Real-time RGB-D semantic keyframe SLAM based on image segmentation learning from industrial CAD models

Howard Mahe, Denis Marraud, Andrew I. Comport

Real-time RGB-D semantic keyframe SLAM based on image segmentation learning from industrial CAD models

Howard Mahé[12], Denis Marraud[1] and Andrew I. Comport[2]

***Abstract*— This paper presents methods for performing *real-time semantic SLAM* aimed at autonomous navigation and control of a humanoid robot in a manufacturing scenario. A novel multi-keyframe approach is proposed that simultaneously minimizes a semantic cost based on class-level features in addition to common photometric and geometric costs. The approach is shown to robustly construct a 3D map with associated class labels relevant to robotic tasks. Alternatively to existing approaches, the segmentation of these semantic classes have been learnt using RGB-D sensor data aligned with an industrial CAD manufacturing model to obtain noisy pixel-wise labels. This dataset confronts the proposed approach in a complicated real-world setting and provides insight into the practical use case scenarios. The semantic segmentation network was fine tuned for the given use case and was trained in a semi-supervised manner using noisy labels. The developed software is real-time and integrated with ROS to obtain a complete semantic reconstruction for the control and navigation of the HRP4 robot. Experiments in-situ at the Airbus manufacturing site in Saint-Nazaire validate the proposed approach.**

I. INTRODUCTION

Recent advances in closed-loop control using visual SLAM have allowed humanoid robots to navigate and locate themselves with centimetric precision in 6D using real-time RGB-D sensing of their environment (see for example [1]). Navigation and planning algorithms, however, require higher-level knowledge about the environment surrounding the robot in order to know which parts of the environment are accessible, which objects can be interacted with and where obstacles are located. State-of-the-art real-time semantic SLAM systems have primarily been applied to generic object segmentation classes without any real connection to robotic task objectives. Whilst some domain specific cases such as autonomous driving exist, useful high-level information is not always readily identifiable in a general setting. A manufacturing environment, however, offers detailed CAD models that provide labels in minute detail for every part of the environment.

The work presented in this paper is directly related to the European H2020 project Comanoid. This project aims at collaborative manufacturing of aircraft using humanoids in a multi-contact control setting. Within this context detailed labels are required, not only for walk planning, but also for stair climbing and the identification of support regions for the hands. The Airbus manufacturing CAD model ideally provides detailed information about the aircraft mock-up and assembly line.

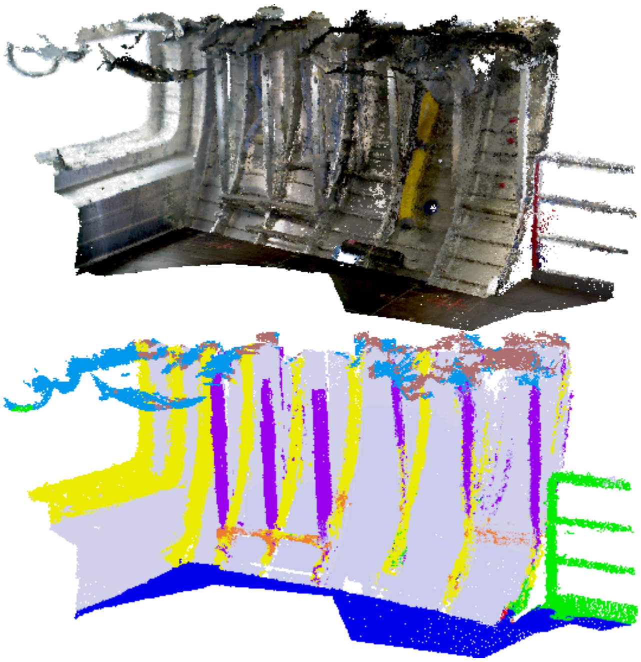


Fig. 1: The output of the proposed system. On top, a dense 3D map of an Airbus A350 mock-up, reconstructed in real-time using a RGB-D sensor. On bottom, the corresponding semantic 3D map is shown.

II. RELATED WORK

A. Real-time semantic segmentation

Since the introduction of fully convolutional networks (FCN) [2], almost all methods have adopted the fully convolutional encoder-decoder paradigm. The encoder gradually reduces the feature map and extracts higher-level perceptual information to yield a feature map downsampled with respect to the input RGB image resolution. The encoded features are successively refined in the decoder which reconstructs sharper object boundaries. The final layer generates a segmentation mask using a softmax function. High performance CNN-based models for semantic segmentation require a large amount of computational resources and generally suffer from a larger number of parameters and floating point operations [3]. These issues make them unsuitable for applications requiring real-time processing. To alleviate those issues, recent efforts have been made to design light-weight and efficient networks which still deliver high accuracy.

ICNet [4] adapts PSPNet [3] to deal with high-resolution images (2048×1024) using an image cascade network in addition to a compression technique. Likewise, LW-

* This work was supported by the H2020 Comanoid project

[1] Airbus Defence & Space, Elancourt, France
[2] CNRS-I3S, Université Côte d'Azur, Sophia-Antipolis, France

TABLE I: Performance on the Cityscapes [12] test set in terms of mean intersection over union (mIoU) defined in Section IV-C. Computational cost is measured for 224×224 inputs in floating point operations per second (FLOPs) and inference speed in terms of frames processed per second (FPS) for 1024×512 inputs on a NVIDIA Titan X, apart from ICNet which uses 2048×1024 inputs.

