2011 IEEE/RSJ International Conference on
Intelligent Robots and Systems
September 25-30, 2011. San Francisco, CA, USA

Towards Semantic SLAM using a Monocular Camera

Javier Civera, Dorian Gálvez-López, L. Riazuelo, Juan D. Tardós and J. M. M. Montiel

Abstract— **Monocular SLAM systems have been mainly focused on producing geometric maps just composed of points or edges; but without any associated meaning or semantic content. In this paper, we propose a semantic SLAM algorithm that merges in the estimated map traditional meaningless points with known objects. The non-annotated map is built using only the information extracted from a monocular image sequence. The known object models are automatically computed from a sparse set of images gathered by cameras that may be different from the SLAM camera. The models include both visual appearance and tridimensional information. The semantic or annotated part of the map –the objects– are estimated using the information in the image sequence and the precomputed object models.**

The proposed algorithm runs an EKF monocular SLAM parallel to an object recognition thread. This latest one informs of the presence of an object in the sequence by searching for SURF correspondences and checking afterwards their geometric compatibility. When an object is recognized it is inserted in the SLAM map, being its position measured and hence refined by the SLAM algorithm in subsequent frames.

Experimental results show real-time performance for a hand-held camera imaging a desktop environment and for a camera mounted in a robot moving in a room-sized scenario.

I. INTRODUCTION

SLAM, standing for Simultaneous Localization and Mapping, aims to self-localize a robot and estimate a model of its environment from sensory information [8]. The richer the environment model and the higher its semantic level, the more useful it becomes for a robot in order to perform autonomous tasks. Currently, the map produced by most of the SLAM algorithms is a joint estimation of geometric entities –in most cases points or lines– without any semantic meaning or annotations attached to it. The aim of this paper is to enrich usual monocular SLAM maps by partially annotating the geometric estimation with object information.

The addition of semantic content to a geometric SLAM is done in this paper by recognizing object instances and registering them into the estimated map. We have chosen monocular vision as the only input for constructing the object models, recognizing them and estimating the geometric map. We believe that there are three reasons why such a minimalistic configuration has a high potential for robotic applications: 1) there is a large body of recognition models from the computer vision community; 2) the Internet is widely populated with pictures from which we could automatically generate the recognition models in the future; and 3) different cameras can be used at different steps. Cameras for building the models could be different for cameras used in SLAM –so they are in our experiments. That allows interoperability, that is, the capability to exploit the same models by diverse robots.

The authors are with the Robotics, Perception and Real-Time Group, Aragón Institute of Engineering Research (I3A), Universidad de Zaragoza, 50018 Zaragoza, Spain. jcivera,dorian,riazuelo,tardos,josemari@unizar.es

Object recognition from visual data is experiencing an extraordinary progress, but most of the research is directed to recognition from just a single image. This is in fact the right approach for some recognition applications like content-based Internet images classification. Nevertheless, in our opinion, the robotic scenario poses some specific constraints that do not fit the main research line. Specifically, there are three main issues that lead us to the combination of local mapping plus object recognition and registration as a feasible solution for the robotic mapping problem.

First, a robot has to interact physically with its environment and then the 3D registration of the object is a must in order to perform any task. The sole presence of the object in an image is not enough in most of the robotic applications (e.g., grasping an object).

Secondly, the usual visual input of most robots is not a single image but a sequence. The recognition algorithms should be adapted to exploit this source of information.

Finally, every robotic task poses severe real-time constraints that should be taken into account. Using again the grasping example, we are interested in the location of the robotic arm with respect to the object *at the specific grasp moment*. In our approach, the objects are registered within the SLAM map when they are recognized and their 3D location is then available at any moment after that. As a desirable feature of our algorithm, once an object is recognized and inserted into the map its position is known for task definition at every moment –even when the object comes out from the camera field of view.

Summing up, our work is motivated by the limitations of both object recognition and standard geometric SLAM for robotic tasks. On the one hand, object recognition algorithms on their own are unable to register several objects in a common 3D reference frame. On the other hand, any standard geometric SLAM algorithm can provide an accurate geometric registration, but cannot be used for a high-level task definition that includes objects (for example, 'grasp a cup'). The semantic SLAM proposed in this paper –object recognition, object registration and local SLAM– forms a basic sensing capability suitable for task execution by diverse robots.

The algorithm proposed in this paper uses the state-of-the-art in three different areas to achieve the realization of this idea: First, an EKF monocular SLAM provides the online real-time estimation for the current camera location and a

978-1-61284-456-5/11/$26.00 ©2011 IEEE

sparse map of point features [6]. Second, Structure from Motion is used to precompute a database of object models from sparse images [24]. Third, visual recognition allows to detect the presence of the object in an image stream [15], [16].

The rest of the paper is organised as follows: Section II discusses related work and section III summarizes the whole algorithm. Next sections are devoted to further details about the different components of the algorithm: section IV to the object model, section V to the object recognition and VI to the backbone point-based monocular SLAM where objects are registered into. Finally, section VII shows experimental results and section VIII concludes and presents lines for future work.

II. RELATED WORK

Visual recognition has been used in robotic SLAM mostly for *scene recognition*, for example loop closing in [7], [2], relocalization in [28] and place classification in a semantic context in [22]. Nevertheless, it has been scarcely combined with SLAM for *object recognition*. [12] is one of the few examples, using a map hierarchy together with object recognition in order to obtain additional information to classify the places where the objects are found in. A similar idea is presented in [26], where the objects are located in the 3D space by means of a stereo camera. Image-based recognition was also used in [9], [17], [29] to register known objects into a laser SLAM map. Differently from these latest works, we do not only use the object appearance but also its estimated geometry in the object model. As a second difference our geometric SLAM algorithm tracks the object features after the insertion; hence being able to refine their 3D pose. Regarding sensors, we rely only on a monocular camera for every step in our algorithm.

