SemanticFusion: Dense 3D Semantic Mapping with Convolutional Neural Networks

John McCormac, Ankur Handa, Andrew Davison, and Stefan Leutenegger
Dyson Robotics Lab, Imperial College London

***Abstract*— Ever more robust, accurate and detailed mapping using visual sensing has proven to be an enabling factor for mobile robots across a wide variety of applications. For the next level of robot intelligence and intuitive user interaction, maps need extend beyond geometry and appearence — they need to contain semantics. We address this challenge by combining Convolutional Neural Networks (CNNs) and a state of the art dense Simultaneous Localisation and Mapping (SLAM) system, ElasticFusion, which provides long-term dense correspondence between frames of indoor RGB-D video even during loopy scanning trajectories. These correspondences allow the CNN's semantic predictions from multiple view points to be probabilistically fused into a map. This not only produces a useful semantic 3D map, but we also show on the NYUv2 dataset that fusing multiple predictions leads to an improvement even in the 2D semantic labelling over baseline single frame predictions. We also show that for a smaller reconstruction dataset with larger variation in prediction viewpoint, the improvement over single frame segmentation increases. Our system is efficient enough to allow real-time interactive use at frame-rates of ≈25Hz.**

I. INTRODUCTION

The inclusion of rich semantic information within a dense map enables a much greater range of functionality than geometry alone. For instance, in domestic robotics, a simple fetching task requires knowledge of both what something is, as well as where it is located. As a specific example, thanks to sharing of the same spatial and semantic understanding between user and robot, we may issue commands such as 'fetch the coffee mug from the nearest table on your right'. Similarly, the ability to query semantic information within a map is useful for humans directly, providing a database for answering spoken queries about the semantics of a previously made map; 'How many chairs do we have in the conference room? What is the distance between the lectern and its nearest chair?' In this work, we combine the geometric information from a state-of-the-art SLAM system ElasticFusion [25] with recent advances in semantic segmentation using Convolutional Neural Networks (CNNs).

Our approach is to use the SLAM system to provide correspondences from the 2D frame into a globally consistent 3D map. This allows the CNN's semantic predictions from multiple viewpoints to be probabilistically fused into a dense semantically annotated map, as shown in Figure 1. ElasticFusion is particularly suitable for fusing semantic labels because its surfel-based surface representation is automatically deformed to remain consistent after the small and large loop closures which would frequently occur during typical interactive use by an agent (whether human or robot). As the surface representation is deformed and corrected, individual surfels remain persistently associated with real-world entities and this enables long-term fusion of per-frame semantic predictions over wide changes in viewpoint.

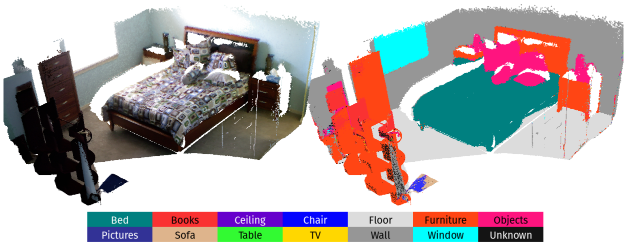

Fig. 1: **The output of our system**: On the left, a dense surfel based reconstruction from a video sequence in the NYUv2 test set. On the right the same map, semantically annotated with the classes given in the legend below.

The geometry of the map itself also provides useful information which can be used to efficiently regularise the final predictions. Our pipeline is designed to work online, and although we have not focused on performance, the efficiency of each component leads to a real-time capable (≈ 25Hz) interactive system. The resulting map could also be used as a basis for more expensive offline processing to further improve both the geometry and the semantics; however that has not been explored in the current work.

We evaluate the accuracy of our system on the NYUv2 dataset, and show that by using information from the unlabelled raw video footage we can improve upon baseline approaches performing segmentation using only a single frame. This suggests the inclusion of SLAM not only provides an immediately useful semantic 3D map, but it suggests that many state-of-the art 2D single frame semantic segmentation approaches may be boosted in performance when linked with SLAM.

The NYUv2 dataset was not taken with full room reconstruction in mind, and often does not provide significant variations in viewpoints for a given scene. To explore the benefits of SemanticFusion within a more thorough reconstruction, we developed a small dataset of a reconstructed office room, annotated with the NYUv2 semantic classes. Within this dataset we witness a more significant improvement in segmentation accuracy over single frame 2D segmentation. This indicates that the system is particularly well suited to longer duration scans with wide viewpoint variation aiding to disambiguate the single-view 2D semantics.