Model	FLOPS	Speed (FPS)	mIoU (%)
ENet [8]	345 M	88	58.3
ESPNet [9]	424 M	112	60.3
ESPNetv2 [10]	332 M	83	62.1
ICNet [4]	-	30	69.5
ERFNet [11]	2.45 B	48	**69.7**

RefineNet [5] adapts RefineNet by removing the Residual Convolutional Unit (RCU) blocks and replacing all 3×3 convolutions by point-wise (1×1) convolutions in the decoder. LW-RefineNet architecture can be mixed with any backbone network and the authors report a slight improvement (+0.5% on PASCAL VOC 12) using their decoder over DeepLabv3 [6] with MobileNetv2 [7] backbone. Alternatively, several approaches have been proposed to create light-weight CNNs by design which relies on factoring computationally expensive convolution operation so that the underlying model learns fewer parameters and has fewer floating point operations. ENet [8], ESPNet [9], [10] and ERFNet [11] all adopt the encoder-decoder paradigm and build their own bottleneck module. ENet [8] and ERFNet [11] both use asymmetric 1D convolutions and dilated convolutions, while ESPNet [9] introduces a bottleneck module (ESP) in which standard convolution relies on point-wise (1×1) convolution and spatial pyramid of dilated convolutions. ESPNetv2 [10] extends the latter module by using group point-wise and depth-wise separable convolutions. It is worth noting that the ESPNetv2 backbone outperforms MobileNetv2 [7] in terms of accuracy and run-time speed for image classification although it is hazardous to transpose such assertion for semantic segmentation since generalisation capabilities of the model must be evaluated in the case of such transfer learning.

Table I summarizes the performance of various real-time semantic segmentation models. Since LW-RefineNet has not been benchmarked on the Cityscapes dataset [12], ERFNet was found to be the best accuracy-speed trade-off.

B. Semantic SLAM

Semantic SLAM approaches can be divided into semantic visual odometry (VO) approaches focusing on exploiting semantic label maps for pose estimation and those approaches that focus on incremental 3D semantic mapping.

Classic VO methods generally perform image alignment using photometric and geometric error terms over coarse-to-fine levels of a pyramid. The authors' initial work on semantic visual odometry [13] focused on studying the importance of semantic-only information in pose estimation but the approach was limited to a single keyframe and did not run in real-time. Czarnowski et al. [14] proposes to replace the usual image pyramid by a hierarchy of convolutional feature maps from a CNN trained for image classification. This work demonstrates rotational robustness to varying lighting conditions by aligning thousands of features maps and real-time performance is achieved thanks to GPU acceleration. Recently, several methods [15], [16], [17], [18], [19] have learnt to solve the non-linear least squares optimization in a differentiable manner so that the network can learn suitable features, subsequently making the minimization problem more tractable.

In terms of *3D semantic mapping* two types of methods have emerged: (a) offline 3D semantic labeling of volumetric reconstructions which use either voxel grids [20], octrees [21], pointclouds [22] or meshes [23], [24] as an underlying 3D map representation and (b) *incremental* 3D semantic mapping approaches which infer semantic segmentation of images at multiple viewpoints and incrementally fuse the semantic information into a 3D map representation such as voxel grids [25], octrees [26], surfels map [27], [28], pointclouds [29] or multi-view [30], [31]. Most methods adopt Bayesian fusion [32] for semantic label fusion [25], [26], [28], [29], [31] except [27] which uses a confidence-based fusion scheme.

Despite the implicit ability to model self occlusions from different parts of the scene, volumetric representations consume a lot of memory and have not been designed to perform loop-closure in a SLAM setting [33] because the camera positions used to acquired the map are completely discarded. Surfel-based map representations [34] solve most of the issues encountered with volumetric representations at the cost of a complex non-rigid map deformation algorithm for handling loop closure. Graph of keyframes representations have been popular in robotics due to their locally accurate representations and their capability to handle incremental drift by adjusting efficiently the graph using loop closure. Contrary to volumetric approaches, keyframes encode raw sensor data and uncertainty that can be accessed with constant time access independently to the number of keyframes in the graph. One disadvantage of keyframe approaches is that the current camera field of view does not completely overlap with the closest keyframe and the image alignment is therefore only computed using partial overlap. In [35] a multi-keyframe blending approach is proposed to solve this problem whilst maintaining the advantages of a keyframe approach.

C. Contribution

The objective of this paper is to propose a complete real-time semantic SLAM system which leverages CNN semantic features not only to perform 3D semantic mapping, but also to improve the robustness of multi-keyframe pose estimation [35] by introducing semantic feature alignment. In addition to previous works, the proposed approach is based on a graph of semantic keyframes, regularly updated with new measurement using an efficient blending function which implements Bayesian fusion of the semantic probabilities.

The main contributions are threefold: (1) A real-time semantic segmentation network is trained using RGB-D images with noisy semantic labels obtained by image-to-model alignment with an industrial 3D CAD model. (2) A real-time semantic SLAM approach is proposed which extends [13] to real-time and adds multi-keyframe semantic mapping and fusion. (3) Experiments are demonstrated in real-time at 30 FPS on a real-world aircraft mock-up as part of an assembly line for robotic applications.

The remainder of this paper is organized as follows. Section III presents the real-time semantic SLAM pipeline including subsection III-A on real-time semantic segmentation, subsection III-B on semantic multi-keyframe pose estimation and subsection III-C on semantic class-level features fusion. Experimental results are presented in Section IV and conclusions are drawn in Section V.

III. REAL-TIME SEMANTIC SLAM

The proposed pipeline, illustrated in Figure 3, is composed of three separate units: a real-time semantic segmentation network, a multi-keyframe pose estimation approach based on semantic feature alignment and a 3D map represented as a graph of semantic keyframes.

A. *Real-time semantic segmentation*

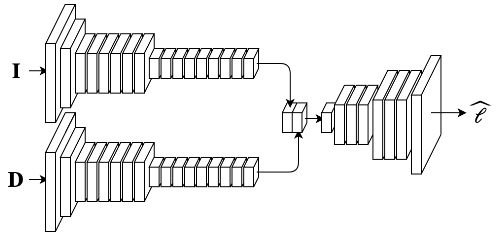


Fig. 2: Multi-modal architecture inspired by ERFNet.

