University of Brasilia

Institute of Exact Sciences
Department of Computer Science

Towards Complete 3D Indoor Scene Understanding from a Single Point-of-View

Aloisio Dourado Neto

Thesis presented for conclusion of the Ph.D. Program in Computer Science

Supervisor
Prof. Dr. Teófilo Emidio de Campos

Brasilia
2022

Ficha catalográfica elaborada automaticamente,
com os dados fornecidos pelo(a) autor(a)

Dourado Neto, Aloisio
DD739t Towards Complete 3D Indoor Scene Understanding from a
Single Point-of-View / Aloisio Dourado Neto; orientador
Teofilo Emidio de Campos. -- Brasília, 2022.
154 p.

Tese(Doutorado em Informática) -- Universidade de
Brasília, 2022.

1. Visão Computacional. 2. Compreensão de Cenas 3D. 3.
Complementação Semântica de Cenas. 4. Redes Neurais
Convolucionais. I. Emidio de Campos, Teofilo, orient. II.
Título.

University of Brasilia

Institute of Exact Sciences
Department of Computer Science

Towards Complete 3D Indoor Scene Understanding from a Single Point-of-View

Aloisio Dourado Neto

Thesis presented for conclusion of the Ph.D. Program in Computer Science

Prof. Dr. Teófilo Emidio de Campos (Supervisor)
CIC/UnB

Dr. Gabriela Csurka Naver Labs Europe	Prof. Dr. Anderson Rocha Unicamp
Prof. Dr. Bruno Luiggi Macchiavello Espinoza CIC/UnB	Prof. Dr. Vinicius Ruela Pereira Borges (suplente) CIC/UnB

Prof. Dr. Ricardo Jacobi
Computer Science Graduate Program Coordinator

Brasilia, November 5, 2022

Dedication

I dedicate this work to my family.

To my loving parents, Luiz and Helena who always encouraged me to continue studying, a special feeling of gratitude.

To my beloved wife, Camila and my dear kids, Aloisio, Juliana e Lucas, many thanks for the support during all this time. Without your understanding and unconditional love it would be impossible to finish my PhD. I love you all!

Acknowledgments

Thanks to Dr. Teo de Campos, my supervisor, for believing in me from the beginning and for the excellent guidance during all these years.

Thanks to the UnBVision group for the insightful chats and precious help.

Thanks to Dr. Adrian Hilton, from University of Surrey, and Dr. Hansung Kim, from University of Southampton, for the supervision during my stay in the UK and for the further collaboration. It was very fruitful.

This work was carried out with the support of the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES), through access to the Portal de Periódicos.

Abstract

While reasoning about scenes in 3D is a natural task for humans, it remains a challenging problem in Computer Vision, despite the great advances we have seen in the last few decades. Automatic understanding of the complete 3D geometry of an indoor scene and the semantics of each occupied 3D voxel many applications, such as robotics, surveillance, assistive computing, augmented reality, and immersive spatial audio reproduction. With this research project, we intend to contribute to enhancing the current computational results on scene understanding, both in accuracy and coverage. We focus on the task of Semantic Scene Completion, one of the most complete tasks related to scene understanding, as it aims to infer the complete 3D geometry and the semantic labels of each voxel in a scene, including occluded regions. In this thesis, we formulate and access a series of hypotheses to improve current Before getting into the problem of 3D SSC, we explored Domain Adaptation methods to address problems related to the scarcity of labeled training data in image segmentation tasks in 2D to further apply to 3D. In the 3D SSC domain, we introduced and evaluated a completely new way to explore the RGB information provided in the RGB-D input and complement the depth information. We showed that this leads to an enhancement in the segmentation of hard-to-detect objects in the scene. We further advanced in the use of RGB data by using semantic priors from the 2D image as semantic guidance to the 3D segmentation and completion in a multi-modal data-augmented 3D FCN. We complete the contributions related to quality improvement by combining a Domain Adaptation technique accessed in the earlier stages of the research to our multi-modal network with impressive results. Regarding the scene coverage, which today is restricted to the limited field of view of regular RGB-D sensors like Microsoft Kinect, we complete our contributions with a new approach to extend the current methods to 360° using panoramic RGB images and corresponding depth maps from 360-degree sensors or stereo 3D 360-degree cameras.