II. RELATED WORK

The works most closely related are Stückler *et al.* [23] and Hermans *et al.* [8]; both aim towards a dense, semantically annotated 3D map of indoor scenes. They both obtain per-pixel label predictions for incoming frames using Random Decision Forests, whereas ours exploits recent advances in Convolutional Neural Networks that provide state-of-the-art accuracy, with a real-time capable run-time performance. They both fuse predictions from different viewpoints in a classic Bayesian framework. Stückler *et al.* [23] used a Multi-Resolution Surfel Map-based SLAM system capable of operating at 12.8Hz, however unlike our system they do not maintain a single global semantic map as local key frames store aggregated semantic information and these are subject to graph optimisation in each frame. Hermans *et al.* [8] did not use the capability of a full SLAM system with explicit loop closure: they registered the predictions in the reference frames using only camera tracking. Their run-time performance was 4.6Hz, which would prohibit processing a live video feed, whereas our system is capable of operating online and interactively. As here, they regularised their predictions using Krähenbühl and Koltun's [13] fully-connected CRF inference scheme to obtain a final semantic map.

Previous work by Salas-Moreno *et al.* aimed to create a fully capable SLAM system, SLAM++ [19], which maps indoor scenes at the level of semantically defined objects. However, their method is limited to mapping objects that are present in a pre-defined database. It also does not provide the dense labelling of entire scenes that we aim for in this work, which also includes walls, floors, doors, and windows which are equally important to describe the extent of the room. Additionally, the features they use to match template models are hand-crafted unlike our CNN features that are learned in an end-to-end fashion with large training datasets.

The majority of other approaches to indoor semantic labelling either focuses on offline batch mapping methods [24], [12] or on single-frame 2D segmentations which do not aim to produce a semantically annotated 3D map [3], [20], [15], [22]. Valentin *et al.* [24] used a CRF and a per-pixel labelling from a variant of TextonBoost to reconstruct semantic maps of both indoor and outdoor scenes. This produces a globally consistent 3D map, however inference is performed on the whole mesh once instead of incrementally fusing the predictions online. Koppula *et al.* [12] also tackle the problem on a completed 3D map, forming segments of the map into nodes of a graphical model and using hand-crafted geometric and visual features as edge potentials to infer the final semantic labelling.

Our semantic mapping pipeline is inspired by the recent success of Convolution Neural Networks in semantic labelling and segmentation tasks [14], [16], [17]. CNNs have proven capable of both state-of-the-art accuracy and efficient test-time performance. They have have exhibited these capabilities on numerous datasets and a variety of data modalities, in particular RGB [17], [16], Depth [1], [7] and Normals [2], [4], [6], [5]. In this work we build on the CNN model proposed by Noh *et. al.* [17], but modify it to take advantage of the directly available depth data in a manner that does not require significant additional pre-processing.

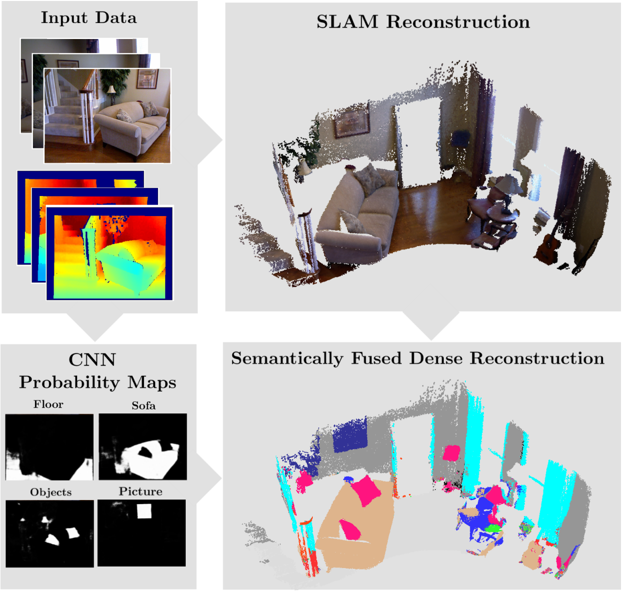

Fig. 2: **An overview of our pipeline**: Input images are used to produce a SLAM map, and a set of probability prediction maps (here only four are shown). These maps are fused into the final dense semantic map via Bayesian updates.

III. METHOD

Our SemanticFusion pipeline is composed of three separate units; a real-time SLAM system ElasticFusion, a Convolutional Neural Network, and a Bayesian update scheme, as illustrated in Figure 2. The role of the SLAM system is to provide correspondences between frames, and a globally consistent map of fused surfels. Separately, the CNN receives a 2D image (for our architecture this is RGBD, for Eigen *et al.* [2] it also includes estimated normals), and returns a set of per pixel class probabilities. Finally, a Bayesian update scheme keeps track of the class probability distribution for each surfel, and uses the correspondences provided by the SLAM system to update those probabilities based on the CNN's predictions. Finally, we also experiment with a CRF regularisation scheme to use the geometry of the map itself to improve the semantic predictions [8], [13]. The following section outlines each of these components in more detail.

