Imperial College London
Department of Computing

Semantic Neural Representation for SLAM and Scene Understanding

Shuaifeng Zhi

21st September 2021

Supervised by Prof. Andrew Davison

Co-supervised by Dr. Stefan Leutenegger

Submitted in part fulfilment of the requirements for the degree of PhD in Computing and the Diploma of Imperial College London. This thesis is entirely my own work, and, except where otherwise indicated, describes my own research.

Abstract

Semantic simultaneous localisation and mapping (SLAM) has advanced remarkably over the past few years with the application of deep learning techniques. We are in the middle of a rapid progression from SLAM systems that simply reconstruct geometry towards understanding *what* is *where* in scenes. Semantically enriched maps will ultimately help intelligent robots improve the range and sophistication of their interactions with the world.

A key enabler of this capability is the *scene representation*, which defines how intelligent robots perceive, store and understand environmental attributes (i.e., physics, semantics, dynamics and interactions) from continuous observations. Geometric representation itself has long been a central research topic, and researchers have developed many ways to reconstruct the shape of scenes accurately and efficiently. We believe that representing geometry and semantics *jointly* is the right direction for scene models which are both optimally efficient and the most useful for actionable intelligence.

The work in this thesis concerns using tools from deep learning to enable joint representation of both geometry and semantics in SLAM systems and scene understanding. First we propose to learn code-based compact scene representations given large-scale datasets of images, depth maps and dense semantic labelling. The code representations of geometry and semantics are learned separately and enables joint optimisation at runtime, leading to a new multi-view label fusion approach and a preliminary dense monocular semantic mapping system. Secondly we explore a scene-specific implicit representation jointly encoding appearance, geometry and semantics. Without any external data, we show that the smoothness and consistency within our approach enable accurate semantic rendering given only sparse or noisy in-place annotation. Lastly, the real impact of this implicit representation in an super-efficient interactive scene labelling tool is shown.

Acknowledgements

Studying in Imperial College London as a PhD student has always been my dream since my first trip to UK as a visiting undergraduate in 2015. I am very grateful to many people who helped me along this special journey.

First and foremost, I would like to thank my supervisor Prof. Andrew Davison and co-supervisor Dr. Stefan Leutenegger for their selfless and remarkable guidance on my research and PhD life, as well as patient and unwavering help to get me through the most difficult time. This thesis is not possible without their enormous efforts of time and expertise. I am grateful for Andy's having me in this wonderful research lab with nice families. To me Andy is the best PhD supervisor I could ever thought of. His vision, attitude, persistence and passion towards truly impactful research and live demonstrations have reshaped my research value with a lifelong benefit in my future career.

I am thankful to Prof. Xiang Li for his commitment to supporting my overseas study. I would also like to thank his group for the generous support of my remote PhD study during the difficult time with global pandemic.

As a member of Dyson Robotics Lab, I appreciate all the time and communications with past and current lab members: Binbin Xu, Jan Czarnowski, Ronald Clark, Sajad Saeedi, Edward Johns, Wenbin Li, Robert Lukierski, Andrea Nicastro, Patrick Bardow, John McCormac, Stephen James, Charlie Houseago, Daniel Lenton, Zoe Landgraf, Dorian Hennings, Joseph Ortiz, Kentaro Wanda, Hide Matsuki; especially my collaborators Michael Bloesch, Tristan Laidlow, Edgar Sucar, Shikun Liu, Andre Mouton, Iain Haughton, lab manager Dr. Iosifina Pournara and everyone else. I also thank CSC and Dyson Technology Ltd. for funding and support my PhD research.

Last but not least, I would like to thank my dear friends sharing happiness and sorrow: Jingyuan Xia, Qi Yu, Junjie Huang, Tianrui Liu, Shengxi Li, Ke Wang, Cong Hu, Kai Mei and everyone else. I show my deepest respect and love to all my families in China. Love you all.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

Contents

1.1 Scene Representation and Understanding for Robotics

Humans can intelligently perceive and understand the surrounding physical world using past experience and various sensing observations. Among all types of sensing modalities including vision, auditory and tactility, etc., the visual sense acts as the main one and offers rich context information. These sparse and incomplete visual observations projected from the 3D world are processed with prior information by human brains to understand the geometry and semantics of the whole scene. The core of this human capability is *scene representation*, which incrementally processes and converts sensed data into a task-specified, compact and well described model [Eslami et al., 2018]. As argued by [Pearl, 2017]: "What humans possessed that other species lacked was a mental representation, a blue-print of their environment which they could manipulate at will to imagine alternative hypothetical environments for planning and learning."

Scene understanding involves many vision tasks such as object recognition, object detection, semantic segmentation, visual localisation and reconstruction, etc.. *Simultaneous localisation and mapping* (SLAM) is one of the most fundamental and critical research issues whose central problem is to enable robots to build a map of the 3D scene and concurrently localise itself within it, and is also closely related to the choice of underlying scene representation. To enrich embodied devices such as robots with increasing intelligence, a suitable scene representation of the environment is essential to fulfilling long-term robust scene perception, serving as a basic building block to understand the world and affect down-stream algorithm design and applications [Davison, 2018]. Robot arms in factories mainly perform in a pre-set environment but with high precision requirement; domestic robots, like vacuum cleaners, adopt representation which gradually evolves from randomly bouncing off the walls (representation-free mode) to actively localising

itself within rooms after constructing internal maps and path planning. Autonomous vehicles and drones incrementally process scanned images from dynamic environments under restrictions of computational power and efficiency. These intelligent embodied devices need to build and maintain representations of their environments which permit inference of geometric and semantic properties, such as the traversability of rooms or the way to grasp objects. Crucially, if this inference is to be scalable in terms of computing resources, these representations must be *efficient*; and if devices are to operate *robustly*, the employed representations must cope with all of the variation present in the real world. However, current real-time scene understanding systems are still a long way from the performance needed for truly ground-breaking applications [Cadena et al., 2016a, Davison, 2018].

An eventual token-like, composable scene understanding may finally give artificial systems the capability to reason about space and shape in the intuitive manner of humans [Wu et al., 2015]. The field of artificial intelligence has long sought to reproduce the process of scene representation. Bringing deep learning into traditional hand-designed estimation methods for SLAM has certainly led to big advances to representations which can capture both shape and semantics (i.e., augmented classical scene representation and neural scene representation) [Cadena et al., 2016b, Weerasekera et al., 2017], but so far these are problematic in various ways. In addition, although there have been tremendous efforts in exploring geometry representation in real-time SLAM systems and 3D perceptions, relatively less attention has been paid to the formulation of semantic representations.

For the reasons presented above, the work presented in this thesis focuses on using machine learning approaches to explore *semantic scene representation* in scene understanding and visual SLAM systems.

1.2 Scene Representations in visual SLAM

Scene representation in visual SLAM (vSLAM) systems has gradually progressed from sparse feature point sets [Davison, 2003, Mur-Artal and Tardós, 2017] to dense geometric 3D maps (e.g. point clouds, meshes and voxels shown in Figure 1.1) [Newcombe et al., 2011a, Whelan et al., 2016, Nießner et al., 2013, Engel et al., 2014, Dai et al., 2017b] and more recently, to neural representations [Bloesch et al., 2018, Sucar et al., 2021], increasingly involving semantics [Salas-Moreno et al., 2013, McCormac et al., 2017a, Sünderhauf et al., 2017, Narita et al., 2019, Zhi et al., 2019, Zhi et al., 2021a].

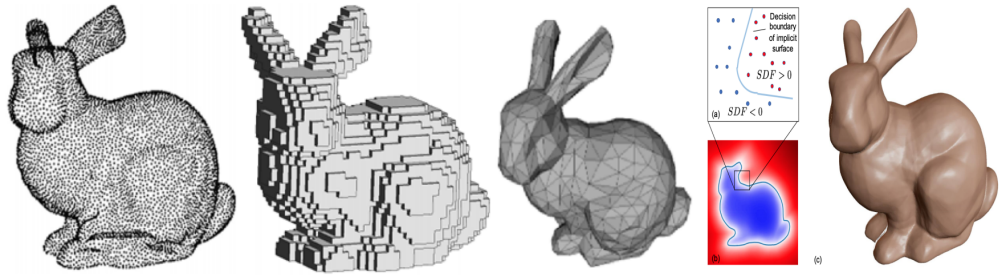


Figure 1.1: Example of classical geometric representations. From left to right are point clouds, volumetric grids (voxels), meshes and signed distance field (SDF).

Correspondingly, the taxonomy of vSLAM systems from the adopted representation can hence be divided into sparse SLAM, (semi-)dense SLAM and recent neural representation-based SLAM.

Geometric SLAM, from Sparse to Dense

Classical sparse SLAM methods track and reconstruct a set of sparse discriminative 2D/3D landmarks (e.g., corners, lines, arcs) within the scene, which can be referred to as a feature-based representation. The selected landmarks are usually called *keypoints* which are geometrically robust and consistent under visual observations from various viewpoints. Keypoints are detected based on low-level images statistics such as high image gradients. Therefore, sparse SLAM systems

assume no prior knowledge of the geometry of the observed scene, and the geometric parameters of all the keypoints are considered conditionally independent given the camera state. The most commonly used keypoint detection in vSLAM systems are Harris Corner Detection [Harris and Stephens, 1988], FAST [Rosten and Drummond, 2006], GFTT [Shi and Tomasi, 1994], etc.

In order to identify keypoints under different camera motions, correspondences between images are established by finding keypoint pairs sharing most similarities via *descriptors*. The descriptors aim to discriminately represent the salient and distinguishable image regions around keypoint locations, of which repeatability, distinctiveness, efficiency and locality are the main characteristics. Most popular local descriptors are SIFT [Lowe, 2004], SURF [Bay et al., 2006], BRIEF [Calonder et al., 2010] ,BRISK [Leutenegger et al., 2011] and ORB [Rublee et al., 2011, Mur-Artal and Tardós, 2014], etc. There have been years of continuous efforts in the area of image processing community on extraction of keypoints, descriptors as well as correspondences, ranging from above-mentioned hand-crafted ones to latest learning-based approaches [Schmidt et al., 2017, DeTone et al., 2018].

Although accurate and efficient tracking and localisation can be enabled by sparse representation, a small set of landmarks is incapable of providing rich information about object shapes and surfaces within scenes and hence is less useful to applications where a dense 3D scene representation is preferred and required. For example, path planning and navigation of mobile robots require a good sense of dense occupancy within scenes to determine if it is possible to safely traverse along certain trajectories to reach a destination; and human-robot interaction with environments needs a detailed dense model of a scene to enable lively visualisation and feedback.

Specifically, dense SLAM attempts to utilise and reconstruct 3D information for all raw pixels from image space with the help of geometric prior knowledge,

typically smoothness over a local image region. Semi-dense representations lie between sparse and dense ones, where a well-constrained subset of pixels are reconstructed, and this is usually discussed together with dense representation [Engel et al., 2014, Engel et al., 2017, Engel, 2017]. Leveraging the advent of modern commodity Graphic Processing Units (GPU), dense representation benefits from its parallel processing power and hence is able to process all raw image information to achieve real-time performance. DTAM [Newcombe et al., 2011b] is one of the first dense SLAM systems operating with a single monocular camera [Pradeep et al., 2013, Pizzoli et al., 2014]. In DTAM the dense geometry of keyframes is estimated by accumulating photometric information from overlapping reference images with a small baseline into a perspective cost volume which is then solved using a variational approach. Robust camera tracking is achieved by dense image alignment. RGB-D based dense mapping appeared after low-cost commercial depth sensors came out [Microsoft Corp, 2010]. KinectFusion [Newcombe et al., 2011a] was one of earliest and most influential attempts in this direction. A global volumetric implicit map using signed distance filed (SDF) is maintained and camera tracking is aligned by dense registration. [Nießner et al., 2013] and [Kahler et al., 2015] improved the memory and speed efficiency by adopting a spatial hashing strategy. ElasticFusion [Whelan et al., 2015] used a surfel based dense representation and allowed for flexible map deformation during mapping and loop closure. BundleFusion [Dai et al., 2017b] addressed the fragile temporal camera tracking, model inconsistency and scalability issue of existing dense SLAM systems for large-scale scenes by estimating a global set of camera poses given complete history with an efficient hierarchical approach.

Semantic Representation

Impressive progress in geometric SLAM has been achieved [Cadena et al., 2016a] and dense geometric representation at different scales with fairly accurate detail can

be reconstructed in real-time with commodity cameras. Nevertheless, there are still numerous relationships and semantic concepts in a scene, which are beyond low-level primitives such as points, lines and raw image patches. More general scene understanding is required, and we need to advance beyond pure geometric representation towards a semantic-aware one so that robots have the capability to fulfil a range of complicated tasks, e.g., bringing a cup of tea to a user from the kitchen to the living room, or tidying and rearranging a room, or robot-human assistance and interaction, etc. [Batra et al., 2020].

Semantic mapping emerged to build a semantic map of scenes containing and organizing detailed semantic concepts. Typically, dense semantic labels are painted on and attached to geometric dense 3D maps. [Hermans et al., 2014] used a randomized decision forest followed by a dense conditional random field (CRF) to predict and refine semantic segmentation results on 3D a point cloud reconstruction. Recently with the rise of deep learning [Krizhevsky et al., 2012], convolutional neural networks (CNN) have demonstrated superior performance in various computer vision tasks powered by modern GPUs and large training corpuses, and have become a popular choice of semantic classifier in the latest semantic mapping systems [Long et al., 2015, Hazirbas et al., 2016, Lin et al., 2017, Chen et al., 2018a]. SemanticFusion [McCormac et al., 2017a], shown in Figure 1.2, is one of the earliest systems to combine a dense geometric SLAM system ElasticFusion with semantic segmentation from CNNs [Tateno et al., 2017]. 2D semantic predictions are re-projected to assign multi-class semantic probabilities to the associated surfels, which are further refined using Bayesian multi-view label fusion. [Nakajima et al., 2018] further improved efficiency and performance with the help of geometric segmentation and a light-weight segmentation CNN. Other work [Zhao et al., 2017a] and [Narita et al., 2019] tackled 3D material reconstruction and panoptic semantic reconstruction in similar pipelines.

An alternative to dense semantic representation, object-level representation is

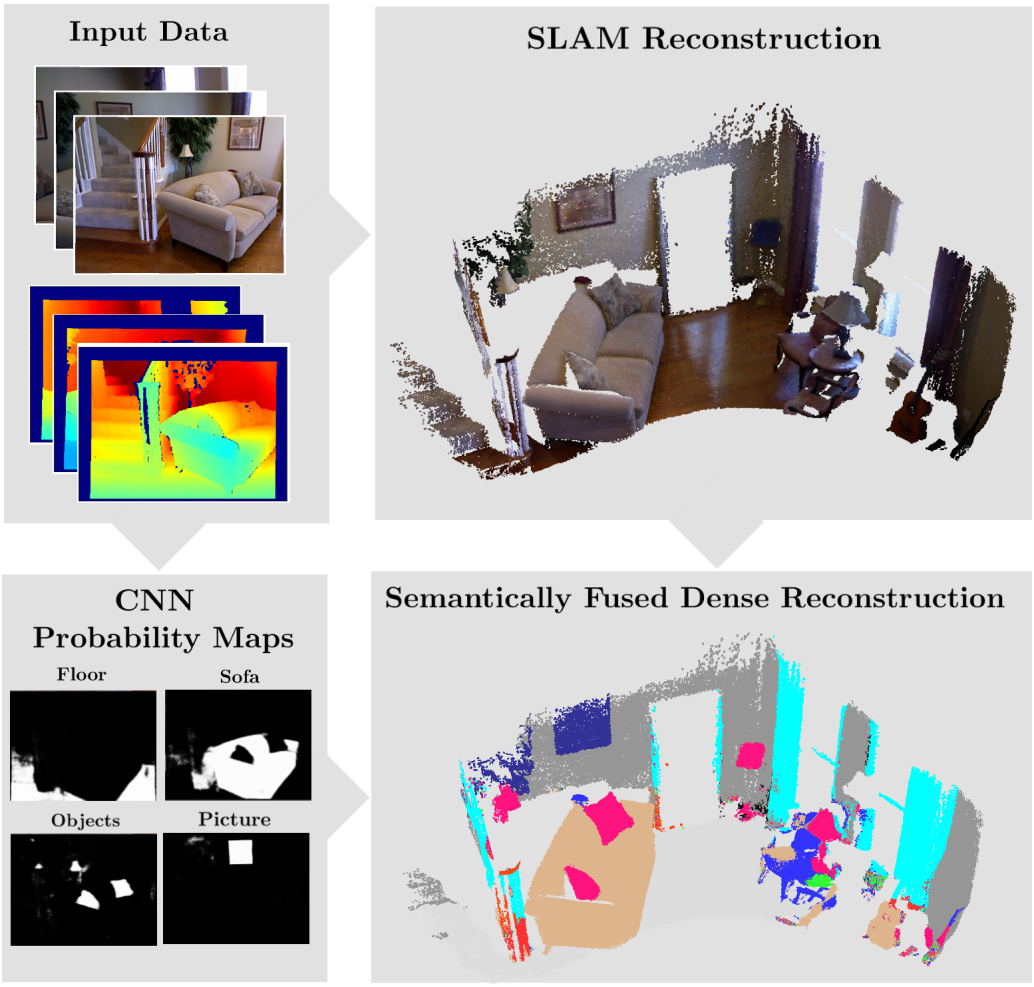


Figure 1.2: Pipeline of the SemanticFusion [McCormac et al., 2017a] system combining a dense geometric SLAM and CNN semantic predictions.

another semantic representation offering the potential of object reasoning. The SLAM++ system [Salas-Moreno et al., 2013] proposed to include detected 3D objects (e.g., chairs, desks) and other structural priors into pose graph optimisation, enabling robust relocalisation and AR applications. However, the object database in SLAM++ is predefined and not scalable to practical complex scenes. Fusion++ [McCormac et al., 2018] and work from Sunderhauf *et al*. [Sünderhauf et al., 2017] alleviate this limitation by leveraging on supervised deep neural network for object detection (e.g., Mask R-CNN [He et al., 2017a], YOLO [Redmon et al., 2016])

and construct an object-centric 3D map. In these works, scene graphs are used as the general underlying representation, which organise all entities of a scene in a graph where objects and their inter-relationships are modelled as nodes and edges, respectively [Wald et al., 2020, Wu et al., 2021]. We envision more applications combing online SLAM systems and scene graph representations to explore more general reasoning between scene entities, which opens a more complicated and task-specific decision about which hierarchical and organisation of semantic concepts are important.

Neural Scene Representation in vSLAM

The computer vision and SLAM communities have long sought to explore the learned scene representations to overcome several problems in the human-designed ones. Point clouds are raw 3D data which can be easily acquired from LiDAR and depth sensors nowadays, advancing geometric deep learning like PointNet [Qi et al., 2016, Qi et al., 2017a, Qi et al., 2017b, Dai and Nießner, 2019, Dai et al., 2021]. However, the lack of topological information makes it less suitable for high-level tasks. In contrast, volumetric representation benefits from its clear typology and regular structure for parallel processing, while sufferring from limited computational and memory efficiency when representing detailed shape due to discretisation. As for meshes, adaptively deforming from fixed templates to target mesh topologies is a nontrivial problem.

As envisioned by "Spatial AI" as the enabling technology for next generation smart robots, coined by Davison [Davison, 2018], a task-focused and persistent scene representation, with both learned and designed elements, should be built which is close to metric 3D geometry, at least locally, and is human understandable. This demands for a learned or hybrid representation which works as efficiently as a sparse one but has the ability to offer dense geometry prediction and semantic reasoning.

2D representation learning has been much explored lately thanks to publicly available large-scale image data. Image-level or pixel-level dense feature representation can be learned in a supervised or self-supervised manner, and further leveraged by down-stream 2D tasks such as 2D object detection, segmentation and even generative tasks with promising performance [Doersch et al., 2015, Zhang et al., 2016, Donahue and Simonyan, 2019, Hou et al., 2021]. Seminal work including GQN [Eslami et al., 2018] and CodeSLAM [Bloesch et al., 2018] proposed to use view-based code representations learned by a variational auto-encoder (VAE) under a multi-view set-up to achieve inference in 3D space. In GQN, view-synthesis at novel viewpoints can be retrieved from the merged representation of a few reference views, but is limited to relatively simple 3D scenes. CodeSLAM also uses a VAE but encodes dense geometry into a compact and optimisable latent space, which allows for joint optimisation of dense geometry and camera poses in a real-time SLAM system (Figure 1.3). SceneCode, which will be discussed in Chapter 3, also encodes dense semantic labelling into compact latent codes and shows the benefits of code-representation in multi-view semantic label fusion. However, although trained with posed images or depth maps, the code representation is still view-based and lacks true awareness of the 3D geometry.

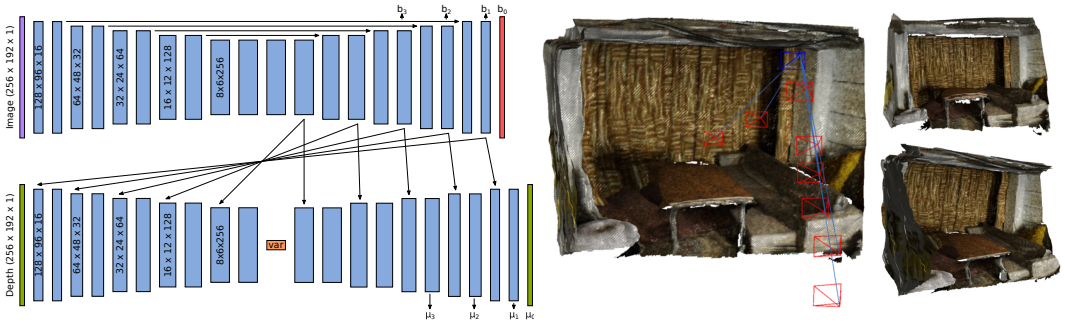


Figure 1.3: Network architecture (left) and qualitative result of a 9-view dense reconstruction using code optimisation (right) in CodeSLAM [Bloesch et al., 2018].

When it comes to 3D-aware neural representation, several methods have tried to learn a 3D-structured latent model [Nguyen-Phuoc et al., 2019, Sitzmann et al.,

2019a, Park et al., 2020], but still suffer from the limitations coming from the adopted data structure; e.g., 3D latent spaces produced by neural networks using computationally and memory expensive 3D convolutions are finally limited by the resolution it can efficiently represent. Instead of directly augmenting classical representation with powerful deep features, at the other end of the spectrum are *implicit* neural representations proposed recently. Pioneering methods including Scene Representation Networks (SRN) [Sitzmann et al., 2019b], DeepSDF [Park et al., 2019], Occupancy Networks [Mescheder et al., 2019] and Neural Radiance Fields (NeRF) [Mildenhall et al., 2020] propose to implicitly represent 3D scenes using the weights of neural networks which can predict the occupancy and colour information at a given 3D query position. iMAP [Sucar et al., 2021] proposed to use a NeRF-like representation with continual learning to establish a real-time dense tracking and mapping system. Though this is an encouraging advance on the geometric representation, neural semantic representation is under-explored as semantic information is usually formulated as an extra property and attached to underlying geometry. We believe that there is much to gain by focusing on semantic representations which potentially enable joint inference and mutual benefits between semantics and geometry.

In this thesis, we concentrate on formulating joint geometric and semantic scene representations for scene understanding and semantic SLAM. Specifically, the representation should be able to infer rich low-level and high-level information of scenes to support semantic-level tasks, e.g., motion and structure estimation, and identifying and interacting with scene entities [Davison, 2018]. In addition, the representation should be efficiently learned and updated in an incremental manner, ideally in real-time, so that new observations can be incorporated to improve its priors and effectiveness. Last but not least, the representation should also have a sense of uncertainty so that instead of being over-confident to outlier samples, users have access to the fidelity of the predictions [Gal, 2016, Kendall and Gal,

2017]. Inspired by these goals, the work discussed in Chapter 3 focuses on learning a compact code-based semantic representation to tackle multi-view label fusion. The prior information learned from a training phase enables semantic refinement at inference time; Chapter 4 alleviates the need for costly external supervision and the problem of the generalisation gap by adopting a scene-specific implicit representation. The joint encoding of appearance, geometry and semantics allows for dense semantic labelling from sparse or noisy in-place supervision. Finally presented in Chapter 5, a real-time interactive scene labelling and understanding tool is discussed based on Semantic-NeRF from Chapter 4.

1.3 Contributions

The main results presented in this thesis have been published in three different research papers. The full list of publications done in conjunction with the thesis, as well as the video material that provides visualisation of the algorithms developed, is given in this Section 1.4. The motivation and contribution from each paper is briefly discussed below.

1.3.1 Paper I: SceneCode: Code-based Semantic Representation

Research Question: Can we design a compact and optimisable semantic representation for monocular dense semantic mapping?

While there has been much work on the correct formulation for geometrical estimation, semantic representation is less explored and state-of-the-art semantic SLAM systems usually rely on simple semantic representations which store and update independent label estimates for each surface element (depth pixels, surfels, or voxels). Consequently, spatial correlation is discarded and fused label maps are

incoherent and noisy, which can be partly addressed by expensive post-processing steps such as CRF.

Inspired by CodeSLAM [Bloesch et al., 2018], we propose SceneCode which represents dense semantic labels as compact and optimisable latent codes by training a variational auto-encoder that is conditioned on a colour image. Using this learned latent space, we can tackle semantic label fusion by jointly optimising the low-dimenional codes associated with each of a set of overlapping images, producing consistent fused label maps which preserve spatial correlation. This is substantially different from the widely adopted element-wise fusion approach. We also show how this approach can be used within a monocular keyframe based semantic mapping system where a similar code approach is used for geometry. The probabilistic formulation allows a flexible formulation where we can jointly estimate motion, geometry and semantics in a unified optimisation.

SceneCode will be described in detail in Chapter 3.

1.3.2 Paper II: Semantic-NeRF: Implicit Semantic Representation based on Neural Radiance Field

Research Question: Can we find an efficient semantic representation that does not scale with scene resolution and does not require external annotation?

Most existing semantic representation learning relies on expensive annotated datasets and generalises poorly to out-of-distribution unseen samples. Recent implicit neural reconstruction techniques are appealing as they do not require prior training data, but the same fully self-supervised approach is not possible for semantics because labels are human-defined properties.

Motivated by the fact that semantic labelling is highly correlated with geometry and radiance reconstruction, as scene entities with similar shape and appearance

are more likely to come from similar classes, we propose Semantic-NeRF, which extends neural radiance fields (NeRF) to jointly encode semantics with appearance and geometry, so that complete and accurate 2D semantic labels can be achieved using a small amount of in-place annotation specific to the scene. The intrinsic multi-view consistency and smoothness of NeRF benefits semantics by enabling sparse labels to efficiently propagate. We show the benefit of efficient learning Semantic-NeRF when labels are either sparse or very noisy in room-scale scenes. We demonstrate its advantageous properties in various interesting applications such as an efficient scene labelling tool, novel semantic view synthesis, label denoising, super-resolution, label interpolation and multi-view semantic label fusion in visual semantic mapping systems.

Semantic-NeRF will be discussed in detail in Chapter 4.

1.3.3 Paper III: iLabel: Interactive Scene Understanding in Real-time using Implicit Scene Representation

Research Question: Can we leverage implicit scene representation to design a real-time interactive scene understanding system?

We present iLabel: the first *online*, *interactive* 3D scene understanding system based on neural implicit scene representations. A user annotates semantic properties in a scene via clicks, while simultaneously scanning and mapping it with a handheld RGB-D sensor. The scene model is updated and visualised in real-time, allowing the user to focus interactions as needed to achieve ultra-efficient labelling. iLabel's underlying model is an MLP trained from scratch in real-time to learn a joint implicit encoding of geometry, appearance and semantics in 3D. The internal smoothness and consistency of the representation of shape and appearance is inherited by the semantic channel, allowing it to make accurate dense predictions from very sparse annotations, and regularly auto-segment objects and other re-

gions. We show that a room or similar scene can be highly accurately labelled into 10+ semantic categories with only a few tens of clicks, where these categories are either known in advance or defined in an interactive 'open-set' manner by the user. The quantitative labelling accuracy scales powerfully with the number of clicks, and rapidly surpasses the accuracy of standard pre-trained semantic segmentation methods. We also demonstrate a variant which uses a binary tree for hierarchical semantic labelling. iLabel has the flexibility to be used in a variety of scenarios: from an interactive, user-friendly data annotation or scene labelling tool to a core perception module enabling intelligent robots to operate in open-set environments.

In this work Edgar Sucar and I contributed equally, where I focused on the implementation of automatic query generation and quantitative experiments, and partly contributed to the initial idea of the iLabel system.

iLabel will be discussed in detail in Chapter 5.

1.4 Publications

The work described in this thesis resulted in the following publications:

- Zhi, S., Bloesch, M., Leutenegger, S., and Davison A. (2019). **Scenecode: Monocular dense semantic reconstruction using learned encoded scene representations**. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [Zhi et al., 2019].

- Zhi, S., Laidlow, L., Leutenegger, S., and Davison A. (2021). **In-Place Scene Labelling and Understanding with Implicit Scene Representation**. *In Proceedings of the International Conference on Computer Vision (ICCV)*. [Zhi et al., 2021a].

- Zhi, S. *, Sucar, E.*, Mouton, A., Haughton, I., Laidlow, T., Davison, A. (2021). **iLabel: Interactive Implicit Scene Labelling and Learning in Real-Time**. *Under submission*. [Zhi et al., 2021b]. (* indicates equal contribution to the paper.)

While not described directly, the following publication was done in conjunction with this thesis:

- Liu, S., Zhi, S., Johns, E., Davison, A. (2021). **Bootstrapping Semantic Segmentation with Regional Contrast**. *Arxiv*. [Liu et al., 2021].

The following material provides visualisation of some algorithms developed in this thesis:

- SceneCode: supplementary video at https://youtu.be/MCgbgW3WA1M.
- Semantic-NeRF:
  - Project page with various qualitative results at https://shuaifengzhi.com/Semantic-NeRF/.
  - Supplementary video at https://youtu.be/FpShWO7LVbM.

1.5 Thesis structure

The remainder of this thesis is structured as follows:

Chapter 2 introduces basic notations and the necessary concepts including the variational auto-encoder (VAE) and neural implicit representations used in this work.

Chapter 3 introduces a compact and optimisable code representation for dense semantic labelling, based on which we propose a new code based multi-view label fusion approach leading to coherent and smooth fusion results. We also show the benefits of code representation in a monocular dense semantic mapping system.

Chapter 4 discusses an efficient scene-specific semantic representation built upon neural radiance field, i.e., Semantic-NeRF. Motivated by the internal coherence and multi-view consistency of Semantic-NeRF, we show that accurate dense semantic rendering can be achieved with only sparse or noisy semantic supervision. Various qualitative and quantitative experiments on Replica and ScanNet datasets validate its potential applications.

Chapter 5 proposes the first real-time interactive 3D scene labelling and understanding system based on deep neural implicit representation. We demonstrate and discuss its capability of ultra-efficient scene labelling in a variety of challenging real-world scenes.

Chapter 6 concludes this thesis with a summary of the results presented and suggestions for future work.

Chapter 2

Preliminaries

Contents

In this chapter, we review the fundamental concepts and components for the algorithms presented ahead. We start with mathematical notation and continue with the pin-hole camera model in Section 2.1.5. Then we present a general in-

troduction to feedforward deep neural networks including the Multi Layer Perceptron (MLP) and Convolutional Neural Networks (CNN). A brief mathematical derivation of Variational Auto-Encoder is shown in Section 2.3.2, followed by a discussion of deep semantic segmentation approaches. The final section introduces implicit neural representations, especially neural radiance fields (NeRF) originally proposed for photorealistic novel view synthesis.

2.1 Notation

In this work we use the following notation:

2.1.1 General notation

a/A Standard mathematical symbol denotes a scalar apart from common capital exceptions.

$\mathbf{a}$ A bold lower-case symbol denotes an m-dimensional column vector with a_i the i^{th} element as:

$$\mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix}, \mathbf{a}^\top = [a_1, a_2, \ldots, a_m]. \tag{2.1}$$

We use $\mathbf{a}_{i:j}$ to denote the vector consisting of the elements of $\mathbf{a}$ with indices in the $[i, j]$ range.

$\mathbf{A}$ A bold capital symbol denotes an $m \times n$ matrix with $a_{i,j}$ the element at the

i^{th} row and j^{th} column:

$$\mathbf{A} = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ldots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{bmatrix}. \tag{2.2}$$

$\mathbf{I}$ The identity matrix, optionally with dimensions as subscript.

$\mathbf{0}$ The zero matrix, optionally with dimensions as subscript.

$[\cdot]^\times$ The cross-product matrix that produces a skew symmetric matrix from a 3D vector such that $\mathbf{a} \times \mathbf{b} = [\mathbf{a}]^\times \mathbf{b}$. Given the vector $\mathbf{a} = [a_x, a_y, a_z]^\top$, $[\mathbf{a}]^\times$ can be computed by:

$$[\mathbf{a}]^\times = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix}. \tag{2.3}$$

$\mathscr{A}$ A calligraphic capital symbol denotes a set.

2.1.2 Probability

$p(\mathbf{x})$ The probability density function of random variable $\mathbf{x}$.

$p(\mathbf{x} \mid \mathbf{y})$ The conditional probability density function of random variable $\mathbf{x}$ given random variable $\mathbf{y}$.

2.1.3 Spaces and manifolds

$\mathbb{R}$ The set of real numbers.

$\mathbb{R}^+$ The set of positive real numbers.

$\mathbb{R}^m$ The vector space of real m-dimensional vectors.

$\mathbb{R}^{m\times n}$ The vector space of real $m \times n$-dimensional matrices.

$\mathbb{Z}$ The set of integers.

$\mathcal{N}(\mu, \Sigma)$ The Normal distribution with mean μ and covariance Σ.

S^3 The 3-sphere group.

$SO(3)$ The group of 3D rotations.

$SE(3)$ The group of 3D rigid transformations.

$\boxplus$ The "box-plus" operator that applies a perturbation expressed in a tangent space to a manifold state.

$\boxminus$ The "box-minus" operator that expresses the difference of two manifold states in the tangent space.

2.1.4 Frames and transformations

$\underrightarrow{\mathcal{F}}_A$ The Cartesian coordinate frame A in $\mathbb{R}^3$.

${}_A\mathbf{v}$ A vector $\mathbf{v}$ expressed in coordinate frame $\underrightarrow{\mathcal{F}}_A$.

$\mathbf{C}_{AB}$ The rotation matrix that transforms the vector ${}_B\mathbf{v}$ expressed in coordinate frame B to one expressed in $\underrightarrow{\mathcal{F}}_A$ as: ${}_A\mathbf{v} = \mathbf{C}_{AB}\,{}_B\mathbf{v}$. The inverse rotation $\mathbf{C}_{BA}$ can be computed as: $\mathbf{C}_{BA} = \mathbf{C}_{AB}^{-1} = \mathbf{C}_{AB}^{\top}$.

$\boldsymbol{T}_{AB}$ The homogeneous transformation matrix that transforms the homogeneous vector from coordinates B to A.

2.1.5 Camera Models

f_x The horizontal focal length of a camera measured in pixels.

f_y The vertical focal length of a camera measured in pixels.

c_x The horizontal offset of a camera centre measured in pixels.

c_y The vertical offset of a camera centre measured in pixels.

$\mathbf{K}$ The intrinsic camera matrix:

$$\mathbf{K} = \begin{bmatrix} f_x & c_x & 0 \\ f_y & 0 & c_y \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.4}$$

$\pi(\cdot)$ The perspective projection function.

$\pi^{-1}(\cdot)$ The inverse perspective projection function.

2.2 Camera Models

The process of projecting a 3D point from world space to a 2D point in image plane can be described by a geometric camera model. Though there are various types of camera models available, the simplest and most commonly used in 3D vision is the *pinhole camera model*, describing the imaging process via an infinitesimally small, single point aperture. At the same time, due to the presence of the camera lens, *distortion* is generated during practical projection. In this thesis, we assume an *ideal pinhole camera model* is used (Figure 2.1) and therefore all distortions are assumed to have been corrected.

The pinhole camera model can be mathematically described by intrinsic matrix $\mathbf{K}$ as in Equation 2.4, where f_x, f_y are the horizontal and vertical focal length, and c_x, c_y are the horizontal and vertical camera center offset, respectively, all in pixels. It is usually assumed that these intrinsic parameters are fixed after manufacturing and will not change during usage.

As shown in Figure 2.1, a 3D point ${}_C\mathbf{p}$ represented in camera coordinate frame $\underrightarrow{\mathcal{F}}_C$ can be projected to a 2D coordinate $\mathbf{u} = [u, v]$ in the pixel plane via compact

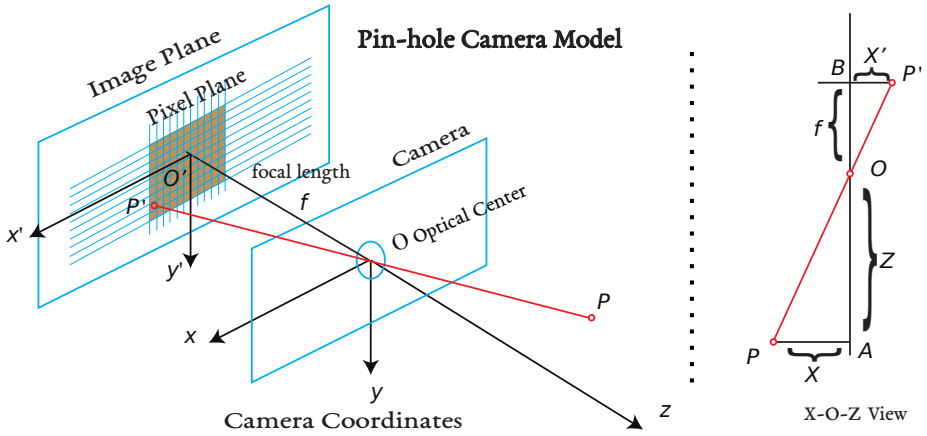


Figure 2.1: Illustration of the perspective projection process using a pin-hole camera model(left) and its X-O-Z view (right). Modified from https://github.com/gaoxiang12/slambook-en/blob/master/resources/cameraModel/cameraModel.pdf with GNU general public liscense.

matrix multiplication with homogeneous coordinates:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \frac{1}{{}_C p_z} \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} {}_C p_x \\ {}_C p_y \\ {}_C p_z \end{bmatrix} \triangleq \frac{1}{{}_C p_z} \mathbf{K}_C \mathbf{p}. \tag{2.5}$$

.