The proposed CNN architecture is an adaptation of ERFNet [11], containing downsampling and upsampling layers groups. The downsampling blocks concatenate the parallel outputs of a 3×3 convolution and a 2×2 max pooling, both with stride 2. The upsampling block uses $3{\times}3$ transposed convolutions with stride 2. Between them, a sequence of residual units is inserted. These blocks are inspired from 2-layer ResNet units without bottleneck [36] except that each $3{\times}3$ convolution are decomposed in two 1D convolutions. The encoder has 5 of these after the second downsampling group, 8 with dilation after the third group, 2 after the first upsampling and 2 after the second upsampling.

Using such an architecture, we observe checkerboard artifacts induced by transposed convolutions with odd-sized kernels [37]. Unlike [11], we replaced the transposed convolutions with a bilinear upsampling followed by a $3{\times}3$ convolution. Built upon it, our multi-modal architecture (Fig. 2) employs a middle fusion scheme for RGB-D fusion. First, both input images have their own modality-specific encoder, then the high-level features from each modality are concatenated and refined using a $3{\times}3$ convolution. The outputs of the semantic segmentation network for the semantic SLAM are the semantic class-level feature maps (i.e. *logits*) before the softmax operation.

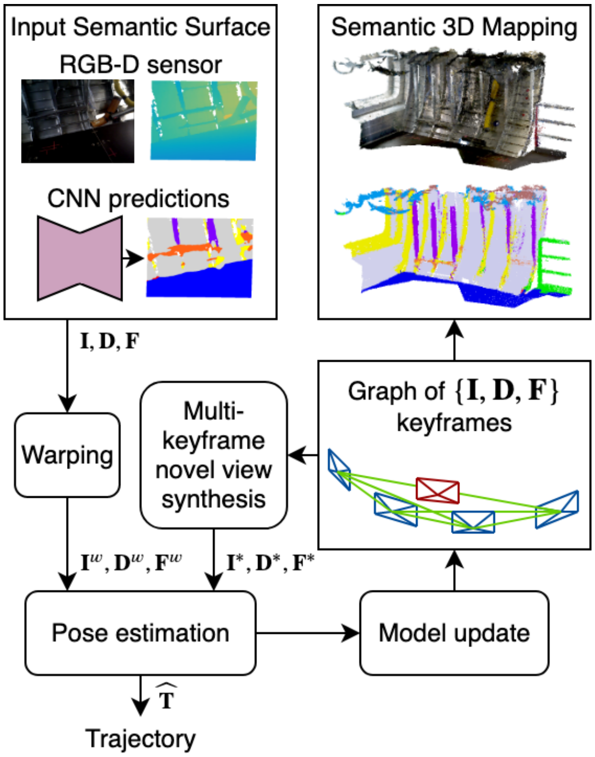

Input Semantic Surface
RGB-D sensor
CNN predictions
Semantic 3D Mapping
$\mathbf{I}, \mathbf{D}, \mathbf{F}$
Warping
Multi-keyframe novel view synthesis
Graph of $\{\mathbf{I}, \mathbf{D}, \mathbf{F}\}$ keyframes
$\mathbf{I}^w, \mathbf{D}^w, \mathbf{F}^w$
$\mathbf{I}^*, \mathbf{D}^*, \mathbf{F}^*$
Pose estimation
Model update
$\widehat{\mathbf{T}}$
Trajectory

Fig. 3: Overview of our pipeline. Input RGB-D images are used to infer class-level features which produce semantic labels. The pose estimation aligns the input surface with a virtual view synthesized from closest semantic keyframes along a graph. The semantic keyframe graph can easily be transformed into a semantic volumetric representation if required.

B. *Semantic multi-keyframe pose estimation*

Consider a calibrated RGB-D sensor with a color brightness function $\mathbf{I}{:}\Omega{\rightarrow}\mathbb{R}^+$, a depth function $\mathbf{D}{:}\Omega{\rightarrow}\mathbb{R}^+$ and a set $\mathbf{F}$ of semantic class-level feature functions $\mathbf{F}_c{:}\Omega{\rightarrow}\mathbb{R}$, where $\Omega{=}[1,n]\times[1,m]\subset\mathbb{R}^2$ and $(\mathbf{p}_1,\mathbf{p}_2,\dots,\mathbf{p}_{nm})^\top{\in}\Omega^{nm}$ are pixel locations within the image of dimension $n{\times}m$. The semantic label prediction function is defined as $\widehat{\boldsymbol{\ell}}{:}\Omega{\rightarrow}[1;C];(\mathbf{p}){\mapsto}\widehat{\boldsymbol{\ell}}(\mathbf{p}){=}argmax(\{\mathbf{F}_c(\mathbf{p})\}_{c=1..C})$ where C is the number of classes. The set $\boldsymbol{\mathcal{S}}{=}\{\mathbf{I},\mathbf{P},\mathbf{F}\}$ is defined to be a *semantic* 3D textured surface. The 3D points $\mathbf{P}_i$ are computed from homogeneous pixel locations $\bar{\mathbf{p}}_i$ by back-projection (1)

$$\mathbf{P}_i{=}\pi^{-1}(\mathbf{p}_i){=}\mathbf{K}^{-1}\bar{\mathbf{p}}_i\ \mathbf{D}(\mathbf{p}_i) \qquad (1)$$

