**QUEEN'S
UNIVERSITY
BELFAST**

A novel vSLAM framework with unsupervised semantic segmentation based on adversarial transfer learning

Jin, S., Chen, L., Sun, R., & McLoone, S. (2020). A novel vSLAM framework with unsupervised semantic segmentation based on adversarial transfer learning. *Applied Soft Computing*, 90, [106153].
<https://doi.org/10.1016/j.asoc.2020.106153>

Published in:
Applied Soft Computing

Document Version:
Peer reviewed version

Queen's University Belfast - Research Portal:
[Link to publication record in Queen's University Belfast Research Portal](#)

Publisher rights

© 2020 Elsevier Ltd.

This manuscript version is made available under the CC-BY-NC-ND 4.0 license <http://creativecommons.org/licenses/by-nc-nd/4.0/>, which permits distribution and reproduction for non-commercial purposes, provided the author and source are cited.

General rights

Copyright for the publications made accessible via the Queen's University Belfast Research Portal is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy

The Research Portal is Queen's institutional repository that provides access to Queen's research output. Every effort has been made to ensure that content in the Research Portal does not infringe any person's rights, or applicable UK laws. If you discover content in the Research Portal that you believe breaches copyright or violates any law, please contact openaccess@qub.ac.uk.

A Novel vSLAM Framework with Unsupervised Semantic Segmentation Based on Adversarial Transfer Learning

Sheng Jin^a, Liang Chen^{a,*}, Rongchuan Sun^a, Seán McLoone^b

^a School of Mechanical and Electric Engineering, Soochow University, Suzhou 215021, PR China

^b School of Electronics, Electrical Engineering and Computer Science, Queen's University Belfast, UK

Abstract:

Significant progress has been made in the field of visual Simultaneous Localization and Mapping (vSLAM) systems. However, the localization accuracy of vSLAM can be significantly reduced in dynamic applications with mobile robots or passengers. In this paper, a novel semantic SLAM framework in dynamic environments is proposed to improve the localization accuracy. We incorporate a semantic segmentation model into the Oriented FAST and Rotated BRIEF-SLAM2 (ORB-SLAM2) system to filter out dynamic feature points, but we encounter one main challenge, i.e. the performance of a segmentation network well-trained with labeled datasets may decrease seriously in a real application without any labeled data due to the inconsistency between the source domain and the target domain. Therefore, we proposed an unsupervised semantic segmentation model with a Residual Neural Network (ResNet) structure, which is trained by the adversarial transfer learning method in the multi-level feature spaces. This work may be the first to perform multi-level feature space adversarial transfer learning for the semantic SLAM task in dynamic environments. In order to evaluate our method, images of indoor scenes from three datasets are used as the source domain, and the dynamic sequences of the TUM dataset are used as the target domain. The extensive experimental results show favorable performance against the state-of-the-art methods in terms of the absolute trajectory accuracy and image semantic segmentation quality.

Keywords—semantic slam; dynamic environments; dynamic object segmentation

1. Introduction

vSLAM is a hot issue in the field of robotics and visual navigation. vSLAM mainly solves two problems. One is to estimate the motion trajectory of the robot through image sequence taken by the camera; the other is to reconstruct the geometry structure in a three-dimensional scene at the same time. In recent years, vSLAM has made significant progress with the rapid development of computing power and sensing technology.

vSLAM systems can be divided into two different types: feature-based methods and direct methods. Among feature-based methods, ORB-SLAM2 [1] is one of the most popular systems. ORB-SLAM2 innovatively uses three threads, including the tracking

* Corresponding author E-mail address: ChenL@suda.edu.cn (L. Chen).

thread, the local mapping thread and the loop closing thread. The tracking thread extracts ORB feature points and estimates the camera pose. The local mapping thread optimizes feature points and camera poses in local maps. The loop closing thread optimizes the pose graph composed of the poses of the key-frames. In direct methods, camera pose is estimated directly from the intensity values of the image. Large Scale Direct monocular SLAM (LSD-SLAM) is a direct (feature-less) monocular SLAM algorithm, which constructs a semi-dense depth map and it is used to match the next key-frame to estimate camera poses [2]. These mainstream vSLAM systems can achieve high absolute trajectory accuracy in a static environment. However, in a dynamic environment, feature-based vSLAM systems are easy to be affected by moving objects. Matching feature points on dynamic objects will result in poor localization accuracy. It greatly limits the application of feature-based vSLAM in many cases, such as working in a real environment where many people are walking around in the long term [3].

Besides, vSLAM systems can only provide the geometric information of the environment for mobile robots. The mobile robot only knows where the surrounding environment can pass and where there are obstacles. The 2D or 3D map constructed by the existing vSLAM systems can only be used to avoid obstacles and plan paths, which is not enough in more advanced tasks [4]. Therefore, it's of great significance to implement semantic SLAM. First, the map information generated by the semantic SLAM contains not only geometric information but also more advanced semantic information. The mobile robot can have a more advanced understanding of the surrounding environment, which can further help the robot to complete more advanced tasks. Second, semantic information can also be exploited to improve localization accuracy [5]. By combining semantic segmentation models [6,7], feature points belonging to dynamic objects can be filtered out. In this way, the localization accuracy of vSLAM can be improved and robotic perception of the environment can be much stronger [8,9].

However, most semantic SLAM systems use the semantic segmentation models trained on datasets that are irrelevant to the vSLAM datasets and are not further fine-tuned on the vSLAM datasets [3,10]. These semantic segmentation models well-trained on training datasets may not generalize well in real applications. In order to improve the segmentation results, it is necessary to fine-tune the semantic segmentation model by supervised learning according to the real application scenes, i.e. we need to re-annotate a large amount of per-pixel ground truths belong to the real application scenes, which is time-consuming and impractical. To address this issue, adversarial transfer learning techniques have been proposed to narrow the gap between the training dataset and the real application scenes and to make features better generalize to unseen scenes without re-annotating per-pixel ground truths [11,12]. Adversarial transfer learning based on generative adversarial networks (GAN) [13] is able to transfer rich information from the training dataset to the real application scenes [14]. Benefited from transferable information that is applicable to both the training dataset and the real application scenes, the model in the real application is not needed to be trained from scratch, which can help carry out unsupervised learning in principle.

In this paper, we propose a novel semantic SLAM framework with Unsupervised Semantic Segmentation (hereafter for short USS-SLAM), which can provide high localization accuracy and semantic information in dynamic environments. First, we combine a semantic segmentation model with the multi-view geometry method in the tracking thread of the ORB-SLAM2 system to filter out feature points belonging to dynamic objects in the environment to improve the localization accuracy. Second, an unsupervised adversarial transfer learning method is proposed to improve the semantic segmentation performance. By adversarial transfer learning, transferable information from the training dataset can be brought to the real application scenes without labeling per-pixel ground truths. In order to transfer more rich information at different levels of the semantic segmentation model, we perform the adversarial transfer learning in the multi-level feature spaces.

2. Related Work

2.1 Semantic Segmentation

Image semantic segmentation is the assignment of an object category label to each pixel in the image at the pixel level [15]. Recently, semantic segmentation models have made many breakthroughs in algorithm design. Convolutional Neural Networks (CNNs) have achieved great success in semantic segmentation. CNN-based methods generally extract visual features of input images through layer-wise abstraction of multiple convolutional layers. A fully convolutional network (FCN) is proposed in [16] based on the existing mature object classification networks (e.g., AlexNet [17], VGGNet [18], or GoogLeNet [19]). FCN modifies the fully connected layer at the end of the classification network to be a full convolutional layer and adds a deconvolution layer (upsampling operation) at the end of the classification network to achieve end-to-end and pixel-to-pixel semantic segmentation. ResNet [20] based on residual learning was proposed to have a deeper network structure. ResNet-101 with 101 layers are more commonly used in many deep learning tasks. Typically, the increase in network depth will increase the number of parameters in the network and increase the computational complexity. ResNet introduces a 1x1 convolution kernel into the residual block for dimensionality reduction. Thus, the ResNet not only guarantees the depth of the network but also reduces the computational complexity. Moreover, it is found that feature maps extracted from ResNet have a very large receptive field, which plays an important role in the classification and segmentation tasks [21]. Therefore, the ResNet is widely used in the semantic segmentation models. Zhang et al. [22] proposed a lightweight semantic segmentation model based on the ResNet, and their work achieved better accuracy and efficiency on road scene datasets. Zhang et al. [23] proposed a novel semantic segmentation framework Context Encoding Network (EncNet) based on ResNet, which improves the feature representation and achieve good performance on several datasets. Chen et al. [24] proposed the DeepLab-v2 framework based on VGGNet and ResNet to improve the performance of FCN. It is found that ResNet structure is much better than the VGGNet structure on many mainstream datasets. Therefore, in this paper, we adopt the ResNet structure as the basic semantic segmentation model.