From the above equation, we can also observe that: (1) once the depth of a certain pixel is given, we can also invert the projection process, i.e., unproject a 2D pixel to its corresponding 3D point in the camera coordinate frame. (2) absolute depth information, i.e. ${}_C p_z$, is lost during projection and hence it is not possible to recover depth information using a single monocular camera.

We have shown how to describe a single monocular camera using the pin-hole camera model. In practice, more types of camera such as stereo cameras, RGB-D cameras or LIDAR are often used to acquire depth information to support challenging vision tasks. In this thesis, Chapters 3 and 4 use a monocular set-up, while Chapter 5 utilises an RGB-D camera which actively measures per-pixel depth.

2.3 Deep Neural Networks

Deep neural networks (DNN) are a broad and general type of machine learning algorithm, and have impressively demonstrated their promise in computer vision in both research and industrial applications. Many challenging vision problems including scene understanding and representation learning have been greatly advanced by deep learning techniques [Eigen and Fergus, 2015, Ronneberger et al., 2015, Long et al., 2015]. Specifically, a deep neural network is a parametric computational graph made of 'neurons' inspired by biological neural networks to learn a mapping $f(\cdot)$ from input domain $\mathscr{X}$ to output domain $\mathscr{Y}$ (i.e., labels, representation). The weights of each neuron are updated with the *back propagation* technique to minimise a pre-designed objective loss function over mini-batches of training data samples, iteratively. After prior information is learned from the training process, it is expected to work well on unseen testing data sampled from a similar distribution.

Though deep neural networks are not the only way to solve these tasks, and have several unsolved issues such as lack of interpretability and dependence on a huge amount of (labelled) data in general, we believe that the encouraging achievements of deep learning are a firm step towards Spatial AI and we use neural representations broadly in this thesis.

We make use of feedforward deep neural networks in this thesis, whose computational graphs are a-cyclic one. Multi Layer Perceptron or Fully Connected Network and the Convolutional Neural Networks are two main categories of feedforward deep neural networks.

2.3.1 Feedforward Neural Networks

Multi Layer Perceptron

A multi layer perceptron is the 'vanilla' deep neural network composed of a series of fully connected layers. In each layer a neuron is connected to all neurons in the previous adjacent layer, whose inputs are added up with trainable weights and passed through a non-linear activation function, such as a sigmoid, Tanh, ReLU or ReLU variations. There are no connections between neurons within the same layer.

Theoretically the representative power of an MLP is very strong [Hornik, 1989] and it has little inductive bias by design, which is the assumptions a machine learning algorithm makes about encountered data. Therefore, despite being a powerful universal approximator, the unconstrained and flexible MLP tends to overfit training data and is less efficient at tackling 2D vision tasks where the network size of MLP scales proportionally with the number of pixels. This leads to the more efficient design of convolutional neural networks which incorporate stronger inductive biases suiting natural images for 2D and 3D vision tasks, as discussed in the next subsection. However, a fully connected layer still acts a key component in several popular architectures [Simonyan and Zisserman, 2015, Qi et al., 2017a, Park et al., 2019] and recently we have witnessed a renaissance of MLPs in various vision tasks, thanks to the injection of inductive biases from overall pipeline design (e.g., multi-view consistency used in implicit representation learning) [Sitzmann et al., 2019b, Mildenhall et al., 2020] or to the huge amount of training dataset (e.g., vision Transformer) [Vaswani et al., 2017, Dosovitskiy et al., 2021].

In Chapter 4 and Chapter 5, we will show how to use a coordinate-based MLP to learn implicit semantic representations of 3D scenes, in which the input to the MLP is queried 3D (x, y, z) positions and predict associated occupancy value.

Convolutional Neural Networks

A convolutional neural network is similar to an MLP in overall design. However, a CNN explicitly assumes the input data to follow a regular grid pattern, typically 2D images or 3D voxel grids. Recently there are also graph neural networks extending the convolution to graph structures.

A convolutional neural network processes input images via a sequence of layers: convolutional layers, pooling layers and optional fully connected layers. Convolutional layers are the core building blocks of CNN where *local connectivity* and *parameter sharing* are the two main properties enabling efficiently processing of visual data. Local connectivity means the spatial extent (height and width) of the connectivity of each neuron, i.e. *receptive field*, is limited to a local region of input; while the channel-wise extent is equal to its input. This design solves the network scaling issue of network size with input dimension. Parameter sharing is another important design choice to control the total number of trainable parameters. The common assumption behind it is that salient features detected at a certain position should also be detectable at other spatial positions as well. Pooling layers are periodically inserted between consecutive convolutional layers to reduce the dimensions of input feature maps to the following network layers. Average pooling and maximum pooling are the two most commonly used pooling types. A fully connected layer seen in popular CNN architectures is placed at the end for feature fusion and classification. Mathematically, convolution can be expressed as matrix multiplication and can efficiently leverage modern computation libraries and GPUs. The trainable weights in a CNN are composed of convolutional kernels and fully connected neurons, and can be optimised using backpropagation from mini-batch stochastic gradient descent (SGD). Although only a subset of deep learning research, numerous CNNs have been proposed and developed with task specific custom architectures and layers to address vision tasks ranging from image understanding including classification, object detection, and semantic

prediction to geometry tasks such as depth prediction, correspondence matching and pose estimation.

In addition to the superior performance of CNNs as a feature extraction via supervised or self-supervised training, recent research also reveals that the strong inductive bias of CNN can act as an effective deep natural image prior [Ulyanov et al., 2018, Chakrabarty and Maji, 2019, Cheng et al., 2019, Gandelsman et al., 2019], with promising performance on image inverse problems. Deep internal learning recently coined by Shocher and Irani [Shocher et al., 2018] leverages the internal redundancy discovered by a CNN in a totally unsupervised way. It has been shown that without any prior knowledge of data, the self-similarity within a single image is enough to train a CNN for a variety of tasks including super resolution, image segmentation, transparency layer separation, or image manipulation [Shocher et al., 2019, Bell-Kligler et al., 2019, Shaham et al., 2019].

2.3.2 Variational Auto-Encoder

The variational auto-encoder (VAE) is a powerful deep generative model and proposed by Kingma *et al*. [Kingma and Welling, 2014] for efficient learning and inference of directed probabilistic models (see Figure 2.2). The name 'auto-encoder' comes from its derived cost function, a part of which resembles the cost function of a traditional auto-encoder network (AE). There are striking differences between variational auto-encoders and classical auto-encoders. Specifically, the VAE is a generative model while AE is usually a deterministic model.

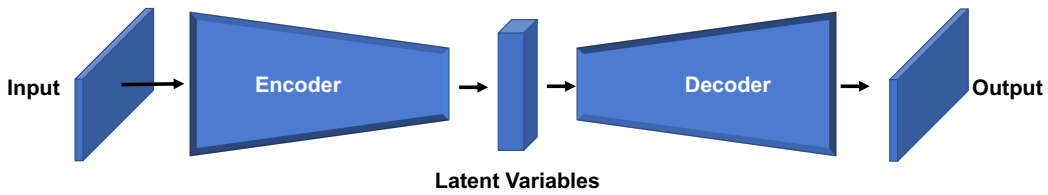


Figure 2.2: Illustration of a vanilla variational auto-encoder.

Let us denote the dataset we are interested in as $\mathbf{X} = \{\mathbf{x}^{(i)}\}_{i=1}^{N}$, containing N i.i.d. (independent and identically distributed) data samples of the variable $\mathbf{x}$ ($\mathbf{x}$ can be either discrete or continuous), under the assumption that all samples are generated from a certain random process with an hidden latent variable $\mathbf{z}$. A typical generation process with a VAE usually has two steps: (1) The unobserved latent variable $\mathbf{z}$ is generated from the prior distribution : $\mathbf{z} \sim p(\mathbf{z})$; (2) a data sample $\mathbf{x}^{(i)}$ is generated from the conditional distribution $\mathbf{x} \sim p(\mathbf{x} \mid \mathbf{z})$. Thereby, as a generative model, the final objective of a VAE is the distribution:

$$p(\mathbf{x}) = \int p(\mathbf{x} \mid \mathbf{z}; \theta)\, p(\mathbf{z})\, d\mathbf{z}. \tag{2.6}$$

However, this integral is usually intractable and the true posterior distribution of the latent variable $p(\mathbf{z} \mid \mathbf{x})$ is also unknown. To solve these problems, another distribution called a recognition model $q(\mathbf{z} \mid \mathbf{x})$ is introduced to approximate the true posterior distribution of the latent variable $\mathbf{z}$. One reason for this approximation is that practically most samples $\mathbf{z}$ from prior distribution $p(\mathbf{z})$ will not contribute to the estimation of $p(\mathbf{x})$; i.e., $p(\mathbf{x} \mid \mathbf{z})$ is almost zero. We expect that $q(\mathbf{z} \mid \mathbf{x})$, the condition distribution of $\mathbf{z}$ given data samples, is more likely to generate data from $\mathbf{x}$, which constrains the space of valid $\mathbf{z}$ values [Doersch, 2016]. Next we will briefly discuss how we can train the VAE in a tractable way by deducing the evidence lower bound (ELBO) of the VAE.

To guarantee the validity of the approximation, the Kullback–Leibler divergence (KL divergence denoted as D) between the approximated posterior distribution $q(\mathbf{z} \mid \mathbf{x})$ and the true one $p(\mathbf{z} \mid \mathbf{x})$ for those $\mathbf{z}$ values from $q(\mathbf{z} \mid \mathbf{x})$ is calculated as shown:

$$\begin{aligned} D(q(\mathbf{z} \mid \mathbf{x}) \,||\, p(\mathbf{z} \mid \mathbf{x})) &= \int q(\mathbf{z} \mid \mathbf{x}) \log \frac{q(\mathbf{z} \mid \mathbf{x})}{p(\mathbf{z} \mid \mathbf{x})} d\mathbf{z} \\ &= E_{\mathbf{z} \sim \mathbf{q}}[\log q(\mathbf{z} \mid \mathbf{x}) - \log p(\mathbf{z} \mid \mathbf{x})]. \end{aligned} \tag{2.7}$$

By applying Bayes rule to the intractable $p(\mathbf{z} \mid \mathbf{x})$ we can bring $p(\mathbf{x})$ into the equation:

$$D(q(\mathbf{z} \mid \mathbf{x}) \,||\, p(\mathbf{z} \mid \mathbf{x})) = E_{\mathbf{z}\sim\mathbf{q}}[\log q(\mathbf{z} \mid \mathbf{x}) - \log p(\mathbf{x} \mid \mathbf{z}) - \log p(\mathbf{z})] + \log p(\mathbf{x}). \tag{2.8}$$

Hence we can obtain:

$$\log p(\mathbf{x}) - D(q(\mathbf{z} \mid \mathbf{x}) \,||\, p(\mathbf{z} \mid \mathbf{x})) = -D(q(\mathbf{z} \mid \mathbf{x}) \,||\, p(\mathbf{z})) + E_{\mathbf{z}\sim\mathbf{q}}[\log p(\mathbf{x} \mid \mathbf{z})]. \tag{2.9}$$

In Equation 2.9, the first LHS term $\log p(\mathbf{x})$ is the unknown likelihood probability of the training set which we are interested in and want to maximize. The second LHS term of the KL divergence $D(q(\mathbf{z} \mid \mathbf{x}) \,||\, p(\mathbf{z} \mid \mathbf{x}))$, albeit not tractable, can be regarded as a non-negative error term indicating whether the approximation fits well or not and needs to be minimised. In contrast, the whole RHS term, called the variational lower bound, can be well optimised. Therefore, Equation 2.9 can be changed to:

$$\log p(\mathbf{x}) \geqslant -D(q(\mathbf{z} \mid \mathbf{x}) \,||\, p(\mathbf{z})) + E_{\mathbf{z}\sim\mathbf{q}}[\log p(\mathbf{x} \mid \mathbf{z})]. \tag{2.10}$$

We can further derive that the empirical cost function of the VAE L with Gaussian latent variables $\mathbf{z}$ [Kingma and Welling, 2014] is equivalent to :

$$L = -D(q(\mathbf{z} \mid \mathbf{x}) \,||\, p(\mathbf{z})) + \frac{1}{L}\sum_{l=1}^{L} \log p\left(\mathbf{x} \mid \mathbf{z}^{(l)}\right), \tag{2.11}$$

where $\mathbf{z}^{(l)} = g\left(\mathbf{x}, \varepsilon^{(l)}\right)$, $\varepsilon^{(l)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and g is the deterministic and differentiable mapping with an auxiliary noise variable ε, which is also called the reparametrization trick [Kingma and Welling, 2014] to allow the error to back-propagate to the latent variables. In addition, the first RHS term in Equation 2.11 can also be computed in a closed form under the Gaussian assumption.

So far we have derived the general formulation of the cost function for a variational auto-encoder. In Equation 2.11, the first KL divergence term can be regarded as a regularisation whereas the second term is the reconstruction error in

auto-encoder parlance. The probability $q(\mathbf{z} \mid \mathbf{x})$ and $p(\mathbf{x} \mid \mathbf{z})$ can be computed using neural networks as an encoder and a decoder, respectively. Hence, we can train a VAE to learn the underlying latent distribution of the data via stochastic gradient descent (SGD), and new samples can be generated during inference via sampling from the prior distribution of latent variable $p(\mathbf{z})$. In Chapter 3, a VAE is adopted to learn the code latent representation of dense semantics and geometry, i.e., variable $\mathbf{z}$.

2.3.3 Semantic Segmentation using Deep Learning

Semantic segmentation is a dense prediction task of classifying each pixel of an image into a pre-defined semantic category, and is one of most important vision tasks in scene understanding. The fully convolutional neural network [Long et al., 2015] and U-Net [Ronneberger et al., 2015] are seminal CNN architectures in segmentation (shown in Figure 2.3 and Figure 2.4), with a huge amount of following work, pushing the state-of-the-art on challenging benchmarks including the PASCAL, MS-COCO, ScanNet datasets [Everingham et al., 2010, Lin et al., 2014, Dai et al., 2017a] and applications [Zhao et al., 2017b, Chen et al., 2017]. To avoid the expensive annotation cost of dense semantic labelling, weakly-/semi-/self-supervised approaches for semantic labelling have had more recent attention to cluster and segment natural images into consistent semantic regions [Zhou et al., 2018, Hsu et al., 2019, Hung et al., 2019].

In semantic mapping systems, a dense SLAM system provides the key geometric component for scene understanding, namely a dense map reconstruction and camera localisation, and CNN based segmentation methods usually enable high-level reasoning beyond geometry. SemanticFusion [McCormac et al., 2017a] assigns 2D semantic segmentations to dense 3D surfel maps; CNN-SLAM [Tateno et al., 2017] uses both dense depth and label predictions to semantically densify the semi-dense map from LSD-SLAM and estimate the scale of geometry (Figure 2.5).

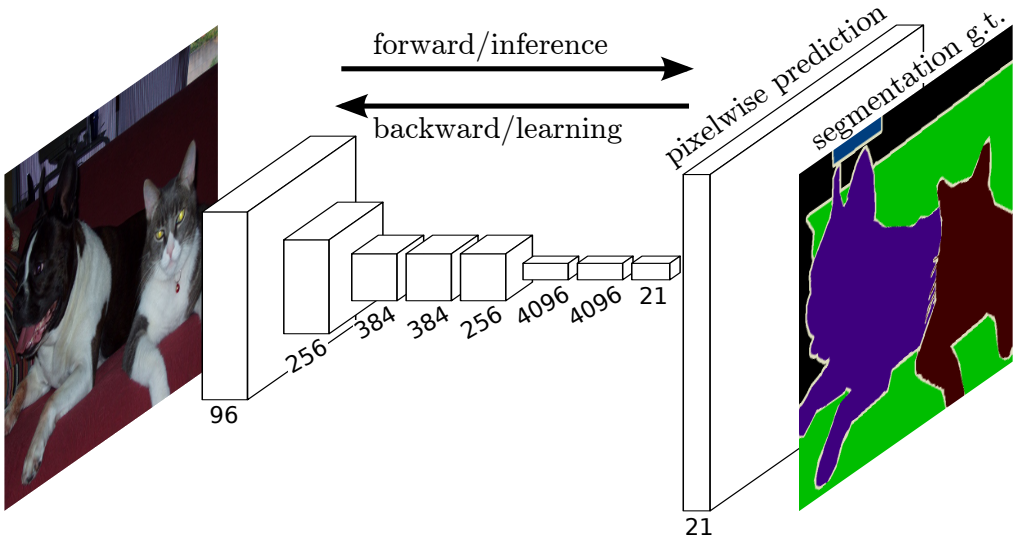


Figure 2.3: Network architecture of a Fully Convolutional Network (FCN) [Long et al., 2015].

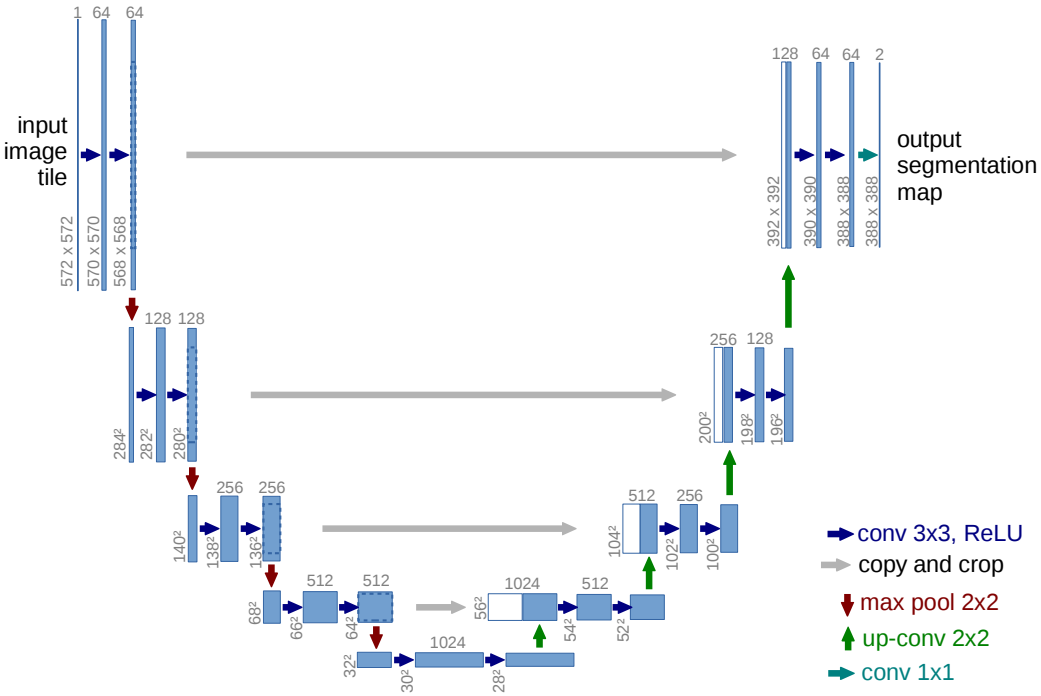


Figure 2.4: Network architecture of the U-Net [Ronneberger et al., 2015].

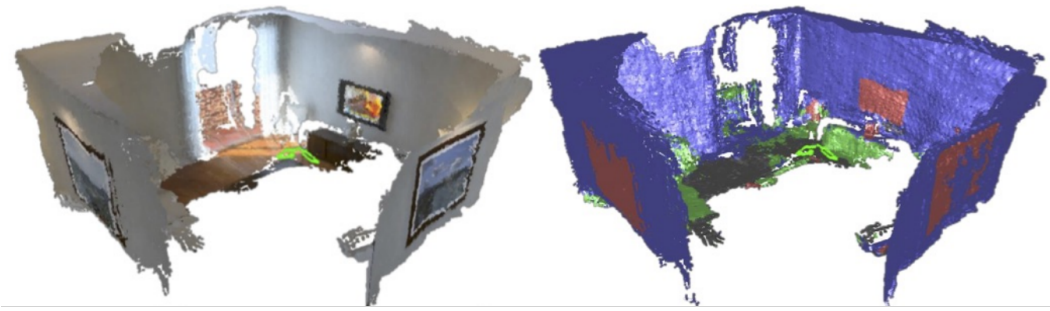


Figure 2.5: Dense 3D reconstruction from CNN-SLAM [Tateno et al., 2017].

When it comes to the evaluation of 2D semantic segmentation, the most commonly used metrics are total pixel accuracy (Pixel Acc.), class average accuracy (Class Avg. Acc.) and mean intersection-over-union (mean IoU, mIoU, or flat mean Jaccard index). Specifically, total pixel accuracy simply measures the percentage of corrected classified pixels out of the total number pixels in the image; class average accuracy computes the mean percentage of correct pixels within each class; mean IoU quantifies the percentage of overlap between the ground-truth mask and our predicted mask averaged over all classes. Though total pixel accuracy describes the overall performance, it is dominated by large regions such as walls, floors, ceilings; while class average metrics, including class average accuracy and mean IoU, equally weight all classes and are relatively more sensitive to small object classes. Assume there are C valid semantic classes, we denote the number of pixels which belong to ground truth class i and are predicted as class j by n_{ij}. Therefore, three above-mentioned metrics can be defined as follows:

Total Pixel Accuracy:

$$\frac{\sum_{i=0}^{C-1} n_{ii}}{\sum_{i=0}^{C-1}\sum_{j=0}^{C-1} n_{ij}}. \tag{2.12}$$

Class Average Accuracy:

$$\frac{1}{C}\sum_{i=0}^{C-1}\frac{n_{ii}}{\sum_{j=0}^{C-1} n_{ij}}. \tag{2.13}$$

mIoU:

$$\frac{1}{C}\sum_{i=0}^{C-1}\frac{n_{ii}}{\sum_{j=0}^{C-1} n_{ij}+\sum_{j=0}^{C-1} n_{ji}-n_{ii}}. \tag{2.14}$$

2.4 Neural Implicit Representations

A fundamental research interest across computer graphics, computer vision and SLAM is to find the best underlying scene representation of an environment's appearance, geometry and semantics, etc. Classical representations have been successfully applied in dense 3D reconstruction. However, they all suffer from discretisation and are hence inefficient in representing detailed shape, as discussed in Chapter 1. Recently proposed neural implicit representation has shown promising results in neural rendering and reconstruction thanks to being continuous by nature [Sitzmann et al., 2019b, Park et al., 2019, Mildenhall et al., 2020]. The actual resolution eventually depends on the capacity of the deep neural networks used.

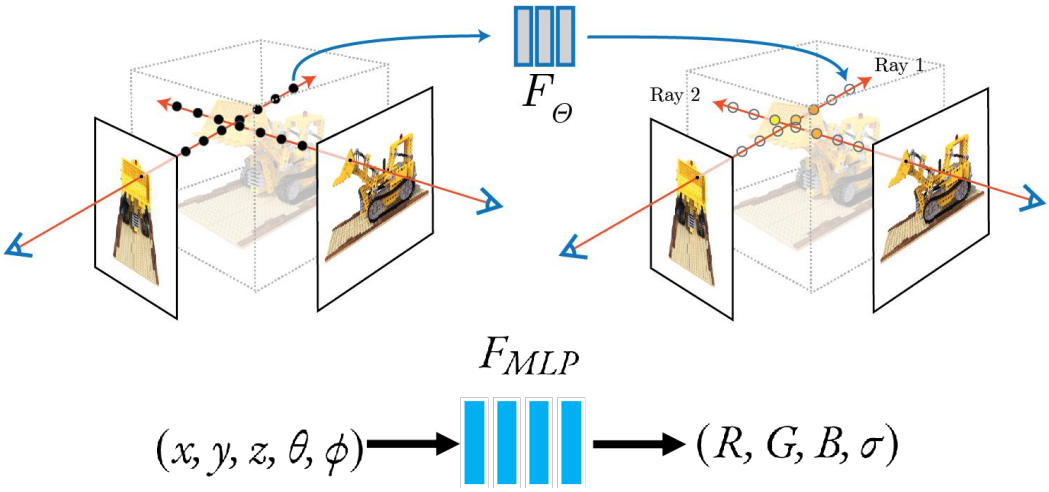


Figure 2.6: Overview of NeRF. 5D input (3D position and 2D viewing direction) are fed to a MLP to obtain the corresponding occupancy and radiance value. Volume rendering technique is used to compute the 2D pixel value traversed by the ray.

In this thesis, the neural radiance fields (NeRF) [Mildenhall et al., 2020] is of par-

ticular interest because of its scene-specific nature which means only in-situ observations are required. Realistic rendering can be obtained at novel viewpoints along with accurate internal 3D geometry after training with only posed 3D colour images, shown in Figure 2.6. Given multiple images of a static scene with known camera intrinsics and extrinsics, NeRF [Mildenhall et al., 2020] uses MLPs to implicitly represent the continuous 3D scene density σ and colour $\mathbf{c} = (r, g, b)$ as a function of continuous 5D input vectors of spatial coordinates $\mathbf{x} = (x, y, z)$ and viewing directions $\mathbf{d} = (\theta, \phi)$. Specifically, $\sigma(\mathbf{x})$ is designed to be a function of only 3D position while the radiance $\mathbf{c}(\mathbf{x}, \mathbf{d})$ is a function of both 3D position and viewing direction. To compute the colour of a single pixel, NeRF [Mildenhall et al., 2020] approximates volume rendering by numerical quadrature, of which *hierarchical volume sampling* and *positional encoding (PE)* are two key design choices enabling high-quality photorealistic rendering.

Hierarchical volume sampling learns two set of MLP, namely a coarse MLP and fine MLP concurrently. As a result, the coarse MLP can provide an initial estimate of scene geometry to the fine network and hence allow it to allocate more samples to regions expected to be visible under a limited sampling budget. Within one hierarchy, if $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$ is the ray emitted from the centre of projection of camera space through a given pixel, traversing between near and far bounds (t_n and t_f), then for selected K random quadrature points $\{t_k\}_{k=1}^{K}$ between t_n and t_f, the approximated expected colour $\hat{\mathbf{C}}$ is given by:

$$\hat{\mathbf{C}}(\mathbf{r}) = \sum_{k=1}^{K} \hat{T}(t_k)\, \alpha\big(\sigma(t_k)\delta_k\big)\, \mathbf{c}(t_k), \tag{2.15}$$

$$\text{where} \quad \hat{T}(t_k) = \exp\left(-\sum_{k'=1}^{k-1} \sigma(t_k)\delta_k\right), \tag{2.16}$$

where $\alpha(x) = 1 - \exp(-x)$, and $\delta_k = t_{k+1} - t_k$ is the distance between two adjacent quadrature sample points.

Positional encoding is shown to be an important 'trick' in making NeRF cap-

able of learning high-frequency details. The axis-aligned positional encoding is applied to lift each normalised and proximate low dimensional input component from $xyz\theta\phi$ to a more distinguishable high dimensional space using trigonometry functions [Vaswani et al., 2017, Mildenhall et al., 2020]:

$$\gamma^L(p) = (\sin(2^0\pi p), \cos(2^0\pi p), \ldots, \sin(2^{L-1}\pi p), \cos(2^{L-1}\pi p)), \tag{2.17}$$

where L represents the maximum frequency of positional encoding. Fourier feature mapping [Tancik et al., 2020] further extends the axis-aligned positional encoding to a more general formulation which alleviates the deterministic and on-axis frequencies by using randomly sampled isotropic Gaussian mapping. Given multi-view training images of the observed scene, NeRF uses stochastic gradient descent (SGD) to optimise σ and $\mathbf{c}$ by minimising photometric discrepancy between rendered images and observed images.

NeRF has led a sparking explosion of interest in the community [Dellaert and Yen-Chen, 2020] and many follow-up pieces of work based on NeRF appeared recently [Zhang et al., 2020, Srinivasan et al., 2021, Martin-Brualla et al., 2020]. Our work in Chapter 4 is also inspired by NeRF. Instead of focusing on photorealistic rendering, we are interested in its potential as a scene-specific semantic representation for robots' scene understanding. Chapter 5 further uses neural implicit representation to enable a real-time interactive scene labelling system.

CHAPTER 3

SceneCode

Contents

Parts of this Chapter appear in: Zhi, S., Bloesch, M., Leutenegger, S. and Davison, A. (2019). SceneCode: Monocular Dense Semantic Reconstruction using Learned Encoded Scene Representations. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. [Zhi et al., 2019]

3.1 Introduction

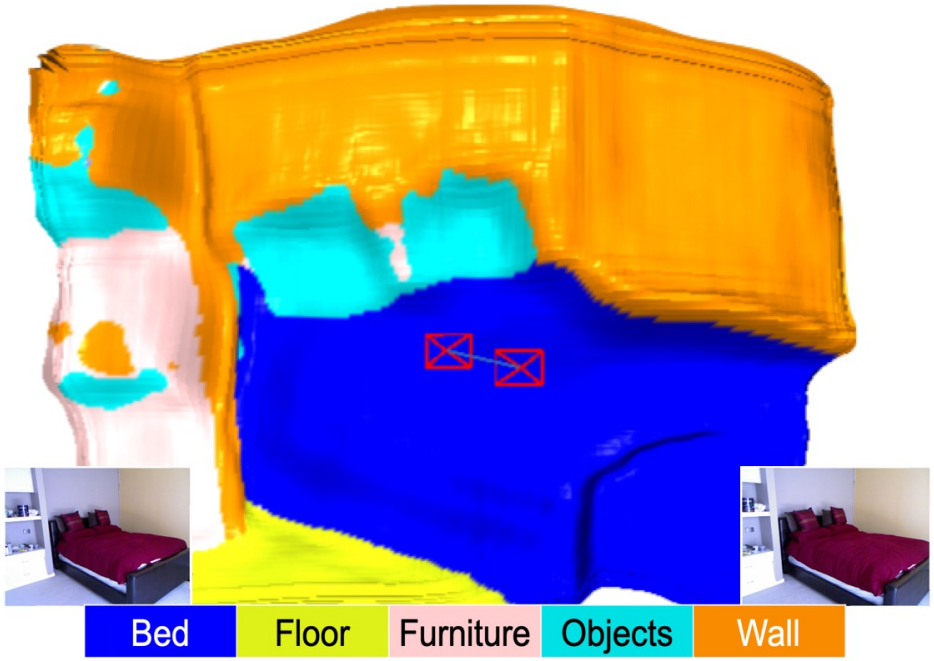


Figure 3.1: Exemplar result of two-view dense semantic structure from motion (SfM) using proposed SceneCode system in this work. Compact representations of semantics and geometry have been jointly optimised with camera pose to obtain smooth and consistent estimates.

Incremental semantic mapping systems which create 3D semantic maps from image sequences must store and update representations of both geometry and semantic entities. The most straightforward approaches, such as [Kahler and Reid, 2013, Hermans et al., 2014, McCormac et al., 2017a, Xiang and Fox, 2017, Xiao and Quan, 2009, Ma et al., 2017] which paint a dense geometric SLAM map with fused semantic labels predicted from views [Long et al., 2015, Lin et al., 2017, He et al.,

2017b], are expensive in terms of representation size; label scenes in an incoherent way where each surface element (depth pixels, surfels, or voxels) independently stores its own class estimates so that spatial correlation is discarded; and do not benefit from semantic labelling improving motion or geometry estimation.

At the other end of the scale are approaches which explicitly recognise object instances and build scene models as 3D object graphs [McCormac et al., 2018, Sünderhauf et al., 2017, Nicholson et al., 2018, Runz et al., 2018]. These representations have the token-like character we are looking for, but are limited to mapping discrete 'blob-like' objects from known classes and leave large fractions of scenes undescribed.

Looking for efficient representations of whole scenes, we are inspired by seminar work CodeSLAM from Bloesch *et al*. [Bloesch et al., 2018] which used a learned encoding to represent the dense geometry of a scene with small codes which can be efficiently stored and jointly optimised in multi-view SLAM. While CodeSLAM encoded only geometry, here we propose SceneCode and show that we can extend the same conditional variational auto-encoder (CVAE) to represent the multimodal distribution of semantic segmentation.

As in CodeSLAM, our learned low-dimensional semantic code is especially suitable for, but not limited to keyframe based semantic mapping systems, and allows for joint optimisation across multiple views to maximise semantic consistency. This joint optimisation alleviates the problems caused by the independence of surface elements assumed by most semantic fusion methods, and allows much higher quality multi-view labellings which preserve whole elements of natural scenes.

We show that compact representations of geometry and semantics can be jointly learned, resulting in the multitask CVAE used in this chapter (Figure 3.2). This network makes it possible to build a monocular dense semantic SLAM system where geometry, poses and semantics can be jointly optimised.

To summarise, the key contributions of this chapter are:

- A compact and optimisable representation of semantic segmentation using an image-conditioned variational auto-encoder.
- A new multi-view semantic label fusion method optimising semantic consistency.
- A monocular dense semantic 3D reconstruction system, where geometry and semantics are tightly coupled into a joint optimisation framework.

3.2 Related Work

In this section we briefly discussed most related works in view-based neural semantic representation of scenes.

2D neural representation of structured semantic segmentation of the type we propose have been studied by several authors. Sohn *et al*. [Sohn et al., 2015] proposed a CVAE to learn the distribution of object segmentation labels using Gaussian latent variables. Due to the learned distribution, the resulting object segmentation was more robust to noisy and partially observed data compared to discriminative CNN models. Pix2Pix from Isola *et al*. [Isola et al., 2017] used a conditional Generative Adversarial Network (GAN) to achieve image to image translation task in which the conditional distribution of semantic labels is implicitly learned. However, when used for semantic prediction from colour images, the GAN training process induces hallucinated objects. In addition, the distributions learned by GANs are not directly accessible and optimisable in the form we need for multi-view semantic fusion.

Closest work to ours is [Kohl et al., 2018] from Kohl *et al*. They proposed a probabilistic U-Net to address the ambiguities of semantic segmentation due to

insufficient context information. A CVAE was designed to learn the multimodal distribution of segmentations given colour images through a low-dimensional latent space, and it was shown that ambiguities can be well modelled by a compact latent code. We build on this idea and show that we can use the learned latent space to integrate multi-view semantic labels, and build a monocular dense SLAM system capable of jointly optimising geometry and semantics.

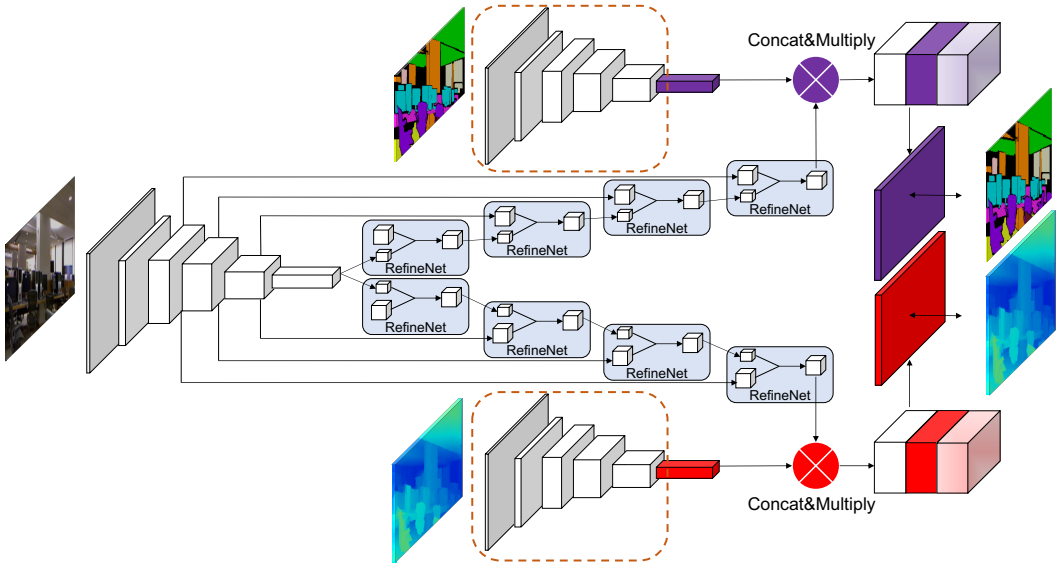


Figure 3.2: The proposed multitask conditional variational auto-encoder (multitask CVAE). Depth images and semantic labels (one-hot encoded) are encoded to two low-dimensional latent codes via VGG-like fully convolutional networks. These recognition models shown in the dashed boxes are not accessible during inference. The RGB images are processed by a U-shaped network with a ResNet-50 backbone. Finally, the sub-parts are combined by $\otimes$ operations standing for a combination of broadcasting, concatenation, and element-wise multiplication.

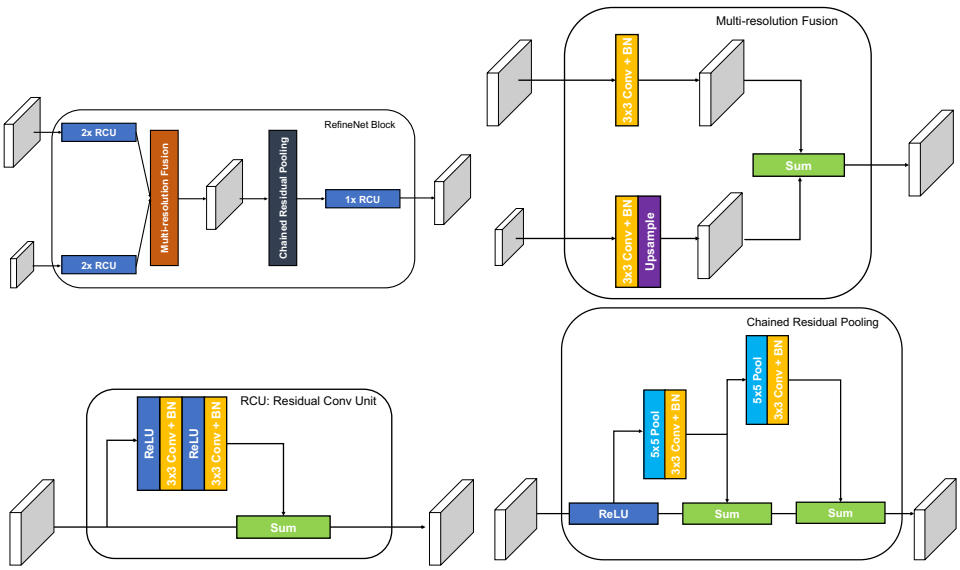


Figure 3.3: Components of the RefineNet Unit used in our multitask CVAE. Unlike the original implementation, batch-normalization (BN) is added after each convolution. BN is necessary in our experiments to stabilise the training of the RefineNet units.