The closest approach to this paper is the work by Castle et al. [3], [4], which registers planar objects into an EKF-based monocular SLAM. Their research make use of SIFT features to construct the appearance model of the objects and insert in the SLAM map three of the boundary corners of the plane. Compared to this approach, we are able to overcome the planar restriction and can deal with any object geometry. Our object model is composed by the appearance; but also by the geometric position of salient point features over the object. When the object is recognized, we insert in the SLAM map the recognized points using general –non-planar– geometric constraints.

As the word *semantic* may have a broad meaning, it is worth remarking that in this paper it is referred to the labeling of certain map features in a SLAM map. Although we find it a promising line of research, we do not consider here the relations between objects in a scene (as done, for example, in [23]).

III. NOTATION AND GENERAL OVERVIEW

Given that geometric SLAM is typically faster than object recognition, our algorithm is divided into two threads: one dedicated to monocular SLAM and other one to object recognition. Figure 1 gives an general overview of the algorithm using figures from our experimental results. This section summarizes briefly our proposal based on this figure, and every part is detailed in next sections.

Let start at step $k-m$. The image I_{k-m} is used both in the monocular SLAM and the recogntion threads. The monocular SLAM thread uses this image to update the geometric state $\mathbf{x}$ from the previous step $k-m-1$ to the current one $k-m$ using a standard EKF formulation. At the same time, the SLAM state vector is augmented with the current camera pose. Such augmentation will be necessary every time the recognition thread starts for a coherent object insertion. The recognition results will arrive after some processing at step k; but the object insertion should be made with respect to the input image I_{k-m}. The details about the state augmentation are described in section VI-B.

We have a model for every object we want to recognize $O_1, \ldots, O_p, \ldots, O_q$ stored in a database. Each one of these models contains appearance and geometric information extracted from a set of images of the object. The appearance information is composed of SURF descriptors extracted from the images; and the geometric information is the 3D position of these SURF points.

The object recognition thread inserts the objects from the database in the SLAM map as follows: First, the algorithm searches for correspondences between the image I_{k-m} and every object $O_1, \ldots, O_p, \ldots, O_q$. SURF point features are extracted from image I_{k-m} and compared against the precomputed SURF descriptors for each object. RANSAC is used to compute a consistent geometric model of each object that maximizes the number of correspondences. If there are enough consistent correspondences, the points are assumed to belong to the object and inserted into the SLAM map. Once in the SLAM map, these points are tracked and their position refined in the rest of the sequence.

We are assuming in this paper that there are not repeated objects nor false object detections (in fact, false object detections did not appear in none of our experiments). Nevertheless, the opinion of the authors is that the proposed combination could easily handle both situations. As objects in our system are geometrically registered, any repeated object detection in a different location than the previous one is a different object of the same class. The geometric object registration also serves for the case of false detections: as the SLAM algorithm keeps measuring the object features once inserted, any geometric inconsistency produced by a false detection would be rejected as spurious by the 1-point RANSAC [6].

IV. OBJECT MODEL

Our model for an object O is based on the information extracted from a set of images I_l of it. In order to construct the object model, SURF features [1] are extracted in first place from the set of images I_l. The SURF descriptors $\mathbf{d}_O^l$ form the appearance model for the object. The geometric position for the SURF points $\mathbf{y}_O$ is computed using pairwise

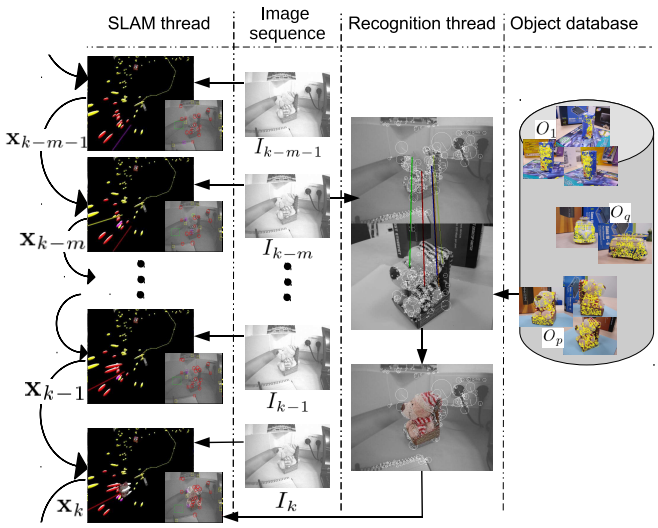

Figure 1. Overview of the algorithm using figures from our experiments. Best seen in color.

Figure 2. The top row and bottom left images show 3 out of the 20 images that were used for the model of the teddy bear in our first experiment. Notice the SURF features used for recognition superimposed over the images. The bottom right image shows the dense 3D model that will be registered in the SLAM map.

Structure from Motion algorithms [14]: correspondences are searched over the images I_l using the SURF descriptors, the geometry is estimated between pairs of images using robust algorithms, and a non-linear optimization step finally produces the 3D estimation for SURF points $\mathbf{y}_O$ and the relative motion between the images I_l. A modern software library called *Bundler* [24] has been used in this paper for constructing this geometric part of the model.

