Robust and Accurate 3D Mapping by combining Geometry and Machine Learning to deal with Dynamic Objects

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 use 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 (for e.g. people) but also other elements that change place in the scene (e.g. books carried by people). Our system detects them and also reconstructs the hidden parts of the scene in some images using information from alternative images.**

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 reconstruct a widely sparse set of salient points. Other works [4], [5], [6] propose a completely dense reconstruction of the scene by the direct minimization of the photometric error –instead of the geometric error of the feature-based methods– and a regularization term by including the total variation norm to the cost function. Lately, the direct photometric-based methods [7], [8] are facing the problem in a more conservative way, by computing only a semi-dense, but more accurate map of the scene. Building upon this idea, 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-based with learning-based methods [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 reconstruction adds several challenging open problems in mapping:

1) How to detect these 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 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, medical imaging, among others.

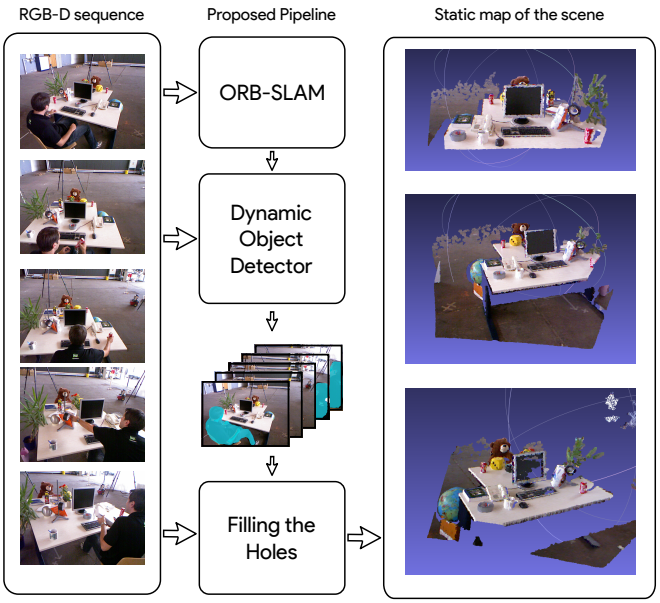

Fig. 1: Overview of our proposed pipeline. Our input is a sequence of RGB-D images. We estimate the pose of the camera using ORB-SLAM [3]. With the RGB-D sequence and the estimated poses, we detect dynamic objects (e.g. people) and remove them from the images. We can then reconstruct a 3D map of the scene, containing only the static part.

The standard approach for dealing with dynamic objects in feature-based SLAM is to detect them and then treat 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

*This work has been supported by NVIDIA Corporation through the donation of a Titan X GPU, by the Spanish Ministry of Economy and Competitiveness (projects DPI2015-68905-P and DPI2015-67275-P, FPI grant BES-2016-077836), and by the Aragón regional government (Grupo DGA T04-FSE).

Berta Bescós, José M. Fácil, Javier Civera and José Neira are with the Instituto de Investigación en Ingeniería de Aragón (I3A), Universidad de Zaragoza, Zaragoza 50018, Spain {bbescos,jmfacil,jcivera,jneira}@unizar.es

Algorithm 1 Features that belong to dynamic objects

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

for each feature point x in each selected keyframe, **do**
  Compute its projection x' in the current frame
  if $\alpha \leq 30°$, **then**
    $H_{dist}(X) \leftarrow \min_d(B_x \oplus B_{x'+d})$
      ▷ $\oplus$: bitwise exclusive-or
    if $H_{dist}(X) > \tau_B$ **then**
      if x' does not lay on an object border, **then**
        x' belongs to a dynamic object
      end if
    end if
  end if
end 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 Wangsiripitak 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 [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. In this work we propose a solution 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 use a combination of geometric and deep-learning-based algorithms. This approach gives a more accurate, realistic and reusable map of the scene. An overview of our proposal can be seen in Fig. 1. We split our problem in two different parts, the detection of moving objects and the reconstruction of the background.

II. Detection of Moving Objects

In order to detect the moving objects we have developed two different approaches, and we use them in a combined and complementary way for more robust results.

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 the keypoints x of the chosen static images (its corresponding 3D point is X) into the current frame, obtaining the keypoints x' . For each keypoint the parallax angle α is calculated. If this angle is greater than 30°, 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_{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, $H_{dist}(X)$, is grater than a threshold τ_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 may 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 have done the region growing of 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. This can carry out some problems due to the time difference between RGB and depth images, and due to the discontinuity of depth inside a moving object itself. Due to the common lack of information of depth in the objects borders, and the already said time difference between RGB and depth images, a dilation of the segmented regions is to be done.