3.3 Compact Geometry + Semantics Encoding

3.3.1 Image Conditioned Auto-Encoding of Depth and Semantics

Dense semantic maps, though capture complete shape information with corresponding semantic labels, own high dimensionality and are computationally expensive to store, process and update. Variational auto-encoder has the right capability to encode high dimensional input data into an explicit, compact and optimisable latent space. Naive auto-encoding of depth maps and semantic labels are not feasible because there is no access to ground truth depths and labels in monocular vision systems. Motivated by CodeSLAM [Bloesch et al., 2018], to enable accurate and efficient encoding with a low-dimensional code, RGB image conditioned auto-encoding is favoured as images are always available from camera and the latent code can focus on representing information which is not able

to retrieve from colour images. A multitask variational auto-encoder (multitask CVAE) is proposed in this work to simultaneously learn two separate codes for depth and semantics, respectively, on top of a shared backbone extracting image features. Multitask design here allows for concise network architecture and potentially benefits individual task with complementary information from another domain.

3.3.2 Multitask CVAE Network Architecture

Our multitask CVAE (see Figure 3.2) learns the conditional probability densities for depths and semantic segmentations conditioned on colour images in a manner similar to the compact representation of geometry in CodeSLAM [Bloesch et al., 2018]. The network consists of three main parts: a U-shaped multitask network with skip connections and two variational auto-encoders for depth and semantic segmentation.

The U-shaped multitask network contains one shared encoder with a ResNet-50 backbone [He et al., 2016] and two separate decoders adopting RefineNet units [Lin et al., 2017]. Unlike the original implementation, batch normalisation is added after each convolution in the RefineNet unit to stabilise training. Each of the two variational auto-encoders consists of a VGG-like fully convolutional recognition model (encoder) followed by a linear generative model (decoder), which is coupled with the U-net and thus conditioned on colour images.

More specifically, in the linear decoder the latent code is first broadcast spatially to have the same height/width and then 1×1 convolved to have the same dimensionality as the image feature maps from the last RefineNet unit. A merged tensor is then computed by doing a three-fold concatenation of the broadcast/convolved code, the RefineNet unit, and an element-wise multiplication of the two. Finally, convolution (without nonlinear activation) and bilinear upsampling is applied to

obtain the prediction. The motivation for this procedure is to obtain a linear relationship between code and prediction which is conditioned on the input image in a nonlinear manner [Bloesch et al., 2018] — the linearity enabling pre-computation of Jacobians during inference at test time (see Section 3.4). The predicted depth D and semantics (unscaled logits before softmax function) S can thus be formulated as:

$$D\left(\mathbf{c}_d, I\right) = D_0(I) + \mathbf{J}_d(I)\mathbf{c}_d, \tag{3.1}$$

$$S\left(\mathbf{c}_s, I\right) = S_0(I) + \mathbf{J}_s(I)\mathbf{c}_s, \tag{3.2}$$

where $\mathbf{J}_{s/d}$ represents the learned linear influence which only depends on the input colour image, and $D_0(I) = D(\mathbf{0}, I)$ and $S_0(I) = S(\mathbf{0}, I)$. Due to our variational setup, $D_0(I)$ and $S_0(I)$ can be interpreted as the most likely prediction given the input image alone. Note the generality of this framework, which could be combined with arbitrary network architectures.

3.3.3 Network Training Configuration

Both the depth and semantic predictions are jointly trained using ground truth data. In addition to the reconstruction losses discussed in the following sections, the variational setup requires a KL-divergence based loss on the latent space to minimise the negative evidence lower bound (ELBO) [Kingma and Welling, 2014]:

$$L_{\phi,\theta}(x) = KL\Big(q_\phi\left(\mathbf{c}_x \mid x\right) \,||\, p_\theta\left(\mathbf{c}_x\right)\Big), \tag{3.3}$$

where x can be either depth image D or semantic logits S, $q_\phi\left(\mathbf{c}_x \mid X\right)$ is the approximated posterior distribution to the unknown and intractable $p_\theta\left(\mathbf{c}_x \mid X\right)$ via the recognition model (encoder).

We assume latent codes with a factorized standard Gaussian prior distribution, i.e., $p\left(\mathbf{c}_x\right) \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, and that the approximated posterior distribution $q_\phi\left(\mathbf{c}_x \mid x\right)$ is also a factorized Gaussian distribution $q\left(\mathbf{c}_x \mid x\right) \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, whose mean and

variance are predicted by the encoder. So the KL term in Equation 3.3 has a closed form formulation:

$$L_{\phi,\theta}(x) = \frac{1}{2}\sum_{j=1}^{J}((1+\log(\sigma_j^2)) - (\mu_j)^2 - (\sigma_j)^2), \tag{3.4}$$

where μ_j and σ_j are the j-th element of predicted μ and diag(Σ), respectively. J is the latent code size. In order to avoid a degrading latent space, we employ a KL annealing strategy [Bowman et al., 2015, Sønderby et al., 2016] where we gradually increase the weights of the KL terms from 0 after 2 training epochs.

To compute the reconstruction loss of depth images, as in [Bloesch et al., 2018], the raw depth values d are first transformed via a hybrid parametrisation called *proximity*, $p = a/(a+d)$, where a is the average depth value, which is set to 2m in all of our experiments. In this way, we can handle raw depth values ranging from 0 to $+\infty$ and assign more precision to regions closer to the camera. An L_1 loss function together with data dependent homoscedastic uncertainty [Kendall and Gal, 2017] is used as the reconstruction error:

$$L_{\phi,\theta}(d) = \sum_{i=1}^{N}\left[\frac{|\widetilde{p}_i - p_i|}{b_i} + \log\left(b_i\right)\right], \tag{3.5}$$

where N is the number of pixels, $\widetilde{p}_i$ and p_i are the predicted proximity and input proximity of the i-th pixel, and b_i is the predicted uncertainty of the ith pixel.

Semantic segmentation labels, which are discrete numbers, are one-hot encoded before being input to the network. Therefore, the multi-class cross-entropy function is a natural option for calculating the reconstruction loss using the predicted softmax probabilities and one-hot encoded labels:

$$L_{\phi,\theta}(s) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} k_c^{(i)} \log p_c^{(i)}, \tag{3.6}$$

where C is the number of classes, $k_c^{(i)}$ is the c-th element of the one-hot encoded labels for the i-th pixel and $p_c^{(i)}$ is the predicted softmax probability in the same position.

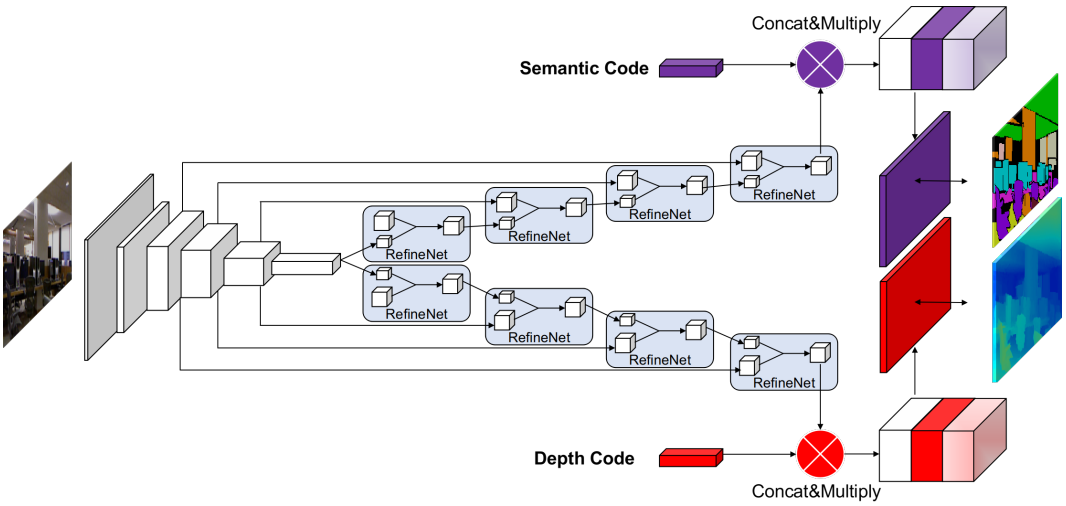


Figure 3.4: Multitask CVAE at inference time. Both semantic code and depth code are initialised to full zero codes.

Hence, the total reconstruction loss L_{total} is a linear combination of $L_{\phi,\theta}(d)$ and $L_{\phi,\theta}(s)$:

$$L_{total} = \alpha_d * L_{\phi,\theta}(d) + \alpha_s * L_{\phi,\theta}(s), \tag{3.7}$$

where the weight on the multitask reconstruction error α_d and α_s are adaptively learned during training as the task-dependent uncertainty [Kendall et al., 2018]. We found this learned weighting scheme provides better overall performance than equal weighting.

In all of our experiments, we train the whole network in an end-to-end manner using the Adam optimiser [Kingma and Ba, 2015] with an initial learning rate of 10^{-4} and a weight decay of 10^{-4}. The ResNet-50 backbone is initialised using ImageNet pre-trained weights, and all other weights are initialised using He *et al.*'s method [He et al., 2015].

3.4 Fusion via Multi-View Code Optimisation

In a multi-view setup, depth, semantics, and motion estimates can be refined based on consistency in overlapping regions by making use of dense correspondences,

Figure 3.5: Illustration of semantic reconstruction from SceneCode system [Zhi et al., 2019].

shown in Figure 3.5. While the use of photometric consistency is well-established, here we also introduce semantic consistency, i.e. any given part of our scene should have the same semantic label irrespective of viewpoint. The semantic consistency is less affected by disturbances such as non-Lambertian reflectance, but may be subject to quantisation errors and cannot be directly measured. The underlying intuition of semantic consistency loss in Equation 3.11 is that corresponding pixels must have the same semantic label, and thus similar (but not necessary the same) softmax categorical probabilities. Unlike the photo-consistency assumption, the semantic consistency assumption is comparatively weak since it is not anchored to any actual measurement, though this is somewhat alleviated by the zero-code prior described above. Nevertheless, as the viewpoint varies, different semantic cues may become available and a previously semantically ambiguous region may become more distinctive. Instead of fusing this information element-wise [McCormac et al., 2017a], the estimates are propagated all the way back to the semantic code, allowing spatial information fusion.

Given no additional information, an all-zero code is the most likely code because of the multivariate Gaussian prior assumption during training (see Section 3.3.3). Hence, the corresponding zero code predictions D_0 and S_0 in Equation 3.1 and Equation 3.2 are the most likely prediction (i.e., monocular prediction) for depths and semantics, respectively. This zero code prediction can thus be used

both as an initialisation value and as a prior during optimisation at test time (during which we have no access to depths or semantic labels). The main motivation to treat zero code predictions as prior information is that, as discussed above, there is no intrinsic semantic cost to guarantee correct convergence during multi-view semantic refinement, therefore zero code can act as a regularisation term which drives the optimisation to find local minima near the origin point of latent space.

The probabilistic formulation of the system allows it to embed depth, semantics and motion into a unified probabilistic framework and thereby combine an arbitrary number of information sources including images, semantic constraints, priors, motion models or even measurements from other sensors.

3.4.1 Geometry Refinement

In analogy to [Bloesch et al., 2018], given an image I_A with its depth code $\mathbf{c}_d^A$, and a second image I_B with estimated relative rigid body transformation $\mathbf{T}_{BA} = (\mathbf{R}_{BA}, \mathbf{t}_{BA}) \in SO(3) \times \mathbb{R}^3$, the dense correspondence for each pixel $\mathbf{u}$ in view A is:

$$w\left(\mathbf{u}_A, \mathbf{c}_d^A, \mathbf{T}_{BA}\right) = \pi\left(\mathbf{T}_{BA}\, \pi^{-1}\left(\mathbf{u}_A, D_A[\mathbf{u}_A]\right)\right), \tag{3.8}$$

where π and π^{-1} are the projection and inverse projection functions, respectively. D_A stands for $D_A = D\left(\mathbf{c}_d^A, I_A\right)$, and the square bracket operation $[\mathbf{u}]$ means a value look-up at pixel location $\mathbf{u}$. We can then establish the photometric error r_i based on the photo-consistency assumption [Kerl et al., 2013]:

$$r_i = I_A[\mathbf{u}_A] - I_B\left[w\left(\mathbf{u}_A, \mathbf{c}_d^A, \mathbf{T}_{BA}\right)\right]. \tag{3.9}$$

Similarly, we can derive the geometric error term r_z as:

$$r_z = D_B[w(\mathbf{u}_A, \mathbf{c}_d^A, \mathbf{T}_{BA})] - [\mathbf{T}_{BA}\pi^{-1}(\mathbf{u}_A, D_A[\mathbf{u}_A])]_Z, \tag{3.10}$$

where $[\cdot]_Z$ refers to the depth value of a point.

3.4.2 Semantics Refinement

Given images I_A, I_B sharing a common field of view (FOV), and their pre-softmax predictions S_A and S_B generated from semantic codes $\mathbf{c}_s^A$ and $\mathbf{c}_s^B$, we propose to establish a semantic error term via dense warping:

$$r_s^{'} = DS\left(S_A[\mathbf{u}_A], S_B\left[w\left(\mathbf{u_A}, \mathbf{c}_d^A, \mathbf{T}_{BA}\right)\right]\right), \tag{3.11}$$

where DS can be an arbitrary function measuring distance/dissimilarity [Cha and Srihari, 2002]. In the scope of this paper, DS is chosen to be the Euclidean distance after applying softmax on the logits. Establishing the semantic error on top of semantic labels is not adopted here due to the loss of multi-class information and the induced non-differentiability from argmax operation.

The underlying intuition of Equation 3.11 is that corresponding pixels must have the same semantic label, and thus similar (but not necessary the same) softmax categorical probabilities. However, unlike photometric and geometric errors (Equation 3.9 and 3.10) which are intrinsic to geometry, so that multi-view optimisation leads to better geometry estimation, semantic loss itself (Equation 3.11) is under-constrained and not intrinsic to semantic refinement. Simply minimising semantic error can be trivially solved by assigning arbitrary consistent but incorrect semantic labels to the correspondences. For example, corresponding pixels belonging to "table" still have zero semantic error if they are all classified to "chair".

To avoid trivial solution, prior information/regularisation is needed. We propose to leverage on the zero code semantic prediction (i.e., the likelihood prediction of single view) and make it a new regularisation term in Equation 3.11. Our final semantic error term is:

$$r_s = r_s^{'} + \lambda\left(\left\|\mathbf{c}_{\mathbf{s_A}}\right\|_2^2 + \left\|\mathbf{c}_{\mathbf{s_B}}\right\|_2^2\right), \tag{3.12}$$

where λ is the weight on the regularisation term.

Unlike the photo-consistency assumption, the semantic consistency assumption in Equation 3.12 is comparatively weak since it is not anchored to any actual measurement, though this is somewhat alleviated by the zero-code prior described above. Nevertheless, as the viewpoint varies, different semantic cues may become available and a previously semantically ambiguous region may become more distinctive. Instead of fusing this information element-wise [McCormac et al., 2017a], the estimates are propagated all the way back to the semantic code, allowing spatial information fusion.

The semantic error term in Equation 3.12 is differentiable not only w.r.t. the semantic code $\mathbf{c}_{\mathbf{s_A}}$ and $\mathbf{c}_{\mathbf{s_B}}$, but to camera pose and the depth of the reference keyframe as well, making it possible to affect geometry using semantic information. This naturally enables semantic information to influence motion and structure estimation, i.e., the framework will for instance attempt to align chairs with chairs and walls with walls. In this chapter, we focus on refining the semantics using estimated motion and geometry while leaving the opposite direction as exciting future research. The Jacobians can be computed using the chain rule:

$$\frac{\partial r_s}{\partial \mathbf{c}_{s_A}} = \frac{\partial r_s}{\partial DS}\frac{\partial DS}{\partial S_A}\frac{\partial S_A}{\partial \mathbf{c}_{s_A}}, \tag{3.13}$$

$$\frac{\partial r_s}{\partial \mathbf{c}_{s_B}} = \frac{\partial r_s}{\partial DS}\frac{\partial DS}{\partial S_B}\frac{\partial S_B}{\partial \mathbf{c}_{s_B}}, \tag{3.14}$$

$$\frac{\partial S_B[\mathbf{v}]}{\partial \mathbf{t}_{BA}} = \frac{\partial S_B[\mathbf{v}]}{\partial \mathbf{v}}\frac{\partial \pi(\mathbf{x})}{\partial \mathbf{x}}, \tag{3.15}$$

$$\frac{\partial S_B[\mathbf{v}]}{\partial \mathbf{R}_{BA}} = \frac{\partial S_B[\mathbf{v}]}{\partial \mathbf{v}}\frac{\partial \pi(\mathbf{x})}{\partial \mathbf{x}}\left[-\mathbf{R}_{BA}\pi^{-1}(\mathbf{u}, d)\right]^{\times}, \tag{3.16}$$

where $\times$ is the skew symmetric matrix form of a 3D vector, together with the

following abbreviations:

$$\mathbf{v} = w\left(\mathbf{u_A}, \mathbf{c}_{d_A}, \mathbf{T}_{BA}\right), \tag{3.17}$$

$$\mathbf{x} = \mathbf{R}_{BA}\pi^{-1}(\mathbf{u}_A, D_A[\mathbf{u}_A]) + \mathbf{t}_{BA}, \tag{3.18}$$

$$\frac{\partial \mathbf{x}}{\partial \mathbf{R}_{BA}} = \left(-\mathbf{R}_{BA}\pi^{-1}(\mathbf{u}_A, D_A[\mathbf{u}_A])\right)^{\times}. \tag{3.19}$$

Benefits of Linear Decoder

Both photometric, geometric and semantic errors are differentiable w.r.t. the input camera poses and latent codes, so that Jacobians can be computed using the chain rule. Due to the designed linear relationship we can *pre-compute* the Jacobian of network predictions w.r.t. the codes which is computationally expensive to evaluate due to dense convolution operations. The depth-to-code and semantic-to-code Jacobians can be evaluated via numerical auto-differentiation using deep learning libraries such as Tensorflow and Pytorch.

Probabilistic Formulation

There is a principled and probabilistic explanation behind our refinement process. Minimizing $r_s^{'}$ in Equation 3.11 can be viewed as Maximum Likelihood Estimation (MLE) of semantic code $\mathbf{c_s}$:

$$\mathbf{c}_s = \underset{\mathbf{c_s}}{\operatorname{argmax}}\, p\left(r_s \mid \mathbf{c_s}\right). \tag{3.20}$$

If the individual term $p\left(r_s \mid \mathbf{c_s}\right)$ of each correspondence obeys an independent and identically distributed (i.i.d.) Gaussian distribution, solving Equation 3.20 becomes a classical least-square problem.

In addition, since the prior of the latent code in our network is a Gaussian (See Section 3.3.3), Equation 3.12 can be formulated as a Maximum A Posteriori (MAP) problem:

$$\mathbf{c_s} = \underset{\mathbf{c_s}}{\operatorname{argmax}}\, p\left(\mathbf{c_s} \mid r_s\right) = \underset{\mathbf{c_s}}{\operatorname{argmax}}\, p\left(r_s \mid \mathbf{c_s}\right) p\left(\mathbf{c_s}\right). \tag{3.21}$$

Although our semantics refinement approach targets a monocular keyframe based SLAM system, it can be adopted as a semantic label fusion module in arbitrary SLAM system such as stereo or RGB-D SLAM systems.

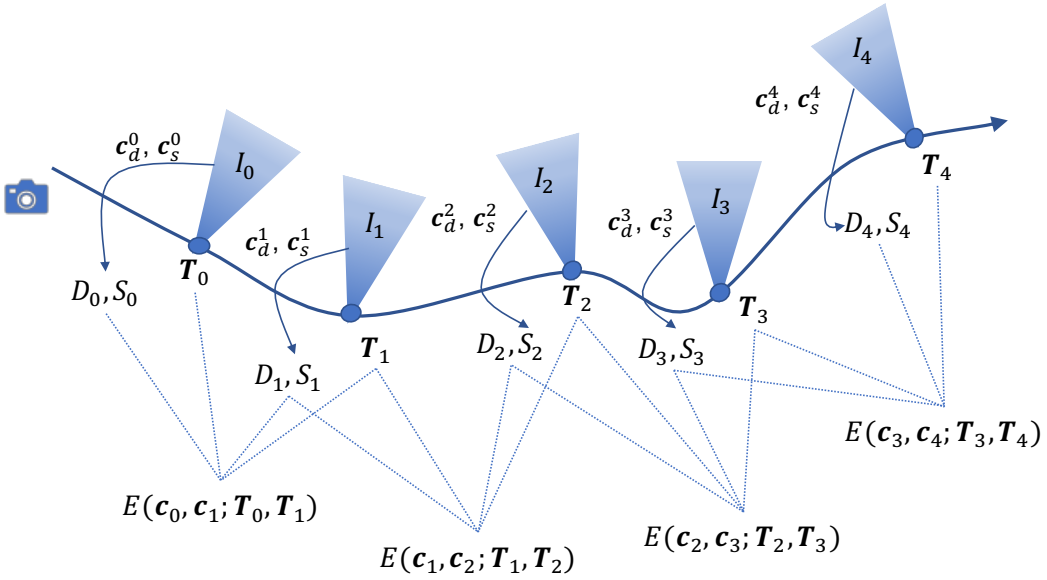


Figure 3.6: Semantic mapping formulation. Each keyframe has a colour image I, depth code $\mathbf{c}_d$ and semantic code $\mathbf{c}_s$. Second order optimisation can be applied to jointly or separately optimise camera motion, geometry and semantics.

3.5 Monocular Dense Semantic SLAM

Here we can integrate the geometry and semantics refinement processes into a preliminary keyframe based monocular SLAM system. The map is represented by a collection of keyframes, each with a camera pose and two latent codes, one for geometry and one for semantics, as shown in Figure 3.6. We follow the standard paradigm of dividing the system into tracking (front-end) and mapping (back-end) and alternate between them [Klein and Murray, 2007]. In the present chapter, for efficiency reasons, the tracking module estimates the relative 3D motion between the current frame and the last keyframe using the photometric residual only [Baker and Matthews, 2004].

The mapping module relies on dense N-frame structure from motion, by minimising photometric, geometric and the proposed semantic residuals with a zero-code prior between any two overlapping frames, which can be formulated as a non-linear least-squares problem. As in CodeSLAM [Bloesch et al., 2018], we employ loss functions that (i) remove invalid correspondences, (ii) perform relative weighting for different residuals, (iii) include robust Huber weighting, (iv) down-weight strongly slanted and potentially occluded pixels. The differentiable residuals are minimised by a damped Gauss-Newton solver. In addition, the linear decoder allows us to pre-compute the Jacobians of the network prediction w.r.t. the code for each keyframe. Because the semantic residual relies not only on the semantic code but also on data association, during mapping we adopt a stage-wise optimisation. We first jointly optimise the geometry and poses, then optimise the semantic residual, and lastly jointly optimise both geometry and semantics. In this way, we tightly couple geometry and semantics into a single optimisation framework.

3.6 Experiments

Please also see our submitted video which includes further demonstrations: https://youtu.be/MCgbgW3WA1M.

To test our method, we use three indoor datasets: the synthetic SceneNet RGB-D dataset [McCormac et al., 2017b], and the real-world NYUv2 [Silberman et al., 2012] and Stanford 2D-3D-Semantic datasets [Armeni et al., 2017]. Compared to outdoor road scenes [Geiger et al., 2012, Cordts et al., 2016], indoor scenes have different challenges with large variations in spatial arrangement and object sizes, and full 6-D motion.

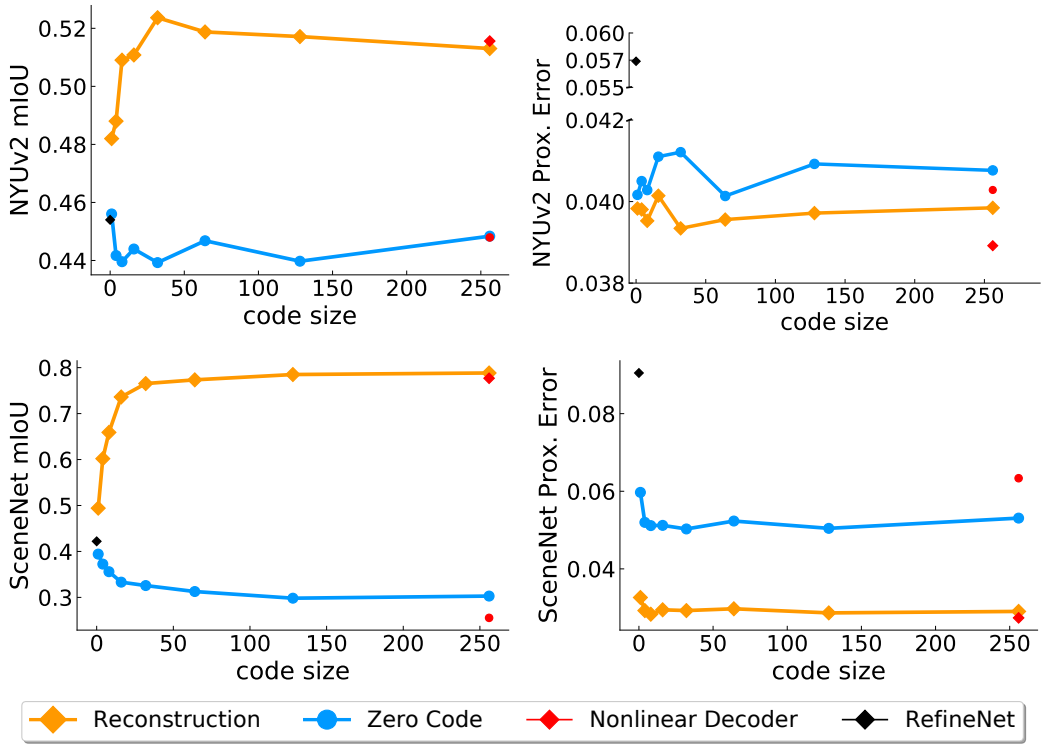


Figure 3.7: Reconstruction and zero code prediction performance of different setups on the NYUv2 and SceneNet RGB-D test sets. mIoU (higher is better) and proximity error (lower is better) are used to evaluate the performance of semantic segmentation and depth prediction, respectively. Reconstruction performance (i.e., given ground truth data as network input) increases with code size. The quality of zero code predictions is comparable to a discriminative baseline model RefineNet for semantic segmentation, and better on depth prediction. Using a nonlinear decoder leads to little improvement and requires expensive re-estimation of Jacobians per step as discussed in Section 3.4.

3.6.1 Datasets

NYUv2 has 1,449 pre-aligned and annotated images (795 in the training set and 654 in the test set) [1]. We cropped all the available images from 640×480 to valid regions of 560×425 before further processing. The 13 class semantic segmentation task is evaluated in our experiments.

Stanford 2D-3D-Semantic is a large scale real world dataset with a different set

[1] https://cs.nyu.edu/~silberman/datasets/nyu_depth_v2.html

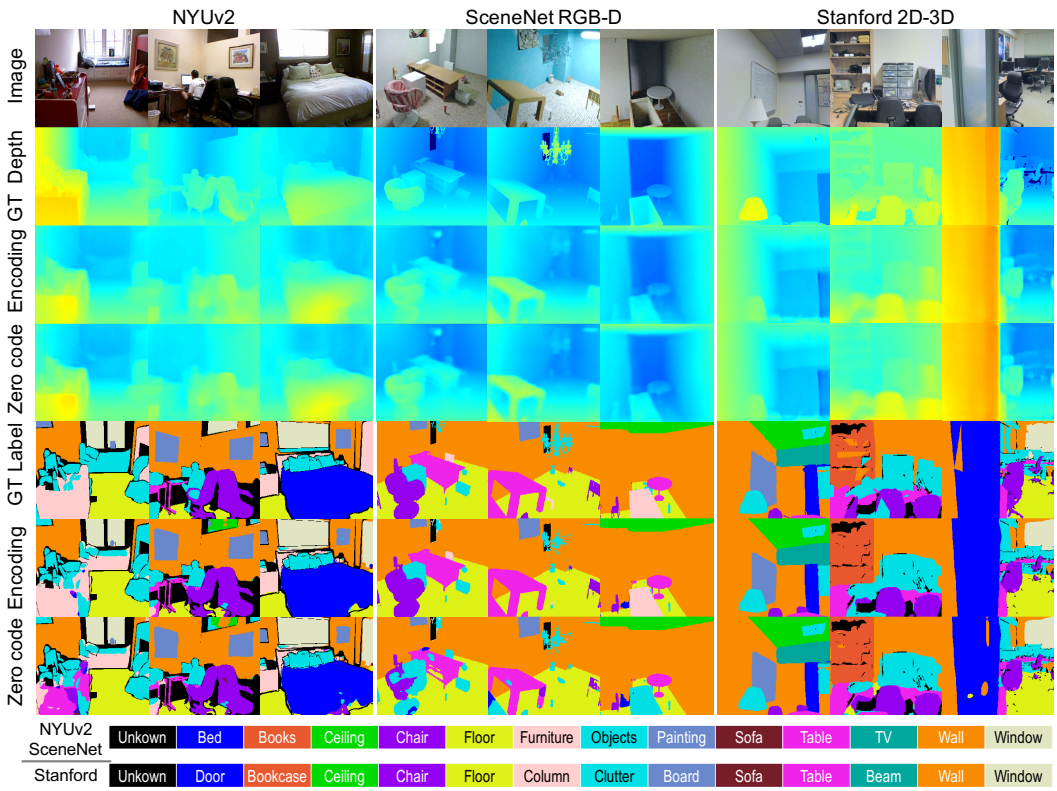


Figure 3.8: Qualitative results on the NYUv2 (left), SceneNetRGB-D (middle) and Stanford (right) datasets. Input colour images are at the top. We show ground truth, encoded predictions (code from encoder) and zero code predictions (monocular predictions) for depth and semantic labels. Incorrect semantic predictions in regions which are ambiguous for monocular predictions are corrected by extra information in the compact latent codes. Black regions are masked unknown classes.

of 13 semantic class definitions [2]. 70,496 images with random camera parameters are split into a training set of 66,792 images (areas 1, 2, 4, 5, 6) and a test set of 3,704 images (area 3). We rectified all images to a unified camera model.

The synthetic **SceneNet RGB-D** dataset provides perfect ground truth annotations for 5M images [3]. We use a subset: our training set consists of 110,000 images by sampling every 30th frame of each sequence from the first 11 original training splits. Our test dataset consists of 3,000 images by sampling every 100th frame

[2] http://buildingparser.stanford.edu/dataset.html
[3] https://robotvault.bitbucket.io/scenenet-rgbd.html

from the original validation set.

All input images are resized to a resolution of 256×192. During training, we use data augmentation including random horizontal flipping and jittering of brightness and contrast. At test time, only single scale semantic prediction is evaluated.

3.6.2 Image Conditioned Scene Representation

We first quantitatively inspect the influence of code size on both the NYUv2 and SceneNet RGB-D datasets by measuring reconstruction performance. We use the same latent code size for depth images and semantic labels for simplicity. We also train a discriminative RefineNet for semantic segmentation and depth estimation separately as a single task prediction-only baseline models (i.e. code size of 0). Figure 3.7 shows results for depth and semantic encoding with different code size and setups. The reconstruction performance indicates the capacity of the latent encoding for variational auto-encoders. Due to the encoded information, the reconstruction is consistently better than single view monocular prediction, which also shows competitive performance compared to discriminative baselines. Furthermore, reconstruction performance does not benefit from a non-linear decoder and we observe diminishing returns when the code size is larger than 32, and therefore choose this code size for later experiments.

The qualitative effects of our image conditioned auto-encoding of size 32 are shown in Figure 3.8. The zero code predictions are usually similar to the encoded predictions, though errors in ambiguous regions are corrected given the additional encoded information. Figure 3.9 displays the learned image dependent Jacobians of the semantic logits w.r.t. entries in the code. We see how each code entry is responsible for certain semantically meaningful **regions** (e.g. examine the sofa Jacobians). Additionally, each code entry also has a tendency to decrease the probability of other ambiguous classes. For two images from different viewpoints, the

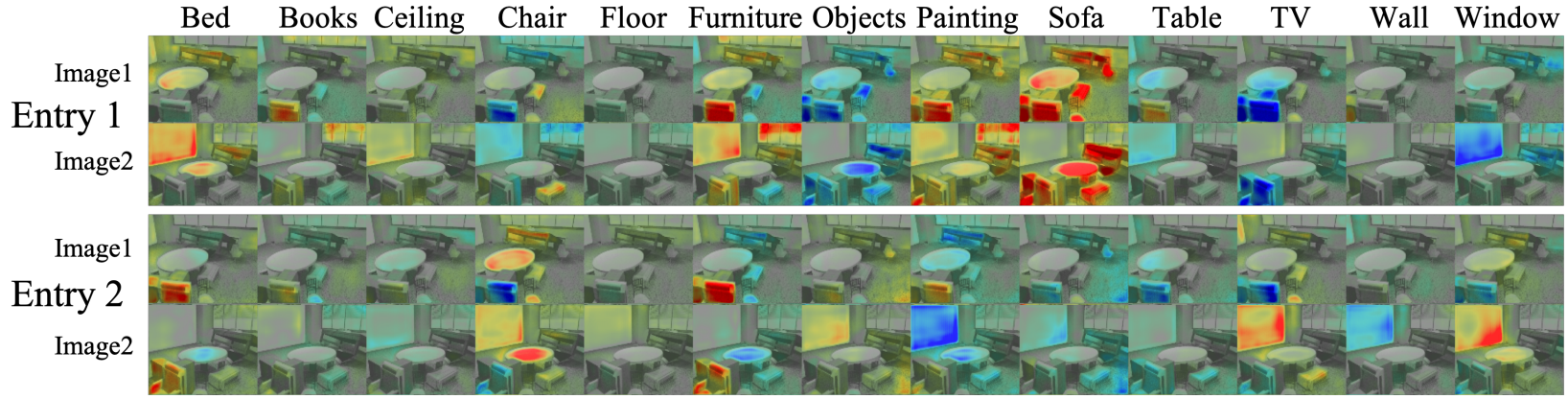


Figure 3.9: The Jacobians of semantic logits w.r.t. two code entries for a pair of wide baseline views. The columns represent the influence of the code entry across each semantic classes. Red and blue mean positive and negative influence, respectively. Semantically meaningful regions can be refined coherently during optimisation, leading to smooth and complete segmentation, and this property is automatically carried over into the semantic fusion process.

image dependent Jacobians show high consistency.

3.6.3 Semantic Label Fusion using Learned Codes

Statistics	Mean	Std	Max	Min
Rotation (degree)	5.950	9.982	163.382	0.028
Translation (meter)	0.149	0.087	0.701	0.001

Table 3.1: The statistics of the relative 3D motion between consecutive frames extracted from SceneNet RGB-D.

Our semantic refinement process can be regarded as a label fusion scheme for multi-view semantic mapping. An important advantage of code-based fusion compared to the usual element-wise fusion approaches for label fusion is its ability to naturally obtain spatially and temporally consistent semantic labels by performing joint estimation in the latent code space. This means that pixels are not assumed i.i.d when their semantic probabilities are updated, leading to smoother and more complete label regions.

To isolate only label estimation, our experiments use the SceneNet RGB-D dataset where precise ground truth depth and camera poses are available to enable perfect data association. We also mask out and ignore occluded regions. We use the zero-code monocular predictions as the initial semantic predictions for all fusion methods.