where $\mathbf{K}{\in}\mathbb{R}^{3{\times}3}$ is the intrinsic camera matrix. The pose of the camera is represented as the homogeneous pose matrix $\mathbf{T}(\mathbf{x}){\in}\mathbb{SE}(3){\subset}\mathbb{R}^{4{\times}4}$ which depends on a minimal parameterisation $\mathbf{x}{\in}\mathfrak{se}(3){\subset}\mathbb{R}^6$. The pose matrix $\mathbf{T}(\mathbf{x})$ can be decomposed into rotational $\mathbf{R}(\mathbf{x}){\in}\mathbb{SO}(3)$ and translational components $\mathbf{t}(\mathbf{x}){\in}\mathbb{R}^3$. The inverse warping function $w(\cdot)$ projects the reference 3D points $\mathbf{P}_i^*$ transformed by $\mathbf{T}(\mathbf{x})$ onto the current frame at the warped pixel coordinates $\mathbf{p}_i^w{=}w(\mathbf{T}(\mathbf{x}),\mathbf{p}_i^*)$ (2)

$$\bar{\mathbf{p}}_i^w{=}\pi\big(\mathbf{R}(\mathbf{x})\pi^{-1}(\mathbf{p}_i^*){+}\mathbf{t}(\mathbf{x})\big) \qquad (2)$$

In this work, we apply the semantic surface alignment of [13] to frame-to-virtual keyframe odometry as part of a keyframe-based SLAM with multi-keyframe fusion [35].

Semantic surface alignment. We adapt the semantic visual odometry proposed in [13] to solve direct motion estimation between a *virtual* reference surface $\boldsymbol{\mathcal{S}}^*$ and a current surface $\boldsymbol{\mathcal{S}}$ by formulating a tri-objective cost function (3) that simultaneously minimizes a geometric error $\mathbf{e}_{\mathcal{G}}$, a photometric error $\mathbf{e}_{\mathcal{I}}$ and a semantic error $\mathbf{e}_{\mathcal{S}}$.

$$\mathbf{e}(\mathbf{x}) = \begin{bmatrix} \mathbf{e}_{\mathcal{G}}(\mathbf{x}) \\ \mathbf{e}_{\mathcal{I}}(\mathbf{x}) \\ \mathbf{e}_{\mathcal{S}}(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} \left(\widehat{\mathbf{R}}\mathbf{R}(\mathbf{x})\mathbf{N}^*\right)^\top \left(\mathbf{P}^m - \Pi_3 \widehat{\mathbf{T}}\mathbf{T}(\mathbf{x})\overline{\mathbf{P}}^*\right) \\ \mathbf{I}^*\left(w(\mathbf{T}(\mathbf{x}),\mathbf{p}^*)\right) - \mathbf{I}\left(w(\widehat{\mathbf{T}},\mathbf{p})\right) \\ \mathbf{F}^*\left(w(\mathbf{T}(\mathbf{x}),\mathbf{p}^*)\right) - \mathbf{F}\left(w(\widehat{\mathbf{T}},\mathbf{p})\right) \end{bmatrix} \tag{3}$$

where $\Pi_3 = [\mathbf{Id_3}, \mathbf{0}] \in \mathbb{R}^{3\times4}$ is the projection matrix, $\mathbf{N}_i^* \in \mathbb{R}^3$ is the surface normal for each homogeneous 3D point $\overline{\mathbf{P}}_i^*$. P_i^m is the closest point obtained by interpolating the warped pixel coordinates $\overline{\mathbf{p}}_i^w$ into the current depth map. The pose estimate $\widehat{\mathbf{T}}$ is computed at each iteration and is updated incrementally by a pose increment $\mathbf{T}(\mathbf{x})$ following an inverse compositional update rule $\widehat{\mathbf{T}} \leftarrow \widehat{\mathbf{T}}\mathbf{T}(\mathbf{x})$. This non-linear error is iteratively minimized using a Gauss-Newton approach with Huber's M-estimator for robust parameter estimation.

Semantic multi-keyframe view synthesis. The *virtual* reference surface $\boldsymbol{\mathcal{S}}^*$ is computed using multi-keyframe view synthesis inspired by [35]. The view synthesis is performed by first extracting the M (e.g. 5) closest keyframes to the current frame based on the distance along the keyframe graph. Each keyframe is then rasterized and blended into a virtual keyframe at the predicted camera viewpoint (4).

$$\boldsymbol{\mathcal{S}}^* = \sum_{k=1}^{M} f\left(\boldsymbol{\mathcal{S}}\left(\Gamma\left(\mathbf{P}^*, \mathbf{E}, w(\widehat{\mathbf{T}}^{-1}\mathbf{T}_k, \mathbf{p}_k)\right)\right)\right) \tag{4}$$

where $\mathbf{E}$ contains the indices of each triangle triangulated from the current depth map, Γ is the rasterization function efficiently implemented in hardware on GPUs and $f(\boldsymbol{\mathcal{S}}(\cdot))$ is a blending function that correctly fuses the synthesized surfaces with an efficient occlusion handling.
Compared to [35], we have added an extra component for *semantic* features to the surfaces $\boldsymbol{\mathcal{S}}$. Thus, a semantic label fusion scheme, described in Section III-C, has been designed to blend the class-level features into the *virtual* keyframe.