As standard cameras usually do not capture a whole object in an image –you never see the front and the back of an object at the same time–, the object model is divided into *faces*. We name *face* F to the tuple $< \mathbf{t}_F^O,\ \mathbf{q}_F^O,\ \mathbf{y}_F^O,\ \mathbf{d}_F >$. That means each *face* corresponds to an image augmented with the appearance –the SURF descriptors $\mathbf{d}_F$– and the geometric information –the 3D position of the SURF points $\mathbf{y}_F^O$– needed for the object insertion. The *face* also includes the position $\mathbf{t}_F^O$ and orientation $\mathbf{q}_F^O$ of the image with respect to an object reference frame O. The object reference frame has been chosen to be the centroid of the 3D point cloud with two axis aligned with its principal components.

Apart from the model for each face F, a global dense point cloud 3D model is also constructed starting from the *faces* and the estimated location of the object images. Multi-View Stereo algorithms (specifically the software package *PMVS* [11]) are used for this purpose. It should be remarked that this dense model is not used for object 3D registration nor recognition, which are done by the appearance and 3D position of the SURF features. We think that a dense model like this one could be of importance for robotic applications like grasping; but in this paper it is only used for visualization purposes.

Figure 2 shows three of the images that compose the object model for a teddy bear, used in our first experiment. These three images are a subset of the twenty that were used to construct the model, covering several points of view. Each one of these twenty images is the basis for a *face*. The SURF features that will be used for recognition are drawn in the images as white circles. Finally, the dense 3D model that will be inserted in the SLAM map is also displayed.

V. OBJECT RECOGNITION

The object recognition algorithm starts by extracting SURF features from the image I_{k-m}. For each object O in the database, correspondences are calculated between I_{k-m} and its *faces* F by applying the distance ratio (NNDR) to fast approximate nearest neighbors [19]. These correspondences are then checked to be geometrically consistent. The RANSAC algorithm is run to find a subset of at least 5 correspondences, between different SURF features, that describe a valid transformation between the SLAM image I_{k-m} and the object image I_F. The Perspective-*n*-Point (PnP) problem [18] is used to estimate the translation $\mathbf{t}_F^{C_{k-m}}$ and orientation $\mathbf{q}_F^{C_{k-m}}$ that define this transformation from the image features and 3D points. However, for planar objects, an homography is estimated instead, using the DLT algorithm [14].

The correspondences which are not consistent with the transformation or the homography are rejected. If we obtain a valid transformation between I_{k-m} and a *face* F, we stop searching the rest of the faces of the object O as it has already been found.

When a face $F \equiv< \mathbf{t}_F^O,\ \mathbf{q}_F^O,\ \mathbf{y}_F^O,\ \mathbf{d}_F >$ of an object O is detected, the final pose of the object is obtained by refining the estimated transformation with only those correspondences which were inliers. The 3D points $\mathbf{y}_F^O$ corresponding to the matches $\mathbf{d}_F$ are then fed into the monocular SLAM system for insertion as detailed in section VI-C.

VI. MONOCULAR SLAM

Our proposal builds on a state-of-the-art EKF-based monocular SLAM. In this section the additions to the standard mode are described.

A. Standard Mode EKF

We follow the 1-point RANSAC EKF proposed in [6]. The estimated parameters are modeled as a multidimensional Gaussian variable $\mathbf{x}$, including camera motion parameters $\mathbf{x}_{C_k}$ at step k and the n point features $\mathbf{y}_i$ that form the map. The geometry of the detected objects will be added to this standard system.

$$\mathbf{x}_k = \left(\begin{array}{cccccc} \mathbf{x}_{C_k}^\top & \mathbf{y}_1^\top & \dots & \mathbf{y}_i^\top & \dots & \mathbf{y}_n^\top \end{array} \right)^\top . \quad (1)$$

Camera parameters $\mathbf{x}_{C_k}$ at step k include camera position $\mathbf{t}_{C_k}$ and orientation $\mathbf{q}_{C_k}$ and also linear and angular velocities $\mathbf{v}_{C_k}$ and ω_{C_k}. Point features $\mathbf{y}_i$ are initialized using inverse depth parametrization and are converted to Euclidean coordinates $\mathbf{y}_i^\top = (X \;\; Y \;\; Z)$ when projection equation shows a high degree of linearity [5]. All the geometric parameters are referred to a common frame W, that has been ommited in the formulation for simplicity.

Robust data association relies on 1-point RANSAC EKF, that has proven to be an efficient implementation of a spurious rejection into an EKF framework and shows a low computational cost even for large number of matches and large rates of spurious data [6].

B. State Augmentation with Past Camera Pose

When the recognition thread starts at step $k-m$, the monocular SLAM state has to be augmented with the current camera location. This is done by copying camera position $\mathbf{t}_{C_{k-m}}$ and orientation $\mathbf{q}_{C_{k-m}}$ into the state vector and propagating covariances accordingly. At step $k-1$, just before the object insertion, the state vector will be then as follows:

$$\mathbf{x}_k = \left(\begin{array}{ccccccc} \mathbf{x}_{C_{k-1}}^\top & \mathbf{y}_1^\top & \dots & \mathbf{y}_n^\top & \mathbf{t}_{C_{k-m}}{}^\top & \mathbf{q}_{C_{k-m}}{}^\top \end{array} \right)^\top \quad (2)$$

Such augmentation lasts until the recognition thread finishes, which usually takes several EKF steps. If an object from the database has been recognized the past camera location will be used for the delayed initialization of the recognized object. The augmented camera is not needed anymore after the recognition has finished so the past camera position $\mathbf{t}_{C_{k-m}}$ and orientation $\mathbf{q}_{C_{k-m}}$ are marginalized out from the state.