The loss functions of the traditional semantic segmentation models are strongly dependent on ground truths. Therefore, a large dataset with accurate annotations is extremely necessary. However, the data labeling work is mainly done manually with high labeling costs, and the quality is uneven, which inevitably leads to some errors. On the contrary, unsupervised adversarial learning does not rely on ground truths. Luc et al. [25] first combined GAN-based unsupervised adversarial learning and semantic segmentation models to correct higher-order inconsistencies of semantic segmentation results. Chen et al. [26] proposed to utilize an unsupervised adversarial learning approach to adapt the semantic segmentation model of road scene across different cities.

Alternative solutions to improve the performance of segmentation models are based on fusing feature maps at different levels of abstraction [27,28] and fusing global context information [29,30]. However, these methods need to add complexity to the semantic segmentation model. Moreover, some weakly-supervised approaches can achieve satisfactory results without a large amount of annotation of the ground truths [31,32]. However, these methods are inseparable from the labeling of the image class, which may be difficult for real applications.

2.2 GAN-based unsupervised learning

GAN proposed by Goodfellow [13] contains a generative network and a discriminative network with a confrontational relationship between two networks. Typically, the generative network is used to generate samples and the discriminative network is a binary classifier that discriminates whether the input is real or generated. GAN has greatly promoted the study of unsupervised learning. Geng et al. [33] proposed an unsupervised learning framework, which adds training constraints for the depth estimation task. Fu et al. [34] proposed a virtual view-generation algorithm based on GAN to improve the performance of binocular vision systems. Moreover, some researchers used GAN to perform unsupervised adversarial transfer learning. Tzeng et al. [11] proposed an improved unsupervised domain adaptation method that combines adversarial learning with discriminative feature learning. This method can narrow the performance gap between the source domain and the target domain. Shen et al. [12] also utilized GAN-based adversarial transfer learning to improve generalization ability on the target domain. This method uses W distance to metric the difference between the two domains and improves performance on several public datasets.

Motivated by these works, we use a GAN-based unsupervised learning method to perform GAN-based unsupervised learning for semantic SLAM.

2.3 Semantic SLAM

Semantic SLAM can provide not only the geometric information but also the semantic information about the environment such as position, posture and functional attributes to complete more intelligent service tasks in complex scenes [35–37]. In fact, semantic information can be used not only to build semantic maps but also to improve the performance of vSLAM. Recently, according to the different use of semantic information, the latest semantic SLAM researches are mainly divided into the following five aspects:

172 (a) *Feature point selection*: Semantic information can be used to select feature
173 points of interest in the image. Liang et al. [38] used deep learning to extract saliency
174 maps, on which the feature points are extracted on these saliency maps. This work
175 achieved highly-accurate and robust visual odometry. Ganti et al. [39] incorporated
176 semantic segmentation network uncertainty into the feature point selection. If the
177 mutual information of a feature point above a predefined threshold, the uncertainty of
178 this feature point is considered to be small and this feature point will be easily selected.

179 (b) *Scale recovering in Monocular SLAM*: Monocular SLAM systems are mainly
180 limited by the unobservable scale of the 3D scene, which makes it difficult to obtain
181 the correct map and trajectory. Sucar et al. [40] proposed a Bayesian filtering algorithm
182 which integrates a deep learning based object detection method to estimate the scale
183 correction.

184 (c) *Long-term localization*: One of the major challenges in long-term autonomous
185 vehicle navigation is that the long-term localization tends to be harder as the map gets
186 older. Stenborg et al. [41] utilized semantically labeled 3D point map and semantically
187 segmented images to deal with the long-term localization problem rather than using
188 hand-crafted feature descriptors, which can save a lot of storage space.

189 (d) *Localization accuracy improvement*: Bowman et al. [42] first integrated
190 semantic information into the back-end optimization part and used expectation
191 maximization (EM) algorithm to achieve the joint optimization of geometric
192 information and semantic information, which improved the localization accuracy of
193 vSLAM. Lianos et al. [43] utilized semantic information as an invariant scene
194 representation to keep tracking objects. This method established a novel cost function
195 for minimizing semantic re-projection errors and geometric errors. Moreover, Wang et
196 al. [44] proved that the vSLAM system with additional semantic information can have
197 better localization accuracy. They used an object detection network called ‘You Only
198 Look Once (YOLO)’ [45] to give each feature point a semantic label. During the feature
199 matching process, only features with the same semantic label are matched.

200 (e) *Tracking in Dynamic environments*: Most vSLAM systems are based on the
201 assumption that the surrounding environment is static [46]. However, the environment
202 in real applications will inevitably contain dynamic objects, like walking people or
203 moving pets or vehicles, which may disturb feature association in the vSLAM systems
204 [47]. If there are not enough correct matching features, the pose estimation is very likely
205 to drift or even lost. Hence, it is necessary to design robust vSLAM systems in dynamic
206 environments. Some researchers have integrated deep learning into vSLAM in dynamic
207 environments. Yu et al. [3] combined SegNet [6] with a moving consistency check
208 method to filter out feature points on moving objects, and the localization accuracy is
209 highly improved than the ORB-SLAM2 system. Bescos et al. [10] proposed a novel
210 vSLAM system named DynaSLAM, which is able to detect the moving objects by
211 Mask R-CNN [7] together with a multi-view geometry method.

212 In this paper, we focus on the semantic SLAM task in dynamic environments. We
213 put more emphasizes on the pixel-level entities, which are not only used to generate a
214 dense 3D semantic map of the environment but also utilized to predict the movable
215 properties of dynamic objects, in order to filter out the corresponding feature points to

achieve a higher localization accuracy. One motivation of this paper is to incorporate a ResNet network based semantic segmentation model and a GAN-based unsupervised learning method into the ORB-SLAM2 system to form a novel framework named USS-SLAM, of which the details and the superiorities will be given in the following part.

3. The Method

3.1 Method Overview

In this section, we describe the main framework of the USS-SLAM. As shown in Fig. 1, the USS-SLAM framework is divided into two parts, i.e. the off-line training part and the on-line implementation part.

In the training process, our target is to narrow the performance gap on the feature space between the source domain and the target domain to achieve better segmentation quality. We propose an unsupervised semantic segmentation based on adversarial transfer learning to figure out the semantic information and to filter out the dynamic feature points on dynamic objects.

Fig. 1. Overview of the USS-SLAM framework.

In the implementation process, the USS-SLAM framework runs four threads in parallel, i.e. tracking thread, local mapping thread, loop closing thread, semantic map

generation thread. The flowchart is shown in Fig. 2. Raw RGB images obtained by the camera are inputs. In the tracking thread, we first extract the ORB features from the raw RGB image. Then, we segment images using the semantic segmentation model trained in the training process. After the segmentation results are generated, the ORB feature points located in dynamic objects will be filtered out. Then, a multi-view geometry method is added to further filter out dynamic feature points among the remaining feature points. In this way, feature points located in movable (a priori dynamic) objects and moving objects can be successfully filtered out. The approaches for dynamic feature filtering and multi-view geometry are detailed in section 3.4 and section 3.5, respectively. The rest of the stable feature points are used in feature matching. If tracking was successful for last frame, a transformation matrix could be computed to predict the camera pose. In the local mapping thread, we use bundle adjustment (BA) to optimize the estimated camera pose. The semantic map generation thread gets new key-frames and segmentation results derived in the tracking thread. For the key-frames, the segmentation results, transformation matrix and the depth images are used to generate the local semantic point cloud, which is further integrated into the global semantic point cloud map. The loop closing thread detects the closed loops for the last key-frame by performing pose optimization.

Fig. 2. The flowchart of the implementation process in USS-SLAM.

3.2 Unsupervised Semantic Segmentation with adversarial transfer learning

In real applications, it is difficult and impossible to label per-pixel ground truth that is a prerequisite to train a semantic segmentation model in a supervised way. Unfortunately, concerning the researches of vSLAM, there's no unsupervised learning method existing in the literature. This is the reason why we want to adopt adversarial transfer learning into the USS-SLAM framework. If the semantic segmentation model is only trained on the source domain, the domain drift may occur in unseen application scenes. Fortunately, researchers have found that it is possible and valuable to narrow the performance gap between the source domain and the target domain. Although the images in these two domains may have large difference in appearance, the distribution in the output space may share a lot of similarities. Therefore, we can adopt the idea of adversarial transfer learning to optimize the distribution in the output space.

270 Inspired by the idea of GAN, the adversarial transfer learning method proposed in
271 this paper consists of a segmentation network (i.e. generation network in GAN) and
272 two discriminator networks.