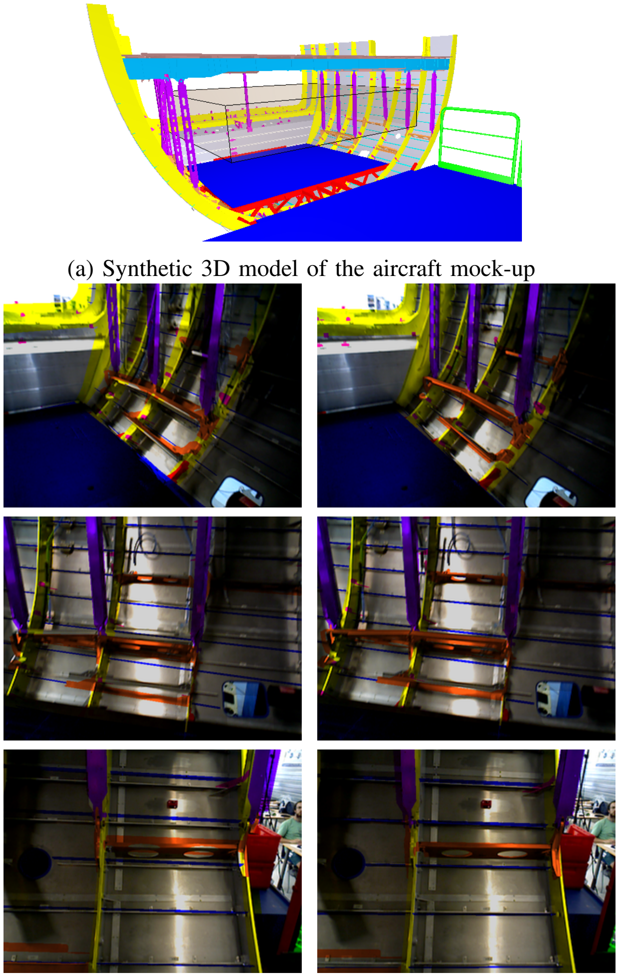
C. Semantic class-level features fusion

In addition to color image, depth map, normals and pose information, each keyframe in our graph, stores semantic class-level features $\mathbf{F}_c$ (i.e. the *logits* or classification scores) over the set of class labels, $c \in [1;C]$. Hence, semantic class-level features fusion must be performed both during *multi-keyframe view synthesis* and *model update* (Figure 3). Fusing semantic information in the model enforces long-term temporal consistency of the semantic image segmentation.

Bayesian fusion [32] provides an approach to integrate several measurements in the probability space. Assuming the data association of vertices from several keyframes has already been performed, let us denote the semantic labeling of a vertex for a given class c by $\boldsymbol{\ell}_c$ and its predictions using keyframe $\boldsymbol{\mathcal{S}}^k$ by $\widehat{\boldsymbol{\ell}}_c^k$. According to Bayes rule and by making strong hypothesis [31], the fused semantic labeling predictions can be computed (5) by taking the product over the semantic labeling likelihoods $\boldsymbol{\sigma}\left(\mathbf{F}_c^k\right)$ (i.e. the softmax $\boldsymbol{\sigma}(\cdot)$ of the *logits* $\mathbf{F}_c^k$).

$$p\left(\boldsymbol{\ell}_c | \widehat{\boldsymbol{\ell}}_c^{1,\ldots,k}\right) \simeq \prod_k \eta_i \boldsymbol{\sigma}\left(\mathbf{F}_c^k\right) = \boldsymbol{\sigma}\left(\sum_k \mathbf{F}_c^k\right) \tag{5}$$

It is worth mentioning that the fused class-level features can be computed directly by summing the individual keyframe's class-level features $\mathbf{F}_c^k$. Applying softmax $\boldsymbol{\sigma}(\cdot)$ on this sum yields the fused labeling probability distribution. For ease of computation, storing semantic class-level features $\mathbf{F}_c^k$ in the keyframe representation was preferred rather than labeling the probability distributions $\boldsymbol{\sigma}\left(\mathbf{F}_c^k\right)$.



(a) Synthetic 3D model of the aircraft mock-up

(b) Samples without ICP

(c) Samples with ICP

Fig. 4: Samples from AirMockUp *train* set. The images depict overlays of RGB images and their ground truth labels projected from the synthetic 3D model of the aircraft mock-up (4a) using global registration before (4b) and after (4c) post refinement using ICP algorithm.

TABLE II: Performance comparison for semantic segmentation on the A350MockUp *test* set in terms of mIoU (%). † indicates that the models were trained for 40K iterations and learning rate were decayed every 10K iterations.

	Fuselage	Frame	Frame support	Stringer	Vertical bar	Horiz. bar	Ground	Ground support	Upper deck	Bracket	Safety rail	mIoU
	⬜	🟨	🟧	🟦	🟪	🟦	🟦	🟥	🟫	🟪	🟩	
Class frequency	49.7%	9.3%	4.4%	2.9%	7.5%	5.2%	9.3%	0.3%	10.7%	0.5%	0.2%	
FCN-8s (RGB)	64.6	28.1	14.7	-	38.5	38.5	78.0	0.1	35.9	-	23.9	29.3
ERFNet (RGB)	72.1	37.0	1.2	-	47.7	41.4	82.2	-	61.9	-	38.4	34.8
Ours (RGB)	72.4	41.6	27.2	-	53.2	42.4	80.6	**3.8**	**62.1**	-	39.3	38.4†
Ours (D)	70.1	41.8	**32.8**	-	53.9	41.9	81.4	3.6	48.6	-	8.4	34.8†
Ours (RGB-D)	**72.9**	**44.3**	29.7	-	**58.1**	**43.7**	**82.3**	-	60.7	-	**39.5**	**39.2**†

IV. EXPERIMENTS

To show the effectiveness of the proposed approach within the particular context of an industrial environment, a new dataset named A350MockUp is described in Section IV-A. The temporal consistency of the semantic segmentation was evaluated through quantitative and qualitative experiments in Section IV-C and Section IV-D, respectively.

The semantic surface alignment has already proven to (1) enlarge the basin of convergence in challenging lighting conditions [14] and (2) to improve the convergence for frame-to-keyframe pose estimation [13]. However, this additional semantic term does not significantly improve the localization and mapping capabilities in nomal settings, hence this section will not evaluate localization nor reconstruction accuracies.