A. SLAM Mapping

We choose ElasticFusion as our SLAM system.[1] For each arriving frame, k, ElasticFusion tracks the camera pose via a combined ICP and RGB alignment, to yield a new pose $\boldsymbol{T}_{WC}$, where W denotes the World frame and C the camera frame. New surfels are added into our map using this camera pose, and existing surfel information is combined with new evidence to refine their positions, normals, and

[1] Available on `https://github.com/mp3guy/ElasticFusion`

colour information. Additional checks for a loop closure event run in parallel and the map is optimised immediately upon a loop closure detection.

The deformation graph and surfel based representation of ElasticFusion lend themselves naturally to the task at hand, allow probability distributions to be 'carried along' with the surfels during loop closure, and also fusing new depth readings to update the surfel's depth and normal information, without destroying the surfel, or its underlying probability distribution. It operates at real-time frame-rates at VGA resolution and so can be used both interactively by a human or in robotic applications. We used the default parameters in the public implementation, except for the depth cutoff, which we extend from 3m to 8m to allow reconstruction to occur on sequences with geometry outside of the 3m range.

B. CNN Architecture

Our CNN is implemented in *caffe* [11] and adopts the Deconvolutional Semantic Segmentation network architecture proposed by Noh *et. al.* [17]. Their architecture is itself based on the VGG 16-layer network [21], but with the addition of max unpooling and deconvolutional layers which are trained to output a dense pixel-wise semantic probability map. This CNN was trained for RGB input, and in the following sections when using a network with this setup we describe it RGB-CNN.

Given the availability of depth data, we modified the original network architecture to accept depth information as a fourth channel. Unfortunately, the depth modality lacks the large scale training datasets of its RGB counterpart. The NYUv2 dataset only consists of 795 labelled training images. To effectively use depth, we initialized the depth filters with the average intensity of the other three inputs, which had already been trained on a large dataset, and converted it from the 0–255 colour range to the 0–8m depth range by increasing the weights by a factor of $\approx 32\times$.

We rescale incoming images to the native 224×224 resolution for our CNNs, using bilinear interpolation for RGB, and nearest neighbour for depth. In our experiments with Eigen *et. al.'s* implementation we rescale the inputs in the same manner to 320×240 resolution. We upsample the network output probabilites to full 640×480 image resolution using nearest neighbour when applying the update to surfels, described in the section below.

C. Incremental Semantic Label Fusion

In addition to normal and location information, each surfel (index s) in our map, $\mathcal{M}$, stores a discrete probability distribution, $P(L_s = l_i)$ over the set of class labels, $l_i \in \mathcal{L}$. Each newly generated surfel is initialised with a uniform distribution over the semantic classes, as we begin with no *a priori* evidence as to its latent classification.

After a prespecified number of frames, we perform a forward pass of the CNN with the image $\boldsymbol{I}_k$ coming directly from the camera. Depending on the CNN architecture, this image can include any combination of RGB, depth, or normals. Given the data $\boldsymbol{I}_k$ of the k^{th} image, the output of the CNN is interpreted in a simplified manner as a per-pixel independent probability distribution over the class labels $P(O_{\boldsymbol{u}} = l_i|\boldsymbol{I}_k)$, with $\boldsymbol{u}$ denoting pixel coordinates.

Using the tracked camera pose $\boldsymbol{T}_{WC}$, we associate every surfel at a given 3D location ${}_W\boldsymbol{x}(s)$ in the map, with pixel coordinates $\boldsymbol{u}$ via the camera projection $\boldsymbol{u}(s,k) = \boldsymbol{\pi}(\boldsymbol{T}_{CW}(k)_W\boldsymbol{x}(s))$, employing the homogeneous transformation matrix $\boldsymbol{T}_{CW}(k) = \boldsymbol{T}_{WC}^{-1}(k)$ and using homogeneous 3D coordinates. This enables us to update all the surfels in the visible set $\mathcal{V}_k \subseteq \mathcal{M}$ with the corresponding probability distribution by means of a recursive Bayesian update

$$P(l_i|\boldsymbol{I}_{1,\ldots,k}) = \frac{1}{Z}P(l_i|\boldsymbol{I}_{1,\ldots,k-1})P(O_{\boldsymbol{u}(s,k)} = l_i|\boldsymbol{I}_k), \quad (1)$$

which is applied to all label probabilities per surfel, finally normalising with constant Z to yield a proper distribution.

It is the SLAM correspondences that allow us to accurately associate label hypotheses from multiple images and combine evidence in a Bayesian way. The following section discusses how the naïve independence approximation employed so far can be mitigated, allowing semantic information to be propagated spatially when semantics are fused from different viewpoints.

