addressable in principle, and generalizing the framework to deal also with category-level recognition.

References

[1] A. Aldoma, F. Tombari, L. Di Stefano, and M. Vincze. A global hypothesis verification method for 3d object recognition. In *Computer Vision (ECCV), IEEE European Conf. on*, Florence, Italy, Oct 2012.

[2] K. S. Arun, T. S. Huang, and S. D. Blostein. Least-squares fitting of two 3-d point sets. *Pattern Analysis and Machine Intelligence (PAMI), IEEE Trans. on*, 9(5):698–700, Sep 1987.

[3] S. Y. Bao, M. Bagra, Y.-W. Chao, and S. Savarese. Semantic structure from motion with points, regions, and objects. In *Computer Vision and Pattern Recognition (CVPR), IEEE Int'l Conf. on*, 2012.

[4] S. Y. Bao and S. Savarese. Semantic structure from motion. In *Computer Vision and Pattern Recognition (CVPR), IEEE Int'l Conf. on*, 2011.

[5] P. J. Besl and H. D. McKay. A method for registration of 3-d shapes. *Pattern Analysis and Machine Intelligence (PAMI), IEEE Trans. on*, 14(2):239–256, 1992.

[6] R. O. Castle, G. Klein, and D. W. Murray. Combining monoslam with object recognition for scene augmentation using a wearable camera. *Image and Vision Computing*, 28(11):1548–1556, 2010.

[7] J. Civera, D. Gálvez-López, L. Riazuelo, J. D. Tardós, and J. M. M. Montiel. Towards semantic SLAM using a monocular camera. In *Intelligent Robot Systems (IROS), IEEE/RSJ Int'l Conf. on*, pages 1277–1284, 2011.

[8] J. Civera, O. G. Grasa, A. J. Davison, and J. M. M. Montiel. 1-point ransac for extended kalman filtering: Application to real-time structure from motion and visual odometry. *Journal of Field Robotics*, 27(5):609–631, Sep 2010.

[9] N. Cornelis, B. Leibe, K. Cornelis, and L. Van Gool. 3d urban scene modeling integrating recognition and reconstruction. *International Journal of Computer Vision (IJCV)*, 78(2-3):121–141, July 2008.

[10] A. J. Davison. Real-time simultaneous localisation and mapping with a single camera. In *Computer Vision (ICCV), IEEE Int'l Conf. on*, page 1403, Washington, DC, USA, 2003.

[11] E. Eade and T. Drummond. Monocular SLAM as a graph of coalesced observations. In *Computer Vision (ICCV), IEEE Int'l Conf. on*, pages 1–8, Rio de Janeiro, Brasil, Oct 2007.

[12] S. Ekvall, P. Jensfelt, and D. Kragic. Integrating active mobile robot object recognition and slam in natural environments. In *Intelligent Robots and Systems, IEEE/RSJ Int'l Conf. on*, Oct 2006.

[13] F. Endres, J. Hess, N. Engelhard, J. Sturm, D. Cremers, and W. Burgard. An evaluation of the RGB-D SLAM system. In *Robotics and Automation (ICRA), IEEE Int'l Conf. on*, St. Paul, MA, USA, May 2012.

[14] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox. Rgb-d mapping: Using depth cameras for dense 3d modeling of indoor environments. In *Experimental Robotics (ISER), Int'l Symp on*, 2010.

[15] D. Hoiem, A. A. Efros, and M. Hebert. Putting objects in perspective. *International Journal of Computer Vision (IJCV)*, 80(1):3–15, Oct 2008.

[16] A. Johnson. *Spin-Images: A Representation for 3-D Surface Matching*. PhD thesis, Robotics Institute, Carnegie Mellon University, Aug 1997.

[17] G. Klein and D. Murray. Parallel tracking and mapping for small ar workspaces. In *Mixed and Augmented Reality (ISMAR), IEEE and ACM Int'l Symp. on*, pages 225 –234, Nov 2007.

[18] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard. g2o: A general framework for graph optimization. In *Robotics and Automation (ICRA), IEEE Int'l Conf. on*, Shanghai, China, May 2011.

[19] L. Ladický, P. Sturgess, C. Russell, S. Sengupta, Y. Bastanlar, W. Clocksin, and P. H. Torr. Joint optimization for object class segmentation and dense stereo reconstruction. *International Journal of Computer Vision (IJCV)*, 100:122–133, 2012.

[20] L.-J. Li, R. Socher, and L. Fei-Fei. Towards total scene understanding: Classification, annotation and segmentation in an automatic framework. In *Computer Vision and Pattern Recognition (CVPR), IEEE Int'l Conf. on*, 2009.

[21] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision (IJCV)*, 60(2):91–119, Jan, 5 2004.

[22] 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*, 56(6):503–511, June 2008.

[23] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *Mixed and Augmented Reality (ISMAR), IEEE and ACM Int'l Symp. on*, pages 127–136, Washington, DC, USA, 2011.

[24] H. Strasdat, A. J. Davison, J. Montiel, and K. Konolige. Double window optimisation for constant time visual SLAM. In *Computer Vision (ICCV), IEEE Int'l Conf. on*, pages 2352–2359, Los Alamitos, CA, USA, 2011.

[25] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of rgb-d slam systems. In *Intelligent Robot Systems (IROS), IEEE/RSJ Int'l Conf. on*, Oct 2012.

[26] F. Tombari, S. Salti, and L. Di Stefano. A combined texture-shape descriptor for enhanced 3D feature matching. In *Image Processing (ICIP), IEEE Int'l Conf. on*, pages 809–812, Sep 2011.

[27] B. Triggs, P. Mclauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment – a modern synthesis. In *Vision Algorithms: Theory and Practice*, LNCS, pages 298–375. Springer Verlag, 2000.

[28] S. Vasudevan, S. Gächter, V. Nguyen, and R. Siegwart. Cognitive maps for mobile robots-an object based approach. *Robotics and Autonomous Systems*, 55(5):359–371, May 2007.

[29] Y. Zhong. Intrinsic shape signatures: A shape descriptor for 3d object recognition. In *ICCV Workshop*, pages 689–696, Oct 2009.