Removing Dynamic Objects from 3D Maps using Geometry and Learning

Berta Bescós, José M. Fácil, Javier Civera and José Neira

Abstract—In this work we present a new approach for the 3D reconstruction of a scene from RGB-D sequences containing dynamic objects. This challenging problem includes the detection of such objects, as well as the reconstruction of those parts of the scene occluded by them. We propose a combination of computer vision geometry (detection and tracking of dynamic keypoints and associated image regions) and machine learning techniques (Fully Convolutional Neural Networks and Generative Adversarial Networks), which allows us to detect not only objects that are known to be dynamic (e.g., people) but also other elements that change place in the scene (e.g., books carried by people). Our system detects them, removes them from the map and also reconstructs the occluded parts of the scene using information from alternative views.

I. INTRODUCTION

Obtaining an accurate 3D reconstruction of a scene from a sequence of images is one of the key challenges in computer vision. The research community has addressed this problem in different ways. Some methods [1], [2], [3] rely on feature-based algorithms and can only reconstruct a sparse set of salient points. Other works [4], [5], [6] are able to estimate a completely dense reconstruction of the scene by the direct minimization of the photometric error—instead of the reprojection error of the feature-based methods—and a regularization term by including the total variation norm to the cost function. Other approaches within such direct—or photometry-based—methods [7], [8] estimate a semi-dense but more accurate map of the scene. Building upon these ideas, following works [9], [10] suggest to include piece-wise assumptions to achieve a dense or quasi-dense fairly accurate reconstruction. More recently, with the well-known boost of deep learning, some ideas have appeared to combine geometry and learning in order to increase the accuracy of monocular maps [11], [12].

None of these methods have proposed a robust algorithm for dealing with the very common problem of dynamic objects in the scene, e.g., people walking around, people moving objects, animals, bicycles or cars, etc. Detecting and dealing with dynamic objects in 3D scene reconstructions reveals several challenging problems in mapping, including:

1) How to detect such dynamic objects in the images.

2) How to prevent the mapping algorithm from including moving objects as part of the 3D map.

3) How to complete the 3D information in the scene occluded by the moving objects.

Many applications would be greatly benefited from a solution for this problem, e.g., augmented reality, autonomous cars and medical imaging, among others. In the main, all of them that, for instance, may reuse maps. Some of these applications may only need detection and removal of dynamic objects to fairly succeed. However, their performance would be improved thanks to the useful information we could provide by the reconstruction of the occluded areas. In some cases the benefits could be present in very critical situations, e.g., avoidance of occluded obstacles or realistic virtual insertions in AR.

In this work we propose a pipeline for both detecting dynamic objects and, after having removed them from the images, filling the holes in the images with the correct information of the scene. For that purpose, we combine geometric models and deep-learning-based algorithms. This
Algorithm 1 Features that belong to dynamic objects

Precondition: Denote $x$ as each of the keypoints from each selected keyframe, $X$ as its corresponding 3D scene point, and $x'$ as its projection in the current frame.

1: for each feature point $x$ in each selected keyframe, do
2: Compute its projection $x'$ in the current frame
3: if $\alpha \leq 30^\circ$, then
4: \[ H_{\text{dist}}(X) \leftarrow \min_d (B_x \oplus B_{x'+d}) \] \hspace{1cm} $\triangledown \oplus$: bitwise exclusive-or
5: if $H_{\text{dist}}(X) > \tau_B$ then
6: if $x'$ does not lay on an object border, then
7: $x'$ belongs to a dynamic object
8: end if
9: end if
10: end if
11: end for

This approach gives a more accurate, realistic and reusable map of the scene. An overview of our proposal is seen in Fig. 1.

II. RELATED WORK

The standard approach for dealing with dynamic objects in feature-based SLAM is to detect them and then label them as outliers. ORB-SLAM [3] generally succeeds in ignoring moving objects by setting their corresponding keypoints as outliers thanks to RANSAC algorithms and the use of distant keyframes (temporally and spatially). However, when dealing with dynamic environments, the system becomes less accurate as the objects that have remained static in several keyframes are mapped in the reconstruction. Another approach to deal with this problem lies on detecting the changes that have taken place in the scene by projecting the features from the keyframes to the current frame for appearance and structure comparison. If features are considered to be dynamic, they are not used anymore [13]. Another work that similarly manages to remove the moving outliers by tracking known 3D objects in the scene is the one proposed by Wlangsiripitak and Murray [14]. The direct method presented by Concha and Civera [9] (among others) includes a robust cost function in the optimization that allows to deal with occlusions, including in some cases those produced by dynamic objects (a more detailed study can be found at the cost function evaluation of Concha and Civera [15]). In the same way as the feature-based algorithms, this technique moderates the influence of outliers in the optimization, e.g., the camera pose estimation. Nevertheless, the non-inclusion of dynamic objects in the map reconstruction is not ensured.