D. Map Regularisation

We explore the benefits of using map geometry to regularise predictions by applying a fully-connected CRF with Gaussian edge potentials to surfels in the 3D world frame, as in the work of Hermans *et al.* [8], [13]. We do not use the CRF to arrive at a final prediction for each surfel, but instead use it incrementally to update the probability distributions. In our work, we treat each surfel as a node in the graph. The algorithm uses the mean-field approximation and a message passing scheme to efficiently infer the latent variables that approximately minimise the Gibbs energy E of a labelling, $\mathbf{x}$, in a fully-connected graph, where $x_s \in \{l_i\}$ denotes a given labelling for the surfel with index s.

The energy $E(\mathbf{x})$ consists of two parts, the unary data term $\psi_u(x_s)$ is a function of a given label, and is parameterised by the internal probability distribution of the surfel from fusing multiple CNN predictions as described above. The pairwise smoothness term, $\psi_p(x_s, x_{s'})$ is a function of the labelling of two connected surfels in the graph, and is parameterised by the geometry of the map:

$$E(\mathbf{x}) = \sum_s \psi_u(x_s) + \sum_{s<s'} \psi_p(x_s, x_{s'}). \quad (2)$$

For the data term we simply use the negative logarithm of the chosen labelling's probability for a given surfel,

$$\psi_u(x_s) = -\log(P(L_s = x_s|\boldsymbol{I}_{1,\ldots,k})). \quad (3)$$

In the scheme proposed by Krähenbühl and Koltun [13] the smoothness term is constrained to be a linear combination of K Gaussian edge potential kernels, where $\mathbf{f}_s$ denotes some feature vector for surfel, s, and in our case $\mu(x_s, x_{s'})$ is given by the Potts model, $\mu(x_s, x_{s'}) = [x_s \neq x_{s'}]$:

$$\psi_p(x_s, x_{s'}) = \mu(x_s, x_{s'}) \left(\sum_{m=1}^{K} w^{(m)} k^{(m)}(\mathbf{f}_s, \mathbf{f}_{s'}) \right). \quad (4)$$

Following previous work [8] we use two pairwise potentials; a bilateral appearance potential seeking to closely tie together surfels with both a similar position and appearance, and a spatial smoothing potential which enforces smooth predictions in areas with similar surface normals:

$$k^1(\mathbf{f}_s, \mathbf{f}_{s'}) = \exp\left(-\frac{|\mathbf{p}_s - \mathbf{p}_{s'}|^2}{2\theta_\alpha^2} - \frac{|\mathbf{c}_s - \mathbf{c}_{s'}|^2}{2\theta_\beta^2} \right), \quad (5)$$

$$k^2(\mathbf{f}_s, \mathbf{f}_{s'}) = \exp\left(-\frac{|\mathbf{p}_s - \mathbf{p}_{s'}|^2}{2\theta_\alpha^2} - \frac{|\mathbf{n}_s - \mathbf{n}_{s'}|^2}{2\theta_\gamma^2} \right). \quad (6)$$

We chose unit standard deviations of $\theta_\alpha = 0.05$m in the spatial domain, $\theta_\beta = 20$ in the RGB colour domain, and $\theta_\gamma = 0.1$ radians in the angular domain. We did not tune these parameters for any particular dataset. We also maintained w^1 of 10 and w^2 of 3 for all experiments. These were the default settings in Krähenbühl and Koltun's public implementation[2] [13] .

IV. EXPERIMENTS

A. Network Training

We initialise our CNNs with weights from Noh *et. al.* [17] trained for segmentation on the PASCAL VOC 2012 segmentation dataset [3]. For depth input we initialise the fourth channel as described in Section III-B, above. We finetuned this network on the training set of the NYUv2 dataset for the 13 semantic classes defined by Couprie *et al.* [1].

For optimisation we used standard stochastic gradient descent, with a learning rate of 0.01, momentum of 0.9, and weight decay of 5×10^{-4}. After 10k iterations we reduced the learning rate to 1×10^{-3}. We use a mini-batch size of 64, and trained the networks for a total of 20k iterations over the course of 2 days on an Nvidia GTX Titan X.

B. Reconstruction Dataset

We produced a small experimental RGB-D reconstruction dataset, which aimed for a relatively complete reconstruction of an office room. The trajectory used is notably more loopy, both locally and globally, than the NYUv2 dataset which typically consists of a single back and forth sweep. We believe the trajectory in our dataset is more representative of the scanning motion an active agent may perform when inspecting a scene.

We also took a different approach to manual annotation of this data, by using a 3D tool we developed to annotate the surfels of the final 3D reconstruction with the 13 NYUv2 semantic classes under consideration (only 9 were present). We then automatically generated 2D labellings for any frame in the input sequence via projection. The tool, and the resulting annotations are depicted in Figure 3. Every 100th frame of the sequence was used as a test sample to validate our predictions against the annotated ground truth, resulting in 49 test frames.