A. A350MockUp dataset

A RGB-D dataset was produced, with the aim of testing the proposed semantic segmentation network by training it on a real-world industrial environment for aircraft assembly. The dataset was collected using a hand-held ASUS Xtion PRO LIVE RGB-D camera within an aircraft mock-up in an aera spanning 8×4m as depicted in Fig. 1. RGB and depth images were captured at a resolution of 640×480 pixels at 30Hz. The acquisition camera traverses the mockup in multiple sequences with various linear and angular motions along with varying lighting conditions. The acquisition conditions are challenging due to low textured structural aircraft parts.

The synthetic 3D CAD model of the mock-up provided by Airbus (Fig. 4a) were exploited for training. Five RGB-D sequences were annotated with noisy labels using a semi-automated registration procedure. For each sequence, a 3D scene reconstruction is computed using SLAM and the 3D reconstruction is globally aligned with the synthetic 3D model. 2D ground truth labels are then automatically rendered for any frame in the input sequence via projection using the trajectory estimated by the SLAM. As the SLAM system sometimes lost tracking when degenerate configurations arose, some of the ground truth labels were not perfectly aligned with the RGB image (Fig. 4b). An iterative refinement step was implemented which aimed at refining the alignment of the real depth map and the synthetic depth map at the estimated pose using an Iterative Closest Point (ICP) algorithm (Fig. 4c).

Despite the refinement step, the annotation procedure still suffers from discrepancies between the 3D model of the mock-up e.g. pipes, cables, tools that are present in the real mock-up but not in the synthetic 3D model and vice versa for some brackets that are not yet mounted in the real mock-up. Thus the *trainval* set is composed of 6'884 images with noisy pixel-wise labels due to the aforementioned discrepancies, while the *test* set is composed of 25 carefully annotated images from sequences not used in the *trainval* set.

Demonstrating our method on a real-world industrial use case was a good opportunity to extract free ground truth labels from an existing 3D digital mock-up. However, our system could work on any environment (e.g. indoor, urban) for which a semantic segmentation model has been trained.

B. Training protocol

The proposed network is trained with an input image of resolution 640×480 without resizing the input images from the Xtion camera. The encoder part of the network is initialized with weights pre-trained on ImageNet [38], while the He initialization scheme [39] is used for the convolutional layers of the decoder. The PyTorch [40] framework is used for minimizing a softmax cross-entropy loss using the Adam optimizer [41] with standard momentum settings of $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\varepsilon = 10^{-8}$ for 20K iterations with a mini-batch size of 12 on a single NVIDIA Titan X GPU. The initial learning rate is set to $\lambda_0 = 5e^{-4}$ and decayed following a step learning rate policy at milestones {2.5K, 7.5K, 15K} by a factor 10^{-1}. It was found to be mandatory to employ general data augmentation techniques during training to prevent over-fitting and increase the effective number of training samples and their variability. A series a augmentation strategy was applied randomly including: scaling (0.5 to 2.0), cropping (640×480), left-right flipping, color (hue: ±0.05), brightness (0.8 to 1.2), saturation (0.8 to 1.2), contrast (0.9 to 1.1).

A multi-stage procedure is employed for training the RGB-D model using the middle fusion scheme, similar to [42]. First each modality-specific ERFNet model was trained individually using the aforementioned training procedure. In the second stage, transfer learning is leveraged by initializing only the encoder weights from the individual modality-specific encoders trained in the previous stage. Again, the He initialization scheme [39] is used for the convolutional layers of the decoder. The learning rate is set to $\lambda_0 = 5e^{-5}$ and

TABLE III: Ablation experiments for semantic segmentation on A350MockUp *test* set. **ICP**: the alignment of the ground truth labels with the RGB-D image were refined using ICP. **AUG**: train set is augmented using data augmentation. **WU**: training starts with a warmup phase. **CA**: decoder employs a combination of bilinear upsampling and convolution 3×3 rather than transposed convolution 3×3 to prevent checkerboard artifacts. **ENC**: encoder is pretrained on ImageNet. **ITER**: training lasts 40K iterations and learning rate were decayed every 10K iterations.

Data	ICP	AUG	WU	CA	ENC	ITER	mIoU (%)
RGB		✓	✓	✓	✓		30.8
	✓						29.2
	✓	✓					30.1 (+0.9)
	✓	✓	✓				31.3 (+1.2)
	✓	✓	✓	✓			32.9 (+1.6)
	✓	✓	✓	✓	✓		35.5 (+2.6)
	✓	✓	✓	✓	✓	✓	**38.4** (+2.9)
D	✓	✓	✓			✓	34.2
	✓	✓	✓	✓		✓	34.2
	✓	✓	✓	✓	✓	✓	**34.8** (+0.6)

decayed every 10K iterations by a factor 10^{-1}. The network is trained with a mini-batch of 8 for 30K iterations.

C. Quantitative results

The performance of the semantic segmentation network is evaluated using a standard metric namely the mean intersection over union (mIoU) defined as

$$\text{mIoU} = \frac{1}{C}\sum_{i=1}^{C}\frac{n_{ii}}{\sum_j n_{ji} + \sum_j n_{ij} - n_{ii}} \tag{6}$$

where C is the number of classes, n_{ij} is the number of pixels of class i classified as class j and $\sum_j n_{ij}$ is the total number of pixels belonging to class i. Multiscale inputs or left-right flips were not applied during testing, nor Conditional Random Field (CRF), as these techniques increase the computational complexity and the runtime.

Table II shows the performance comparison with baseline models including FCN-8s [2] and ERFNet [11] for the A350MockUp dataset. Our segmentation model performs well on most of the classes with class IoU greater than 40% for most of them. However, it was noticed that none of the models succeed in learning to segment the *stringer* and *bracket* classes. On one hand, *stringers* are objects which are too thin and their ground truth labels are too noisy to be successfully segmented. On the other hand, *brackets* are small objects and, most of all, they suffer from severe discrepancies between the real mock-up and its synthetic 3D model. It can be seen that the depth modality performs better for geometric parts e.g. *frame support* while the RGB modality performs better on all other classes. Averaged on all classes, the proposed RGB-D network improves by +4.2% over ERFNet and +9.9% over FCN-8s.