In Figure 3.10 we show the result of semantic label fusion given two views taken with a large baseline. The RHS zero code prediction struggles to recognise the table given the ambiguous context. The high information entropy indicates that the predicted semantic labels are uncertain and are likely to change during optimisation. In contrast, the LHS zero code prediction is able to accurately segment the table with relatively low entropy. By minimising the semantic cost between two views, the optimised semantic representations are able to generate consistent

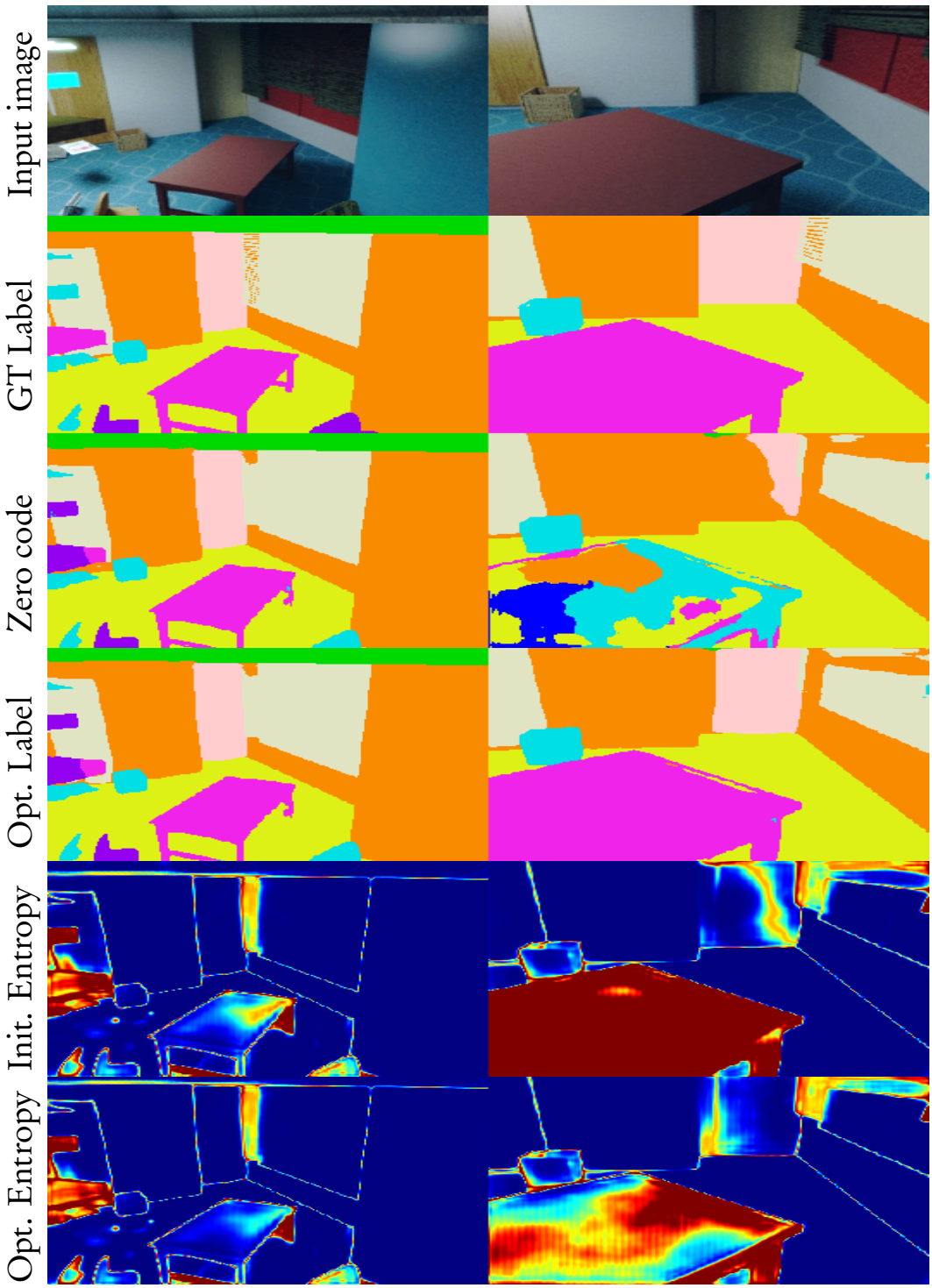


Figure 3.10: An example of two-view semantic label fusion with our method. From top to bottom rows: input colour image, ground truth semantic label, zero code prediction, optimised label (minimising semantic cost), information entropy of the zero-code softmax probabilities, information entropy of the optimised softmax probabilities.

predictions, successfully leading to the disambiguation of the RHS into a well segmented and smooth prediction. The entropy of both views is reduced as well. Similar improvements can also be observed in other regions. In addition, it is interesting to observe that the entropy map exhibits consistency with the scene structure, showing that the network can recognise the spatial extent of an object but struggles with the precise semantic class.

Qualitative results for different label fusion methods are shown in Figure 3.11. The results of both element-wise fusion approaches are obtained by integrating the probabilities of the other images into each current frame, while our result simply comes from pairing all the later frames to the first frame. For a sequence of 5 consecutive frames with small baselines, the zero code predictions are all similar and show consistent incorrect predictions in certain regions, for example, in the front object. We can observe that proposed code optimisation is able to generate much smoother predictions, while it is challenging for other approaches to improve in this case.

As a result, when there is a difficult, ambiguous region (indicated by low quality zero code predictions and high entropy), the element-wise label fusion methods lead to results which are only marginally better. However, the representation power in the learned compact code enables much smoother predictions with correct labels to be obtained through optimisation. After optimisation, the reduced entropy for these regions indicates that the network is much more confident.

As indicated by Figure 3.9, semantically meaningful regions are refined coherently during optimisation, leading to smooth and complete segmentation, and this property is automatically carried over into our code-based semantic fusion results. This makes our approach strongly different from element-wise fusion approaches which neglect local correlations and can result in noisy and incoherent labels, although this can be partly addressed by an expensive post-CRF process [Chen et al.,

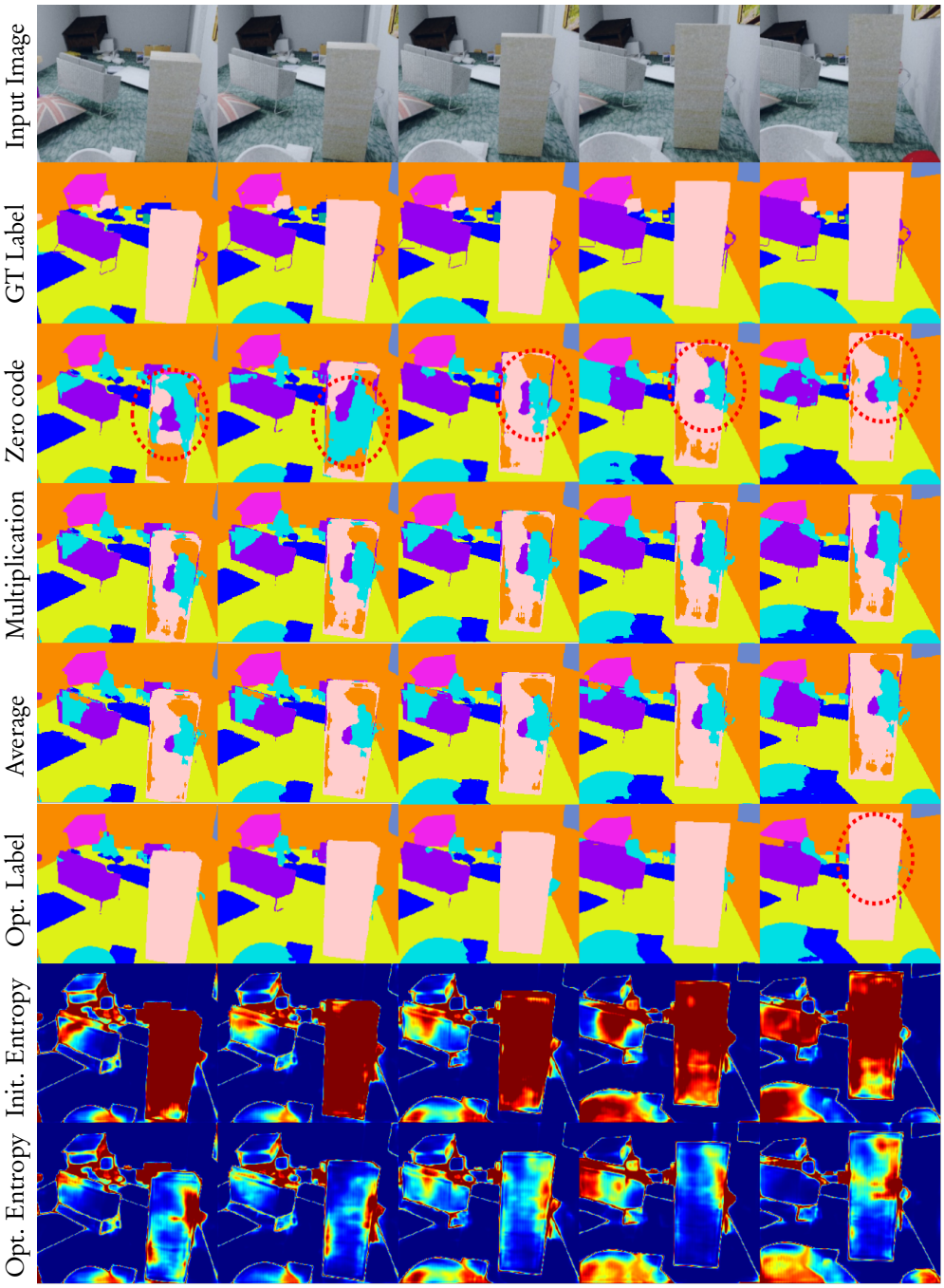


Figure 3.11: Qualitative comparison of different label fusion methods. 5 consecutive frames with a small baseline are chosen. Our method can effectively fuse multi-view semantic labels to generate smoother semantic predictions even there are consistent incorrect predictions in monocular predictions, highlighted by dashed red circles.

2018a, McCormac et al., 2017a].

Next we provide a quantitative comparison of various label fusion methods. 2000 images sampled from 1000 sequences (2 images per sequence) from SceneNet RGB-D validation set are used to evaluate the performance. We augment every extracted image with a variable number of subsequent images in the sequence to obtain short multi-view sequences (1-4 frames). Since the trajectories of SceneNet RGB-D are randomly generated, a good variety of relative transformations and baselines are included in this set. Table 3.1 shows the motion statistics from the sampled subsets.

Table 3.2 shows the effectiveness of three multi-view label fusion methods given a various number of views. Our label fusion approach using code optimisation outperforms others methods. The improvement in total pixel accuracy is not significant because of the large area of walls and floors in the dataset. However, the large improvement in the mIoU metric shows that our method is able to consider more on high-order statistics, indicating smoother predictions and better results on other small objects or fine structures.

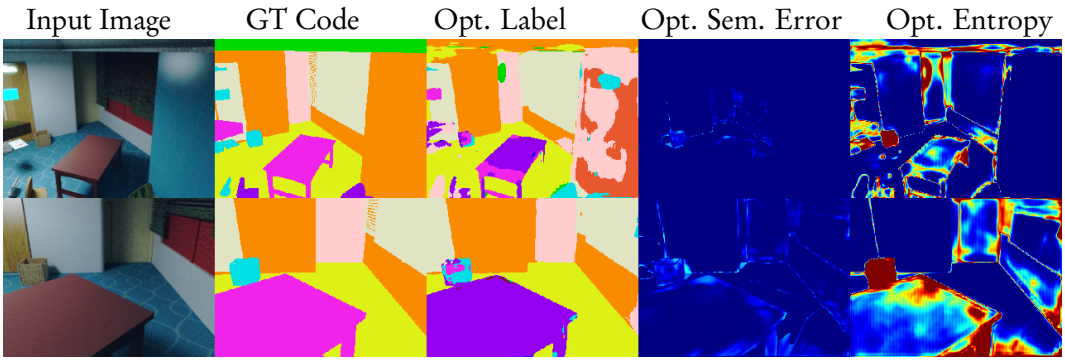


Figure 3.12: Two-view semantic label fusion **without zero code prior**. The semantic error has been minimised with higher confidence while the desks have consistent but incorrect predictions.

#Views	Method	Pix. Acc.	Cls. Acc.	mIoU
1	-	75.167	63.330	41.713
2	Multiplication	75.424	63.629	42.326
	Average	75.374	63.549	42.220
	Ours	**75.725**	**63.750**	**43.842**
	Ours (w/o prior)	74.498	60.646	39.600
3	Multiplication	75.542	63.815	42.692
	Average	75.451	63.754	42.213
	Ours	**75.815**	**63.827**	**44.231**
4	Multiplication	75.578	**63.950**	42.795
	Average	75.358	63.767	42.102
	Ours	**75.668**	63.720	**44.263**

Table 3.2: The effectiveness of different label fusion methods on 2000 images sampled from SceneNet RGB-D. The large improvement on the metric of intersection over union shows that our label fusion lead to smoother predictions.

Effect of Code Prior during Semantic Optimisation

During semantic optimisation we use a zero-code regularisation term. Without this term, the optimisation may be drawn to locally consistent but incorrect semantic labels. Figure 3.12 demonstrate the necessity of this prior during semantic refinement, incorrect labels are predicted while the entropy and semantic error are low. Table 3.2 shows that the accuracy of two-view label fusion without a zero-code prior is even lower than single view prediction, underlining the importance of this prior.

3.6.4 Monocular Dense Semantic SLAM

We present example results from our preliminary full monocular dense semantic SLAM system. Due to the prior information on geometry encoded in the system, the system is very robust during initialisation and can manage pure rotational motion. The system currently runs in a sliding window manner. Figures 3.1 and 3.13

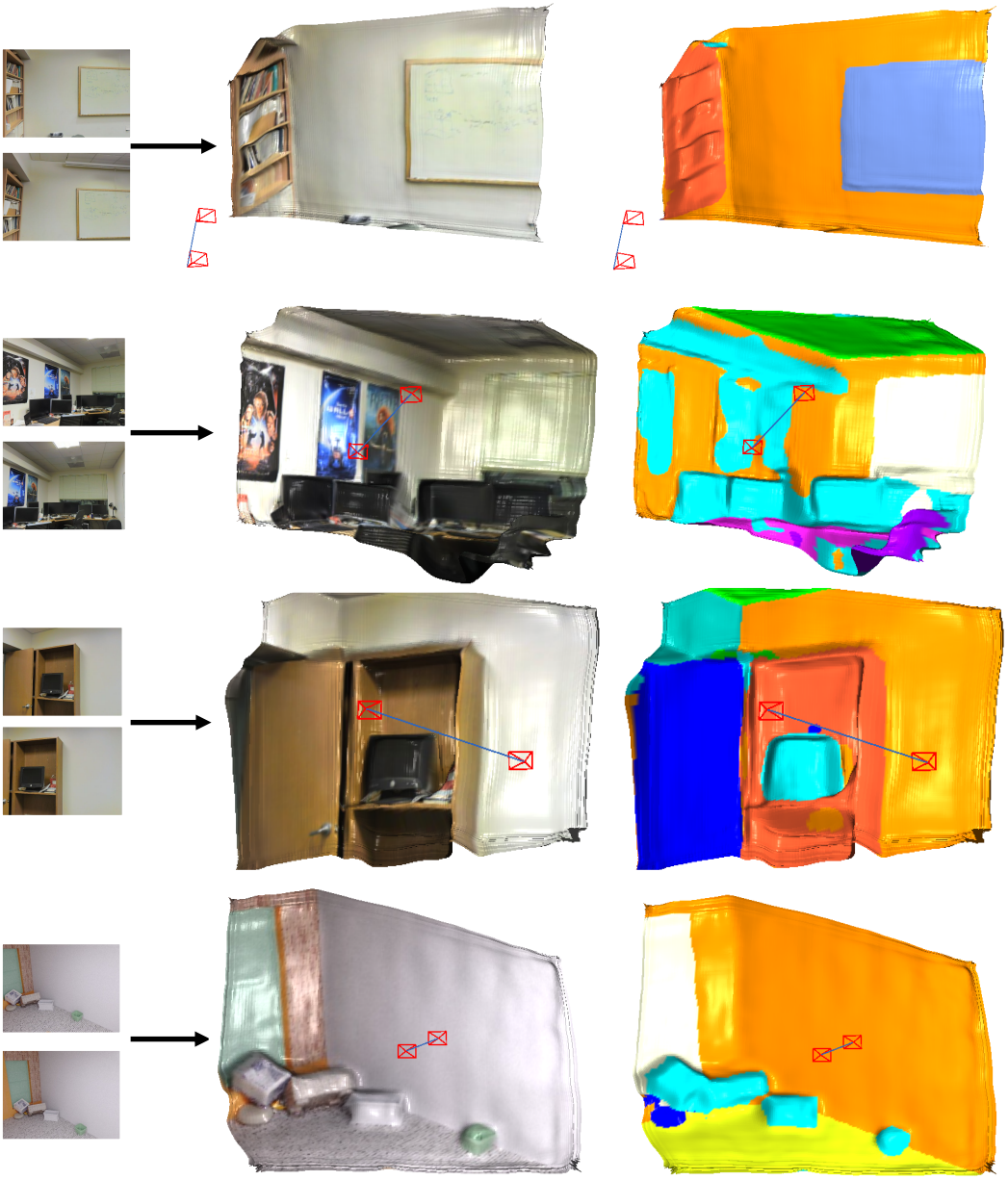


Figure 3.13: Qualitative results of two-view structure from motion on two selected frames from Stanford dataset (first 3 rows) and the SceneNet RGB-D dataset (last row). The compact representations of both semantics and geometry are (jointly) optimised with camera pose to obtain a dense map with consistent semantic labels and relative camera motion.

Figure 3.14: Qualitative result of monocular dense semantic SLAM system in bathroom and kitchen sequences of the NYUv2 dataset.

show examples of two-view dense semantic structure from motion from different datasets. We also build a preliminary keyframe-based monocular dense semantic SLAM system, shown in Figure 3.14. A 7-frame sliding window is applied to trade-off between optimisation efficiency and reconstruction quality.

3.7 Conclusion and Future Work

In this chapter, we have shown that an image-conditioned learned compact representation can coherently and efficiently represent semantic labels. This semantic code can be optimised across multiple overlapping views to implement semantic fusion with many advantages over the usual methods which operate in a per-surface-element independent manner. As well as proving this fusion capability experimentally, we have built and demonstrated a prototype full dense, semantic mon-

ocular SLAM system based on learned codes where geometry, poses and semantics can all be jointly optimised.

It is true that our approach, like element-wise fusion methods, relies on robust data association via successful geometric reconstruction. In this chapter we have still not proven that simultaneous joint optimisation of geometry and semantics from scratch will converge, but show very promising results from a staged approach where we first jointly optimise geometry and motion to reach reliable dense correspondence, then semantics, and finally full joint optimisation. Practical dense SLAM systems usually rely on some kind of staged optimisation as well. From the rigorous probabilistic point of view, joint optimisation allows the consideration of the full correlated distribution of structure, motion and semantics, and we certainly hope and plan to keep moving towards systems which work in this way. We believe that our work here on probabilistic and efficient semantics representation is a step in the right direction.

However, there are still a number of unsolved problems which were encountered in this chapter and will direct the research in later chapters and future work.

Current code-based representation is attached to keyframes, and hence is a 2D representation focusing on local scene properties. In the longer term, it is promising to distribute the encoding ability of such compact representation to larger-scale scene properties such as scene-level geometry and semantics, or to more specific ones such as object instances [Sucar et al., 2020, Li et al., 2020], which could possibly be used to compose an abstract semantic scene graphs where each concept can be described as an efficient latent code.

In addition, exploring the possibility of using semantics to refine geometry with intrinsic semantic error terms is also exciting future research direction. Currently the zero code regularisation acts as a strong prior to avoid trivial solutions during optimisation and semantics mainly benefits from geometry, while the opposite

direction is still not well investigated.

Like other supervised learning systems, SceneCode still relies on a collection of paired training samples of RGB image, depth map and dense semantic class labels. In addition to costly dense semantic annotations, the prior information learned by a CVAE is also limited to the distribution of training data, though the generalisation gap can be mitigated to some extent by the capabilities of code optimisation at inference time. This asks us to seek more diverse datasets or efficient learning strategies for real world applications. One possible solution is to adopt weakly or semi-supervised learning to reduce the labelling demand and better leverage the correlation between these related dense prediction vision tasks [Zamir et al., 2018, Zamir et al., 2020].

In the next chapter, we will demonstrate a scene-specific implicit scene representation to alleviate the generalisation gap and reliance on external datasets.

CHAPTER 4

Semantic-NeRF

Contents

Parts of this Chapter appear in: Zhi, S., Laidlow, T., Leutenegger, S. and Davison, A. (2021). In-Place Scene Labelling and Understanding with Implicit Scene Representation.*In Proceedings of the International Conference on Computer Vision (ICCV)*. [Zhi et al., 2021a]

4.1 Introduction

Enabling intelligent agents, such as indoor mobile robots, to plan context-sensitive actions in their environment requires both a geometric and semantic understanding of the scene. Machine learning methods have proven to be valuable in both geometric and semantic prediction tasks, but the performance of these methods suffers when the distribution of the training data does not match the scenes observed at test-time. Though the issue can be mitigated by gathering costly annotated data or semi-supervised learning, it is not always feasible in open-set scenarios with various known and unknown classes. For this reason, it is advantageous to have methods that can self-supervise. In particular, there has been recent success in using appealing scene-specific methods (e.g. NeRF [Mildenhall et al., 2020]) that implicitly represent the shape and radiance of a single scene with a neural network trained from scratch using only images and associated camera poses. But the same fully self-supervised approach is not possible for semantics of a novel scene because labels are human-defined properties. The best that could be achieved would be to cluster self-similar structures of a scene into categories; but some labelling would always be needed to associate these clusters with human-defined semantic classes. It is worth investigating whether scene-specific representation can be applied to semantics and enable semantic scene understanding of robots in open-set environments, i.e., attaching custom class labels to a geometric model.

The tasks of estimating the geometry of a scene and predicting its semantic labels are strongly related, as parts of a scene that have similar shape and appear-

ance are more likely to belong to the same semantic category than those which differ greatly, which has also been shown in work on multitask learning [Zamir et al., 2018, Liu et al., 2019] where networks that simultaneously predict both shape and semantics perform better than when the tasks are tackled separately. In this work we show how to design a scene-specific network for joint geometric and semantic prediction and train it on images from a single scene with only weak semantic supervision and no geometric supervision. Specifically, we extend neural radiance fields (NeRF) to jointly encode semantics with appearance and geometry, i.e. Semantic-NeRF, so that complete and accurate 2D semantic labels can be achieved using a small amount of in-place annotations specific to the scene. Because our single network must generate both geometry and semantics, the correlation between these tasks means that semantics prediction can benefit from the smoothness, coherence and self-similarity learned by self-supervision for geometry, enabling sparse labels to efficiently propagate. We show the benefit of this approach when labels are either sparse or very noisy in room-scale scenes. In addition, multi-view consistency is inherent to the training process and enables the network to produce accurate semantic labels of the scene, including for views that are different from any in the input set.

Our system takes as input a set of RGB images with associated known camera poses. We also supply some partial or noisy semantic labels for the images, such as ground truth labels for a small fraction of the images, or noisy or coarse label maps for a higher number of images. We train our network to jointly produce implicit 3D representations of both the geometry and semantics for the whole scene, and evaluate our system both quantitatively and qualitatively on scenes from the Replica dataset [Straub et al., 2019], and qualitatively on real-world scenes from the ScanNet dataset [Dai et al., 2017a]. Generating dense semantic labels for a whole scene from partial or noisy input labels is important for practical applications, like when a robot encounters a new scene and either only a small amount of

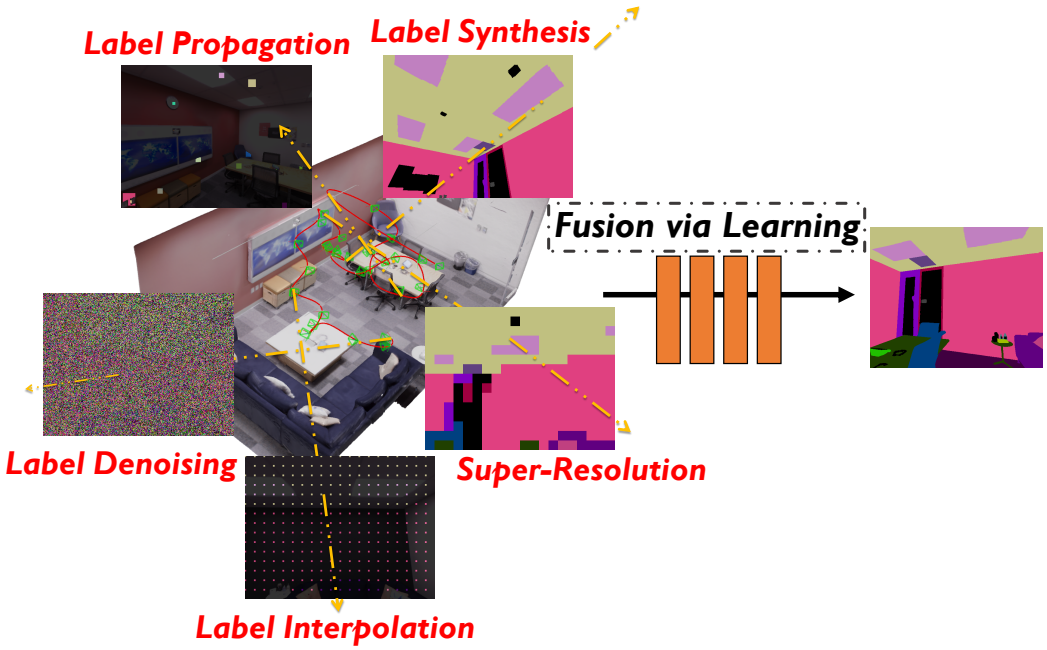


Figure 4.1: Neural radiance fields (NeRF) jointly encoding appearance and geometry contain strong priors for segmentation and clustering. We build upon this to create a scene-specific 3D semantic representation, Semantic-NeRF, and show that it can be efficiently learned with in-place supervision to perform various potential applications.

in-situ labelling is feasible, or only an imperfect single-view network is available. We demonstrate advantageous properties of Semantic-NeRF in various interesting applications such as an efficient scene labelling tool, novel semantic view synthesis, label denoising, super-resolution, label interpolation and multi-view semantic label fusion in visual semantic mapping systems.

4.2 Related Work

Most existing 3D semantic mapping and understanding systems work by attaching (fused) semantic labels to a 3D geometric representation created by a standard reconstruction method. For example, [Hermans et al., 2014] uses point clouds, [McCormac et al., 2017a] and [Runz et al., 2018] use surfels, [Narita et al., 2019]

uses voxels, and [McCormac et al., 2018] uses signed distance fields. These classical 3D geometric representations are all limited in their ability to efficiently represent fine details in complex topologies. Volumetric representations, for example, have a convenient structure for parallel processing or use with convolutional neural networks, but suffer from large memory requirements due to discretisation that ultimately limits the resolution it can represent.

To help overcome these limitations, many learning-based representations have been developed. Code-based representations, for example, use the latent code of an auto-encoder as a compact representation of the scene. However, as discussed in Section 1.2, although trained with depth maps or camera poses, they are still view-based representations and lacked true awareness of 3D geometry.

There has been much promising recent work on using neural implicit scene representations. As these are continuous representations, they can easily handle complicated topologies and do not suffer from discretisation error, with the actual representative resolution depending on the capacity of the neural network used. The Scene Representation Network (SRN) [Sitzmann et al., 2019b] was one of the first methods to use a multi layer perceptron (MLP) as the neural representation of a learned scene given a collection of images and associated poses. DeepSDF [Park et al., 2019] and DIST [Liu et al., 2020] used deep decoders to learn implicit signed distance functions (SDFs) of various shape instances of the same class, and Occupancy Networks [Mescheder et al., 2019, Peng et al., 2020] learned an implicit 3D occupancy function for shapes or large scale scenes given 3D supervision. Kohli *et al*. [Kohli et al., 2020] also proposed to learn a joint implicit representation of appearance and semantics for 3D shapes on top of an SRN using a linear segmentation renderer. After being trained in a two-step semi-supervised manner, the network can synthesise novel view semantic labels from either colour or semantic observations.

The methods mentioned above involve extensive pre-training on collections of data to learn priors about the shapes or scenes they are used to represent. Although promising generalisation capability has been shown across different instances or scenes, it is not always possible to get adequate data for various unseen environments. The alternative is a scene-specific representation which requires minimum in-place labelling effort.

NeRF [Mildenhall et al., 2020] and other systems based on it [Zhang et al., 2020, Martin-Brualla et al., 2021, Trevithick and Yang, 2021, Srinivasan et al., 2021] use MLPs to overfit input from a single bounded scene and act as an implicit volumetric representation for realistic view-synthesis. In 2D representation/view-based Semantic SLAM system such as SceneCode [Zhi et al., 2019], if we have a very close look to a desk, it is likely to be mistakenly recognised as floor. In 3D-aware representation-based system such as NeRF [Mildenhall et al., 2020], since we have access to 3D scene space, once we know that certain 3D position has certain semantic class, we should still predict the correct semantic even the 2D view point becomes challenging.

In this work, we treat NeRF as a powerful scene-specific 3D implicit representation, and extend it to include semantic representation which can be efficiently learned from sparse or noisy annotations (Figure 4.1).

4.3 Method

4.3.1 Semantic-NeRF

We now show how to extend NeRF to jointly encode appearance, geometry and semantics. As shown in Figure 4.2, we augment the original NeRF by appending a segmentation renderer before injecting viewing directions into the MLP.

We formalise semantic segmentation as an inherently *view-invariant* function

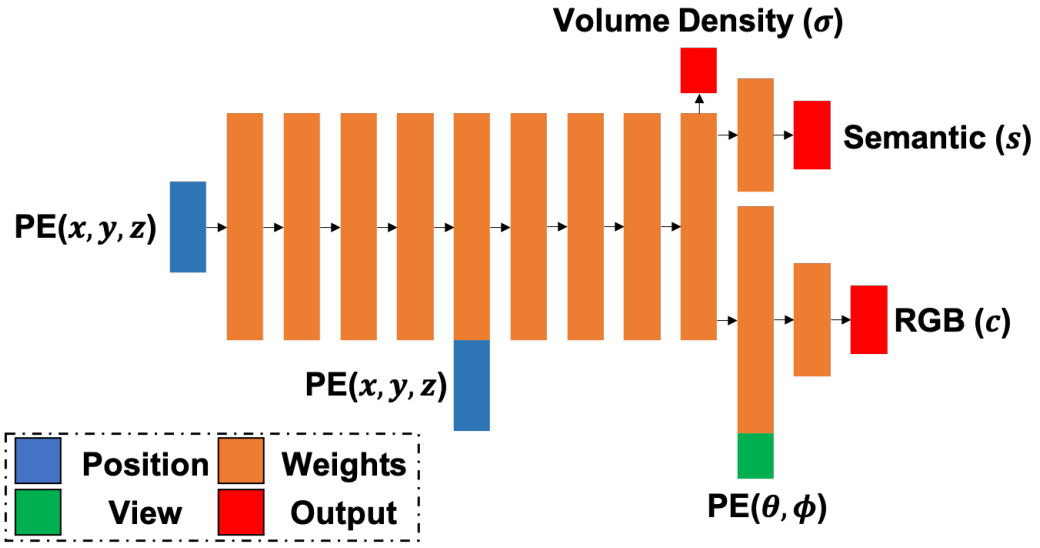


Figure 4.2: Semantic-NeRF network architecture. All fully connected layers have 256 neurons and ReLU activations except those with 128 neurons before semantics and radiance predictions. 3D position (x, y, z) and viewing direction (θ, ϕ) are fed into the network after positional encoding. Volume density σ and semantic logits $\mathbf{s}$ are functions of 3D position only while colours $\mathbf{c}$ additionally depend on viewing direction.

that maps only a world coordinate $\mathbf{x}$ to a distribution over C semantic labels via pre-softmax semantic logits $\mathbf{s}(\mathbf{x})$:

$$\mathbf{c} = F_\Theta(\mathbf{x}, \mathbf{d}), \quad \mathbf{s} = F_\Theta(\mathbf{x}), \tag{4.1}$$

where F_Θ represents the learned MLPs.

Similar to Equation 2.15, the approximated expected semantic logits $\hat{\mathbf{S}}(\mathbf{r})$ of a given pixel in the image plane can be written as:

$$\hat{\mathbf{S}}(\mathbf{r}) = \sum_{k=1}^{K} \hat{T}(t_k)\,\alpha\big(\sigma(t_k)\delta_k\big)\mathbf{s}(t_k), \tag{4.2}$$

$$\text{where} \quad \hat{T}(t_k) = \exp\left(-\sum_{k'=1}^{k-1} \sigma(t_k)\delta_k\right), \tag{4.3}$$

with $\alpha(x) = 1 - \exp(-x)$ and $\delta_k = t_{k+1} - t_k$ is the distance between adjacent sample points. Semantic logits can then be transformed into multi-class probabilities through a softmax normalisation layer.

4.3.2 Network Training

We train the whole network from scratch under photometric loss L_p and semantic loss L_s:

$$L_p = \sum_{\mathbf{r}\in\mathscr{R}} \Big[\big\| \hat{\mathbf{C}}_c(\mathbf{r}) - \mathbf{C}(\mathbf{r}) \big\|_2^2 + \big\| \hat{\mathbf{C}}_f(\mathbf{r}) - \mathbf{C}(\mathbf{r}) \big\|_2^2 \Big], \tag{4.4}$$

$$L_s = -\sum_{\mathbf{r}\in\mathscr{R}} \Big[\sum_{c=1}^{C} p^c(\mathbf{r}) \log \hat{p}_c^c(\mathbf{r}) + \sum_{c=1}^{C} p^c(\mathbf{r}) \log \hat{p}_f^c(\mathbf{r}) \Big], \tag{4.5}$$

where $\mathscr{R}$ are the sampled ray batches within each optimisation step, and $\mathbf{C}(\mathbf{r})$, $\hat{\mathbf{C}}_c(\mathbf{r})$ and $\hat{\mathbf{C}}_f(\mathbf{r})$ are the ground truth, coarse volume predicted and fine volume predicted RGB colours for ray $\mathbf{r}$, respectively. Similarly, p^c, $\hat{p}_c^c$ and $\hat{p}_f^c$ are the multi-class semantic probability at class l of the provided ground truth map, coarse volume and fine volume predictions for ray $\mathbf{r}$, respectively. L_s is chosen as a multi-class cross-entropy loss to encourage the rendered semantic labels to be consistent with the provided labels, whether these are ground-truth, noisy or partial observations. Hence, the total training loss L is:

$$L = L_p + \lambda L_s, \tag{4.6}$$

where λ is the weight of the semantic loss and is set to 0.04 to balance the magnitude of both losses [Kohli et al., 2020]. In practice we find that actual performance is not sensitive to λ value and setting λ to 1 gives us similar performance. These photometric and semantic losses naturally encourage the network to generate multi-view consistent 2D renderings from the underlying joint representation.

4.3.3 Implementation

A scene-specific semantic representation is obtained by training the network from scratch for each scene individually. We use setup and hyper-parameters similar to [Mildenhall et al., 2020]. Specifically, we use hierarchical volume sampling to jointly optimise coarse and fine networks, where the former provides importance

sampling bias so that the latter can distribute more samples to positions likely to be visible. At each optimisation step we randomly pick one image from the training pool and randomly select a ray batch of 1024 rays due to memory limitation. The number of samples per ray through the coarse network N_c and fine network N_f is set to 64 and 192 ($64 + 128$), respectively.

Axis-aligned positional encoding $\gamma(\cdot)$ of length 10 and 4 are applied to 3D positions and viewing directions and therefore the length of inputs after PE is $L_{\gamma(\mathbf{x})} = 2 * 3 * 10 = 60$ and $L_{\gamma(\mathbf{d})} = 2 * 3 * 4 = 24$. In addition, since we have no depth information, we simply set the bounds of ray sampling to 0.1m and 10m respectively across experiments without careful tuning to span indoor scenes. Note that a well-adjusted bound per scene or frame will potentially take better advantage of sampling budget, which is not the scope of this work and can be part of future work.

Training images are resized to 320x240 for all the experiments. We implement our model in PyTorch [Paszke et al., 2019] and train it on a single RTX2080-Ti GPU with 11GB memory. We train the neural network using the Adam optimiser [Kingma and Ba, 2015] with a learning rate of 5e-4 for 200,000 iterations (approximately 10-11 hours).

4.4 Experiments and Applications

After training on colour images and semantic labels with associated poses, we obtain a scene-specific implicit 3D semantic representation. We evaluate its effectiveness quantitatively by projecting the 3D representation back into 2D image space where we have direct access to explicit ground truth data. We aim to show the benefits and promising applications of efficiently learning such a joint 3D representation for semantic labelling and understanding. We kindly urge readers to inspect more qualitative results in the project page https://shuaifengzhi.com/

Semantic-NeRF/ and video https://youtu.be/FpShWO7LVbM.

4.4.1 Indoor Scene Datasets and Data Preparation

Replica Replica [Straub et al., 2019] is a reconstruction-based 3D dataset of 18 high fidelity scenes with dense geometry, HDR textures and semantic annotations. We use the Habitat simulator [Savva et al., 2019] to render RGB colour images, depth maps and semantic labels from randomly generated 6-DOF trajectories similar to hand-held camera motions. We follow the procedure from SceneNet RGB-D [McCormac et al., 2017b], and lock the roll angle with the camera up-vector pointing along the y-axis.

We use the provided 88 semantic classes from Replica in scene-specific experiments and also manually map these labels to the popular NYUv2-13 definition [Silberman et al., 2012, Eigen and Fergus, 2015] in Section 4.4.4 for multi-view semantic label fusion, following the mapping convention from ScanNet [Dai et al., 2017a]. For each Replica scene of rooms and offices, we render 900 images at resolution 640x480 using the default pin-hole camera model with 90 degree horizontal field of view. We sample every 5th frame from the sequence to compose the training set and also sample intermediate frames to make the test set.

ScanNet ScanNet [Dai et al., 2017a] is a large-scale real-world indoor RGB-D video dataset of 2.5M views in 1513 scenes with rich annotations including semantic segmentation, camera poses and surface reconstructions. We train our Semantic-NeRF on ScanNet scenes using only the provided colour images, camera poses and 2D semantic labels. The sequences in each scene are evenly sampled so that the total amount of training data is roughly 300 frames. During experiments we select several indoor room-scale scenes and train one Semantic-NeRF per scene using posed images and semantic labels from the NYUv2-40 definition.

4.4.2 Semantic Neural Radiance Fields

In this work we introduce semantic neural radiance fields, though there are no physical semantic photons which makes its definition less intuitive, we argue that the volume rendering equation discretises an integral over the radiance field as a weighted summation. The weights can be interpreted as termination probabilities, which means a higher weight is given to the first intersected surface. This interpretation shows how we can use the volume rendering equation to render any field defined in 3D, such as colour, depth or semantics. As will be demonstrated in later experiments, the advantages of learning such implicit semantic fields over explicit ones lie in the compactness and efficiency in representing complex shapes as well as the baked-in multi-view consistency, which alleviates the discretisation error and post-optimisation in Chapter 3.

We check the influence of semantics on appearance and geometry by quantitatively computing the quality of rendered RGB images and depth maps on Replica scenes with and without semantic prediction enabled. Peak signal-to-noise ratio (PSNR) is used to measure the quality of the rendered colour images and the metrics used to evaluate the 2D depth maps are shown in Table 4.1.