C. Object Insertion

The output of the recognition thread is a set of points $\mathbf{y}_F^O$ in an object reference frame O that are known to appear in the monocular SLAM sequence. Also, the relative motion $\mathbf{t}_F^{C_{k-m}}, \mathbf{q}_F^{C_{k-m}}$ between the SLAM camera C_{k-m} and the *face* F from the object model has been estimated. Each point $\mathbf{y}_{F,j}^O$ has to be referred to the SLAM reference frame W for its insertion; which is done as follows:

$$\mathbf{y}_{F,j}^W = \mathbf{H}_{C_{k-m}}^W \left(\mathbf{t}_{C_{k-m}}, \mathbf{q}_{C_{k-m}} \right) \mathbf{H}_O^{C_{k-m}} \mathbf{y}_{F,j}^O , \quad (3)$$

where $\mathbf{H}_a^b$ stands for the homogeneous transformation matrix encoding the motion from reference frame b to a. $\mathbf{t}_{C_{k-m}}$ and $\mathbf{q}_{C_{k-m}}$ are the position and orientation of the SLAM camera when the object recognition was initiated and were stored in the state vector $\mathbf{x}$ as described in section VI-B. $\mathbf{H}_O^{C_{k-m}}$ represents the motion between the SLAM camera C_{k-m} and the object reference frame O. It can be computed by composition:

$$\mathbf{H}_O^{C_{k-m}} = \mathbf{H}_F^{C_{k-m}} \left(\mathbf{t}_F^{C_{k-m}}, \mathbf{q}_F^{C_{k-m}} \right) \mathbf{H}_O^F \left(\mathbf{t}_F^O, \mathbf{q}_F^O \right) , \quad (4)$$

where $\mathbf{t}_F^O, \mathbf{q}_F^O$ are the position and orientation of the *face* F in the object frame O and are available from the object model (section IV). $\mathbf{t}_F^{C_{k-m}}, \mathbf{q}_F^{C_{k-m}}$ encode the motion between the *face* F and the SLAM camera C_{k-m}; and are estimated by the recognition thread (section V). Point covariances are propagated accordingly, and points $\mathbf{y}_F^W$ in the W reference are finally inserted into the SLAM map.

$$\mathbf{x}_k = \left(\begin{array}{cccccc} \mathbf{x}_{C_k}^\top & \mathbf{y}_1^\top & \dots & \mathbf{y}_n^\top & \mathbf{y}_F^{W\top} \end{array} \right)^\top . \quad (5)$$

As points in the object model already have a tight 3D estimation, they are inserted in the SLAM state using their Euclidean coordinates. After its insertion, the object points are tracked and hence its position refined by the estandar EKF monocular SLAM formulation [5].

D. Relocalization

In the practical use of any monocular SLAM system, tracked features may be lost due to varied reasons: a dynamic object covering most of the image, lost images or sudden motions producing blur. A relocation system able to recover the camera location with respect to a previous map is essential for any practical system. In this paper we are using the relocation system in [28]. In the video accompanying the paper it can be observed how the EKF monocular SLAM loses its tracked features in our first experiment (section VII-A) and succesfully recovers after a few frames.

VII. EXPERIMENTAL RESULTS

The two experiments presented here were recorded with the same camera, a low-cost black-and-white Unibrain camera with a resolution of 320×240. The model of the objects were built from images taken with a standard consumer digital camera. The image sequences for both experiments were gathered by the Unibrain camera at 30 frames per second and used at this frequency as the input to our experiments. As the computational cost of the proposed algorithm is higher than 33 milliseconds for the map sizes used, some of the frames may be skipped by the semantic SLAM.

A. Desktop Environment

The sequence for this experiment has 8951 frames and was recorded moving a hand held camera over a desktop in one of our laboratories for about 5 minutes. The lighting conditions are particularly bad for this sequence, as can be observed in figure 4. The desktop contains four 3D objects –a tetra pack of fruit juice, a replica of a Volkswagen van,

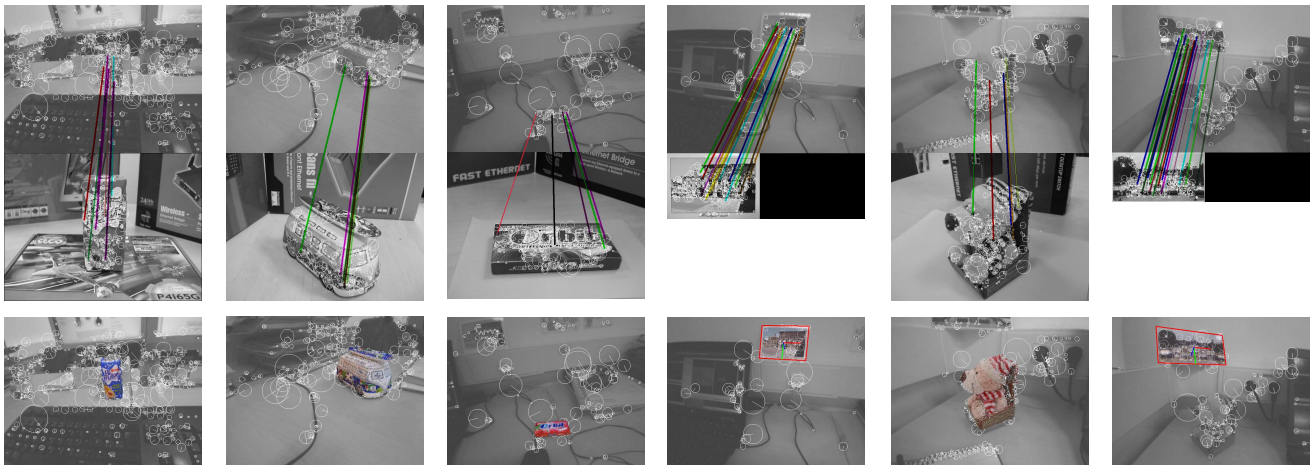

Figure 3. Object recognition thread results, showing columnwise the specific frames where the six objects were detected. The top row shows the image in the monocular sequence input to the semantic SLAM, the middle row shows the *face* of the object model, and the coloured lines are the matches. The object is inserted in the map based on those correspondences. The bottom row shows the dense point cloud of the object over the image, proving the correct alignment.