Table III shows ablation experiments of different contributions. The refinement of the ground truth labels using **ICP** gives an improvement of +4.7%. Data augmentation (**AUG**) improves by +0.9% but significantly helps in preventing overfitting during training. Warmup (**WU**) is a good practice for adaptive optimizers to offset excessive variance when the optimizer has only worked with limited training data. Unlike [11], we adjoin a warmup (an initial period of training with a much lower learning rate) to Adam for a +1.2% gain. Replacing the transposed convolutions in the decoder with bilinear upsampling followed by convolutions to fix checkerboard artifacts (**CA**) achieves an improvement of +1.6% and initializing the encoder with weights pretrained on ImageNet (**ENC**) improves by +2.6%. Finally, we train our best model for extra iterations (**ITER**) with a slower learning rate policy and we improve by 2.9%.

In practice, the multi-keyframe novel view synthesis has improved the semantic image segmentation by almost 1.5%.

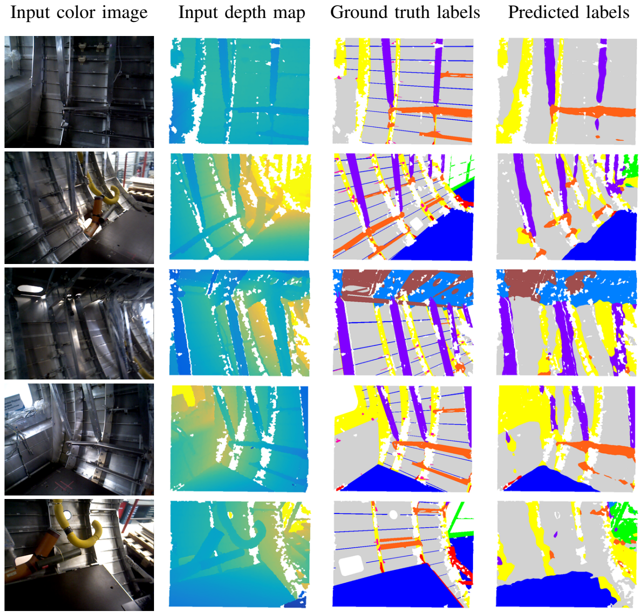


Fig. 5: Qualitative A350MockUp test set results at varying viewpoints. The semantic categories are color coded as follows: ⬜ Fuselage, 🟨 Frame, 🟧 Frame support, 🟦 Stringer, 🟪 Vertical bar, 🟦 Horizontal bar, 🟦 Ground, 🟥 Ground support, 🟫 Upper deck, 🟪 Bracket, 🟩 Safety rail.

D. Qualitative results

Fig. 1 illustrates the semantic 3D mapping output that the system produces from the graph of semantic keyframes. Most classes are well segmented except the *frame supports* which are occasionally confused with the *fuselage*.

Fig. 5 shows label predictions inferred by the proposed semantic segmentation network. It is able to accurately segment the scene while being robust to challenging lighting conditions, reflective surfaces despite a training with noisy labels due to discrepancies due to the real-world and the synthetic mock-up. As mentioned in Section IV-C, the trained model is not able to segment *stringers* and *brackets* and it can be noted that *ground supports* are extremely underrepresented in the dataset.

The last two rows show failure modes. In the third row, the model does not predict a *void* category for objects that are outside the mock-up as it was not trained to do so. Similarly, the model is not able to generalize to never seen objects e.g. wooden pallets on the floor which are categorized as *frame* in the last row. Alternatively, pipes were seen in the 'trainval' set but handled as though they were transparent i.e. the 'pipes' class is assigned the labels of the object they are occluding. In the last row, pipes are successfully inferred with the *fuselage* they were occluding.

Comparing the segmentation accuracies of Fig. 1 and Fig. 5 shows that the graph of semantic keyframes as a 3D map representation permits qualitatively more accurate 3D segmentation results in particular for thin objects such as *frame support*, *vertical bar* and *safety rail*.

E. Real-time implementation

The SLAM system is fully integrated into ROS and optimized on the GPU. Its performance is evaluated on random sequences from the A350MockUp test set. The whole system processes every frame at 30 FPS on an Intel Xeon E5-1620 3.50GHz CPU and a NVIDIA GTX 980 Ti GPU. The semantic segmentation network processes every frame and the semantic class-level feature alignment does not incur any time overhead since it only aligns a single feature map per pyramid level.

V. CONCLUSIONS

A novel real-time semantic multi-keyframe SLAM approach was proposed for autonomous navigation and control of a humanoid robot in a manufacturing scenario. An industrial CAD model was used for training a custom segmentation network. Experimental results show large-scale 3D maps with associated object labels relevant to a real-world robotic manufacturing scenario.

Future work will be dedicated to design an adaptive multi-view feature fusion layer similarly to [42] whose fusion scheme learns the most favorable element-wise weighting for the fusion. Further evaluations will be performed using data that have been acquired using an external motion capture system for comparing with more ground truth poses.