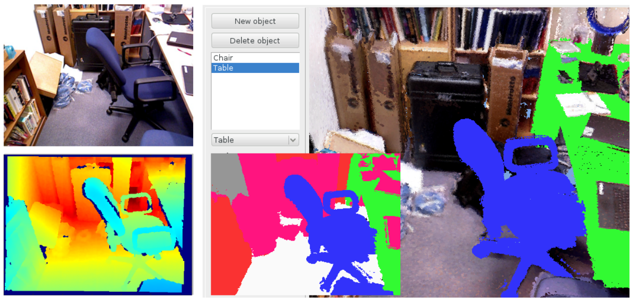

Fig. 3: **Our office reconstruction dataset**: On the left are the captured RGB and Depth images. On the right, is our 3D reconstruction and annotation. Inset into that is the final ground truth rendered labelling we use for testing.

C. CNN and CRF Update Frequency Experiments

We used the dataset to evaluate the accuracy of our system when only performing a CNN prediction on a subset of the incoming video frames. We used the RGB-CNN described above, and evaluated the accuracy of our system when performing a prediction on every 2^n frames, where $n \in \{0..7\}$. We calculate the average frame-rate based upon the run-time analysis discussed in Section IV-F. As shown in Figure 4, the accuracy is highest (52.5%) when every frame is processed by the network, however this leads to a significant drop in frame-rate to 8.2Hz. Processing every 10th frame results in a slightly reduced accuracy (49-51%), but over three times the frame-rate of 25.3Hz. This is the approach taken in all of our subsequent evaluations.

We also evaluated the effect of varying the number of frames between CRF updates (Figure 5). We found that when applied too frequently, the CRF can 'drown out' predictions of the CNN, resulting in a significant reduction in accuracy. Performing an update every 500 frames results in a slight improvement, and so we use that as the default update rate in all subsequent experiments.

D. Accuracy Evaluation

We evaluate the accuracy of our SemanticFusion pipeline against the accuracy achieved by a single frame CNN segmentation. The results of this evaluation are summarised in Table I. We observe that in all cases semantically fusing additional viewpoints improved the accuracy of the segmentation over a single frame system. Performance improved from 43.6% for a single frame to 48.3% when projecting the predictions from the 3D SemanticFusion map.

We also evaluate our system on the office dataset when using predictions from the state-of-the-art CNN developed by Eigen *et al.*[3] based on the VGG architecture. To maintain

[2] Available from: `http://www.philkr.net/home/densecrf`

[3] We use the publicly available network weights and implementation from: `http://www.cs.nyu.edu/~deigen/dnl/`.

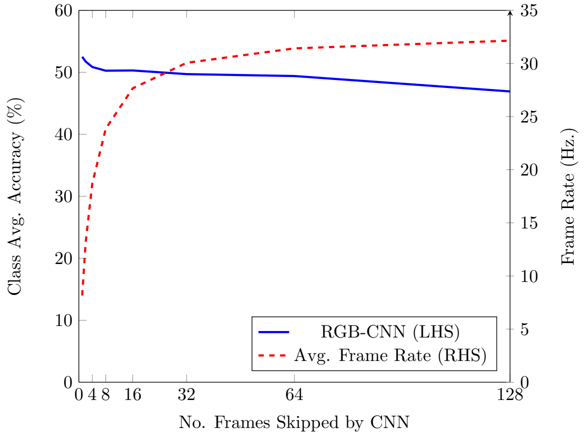

Fig. 4: The class average accuracy of our RGB-CNN on the office reconstruction dataset against the number of frames skipped between fusing semantic predictions. We perform this evaluation without CRF smoothing. The right hand axis shows the estimated run-time performance in terms of FPS.

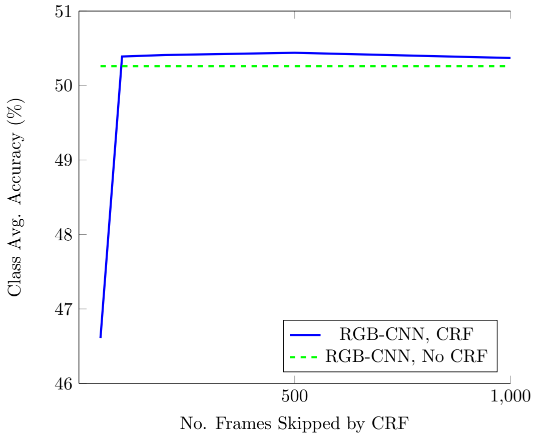

Fig. 5: The average class accuracy processing every 10th frame with a CNN, with a variable number of frames between CRF updates. If applied too frequently the CRF was detrimental to performance, and the performance improvement from the CRF was not significant for this CNN.

consistency with the rest of the system, we perform only a single forward pass of the network to calculate the output probabilities. The network requires ground truth normal information, and so to ensure the input pipeline is the same as in Eigen *et al.* [2], we preprocess the sequence with the MATLAB script linked to in the project page to produce the ground truth normals. With this setup we see an improvement of 2.9% over the single frame implementation with SemanticFusion, from 57.1% to 60.0%.