273 The network structure of the semantic segmentation model is shown in Figure 3.
274 We use Deeplab-v2 [24] with ResNet-101 [20] as the baseline network to segment
275 images. DeepLab-v2 is a widely used semantic segmentation framework with the
276 following advantages. (1) DeepLab-v2 uses Atrous convolution to effectively enlarge
277 the receptive field. Larger context information is incorporated by employing multiple
278 atrous rates. (2) Atrous Spatial Pyramid Pooling (ASPP) is used in the DeepLab-v2 to
279 segment objects at multiple scales [24]. ASPP is proposed to use atrous convolutional
280 layers with different sampling rates to extract feature maps. These feature maps are then
281 concatenated to obtain the output feature map, which contains different sizes of
282 receptive fields so that multi-scale information is encoded. ASPP is widely used in
283 many semantic segmentation models to concatenate multi-scale information for
284 prediction [24,48,49]. (3) Conditional Random Field (CRF) is integrated into the
285 DeepLab-v2 framework, which post-processes the semantic segmentation results by
286 modeling positional relationships of pixels in images [24,50,51]. (4) DeepLab-v2 used
287 a multi-scale fusion strategy. Images with different scales are fed into the model as
288 inputs, and their score maps are fused by taking the maximum response across scales.

289 Different from the original DeepLab-v2, the USS-SLAM framework abandoned
290 the use of CRF due to the extremely slow inference speed of CRF[52]. Some researches
291 proposed to down-sample the input image to speed up the inference time of CRF [53,54].
292 Teichmann et al. [52] proposed a novel method called convolutional CRFs (ConvCRFs),
293 which greatly improve the inference speed of CRF without deteriorating the
294 performance of semantic segmentation. Moreover, CRF is a post-processing process,
295 which is not suitable for joint training of the semantic segmentation model and the
296 discriminator in the USS-SLAM.

297 The USS-SLAM framework abandoned the use of a multi-scale fusion strategy.
298 Although multi-scale inputs will improve the segmentation performance, it will
299 significantly increase the computation time and memory size. Therefore it may be not
300 suitable for real-time vSLAM applications.

301 The network structure of the semantic segmentation model in the USS-SLAM
302 framework is designed in Fig. 3. The blocks are the basic modules of ResNet with
303 different parameters. It's of great significance for these block modules used in ResNet-
304 101 to reduce calculations and parameter quantities. In the USS-SLAM framework, we
305 removed the final classification layer of ResNet-101 and modified the stride of block3
306 and block4 from 2 to 1, making the size of the output feature maps 8 times smaller than
307 the input image size. Also, we use dilated convolutions with a dilation rate of 2 and 4
308 in block3 and block4 to enlarge the receptive field and to widen the range of information.

309 In order to perform adversarial transfer learning in the multi-level feature spaces,
310 we extract low-level features and high-level features simultaneously. Low-level feature
311 refers to local features of the image, e.g. edge, corner and color, and high-level feature
312 refers to the images' global representations that contain rich semantic information of
313 the image [55,56]. Recently, CNN has demonstrated superior performance for its ability

to automatically learn high-level features via a layer-to-layer propagation [57]. Typically, features extracted from the top layers of CNN are high-level features and features extracted from the intermediate layers are low-level features [56]. Thus, we can form multi-level feature spaces that are crucial for adversarial transfer learning.

In Fig. 3, ASPP is used after the block3 module, which combines dilated convolutions of four different dilation rates in the output feature space. Then we up-sample the output features along with the softmax to match the size of the input image to obtain the low-level features. Similarly, after the block3 module, we add block4 with a dilation rate of 4, ASPP and up-sample layer to obtain the high-level features.

Fig. 3. The network structure of the proposed semantic segmentation model

The structure of the discriminator network in USS-SLAM is shown in Fig. 4. We use an architecture similar to the work [58]. It is a CNN of n layers with a stride of 2. The first $n-1$ layers are comprised of a convolutional operation followed by leaky ReLU, and the last layer is a convolutional operation that outputs a feature map with the size of $H/2^n \times W/2^n$, where H and W represent the height and width of the input image. The proper number of n is discussed in the following section.

Fig. 4. The structure of discriminator network

The flowchart of our unsupervised semantic segmentation with adversarial transfer learning in the multi-level feature spaces is summarized in Fig. 5. The training details will be illustrated in the next section.

Fig. 5. Flowchart of the unsupervised semantic segmentation with adversarial transfer learning

3.3 Training Details

Before describing the training details, we first introduce the parameters definitions as shown in Table 1.

TABLE 1. The parameters involved in the training process and their definitions.

Parameters	Definitions
I_s	Source image
Y_s	The ground truths of source image I_s
I_t	Target image
L_s	The semantic segmentation outputs in the low-level feature space for the source image I_s
H_s	The semantic segmentation outputs in the high-level feature space for the source image I_s
L_t	The semantic segmentation outputs in the low-level feature space for the target image I_t
H_t	The semantic segmentation outputs in the high-level feature space for the target image I_t
h	The height of each source/target image
w	The width of each source/target image
c	The number of semantic classes
λ_{seg}^L	Weight used to balance low-level features and high-level features in training the semantic segmentation model
λ_{seg}^H	
λ_{adv_1}	Weight used to balance low-level features and high-level features in training the discriminators
λ_{adv_2}	

Given the source image set A which contains u images.

$$A = \{(I_s, Y_s)\}, s = 1, 2, \dots, u \quad (1)$$

Given the target image set B which contains v images.

$$B = \{(I_t)\}, t = 1, 2, \dots, v \quad (2)$$

The training procedure can be divided into the following four steps. It is notable that the semantic segmentation model and discriminators are trained jointly.

- (1) Randomly select a source image I_s from the source image set and input it into the semantic segmentation model to obtain the semantic segmentation outputs L_s and H_s . Up-sample L_s and H_s to the same size as the source image I_s . The segmentation loss can be computed according to the ground truths by the equation (3). Then, we can train the semantic segmentation model by back-propagating $Loss_{seg}$.

$$Loss_{seg} = - \sum_{h,w} \sum_{c \in C} (\lambda_{seg}^L Y_s^{(h,w,c)} \log(L_s^{(h,w,c)}) + \lambda_{seg}^H Y_s^{(h,w,c)} \log(H_s^{(h,w,c)})) \quad (3)$$

- (2) Randomly select a target image I_t from the target image set and input it into the

semantic segmentation model to obtain semantic segmentation outputs L_t and H_t . Up-sample L_t and H_t to the same size of the target image I_t . Then, we input L_s into the discriminator D_1 and input H_s into the discriminator D_2 . Here, D_1 is used to distinguish whether the low-level features are from the source or target domain. D_2 is used to distinguish whether the high-level features are from the source or target domain. D_1 and D_2 have the same network architecture. We can obtain predicted outputs $D_1(L_t)$ and $D_2(H_t)$. In order to fool the discriminator network, we maximize the probability that the target images are from the source image set. We use 0 to represent the source domain. Therefore, the adversarial loss can be defined as the equation (4) to back-propagate $Loss_{adv}$ to train the semantic segmentation model.

$$Loss_{adv} = -\sum_{h,w} (\lambda_{adv_1} \log(D_1(L_t)^{(h,w,0)}) + \lambda_{adv_2} \log(D_2(H_t)^{(h,w,0)})) \quad (4)$$

- (3) Input L_s and H_s to the discriminator D_1 and D_2 respectively. To make the discriminator D_1 distinguish that low-level feature L_s belongs to the source domain. We train the discriminator D_1 using the loss defined as:

$$Loss_{D_1}(L_s) = -\sum_{h,w} \log(D_1(L_s)^{(h,w,0)}) \quad (5)$$

We train the discriminator D_2 using the loss defined as:

$$Loss_{D_2}(H_s) = -\sum_{h,w} \log(D_2(H_s)^{(h,w,0)}) \quad (6)$$

Back-propagate $Loss_{D_1}(L_s)$ and $Loss_{D_2}(H_s)$ to train the discriminator D_1, D_2 .

- (4) Input L_t and H_t to the discriminator D_1 and D_2 respectively. To make the discriminator D_1 distinguish that low-level feature L_t belongs to the target domain. We use 1 to represent the target domain. We train the discriminator D_1 using the loss defined as:

$$Loss_{D_1}(L_t) = -\sum_{h,w} \log(D_1(L_t)^{(h,w,1)}) \quad (7)$$

We train the discriminator D_2 using the loss defined as:

$$Loss_{D_2}(H_t) = -\sum_{h,w} \log(D_2(H_t)^{(h,w,1)}) \quad (8)$$

Back-propagate $Loss_{D_1}(L_t)$ and $Loss_{D_2}(H_t)$ to train the discriminator D_1, D_2 .

The training procedure is summarized in Algorithm 1.