(a) Geometrical Methods

(b) Learning Based Methods

(c) Combining geometrical and learning methods

Fig. 2: Detection and segmentation of dynamic objects using geometrical methods (top, the person in front moving carrying a book), learning based methods (middle, detection of people, even if no motion or if motion is small), and a combination of both geometrical and learning methods (bottom). Figure best viewed in electronic format.

B. Machine Learning Approach

For detecting dynamic objects we propose to use a Fully Convolutional Neural Networks (FCN) to get a semantic segmentation of the images. In our experiments we have used the implementation of Shelhamer *et al.* [18], [19]. Concretely, we are using the 8 pixel prediction stride version, trained with the PASCAL VOC 2010. We feed the FCN with the RGB original image, subtracting firstly the RGB mean of the training data and transforming the image to BGR. 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. In the current version of our work, 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, we consider that our proposal will be benefited from their advances.

C. Combined use of Geometrical and Machine Learning Approaches

We can find several advantages and disadvantages in both methods. Firstly, while using the geometrical approach, 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. With the use of the machine learning method there is no initialization issue. On the other hand, the main constraint 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 leads to a wrong reconstruction of the scene. This issue can be solved by using the geometrical approach.

These two ways of facing the moving objects detection problem can be seen in the Fig. 2. In the 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 the advantages and disadvantages of both methods are complementary, 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 contain no dynamic objects will 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.

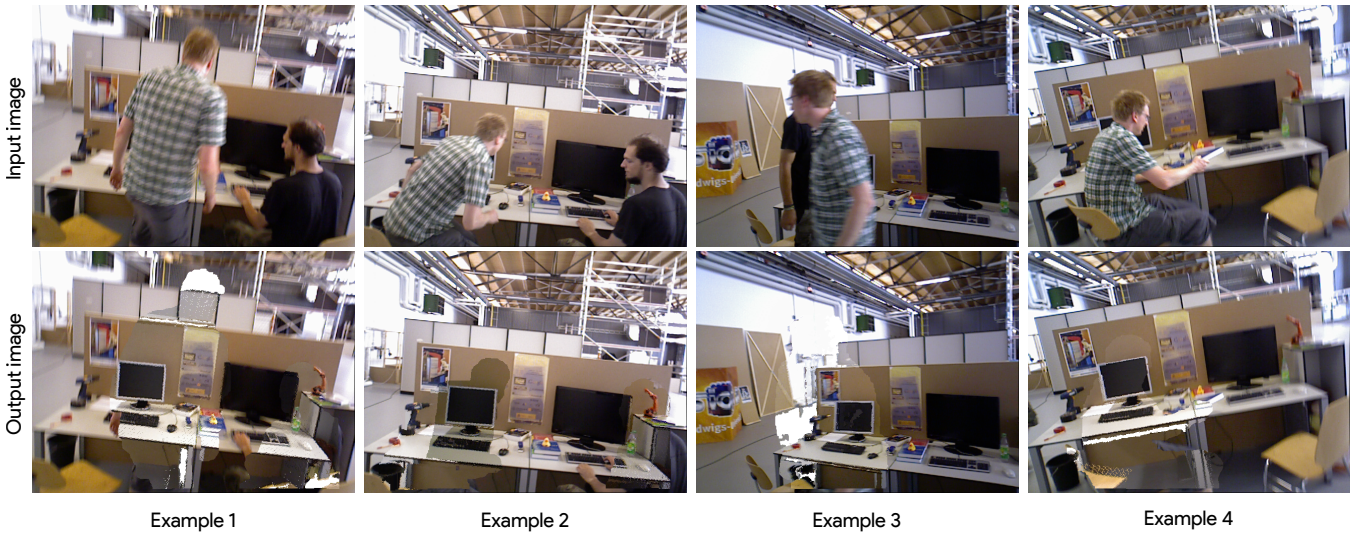

Fig. 3: Our preliminary results for 4 different examples. *The first row* is the original image containing dynamic objects and *the second row* is the current state of our proposal. Notice that the dynamic objects are removed from the image and replaced with real information of the scene. However, there is still room for improvement as the images do not look as realistic as they should.

Once the initialization issue 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. To obtain 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.

III. Reconstruction of the Background

For each of the objects of the frame that have been removed, we aim at reconstructing the background so that it looks like a realistic image with no moving objects. We first project both the color and depth from the static image with more overlapping into the gaps of the frame. The more overlapping the static image has with the current frame, the more similar conditions of illumination there will exist. Secondly, we do the same with the next static image that has the greatest overlapping, and so on.

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 reconstructed by adopting the color and the depth of the closest pixel. The second ones can not be reconstructed with geometrical methods if that part of the scene has never been seen.

For the moment the images are reconstructed, but some of them contain some blank gaps, and some others show slightly different colors between the parts of the original image and the parts that have been reconstructed due to illumination effects. This issue, together with remaining small parts of the moving objects, can lead to a non-realistic result of the image.