As shown in Table 4.2, there is no clear difference which suggests that the current network has the capacity to learn these tasks jointly. Note that we might expect that significant high quality semantic labelling information could feasibly improve reconstruction quality, but in this paper we are focused on how geometry can help semantics in the opposite situation where semantic labelling is sparse or noisy.

4.4.3 Semantic View Synthesis with Sparse Labels

We first train our Semantic-NeRF framework for novel view semantic label synthesis using all available RGB images with camera poses and corresponding se-

2D Depth Metrics	
Abs Rel	$\frac{1}{n}\sum\|d-d_{gt}\|/d_{gt}$
Abs Diff	$\frac{1}{n}\sum\|d-d_{gt}\|$
Sq Rel	$\frac{1}{n}\sum\|d-d_{gt}\|^2/d_{gt}$
RMSE	$\sqrt{\frac{1}{n}\sum\|d-d_{gt}\|^2}$
$\delta < 1.25^i$	$\frac{1}{n}\sum(\max(\frac{d}{d_{gt}}, \frac{d_{gt}}{d}) < 1.25^i)$

Table 4.1: Definitions of depth metrics used in Table 4.2. n is the number of valid depth pixels, d and d_{gt} are rendered depths at testing poses and corresponding ground truth depths, respectively.

Network Set-up	Depth							RGB
	AbsRel↓	AbsDiff↓	SqRel↓	RMSE↓	$\delta < 1.25$↑	$\delta < (1.25)^2$↑	$\delta < (1.25)^3$↑	PSNR↑
W/ Semantics	0.017	0.032	0.007	0.096	0.993	0.997	0.998	32.27
W/O Semantics	0.018	0.032	0.009	0.102	0.993	0.996	0.998	32.80

Table 4.2: Quantitative evaluation of effects of predicting semantics on appearance and geometry on Replica dataset.

mantic labels (i.e., 180 images) from a randomly generated sequence of a certain scene. This fully-supervised setup acts as an upper bound on the semantic segmentation performance of Semantic-NeRF given abundant labelled training data.

However, in practice it is expensive and time-consuming to acquire accurate dense semantic annotations for all observed images in a scene. Considering the redundancy in semantic labels among overlapping frames, we borrow the idea of key-framing from SLAM systems and hypothesise that providing labels for only selected frames should be enough to train the semantic representation efficiently. We choose key-frames by evenly sampling from sequences and train the networks from scratch with semantic labels only coming from those selected key-frames, while the synthesis performance is always evaluated on all test frames.

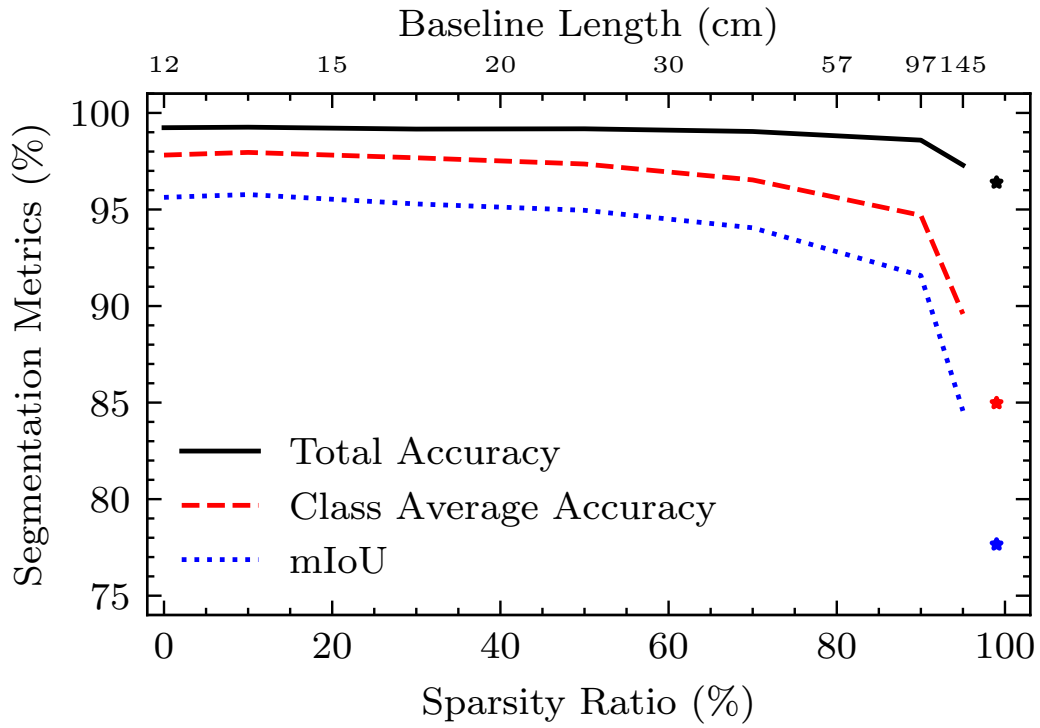


Figure 4.3: Quantitative performance of Semantic-NeRF trained on Replica with sparse semantic labels. Sparsity ratio is the percentage of frames dropped compared to full sequence supervision. Three standard metrics are used to evaluate semantic segmentation performance on test poses (higher is better). Performance gracefully degrades with fewer labels due to uncovered or occluded regions, indicating the possibility of efficient dense labelling from fewer annotations. Results with only two labelled key-frames ($\star$) show remarkably competitive performance. The baseline length metric shows the average camera translation distance between two spatially consecutive keyframes with semantic labelling.

Figure 4.3 validates our assumption that semantics can be efficiently learned from sparse annotations with a sparsity ratio ranging from 0% to 95%, together with the corresponding camera motion baselines as a complementary indication. Only marginal performance loss occurs when less than 10% semantic frames are used, and this is mainly caused by renderings of regions which are unobserved or occluded from key-frames. To take this even further, we manually select just two key-frames (99% sparsity ratio) from each scene to cover as much of the scene as possible. It turns out that our network, trained only with two labelled keyframes,

can render accurate labels from various viewpoints. Corresponding qualitative results of view synthesis are shown in Figure 4.4.

4.4.4 Semantic Fusion

In addition to being able to learn the semantic representation with sparse annotations due to the redundancy present in the semantic labels, another important property of Semantic-NeRF is that multi-view consistency between semantic labels is enforced by design, as we formulate volume density σ and semantic logits $\mathbf{s}$ to be only a function of 3D location $\mathbf{x}$.

In semantic mapping systems (e.g. [Sünderhauf et al., 2017, McCormac et al., 2017a, Narita et al., 2019]), multiple 2D semantic observations are integrated into a 3D map or target frames to produce a more consistent and accurate semantic segmentation. Multi-view consistency is the key concept and motivation in semantic fusion, and the training process of Semantic-NeRF itself can be seen as a multi-view label fusion process. Given multiple noisy or partial semantic labels, the network can fuse them into a joint implicit 3D space so that we can extract denoised labels when we re-render the semantic labels from the learned representation back to input training frames.

We show the capability of Semantic-NeRF to perform multi-view semantic label fusion under a number of different scenarios: pixel-wise label noise, region-wise label noise, low-resolution dense or sparse labelling, partial labelling, and using the output of an imperfect CNN.

Semantic Label Denoising

Labels with Pixel-wise Noise We corrupt ground-truth training semantic labels by adding independent pixel-wise noise. Specifically, we randomly select a fixed portion of pixels per training frame and randomly flip their labels to arbitrary ones

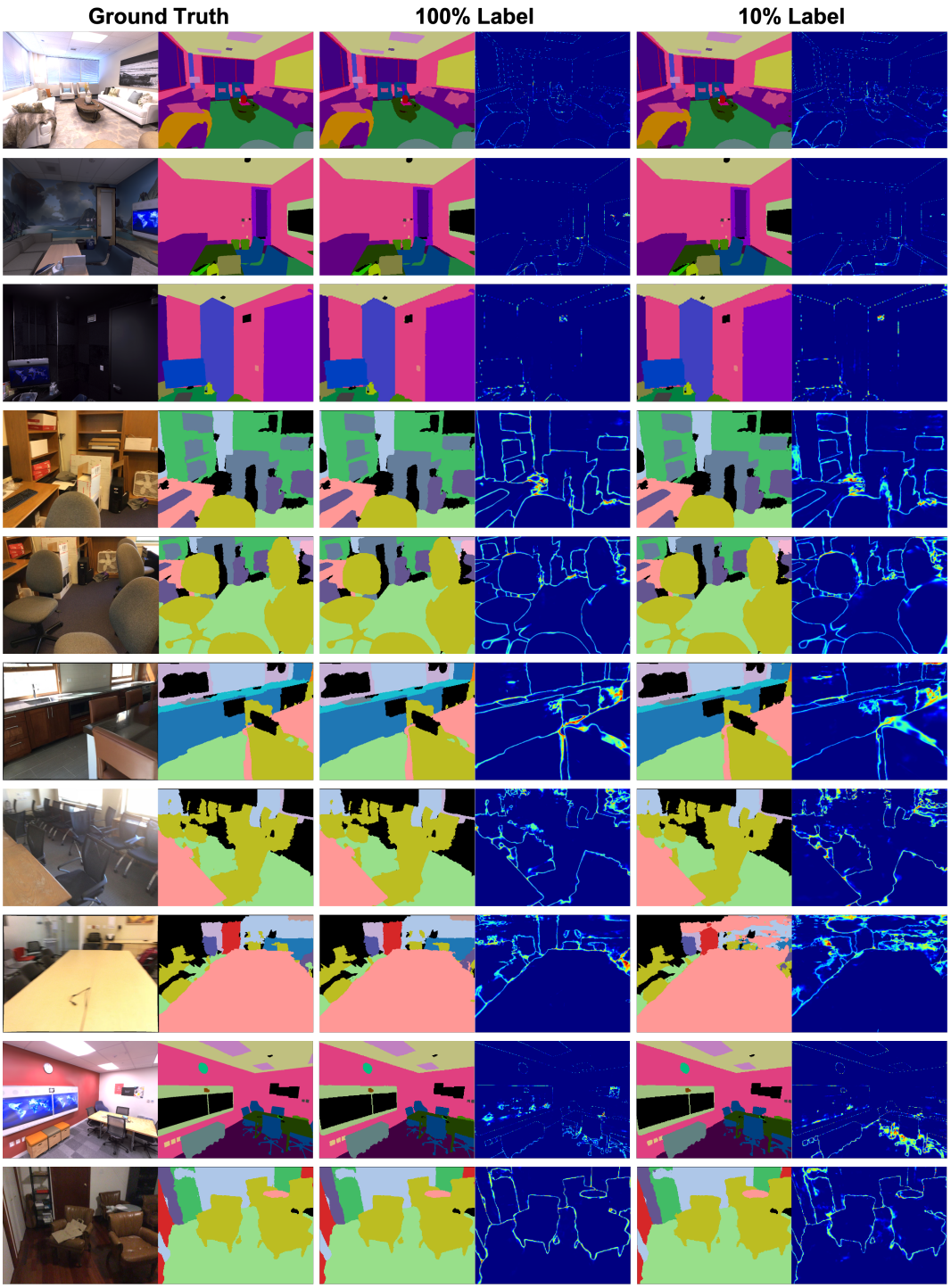


Figure 4.4: Synthesised semantic labels at testing poses given 100% and 10% of ground truth labels during training. From left to right we show the ground truth colour and semantic images for reference, and rendered semantic labels and their information entropy given 100% and 10% supervision, respectively. Bright parts of the entropy map match well to object boundaries or ambiguous/unknown regions in the corresponding training set-up.

(including the void class). After training using only these noisy labels, we obtain denoised semantic labels by rendering back to the same training poses.

Figure 4.5 and 4.6 shows qualitative results from label denoising. When 90% of training pixels are randomly flipped, and it is difficult even for a human to recognise the underlying structure of the scene, the denoised labels still retain accurate boundaries and detail, especially for fine structures. Compared with Figure 4.4, the entropy in this denoising task is higher because the noisy training labels lack the multi-view consistency of clean ones. In addition, regions with void class tend to have the highest uncertainty since noisy pixels in void regions are not optimised during training. Quantitative results shown in Table 4.3 also confirm that accurate denoised labels are obtained after training-as-fusion.

While pixel-wise denoising with such severe corruption is not a realistic application, it is still a very challenging task and, more importantly, highlights our key observation that training itself is a fusion process which enables coherent renderings benefiting from the internal consistency and smoothness of implicit joint representation.

Labels with Region-wise Noise We further validate the effectiveness of semantic consistency by randomly flipping the class labels of certain whole instances instead of pixels in the label maps. This is a better simulation of the behaviour of real single-view CNNs because a whole object can easily be labelled as a similar but incorrect class from an obstructed or ambiguous view.

We choose Replica Room_2 containing 8 instances of chairs as the testing scene. For each chair instance, we compute the occupied area ratio (i.e., ratio of the number of pixels belonging to that instance to the total number of pixels in the image for each ground truth label frame) and then sort the label maps in the sequence based on this occupied area ratio. Two criteria are used for selecting frames in

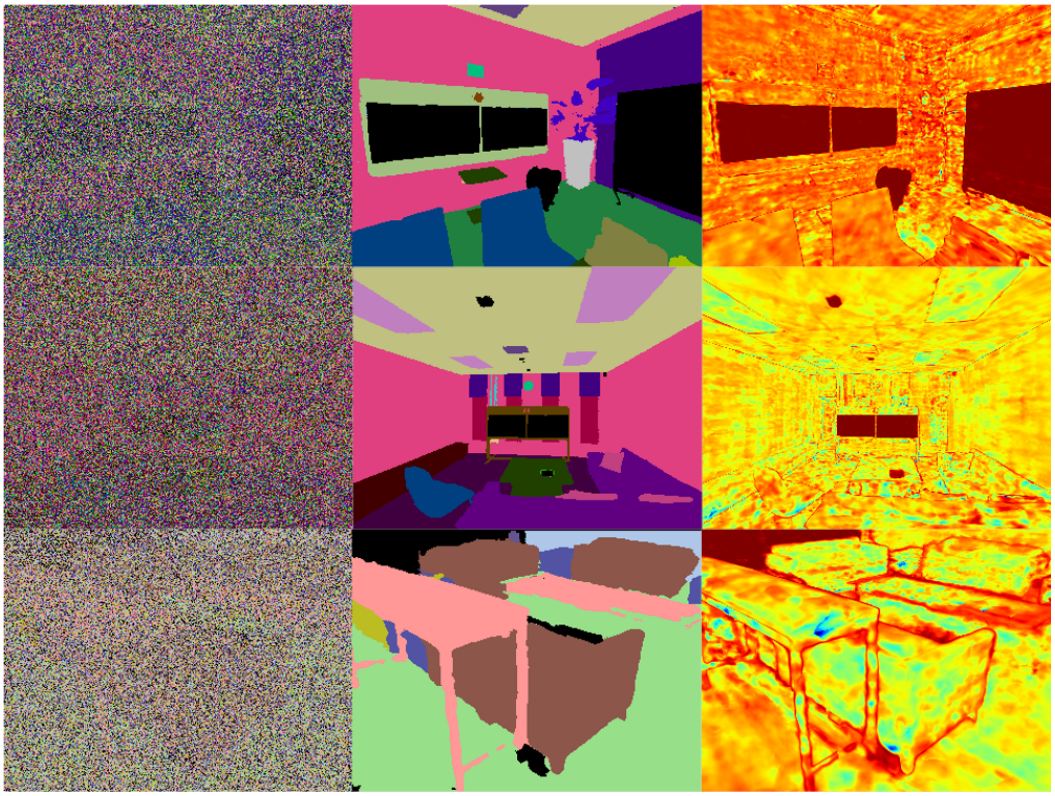


Figure 4.5: Qualitative results for semantic denoising. Even when 90% of all training labels are randomly corrupted, we can recover an accurate denoised semantic map. From left to right are noisy training labels, denoised labels rendered from the same poses after training, and information entropy. The overall high entropy we see in denoising tasks indicates the large inconsistency among noisy training labels.

which to randomly perturb each instance: (1) **Sort**: Select label maps with the least occupied area ratio. The intuition for this is that frames with partial observations are more likely to be mislabelled by semantic label prediction networks due to ambiguous context. (2) **Even**: Select label maps evenly from the sorted sequences introducing more large inconsistent regions into the training process.

Figure 4.7 shows the qualitative results of the re-rendered semantic labels after training. We indeed observe that semantic labels of the chair instances can be corrected due to the enforcement of multi-view consistency during training. Table 4.3 also shows that there are steady improvements while it becomes much harder

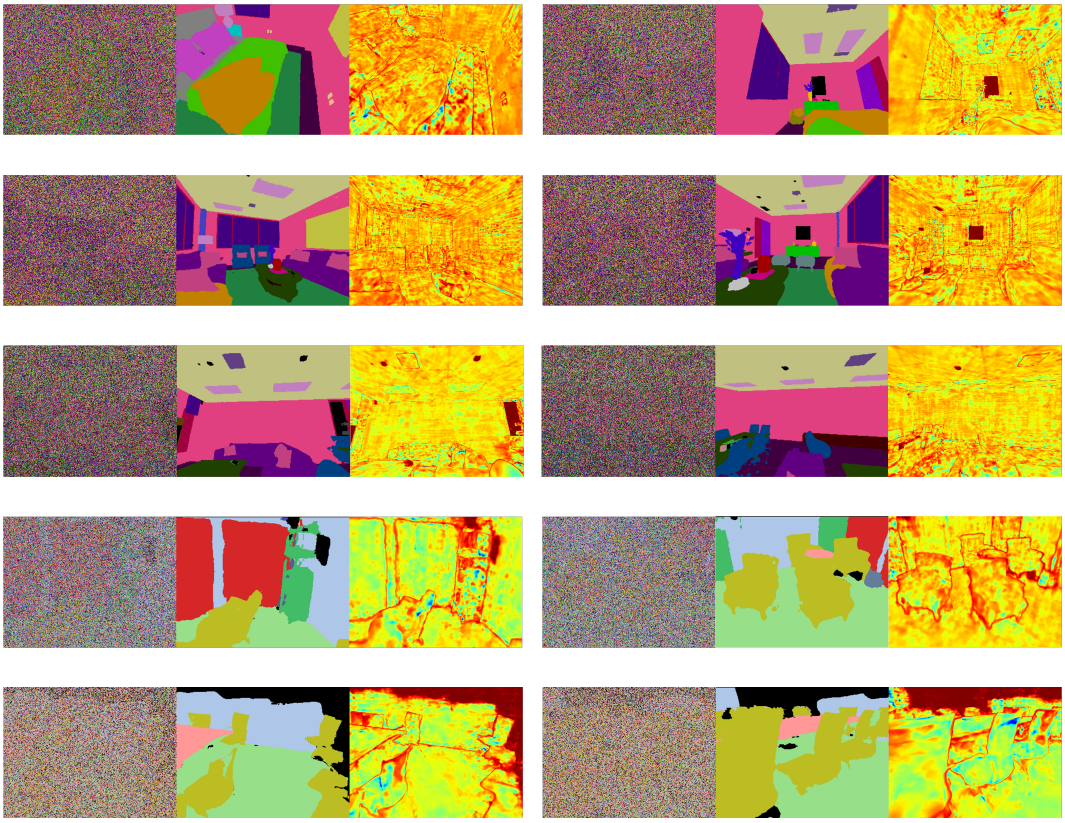


Figure 4.6: Additional results of pixel-wise semantic denoising with 90% noise ratio.

to render improved labels when a larger fraction of labels are perturbed.

Super-Resolution

Semantic label super-resolution is a useful application for scene labelling as well. In an incremental real-time semantic mapping system, a light-weight CNN predicting low-resolution semantic labels might be adopted to reduce computational cost (e.g. [Nakajima et al., 2018]). Another possible use case is in a scene labelling tool, since manual annotation in coarse images is much more efficient.

Here we show that we can train Semantic-NeRF with only low-resolution semantic information but then render accurate super-resolved semantics for either the input viewpoints or novel views. We test two different strategies to generate

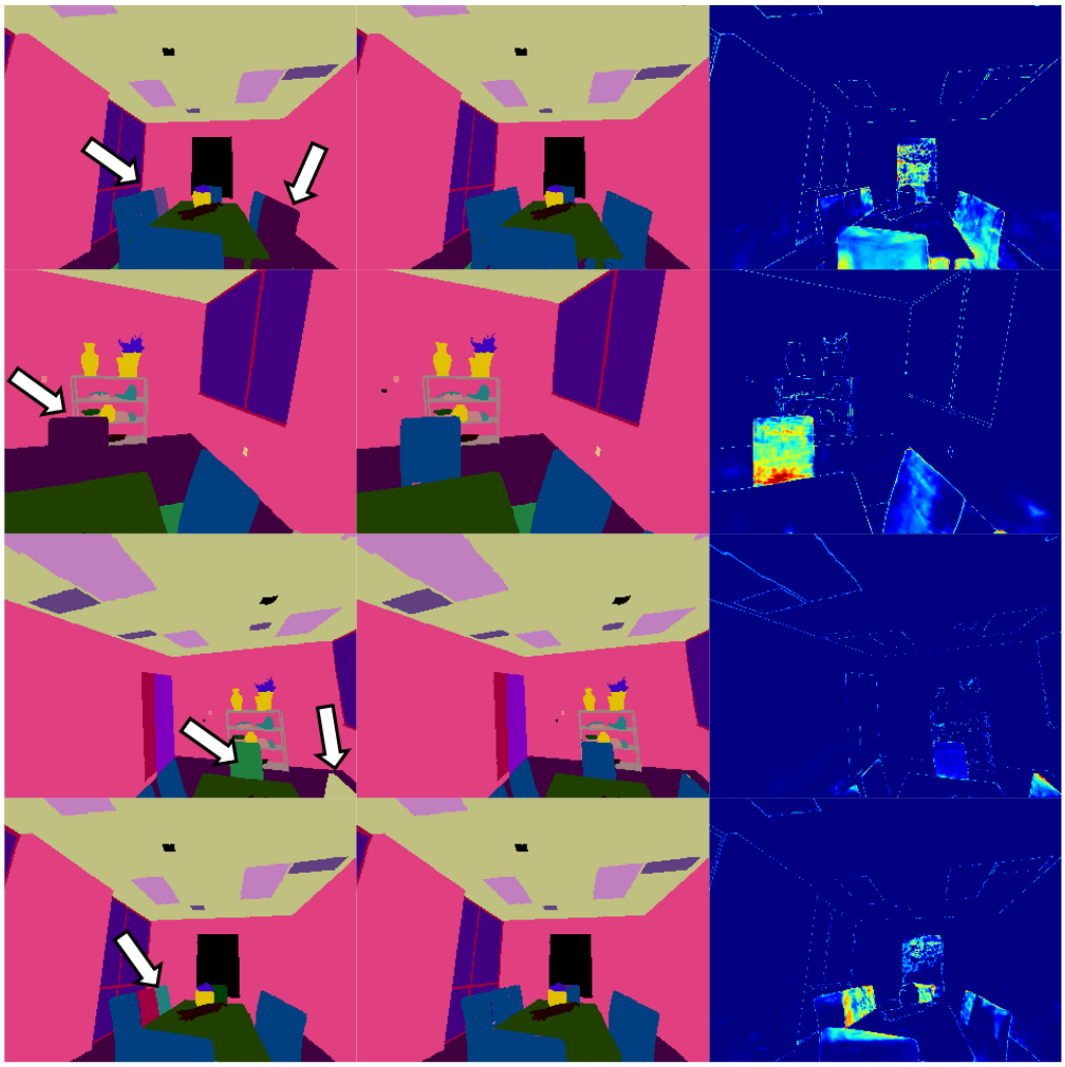
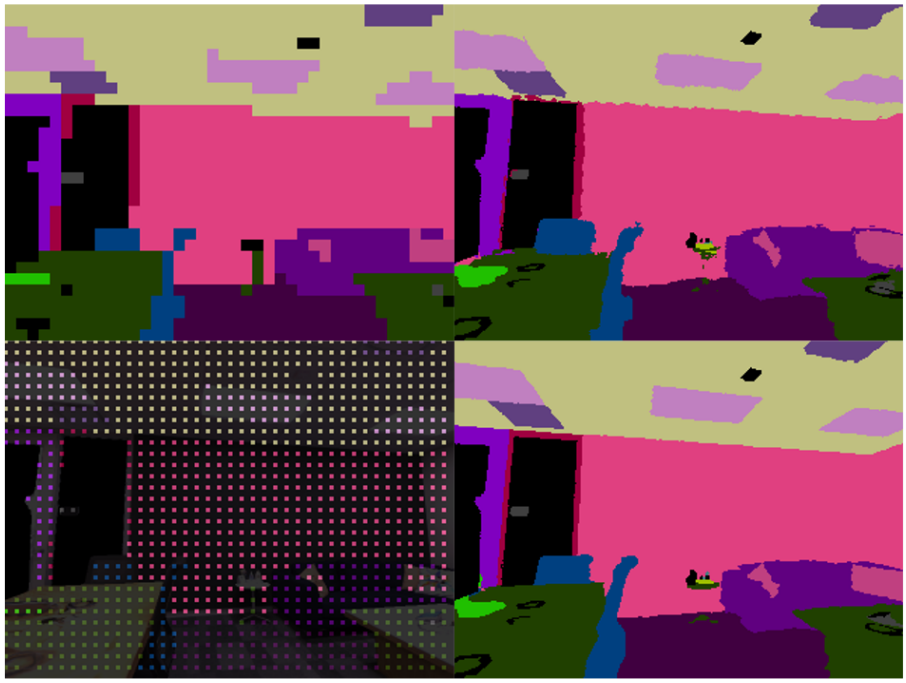
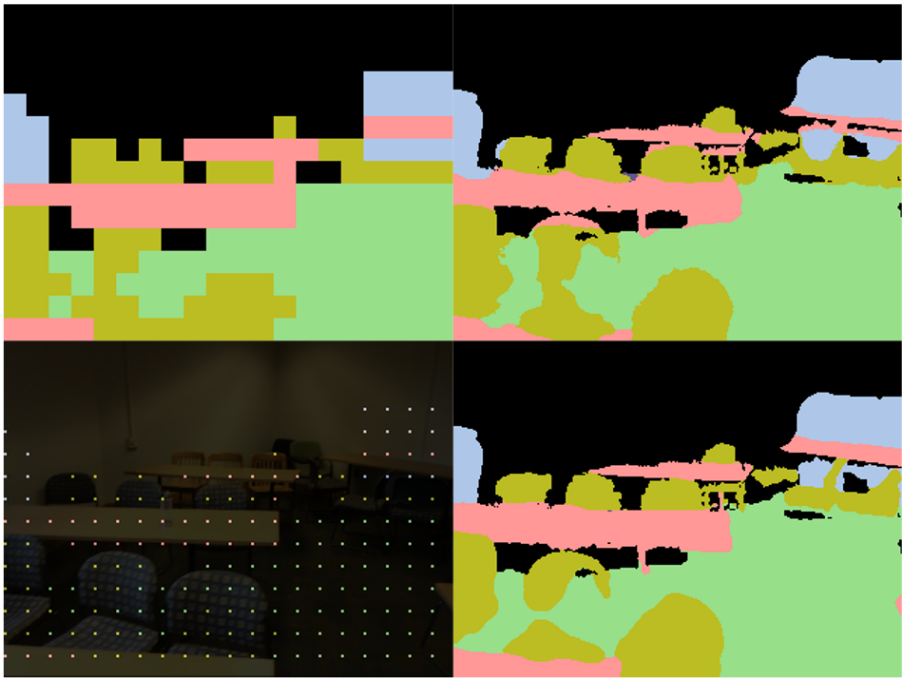


Figure 4.7: Qualitative results of rendered labels when we randomly change the training semantic class label (blue) of chair instances. From left to right: training label with region-wise noise; recovered semantic labels rendered from the same poses; and information entropy, highlighting regions with noisy predictions.

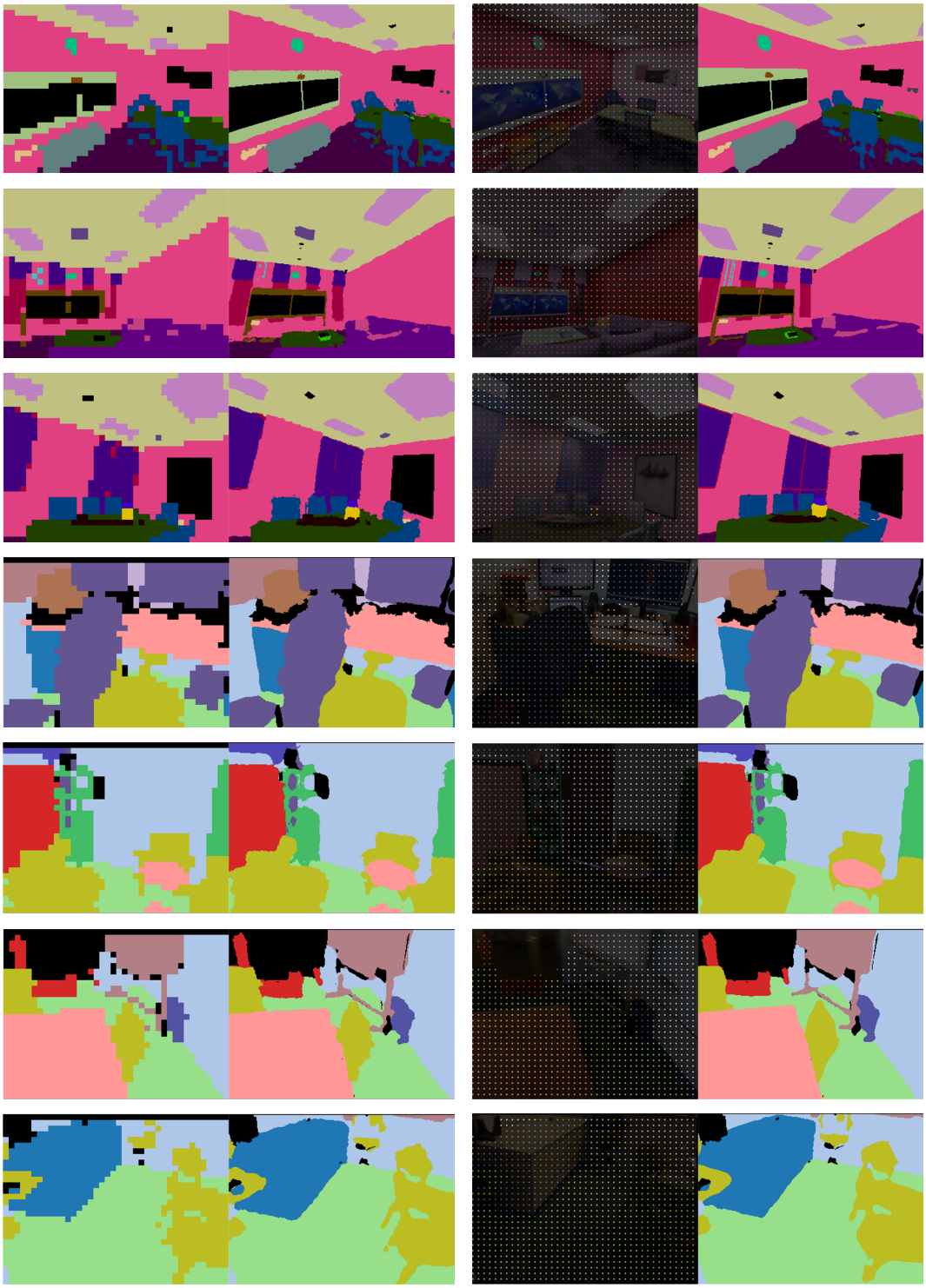


(a) Super-resolution with scale $\times 8$



(b) Super-resolution with scale x16

Figure 4.8: Qualitative results of semantic label super-resolution. We train Semantic-NeRF with only low resolution labels (interpolated or sparsely sampled) and obtain super-resolved labels by re-rendering semantics from the same poses. Left and right shows training labels and super-resolved labels, respectively. Note that the sparse labels have been zoomed-in 4 times and overlaid on top of colour images for the ease of visualisation.



(a) Super-resolution using coarse label (b) Super-resolution using sparse label

Figure 4.9: Additional qualitative results of semantic label super-resolution with scale $\times 8$.

Pixel-Wise Denoising		Metrics		
Noise Ratio		mIoU	Avg Acc	Total Acc
50%	Input Label	0.191	0.534	0.533
	Denoised Label	0.951	0.969	0.994
90%	Input Label	0.041	0.145	0.145
	Denoised Label	0.877	0.908	0.989

Region-Wise Denoising		mIoU		
Noise Ratio		30%	40%	50%
Sort	Input Label	0.866	0.842	0.793
	Denoised Label	0.895	0.893	0.803
Even	Input Label	0.741	0.692	0.684
	Denoised Label	0.796	0.747	0.733

Table 4.3: Quantitative evaluation for label denoising on Replica dataset. Noise ratio is the percentage of changed pixels per frame, and for each instance the percentage of changed frames meeting selected criterion, respectively. mIoU is used for region-wise denoising as it is more sensitive to the incorrect predictions on chair classes within the scene. Both tables are computed against clean training labels.

low-resolution training labels, with and without interpolation as shown in Figure 4.8. Given a down-scaling factor $S = 8$ for instance:

(1) All ground truth labels are down-scaled from 320×240 to 40×30 before being up-scaled back to the original size using nearest neighbour interpolation.

(2) All pixels except those from the low-resolution label maps (row and column

divisible by 8) are masked by the void class so as not to contribute to the training loss.

While method (1) uses interpolated labels to provide 'dense' supervision to a sampled ray batch but will incorrectly interpolate some pixels, method (2) provides sparse but geometrically accurate labels. We report super-resolution performance on training poses from all Replica scenes with two scales $S = 8$ and $S = 16$ in Table 4.4. Figures 4.1, 4.8 and 4.9 show examples where detailed semantic information is recovered through the fusion of many low-resolution or sparsely annotated semantic frames.

Though both setups contain the same amount of information within input labels, as can be observed, method (2) tends to reach relatively better quality and is more helpful to achieve accurate labels for fine structures at high resolution, indicating that sparsely sampled pixels can be correctly interpolated guided by underlying appearance and geometry information, while the possible misalignments between semantics and geometry in method (1) potentially lead to some degradation in performance especially on object boundaries.

The promising results in semantic label denoising and super-resolution tasks mainly benefit from the fact that a joint representation of appearance, geometry and semantics is learned implicitly by Semantic-NeRF. For example, missing semantic information in one frame may be observed in other views; corrupted or incorrect hypothesis from ambiguous context are less usual to appear than the correct one; semantic label of the same 3D position tend to be similar; semantic information of local regions with similar appearance are likely to be the same. These are all taken into consideration into the proposed joint representation.

Super-Resolution		Metrics		
Down-Scaling Factor		mIoU	Avg Acc	Total Acc
Dense	S=8	0.610	0.710	0.923
	S=16	0.433	0.535	0.855
Sparse	S=8	0.887	0.928	0.987
	S=16	0.800	0.866	0.977

Table 4.4: Quantitative evaluation of label super-resolution, with good performance with either sampled or interpolated low-resolution labels. The mIoU metric shows that sparse but geometrically accurate labels are more helpful for fine structures at high resolution.

Label Propagation

Our super-resolution experiments have shown the ability of Semantic-NeRF to interpolate rich details from low-resolution annotations. For a practical scene-annotation tool, straightforward annotations from a user in the form of clicks or scratches or strokes are desirable, and expected that those sparse clicks can expand and propagate to accurately and densely label the scene. The practical cases are when we bring an intelligent robot into a brand new environment, we allow the robots to spend some time on learning from the scenes while there are some sparse manual annotations provided by the user.

To simulate user annotations, for each class within label maps we randomly select a continuous subregion with which to apply a ground-truth label while leaving the rest unlabelled. Results in Figure 4.10 and Table 4.5 show that supervision from one single pixel per class/frame can lead to surprisingly high quality rendered labels with well preserved global and fine structure. Object boundaries are gradually refined when more supervision is available and the incremental improvements

Figure 4.10: Label propagation results using partial annotations of a single-pixel, 1% or 5% of pixels per class within frames, respectively. Accurate labels can be achieved even from single-clicks, which are zoomed-in 9 times for visualisation purposes.

from more sparse labels tend to saturate.

Multi-view Semantic Fusion of Monocular CNN Predictions

We have shown that a semantic representation can be learned from sparse or noisy or partial supervisions. Here we further validate its practical value in multi-view semantic fusion using CNN predictions.

There have been several classical pixel-wise semantic fusion approaches [Hermans et al., 2014, McCormac et al., 2017a, McCormac et al., 2018] to integrate monocular CNN predictions from multiple viewpoints to refine segmentation. For fair comparison, here we have separated out the widely-adopted multi-view

Label Propagation	Metrics		
# Labelling per Class	mIoU	Avg Acc	Total Acc
Single Click	0.602	0.937	0.908
1%	0.706	0.934	0.944
5%	0.836	0.946	0.971
10%	0.884	0.957	0.980

Table 4.5: Evaluation of label interpolation and propagation on Replica scenes using test poses. Even single-pixel supervision leads to competitive performance on the accuracy metrics, which highlights the effectiveness of the representation for interactive scene labelling.

fusion approaches from such systems. Two baseline techniques are: Bayesian fusion, where multi-class label probabilities of corresponding pixels are multiplied together and then re-normalised (e.g. [McCormac et al., 2017a]), and average fusion, which simply takes the average of all label distributions (e.g. [McCormac et al., 2018]).

To prepare training data in Replica dataset, we render two different sequences per Replica scene to cover various parts of scenes. Each sequence consists of 90 frames evenly sampled from 900 renderings of size 640×480 with semantic labels remapped to NYUv2-13 class convention.