With hand-held RGB-D cameras, our interest here lies in detecting the dynamic objects, removing them from the images and reconstructing the scene as if it only contained static objects. An intuitive idea of our purpose is the diminished reality [16], i.e., we want to remove some parts of the real world. The problem becomes more challenging as we want to remove dynamic objects and reconstruct real information.

III. DETECTION OF MOVING OBJECTS

In order to detect the moving objects we propose two different approaches, that can be used in a combined and complementary manner for improving the robustness.

A. Geometrical Approach

The first approach deals with the problem by comparing extracted point features with their reprojection into other frames. We assume that we have a subset of images from the dataset that contain no moving objects—we will explain how to obtain this subset later on in the survey... These images are used as reference (ten static images are enough in our experiments). Differently to Tan et al. [13], once we have this subset we extract their ORB features, instead of the SIFT features. ORB are binary features invariant to rotation and scale (in a certain range), resulting in a very fast recognizer with good viewpoint invariance [17].

For each input frame, since we know its pose from ORB-SLAM, we select those images within the initial subset that have the maximum overlapping of the scene with itself. This is done taking into account both the distance and the rotation between the new frame and each of the static frames, similarly to Tan et al. [13]. The number of close frames has been set to five in our experiments.

We then compute the projection of each keypoint $x$ of the chosen static images (its corresponding 3D point is $X$) into the current frame, obtaining the keypoint $x'$. For each keypoint the parallax angle $\alpha$ is calculated. If this angle is greater than $30^\circ$, its corresponding keypoint might be subject to occlusions, and will be ignored from now on. We calculate the ORB descriptors of the left keypoints in the current frame $B_{x'}$, taking into account the reprojection error (using a small translational vector $d$), and we compare them with the already computed ORB descriptors of the reference frames $B_x$. The difference between them, $H_{\text{dist}}$, is calculated with the Hamming-distance, that in the case of binary vectors can be calculated using the bitwise exclusive or operator. If the Hamming-distance of a 3D point $X$, $H_{\text{dist}}(X)$, is greater than a threshold $\tau_B$, the keypoint correspondent to the new frame $x'$ will be considered as dynamic, if not, the keypoint will be considered as static. This is all described in Algorithm 1.

Some of the keypoints that have been previously set to dynamic might lay on the border of a moving object, potentially causing future problems. In order to avoid this, we use the information given by the depth images. If a keypoint is considered to be dynamic, but a patch around itself has a high variance, this keypoint will no longer be tagged as dynamic.

So far, we know which keypoints belong to dynamic objects, and which ones do not. In order to classify all the pixels belonging to these objects, we grow the region around those pixels that were set as dynamic in the depth image. This segmented image projected on the RGB frame can be seen in Fig. 2a. Our results show small misalignments and errors due to the time difference between RGB and depth images, and depth discontinuities inside a moving object itself. We dilate the segmented regions in order to avoid both effects.
B. Machine Learning Approach

For detecting dynamic objects we propose to use a Fully Convolutional Neural Networks (FCN) to obtain a semantic segmentation of the images. In our experiments we have used the implementation of Shelhamer et al. [18], [19] (specifically the 8 pixel prediction stride version, trained with the PASCAL VOC 2010). The input to the FCN is the RGB original image. The idea is to manually select those classes that are potentially dynamic in the image, e.g., people, dog, cat, car, and remove them from the image. Currently, we are only considering the “person” class in the segmentation as dynamic object. A qualitative result can be seen in the Fig. 2b. Recent results on instance object segmentation [20] show an impressive performance, and we consider that our proposal will benefit from their advances.

C. Geometry-Learning Combination

We can find several advantages and disadvantages in both methods. Firstly, using geometric approaches, the main problem is that initialization is not trivial. A few frames need to be selected manually such that they are different enough and that are also known to contain no dynamic objects. Learning methods, and their impressive performance using a single view, do not have such initialization problems.

On the other hand, the main limitation of the machine learning method is that objects that are supposed to be static (for example a book) can be moved by dynamic objects, and the method is not able to identify them. This causes defects on the reconstruction of the scene. Such defects can be solved by checking the multi-view compatibility.

These two ways of facing the moving objects detection problem are illustrated in Fig. 2. In Fig. 2a we see that the person in the back, which is potentially a dynamic object, is not detected. This is due to both the difficulties that RGB-D cameras face when measuring the depth of objects that are far, and the fact that reliable features lie on defined, and therefore nearby, parts of the image. However, this person is detected by the machine learning method (Fig. 2b). Apart from this, on one hand we see in the Fig. 2a that not only is detected the person in the front of the image, but also the book he is holding and the chair he is sitting on. On the other hand, in the Fig. 2b the two people are the only objects detected as dynamic, and also their segmentation is less accurate. If only the machine learning method is used, a floating book would be left in the images and would incorrectly become part of the 3D map.