Algorithm 1: Training algorithm for USS-SLAM

- 01 for each episode iteration do:
 - 02 Turn off gradient calculation for discriminator D_1 and D_2 ;
 - 03 Randomly select a source image I_s from the source image set A ;
-

```

04   Input  $I_s$  into the semantic segmentation model to obtain  $L_s$  and  $H_s$ ;
05   Up-sample  $L_s$  and  $H_s$ ;
06   Calculate  $Loss_{seg}$  and back-propagate it to train the semantic segmentation
      model;
07   Randomly select a source image  $I_t$  from the source image set  $B$ ;
08   Input  $I_t$  into the semantic segmentation model to obtain  $L_t$  and  $H_t$ ;
09   Up-sample  $L_t$  and  $H_t$ ;
10   Compute  $Loss_{adv}$  and back-propagate it to train the semantic segmentation
      model;
11   Turn on gradient calculation for discriminator  $D_1$  and  $D_2$ ;
12   Input  $L_s$  and  $H_s$  to the discriminator  $D_1$  and  $D_2$  to obtain  $D_1(L_s)$  and  $D_2(H_s)$ ;
13   Compute  $Loss_{D_1}(L_s)$  and  $Loss_{D_2}(H_s)$ ;
14   Back-propagate  $Loss_{D_1}(L_s)$  and  $Loss_{D_2}(H_s)$  to train  $D_1$  and  $D_2$ ;
15   Input  $L_t$  and  $H_t$  to the discriminator  $D_1$  and  $D_2$  to obtain  $D_1(L_t)$  and  $D_2(H_t)$ ;
16   Compute  $Loss_{D_1}(L_t)$  and  $Loss_{D_2}(H_t)$ ;
17   Back-propagate  $Loss_{D_1}(L_t)$  and  $Loss_{D_2}(H_t)$  to train  $D_1$  and  $D_2$ ;
18   end for

```

```

377   We use the Stochastic Gradient Descent (SGD) optimizer to train the semantic
378   segmentation model. Nesterov acceleration is used to update gradient, where the
379   momentum is 0.9 and the weight decay is  $10^{-4}$ . The initial learning rate is  $2.5 \times 10^{-4}$ 
380   and is decreased using the decay with power of 0.9. Adam optimizer is used to train the
381   discriminator. Other training parameters of the discriminator are consistent with the
382   semantic segmentation model. By design of experiments, we found it is more suitable
383   to set the weights  $\lambda_{adv\_2}$  and  $\lambda_{seg}^H$  to be 0.01 and 1.0 respectively in the high-level
384   feature space. For the low-level features with less semantic information, we use smaller

```

weights in the low-level feature space. Through trial and error, we set λ_{adv_1} and λ_{seg}^L to be 0.002 and 0.1 respectively. It may be possible to derive an adaptive method to tune these parameters in the future research.

3.4 Dynamic features filtering

To filter out dynamic features we use a semantic segmentation model trained with adversarial transfer learning to segment the classes that are potentially movable. Because human activities interfere with the localization seriously in most vSLAM scenarios, we would take the human as a typical representative of dynamic objects in the rest of this paper.

The output of the semantic segmentation model, assuming that the input is an RGB image of size $m \times n \times 3$, is a matrix of size $m \times n \times l$, where l is the number of objects in the image. For each output channel $i \in l$, a binary mask is obtained. By combining all the channels into one, we can obtain the segmentation of all dynamic objects appearing in one image. After the semantic segmentation result comes out, the ORB feature points located in dynamic objects will be filtered out. By using this approach, most of the dynamic objects can be segmented and would no longer be used in tracking and mapping threads.

3.5 Multi-view Geometry

After the semantic segmentation model is well-trained, most of the dynamic feature points can be detected successfully. However, some objects are not a priori dynamic, but are moving, e.g. the chairs being dragged by a person. Therefore, we utilize the multi-view geometry method to deal with these cases [47]. The reason to use the multi-view geometry method is explained below. For instance, Fig. 6 shows the effect of dynamic feature points extracted by the multi-view geometry method in such cases. As we can see, one person is dragging a chair. Note that the chair at the bottom right of the image is not dragged. In the left image, the feature points on the moving chair are not removed at all. In the right image, all the feature points of the moving chair are removed by the multi-view geometry method.

Fig. 6. Example result of multi-view geometry method. The image on the left is the result without multi-view geometry method, and the image on the right is the result with multi-view geometry method.

The procedure of our multi-view geometry method in the USS-SLAM framework is similar to the work [47]. For the current key-frame, we will find out the reference key-frames that have the highest overlaps from the key-frame database by comparing their camera positions. As shown in Fig.7, for each feature point x , we compute the parallax angle $\alpha_{x,x'}$ between the back-projections of x and x' . Then, we can perform the multi-view geometry method by using the algorithm summarized in Algorithm 2 to obtain the dynamic feature points (dynamic $V(x')=0$ and static $V(x')=1$) in the current key-frame, where the parameter τ_a is set to 30° , and τ_z is set to 0.4m, and Δz is the difference between the true depth and the projected depth. Then, we can grow the region around the dynamic feature points as the dynamic region [59]. At last, the ORB feature points in the dynamic region of the current key-frame are considered as dynamic and filtered out.

Fig. 7. Keypoint x from the key frame is projected into the current frame by using the camera pose.

Algorithm 2: Procedure of the multi-view geometry method

```

01  for each feature point  $x$  in each reference key-frame do

02      Compute its projection  $x'$  in the current key-frame;

03      if  $\alpha_{x,x'} < \tau_a$  then

04          if  $\Delta z < \tau_z$ , then

05              set  $V(x') = 1$ ;

06          else

07              set  $V(x') = 0$ ;

08          end if

09      else

10          set  $V(x') = 0$ ;

11      end if

12  end for

13  Generate dynamic region and filter out ORB feature points in this region.

```

3.6 Pose estimation and optimization

In the above sections, we use the semantic segmentation model and multi-view geometry method to detect the dynamic feature points. Then, we match the stable feature points. Once the feature matching is successful, BA is used to optimize the estimated coordinates of 3D map points and the pose of the key-frames. Suppose a point P in the world coordinate system is projected on two adjacent image frames, and the projected non-homogeneous pixel coordinates are p_1 and p_2 . Then, we can get the relationship between point p and pixel coordinates.

$$\begin{cases} \lambda_1 p_1 = KP \\ \lambda_2 p_2 = K(RP + t) \end{cases} \quad (9)$$

where R and t represent the transformation matrix of camera's position. K represents the camera intrinsic. λ represents the scale factor. The problem can be defined to optimize the transformation matrix R and t and the point P with p_1 , p_2 and K . The optimization goal can be defined as the following expression.

$$\min \sum_{i=1}^N \left\| \lambda_1^i p_1^i - KP_i \right\|^2 + \left\| \lambda_2^i p_2^i - K(RP_i + t) \right\|^2 \quad (10)$$

where i is the number of feature points. We can use g2o framework to implement the optimization [60].

3.7 Dense semantic 3D point cloud map building

Semantic map generation thread gets new key-frames and semantic segmentation results from the tracking thread. The depth images and the key-frames' transform matrix are used to generate the key-frame's point cloud. Then, we incorporate the key-frame's point cloud into the global point cloud map according to the key-frames' poses and camera parameters. Meanwhile, semantic information is also incorporated into the point cloud map. The segmented dynamic parts are removed from the current key-frame. Each point in the map is associated with a specific color. For example, the point in blue means it belongs to a monitor, the point in red means it belongs to a chair. In this way, the 3D semantic point cloud map can be generated.

4. Experimental Results

In this section, we verify and compare experimental results based on some benchmark datasets and vSLAM methods. We first give a detailed description of the datasets used in the source domain and the target domain. Then we illustrate the performance index adopted in this experiment. After finding the optimal discriminator structure, we compare the experimental results by using the original ORB-SLAM2 system, DS-SLAM and the proposed USS-SLAM. In order to prove the superiority of the semantic segmentation model with adversarial transfer learning, we analyze the performance in terms of localization accuracy and segmentation quality. In order to cope with the situations where static objects are moving, we analyze the performance of localization accuracy by combining the multi-view geometry method.

4.1 Datasets Descriptions

To make the USS-SLAM more comprehensive, in the source domain, we adopt several datasets that are representative for semantic segmentation, i.e. Pascal Visual Object Classes (VOC) dataset, Semantic Boundaries Dataset (SBD) dataset, Microsoft Common Objects in Context (COCO) dataset.

The VOC dataset is the most popular dataset for semantic segmentation. It provides the ground truth of each pixel of each image [61]. The dataset includes 2,913 images with ground truths from 21 categories of objects.

The SBD dataset is an extended version of the Pascal VOC dataset, which provides 11,355 images with ground truths for semantic segmentation [62]. Due to the large amount of training data, this dataset is currently being widely used for semantic segmentation.

The COCO dataset is a large-scale dataset for image recognition and image segmentation [63]. The COCO dataset contains more than 80 object categories with more than 120,000 labeled images. This dataset is very common and can be used in a variety of competitions, which is important for the development of computer vision.