Keywords: Computer Vision, 3D Scene Understanding, Semantic Scene Completion, Convolutional Neural Networks

Resumo

A nossa percepção visual é a habilidade de interpretar e inferir informações sobre o ambiente que nos cerca usando a luz refletida que entra em nossos olhos através da córnea e atinge a retina. Por meio do nosso sistema de visão binocular, nós podemos naturalmente realizar tarefas como identificar o tipo de ambiente no qual nos encontramos, estimar a distância dos objetos na cena e ainda identificar quais objetos são estes. Para os humanos, realizar inferências como estas sobre cenas em 3D é algo natural. Entretanto, em Visão Computacional, este é ainda um problema muito desafiador e com muito espaço para melhorias, para o qual existem inúmeras aplicações, incluindo robótica, segurança, computação assistiva, realidade aumentada e reprodução de áudio espacial imersivo.

Visando contribuir para o alcance de uma compreensão automática de cenas mais efetiva e abrangente, nesta tese, nós elegemos como foco a tarefa de Complementação Semântica de Cenas (em inglês *Semantic Scene Completion*), por ser uma das mais completas tarefas relacionadas à compreensão de cenas, já que visa inferir a geometria completa do campo de visão da cena e os rótulos semânticos de cada um dos voxels do espaço 3D sob análise, incluindo regiões oclusas. A entrada para esta tarefa é uma imagem RGB-D, que consiste em uma imagem RGB regular adicionada de um quarto canal contendo um mapa de profundidade da cena. Tal imagem geralmente é obtida por meio de sensores de luz estruturada como o Microsoft Kinect, mas pode também ser obtida por câmeras estereoscópicas associadas a um algoritmo de estimação de profundidade. As redes profundas já atingiram os níveis de acurácia humana em uma série de tarefas da visão computacional. Entretanto, este não é o caso dos modelos de compreensão semântica de cenas. Nós identificamos quatro principais deficiências nas soluções atuais:

- a parte RGB e outros modos das imagens RGB-D não são completamente explorados;
- algumas técnicas de treinamento amplamente utilizadas em 2D têm sido negligenciadas em 3D;
- nenhum dos trabalhos anteriores que identificamos exploraram o uso de dados não rotulados por meio de treinamento semi-supervisionado;

- as soluções atuais são limitadas ao campo de visão restrito dos sensores de profundidade

Assim sendo, o objetivo geral deste trabalho é propor, implementar e avaliar novas ferramentas e modelos que possam elevar o nível das soluções em Complementação Semântica de Cenas, no sentido de uma compreensão ampla da cena. Nossos objetivos específicos são:

1. avaliar os benefícios das técnicas de adaptação domínio e treinamento semi- supervisionado no contexto de segmentação de imagens em 2D, visando posteriormente explorar o uso de dados não rotulados em 3D;
2. aplicar as tendências atuais dos protocolos de treinamento de redes 2D profundas, nas redes 3D de Complementação Semântica de Cenas;
3. propor e avaliar um novo modelo de rede 3D que utilize a informação RGB presente nas imagens RGB-D e supere os problemas de esparsidade de dados ao projetar dados em 2D para 3D;
4. propor e avaliar uma rede neural multimodal para explorar os múltiplos modos da imagem RGB-D;
5. propor e avaliar os benefícios do uso de dados não rotulados no treinamento semi-supervisionado de redes 3D.
6. propor e avaliar uma solução para a realização de complementação semântica de cenas em 3D usando datasets RGB-D convencionais para treinamento.