a packet of chewing gum and a teddy bear– and two planar objects –two postcards. The planar model for each one of the postcards was built from a single image, and the models of the other four objects were constructed from several images taken around each of them (5, 14, 15 and 20, respectively) following the technique described in section IV.

All the six objects that compose our database are detected along the sequence, inserted in the 3D map and tracked the rest of the sequence. Figure 3 shows the results for the recognition thread at the specific frames where the six objects were recognized. The top row shows the frame in the sequence where they were recognized; the middle row the object *face* that was recognized; and the coloured lines stand for the correspondences between the two. In the bottom row it is displayed the reprojection of the dense point cloud model over the top row images.

Figure 4 summarizes the results of this experiment for several steps of the estimation. For each step it is shown the current frame and a 3D view of the estimation. The 3D view show the uncertainty ellipses for each point in the 3D map, the dense point clouds modeling the objects and the camera trajectory as a yellow line. The tracked points are also displayed over the current frame –an ellipse shows the search area and a square frames the actual match– in different colours: red stands for successfully tracked points, blue for those rejected by low patch cross-correlation, magenta for those rejected by our 1-point RANSAC, white for points succesfully tracked in known objects and orange for points not matched in a known object.

Figure 4.a shows the estimation results at step $\#610$, when no object has been inserted yet. Figure 4.b shows frame $\#1359$, just after the tetra pack has been detected and inserted. Figures 4.c, 4.d and 4.e show respectively that the Volkswagen replica, the chewing gum packet and the postcard have been inserted in the map and are being tracked. Their correspondent frames in the sequence are $\#2764, \#4062, \#4725$. Figure 4.f shows the results at step $\#7102$, when the two latest objects –the teddy bear and the postcard– have been inserted. In figure 4.g the camera has gone back to the starting point (frame $\#8538$), imaging again the tetra pack and the Volkswagen replica. Finally, figure 4 is the last frame of the sequence, showing only the objects in the SLAM map. We strongly recommend to watch the video accompanying the paper for a better understanding of this experiment [1].

Finally, figure 5 shows the computational time (thick blue line) along with the state vector size (thin red line) for this experiment. The experiment was run in an Intel Core i7 at 2.66 GHz. In the worst case, the proposed algorithm runs succesfully at 7 Hz for a state size of around 600 and typical hand-held camera motions.

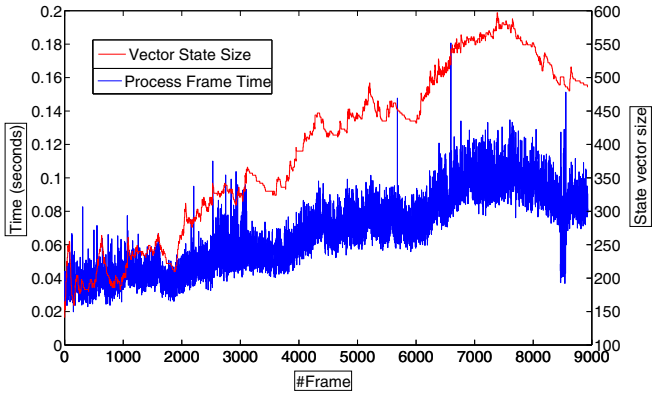

Figure 5. Computational cost (thick blue line) and state vector size (thin red line) for the experiment. Notice that the algorithm runs, in the worst case, at around 7 Hz for a state vector of size 600.

[1] A high resolution version of the video can be found at `http://webdiis.unizar.es/~jcivera/videos/iros11_desktop.avi`