In the target domain, the Technical University of Munich (TUM) dataset [64] is adopted. It provides a large amount of RGB-D data and ground-truth trajectories to evaluate the vSLAM systems. The RGB-D data is collected by Microsoft Kinect sensors, which was recorded at a 30 Hz frame rate and 640×480 resolution. A high-accuracy motion-capture system with tracking cameras generates the ground-truth trajectory. The TUM dataset provides the walking sequences which could be regarded as high-dynamic scenes. The sitting sequences in TUM are also used for our experiments which contain persons just moving a little bit as slight-dynamic scenes.

Based on the above datasets, we can construct the train set and test set for USS-SLAM. Because the TUM dataset mainly includes three types of objects (people, chairs and monitors), we select 49 images from the VOC dataset, 557 images from the SBD dataset and 9,598 images from the COCO dataset that contain those three types of objects as the train set. The TUM dataset is used as the test set.

4.2 Performance Index

The TUM dataset provides the ground truth for each key-frame's pose, which can be used to generate the motion trajectory. When testing the vSLAM systems, we will get the observed pose of each key-frame to obtain the motion trajectory of the robot. By comparing the differences between these two trajectories, we can evaluate the performance of the vSLAM systems. Absolute Trajectory Error (ATE) is widely used for quantitative evaluation, which evaluates the absolute pose differences between two trajectories [3,10,64]. Indicators such as RMSE, Mean Error, Median Error and Standard Deviation (SD) are adopted to measure the ATE. Among these indicators, RMSE and SD are more important in reflecting the performance of the vSLAM system [64]. RMSE is the standard deviation of the measurement errors, which is a good reflection of the precision of the measurement. The smaller RMSE is, the closer the estimated trajectory is to the ground truth trajectory.

The calculation RMSE is as follows:

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (X_{obs,i} - X_{model,i})^2}{n}} \quad (9)$$

where n represents the number of observations, and i represents the i_{th} key-frame, and $X_{obs,i}$ represents the ground truth of the i_{th} key-frame's pose, and $X_{model,i}$ is the observed pose of the i_{th} key-frame.

SD can reflect the degree of dispersion of observations [64]. The calculation standard deviation is as follows:

$$SD = \sqrt{\frac{1}{n} \sum_{i=1}^n (X_{model,i} - \mu)^2} \quad (10)$$

where n represents the number of key-frames and $X_{model,i}$ is the observed pose of the i_{th} key-frame in our method, and μ is the mean value of $X_{model,i}$.

Mean Error (hereafter for short Mean) is the arithmetic mean of the error between the observed pose and the ground truth.

$$Mean = \frac{1}{n} \sum_{i=1}^n |X_{obs,i} - X_{model,i}| \quad (11)$$

where n represents the number of observations, and i represents the i_{th} key-frame, and $X_{obs,i}$ represents the ground truth of the i_{th} key-frame's pose, and $X_{model,i}$ is the observed pose of the i_{th} key-frame.

Median Error (hereafter for short Median) is the median value of the errors between the observed poses and the ground truths.

4.3 Different layers in discriminators

As shown in Figure 4, the discriminator is a CNN with n layers with a stride of 2. In this section, we set n as 4, 5, 6 and 7 respectively to compare the corresponding performances on the *walking_rpy* sequence. The network parameters are set to be identical. We determine the appropriate number of layers empirically in discriminator by observing the semantic segmentation results and the localization accuracy.

First, we compared the semantic segmentation results of the proposed USS-SLAM with different layers. Fig. 8 presents some example results. These images are real-time semantic segmentation results during running. The 1st column shows the original images. The 2nd, 3rd, 4th, 5th column shows the segmentation results when n is set to 4, 5, 6, 7 respectively.

In the first row image, all objects are static. The generated results using 4 layers and 5 layers are similar. When $n=5$, the wallpaper was mistakenly recognized as a part of the human body. When n is set to 6, the area of the computer on the left side becomes small. When $n=7$, the computer on the left side is not recognized and the human body is not segmented ideally. In the second row image, the moving human only appears half in the image. The generated results using 4 layers and 5 layers are similar. In the result with 6 layers, the lower part of the human body is not segmented. The generated result using 7 layers is not segmented well. The original image in the third row is extremely

blurred. Segmentation results using 4 and 7 layers are not ideal. The generated results using 5 layers and 6 layers are satisfactory. The original image in the fourth row is rotated and highly dynamic. The generated result using 4 layers and 7 layers are not segmented well. When n is set to 6, the segmented area belongs to the human body is unreasonably large. The original image in the fifth row is dynamic with two moving people. Only in the result with 6 layers, the head of the human body in the distance is successfully segmented. It is noted that the segmentation results obtained using 5 layers and 6 layers are satisfactory.

Fig. 8. Example results of semantic segmentation using USS-SLAM with different layers.

Second, we compare the ATE of different layers. Table 2 shows the results. The data used here is the average value after multiple experiments. USS-SLAM with 5 layers and 6 layers perform better than 4 layers and 7 layers.

TABLE 2. Comparison of Metrics ATE by using different layers.

Number of layers \ Performance Index	4 layers	5 layers	6 layers	7 layers
RMSE	0.0364	0.0333	0.0333	0.0368
Mean	0.0273	0.0263	0.0267	0.0300
Median	0.0214	0.0201	0.0216	0.0249
SD	0.0241	0.0204	0.0200	0.0214

According to the segmentation results and the ATE results, too many or too few network layers of the discriminator will decrease the performance. The number of

network layers of 5 or 6 layers in discriminators are suitable. Based on the above analysis, we set the number of network layers in discriminators to be 5 layers in the following experiments.

4.4 Comparisons

In this section, we compare the semantic segmentation results, 3D point cloud maps, ATEs and generated trajectories of ORB-SLAM, DS-SLAM and USS-SLAM.

First, we compared the semantic segmentation results of the proposed USS-SLAM with the DS-SLAM. Fig. 9 presents some example results. These images are real-time semantic segmentation results during running. The first column shows the original images. The second column shows the segmentation results which are saved by running the DS-SLAM system. The third column shows our semantic segmentation results by using USS-SLAM.

In the first row, all objects are static. The generated results using these two methods are not much different. As we can see, the chair is not segmented by DS-SLAM, while the segmentation quality of the chair and monitors are improved a little in USS-SLAM.

In the second row, one person is leaving the chair. In the segmentation result of DS-SLAM, the monitors have only been segmented a little part and part of the human body is mistakenly segmented into chairs. There are two different semantic categories on the human body. Compared to DS-SLAM, the segmentation qualities of chairs and monitors are improved by using USS-SLAM.

In the third row, one person is reading a book nearby, and one person who only shows his head is moving in the distance. There is no chair in the segmentation result of DS-SLAM, and a monitor close to the person is not fully segmented. Compared to DS-SLAM, it is obvious that USS-SLAM segments the monitor and chair more reasonably. However, the head in the distance is not segmented accurately in both methods. It is noted that for these small objects, the segmentation results are not well because they are easily classified into the background.

The scenes of the above three images are slightly dynamic. Below, we present two images from highly dynamic scenes, which is very difficult for traditional semantic segmentation.

The fourth image is rotated and blurred. We can see that the segmentation result by using the DS-SLAM is wrong. Semantic segmentation results contain some wrong semantic categories and only the hand part of the human body is successfully segmented. The common semantic segmentation models without special training are difficult to cope with these images. Obviously, the segmentation quality by using USS-SLAM is highly improved. First, it segments the monitor successfully. Second, the segmentation result of the person is better than those of DS-SLAM.

The fifth image is highly blurred. It is expected that the segmentation result of DS-SLAM is bad, which only segments a small part of the human body and divides the wallpaper on the left side of the image into a monitor wrongly. These fast-moving scenes limit the semantic segmentation performance of DS-SLAM. In USS-SLAM, benefited from GAN-based unsupervised adversarial transfer learning, the segmentation quality is better. The USS-SLAM segments people with high quality.

However, USS-SLAM is still unable to segment part of the monitor, which is so small that is easily classified into background class.

Fig. 9. Example results of semantic segmentation using DS-SLAM and USS-SLAM for dynamic TUM sequences.

Second, we compared the 3D point cloud map generated by different methods. It is noted that ORB-SLAM2 is a common vSLAM system that does not incorporate semantic information, while DS-SLAM and USS-SLAM incorporate semantic information. Fig. 10(a) shows the result of the ORB-SLAM2. In ORB-SLAM2 system, there is not a semantic segmentation model and it is clearly shown that the dynamic feature points are not removed. Therefore, the dynamic objects are kept in the map. Fig. 10(b) shows the result of the DS-SLAM. Note that we only fuse the segmented object categories that appear in the scene to the 3D semantic point cloud map because DS-SLAM is easy to be interference from unrelated semantic information. Compared with ORB-SLAM2, most dynamic portions of the scene have been filtered in DS-SLAM.