REFERENCES

[1] A. Tanguy, D. De Simone, A. I. Comport, G. Oriolo, and A. Kheddar, "Closed-loop MPC with Dense Visual SLAM-Stability through Reactive Stepping," in *IEEE International Conference on Robotics and automation (ICRA)*, 20–24 May 2019, working paper or preprint. 1

[2] 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*, 2015, pp. 3431–3440. 1, 6

[3] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2881–2890. 1

[4] H. Zhao, X. Qi, X. Shen, J. Shi, and J. Jia, "Icnet for real-time semantic segmentation on high-resolution images," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 405–420. 1, 2

[5] V. Nekrasov, C. Shen, and I. Reid, "Light-weight refinenet for real-time semantic segmentation," in *Proceedings of the British Machine Vision Conference (BMVC)*, 2018, p. 125. 2

[6] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv preprint arXiv:1706.05587*, 2017. 2

[7] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4510–4520. 2

[8] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, "Enet: A deep neural network architecture for real-time semantic segmentation," *arXiv preprint arXiv:1606.02147*, 2016. 2

[9] S. Mehta, M. Rastegari, A. Caspi, L. Shapiro, and H. Hajishirzi, "Espnet: Efficient spatial pyramid of dilated convolutions for semantic segmentation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 552–568. 2

[10] S. Mehta, M. Rastegari, L. Shapiro, and H. Hajishirzi, "Espnetv2: A light-weight, power efficient, and general purpose convolutional neural network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9190–9200. 2

[11] E. Romera, J. M. Alvarez, L. M. Bergasa, and R. Arroyo, "Erfnet: Efficient residual factorized convnet for real-time semantic segmentation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 1, pp. 263–272, 2018. 2, 3, 6

[12] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3213–3223. 2

[13] H. Mahé, D. Marraud, and A. I. Comport, "Semantic-only visual odometry based on dense class-level segmentation," in *2018 24th International Conference on Pattern Recognition (ICPR)*. IEEE, 2018, pp. 1989–1995. 2, 3, 4, 5

[14] J. Czarnowski, S. Leutenegger, and A. J. Davison, "Semantic texture for robust dense tracking," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 860–868. 2, 5

[15] R. Clark, M. Bloesch, J. Czarnowski, S. Leutenegger, and A. J. Davison, "Learning to solve nonlinear least squares for monocular stereo," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 284–299. 2

[16] C. Wang, H. K. Galoogahi, C.-H. Lin, and S. Lucey, "Deep-lk for efficient adaptive object tracking," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 627–634. 2

[17] L. Han, M. Ji, L. Fang, and M. Nießner, "Regnet: Learning the optimization of direct image-to-image pose registration," *arXiv preprint arXiv:1812.10212*, 2018. 2

[18] Z. Lv, F. Dellaert, J. M. Rehg, and A. Geiger, "Taking a deeper look at the inverse compositional algorithm," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4581–4590. 2

[19] C. Tang and P. Tan, "Ba-net: Dense bundle adjustment network," in *International Conference on Learning Representations*, 2019. 2

[20] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "Scannet: Richly-annotated 3d reconstructions of indoor scenes," in *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017. 2

[21] G. Riegler, A. Osman Ulusoy, and A. Geiger, "Octnet: Learning deep 3d representations at high resolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3577–3586. 2

[22] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, vol. 1, no. 2, p. 4, 2017. 2

[23] J. P. Valentin, S. Sengupta, J. Warrell, A. Shahrokni, and P. H. Torr, "Mesh based semantic modelling for indoor and outdoor scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 2067–2074. 2

[24] S. Sengupta, E. Greveson, A. Shahrokni, and P. H. Torr, "Urban 3d semantic modelling using stereo vision," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 580–585. 2

[25] C. Zhao, L. Sun, and R. Stolkin, "A fully end-to-end deep learning approach for real-time simultaneous 3d reconstruction and material recognition," in *Advanced Robotics (ICAR), 2017 18th International Conference on*. IEEE, 2017, pp. 75–82. 2

[26] J. Stückler, B. Waldvogel, H. Schulz, and S. Behnke, "Dense real-time mapping of object-class semantics from rgb-d video," *Journal of Real-Time Image Processing*, vol. 10, no. 4, pp. 599–609, 2015. 2

[27] K. Tateno, F. Tombari, and N. Navab, "Real-time and scalable incremental segmentation on dense slam," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE, 2015, pp. 4465–4472. 2

[28] J. McCormac, A. Handa, A. Davison, and S. Leutenegger, "Semanticfusion: Dense 3d semantic mapping with convolutional neural networks," in *2017 IEEE International Conference on Robotics and automation (ICRA)*. IEEE, 2017, pp. 4628–4635. 2

[29] A. Hermans, G. Floros, and B. Leibe, "Dense 3d semantic mapping of indoor scenes from rgb-d images," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 2631–2638. 2

[30] Y. He, W.-C. Chiu, M. Keuper, and M. Fritz, "Std2p: Rgbd semantic segmentation using spatio-temporal data-driven pooling," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4837–4846. 2

[31] L. Ma, J. Stückler, C. Kerl, and D. Cremers, "Multi-view deep learning for consistent semantic mapping with rgb-d cameras," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 598–605. 2, 4

[32] S. Thrun, W. Burgard, and D. Fox, "Probabilistic robotics, ser. intelligent robotics and autonomous agents," *Massachusetts Institute of Technology, Cambridge*, 2005. 2, 4

[33] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking," in *2011 IEEE International Symposium on Mixed and Augmented Reality*. IEEE, 2011, pp. 127–136. 2

[34] T. Whelan, S. Leutenegger, R. Salas-Moreno, B. Glocker, and A. Davison, "Elasticfusion: Dense slam without a pose graph." Robotics: Science and Systems, 2015. 2

[35] M. Meilland and A. I. Comport, "On unifying key-frame and voxel-based dense visual slam at large scales," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 3677–3683. 2, 4

[36] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778. 3

[37] A. Odena, V. Dumoulin, and C. Olah, "Deconvolution and checkerboard artifacts," *Distill*, vol. 1, no. 10, p. e3, 2016. 3

[38] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," 2009. 5

[39] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034. 5

[40] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017. 5

[41] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2015. 5

[42] A. Valada, R. Mohan, and W. Burgard, "Self-supervised model adaptation for multimodal semantic segmentation," *International Journal of Computer Vision (IJCV)*, 2019. 5, 7