We choose DeepLabV3+ [Chen et al., 2018b] with a ResNet-101 backbone as the CNN model for monocular label predictions. To generate decent monocular CNN predictions and avoid over-fitting, we first train DeepLab on SUN-RGBD dataset [Song et al., 2015], and then fine-tune it using data from all Replica scenes except the one chosen for training Semantic-NeRF and label fusion evaluation. We repeat this fine-tuning process and train one individual Deeplab CNN model for

each test scene.

Monocular CNN predictions of the test scene are used for two purposes: (1) training supervision for our scene-specific Semantic-NeRF model; (2) monocular predictions (per-pixel dense softmax probabilities) for baseline multi-view semantic fusion methods. We train Semantic-NeRF using posed colour images together with CNN-predicted labels for 200,000 steps and then re-render the fused semantic labels back to the training poses as fusion results.

It is important to note that both baseline fusion techniques require depth information to compute the dense correspondences between frames while ours only requires posed images. We report the average performance across all testing scenes in Table 4.6, in which ground truth depth maps are used for the two baseline approaches to represent a 'best case scenario'. Our method achieves the highest improvement across all metrics, showing the effectiveness of our joint representation in label fusion.

4.4.5 Ablation Studies on Positional Encoding

Axis-aligned positional encoding (PE) of 3D positions are used in this paper as discussed in Section 4.3.3 and the length of positional encoding L relates to the maximum frequency used and affects the rendering quality.

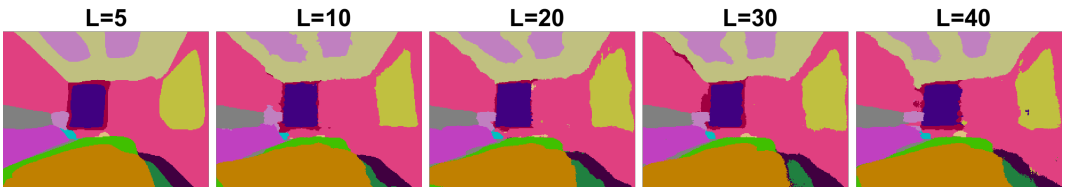


Figure 4.11: Label propagation results using partial annotations of a single-pixel with various positional encoding length.

We show results of label propagation from a single click in Replica Room_1 with L ranging from 5 to 40, shown in Table 4.7 and Figure 4.11. Using only

Semantic Fusion	mIoU	Avg Acc	Total Acc
Monocular	0.659	0.763	0.855
Bayesian Fusion *	0.668	0.764	0.865
Average Fusion *	0.586	0.703	0.814
Bayesian Fusion †	0.666	0.761	0.862
Average Fusion †	0.586	0.708	0.808
NeRF-Training (Ours)	**0.680**	**0.772**	**0.870**

* Using ground truth depth for data association.
† Using learned depth of Semantic-NeRF for data association.

Table 4.6: Comparison of multi-view semantic label fusion methods. Our approach relying on consistency of scene representations outperforms baselines aided with depth maps.

L	mIOU	Avg Acc	Total Acc
5	54.74	91.72	85.95
10	**61.23**	93.21	**89.81**
20	57.57	**93.59**	89.12
30	58.36	92.49	88.51
40	58.07	92.65	87.90

Table 4.7: Quantitative evaluation of various positional encoding length in label propagation task on Replica Room_1.

low-frequency components ($L = 5$) leads to over-smoothed 2D renderings, while using high-frequency ones ($L = 40$) leads to noisy interpolations, which aligns with findings in [Mildenhall et al., 2020, Tancik et al., 2020]. L of 10 empirically performs the best.

We further check the effect of raw 3D coordinates within positional encoding. By default, all experiments in this chapter have raw input xyz value concatenated with its positional encoding. We perform its ablation study on single click label propagation using all Replica scenes with and without raw xyz, respectively. As shown in Table 4.8, there is no clear difference between two set-ups and including raw xyz value performs better in mIoU metric and comparably in accuracy metrics, indicating that the raw xyz value encourages the final segmentation to be locally smooth and coherent.

PE Set-up	mIOU	Avg Acc	Total Acc
w/ xyz (default)	59.34	93.30	90.02
w/o xyz	57.62	93.22	89.87

Table 4.8: Ablation study of raw xyz value in positional encoding on the Replica dataset.

4.4.6 Semantic 3D Reconstruction from Posed Images

After training Semantic-NeRF with in-place semantic annotation, we can also extract an explicit 3D scene from it to inspect the implicit 3D representation.

Geometric meshes are extracted by first querying the MLP on dense 3D grids of the scene and then applying marching cubes. The attached semantic texture is rendered by treating the *negative* normal direction of vertices in the mesh as the ray marching directions during volume rendering. Specifically, a ray is emitted starting from a certain distance away (e.g., set to 0.1m in our experiments) along normal direction of the vertex, and traverses through the vertex following

its negative normal direction. Volume rendering is applied along the ray to obtain semantic label.

Qualitative results of semantic 3D reconstruction (voxel grid of resolution 256^3) for three Replica room scenes are shown in Figure 4.12. Note that Semantic-NeRF is able to predict decent geometry and semantics even in occluded regions (e.g., areas behind the sofa) and fill the holes to some extent in unobserved regions.

4.5 Conclusion

We have shown that adding a view-invariant semantic output to a scene-specific implicit MLP model of geometry and appearance means that complete and high resolution semantic labels can be generated for a scene when only partial, noisy or low-resolution semantic supervision is available, motivated by the redundancy in semantic labelling as well as the consistency and smoothness inherent in the proposed representation. This method has practical uses in robotics or other applications where scene understanding is required in new scenes where only limited labelling is possible.

Enabling real time rendering from NeRF-like neural implicit representations is an important and active research area as well. There have been many recent attempts [Sitzmann et al., 2021, Garbin et al., 2021, Yu et al., 2021a, Reiser et al., 2021] to accelerate volume rendering of NeRF. In addition to inference, the burden of expensive training can be mitigated by cloud computing or improved generalisation capability given extra priors [Yu et al., 2021b, Trevithick and Yang, 2021]. Online learning of NeRF has already been shown to be possible given RGB-D observations, where depth input helps fast convergence of implicit dense geometry, enabling SLAM applications [Sucar et al., 2021]. Efforts in these areas will result in reductions in time and memory complexity and step towards deployment on mobile computing platforms as an exciting future direction. However, the scalability

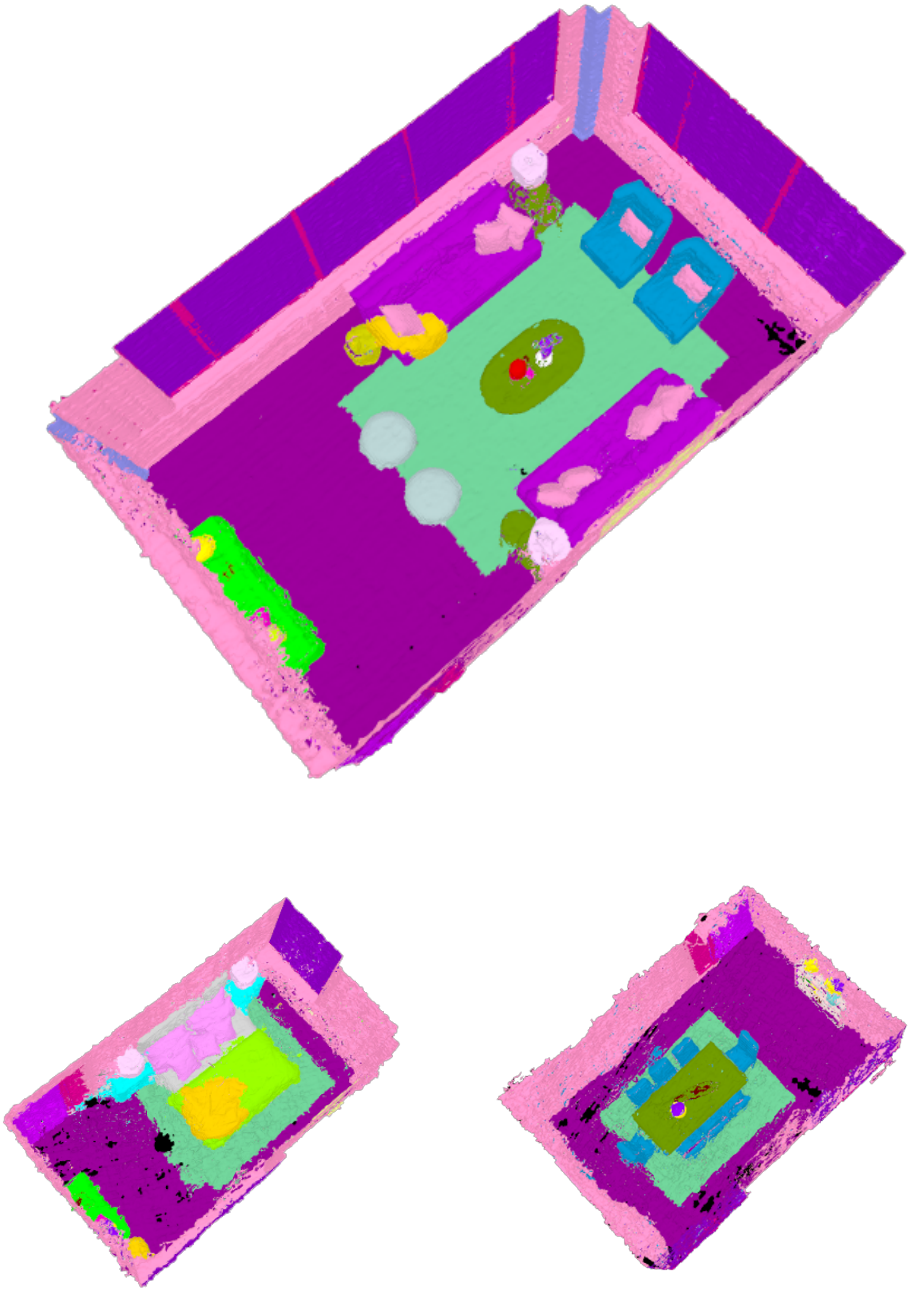


Figure 4.12: Semantic 3D reconstruction obtained using Semantic-NeRF. Note that our learned scene-specific 3D representation predicts decent geometry and semantics in occluded regions and fills the holes caused by unobserved regions to some extent.

of NeRF to faithfully learn accurate geometry of larger and more cluttered scenes is still under exploration. Instead of increasing the sampling budgets and computational resources, more fundamental modifications of pipeline formulations are expected [Barron et al., 2021, Oechsle et al., 2021].

An interesting direction for future research of Semantic-NeRF is interactive labelling, where the continually training network asks for the new labels which will most resolve semantic ambiguity for the whole scene, leading to our work in Chapter 5.

CHAPTER 5

iLabel

Contents

Work within this chapter describes the system iLabel and was conducted under close collaboration with Edgar Sucar, leading to the paper: Zhi, S.*, Sucar, E.*, Mouton, A., Haughton, I., Laidlow, T., Davison, A. (2021). iLabel: Interactive Implicit Scene Labelling and Learning in Real-Time. *Under submission*. [Zhi et al., 2021b]
(* indicates equal contribution to the paper.)

5.1 Introduction

An intelligent autonomous agent must build an internal representation of its environment which goes beyond geometry and colour to include a semantic understanding of the scene. Research on neural implicit representation has shown that a coordinate-based MLP network can be trained from scratch in a single scene via automatic self-supervision to accurately and flexibly represent geometry and appearance [Park et al., 2019, Mescheder et al., 2019, Mildenhall et al., 2020]. Semantic-NeRF [Zhi et al., 2021a] from Chapter 4 demonstrated that the compression, intrinsic smoothness and multi-view consistency of these representations are inherited by additional output channels which can be used to predict dense semantic properties over the scene, allowing sparse supervised labels to propagate efficiently. The iMAP system [Sucar et al., 2021] showed for the first time that a neural implicit MLP can be trained in real-time, without any prior data, while capturing a scene with a handheld RGB-D camera.

In this chapter, we build on iMAP and Semantic-NeRF to introduce iLabel, the first *online*, *interactive* 3D scene understanding system based on neural implicit scene representations. It allows users to annotate semantic properties in a scene via clicks, while simultaneously scanning and mapping it with a handheld RGB-D sensor. Correspondingly, high-quality dense semantic scene reconstruction can be obtained from scratch with only a few minutes of scanning and a few tens of semantic click annotations. The scene model is updated and visualised in real-

time, allowing the user to focus interactions as needed to achieve ultra-efficient labelling. iLabel's underlying model is an MLP trained from scratch in real-time to learn a joint implicit encoding of geometry, appearance and semantics in 3D. The internal smoothness and consistency of the representation of shape and appearance is inherited by the semantic channel, allowing it to make accurate dense predictions from very sparse annotations, and regularly auto-segment objects and other regions. Our approach requires no prior training on semantic datasets and can therefore be applied in novel contexts, with categories decided on-the-fly by the user in an open-set manner. Additionally, semantic categories can be defined either as a flat set of classes or hierarchical tree, where the user can choose the specificity of their labels at run time. For example, a user could start by labelling some objects as 'furniture' and later break down that class into 'chair', 'table', 'bed', etc.

Typically, a deep neural network trained on datasets of thousands of images with dense, high-quality human annotations is used to semantically label a new scene. Not only is there a high cost to creating these training datasets, but the quality of labelling produced by these networks can be poor due to the distance between the distributions of the training data and the test scene. Instead, our approach allows for high-quality labelling with minimal human in-situ interaction because the user monitors the semantic map as it updates in real-time and clicks only as needed to correct it. The smoothness properties of the MLP mean that regions and objects are coherently represented, and can frequently be labelled with only one click. Sometimes, not even a single click is required as the correct properties will be transferred from already labelled parts of the scene. We have shown that a room or similar scene can be highly accurately labelled into 10+ semantic categories with only a few tens of clicks, where these categories are either known in advance or defined in an interactive 'open-set' manner by the user.

As the representation also includes a notion of semantic uncertainty, the user can alternatively take on a more passive role in the labelling process. iLabel can

examine the entropy of the semantic classes at sampled surface points and ask the user to provide labels where uncertainty is greatest, which is named automatic query generation in this work. Such interaction may ease the burden of manual annotation in practical applications, such as a future intelligent household robot scanning a new environment for the first time.

In numerous qualitative examples, we demonstrate the power and flexibility of iLabel to provide high-quality, dense semantic labelling in a wide variety of scenes with minimal human interaction. We also show the benefits of a hierarchical semantic class representation qualitatively, and perform ablation studies to investigate the properties of the underlying MLP. Quantitative experiments on both synthetic and real-world datasets show that the labelling accuracy of iLabel scales powerfully with the number of clicks, and that with just a few tens of clicks, iLabel can outperform a state-of-the-art RGB-D semantic segmentation method trained on datasets with thousands of dense annotations.

Overall, iLabel has the flexibility to be used in a variety of scenarios: from an interactive, user-friendly data annotation or scene labelling tool to a core perception module enabling intelligent robots to operate in open-set environments. The main contributions of this work can be summarised as follows:

- Propose the first real-time, online, interactive scene understanding system, of which an MLP is the sole representation.

- Introduce a novel 3D-aware hierarchical segmentation encoded by binary tree structures.

- Automatic query generation leveraging on uncertainty sampling strategies for localisation-free labelling.

5.2 Related Work

Online Scene Understanding and Labelling Existing real-time, dense semantic mapping systems typically contain two parallel modules: 1) an RGB-D based geometric SLAM system, maintaining a dense 3D map of the scene, and 2) a semantic segmentation module that predicts dense semantic labels of the scene [Hermans et al., 2014, Stückler and Behnke, 2014, McCormac et al., 2018, Nakajima et al., 2019]. Multi-view semantic predictions are incrementally fused into the geometric model, yielding densely-labelled, coherent 3D scenes. While semantic segmentation has been performed using a variety of techniques [Krähenbühl and Koltun, 2011, Nguyen et al., 2017, Krähenbühl and Koltun, 2011, Long et al., 2015, Chen et al., 2018b], it is an inherently user-dependent and subjective problem [Martin et al., 2001]. User-in-the-loop systems are therefore crucial in enabling full flexibility when defining semantic relations between entities in a scene. In this context, the works most closely related to ours are SemanticPaint [Valentin et al., 2015] and Semantic Paintbrush [Miksik et al., 2015].

SemanticPaint [Valentin et al., 2015] is an online, user-in-the-loop system that allows the user to label a scene during capture. To this end, the user interacts with a 3D volumetric map, built from an RGB-D SLAM system, via voice and hand gestures [Nießner et al., 2013]. A streaming random forest classifier, using hand-crafted features called Voxel-Oriented Patches (VOPs), learns continuously from the user gestures in 3D space. The forest predictions are used as unary terms in a conditional random field (CRF) to propagate the user annotations to unseen regions. As the CRFs are built upon the reconstructed data, there is an underlying assumption that this data is good enough to support label propagation. In practice SemanticPaint requires a full initial scan of scenes before any labelling can be done, which additionally has four distinct modes: 1) Labelling; 2) Propagation; 3) Training and 4) Testing, the user must manually switch between them which makes

the whole process cumbersome and slow. SemanticPaint is therefore restricted to comparably simple scenes and its efficacy in complex real-word scenarios is limited. Semantic Paintbrush [Miksik et al., 2015] extends this framework to operate in outdoor scenes. Using a purely passive stereo setup for extended range and outdoor depth estimation, users visualise the reconstruction through a pair of optical see-through glasses and can draw directly onto it using a laser pointer to annotate the objects in the scene. The system learns in an online manner from the these annotations and is thus able to segment other regions in the 3D map.

In contrast to [Valentin et al., 2015, Miksik et al., 2015], iLabel does not rely on hand-crafted features and complex engineering pipelines, benefiting instead from a powerful joint internal representation of shape and appearance; and provides a unified interface for online reconstruction, segmentation and labelling which leads to a much simpler and intuitive system overall.

Hierarchical Segmentation Finding the underlying hierarchical structure of complex scenes is a long-standing problem in computer vision for scene understanding. Hierarchical segmentation focused on representing the scene using a tree-structure, i.e., a set of segmentations with different detail levels where the segmentations at finer levels (child nodes) are nested with respect to those at coarser levels (parent nodes). Because the real-world scenes are hierarchical in nature, hierarchical semantic representation provides a holistic, compact and comprehensive reasoning over components of scene graphs than a standard flat segmentation in which each segment is assumed to be independent to others.

Early attempts [Arbelaez et al., 2010, Arbeláez et al., 2014] used low-level image statistics to extract an ultrametric contour map (UCM), leading to further work on using CNNs for hierarchical segmentation in a supervised manner [Xie and Tu, 2015, Maninis et al., 2016, Mo et al., 2019, Hiroaki Aizawa, 2021]. It is still less evident in the community how to well tackle hierarchical segmentation using

deep neural networks, especially given that various datasets include a significant variation in the scene, such as object/stuff categories, arrangement and their layouts. We show that iLabel alleviate these problems and can build a user-defined hierarchical scene segmentation interactively and store it within the weights of a single MLP.

5.3 iLabel: Online, Interactive Open-Set Labelling and Learning

5.3.1 iLabel System Overview

iLabel is built on top of the iMAP SLAM system [Sucar et al., 2021]. iMAP represents 3D scenes using an implicit neural representation, parameterised by a multi layer perceptron (MLP) that maps a 3D coordinate to a colour and volume density. It jointly optimises the MLP and the camera poses of selected keyframes through differential volume rendering with actively sampled sparse pixels for higher efficiency, while tracking the position of a moving RGB-D camera against the neural implicit representation. These design choices lead to a real-time and long-term RGB-D SLAM enabling complete and accurate construction.

The overview pipeline of the iLabel system is shown in Figure 5.1. The system works in three concurrent processes: tracking, mapping and labelling. Following the design spirit of PTAM (Parallel Tracking and Mapping) [Klein and Murray, 2007], the tracking operates at around 10Hz and the mapping works at 2Hz on our desktop or laptop configuration. The GUI visualisation and labelling process is capable of rendering full frames at $\frac{1}{6}$ of the original image resolution at 8FPS.

iLabel augment iMAP with an extra semantic head to the MLP that predicts either a flat class distribution or a binary hierarchical tree (see Section 5.3.3). In parallel to iMAP, a user provides annotations via clicks in the keyframes. Scene

semantics are then optimised through semantic rendering of these user-selected pixels. The smoothness and compactness priors present in the MLP enable the user-supplied labels to be automatically and efficiently propagated throughout the scene. iLabel is thus able to produce accurate, dense predictions from very sparse annotations and to regularly auto-segment objects and other regions not labelled by the user. The ability to simultaneously reconstruct and label a scene in real-time allows for ultra-efficient labelling of new regions and for easy correction of errors in the current semantic predictions.

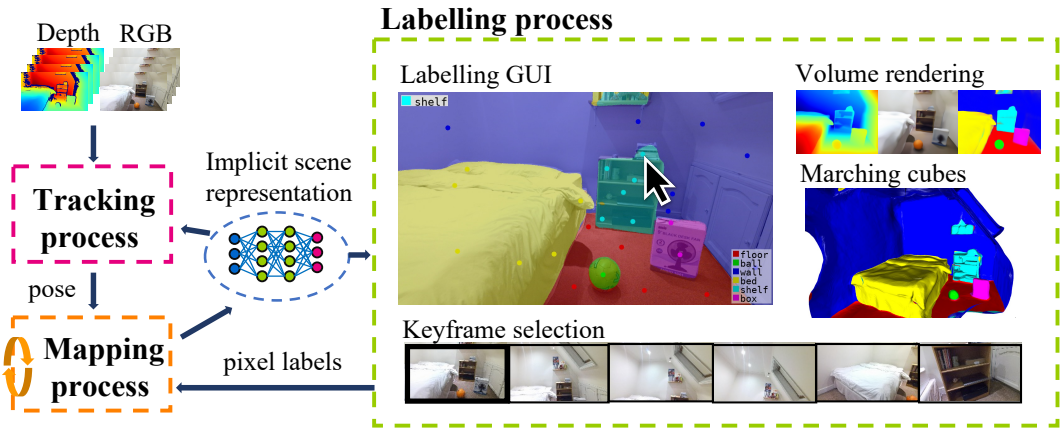


Figure 5.1: Overview of the iLabel system pipeline, including three processes: tracking, mapping and labelling work in parallel. Tracking process enables frame-rate camera tracking of input RGB-D frames; labelling process supports user interaction with GUI and visualisation; mapping process is responsible for learning scene representation using joint optimisation of appearance, geometry and semantics from the labelling process.

5.3.2 Network Architecture

The MLP adopted in iLabel is shown in Figure 5.2. Similar to iMAP [Sucar et al., 2021], optimisable Gaussian Fourier feature mapping of dimension 93 with sinusoidal activation is applied to input 3D position $\mathbf{x} = (x, y, z)$ [Tancik et al., 2020, Sitzmann et al., 2020], which is implemented as an extra MLP layer. The sizes of four hidden fully-connected (FC) layers are all set to 256, shared by three predictions heads of volume density, RGB colour and semantics. Different from

NeRF [Mildenhall et al., 2020], viewing direction is not taken for colour prediction because photorealistic rendering is not the target, which also eases the learning task. The size of semantic head is set to a relatively large value (i.e., 20) which is enough to cover maximum number of potential incoming classes within a real world scene.

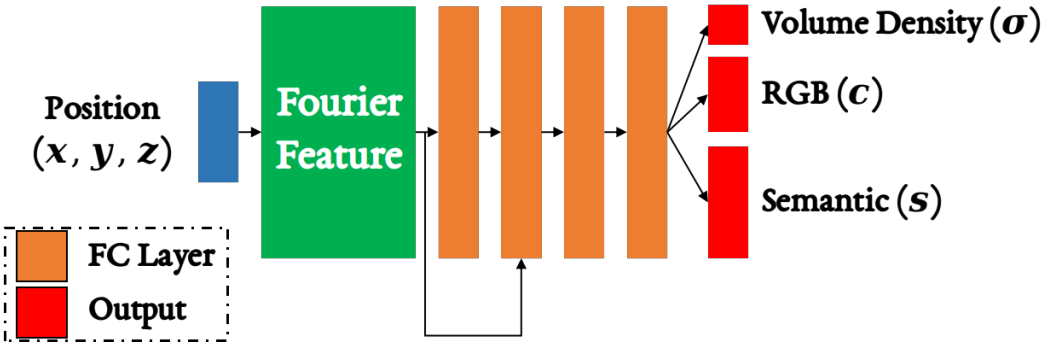


Figure 5.2: MLP architecture used in iLabel. The network predicts three different properties of given position (x, y, z) including volume density, RGB and semantics, of which all are formulated as view-invariant mapping.

5.3.3 Semantics Representation and Optimisation

At the heart of iLabel is the continuous optimisation of the underlying implicit scene representation described in Figure 5.2:

$$F_\theta(\mathbf{x}) = (\mathbf{c}, \mathbf{s}, \sigma), \tag{5.1}$$

where F_θ is an MLP parameterised by θ; $\mathbf{c}$, $\mathbf{s}$ and σ are the radiance, semantic logits and volume density at the 3D position $\mathbf{x} = (x, y, z)$, respectively. The scene representation is optimised with respect to volumetric renderings of depth, colour and semantics, computed by compositing the queried network values along the back-projected ray of pixel $[u, v]$:

$$\hat{D}[u,v] = \sum_{i=1}^{N} w_i d_i, \quad \hat{I}[u,v] = \sum_{i=1}^{N} w_i \mathbf{c}_i, \quad \hat{S}[u,v] = \sum_{i=1}^{N} w_i \mathbf{s}_i, \tag{5.2}$$

where $w_i = o_i \prod_{j=1}^{i-1}(1-o_j)$ is the ray-termination probability of sample i at depth d_i along the ray; $o_i = 1-\exp(-\sigma_i \delta_i)$ is the occupancy activation function; $\delta_i = d_{i+1} - d_i$ is inter-sample distance.

As in [Sucar et al., 2021], geometry and keyframe camera poses $\{T_{WC}\}$ are optimised by minimising the discrepancy between the captured and rendered RGB-D images from sparsely sampled pixels. Semantics are optimised with respect to only the user-labelled pixels, with two different activations and losses, corresponding to the two semantic modes described below. Figure 5.3 gives an overview of the semantic rendering process and the activation functions applied to the rendered logits.

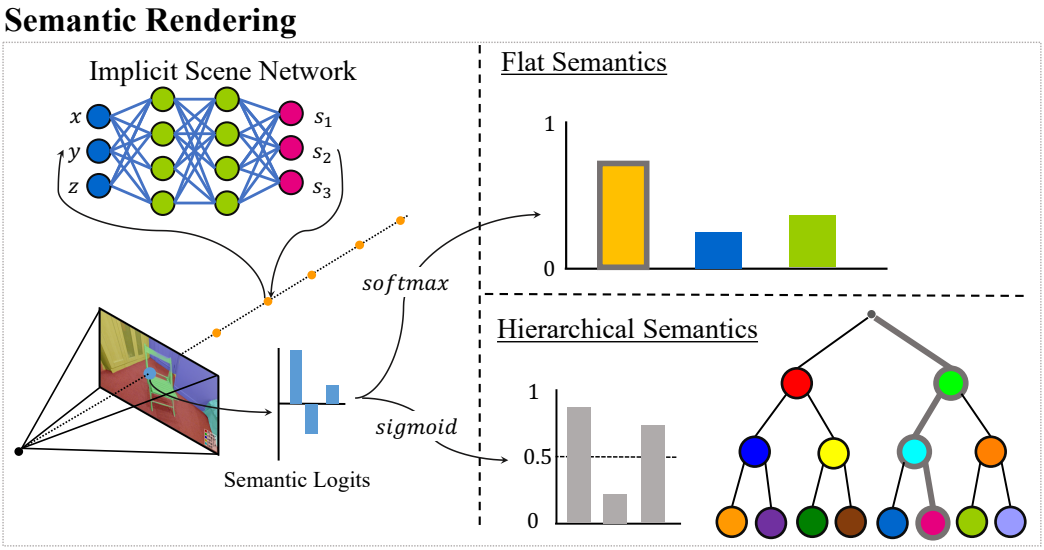


Figure 5.3: Illustration of semantic rendering in iLabel. Implicit scene network is queried for rendering semantic logits; *softmax* or *sigmoid* activations are applied for either flat or hierarchical segmentation modes, respectively.

Flat Semantics As per [Zhi et al., 2021a], the semantic outputs of network, $\mathbf{s}_i$, are multi-class semantic logits which are converted into the image space by differential volume rendering (Equation 5.2) followed by a *softmax* activation: $\hat{\mathbf{S}}[u,v] = softmax(\hat{S}[u,v])$. Semantics are then optimised using the image cross-entropy loss between the provided class ID and the rendered predictions.

Hierarchical Semantics We propose a novel hierarchical semantic representation through a binary tree, which allows for labelling and predicting semantics at different hierarchical levels. While the network output, $\mathbf{s}_i$, is still represented by an n-dimensional flat vector, n now corresponds to the depth of the binary tree as opposed to the number of semantic classes. The semantic logits are rendered in the same manner, but the image activation and loss functions differ.

A *sigmoid* activation function is applied to the rendered logits, producing values in the range $[0,1]$. The j^{th} rendered output value, $\hat{\mathbf{S}}_j[u,v] = \mathit{sigmoid}(\hat{S}_j[u,v])$, corresponds to the branching factor at tree level j. To obtain a hierarchical semantic prediction, each value $\hat{\mathbf{S}}_j[u,v]$ is set to 0 or 1 by thresholding $\hat{\mathbf{S}}_j[u,v]$ at 0.5. In the hierarchical setting, the user-supplied label corresponds to selecting a specific node in the binary tree. This label is transformed into a binary branching representation, and a binary cross entropy loss is computed for each rendered value. A label selecting a tree node at level L only conditions the loss on the output values up to and including level L: $\hat{\mathbf{S}}_j[u,v], j \in \{1,...,L\}$.

With reference to the top half of Figure 5.13, the network outputs three values corresponding to the three levels in the tree. First, the user separates the scene into *foreground* and *background* classes. A background label corresponds to the vector $[0,*,*]$ where $*$ indicates that no loss is calculated for the second and third rendered values. The user then divides the background class further into *wall* and *floor*, where the *wall* label corresponds to vector $[0,1,*]$. The binary hierarchical representation allows the user to separate objects in stages. For example the user first separates a whole bookshelf from the rest of the scene, and later separates the books from the shelf without contradicting the initial labels, meaning that no labelling effort is wasted.

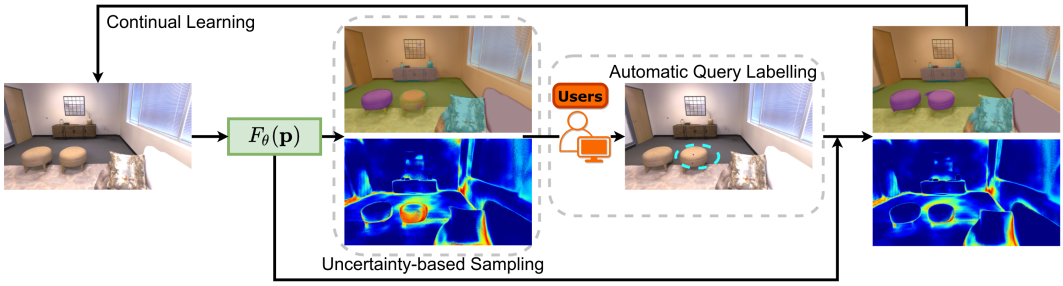


Figure 5.4: Overview of automatic query generation process. Uncertainty-based sampling is used to decide which keyframe and pixel are selected to query label from the oracle. The right chair gets incorrect label propagation result and owns relative higher uncertainty. Generated queries on this chair will correct the segmentation result and reduce the overall uncertainty on chair regions.

5.3.4 Semantic User Interaction Modes

Our system allows for two modes of interaction: 1) **manual interaction**, the usual interactive mode of iLabel, where users provide semantic labels in image space via clicks, and 2) **automatic query generation**, where the system generates automatic queries for the labels of informative pixels, driven by semantic prediction uncertainty (Figure 5.4). The latter mode eases the burden of localisation task in manual annotation, and users could provide labels via text or voice. Several query generation strategies are explored such as random sampling, softmax entropy, least confidence and margin sampling [Settles, 2009] and these uncertainty-based sampling strategies can integrate seamlessly with deep neural networks with little computational overhead [Settles, 2009, Ren et al., 2020].

Here we briefly introduce the applied strategies of uncertainty-based sampling and how they are applied to iLabel:

Random Sampling: Randomly sample a pixel position from all candidates as the query pixel position.

Softmax Entropy: Information entropy computed from softmax probabilities is used to measure semantic uncertainty. Coordinate of pixel owning highest

entropy is selected as the query position.

Least Confidence: Coordinate of the pixel whose maximum softmax probability value over classes is the minimum among all candidates is selected as the query pixel position.

Margin Sampling: Coordinate of the pixel whose most and second most probabilities have the minimum discriminativeness is selected as the query pixel position.

Under automatic query generation mode, iLabel relies on the adopted sampling strategy to decide which keyframe to select and then which pixel to query on the selected keyframe. Specifically, dense uncertainty maps for each keyframe are periodically updated and summed to have a frame-wise uncertainty, indicating the overall uncertainty of the keyframe. As a result, keyframe with higher total uncertainty is more likely to be selected, on which the query pixel position is then determined as well.

5.3.5 Implementation Details

iLabel operates in a multiprocessing, single or multi-GPU framework, running three concurrent processes: 1) tracking, 2) mapping, and 3) labelling (see Figure 5.1). The mapping process encompasses optimising the MLP parameters with respect to a growing set of W keyframes and associated RGB-D observations: $\{(I_i, D_i, T_i)\}_{i=1}^{W}$. As per [Sucar et al., 2021], the photometric loss L_p and geometric loss L_g are minimised on sparse, information-guided pixels. iLabel performs an additional optimisation on K user-selected pixels (ξ_i) in each keyframe and introduces a semantic loss L_s, minimising the following objective function:

$$\underset{\theta}{\operatorname{argmin}} \frac{1}{K} \sum_{i=1}^{W} \sum_{(u,v)\in\xi_i} \underbrace{e_i^g[u,v]}_{L_g} + \alpha_p \underbrace{e_i^p[u,v]}_{L_p} + \alpha_s \underbrace{e_i^s[u,v]}_{L_s}, \tag{5.3}$$

where:

$$e_i^p[u,v] = \left| I_i[u,v] - \hat{I}_i[u,v] \right|, \qquad e_i^s[u,v] = -\sum_{c=1}^{C} \mathbf{S}_i^c[u,v] \log(\hat{\mathbf{S}}_i^c[u,v])$$

$$e_i^g[u,v] = \frac{\left| D_i[u,v] - \hat{D}_i[u,v] \right|}{\sqrt{\hat{D}_{var}[u,v]}}, \quad \hat{D}_{var}[u,v] = \sum_{i=1}^{N} w_i (\hat{D}[u,v] - d_i)^2,$$

and in the hierarchical setting:

$$e_i^s[u,v] = \sum_{l=1}^{L} -\mathbf{S}_i^c[u,v] \log(\hat{\mathbf{S}}_i^c[u,v]) - (1 - \mathbf{S}_i^c[u,v]) \log(1 - \hat{\mathbf{S}}_i^c[u,v]), \quad (5.4)$$

α_c and α_s are set to 5 and 8, the Adam optimiser is used with poses and map learning rates of 0.003 and 0.001.

iLabel does not have explicit/specific refinement process, and all user clicks are involved in the joint optimisation in Equation 5.3. The optimisation keeps working and growing with changing sparse samples for colour and geometry reconstruction, and increasing annotated pixels for semantics, colour and depth as well.

The labelling process coordinates user interactions (clicks and labels) and controls the rendering of semantic images and meshes (via marching cubes on a dense voxel grid queried from the MLP). In practice the user can choose to perform annotations while scanning, or to perform scanning first followed by labelling or to perform both, and we leave this choice to the user depending on the application scenario. From a probabilistic point of view, joint optimisation in Equation 5.3 allows the consideration of the joint distribution of structure, motion and semantics, including their cross correlations. In iLabel, since the semantic labelling is extremely sparse, we expect the corrections of structure estimates to propagate to semantic estimates. In addition, the joint optimisation in Equation 5.3 leads to a simple formulation for the online learning process which makes iLabel more straightforward for users to operate than similar interactive methods, which we will address more in Section 5.4.1.

5.4 Experiments and Applications

iLabel is an interactive tool intended for real-time use and we therefore emphasise that its strengths are best illustrated *qualitatively*. To this end, we provide extensive examples to demonstrate iLabel in a variety of interesting scenes. Additionally, we perform a quantitative comparison to a state-of-the-art, fully-supervised RGB-D segmentation baseline [Chen et al., 2020], in real and synthetic scenes from Replica and ScanNet datasets [Straub et al., 2019, Dai et al., 2017a], representative of the intended operating environment of iLabel.

Live images captured from MS Kinect Azure RGB-D sensor of 720p resolution (1280x720) are used in all the qualitative experiments. Images in quantitative experiments from Replica and ScanNet datasets have resolutions of 1200x680 and 640x480, respectively. These images serve as input to the iLabel system. For visualising the rendering in the online system we render the images at 1/6 resolution for efficiency on a different process running at 8fps. For quantitative evaluations, we render the images in full resolution and compute corresponding metrics.

5.4.1 Qualitative evaluation

As the geometry, colour and semantic heads share a single MLP backbone, user annotations are naturally propagated to untouched regions of the scene without specifying an explicit propagation mechanism (e.g. pairwise terms of a CRF used in [Valentin et al., 2015]). This, together with a user-in-the-loop, enables ultra-efficient scene labelling with only a small number of well-placed clicks.