However, at the bottom of this image, some of the feature points belonging to people are not well removed. Compared with DS-SLAM, the result of USS-SLAM contains less feature points belonging to those people, which is shown in Fig. 10(c).

Fig. 10. 3D point cloud map (a) using ORB-SLAM2; (b) using DS-SLAM; (c) using the proposed USS-SLAM.

Third, we compare the ATE index of USS-SLAM against those of ORB-SLAM2, DS-SLAM. Table 3 shows the results using our method performs favorably against the other methods. The data used here is the average value after multiple experiments. The ORB-SLAM2 system does not filter out any dynamic feature points, which easily cause tracking failures. The proposed USS-SLAM performs favorably against the DS-SLAM system in most sequences, especially in the high-dynamic sequence: w_rpy. However, in low-dynamic sequence: s_static, the performance is not improved much. The reason is that the semantic segmentation network can handle the low-dynamic scenes easily and achieve good performance, left little space for our method to improve.

TABLE 3. Comparison of Metrics Absolute Trajectory Error (ATE) of using different algorithms.

Sequences Methods		w_xyz	w_static	w_rpy	w_half	s_static
ORB-SLAM2	RMSE	0.7521	0.3900	0.8705	0.4863	0.0087
	Mean	0.6492	0.3554	0.7425	0.4272	0.0076
	Median	0.5857	0.3087	0.7059	0.3964	0.0066
	SD	0.3759	0.1602	0.4520	0.2290	0.0043
DS-SLAM	RMSE	0.0247	0.0081	0.4442	0.0303	0.0065
	Mean	0.0186	0.0073	0.3768	0.0258	0.0055
	Median	0.0151	0.0067	0.2835	0.0222	0.0055
	SD	0.0161	0.0036	0.2350	0.0159	0.0033
USS-SLAM	RMSE	0.0139	0.0062	0.0333	0.0253	0.0063
	Mean	0.0120	0.0055	0.0263	0.0217	0.0055
	Median	0.0105	0.0050	0.0201	0.0191	0.0050
	SD	0.0070	0.0029	0.0204	0.0130	0.0030

In the next part, we present the estimated trajectories of our method and the trajectories' ground truths, which is shown in Fig. 11. After running the vSLAM system, we can obtain the pose of each key-frame, and get the trajectory of the camera. The smaller the difference between the obtained trajectory and the ground truth trajectory is, the higher the localization accuracy of the vSLAM system is. As we can see, the

trajectories estimated by USS-SLAM are the closest to the ground truths in these dynamic sequences.

Fig. 11. Comparison of trajectories. First row uses w_{xyz} . Second row uses w_{static} . Third row uses w_{rpy} . Fourth row uses w_{half} . Fifth row uses s_{static} .

By analyzing the above results, we can get the following conclusions:

Remark 1: traditional vSLAM such as ORB-SLAM2, performs poorly in dynamic environments. This is because the feature points on the dynamic objects are very unstable, and matching these feature points produces a large number of erroneously estimated poses. In USS-SLAM, benefited from the semantic information derived, it is able to filter out the feature points on the dynamic objects, and the localization accuracy is significantly improved.

Remark 2: compared with USS-SLAM, DS-SLAM is susceptible to interference from some unrelated semantic information. DS-SLAM often wrongly recognizes some objects, even contains several different semantic categories on one object. The USS-SLAM is rarely interfered by unrelated semantic information because in the USS-SLAM the types of objects that can be recognized are fewer. Therefore, we believe that training a semantic segmentation model for a specific real application environment in the semantic SLAM task can achieve better results, rather than directly using a pre-trained semantic segmentation model.

Remark 3: the power of adversarial transfer learning is effectively performed in USS-SLAM, resulting in more accurate segmentation results than those of DS-SLAM.

4.5 With or without adversarial transfer learning

Semantic segmentation models are often trained by a fully supervised learning method. However, the trained model may not generalize well to real application scenes. Therefore, as we mentioned above, adversarial transfer learning (ATL for short) is critical to narrow the performance gap without labeled per-pixel ground truths. In this section, we want to prove that a semantic segmentation with ATL is more superior to those without ATL. The same idea can be extended to other SLAM systems and is not limited in USS-SLAM. It is noted that the number of training epochs and training parameters used in the semantic segmentation model are the same.

The comparison results are summarized in Fig. 12. These semantic segmentation results are saved during the vSLAM is running. The first column shows the original images. The second column shows the segmentation results without ATL. The third column shows the results of the semantic segmentation with ATL.

In the first row, one person is moving. With ATL, it is obvious that the segmentation results of both monitors and people are more reasonable and the object boundaries are more coordinated.

In the second row, one person is moving on the left side. Without ATL, it is found that the chair is not segmented well. Also, there exist two monitors in this image, and one monitor is not segmented well. As we can see, in the segmentation result of USS-SLAM with ATL, the chair is segmented more completely and the monitors on the right side are partially segmented.

In the third row, one person is sitting on a chair. Results show that the chair and the monitors are segmented better with ATL, compared with those with traditional supervised learning. Without ATL, the pixels that should belong to the sitting person are classified into backgrounds and some pixels in the top right corner of the image are wrongly classified as backgrounds as well.

In the fourth row, the segmentation results of the person and the chair are similar

with or without ATL. However, the monitor is segmented more reasonably with ATL.

In the fifth row, one person is sitting on a chair and the other person is moving fast. As we can see, the chairs and the monitor are segmented slightly better with ATL. The moving person is not segmented well in the result without ATL. Although the person is segmented slightly more reasonably in the result of USS-SLAM with ATL, the lower part of the body is still not well segmented. Our USS-SLAM with ATL still needs further improvement on these fast-moving objects.

Fig. 12. Example results of semantic segmentation for dynamic TUM sequences.

Furthermore, we compare the localization accuracy with or without ATL. Table 4

shows the comparison results of experiments and Table 5 shows the improvements. The data used here is the average value after multiple experiments. According to Table 4 and Table 5, USS-SLAM with ATL achieves better performance in all sequences. It proves that adapting the semantic segmentation model with ATL in the real application scenes can indeed improve the localization accuracy to some extent. It is noted that in high-dynamic sequences, the performance can be improved by 7%-25%. These sequences contain many rotating blurred images, which increases the difficulty of segmentation and dynamic feature extraction. By using adversarial transfer learning, the semantic segmentation quality and the localization accuracy can be improved. However, in low-dynamic sequence: s_static, the improvements are not as obvious as in the high-dynamic sequences.

Based on above results, we may get the following conclusions:

Remark 4: compared to USS-SLAM without ATL, USS-SLAM with ATL has some improvement in semantic segmentation results and localization accuracy. It shows that after trained with GAN-based unsupervised adversarial transfer learning, the performance gap between the source domain (training dataset) and the target domain (real application) is indeed narrowed.

Remark 5: USS-SLAM does not recognize some small objects such as monitors and chairs that are partially blocked by people. These small objects are easily recognized as backgrounds. It shows that the USS-SLAM has limited ability to segment these small objects.

Remark 6: USS-SLAM with a semantic segmentation network can achieve satisfying results in low-dynamic scenes even without ATL. USS-SLAM with ATL and without ATL have similar performance.

TABLE 4. Comparison of ATE with or without ATL in USS-SLAM.

Sequences		w_xyz	w_static	w_rpy	w_half	s_static
Methods						
USS-SLAM with ATL	RMSE	0.0139	0.0063	0.0333	0.0253	0.0063
	Mean	0.0120	0.0055	0.0263	0.0217	0.0055
	Median	0.0105	0.0051	0.0201	0.0191	0.0050
	SD	0.0070	0.0030	0.0204	0.0130	0.0030
USS-SLAM without ATL	RMSE	0.0159	0.0075	0.0392	0.0289	0.0067
	Mean	0.0138	0.0063	0.0311	0.0250	0.0058
	Median	0.0122	0.0056	0.0239	0.0207	0.0053
	SD	0.0079	0.0040	0.0238	0.0146	0.0033

TABLE 5. The Improvement of index performance with or without ATL.

Sequences	w_xyz	w_static	w_rpy	w_half	s_static
Index					
RMSE	12.58%	16.00%	15.05%	12.46%	5.97%
Mean	13.04%	12.70%	15.43%	13.20%	5.17%
Median	13.93%	8.93%	15.90%	7.73%	5.66%
SD	11.39%	25.00%	14.29%	10.96%	9.09%

4.6 With or without multi-view geometry

We further evaluated the effect of USS-SLAM with or without multi-view geometry on five sequences. The results are shown in Table 6 and Table 7.