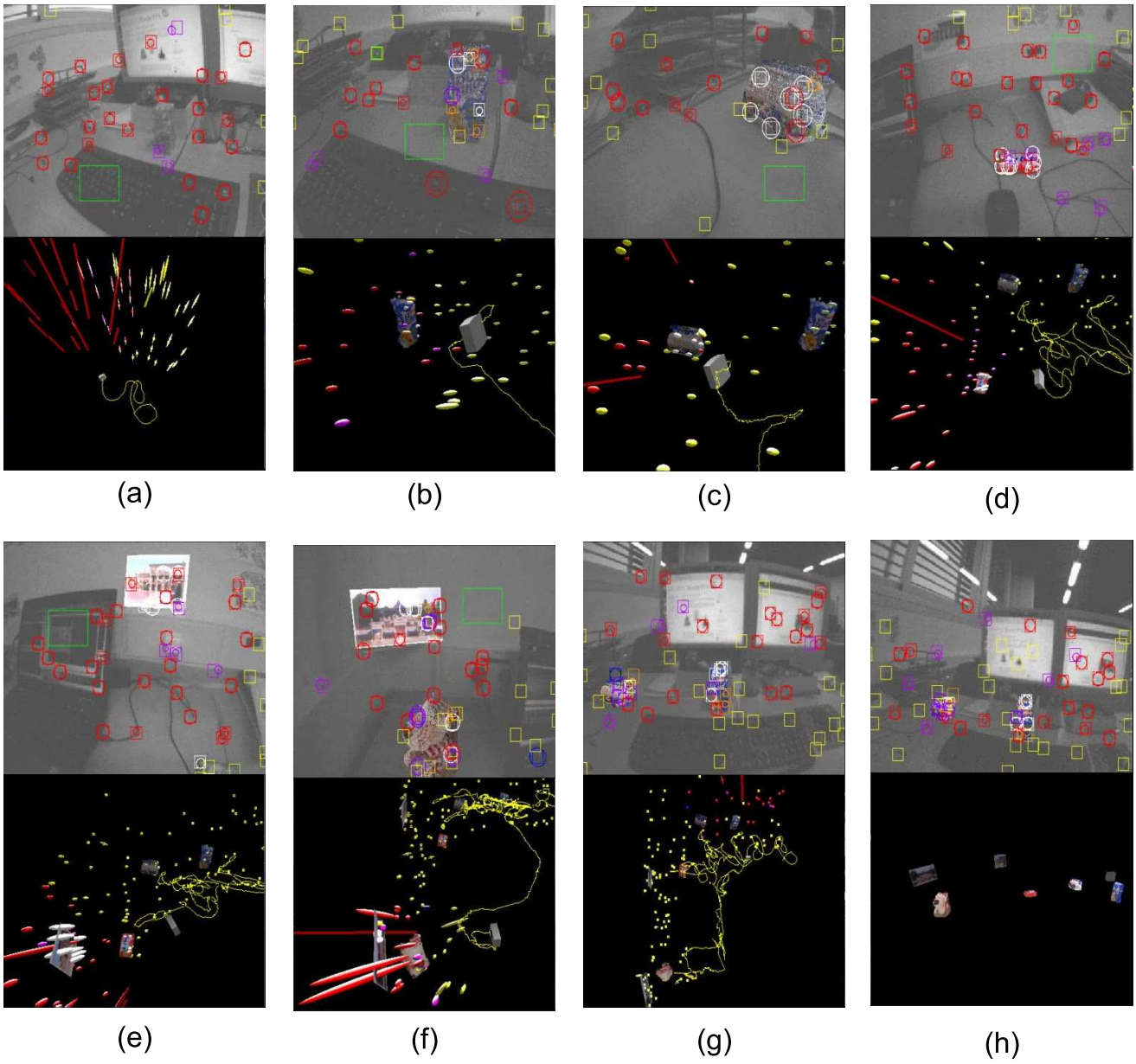

Figure 4. Representative images (at the top) and 3D estimation at their respective times (at the bottom) for the desktop experiment. (a) Initial map, still with no recognized objects. (b) The tetra pack has been recognized, inserted and is being tracked. (c), (d) and (e): the Volkswagen replica, chewing gum packet and the postcard has been inserted. (g) The two remaining objects –postcard and teddy bear– are inserted. (h) The camera moves back close to the starting position, revisiting all previous objects. (h) Final frame of the sequence and 3D view of the objects registered in a common reference frame.

B. Hospital Room Environment –RoboEarth Project

Before going into detail about the performance of the semantic Monocular SLAM, the general aim of this experiment should be stated. This experiment is framed into the RoboEarth project [27]; dedicated to construct a giant network and database repository for robots to share knowledge by uploading and downloading actions and sensory data. In this specific experiment a robot enters a hospital room and downloads from the RoboEarth 1) the recognition models for the objects it may encounter in the hospital room, and 2) an action recipe consisting on the task of serving a tetra pack juice to a patient in the bed. The semantic Monocular SLAM algorithm described in this paper, using the downloaded tetra pack recognition model among others, estimated the partially annotated map that allowed the robot to succesfully grasp the tetra pack and serve it to a patient [2].

The sequence for this experiment has 6003 frames. The object models considered in this experiment were the cabinet, the bed and the tetra pack of juice. The tetra pack is the same than the one in the previous experiment. The cabinet and bed are roughly modeled as delimited by planar faces. Figure 6 shows several frames extracted from the experiment.

[2]A video of the complete RoboEarth experiment can be seen at `http://www.youtube.com/watch?v=RUJrZJyqftU`