The performance benefit of the CRF was less clear. It provided a very small improvement of 0.5% for the Eigen network, but a slight detriment to the RGBD-CNN of 0.2%.

E. NYU Dataset

We choose to validate our approach on the NYUv2 dataset [20], as it is one of the few datasets which provides all of the information required to evaluate semantic RGB-D reconstruction. The SUN RGB-D [22], although an order of magnitude larger than NYUv2 in terms of labelled images, does not provide the raw RGB-D videos and therefore is could not be used in our evaluation.

The NYUv2 dataset itself is still not ideally suited to the role. Many of the 206 test set video sequences exhibit significant drops in frame-rate and thus prove unsuitable for tracking and reconstruction. In our evaluations we excluded any sequence which experienced a frame-rate under 2Hz. The remaining 140 test sequences result in 360 labelled test images of the original 654 image test set in NYUv2. The results of our evaluation are presented in Table II and some qualitative results are shown in Figure 6.

Overall, fusing semantic predictions resulted in a notable improvement over single frame predictions. However, the total relative gains of 2.3% for the RGBD-CNN was approximately half of the 4.7% improvement witnessed in the office reconstruction dataset. We believe this is largely a result of the style of capturing NYUv2 datasets. The primarily rotational scanning pattern often used in test trajectories does not provide as many useful different viewpoints from which to fuse independent predictions. Despite this, there is still a significant accuracy improvement over the single frame predictions.

We also improved upon the state-of-the-art Eigen *et al.* [2] CNN, with the class average accuracy going from 59.9% to 63.2% (+3.3%). This result clearly shows, even on this challenging dataset, the capacity of SemanticFusion to not only provide a useful semantically annotated 3D map, but also to improve the predictions of state-of-the-art 2D semantic segmentation systems.

The improvement as a result of the CRF was not particularly significant, but positive for both CNNs. Eigen's CNN saw +0.4% improvement, and the RGBD-CNN saw +0.3%. This could possibly be improved with proper tuning of edge potential weights and unit standard deviations, and the potential exists to explore many other kinds of map-based semantic regularisation schemes. We leave these explorations to future work.

F. Run-time Performance

We benchmark the performance of our system on a random sample of 30 sequences from the NYUv2 test set. All tests were performed on an Intel Core i7-5820K 3.30GHz CPU and an NVIDIA Titan Black GPU. Our SLAM system requires 29.3ms on average to process each frame and update the map. For every frame we also update our stored surfel probability table to account for any surfels removed by the SLAM system. This process requires an additional 1.0ms. As discussed above, the other components in our system do not need to be applied for every frame. A forward pass of our CNN requires 51.2ms and our Bayesian update scheme requires a further 41.1ms. Our standard scheme performs

Office Reconstruction: 13 Class Semantic Segmentation

Method	books	ceiling	chair	floor	objects	painting	table	wall	window	class avg.	pixel avg.
RGBD	61.8	48.2	28.6	63.9	41.8	39.5	9.1	80.6	18.9	43.6	47.0
RGBD-SF	**66.4**	**78.7**	36.8	63.4	41.9	26.2	12.1	84.2	25.3	48.3	54.7
RGBD-SF-CRF	**66.4**	78.1	37.2	64.2	40.8	27.5	10.6	85.1	22.7	48.1	54.8
Eigen [2]	57.8	54.3	57.8	72.8	49.4	77.5	24.1	81.6	38.9	57.1	62.5
Eigen-SF	60.8	58.0	62.8	74.9	**53.3**	**80.3**	**24.6**	86.3	38.8	60.0	65.8
Eigen-SF-CRF	65.9	53.3	**65.1**	**76.8**	53.1	79.6	22.0	**87.7**	**41.4**	**60.5**	**67.0**

TABLE I: **Reconstruction dataset results**: SF denotes that the labels were produced by SemanticFusion, and the results where captured immediately if a frame with ground truth labelling was present. When no reconstruction is present for a pixel, we fall back to the predictions of the baseline single frame network. All accuracy evaluations were performed at 320 × 240 resolution.