Os primeiros trabalhos de Visão Computacional remontam aos anos 70. Entretanto, dado o baixo poder computacional das máquinas da época, as tarefas possíveis de serem realizadas eram muito simples e os resultados eram pobres. Os primeiros resultados promissores começaram a surgir a partir do ano 2000, com o aumento do poder computacional, com um salto representativo em 2012, com a disponibilização de grandes bases de dados de imagens para treinamento. No Capítulo 2 detalhamos este histórico da evolução do campo da Inteligência Artificial e da Visão Computacional, desde os seus pioneiros até as grandes redes convolucionais profundas atuais. Neste capítulo, também apresentamos conceitos importantes relativos à visão 3D, estimação de profundidade e codificação de volumes, importantes para a compreensão de cenas.

A capacidade de realização de inferências sobre cenas em 3D é considerada um dos problemas fundamentais da Visão Computacional e a tarefa de Segmentação Semântica de Cenas é uma das mais ambiciosas, no sentido de uma compreensão completa da cena. No

Capítulo 3, referente aos trabalhos anteriores, apresentamos a bibliografia estreitamente relacionada com o nosso trabalho, com destaque para o trabalho seminal em Segmentação Semântica de Cenas que introduziu uma série de inovações, que são usadas até hoje, tais como: o uso de convoluções 3D dilatadas para ampliar o campo receptivo e ampliar a captura de contexto; a codificação F-TSDF para destacar as regiões de maior interesse da cena; e SUNCG, um dataset sintético de cenas 3D, muito útil no treinamento das redes. Além disso, o capítulo ainda apresenta trabalhos relativos à compreensão de cenas panorâmicas e os datasets utilizados neste trabalho.

Considerando que o estado da arte atual para este problema utiliza redes neurais totalmente convolucionais (em inglês *Fully Convolutional Network* - FCN), que normalmente requerem quantidades elevadas de dados para treinamento, e considerando também a dificuldade de obtenção de dados totalmente rotulados em 3D, antes de entrar no problema de Complementação Semântica em 3D propriamente dito, no Capítulo 4, nós exploramos alternativas para contornar este dificuldade em um problema mais simples: segmentação semântica em 2D.

Em 2D, nós exploramos o uso de Transferência de Aprendizado (*Transfer Learning*) e Adaptação de Domínio (*Domain Adaptation*) na tarefa de segmentação de pele. Tais conceitos foram adaptados para 3D e amplamente explorados posteriormente na tarefa de complementação semântica de cenas.

Tendo em vista que as soluções anteriores de complementação semântica de cenas não exploravam completamente a informação presente na parte RGB da imagem de entrada, no Capítulo 5 nós endereçamos o problema da esparsidade ao projetar os dados RGB para 3D, por meio de uma maneira completamente nova de explorar a informação RGB presente na imagem RGB-D. A solução consiste em extrair as bordas da imagem RGB e projetá-las para 3D. Por ser uma informação binária, o volume 3D correspondente às bordas projetadas pode ser submetido ao algoritmo F-TSDF, para eliminar o problema de esparsidade. A rede 3D pode então fazer a fusão do volume proveniente do mapa de profundidade com o volume proveniente das cores. A Utilização das bordas da imagem RGB, permite detectar objetos que não seriam detectáveis nas soluções anteriores baseadas exclusivamente no mapa de profundidade, a exemplo de quadros planos ou TVs de tela plana colocados em paredes. Esta solução recebeu o nome de EdgeNet e atingiu resultados promissores na época de seu lançamento.