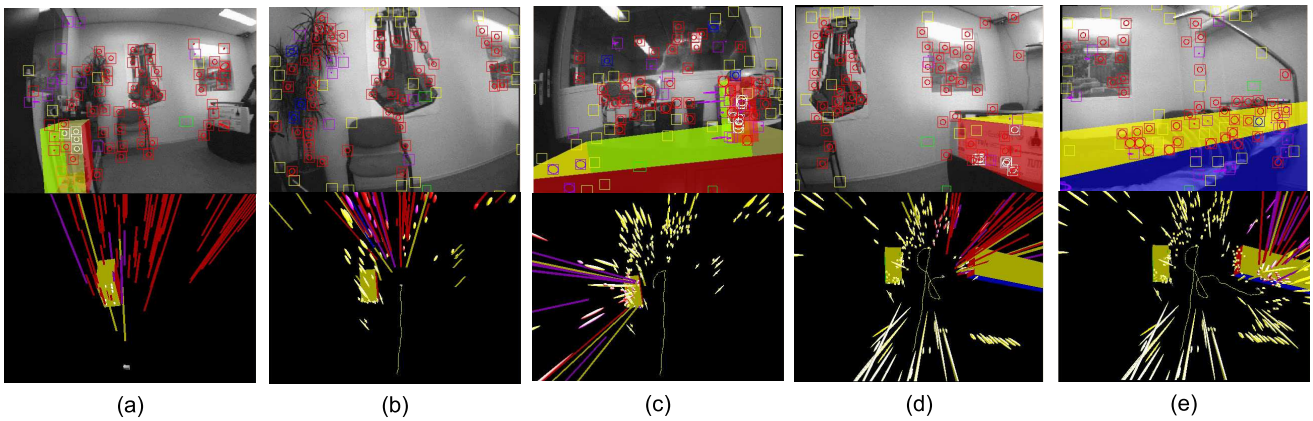

Figure 6. Representative images (at the top) and 3D estimation at their respective times (at the bottom) for the hospital room experiment. The tracked features are displayed in the images as circles. The inserted objects, represented as coloured prismatic solids, are also reprojected at the images. The 3D views show the camera trajectory up to this frame as a yellow line, the point feature uncertainties as ellipses and the inserted objects. (a), frame #34, the cabinet has been already recognized and inserted. (b), frame #241, the robot goes forward. Notice that, although the cabinet is not seen in the image, its 3D position remains registered. (c), frame #912, the robot turns left and faces the cabinet and the tetra pack, which is detected and registered. (d), the robot turns, recognizing and registering the bed. (e) Robot location at the end of the experiment.

Figure 6.a (top) shows the estimation at frame #34, where the cabinet has been already recognized and inserted. Notice the accuracy of the insertion by the overlap between the model reprojection and the real cabinet. In the bottom it can be seen a top view of the 3D map: the ellipses stand for the uncertainty regions of the salient point features and the coloured prism is the cabinet model. As every object in this experiment is composed of planar faces, they are represented by coloured prisms instead of the point clouds of the previous experiment.

The following images in figure 6 stand for other frames of the sequence temporally ordered. Notice that in figure 6.c the tetra pack was already recognized and inserted. In figure 6.d the bed has also been recognized and registered. Finally, figure 6.e shows the final frame of the sequence along with the final estimation results[3]. For a better visualization, figure 7 shows a detail of the final map estimation and the camera trajectory. Notice the accurate registration of the tetra pack over the cabinet in figure 7.b.

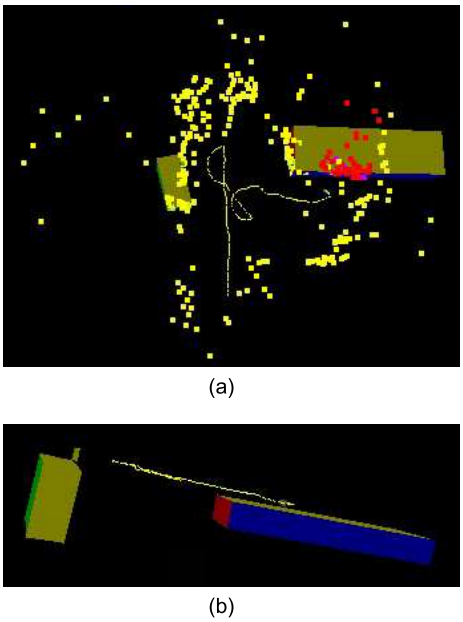

Figure 7. SLAM results at the end of the hospital room experiment. (a), top-view of the camera trajectory, estimated salient point features and recognized objects. (b), side-view of the camera trajectory and recognized objects: the tetra pack over the cabinet, both at the left; and the bed at the right.

VIII. CONCLUSIONS AND FUTURE WORKS

State-of-the-art monocular SLAM, Structure from Motion and object recognition are combined in this paper to allow the insertion of precomputed known objects into a standard point-based monocular SLAM map. The only input to the algorithm is visual information: a monocular sequence feeds the EKF SLAM algorithm; the appearance and geometric models for the known objects are precomputed from a set of sparse images; and also object recognition is driven by visual features. Experimental results show the feasibility of the algorithm and its real-time capabilities for room-sized scenarios.

The approach described on the paper is the first one introducing general 3D objects in a geometric SLAM map in real-time. But we still believe that the main value of our paper resides on the concept behind it. On the one hand recent research on visual object recognition allows to robustly recognize a wide extent of objects from visual input; but 3D information is rarely considered in those approaches. On the other hand, monocular SLAM and Structure from Motion currently offer real-time camera motion and 3D scene estimation but without a semantic meaning. The combination presented on this paper, providing a partially annotated local map and the current robot position, could be of high value for certain robotic tasks like grasping (as demonstrated in

[3]A high resolution version of the video can be found at `http://webdiis.unizar.es/~jcivera/videos/iros11_hospital_room.avi`

the RoboEarth experiment).

Regarding the cameras used: Only the calibration for the SLAM camera is needed, and the camera used for building the models can be different from the SLAM one. Any robot with a calibrated camera would be able to exploit then the precomputed models, what makes the system interoperable. The robot ends up with the location of the object it is supposed to interact with under quite general circunstantes. Finally just by recognizing an object face, the map can incorporate information about object regions that are not observed. This might be quite useful for tasks like robot navigation.

Several interesting lines for future work arise from the results on this paper. First, if would be very interesting to increase the quality and density of the semantic annotations. For example, the classical object recognition algorithms used in this paper could be upgraded to the most recent *category* recognition algorithms [10]. This would allow to recognize generic categories (e.g., the category chair) instead of objects, which are specific instantiations of a category (e.g., a specific chair). Context-based object detection [20] or image segmentation [13] could also help to augment the density of the annotated objects. Second, monocular SLAM algorithms providing with denser geometric maps [21], [25] could be used in order to help robotic tasks like navigation or path planning.