NYUv2 Test Set: 13 Class Semantic Segmentation

Method	bed	books	ceiling	chair	floor	furniture	objects	painting	sofa	table	tv	wall	window	class avg.	pixel avg.
RGBD	62.5	**60.5**	35.0	51.7	92.1	54.5	**61.3**	**72.1**	34.7	26.1	32.4	86.5	53.5	55.6	62.0
RGBD-SF	61.7	58.5	43.4	58.4	92.6	63.7	59.1	66.4	47.3	34.0	33.9	86.0	60.5	58.9	67.5
RGBD-SF-CRF	62.0	58.4	43.3	59.5	**92.7**	**64.4**	58.3	65.8	**48.7**	34.3	34.3	86.3	**62.3**	59.2	67.9
Eigen [2]	42.3	49.1	73.1	72.4	85.7	60.8	46.5	57.3	38.9	42.1	68.5	85.5	55.8	59.9	66.5
Eigen-SF	47.8	50.8	79.0	73.3	90.5	62.8	46.7	64.5	45.8	**46.0**	70.7	88.5	55.2	63.2	69.3
Eigen-SF-CRF	48.3	51.5	79.0	**74.7**	90.8	63.5	46.9	63.6	46.5	45.9	**71.5**	**89.4**	55.6	**63.6**	**69.9**
Hermans *et al.* [8]	**68.4**	45.4	**83.4**	41.9	91.5	37.1	8.6	35.8	28.5	27.7	38.4	71.8	46.1	48.0	54.3

TABLE II: **NYUv2 test set results**: SF denotes that the labels were produced by SemanticFusion, and the results were captured immediately if a keyframe was present. When no reconstruction is present for a pixel, we fall back to the predictions of the baseline single frame network. Note that we calculated the accuracies of [2] using their publicly available implementation. Our results are not directly comparable with Hermans *et al.* [8] as we only evaluate on a subset of the test set, and their annotations are not available. However, we include their results for reference. Following previous work [8] we exclude pixels without a corresponding depth measurement. All accuracy evaluations were performed at 320 × 240 resolution.

this every 10 frames, resulting in an average frame-rate of 25.3Hz.

Our experimental CRF implementation was developed only for the CPU in C++, but the message passing algorithm adopted could lend itself to an optimised GPU implementation. The overhead of copying data from the GPU and performing inference on a single threaded CPU implementation is significant. Therefore on average, it takes 20.3s to perform 10 CRF iterations. In the evaluation above, we perform a CRF update once every 500 frames, but for online use it can be disabled entirely or applied once at the conclusion of a sequence.

V. CONCLUSIONS

Our results confirm the strong expectation that using a SLAM system to provide pixel-wise correspondences between frames allows the fusion of per-frame 2D segmentations into a coherent 3D semantic map. It is the first time that this has been demonstrated with a real-time, loop-closure capable approach suitable for interactive room scanning. Not only that, the incorporation of such a map led to a significant improvement in the corresponding 2D segmentation accuracy.

We exploited the flexibility of CNNs to improve the accuracy of a pretrained RGB network by incoporating an additional depth channel. In this work we opted for the simplest feasible solution to allow this new modality. Some recent work has explored other ways to incorporate depth information [9], but such an approach requires a duplication of the lower network parameters and was infeasible in our system due to GPU memory limitations. However, future research could also incorporate recent breakthroughs in CNN compression [10], which would not only enable improvements to the incorporation of other modalities, but also offer exciting new directions to enable real-time semantic segmentation on low memory and power mobile devices.

We believe that this is just the start of how knowledge from SLAM and machine-learned labelling can be brought together to enable truly powerful semantic and object-aware mapping. Our own reconstruction-focused dataset shows a much larger improvement in labelling accuracy via fusion than the NYU dataset with less varied trajectories, this underlines the importance of viewpoint variation. It also hints at the improvements that could be achieved with significantly longer trajectories, such as those of an autonomous robot in