By analyzing the results, we can get the following conclusions:

Remark 7: compared the results of ORB-SLAM2 in Table 3, the performance of USS-SLAM no matter with or without multi-view geometry is significantly improved. It means that the semantic information can indeed improve the localization accuracy in vSLAM applications.

Remark 8: It is found that the USS-SLAM with a multi-view geometry method is better in all sequences. However, accuracy is not improved so much. The reason is easy to explain, i.e. because most of the dynamic feature points have been successfully detected and filtered out by using the semantic segmentation model and there is not much space for further improvement. We may have a conclusion that multi-view geometry may not be a prerequisite and the framework of USS-SLAM can be simplified, but to verify this we need to do more comprehensive experiments.

TABLE 6. Comparison of index performance with or without Multi-view Geometry in USS-SLAM.

Sequences		w_xyz	w_static	w_rpy	w_half	s_static
Methods						
USS-SLAM Without Multi- view Geometry	RMSE	0.0151	0.0065	0.0358	0.0277	0.0065
	Mean	0.0131	0.0058	0.0281	0.0235	0.0056
	Median	0.0115	0.0054	0.0222	0.0204	0.0050
	SD	0.0076	0.0031	0.0222	0.0142	0.0031
USS-SLAM With Multi- view Geometry	RMSE	0.0139	0.0062	0.0333	0.0253	0.0063
	Mean	0.0120	0.0055	0.0263	0.0217	0.0055
	Median	0.0105	0.0050	0.0201	0.0191	0.0050
	SD	0.0070	0.0029	0.0204	0.0130	0.0030

TABLE 7. The Improvement of performance in USS-SLAM by using Multi-view Geometry.

Sequences		w_xyz	w_static	w_rpy	w_half	s_static
Index						
RMSE		7.95%	4.62%	6.98%	8.66%	3.07%
Mean		8.40%	5.17%	6.41%	7.66%	1.79%
Median		8.70%	7.41%	9.46%	6.37%	0.00%
SD		7.89%	6.45%	8.11%	8.45%	3.24%

4.7 Cost Analysis

To verify that the proposed method is suitable for real time applications, this section calculates the cost time of some major modules in USS-SLAM. The results are summarized in Table 8.

TABLE 8. Cost Time of the USS-SLAM Modules.

Module	ORB feature extraction	Semantic segmentation	Multi-view Geometry
Time(s)	0.0098	0.3067	0.3836

All the experiments are carried out on a single computer with Intel i7 CPU, GTX1080 GPU, and 32GB memory. In the experiment, the input image size is 640×480. For USS-SLAM without multi-view geometry, the average time to process each frame is 0.3224s, which can run at 3.1Hz. For USS-SLAM with multi-view geometry, the average time to process each frame is 0.6878s. It can run at 1.45Hz, which is sufficient for most online indoor applications. It is notable that the cost of required resources is not so high, and we can conclude that the USS-SLAM can be performed on mainstream embedded platforms.

5. Conclusion

The fusion of semantic information and geometric information in vSLAM is an emerging research field, which may achieve good performance about localization accuracy and semantic information in a 3D semantic map. This paper adopts the idea of adversarial transfer learning to improve the semantic segmentation network, and we proposed an unsupervised semantic model with a ResNet structure in the multi-level feature spaces. This work may be the first attempt to perform multi-level output space adversarial transfer learning for the semantic slam task in a dynamic environment. The effectiveness of our method is tested on the five challenging dynamic sequences from the TUM dataset. The experimental results show that our method performs favorably against DS-SLAM and ORB-SLAM2 on the index of RMSE, Mean, Median and SD. The proposed USS-SLAM has a unique value for real applications of robots and autonomous vehicles in the dynamic environment.

Future research directions include the following points. First, USS-SLAM only considers the common categories such as people, chairs and monitors in the TUM dataset. Other types of objects are treated as irrelevant semantic information. However, the USS-SLAM framework could be extended to more extensive object categories in the future. Second, the real-time performance of USS-SLAM can be further improved by optimizing the semantic segmentation model and the multi-view geometry method. Third, the generated dense semantic point cloud map should be applied for the robot to complete advanced tasks, e.g. intelligent navigation, human interaction, etc. Fourth, we have not considered the effects of noise. Some image denoising and filtering algorithms would be adopted for the USS-SLAM framework to improve the performance.

Acknowledgement

This research is supported by the national natural science foundation of China (61673288)

REFERENCES

- [1] R. Mur-Artal, J.D. Tardos, ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras, IEEE Trans. Robot. 33 (2017) 1255–1262.

- [2] J. Engel, T. Schops, and D. Cremers, LSD-SLAM: Large-Scale Direct Monocular SLAM, in: ECCV, Springer, 2014, pp. 834-849.
- [3] C. Yu, Z. Liu, X. Liu, F. Xie, Y. Yang, Q. Wei, Q. Fei, DS-SLAM: A Semantic Visual SLAM towards Dynamic Environments., Robotics, in: IROS, IEEE, 2018, pp. 1168-1174.
- [4] J. Civera, D. Galvez-Lopez, L. Riazuelo, J.D. Tardos, J.M.M. Montiel, Towards semantic SLAM using a monocular camera, Proc. 2011 IEEE/RSJ Int Conf Intell Robot Syst, 2011, pp. 1277-1284.
- [5] C. Cadena, L. Carlone, H. Carrillo, et al. Past, present, and future of simultaneous localization and mapping: towards the robust-perception age, IEEE Trans Robot. 32 (6) (2016) 1309-1332.
- [6] V. Badrinarayanan, A. Kendall, R. Cipolla, Segnet: A deep convolutional encoder-decoder architecture for image segmentation, IEEE Trans Pattern Anal Mach Intell. 39 (12) (2017) 2481-2495.
- [7] K. He, G. Georgia, D. Piotr, G. Ross, Mask R-CNN, in: ICCV, Vol. 1, IEEE, 2017, pp. 2980-2988.
- [8] R.F. Salas-Moreno, R.A. Newcombe, H. Strasdat, P.H.J. Kelly, A.J. Davison, SLAM++: Simultaneous localisation and mapping at the level of objects, in: CVPR, IEEE, 2013, pp. 1352-1359.
- [9] S. Vasudevan, S. Gächter, V. Nguyen, R. Siegwart, Cognitive maps for mobile robots-an object based approach, Rob Auton Syst. 55 (2007) 359-371.
- [10] B. Bescos, J.M. Fàcil, J. Civera, J. Neira, DynaSLAM: Tracking, Mapping and Inpainting in Dynamic Scenes, IEEE Robotics and Automation Letters. 3 (2018) 4076-4083.
- [11] E. Tzeng, J. Hoffman, K. Saenko, T. Darrell, Adversarial Discriminative Domain Adaptation, in: CVPR, Vol. 1, IEEE, 2017, pp. 2962-2971.
- [12] J. Shen, Y. Qu, W. Zhang, Y. Yu, Wasserstein Distance Guided Representation Learning for Domain Adaptation, in: Conference on Artificial Intelligence (AAAI), 2018.
- [13] I.J. Goodfellow, J. Pouget-abadie, M. Mirza, B. Xu, D. Warde-farley, et al. Generative Adversarial Nets, in: Conference on Neural Information Processing Systems (NIPS), 2014, pp. 2672-2680.
- [14] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, C. Liu, A survey on deep transfer learning, in: International Conference on Artificial Neural Networks (ICANN), 2018, pp. 270-279.
- [15] A. Garcia-garcia, S. Orts-escolano, S. Oprea, V. Villena-martinez, P. Martinez-gonzalez, J. Garcia-rodriguez, A survey on deep learning techniques for image and video semantic segmentation, Appl Soft Comput J. 70 (2018) 41-65.
- [16] E. Shelhamer, J. Long, T. Darrell, Fully Convolutional Networks for Semantic Segmentation, IEEE Trans Pattern Anal Mach Intell. 39 (2017) 640-651.
- [17] A. Krizhevsky, G.E. Hinton, ImageNet Classification with Deep Convolutional Neural Networks, in: Conference on Neural Information Processing Systems (NIPS), 2012, pp. 1-9.
- [18] K. Simonyan, A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, in: ICLR, 2015, pp. 1-14.
- [19] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going Deeper with Convolutions, in: CVPR, IEEE, 2015, pp. 1-9.
- [20] K. He, X. Zhang, S. Ren, J. Sun, Deep Residual Learning for Image Recognition, in: CVPR, Vol. 1, IEEE, 2016, pp. 770-778.
- [21] C. Peng, X. Zhang, G. Yu, G. Luo, J. Sun, Large Kernel Matters—Improve Semantic Segmentation by Global Convolutional Network, in: CVPR, Vol. 1, IEEE, 2017, pp. 1743-1751.
- [22] Z. Chen, Q.M.J. Wu, S. Member, L. Cai, D. Lu, X. Li, Fast Semantic Segmentation for Scene Perception, IEEE Trans Ind Informatics. 15 (2019) 1183-1192.
- [23] H. Zhang, K. Dana, J. Shi, Z. Zhang, X. Wang, A. Tyagi, A. Agrawal, Context Encoding for Semantic Segmentation, in: CVPR, IEEE, 2018, pp. 7151-7160.