A core strength of iLabel is the highly compressed scene representation learnt by the lightweight MLP. We have observed that the resulting embeddings are highly correlated for coherent 3D entities in the scene (e.g. objects, surfaces, etc.). Consequently, iLabel is able to segment these entities very efficiently, even with a single click. This is illustrated in Figures 5.5 and 5.6, where only a few clicks generate

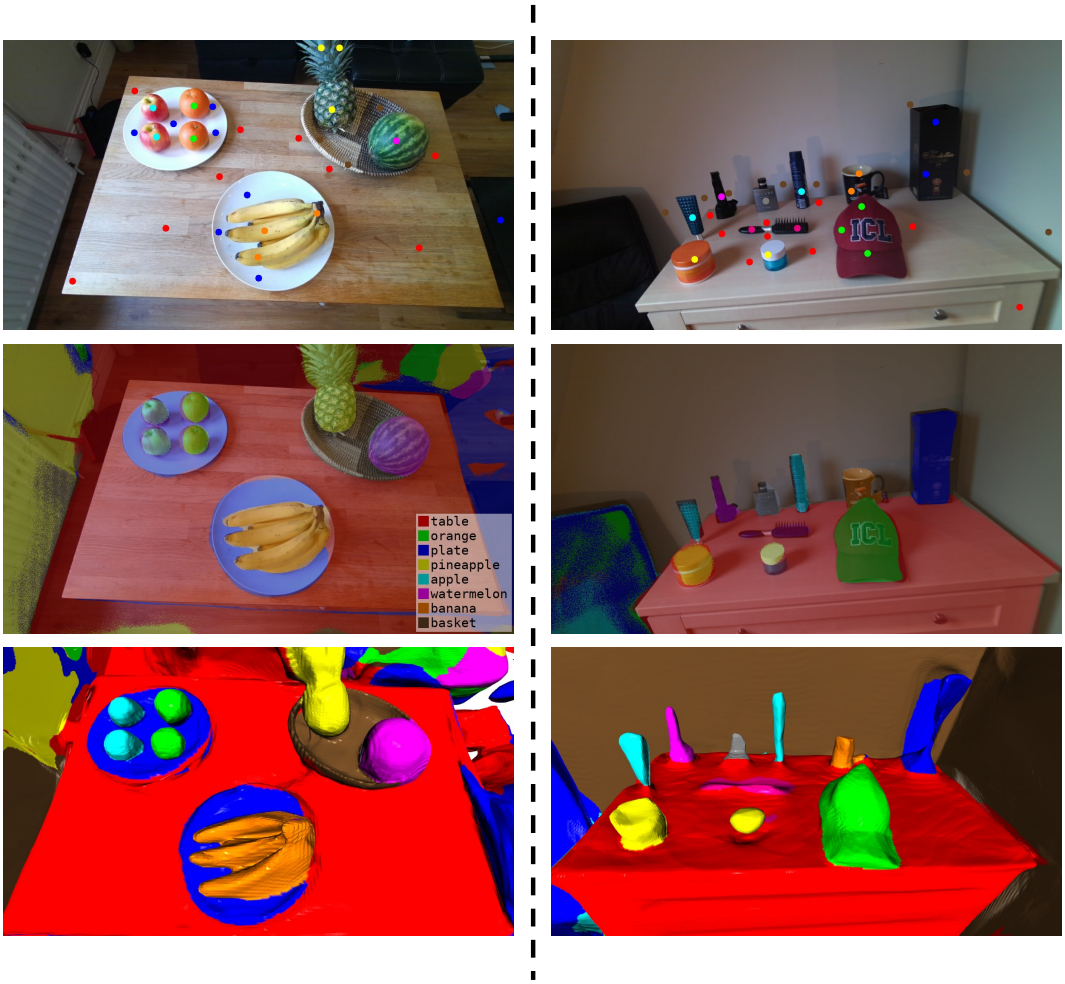


Figure 5.5: Precise segmentations can be obtained from just 1 or 2 clicks per object. From top to bottom are keyframes with user clicks, rendered 2D segmentation and semantically annotated 3D mesh, respectively.

complete and precise segmentations for a wide range of objects and entities, ranging from small, coherent objects (e.g. fruit) to deformable and intricate entities (clothing and furniture). The ability of iLabel to propagate labels across coherent shapes is highlighted further in Figure 5.7, where we show a comparably high-quality segmentation of the fruit scene with colour optimisation disabled (i.e. the system relies entirely on depth). The fact that iLabel is able to produce high-quality segmentations with no colour information demonstrates the power of the underlying representation learnt by the MLP and shows that iLabel is not solely dependent

Figure 5.6: Ultra-efficient label propagation: iLabel produces high-quality segmentations of coherent 3D entities with very few user clicks, approximately 20–30 per scene.

Figure 5.7: In removing the use of colour optimisation for scene reconstruction, only a few extra clicks are required to achieve a comparable quality of segmentation to that shown in Figure 5.5.

on colour and can continue to operate when colour information is not available or is poor (e.g. under low lighting). However, the addition of colour absolutely improves labelling efficiency.

The coordinate-based representation avoids quantisation and allows the network to be queried at arbitrary resolutions. This property allows reconstruction of detailed geometry and skeletal shapes that, when semantically labelled, render very precise segmentations. Figure 5.8 illustrates high-fidelity segmentations of objects which would be challenging for a standard CNN.

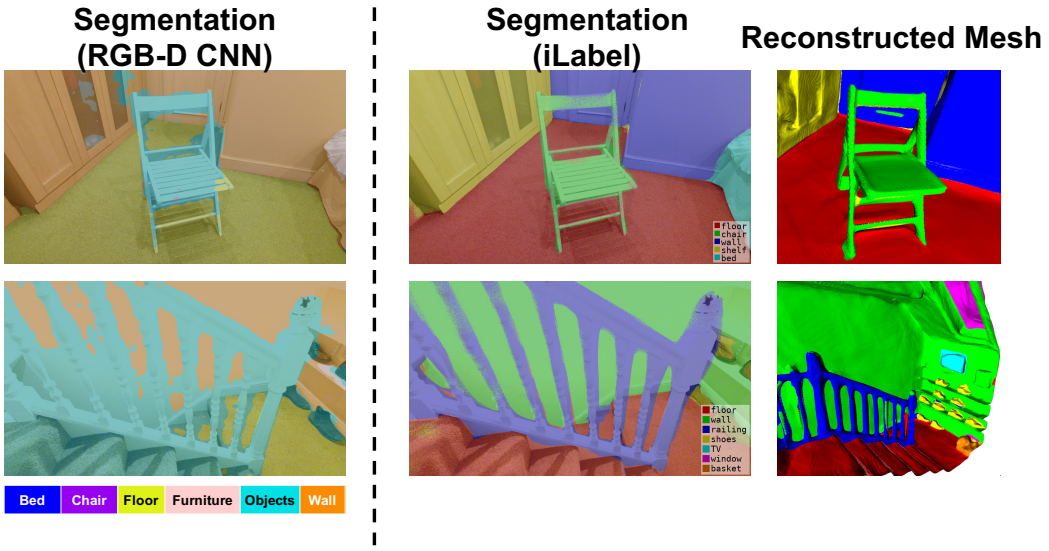


Figure 5.8: Segmentation results for challenging skeletal objects; left: pre-trained CNN SA-Gate on ScanNet (see Section 5.4.2), right: iLabel. CNNs struggle to predict correct and accurate segmentations on objects with fine structures, while iLabel is capable of recognising them leveraging on the learned implicit representation.

iLabel can be used as an efficient tool for generating labelled scene datasets. For example, a scene of a complete room with 13 classes, can be fully segmented with high precision with only 140 user clicks (Figure 5.9). Alternatively, iLabel can be used to tag individual objects for generating object-asset catalogues (Figure 5.10) to aid robotic manipulation tasks, for example.

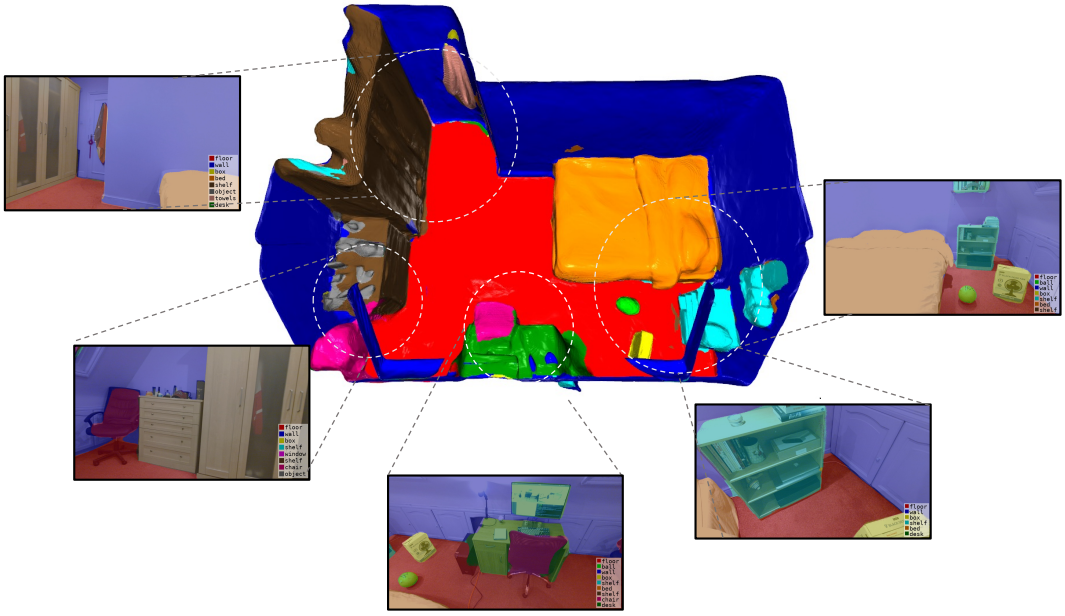


Figure 5.9: Whole-room semantic mesh and selected image semantic projections from only 140 clicks. We reconstruct and semantically label a whole room in under 5 mins.

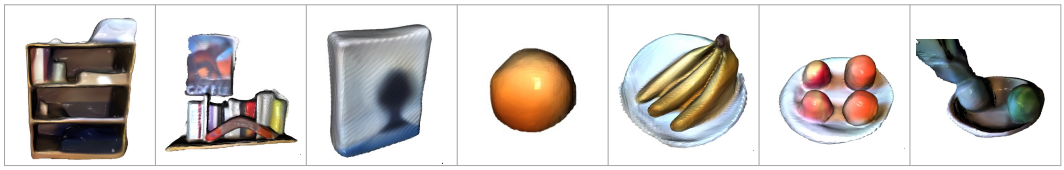


Figure 5.10: Catalogue of object mesh assets separated with iLabel.

While iLabel is particularly powerful at segmenting coherent entities, Figure 5.11 also demonstrates its ability to propagate user-supplied labels to disjoint objects exhibiting similar properties to the labelled objects. Each example shows label transfer between similar objects where only one has been labelled (e.g. (a) boxes on the bed, (b) food boxes and plastic cups and (c) toy dinosaurs). The table and chairs scene in Figure 5.11 (d) is especially interesting. Only four clicks are supplied: the label for the chair leg (blue) propagates to the leg of the table and the legs of the other chairs, while the table-top label (yellow) propagates to the seats of the chairs.

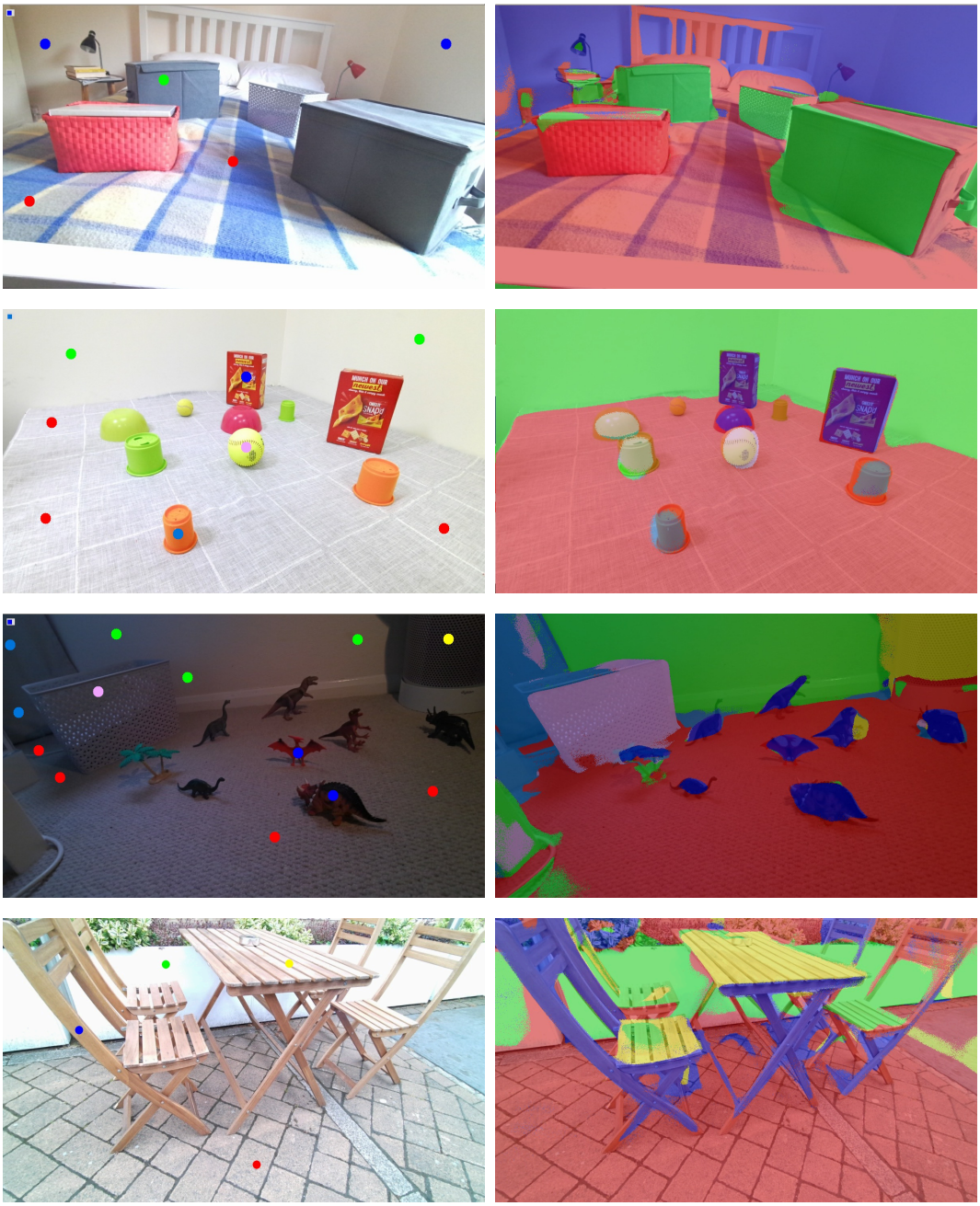


Figure 5.11: Generalisation: iLabel is able to transfer user labels to objects exhibiting similar properties. It is worth highlighting that the segmentation in the outdoor café scene (bottom row) was achieved with only 4 clicks.

User interactions We have demonstrated the labelling efficiency of iLabel when using the simplest user interactions (clicks). Here we note that a stroke (e.g. with a finger or stylus) is equivalent, in terms of user effort, to a single click and arguably more natural from a UI perspective. Figure 5.12 qualitatively compares the labelling efficiency of clicks versus strokes and demonstrates that strokes (represented as dense collections of clicks) yield superior segmentations for the same number of user interactions, as strokes can be regarded as a dense sequence of user clicks.

In this work we choose user annotations to be the challenging clicks to highlight the labelling efficiency of iLabel.

Hierarchical scene segmentation Figure 5.13 demonstrates iLabel's hierarchical mode. The colour-coded hierarchy (defined on-the-fly) is shown together with segmentations and scene reconstructions from each level. The results show the capacity of this representation to group objects at different levels, which has potential in applications where different tasks demand different groupings.

The advantages of the hierarchical semantic representation are qualitatively different from a flat one, allowing users/robots to simultaneously and compactly label a scene at different semantic levels without contradicting labels, for example an object can be both labelled as book and shelf at different levels in the hierarchy, is not fundamentally possible with a flat representation.

Comparison to SemanticPaint As an online interactive scene understanding system, SemanticPaint (SPaint) [Valentin et al., 2015] is the only other comparable system. SemanticPaint requires a full initial scan of scenes before any labelling can be done, which additionally has four distinct modes: 1) Labelling; 2) Propagation; 3) Training and 4) Testing, the user need to switch between them which makes the whole process relatively cumbersome and slow. In contrast, iLabel provides a

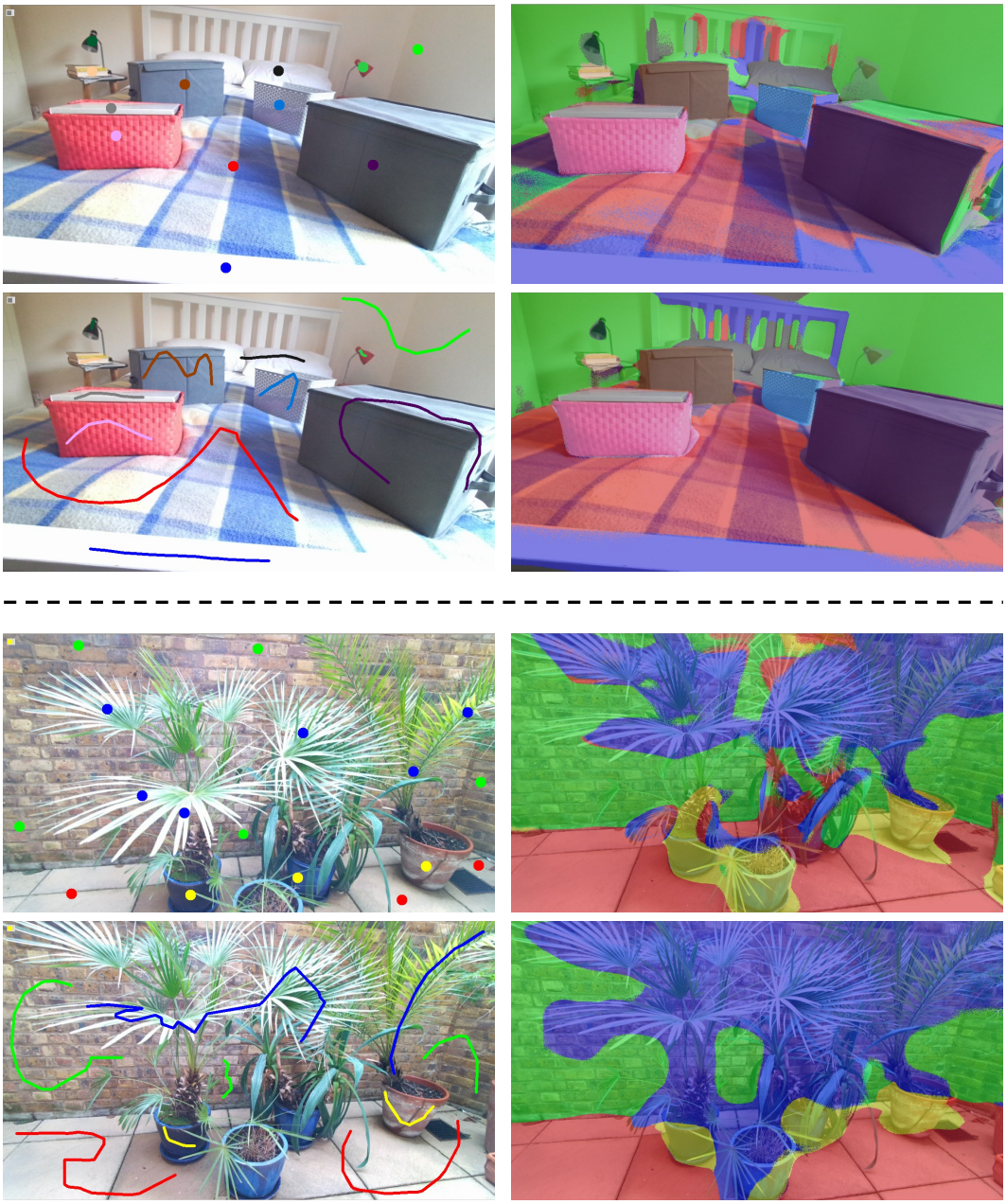


Figure 5.12: Clicks vs. strokes: Scenes can be labelled more efficiently and naturally using strokes.

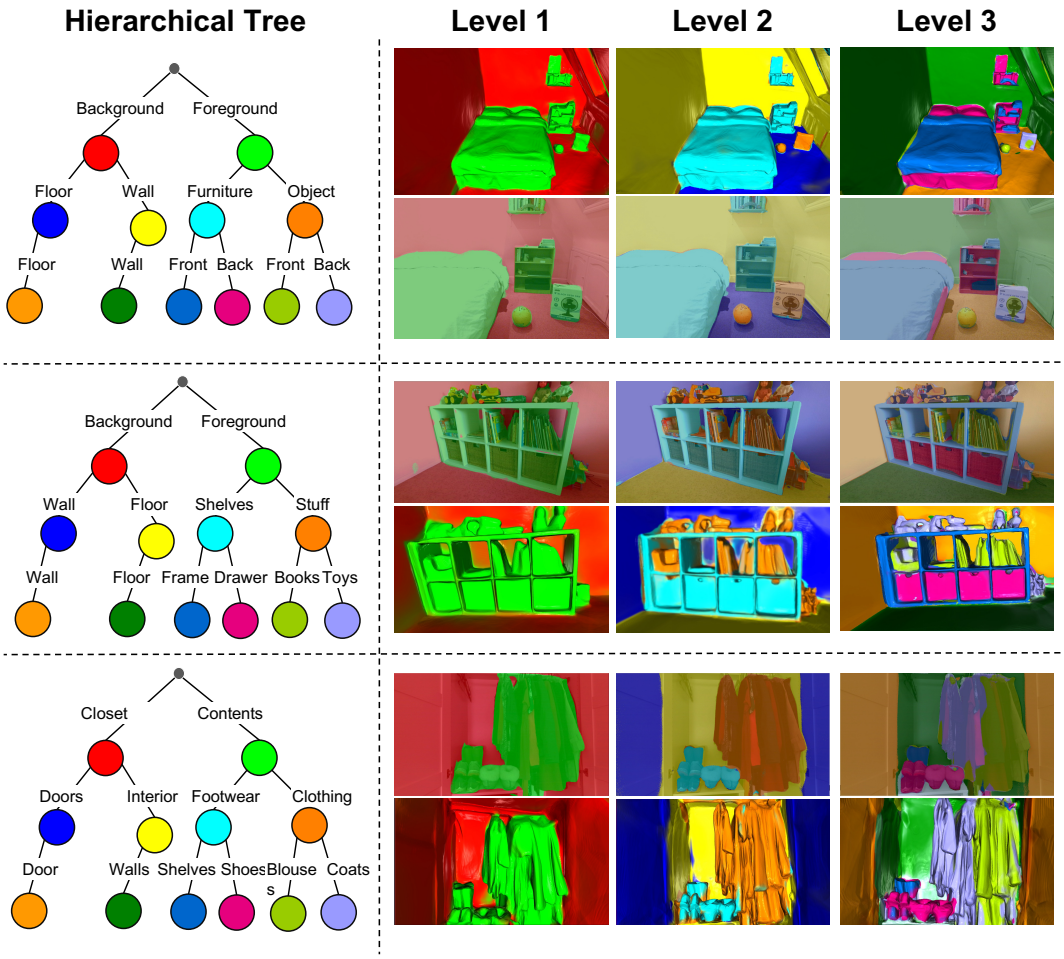


Figure 5.13: Binary tree as well as the segmentations at each level from the hierarchical mode of iLabel.

unified interface for online reconstruction, segmentation and labelling which leads to a much simpler and intuitive system overall.

A qualitative comparison is given in Figure 5.14. We can observe that given the same initial user annotations, the initial label propagation results of SemanticPaint are less complete and noisier compared to those of iLabel. With additional 5-10 subsequent corrective strokes and multiple switches among different working modes, SemanticPaint achieves qualitatively comparable segmentations to iLabel. In contrast, iLabel gives smooth and accurate segmentation with <10 user strokes with a single pass of sequences. Overall, iLabel demonstrated competitive seg-

mentation quality with a much simpler and easier-to-use system design.

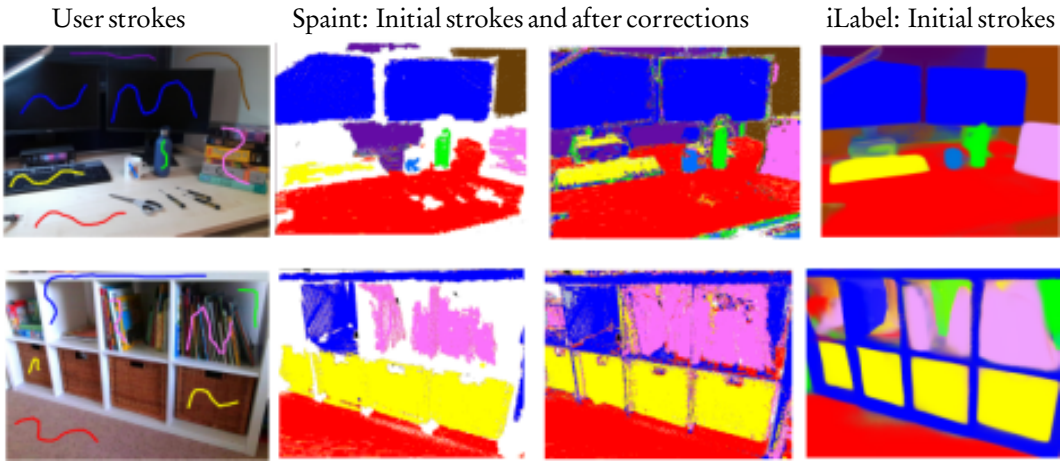


Figure 5.14: Qualitative comparison between SemanticPaint and proposed iLabel system. The first column shows the keyframe and user supplied annotation in form of strokes which are well supported by both systems. There are a total of 9 and 7 strokes for desk and shelves scenes, respectively. The middle two columns demonstrate, after a full scan of the scene, the label propagation result given initial strokes and the one after additional training, inference and user corrections of SemanticPaint, respectively. The right column shows the performance of iLabel given only initial strokes without further operations.

5.4.2 Quantitative evaluation

We evaluate iLabel's 2D semantic segmentation performance in both typical user-interaction mode and automatic query generation mode, with varying numbers of clicks per scene, on the public datasets Replica [Straub et al., 2019] and ScanNet [Dai et al., 2017a]. We report the mean intersection-over-union (mIoU), averaged over 13 classes.

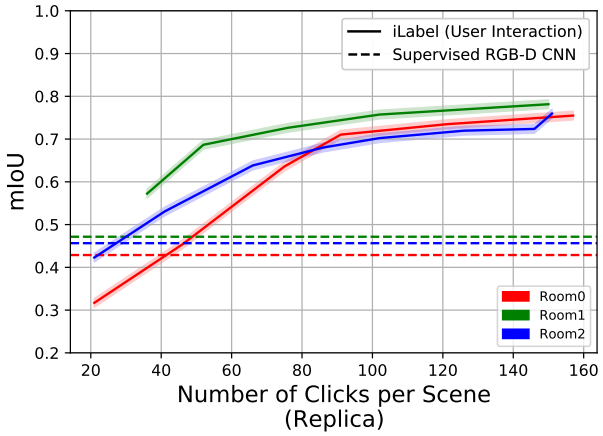
Datasets The Replica dataset [Straub et al., 2019] is a reconstruction-based synthetic dataset, containing 18 scenes with high-quality 3D meshes, photo-realistic textures and rich annotations. ScanNet [Dai et al., 2017a] is a large-scale real-world indoor dataset composed of approximately 2.5M views obtained from 1513 scenes. We test on the official ScanNet validation sets and generate RGB-D test se-

quences and ground-truth semantic labels from the Replica dataset using Habitat-Sim [Savva et al., 2019] with randomly-generated 6DoF trajectories. All semantic labels are remapped to the popular NYUv2-13 standard [Eigen and Fergus, 2015].

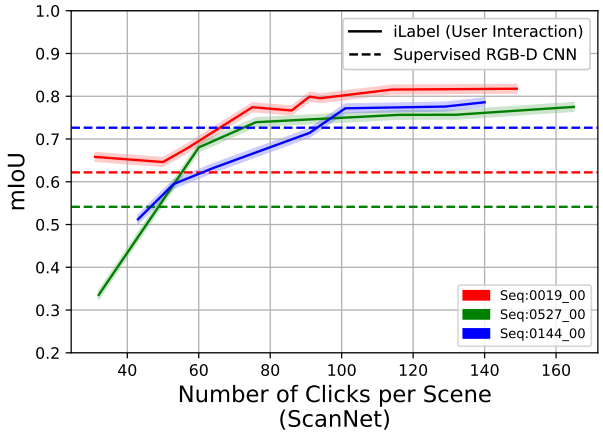
Evaluation is performed on uniformly-sampled frames from each test sequence, to ensure a faithful comparison against the supervised baseline. For each candidate test frame, we record its relative pose with respect to the nearest keyframe during scanning from the tracking process, as keyframe poses are continually optimised in the mapping process. During evaluation, the camera poses of the candidate test frames are then obtained using the recorded relative pose and the latest updated pose of its corresponding nearest keyframe. These retrieved poses are finally used to render the 2D segmentation masks for iLabel at specific viewpoints.

Baselines Performance is evaluated against SA-Gate with a ResNet-101 based DeepLabV3+ backbone [Chen et al., 2018b, Chen et al., 2020], which is the current state-of-the-art CNN model in RGB-D segmentation.

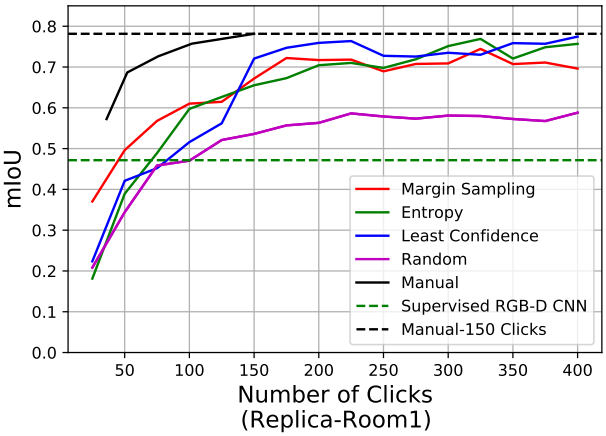
For Replica, we pre-train SA-Gate using the SUN-RGBD dataset [Song et al., 2015] and fine-tune on our generated test sequences. We adopt a leave-one-out strategy, whereby fine-tuning is performed independently for each test sequence using the remaining sequences in the test set. For ScanNet, we train SA-Gate directly on the official training sets, achieving 63.98% mIoU on the validation sets. Approximately 11k and 25k training images were used for training on Replica and ScanNet respectively. The ResNet-101 backbone is initialised with ImageNet pre-trained weights [Russakovsky et al., 2015] in all experiments. As per [Chen et al., 2020], depth maps are encoded using the HHA encoding [Gupta et al., 2014], while the fast depth completion technique [Ku et al., 2018] is used for hole-filling in the ScanNet dataset.



(a) Replica.



(b) ScanNet.



(c) Automatic query generation.

Figure 5.15: Quantitative evaluation of 2D semantic segmentation. Both interaction modes are evaluated and outperform supervised baselines with a small annotation budget.

Results Figure 5.15 show the performance of iLabel with user-supplied clicks compared against the supervised baseline (dashed horizontal line) for the Replica (left) and ScanNet (middle) datasets. With as few as 40 user clicks, iLabel matches the state-of-the-art. On the Replica scenes, iLabel significantly outperforms the baseline with >60 clicks and performance continues to scale powerfully as the number of user interactions is increased. Despite the huge difference between supervised baselines and iLabel, the quantitative results aim at demonstrating the labelling efficiency and quality of iLabel given only few clicks.

iLabel outperforms the baseline on ScanNet with few clicks as well, while the performance gap is not as pronounced. This can be attributed to the lower-quality depth images in ScanNet, inaccurate 2D ground-truth masks and void regions in the ground-truth masks arising from the projection process used to generate them [Dai et al., 2017a]. While the performance of iLabel is dependent on reliable pose tracking and hence high-quality depth data, tracking is not a core contribution of this work and we argue that the performance observed on the Replica dataset is thus more representative of the gains obtainable by iLabel, given suitable operating conditions. Figure 5.8 additionally shows a qualitative comparison of the segmentation masks on challenging skeletal objects generated by iLabel and SA-Gate trained on ScanNet.

Figure 5.15c confirms the effectiveness of automatic query generation, which opens the possibility for contactless scene labelling, e.g., by voice command. All the discussed uncertainty sampling strategies reach human performance except random sampling strategy, as rare classes are hardly sampled during evaluation, leading to degradation in mIoU metrics. Best performance of each strategy during different rounds are reported. As expected, automatic mode would be less efficient at labelling than human interaction, however, it is still very promising as similar performance can be achieved in only 400 clicks. One reason for this result is that users tend to focus on fixing incorrect segmentations directly while uncertainty-

based sampling requires more samples to allocate clicks on those regions, particularly small instances. One possible solution to reduce this gap between manual labelling and automatic labelling is the combination of both modes, where manual clicks serve as bootstrapping before switching to automatic query mode to have a better estimation of uncertainty. How to estimate good uncertainty with in-situ labelling is an exciting area for future work and worth further exploration.

5.5 Conclusion

We have presented iLabel and shown that online, scene-specific training of a compact MLP model which encodes scene geometry, appearance and semantics allows ultra-sparse interactive labelling to produce accurate dense semantic segmentation, far surpassing the performance of standard pre-trained approaches. As real-time scene understanding system, iLabel works only with in-situ supervision and therefore is able to work in various live and open-set environments with either user-supplied clicks or automatically predicted queries. We would also like to explore more applications of iLabel such as custom dataset creation for supervised methods, bootstrapping for weakly/semi-supervised learning techniques and robot learning.

Despite promising results, the label propagation mechanism of iLabel works well mainly for proximal regions and/or those sharing similar geometry or texture. A deeper understanding of this mechanism is necessary to enable better control of this process and to improve generalisation performance (e.g. class-based generalisation), which could further improve labelling efficiency and segmentation quality of the system. In addition, the formulation of hierarchical segmentation is currently limited to a binary tree, though the tree does not necessarily need to be balanced as the user can select which nodes to further divide. A more general hierarchical representation where each node can have an arbitrary number of

children is an interesting problem for future work.

Fortunately, as architectures and methods for neural implicit representation of scenes continue to improve, we expect these gains to be passed on to our labelling approach, and for tools like iLabel to become highly practical for applications where users are able to teach AI systems efficiently about useful scene properties.

CHAPTER 6

Conclusions and Future Work

All the methods developed in this thesis focus on neural semantic scene representations, using either external datasets or in-place labelling. While our qualitative and quantitative evaluation on extensive datasets and real-world scenes have shown the benefits of such semantic representations, we have also discussed their shortcomings. In this final chapter, we summarise the key novel contributions and results presented in this thesis and further discuss their current limitations and potential directions for improvement and future research.

Chapter 3 presented SceneCode, a compact and optimisable code representation for dense semantic labelling. The conditional distribution of semantic segmentation given colour images is learned within latent codes and the prior information is used for tackling multi-view semantic label fusion. Through minimising a multi-view semantic error term, our approach reaches coherent and smooth labelling efficiently by code optimisation and outperforms element-wise fusion baselines. The use of compact code representations of both geometry and semantics from a multitask CVAE allows for a concise monocular dense semantic reconstruction system.

While the system achieves promising results on various sequences from a similar

distribution to its training data, the generalisation capability of fully supervised representations to challenging unseen real world scenes remains barely satisfactory. Despite great progress in semantic segmentation techniques leveraging high-quality datasets with predefined semantic categories, how to enable better transferability of such semantic representation to new environments and even unknown classes in the open-set world remains as an open research problem [Pham et al., 2018b, Pham et al., 2018a, Nakajima et al., 2019]. Weakly-/semi-/self-supervision based deep learning approaches have gained more attention recently to alleviate the expenses of label collection and take better advantage of unlabelled natural images [Liu et al., 2021, Zou et al., 2021, Hung et al., 2019, Araslanov and Roth, 2020].

Another exciting venue of future search is enabling joint optimisation of both comes from the fact that joint optimisation of dense geometry and semantics from scratch still does not guarantee better estimation of both modalities, therefore a stage-wise optimisation is taken in SceneCode. Though we believe better network architecture design and stronger backbone would potentially alleviate this issue by much improved monocular predictions and proper initial code value [Czarnowski et al., 2020], a key missing factor lies in semantic error term which is not intrinsic to describe semantic labelling and extra regularisation is required to prevent trivial solutions. Considering the subjective nature of semantic concept, it is worth investigating if it is feasible to find such an intrinsic description of semantic labelling, either by design or by learning from data, so that joint inference of both geometry and semantics can be realised. Perhaps a more interesting way to enable joint inference is to explore a joint code representation so that cross-correlation is naturally taken into consideration during optimisation .

Driven by discussed limitations above imposed by fully supervised scene representations, Chapter 4 turned to scene-specific semantic representations and introduced Semantic-NeRF. Scene-specific representations are at the other end of the

spectrum from supervised ones as only in-situ annotation is required. A 3D-aware joint implicit representation of appearance, geometry and semantics is leaned in our work by augmenting NeRF with extra semantic outputs. We find that the underlying smoothness, coherence and multi-view consistency of the self-supervised geometric reconstruction enable semantics to efficiently propagate from a sparse set of noisy annotations. Therefore, a collection of posed colour images together with a small amount of semantic supervision is enough to learn a joint implicit scene representation capable of rendering accurate dense labels at novel viewpoints. This idea has been extensively validated in different applications including semantic view synthesis, semantic label denoising, semantic super-resolution, sparse label propagation and multi-view semantic fusion.

In this work we have demonstrated the benefits of encoding semantics within the 5D manifold represented by coordinate-based MLPs leveraging its strong correlation to appearance and geometry, and we are confident that not only semantics, but a wide variety of scene properties beyond semantic classes could be encoded as well such as material type, reflectance, and affordances. Representation like this could be especially useful to intelligent robots. Robots could efficiently learn such representations of their working environments from user annotations and need not rely on generalisation from prior datasets.