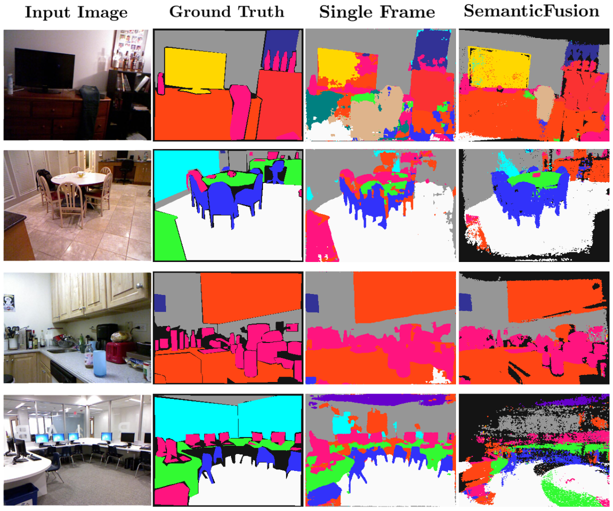

Fig. 6: **Qualitative NYUv2 test set results**: The results of SemanticFusion are using the RGBD-CNN with CRF after the completed trajectory, against the same networks single frame predictions. For evaluation, the black regions of SemanticFusion denoting areas without a reconstruction, are replaced with the baseline CNN predictions. Here we show only the semantic reconstruction result for clarity. The first two rows show instances where SemanticFusion has clearly improved the accuracy of the 2D annotations. The third row shows an example of a very rotational trajectory, where there is little difference as a result of fusing predictions. The final row shows an example where the trajectory was clearly not taken with reconstruction in mind, and the distant geometry leads to tracking and mapping problems even within our subset requiring 2Hz frame-rate. Cases such as this provide an advantage to the accuracy of the single frame network.

the field making direct use of the semantically annotated 3D map.

Going further, it is readily apparent, as demonstrated in a so far relatively simple manner in systems like SLAM++ [19] that not just should reconstruction be used to provide correspondence to help labelling, but that labelling/recognition can make reconstruction and SLAM much more accurate and efficient. A loop-closure capable surfel map as in ElasticFusion is highly suitable for applying operations such as class-specific smoothing (as in the extreme case of planar region recognition and fitting [18]), and this will be an interesting direction. More powerful still will be to interface with explicit object instance recognition and to replace elements of the surfel model directly with 3D object models once confidence reaches a suitable threshold.

References

[1] C. Couprie, C. Farabet, L. Najman, and Y. LeCun, "Indoor semantic segmentation using depth information," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2013.

[2] D. Eigen and R. Fergus, "Predicting Depth, Surface Normals and Semantic Labels with a Common Multi-Scale Convolutional Architecture," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2015.

[3] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *International Journal of Computer Vision (IJCV)*, no. 2, pp. 303–338, 2010.

[4] S. Gupta, R. Girshick, P. Arbelaez, and J. Malik, "Learning Rich Features from RGB-D Images for Object Detection and Segmentation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014.

[5] S. Gupta, P. A. Arbeláez, R. B. Girshick, and J. Malik, "Aligning 3D models to RGB-D images of cluttered scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[6] S. Gupta, P. A. Arbeláez, R. B. Girshick, and J. Malika, "Aligning 3D Models to RGB-D Images of Cluttered Scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[7] A. Handa, V. Pătrăucean, V. Badrinarayanan, S. Stent, and R. Cipolla, "SceneNet: Understanding Real World Indoor Scenes With Synthetic Data," *arXiv preprint arXiv:1511.07041*, 2015.

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

[9] J. Hoffman, S. Gupta, J. Leong, G. S., and T. Darrell, "Cross-Modal Adaptation for RGB-D Detection," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2016.

[10] F. N. Iandola, M. W. Moskewicz, K. Ashraf, S. Han, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <1mb model size," *CoRR*, 2016.

[11] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, 2014.

[12] H. S. Koppula, A. Anand, T. Joachims, and A. Saxena, "Semantic Labeling of 3D Point Clouds for Indoor Scenes," in *Neural Information Processing Systems (NIPS)*, 2011.

[13] P. Krähenbühl and V. Koltun, "Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials," in *Neural Information Processing Systems (NIPS)*, 2011.

[14] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," in *Neural Information Processing Systems (NIPS)*, 2012.

[15] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014, pp. 740–755.

[16] 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 (CVPR)*, 2015.

[17] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," *arXiv preprint arXiv:1505.04366*, 2015.

[18] R. F. Salas-Moreno, B. Glocker, P. H. J. Kelly, and A. J. Davison, "Dense Planar SLAM," in *Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR)*, 2014.

[19] R. F. Salas-Moreno, R. A. Newcombe, H. Strasdat, P. H. J. Kelly, and A. J. Davison, "SLAM++: Simultaneous Localisation and Mapping at the Level of Objects," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2013.178

[20] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from RGBD images," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2012.

[21] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

[22] S. Song, S. P. Lichtenberg, and J. Xiao, "SUN RGB-D: A RGB-D scene understanding benchmark suite," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 567–576.

[23] J. Stückler, B. Waldvogel, H. Schulz, and S. Behnke, "Multi-resolution surfel maps for efficient dense 3d modeling and tracking," *Journal of Real-Time Image Processing JRTIP*, vol. 10, no. 4, pp. 599–609, 2015.

[24] J. Valentin, S. Sengupta, J. Warrell, A. Shahrokni, and P. Torr, "Mesh Based Semantic Modelling for Indoor and Outdoor Scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[25] T. Whelan, S. Leutenegger, R. F. Salas-Moreno, B. Glocker, and A. J. Davison, "ElasticFusion: Dense SLAM without a pose graph," in *Proceedings of Robotics: Science and Systems (RSS)*, 2015.