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. Hitherto, 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 using GAN.

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 quantity of data needed for training a deep neural network remains, given that there are not a lot of dynamic-objects-based RGB-D datasets. Thus, 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. In this way, we have enough non-realistic but real images with their respective ideal ground-truth.

IV. PRELIMINAR 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 processed images. All the dynamic objects have been successfully detected. We can also see that some small parts of them are left because the segmentation is not always accurate. Most of the segmented parts have been properly reconstructed with the static background. The blank parts left and the difference of color between the reconstructed and the original parts of the image are to be improved in further experiments using GAN.

To sum up, we have presented a new approach to coompute 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 left behind 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 re-usable maps, regardless of the dynamic objects that can be found in the scene.

As it can be seen in the preliminary results there is still room for improvement. We are currently working to improve the non-realistic appearance of the images by the use of generative models with GANs. We hope to report on these results very soon. Future work will also consider improving dynamic object detection by incorporating a deeper and complex learning model (e.g. [20]). Another open way for future work is to incorporate this study to a SLAM system working in real time, in order to able to use these advances in the tracking optimization, and therefore, creating a more robust and accurate SLAM system.

REFERENCES

[1] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd, "Real time localization and 3D reconstruction," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 1, pp. 363–370, IEEE, 2006.

[2] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*, pp. 225–234, IEEE, 2007.

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

[4] J. Stühmer, S. Gumhold, and D. Cremers, "Real-time dense geometry from a handheld camera," in *Joint Pattern Recognition Symposium*, pp. 11–20, Springer, 2010.

[5] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "Dtam: Dense tracking and mapping in real-time," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, pp. 2320–2327, IEEE, 2011.

[6] G. Graber, T. Pock, and H. Bischof, "Online 3D reconstruction using convex optimization," in *2011 IEEE International Conference on Computer Vision Workshops*, pp. 708–711, IEEE, 2011.

[7] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *European Conference on Computer Vision*, pp. 834–849, Springer, 2014.

[8] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[9] A. Concha and J. Civera, "DPPTAM: Dense piecewise planar tracking and mapping from a monocular sequence," in *IEEE/RSJ international conference on intelligent robots and systems*, 2015.

[10] P. Pinies, L. M. Paz, and P. Newman, "Dense mono reconstruction: Living with the pain of the plain plane," in *2015 IEEE International Conference on Robotics and Automation*, pp. 5226–5231, 2015.

[11] J. M. Fácil, A. Concha, L. Montesano, and J. Civera, "Single-View and Multiview Depth Fusion," *IEEE Robotics and Automation Letters*, vol. 2, no. 4, pp. 1994–2001, 2017.

[12] K. Tateno, F. Tombari, I. Laina, and N. Navab, "CNN-SLAM: Real-time dense monocular SLAM with learned depth prediction," *arXiv preprint arXiv:1704.03489*, 2017.

[13] W. Tan, H. Liu, Z. Dong, G. Zhang, and H. Bao, "Robust monocular SLAM in dynamic environments," in *Mixed and Augmented Reality (ISMAR), 2013 IEEE International Symposium on*, pp. 209–218, IEEE, 2013.

[14] S. Wangsiripitak and D. W. Murray, "Avoiding moving outliers in visual SLAM by tracking moving objects," in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pp. 375–380, IEEE, 2009.

[15] A. Concha and J. Civera, "An evaluation of robust cost functions for rgb direct mapping," in *Mobile Robots (ECMR), 2015 European Conference on*, pp. 1–8, IEEE, 2015.

[16] J. Herling and W. Broll, "Advanced self-contained object removal for realizing real-time diminished reality in unconstrained environments," in *Mixed and Augmented Reality (ISMAR), 2010 9th IEEE International Symposium on*, pp. 207–212, IEEE, 2010.

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

[18] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3431–3440, 2015.

[19] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 4, pp. 640–651, 2017.

[20] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," *arXiv preprint arXiv:1703.06870*, 2017.

[21] M. Tanner, P. Piniés, L. M. Paz, and P. Newman, "What lies behind: Recovering hidden shape in dense mapping," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pp. 979–986, IEEE, 2016.

[22] V. Guizilini and F. Ramos, "Learning to reconstruct 3d structures for occupancy mapping.," in *Robotics: Science and Systems*, 2017.

[23] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, pp. 2672–2680, 2014.

[24] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[25] H. Zhang, T. Xu, H. Li, S. Zhang, X. Huang, X. Wang, and D. Metaxas, "Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks," *arXiv preprint arXiv:1612.03242*, 2016.

[26] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2536–2544, 2016.

[27] P. K. Nathan Silberman, Derek Hoiem and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *ECCV*, 2012.

[28] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pp. 573–580, IEEE, 2012.