Semantic-NeRF requires computationally expensive off-line training and volume rendering, which prohibits real-time application and is inefficient to update with new observations. Many papers building on NeRF [Garbin et al., 2021, Lindell et al., 2021, Yu et al., 2021a] have attempted to accelerate rendering speed given a trained NeRF model, or improve training efficiency given extra information and priors [Yu et al., 2021b, Trevithick and Yang, 2021, Sucar et al., 2021]. Nevertheless, it does mean that explicit representations are useless because there are still tremendous applications where an explicit map is essential. For example, tasks including path planning, manoeuvring and obstacle avoidance usually require an

explicit and easy to process map representation. We envision the advantages of both types will eventually merged and there have been promising recent work attempting hybrid representations [Popov et al., 2020, Martel et al., 2021].

Our final contribution presented in Chapter 5 is iLabel, the first scene understanding system capable of real-time, interactive and incremental ultra-efficient labelling with only a single MLP as the underlying representation. Inspired by the success of implicit representation in SLAM [Sucar et al., 2021] and sparse label propagation [Zhi et al., 2021a], iLabel aims to address issues related to off-line approaches and bring us closer to a challenging and practical scenario. Instead of conducting off-line data collection and annotation/labelling with pre-determined concepts and training, we show that a user or robot can create a personalised semantic reconstruction tailored to their own intentions, knowledge and experience. iLabel offers the opportunity of performing all parts in an online and interactive manner, from low-level camera localisation and dense mapping up to high level semantic segmentation and system training. Meanwhile, the user keeps receiving immediate feedback from the system and can provide continuous corrections and new annotations. In addition to the overall system contribution, several novel components are also proposed including hierarchical segmentation, automatic query generation guided by uncertainty sampling. Qualitative results on a wide range of real word scenes demonstrate the ultra-efficient labelling performance of iLabel.

During experiments we have qualitatively found that similarity in appearance and geometry as well as proximity influence label transfer, which performs the best on coherent entities. Therefore, if users want to segment many instances of the same class, this is possible by clicking on each of the instances (if they differ significantly in the aforementioned properties), or by clicking on a subset of the instances if they do exhibit some similarities. In the future, we would like to explore a deeper understanding of the propagation mechanism within iLabel so we could investigate the most effective places to assign clicks, reaching higher labelling

efficiency and segmentation quality.

We would also like to work on the improving geometric performance of the iLabel system. Current camera tracking and mapping are both driven by volume rendering from implicit representation, yet they face challenges in robustly working with rapid motions and predicting crisp reconstruction. A possible path is to combine the camera tracker from mature geometric SLAM systems (e.g., ORB-SLAM) and our neural tracker: correspondences established by sparse features could serve as candidates for active sampling and provide initial camera transformations [Matsuki et al., 2021, Zuo et al., 2021]. The reconstruction from iLabel tends to be over-smooth due to the continuity and the way we extract it, i.e., repeatedly querying the MLP on a dense voxel grid and then applying marching cubes to extract a mesh. It is still under active exploration what is the best way to discover and retrieve the represented scenes. More structural and semantic priors could be injected into the mapping process, e.g., regions annotated as walls, tabletops and floors are likely to be planar and explicitly regularise the reconstruction. Hybrid representations, mentioned in the paragraphs above, are another promising direction where classical representations and neural implicit representation could be merged. For example, the lattice structure from volumetric grids allows convolution operations bringing strong inductive biases like translation equivariance, while implicit representation enables efficient encoding of complex shape topologies without an excessive memory consumption. As an exciting new research area, we expect future advances in implicit representation to further benefit iLabel.

There are of course many remaining research problems unsolved before reaching a truly useful representation for Spatial AI systems [Rosen et al., 2021]. The semantic representations throughout this thesis assume a static environment, while the extension of them to deformable, dynamic scenes are challenging and still open problems. In addition, with increasing properties and relationship among entities to present high-level comprehension, we expect that a graph-like distributed

scene representation with hierarchical structure would finally emerge to fulfil this requirement, which could also require a suitable computational hardware.

Bibliography

[Araslanov and Roth, 2020] Araslanov, N. and Roth, S. (2020). Single-stage semantic segmentation from image labels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Arbelaez et al., 2010] Arbelaez, P., Maire, M., Fowlkes, C., and Malik, J. (2010). Contour detection and hierarchical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 33(5):898–916.

[Arbeláez et al., 2014] Arbeláez, P., Pont-Tuset, J., Barron, J., Marques, F., and Malik, J. (2014). Multiscale combinatorial grouping. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Armeni et al., 2017] Armeni, I., Sax, A., Zamir, A. R., and Savarese, S. (2017). Joint 2D-3D-Semantic Data for Indoor Scene Understanding. *arXiv preprint arXiv:1702.01105*.

[Baker and Matthews, 2004] Baker, S. and Matthews, I. (2004). Lucas-Kanade 20 years on: A unifying framework: Part 1. *International Journal of Computer Vision (IJCV)*, 56(3):221–255.

[Barron et al., 2021] Barron, J. T., Mildenhall, B., Tancik, M., Hedman, P., Martin-Brualla, R., and Srinivasan, P. P. (2021). Mip-NeRF: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the International Conference on Computer Vision (ICCV)*.

[Batra et al., 2020] Batra, D., Chang, A. X., Chernova, S., Davison, A. J., Deng, J., Koltun, V., Levine, S., Malik, J., Mordatch, I., Mottaghi, R., et al. (2020). Rearrangement: A challenge for embodied ai. *arXiv*.

[Bay et al., 2006] Bay, H., Tuytelaars, T., and Van Gool, L. (2006). SURF: Speeded up robust features. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

[Bell-Kligler et al., 2019] Bell-Kligler, S., Shocher, A., and Irani, M. (2019). Blind super-resolution kernel estimation using an Internal-GAN. In *Neural Information Processing Systems (NeurIPS)*.

[Bloesch et al., 2018] Bloesch, M., Czarnowski, J., Clark, R., Leutenegger, S., and Davison, A. J. (2018). CodeSLAM — learning a compact, optimisable representation for dense visual SLAM. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Bowman et al., 2015] Bowman, S. R., Vilnis, L., Vinyals, O., Dai, A. M., Jozefowicz, R., and Bengio, S. (2015). Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349*.

[Cadena et al., 2016a] Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., Reid, I., and Leonard, J. J. (2016a). Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics (T-RO)*, 32(6):1309–1332.

[Cadena et al., 2016b] Cadena, C., Dick, A., and Reid, I. D. (2016b). Multi-modal Auto-Encoders as Joint Estimators for Robotics Scene Understanding. In *Proceedings of Robotics: Science and Systems (RSS)*.

[Calonder et al., 2010] Calonder, M., Lepetit, V., Strecha, C., and Fua, P. (2010). BRIEF: Binary Robust Independent Elementary Features. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

[Cha and Srihari, 2002] Cha, S.-H. and Srihari, S. N. (2002). On measuring the distance between histograms. *Pattern Recognition*, 35(6):1355–1370.

[Chakrabarty and Maji, 2019] Chakrabarty, P. and Maji, S. (2019). The Spectral Bias of the Deep Image Prior. In *Neural Information Processing Systems Workshops (NeurIPSW)*.

[Chen et al., 2018a] Chen, L. C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. (2018a). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 40(4):834–848.

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

[Chen et al., 2018b] Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., and Adam, H. (2018b). Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

[Chen et al., 2020] Chen, X., Lin, K.-Y., Wang, J., Wu, W., Qian, C., Li, H., and Zeng, G. (2020). Bi-directional cross-modality feature propagation with separation-and-aggregation gate for rgb-d semantic segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

[Cheng et al., 2019] Cheng, Z., Gadelha, M., Maji, S., and Sheldon, D. (2019). A Bayesian Perspective on the Deep Image Prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Cordts et al., 2016] Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., and Schiele, B. (2016). The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Czarnowski et al., 2020] Czarnowski, J., Laidlow, T., Clark, R., and Davison, A. J. (2020). Deepfactors: Real-time probabilistic dense monocular slam. *IEEE Robotics and Automation Letters (RA-L)*, 5(2):721–728.

[Dai et al., 2017a] Dai, A., Chang, A. X., Savva, M., Halber, M., Funkhouser, T., and Nießner, M. (2017a). ScanNet: Richly-annotated 3d reconstructions of indoor scene. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Dai and Nießner, 2019] Dai, A. and Nießner, M. (2019). Scan2mesh: From unstructured range scans to 3d meshes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Dai et al., 2017b] Dai, A., Nießner, M., Zollhöfer, M., Izadi, S., and Theobalt, C. (2017b). BundleFusion: Real-time Globally Consistent 3D Reconstruction using On-the-fly Surface Re-integration. *ACM Transactions on Graphics (TOG)*, 36(3):24:1–24:18.

[Dai et al., 2021] Dai, A., Siddiqui, Y., Thies, J., Valentin, J., and Nießner, M. (2021). SPSG: Self-supervised photometric scene generation from rgb-d scans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Davison, 2003] Davison, A. J. (2003). Real-Time Simultaneous Localisation and Mapping with a Single Camera. In *Proceedings of the International Conference on Computer Vision (ICCV)*.

[Davison, 2018] Davison, A. J. (2018). FutureMapping: The computational structure of Spatial AI systems. *arXiv preprint arXiv:arXiv:1803.11288*.

[Dellaert and Yen-Chen, 2020] Dellaert, F. and Yen-Chen, L. (2020). Neural volume rendering: Nerf and beyond. *arXiv preprint arXiv:2101.05204*.

[DeTone et al., 2018] DeTone, D., Malisiewicz, T., and Rabinovich, A. (2018). Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*.

[Doersch, 2016] Doersch, C. (2016). Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908*.

[Doersch et al., 2015] Doersch, C., Gupta, A., and Efros, A. A. (2015). Unsupervised visual representation learning by context prediction. In *Proceedings of the International Conference on Computer Vision (ICCV)*.

[Donahue and Simonyan, 2019] Donahue, J. and Simonyan, K. (2019). Large scale adversarial representation learning. In *Neural Information Processing Systems (NeurIPS)*.

[Dosovitskiy et al., 2021] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

[Eigen and Fergus, 2015] Eigen, D. and Fergus, R. (2015). Predicting Depth, Surface Normals and Semantic Labels with a Common Multi-Scale Convolutional

Architecture. In *Proceedings of the International Conference on Computer Vision (ICCV)*.

[Engel et al., 2017] Engel, J., Koltun, V., and Cremers, D. (2017). Direct sparse odometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*.

[Engel et al., 2014] Engel, J., Schoeps, T., and Cremers, D. (2014). LSD-SLAM: Large-scale direct monocular SLAM. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

[Engel, 2017] Engel, J.-J. (2017). *Large-Scale Direct SLAM and 3D Reconstruction in Real-Time*. PhD thesis, Technische Universität München.

[Eslami et al., 2018] Eslami, S. A., Rezende, D. J., Besse, F., Viola, F., Morcos, A. S., Garnelo, M., Ruderman, A., Rusu, A. A., Danihelka, I., Gregor, K., et al. (2018). Neural scene representation and rendering. *Science*, 360(6394):1204–1210.

[Everingham et al., 2010] Everingham, M., Van Gool, L., Williams, C. K., Winn, J., and Zisserman, A. (2010). The pascal visual object classes (VOC) challenge. *International Journal of Computer Vision (IJCV)*, 2:303–338.

[Gal, 2016] Gal, Y. (2016). Uncertainty in deep learning.

[Gandelsman et al., 2019] Gandelsman, Y., Shocher, A., and Irani, M. (2019). "double-dip": Unsupervised image decomposition via coupled deep-image-priors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Garbin et al., 2021] Garbin, S. J., Kowalski, M., Johnson, M., Shotton, J., and Valentin, J. (2021). Fastnerf: High-fidelity neural rendering at 200fps. *arXiv preprint arXiv:2103.10380*.

[Geiger et al., 2012] Geiger, A., Lenz, P., and Urtasun, R. (2012). Are we ready for autonomous driving? The KITTI vision benchmark suite. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

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

[Harris and Stephens, 1988] Harris, C. G. and Stephens, M. (1988). A Combined Corner and Edge Detector. In *Proceedings of the Alvey Vision Conference*, pages 147–151.

[Hazirbas et al., 2016] Hazirbas, C., Ma, L., Domokos, C., and Cremers, D. (2016). FuseNet: incorporating depth into semantic segmentation via fusion-based CNN architecture. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*.

[He et al., 2017a] He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017a). Mask r-cnn. In *Proceedings of the International Conference on Computer Vision (ICCV)*.

[He et al., 2015] He, K., Zhang, X., Ren, S., and Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the International Conference on Computer Vision (ICCV)*.

[He et al., 2016] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[He et al., 2017b] He, Y., Chiu, W.-C., Keuper, M., and Fritz, M. (2017b). STD2P: RGBD Semantic Segmentation Using Spatio-Temporal Data-Driven

Pooling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Hermans et al., 2014] Hermans, A., Floros, G., and Leibe, B. (2014). Dense 3D semantic mapping of indoor scenes from RGB-D images. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*.

[Hiroaki Aizawa, 2021] Hiroaki Aizawa, Yukihiro Domae, K. K. (2021). Hierarchical pyramid representations for semantic segmentation. *arXiv preprint arXiv 2104.01792*.

[Hornik, 1989] Hornik, K. (1989). Multilayer Feedforward Networks are Universal Approximators. *Journal of Neural Networks*, 2:359–366.

[Hou et al., 2021] Hou, J., Xie, S., Graham, B., Dai, A., and Nießner, M. (2021). Pri3d: Can 3d priors help 2d representation learning? In *Proceedings of the International Conference on Computer Vision (ICCV)*.

[Hsu et al., 2019] Hsu, K.-J., Lin, Y.-Y., and Chuang, Y.-Y. (2019). Deepco3: Deep instance co-segmentation by co-peak search and co-saliency detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Hung et al., 2019] Hung, W.-C., Jampani, V., Liu, S., Molchanov, P., Yang, M.-H., and Kautz, J. (2019). Scops: Self-supervised co-part segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Isola et al., 2017] Isola, P., Zhu, J., Zhou, T., and Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Kahler et al., 2015] Kahler, O., Prisacariu, V. A., Ren, C. Y., Sun, X., Torr, P. H. S., and Murray, D. W. (2015). Very High Frame Rate Volumetric Integration of Depth Images on Mobile Device. In *Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR)*.

[Kahler and Reid, 2013] Kahler, O. and Reid, I. (2013). Efficient 3d scene labeling using fields of trees. In *Proceedings of the International Conference on Computer Vision (ICCV)*.

[Kendall and Gal, 2017] Kendall, A. and Gal, Y. (2017). What uncertainties do we need in bayesian deep learning for computer vision? In *Neural Information Processing Systems (NeurIPS)*.

[Kendall et al., 2018] Kendall, A., Gal, Y., and Cipolla, R. (2018). Multi-Task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Kerl et al., 2013] Kerl, C., Sturm, J., and Cremers, D. (2013). Robust odometry estimation for RGB-D cameras. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*.

[Kingma and Ba, 2015] Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

[Kingma and Welling, 2014] Kingma, D. P. and Welling, M. (2014). Auto-Encoding Variational Bayes. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

[Klein and Murray, 2007] Klein, G. and Murray, D. W. (2007). Parallel Tracking and Mapping for Small AR Workspaces. In *Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR)*.

[Kohl et al., 2018] Kohl, S. A., Romera-Paredes, B., Meyer, C., De Fauw, J., Ledsam, J. R., Maier-Hein, K. H., Eslami, S., Rezende, D. J., and Ronneberger, O. (2018). A Probabilistic U-Net for Segmentation of Ambiguous Images. In *Neural Information Processing Systems (NeurIPS)*.

[Kohli et al., 2020] Kohli, A., Sitzmann, V., and Wetzstein, G. (2020). Inferring semantic information with 3d neural scene representations. In *Proceedings of the International Conference on 3D Vision (3DV)*.

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

[Krizhevsky et al., 2012] Krizhevsky, A., Sutskever, I., and Hinton, G. (2012). ImageNet classification with deep convolutional neural networks. In *Neural Information Processing Systems (NeurIPS)*.

[Ku et al., 2018] Ku, J., Harakeh, A., and Waslander, S. L. (2018). In defense of classical image processing: Fast depth completion on the cpu. In *Proceedings of the Canadian Conference on Computer and Robot Vision (CRV)*.

[Leutenegger et al., 2011] Leutenegger, S., Chli, M., and Siegwart, R. (2011). BRISK: Binary robust invariance scalable keypoints. In *Proceedings of the International Conference on Computer Vision (ICCV)*.

[Li et al., 2020] Li, K., Rünz, M., Tang, M., Ma, L., Kong, C., Schmidt, T., Reid, I., Agapito, L., Straub, J., Lovegrove, S., et al. (2020). Frodo: From detections to 3d objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Lin et al., 2017] Lin, G., Milan, A., Chunhua, S., and Reid, I. (2017). RefineNet: Multi-Path Refinement Networks for High-Resolution Semantic Segmenta-

tion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

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

[Lindell et al., 2021] Lindell, D. B., Martel, J. N., and Wetzstein, G. (2021). Autoint: Automatic integration for fast neural volume rendering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Liu et al., 2019] Liu, S., Johns, E., and Davison, A. J. (2019). End-to-end multi-task learning with attention. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Liu et al., 2020] Liu, S., Zhang, Y., Peng, S., Shi, B., Pollefeys, M., and Cui, Z. (2020). Dist: Rendering deep implicit signed distance function with differentiable sphere tracing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Liu et al., 2021] Liu, S., Zhi, S., Johns, E., and Davison, A. J. (2021). Bootstrapping semantic segmentation with regional contrast. *arXiv preprint arXiv:2104.04465*.

[Long et al., 2015] Long, J., Shelhamer, E., and Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Lowe, 2004] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision (IJCV)*, 60(2):91–110.

[Ma et al., 2017] Ma, L., Stückler, J., Kerl, C., and Cremers, D. (2017). Multi-View Deep Learning for Consistent Semantic Mapping with RGB-D Cameras. In *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS)*.

[Maninis et al., 2016] Maninis, K.-K., Pont-Tuset, J., Arbeláez, P., and Van Gool, L. (2016). Convolutional oriented boundaries. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

[Martel et al., 2021] Martel, J. N., Lindell, D. B., Lin, C. Z., Chan, E. R., Monteiro, M., and Wetzstein, G. (2021). Acorn: Adaptive coordinate networks for neural scene representation. *Proceedings of SIGGRAPH*.

[Martin et al., 2001] Martin, D., Fowlkes, C., Tal, D., and Malik, J. (2001). A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings of the International Conference on Computer Vision (ICCV)*.

[Martin-Brualla et al., 2020] Martin-Brualla, R., Radwan, N., Sajjadi, M. S., Barron, J. T., Dosovitskiy, A., and Duckworth, D. (2020). Nerf in the wild: Neural radiance fields for unconstrained photo collections. *arXiv preprint arXiv:2008.02268*.

[Martin-Brualla et al., 2021] Martin-Brualla, R., Radwan, N., Sajjadi, M. S., Barron, J. T., Dosovitskiy, A., and Duckworth, D. (2021). NeRF in the wild: Neural radiance fields for unconstrained photo collections. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Matsuki et al., 2021] Matsuki, H., Scona, R., Czarnowski, J., and Davison, A. J. (2021). Codemapping: Real-time dense mapping for sparse slam using compact scene representations. *IEEE Robotics and Automation Letters (RA-L)*, 6(4):7105–7112.

[McCormac et al., 2018] McCormac, J., Clark, R., Bloesch, M., Davison, A. J., and Leutenegger, S. (2018). Fusion++:volumetric object-level slam. In *Proceedings of the International Conference on 3D Vision (3DV)*.

[McCormac et al., 2017a] McCormac, J., Handa, A., Davison, A. J., and Leutenegger, S. (2017a). SemanticFusion: Dense 3D semantic mapping with convolutional neural networks. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*.

[McCormac et al., 2017b] McCormac, J., Handa, A., Leutenegger, S., and Davison, A. J. (2017b). SceneNet RGB-D: Can 5M synthetic images beat generic ImageNet pre-training on indoor segmentation? In *Proceedings of the International Conference on Computer Vision (ICCV)*.

[Mescheder et al., 2019] Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., and Geiger, A. (2019). Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Microsoft Corp, 2010] Microsoft Corp (2010). Microsoft Kinect. https://www.xbox.com/en-US/xbox-one/accessories/kinect.

[Miksik et al., 2015] Miksik, O., Vineet, V., Lidegaard, M., Prasaath, R., Nießner, M., Golodetz, S., Hicks, S. L., Pérez, P., Izadi, S., and Torr, P. H. (2015). The semantic paintbrush: Interactive 3d mapping and recognition in large outdoor spaces. In *ACM Conference on Human Factors in Computing Systems (CHI)*.

[Mildenhall et al., 2020] Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., and Ng, R. (2020). NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

[Mo et al., 2019] Mo, K., Zhu, S., Chang, A. X., Yi, L., Tripathi, S., Guibas, L. J., and Su, H. (2019). Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Mur-Artal and Tardós, 2014] Mur-Artal, R. and Tardós, J. D. (2014). ORB-SLAM: Tracking and Mapping Recognizable Features. In *Workshop on Multi View Geometry in Robotics (MVIGRO) - RSS 2014*.

[Mur-Artal and Tardós, 2017] Mur-Artal, R. and Tardós, J. D. (2017). ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. *IEEE Transactions on Robotics (T-RO)*, 33(5):1255–1262.

[Nakajima et al., 2019] Nakajima, Y., Kang, B., Saito, H., and Kitani, K. (2019). Incremental class discovery for semantic segmentation with rgbd sensing. In *Proceedings of the International Conference on Computer Vision (ICCV)*.

[Nakajima et al., 2018] Nakajima, Y., Tateno, K., Tombari, F., and Saito, H. (2018). Fast and accurate semantic mapping through geometric-based incremental segmentation. In *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS)*.

[Narita et al., 2019] Narita, G., Seno, T., Ishikawa, T., and Kaji, Y. (2019). Panopticfusion: Online volumetric semantic mapping at the level of stuff and things. In *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS)*.

[Newcombe et al., 2011a] Newcombe, R. A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A. J., Kohli, P., Shotton, J., Hodges, S., and Fitzgibbon, A. (2011a). KinectFusion: Real-Time Dense Surface Mapping and Tracking. In *Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR)*.

[Newcombe et al., 2011b] Newcombe, R. A., Lovegrove, S., and Davison, A. J. (2011b). DTAM: Dense Tracking and Mapping in Real-Time. In *Proceedings of the International Conference on Computer Vision (ICCV)*.

[Nguyen et al., 2017] Nguyen, D. T., Hua, B.-S., Yu, L.-F., and Yeung, S.-K. (2017). A robust 3d-2d interactive tool for scene segmentation and annotation. *IEEE Transactions on Visualization and Computer Graphics (VGC)*.

[Nguyen-Phuoc et al., 2019] Nguyen-Phuoc, T., Li, C., Theis, L., Richardt, C., and Yang, Y.-L. (2019). Hologan: Unsupervised learning of 3d representations from natural images. In *Proceedings of the International Conference on Computer Vision (ICCV)*.

[Nicholson et al., 2018] Nicholson, L., Milford, M., and Sünderhauf, N. (2018). QuadricSLAM: Constrained Dual Quadrics from Object Detections as Landmarks in Object-oriented SLAM. *IEEE Robotics and Automation Letters (RA-L)*.

[Nießner et al., 2013] Nießner, M., Zollhöfer, M., Izadi, S., and Stamminger, M. (2013). Real-time 3D Reconstruction at Scale using Voxel Hashing. In *Proceedings of SIGGRAPH*.

[Oechsle et al., 2021] Oechsle, M., Peng, S., and Geiger, A. (2021). Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *Proceedings of the International Conference on Computer Vision (ICCV)*.

[Park et al., 2019] Park, J. J., Florence, P., Straub, J., Newcombe, R., and Lovegrove, S. (2019). Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Park et al., 2020] Park, K., Mousavian, A., Xiang, Y., and Fox, D. (2020). Latentfusion: End-to-end differentiable reconstruction and rendering for unseen

object pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Paszke et al., 2019] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. (2019). Pytorch: An imperative style, high-performance deep learning library. In *Neural Information Processing Systems (NeurIPS)*.

[Pearl, 2017] Pearl, J. (2017). Theoretical impediments to machine learning, with seven sparks from the causal revolution. Technical report, University of California, Los Angeles. Technical Report R-275.

[Peng et al., 2020] Peng, S., Niemeyer, M., Mescheder, L., Pollefeys, M., and Geiger, A. (2020). Convolutional occupancy networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

[Pham et al., 2018a] Pham, T., Do, T.-T., Carneiro, G., Reid, I., et al. (2018a). Bayesian semantic instance segmentation in open set world. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

[Pham et al., 2018b] Pham, T. T., Do, T.-T., Sünderhauf, N., and Reid, I. (2018b). Scenecut: Joint geometric and object segmentation for indoor scenes. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*.

[Pizzoli et al., 2014] Pizzoli, M., Forster, C., and Scaramuzza, D. (2014). REMODE: Probabilistic, monocular dense reconstruction in real time. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*.

[Popov et al., 2020] Popov, S., Bauszat, P., and Ferrari, V. (2020). Corenet: Coherent 3d scene reconstruction from a single rgb image. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

[Pradeep et al., 2013] Pradeep, V., Rhemann, C., Izadi, S., Zach, C., Bleyer, M., and Bathiche, S. (2013). MonoFusion: Real-time 3D reconstruction of small scenes with a single web camera. In *Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR)*.

[Qi et al., 2017a] Qi, C. R., Su, H., Mo, K., and Guibas, L. J. (2017a). Pointnet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Qi et al., 2016] Qi, C. R., Su, H., Nießner, M., Dai, A., Yan, M., and Guibas, L. J. (2016). Volumetric and multi-view cnns for object classification on 3d data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Qi et al., 2017b] Qi, C. R., Yi, L., Su, H., and Guibas, L. J. (2017b). Pointnet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In *Neural Information Processing Systems (NeurIPS)*.

[Redmon et al., 2016] Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Reiser et al., 2021] Reiser, C., Peng, S., Liao, Y., and Geiger, A. (2021). Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps. In *Proceedings of the International Conference on Computer Vision (ICCV)*.

[Ren et al., 2020] Ren, P., Xiao, Y., Chang, X., Huang, P.-Y., Li, Z., Chen, X., and Wang, X. (2020). A survey of deep active learning. *arXiv preprint arXiv:2009.00236*.

[Ronneberger et al., 2015] Ronneberger, O., Fischer, P., and Brox, T. (2015). U-Net: Convolutional networks for biomedical image segmentation. In *Proceed-*

ings of the International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI).

[Rosen et al., 2021] Rosen, D. M., Doherty, K. J., Terán Espinoza, A., and Leonard, J. J. (2021). Advances in inference and representation for simultaneous localization and mapping. *Annual Review of Control, Robotics, and Autonomous Systems*, 4:215–242.

[Rosten and Drummond, 2006] Rosten, E. and Drummond, T. (2006). Machine learning for high-speed corner detection. In *Proceedings of the European Conference on Computer Vision (ECCV).*

[Rublee et al., 2011] Rublee, E., Rabaud, V., Konolige, K., and Bradski, G. (2011). ORB: an efficient alternative to SIFT or SURF. In *Proceedings of the International Conference on Computer Vision (ICCV).*

[Runz et al., 2018] Runz, M., Buffier, M., and Agapito, L. (2018). Maskfusion: Real-time Recognition, Tracking and Reconstruction of Multiple Moving Objects. In *Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR).*

[Russakovsky et al., 2015] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252.

[Salas-Moreno et al., 2013] Salas-Moreno, R. F., Newcombe, R. A., Strasdat, H., Kelly, P. H. J., and Davison, A. J. (2013). SLAM++: Simultaneous Localisation and Mapping at the Level of Objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

[Savva et al., 2019] Savva, M., Kadian, A., Maksymets, O., Zhao, Y., Wijmans, E., Jain, B., Straub, J., Liu, J., Koltun, V., Malik, J., et al. (2019). Habitat: A

platform for embodied ai research. In *Proceedings of the International Conference on Computer Vision (ICCV)*.

[Schmidt et al., 2017] Schmidt, T., Newcombe, R., and Fox, D. (2017). Self-supervised visual descriptor learning for dense correspondence. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*.

[Settles, 2009] Settles, B. (2009). Active learning literature survey.

[Shaham et al., 2019] Shaham, T. R., Dekel, T., and Michaeli, T. (2019). SinGAN: Learning a generative model from a single natural image. In *Proceedings of the International Conference on Computer Vision (ICCV)*.

[Shi and Tomasi, 1994] Shi, J. and Tomasi, C. (1994). Good features to track. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Shocher et al., 2019] Shocher, A., Bagon, S., Isola, P., and Irani, M. (2019). InGAN: Capturing and retargeting the "DNA" of a natural image. In *Proceedings of the International Conference on Computer Vision (ICCV)*.

[Shocher et al., 2018] Shocher, A., Cohen, N., and Irani, M. (2018). "zero-shot" super-resolution using deep internal learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

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

[Simonyan and Zisserman, 2015] Simonyan, K. and Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

[Sitzmann et al., 2020] Sitzmann, V., Martel, J., Bergman, A., Lindell, D., and Wetzstein, G. (2020). Implicit neural representations with periodic activation functions. In *Neural Information Processing Systems (NeurIPS)*.

[Sitzmann et al., 2021] Sitzmann, V., Rezchikov, S., Freeman, W. T., Tenenbaum, J. B., and Durand, F. (2021). Light field networks: Neural scene representations with single-evaluation rendering. *arXiv preprint arXiv:2106.02634*.

[Sitzmann et al., 2019a] Sitzmann, V., Thies, J., Heide, F., Nießner, M., Wetzstein, G., and Zollhofer, M. (2019a). Deepvoxels: Learning persistent 3d feature embeddings. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Sitzmann et al., 2019b] Sitzmann, V., Zollhöfer, M., and Wetzstein, G. (2019b). Scene representation networks: Continuous 3d-structure-aware neural scene representations. In *Neural Information Processing Systems (NeurIPS)*.

[Sohn et al., 2015] Sohn, K., Lee, H., and Yan, X. (2015). Learning structured output representation using deep conditional generative models. In *Neural Information Processing Systems (NeurIPS)*.

[Sønderby et al., 2016] Sønderby, C. K., Raiko, T., Maaløe, L., Sønderby, S. K., and Winther, O. (2016). How to train deep variational autoencoders and probabilistic ladder networks. *arXiv preprint arXiv:1602.02282*.

[Song et al., 2015] Song, S., Lichtenberg, S. P., and Xiao, J. (2015). SUN RGB-D: A RGB-D scene understanding benchmark suite. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Srinivasan et al., 2021] Srinivasan, P. P., Deng, B., Zhang, X., Tancik, M., Mildenhall, B., and Barron, J. T. (2021). NeRV: Neural reflectance and visibility fields for relighting and view synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Straub et al., 2019] Straub, J., Whelan, T., Ma, L., Chen, Y., Wijmans, E., Green, S., Engel, J. J., Mur-Artal, R., Ren, C., Verma, S., Clarkson, A., Yan, M., Budge, B., Yan, Y., Pan, X., Yon, J., Zou, Y., Leon, K., Carter, N., Briales, J., Gillingham, T., Mueggler, E., Pesqueira, L., Savva, M., Batra, D., Strasdat, H. M., Nardi, R. D., Goesele, M., Lovegrove, S., and Newcombe, R. (2019). The Replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*.

[Stückler and Behnke, 2014] Stückler, J. and Behnke, S. (2014). Multi-resolution surfel maps for efficient dense 3d modeling and tracking. *Journal of Visual Communication and Image Representation*, 25(1):137–147.

[Sucar et al., 2021] Sucar, E., Liu, S., Ortiz, J., and Davison, A. J. (2021). iMAP: Implicit mapping and positioning in real-time. In *Proceedings of the International Conference on Computer Vision (ICCV)*.

[Sucar et al., 2020] Sucar, E., Wada, K., and Davison, A. (2020). NodeSLAM: Neural object descriptors for multi-view shape reconstruction. In *Proceedings of the International Conference on 3D Vision (3DV)*.

[Sünderhauf et al., 2017] Sünderhauf, N., Pham, T. T., Latif, Y., Milford, M., and Reid, I. (2017). Meaningful maps with object-oriented semantic mapping. In *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS)*.

[Tancik et al., 2020] Tancik, M., Srinivasan, P. P., Mildenhall, B., Fridovich-Keil, S., Raghavan, N., Singhal, U., Ramamoorthi, R., Barron, J. T., and Ng, R. (2020). Fourier features let networks learn high frequency functions in low dimensional domains. In *Neural Information Processing Systems (NeurIPS)*.

[Tateno et al., 2017] Tateno, K., Tombari, F., Laina, I., and Navab, N. (2017). CNN-SLAM: Real-time dense monocular slam with learned depth prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Trevithick and Yang, 2021] Trevithick, A. and Yang, B. (2021). GRF: Learning a general radiance field for 3d scene representation and rendering. In *Proceedings of the International Conference on Computer Vision (ICCV)*.

[Ulyanov et al., 2018] Ulyanov, D., Vedaldi, A., and Lempitsky, V. (2018). Deep image prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Valentin et al., 2015] Valentin, J., Vineet, V., Cheng, M.-M., Kim, D., Shotton, J., Kohli, P., Nießner, M., Criminisi, A., Izadi, S., and Torr, P. (2015). Semanticpaint: Interactive 3d labeling and learning at your fingertips. *ACM Transactions on Graphics (TOG)*, 34(5).

[Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Neural Information Processing Systems (NeurIPS)*.

[Wald et al., 2020] Wald, J., Dhamo, H., Navab, N., and Tombari, F. (2020). Learning 3d semantic scene graphs from 3d indoor reconstructions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Weerasekera et al., 2017] Weerasekera, C. S., Latif, Y., Garg, R., and Reid, I. (2017). Dense monocular reconstruction using surface normals. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*.

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

[Whelan et al., 2016] Whelan, T., Salas-Moreno, R. F., Glocker, B., Davison, A. J., and Leutenegger, S. (2016). ElasticFusion: Real-time dense SLAM

and light source estimation. *International Journal of Robotics Research (IJRR)*, 35(14):1697–1716.

[Wu et al., 2015] Wu, J., Yildirim, I., Freeman, W., and Tenenbaum, J. (2015). Perceiving physical object properties by integrating a physics engine with deep learning. In *Neural Information Processing Systems (NeurIPS)*.

[Wu et al., 2021] Wu, S.-C., Wald, J., Tateno, K., Navab, N., and Tombari, F. (2021). SceneGraphFusion: Incremental 3d scene graph prediction from rgb-d sequences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Xiang and Fox, 2017] Xiang, Y. and Fox, D. (2017). DA-RNN: Semantic mapping with data associated recurrent neural networks. In *Proceedings of Robotics: Science and Systems (RSS)*.

[Xiao and Quan, 2009] Xiao, J. and Quan, L. (2009). Multiple view semantic segmentation for street view images. In *Proceedings of the International Conference on Computer Vision (ICCV)*.

[Xie and Tu, 2015] Xie, S. and Tu, Z. (2015). Holistically-nested edge detection. In *Proceedings of the International Conference on Computer Vision (ICCV)*.

[Yu et al., 2021a] Yu, A., Li, R., Tancik, M., Li, H., Ng, R., and Kanazawa, A. (2021a). Plenoctrees for real-time rendering of neural radiance fields. In *Proceedings of the International Conference on Computer Vision (ICCV)*.

[Yu et al., 2021b] Yu, A., Ye, V., Tancik, M., and Kanazawa, A. (2021b). pixelNeRF: Neural radiance fields from one or few images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Zamir et al., 2020] Zamir, A. R., Sax, A., Cheerla, N., Suri, R., Cao, Z., Malik, J., and Guibas, L. J. (2020). Robust learning through cross-task consistency. In

Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[Zamir et al., 2018] Zamir, A. R., Sax, A., Shen, W. B., Guibas, L. J., Malik, J., and Savarese, S. (2018). Taskonomy: Disentangling task transfer learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

[Zhang et al., 2020] Zhang, K., Riegler, G., Snavely, N., and Koltun, V. (2020). NeRF++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492.*

[Zhang et al., 2016] Zhang, R., Isola, P., and Efros, A. A. (2016). Colorful image colorization. In *Proceedings of the European Conference on Computer Vision (ECCV).*

[Zhao et al., 2017a] Zhao, C., Sun, L., and Stolkin, R. (2017a). A fully end-to-end deep learning approach for real-time simultaneous 3d reconstruction and material recognition. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA).*

[Zhao et al., 2017b] Zhao, H., Shi, J., Qi, X., Wang, X., and Jia, J. (2017b). Pyramid scene parsing network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

[Zhi et al., 2019] Zhi, S., Bloesch, M., Leutenegger, S., and Davison, A. J. (2019). SceneCode: Monocular dense semantic reconstruction using learned encoded scene representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

[Zhi et al., 2021a] Zhi, S., Laidlow, T., Leutenegger, S., and Davison, A. J. (2021a). In-place scene labelling and understanding with implicit scene representation. In *Proceedings of the International Conference on Computer Vision (ICCV).*

[Zhi et al., 2021b] Zhi, S., Sucar, E., Mouton, A., Haughton, I., Laidlow, T., and Davison, A. J. (2021b). iLabel: Interactive implicit scene labelling and learning in real-time. *Under Submission*.

[Zhou et al., 2018] Zhou, Y., Zhu, Y., Ye, Q., Qiu, Q., and Jiao, J. (2018). Weakly supervised instance segmentation using class peak response. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Zou et al., 2021] Zou, Y., Zhang, Z., Zhang, H., Li, C.-L., Bian, X., Huang, J.-B., and Pfister, T. (2021). PseudoSeg: Designing pseudo labels for semantic segmentation. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

[Zuo et al., 2021] Zuo, X., Merrill, N., Li, W., Liu, Y., Pollefeys, M., and Huang, G. (2021). CodeVIO: Visual-inertial odometry with learned optimizable dense depth. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*.