IX. ACKNOWLEDGMENTS

The research leading to these results has received funding from the European Union Seventh Framework Programme FP7/2007-2013 under grant agreement number 248942 RoboEarth, the Dirección General de Investigación of Spain under projects DPI2009-13710 and DPI2009-07130, and the Ministerio de Educación (scholarship FPU-AP2008-02272). The authors are grateful to the University of Oxford (Brian Williams and Ian Reid) and to Imperial College London (Andrew J. Davison) for software collaboration and to the members of RoboEarth consortium.

REFERENCES

[1] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool. Surf: Speeded up robust features. *Computer Vision and Image Understanding*, 110(3):346–359, 2008.

[2] C. Cadena, D. Gálvez-López, F. Ramos, J. Tardós, and J. Neira. Robust place recognition with stereo cameras. In *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems*, Taipei, Taiwan, October 2010.

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

[4] R. O. Castle, D. J. Gawley, G. Klein, and D. W. Murray. Towards simultaneous recognition, localization and mapping for hand-held and wearable cameras. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 4102–4107, 2007.

[5] J. Civera, A. J. Davison, and J. M. M. Montiel. Inverse depth parametrization for monocular SLAM. *IEEE Transactions on Robotics*, 24(5):932–945, October 2008.

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

[7] M. Cummins and P. Newman. FAB-MAP: Probabilistic localization and mapping in the space of appearance. *The International Journal of Robotics Research*, 27(6):647, 2008.

[8] H. Durrant-Whyte and T. Bailey. Simultaneous localisation and mapping (SLAM): Part I the essential algorithms. *Robotics and Automation Magazine*, 13(2):99–110, 2006.

[9] S. Ekvall, P. Jensfelt, and D. Kragic. Integrating active mobile robot object recognition and slam in natural environments. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5792–5797, 2006.

[10] V. Ferrari, F. Jurie, and C. Schmid. From images to shape models for object detection. *International journal of computer vision*, 87(3):284–303, 2010.

[11] Y. Furukawa and J. Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(8):1362–1376, 2010.

[12] C. Galindo, A. Saffiotti, S. Coradeschi, P. Buschka, J. Fernandez-Madrigal, and J. Gonzalez. Multi-hierarchical semantic maps for mobile robotics. In *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 2278 – 2283, aug. 2005.

[13] S. Gould, R. Fulton, and D. Koller. Decomposing a scene into geometric and semantically consistent regions. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1–8. IEEE, 2010.

[14] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, 2004.

[15] S. Hinterstoisser, V. Lepetit, S. Ilic, P. Fua, and N. Navab. Dominant orientation templates for real-time detection of texture-less objects. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2257 –2264, jun. 2010.

[16] M. Martinez, A. Collet, and S. Srinivasa. Moped: A scalable and low latency object recognition and pose estimation system. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 2043 –2049, may. 2010.

[17] D. Meger, P. Forssén, K. Lai, S. Helmer, S. McCann, T. Southey, M. Baumann, J. Little, and D. Lowe. Curious George: An attentive semantic robot. *Robotics and Autonomous Systems*, 56(6):503–511, 2008.

[18] F. Moreno-Noguer, V. Lepetit, and P. Fua. Accurate non-iterative o(n) solution to the pnp problem. *Computer Vision, IEEE International Conference on*, 0:1–8, 2007.

[19] M. Muja and D. G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *International Conference on Computer Vision Theory and Application VISSAPP'09)*, pages 331–340. INSTICC Press, 2009.

[20] K. Murphy, A. Torralba, and W. Freeman. Using the forest to see the trees: a graphical model relating features, objects and scenes. *Advances in Neural Information Processing Systems*, 16, 2003.

[21] R. Newcombe and A. Davison. Live dense reconstruction with a single moving camera. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1498–1505. IEEE, 2010.

[22] A. Pronobis, O. Martinez Mozos, B. Caputo, and P. Jensfelt. Multi-modal semantic place classification. *The International Journal of Robotics Research*, 29(2-3):298, 2010.

[23] A. Ranganathan and F. Dellaert. Semantic modeling of places using objects. In *Robotics: Science and Systems*, 2007.

[24] N. Snavely, S. Seitz, and R. Szeliski. Modeling the world from internet photo collections. *International Journal of Computer Vision*, 80(2):189–210, 2008.

[25] H. Strasdat, J. Montiel, and A. Davison. Scale drift-aware large scale monocular SLAM. In *Proceedings of Robotics: Science and Systems (RSS)*, 2010.

[26] S. Vasudevan, S. Gächter, V. Nguyen, and R. Siegwart. Cognitive maps for mobile robots-an object based approach. *Robot. Auton. Syst.*, 55:359–371, May 2007.

[27] M. Waibel, M. Beetz, R. D'Andrea, R. Janssen, M. Tenorth, J. Civera, J. Elfring, D. Gálvez-López, K. Haussermann, J. Montiel, A. Perzylo, B. Schiele, O. Zweigle, and R. van de Molengraft. Roboearth - a world wide web for robots. *IEEE Robotics and Automation Magazine*, 18(2):69–82, june 2011.

[28] B. Williams, G. Klein, and I. Reid. Real-time SLAM relocalisation. In *IEEE 11th International Conference on Computer Vision*, page 1:8, 2007.

[29] H. Zender, O. Martínez Mozos, P. Jensfelt, G. Kruijff, and W. Burgard. Conceptual spatial representations for indoor mobile robots. *Robotics and Autonomous Systems*, 56(6):493–502, 2008.