Because of the advantages and disadvantages of both methods, we consider they are complementary and therefore the combined use of both is an effective way to achieve an accurate mapping. The initialization can be done automatically by using the learning based method for the detection of known dynamic objects. A subset of the images that do not contain dynamic objects can be used as reference for the geometrical approach. In case that this subset did not contain enough static images, regions of images that contain no moving objects would be used as reference. Once the initialization is solved, the segmented frame should show all
the objects that have moved with respect to other frames, including objects that are not considered as dynamic, e.g., a book. In order to achieve this goal, if an object has been detected with both approaches, the segmentation mask should be that of the geometrical method; but if an object has only been detected by the learning based method, the segmentation mask should contain this information too. The final segmented image of the example explained above can be seen in the Fig. 2c. These segmented parts considered dynamic are then removed from the processed frame.

IV. RECONSTRUCTION OF THE BACKGROUND

For every dynamic object that is removed, we also aim at reconstructing the occluded background, so that we can synthesize a realistic image without moving objects. First we project into the dynamic segments both the color and depth from overlapping static images. The more overlapping the static image, the more similar illumination conditions and the better the synthesis. We do the same with the next static image that has the greatest overlapping, and repeat for all the images that show overlap.

Some gaps have no correspondences and are left blank: it can be due to the difference of field of view between the cameras –two pixels in the static image might correspond to three pixels in the input frame–. Besides, another reason why some areas can not be reconstructed is because the correspondent part of the scene does not appear in the static images. The first ones usually have a small size and can easily be in-painted with the color and the depth from the neighbours. The second ones can not be reconstructed with geometrical methods if that part of the scene has never been seen and would need a more elaborated in-painting.

In order to fill in the already mentioned blank gaps, Tanner et al. [21] suggest a method that consists in surface interpolation. More recently, Guizilini and Ramos [22] deal with partial occlusions and sensor failure by using both a Bayesian Convolutional Variational Auto-Encoder and Hilbert Maps, i.e., they learn to complete partially occluded structures and objects, based on partial views.

In contrast, we propose the use of Generative Adversarial Networks (GAN) to deal with both the blank gaps problem and the non-realist RGB appearance. GAN is a model proposed by Goodfellow et al. [23] for training generative neural networks to produce realistic results. This model has been used to generate from hand-written digits [23] to quasi-indistinguishable realistic-level generations of faces, indoor environments [24], natural images [25] among others. More related to our proposal, Pathak et al. [26] proposed an in-painting model to learn features in an unsupervised way. Focusing on the realistic in-painting model itself, we want to include a similar model into our pipeline in order to fill the holes with real and realistic information of the scene. Real by using the 3D map, reconstructed from the sequence, and realistic by using the in-painting network, trained with RGB-D data.

Our initial proposal is to create a Fully Convolutional Neural Network (FCN) whose work is to transform the initial solution given by the geometric method into a smoother, more realistic solution. This FCN will have as input the output that we get by using multiple views, see Fig. 3. In order to train this FCN we propose to use GAN, based on Pathak et al. [26]. We plan to use a similar architecture, including the two different losses proposed in their work. The adversarial loss to make the image realistic and the $L_2$ to keep it real. However, the problem of the amount of data needed for training a deep neural network remains, given the small number of RGB-D datasets containing dynamic objects. We propose to use the NYU Depth Dataset V2 [27]
to generate synthetic training data. We will crop random but object-like shapes in the images and we will fill them with the geometric method.

V. EXPERIMENTAL RESULTS

We have conducted several experiments in 5 hand-held indoor sequences with dynamic objects of the TUM RGB-D benchmark [28], evaluating the general performance of the system: the detection of moving objects has been done using a combination of the geometrical and machine learning approaches, and the reconstruction of the background has been done so far by projection.

As an example of our procedure, we show in Fig. 3 four input frames from the TUM RGB-D benchmark, and their corresponding synthesized images without the dynamic objects. All the dynamic content have been successfully detected. We can also see that small parts are left because of the segmentation inaccuracies. Most of the segmented parts have been properly reconstructed with the static background. Improving the images appearance, mainly the blank parts and the difference of color between the reconstructed and the original parts of the image, is the goal of our future work.

VI. CONCLUSIONS

We have presented a new approach to compute an accurate 3D reconstruction of a scene that contains moving objects. There are two main and differentiated parts along this study: the detection of the moving objects, and the reconstruction of the background occluded by these objects. In order to deal with these two problems, we propose a combination of geometry-based and machine learning-based methods. The main interest of this study is the possibility of creating reusable maps of the static part of the scene, regardless of the dynamic objects that can appear in a particular moment.

Our preliminary results are promising, we are able to detect all the dynamic objects and reconstruct a coherent background from overlapping views. We are currently working to improve the non-realistic appearance of the images by the use of generative models with GANs. Future work will also consider improving dynamic object detection by incorporating a deeper and complex learning model (e.g. [20]). Another open line for future work is to incorporate this study to a real-time SLAM system, in order to use these advances in the tracking thread, which may improve its accuracy and robustness.

REFERENCES