- [24] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A.L. Yuille, DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs, *IEEE Trans Pattern Anal Mach Intell.* 40 (4) (2018) 834-848.
- [25] P. Luc, C. Couprie, J. Verbeek, L.J. Kuntzmann, Semantic Segmentation using Adversarial Networks, in: *Conference on Neural Information Processing Systems (NIPS)*, 2016.
- [26] Y. Chen, W. Chen, Y. Chen, B. Tsai, Y.F. Wang, M. Sun, No More Discrimination : Cross City Adaptation of Road Scene Segmenters, in: *ICCV, IEEE*, 2017, pp. 1992-2001.
- [27] G. Huang, Z. Liu, L. Van Der Maaten, K.Q. Weinberger, Densely Connected Convolutional Networks, in: *CVPR, IEEE*, 2017, pp. 4700-4708.
- [28] J. Simon, M. Drozdal, D. Vazquez, A. Romero, Y. Bengio, The One Hundred Layers Tiramisu : Fully Convolutional DenseNets for Semantic Segmentation, in: *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, IEEE, 2017, pp. 1175-1183.
- [29] H. Li, P. Xiong, J. An, L. Wang, Pyramid Attention Network for Semantic Segmentation, *arXiv preprint arXiv: 1805.10180*.
- [30] W. Yang, Q. Zhou, Y. Fan, G. Gao, S. Wu, W. Ou, H. Lu, J. Cheng, L.J. Latecki, Deep Context Convolutional Neural Networks for Semantic Segmentation, in: *Chinese Conference on Computer Vision (CCCV)*, Springer, 2017, pp. 696-704.
- [31] A. Khoreva, R. Benenson, J. Hosang, M. Hein, B. Schiele, Simple Does It: Weakly Supervised Instance and Semantic Segmentation, in: *CVPR, Vol. 1, IEEE*, 2017, pp. 1665-1674.
- [32] Z. Huang, X. Wang, J. Wang, W. Liu, J. Wang, Weakly-Supervised Semantic Segmentation Network With Deep Seeded Region Growing, in: *CVPR, IEEE*, 2018, pp. 7014-7023.
- [33] M. Geng, S. Shang, B. Ding, H. Wang, Unsupervised Learning-based Depth Estimation aided Visual SLAM Approach, *arXiv preprint arXiv: 1901.07288*.
- [34] J. Fu, J. Liang, Virtual View Generation Based on 3D-Dense-Attentive GAN Networks, *Sensors.* 19 (2) (2019) 344.
- [35] R. F. Salas-Moreno, Dense semantic SLAM, London, UK: Imperial College, 2014.
- [36] I. Kostavelis, A. Gasteratos, Semantic mapping for mobile robotics tasks: A survey, *Rob Auton Syst.* 66 (2015) 86-103.
- [37] M. Tenorth, L. Kunze, D. Jain, M. Beetz, KNOWROB-MAP – Knowledge-Linked Semantic Object Maps, in: *International Conference on Humanoid Robots*, 2010, pp. 430-435.
- [38] H. Liang, N.J. Sanket, C. Ferm, Y. Aloimonos, SalientDSO : Bringing Attention to Direct Sparse Odometry, *IEEE Transactions on Automation Science and Engineering.* (2019) 1-8.
- [39] P. Ganti, S. Waslander, Network Uncertainty Informed Semantic Feature Selection for Visual SLAM, in: *Conference on Computer and Robot Vision (CRV)*, 2019, pp. 121-128.
- [40] E. Sucar, J. Hayet, Bayesian Scale Estimation for Monocular SLAM Based on Generic Object Detection for Correcting Scale Drift, in: *ICRA, IEEE*, 2018, pp. 5152-5158.
- [41] E. Stenborg, C. Toft, L. Hammarstrand, Long-term Visual Localization using Semantically Segmented Images, in: *ICRA, IEEE*, 2018, pp. 6484-6490.
- [42] S.L. Bowman, G.J. Pappas, Probabilistic Data Association for Semantic SLAM, in: *ICRA, IEEE*, 2017, pp. 1722-1729.
- [43] K. Lianos, J.L. Schonberger, M. Pollefeys, T. Sattler, VSO : Visual Semantic Odometry, in: *ECCV, Springer*, 2018, pp. 246-263.
- [44] Y. Wang, A. Zell, Improving Feature-based Visual SLAM by Semantics, in: *International Conference on Image Processing, Applications and Systems (IPAS)*, IEEE, 2018, pp. 7-12.
- [45] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You Only Look Once: Unified, Real-Time Object Detection, in: *CVPR, IEEE*, 2016, pp. 779-788.
- [46] M.R.U. Saputra, A. Markham, N. Trigoni, Visual SLAM and Structure from Motion in Dynamic Environments : A Survey, 51 (2018).

- [47] W. Tan, H. Liu, Z. Dong, G. Zhang, H. Bao, Robust Monocular SLAM in Dynamic Environments, Proc IEEE International Symposium on Mixed and Augmented Reality, 2013, pp. 209-218.
- [48] L. Chen, G. Papandreou, F. Schroff, H. Adam, Rethinking Atrous Convolution for Semantic Image Segmentation, in: CVPR, IEEE, 2017.
- [49] M. Yang, K. Yu, C. Zhang, Z. Li, K. Yang, DenseASPP for Semantic Segmentation in Street Scenes, in: CVPR, 2018, pp. 3684-3692.
- [50] B. Romera-paredes, M. Larsson, A. Kirillov, B. Savchynskyy, C. Rother, F. Kahl, P.H.S. Torr, Conditional Random Fields Meet Deep Neural Networks for Semantic Segmentation: Combining Probabilistic Graphical Models with Deep Learning for Structured Prediction, IEEE Signal Processing Magazine. 35 (1) (2018) 37-52.
- [51] W. Zhao, Y. Fu, X. Wei, H. Wang, An Improved Image Semantic Segmentation Method Based on Superpixels and Conditional Random Fields, Appl. Sci. 8(5) (2018) 837.
- [52] M.T.T. Teichmann, R. Cipolla, Convolutional CRFs for Semantic Segmentation, arXiv preprint arXiv: 1805.04777.
- [53] S. Chandra, I. Kokkinos, Fast, exact and multi-scale inference for semantic image segmentation with deep gaussian CRFs, in: ECCV, Springer, 2016, pp. 402-418.
- [54] G. Lin, C. Shen, A. Van Den Hengel, I. Reid, Efficient Piecewise Training of Deep Structured Models for Semantic Segmentation, in: CVPR, IEEE, 2016.
- [55] F. Takarli, A. Aghagolzadeh, H. Seyedarabi, Combination of high-level features with low-level features for detection of pedestrian, Signal, Image Video Process. 10 (2016) 93-101.
- [56] L. Zheng, Y. Zhao, S. Wang, J. Wang, Q. Tian, Good Practice in CNN Feature Transfer, arXiv preprint arXiv: 1604.00133.
- [57] H. Li, H. Lu, Z. Lin, X. Shen, B. Price, LCNN: Low-level Feature Embedded CNN for Salient Object Detection, arXiv preprint arXiv: 1508.03928.
- [58] A. Radford, L. Metz, S. Chintala, Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks, in: ICLR, 2016.
- [59] N.L. Gerlach, G.J. Meijer, D. Kroon, E.M. Bronkhorst, S.J. Bergé, et al., Evaluation of the potential of automatic segmentation of the mandibular canal using cone-beam computed tomography, British Journal of Oral and Maxillofacial Surgery. 52(9) (2014) 838-844.
- [60] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, W. Burgard, G2o: A general framework for graph optimization, in: IEEE international conference on robotics and automation (ICRA), IEEE, 2011, pp. 3607-3613.
- [61] M. Everingham, L. Van Gool, C.K.I. Williams, J. Winn, The PASCAL Visual Object Classes (VOC) Challenge, International Journal of Computer Vision. 88 (2010) 303-338.
- [62] B. Hariharan, P. Arbel, L. Bourdev, S. Maji, J. Malik, U.C. Berkeley, A. Systems, P. Ave, S. Jose, Semantic Contours from Inverse Detectors, in: ICCV, Vol. 1, IEEE, 2011, pp. 991-998.
- [63] T. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Doll, C.L. Zitnick, Microsoft COCO : Common Objects in Context, in: ECCV, Springer, 2014, pp. 740-755.
- [64] J. Sturm, N. Engelhard, F. Endres, W. Burgard, D. Cremers, A benchmark for the evaluation of RGB-D SLAM systems, in: IROS, IEEE, 2012, pp. 573-580.