Posteriormente, nós avançamos no uso dos dados RGB por meio das probabilidades *a priori* extraídas a partir de uma de rede segmentação semântica 2D. No capítulo 6, nós apresentamos SPAwN, uma solução multi-multimodal, leve e direta que que explora a segmentação semântica 2D de uma forma completamente nova. Nos trabalhos anteriores que exploravam a segmentação semântica 2D, devido ao alto consumo de memória, o

procedimento comum era projetar não a saída final da rede, o que consumiria muitos recursos, mas sim, projetar as *features* internas da rede 2D. Outras soluções que usavam a saída da rede, eram obrigadas a aplicar algum tipo de codificação no volume projetado para reduzir seu tamanho. Ambas as soluções tinham como efeito colateral a redução do potencial semântico advindo da rede 2D. A nossa solução consiste em alimentar uma rede de segmentação 2D bimodal com dois modos da imagem RGB-D de entrada: RGB e as normais de superfície. Após isso, nós submetemos a saída da rede 2D a uma função Softmax para obter as probabilidades *a priori* que são projetadas para um volume 3D de baixa resolução. O terceiro modo de entrada, o mapa de profundidade, é projetado para um volume 3D de alta resolução que é codificado com F-TSDF.

Os dados *a priori* foram usados como guia semântico enquanto o volume proveniente do mapa de profundidade fornece a base estrutural da cena. SPAwN também introduziu o uso de *data augmentation* aplicado diretamente aos volumes 3D.

Nós completamos nossas contribuições relativas à melhoria da qualidade das inferências no Capítulo 7, combinado a técnica de Adaptação de Domínio explorada nos estágios iniciais da nossa pesquisa com a nossa rede 3D multi-modal atingindo resultados impressionantes.

Em relação à cobertura da cena, que hoje é restrita ao campo de visão limitado de sensores RGB-D convencionais, como o Microsoft Kinect, no Capítulo 8, nós propusemos uma abordagem para estendê-la para 360° usando imagens RGB panorâmicas e mapas de profundidade obtidos a partir de sofisticados sensores de 360° ou a partir de câmeras panorâmicas de baixo custo, montadas em uma configuração estereoscópica. Os resultados promissores obtidos com a abordagem proposta foram usados com sucesso em um sistema de reprodução de áudio espacial imersivo.

Nossos estudos preliminares em 2D foram publicados na *34th SIBGRAPI Conference on Graphics, Patterns and Images* (**SIBGRAPI 2021**). Nossas contribuições no domínio 3D foram publicadas em 3 conferências de visão computacional de alto nível: *International Conference on Pattern Recognition* (**ICPR 2020**); *IEEE/CVF Winter Conference on Applications of Computer Vision* (**WACV 2022**); e *Conference on Computer Vision Theory and Applications* (**VISAPP 2020**); O sistema de reprodução de áudio espacial imersivo usando a nossa solução 3D em 360° foi publicado na revista Virtual Reality Journal (**VIRE**).

Palavras-chave: Visão Computacional, Compreensão de Cenas 3D, Complementação Semântica de Cenas, Redes Neurais Convolucionais

Contents

List of Acronyms and Abbreviations

CCD Charged Coupled Device

CNN Convolutional Neural Network

CRFs Conditional Random Fields

CVSSP Centre for Vision, Speech and Signal Processing, University of Surrey

DA Domain Adaptation

F-TSDF Flipped Truncated Signed Distance Function

FCN Fully Convolutional Network

FOV Field of View

GPU Graphics Processing Unit

HOG Histograms of Oriented Gradients

IoU Interception over Union

IR Infra Red

RF Receptive Field

RGB-D Sensor which capture RGB image and depth maps

RNNs Recurrent Neural Networks

SSC Semantic Scene Completion

SVD Singular Value Decomposition

TL Transfer Learning

TSDF Truncated Signed Distance Function

List of Symbols

$P(\mathrm{Y}\vert\mathrm{X})$	conditional probability distribution of Y given X
$\mathcal{D}$	domain
O	asymptotic complexity (big-O notation)
P	probability distribution
$\mathbb{R}$	set of Real numbers
$\mathcal{T}$	task
$\mathcal{X}$	feature space
$\mathcal{Y}$	label space

Chapter 1

Introduction

Human visual perception is the ability to interpret and infer information from the environment using the reflected light that enters the eyes through the cornea and reaches the retina [6]. Using our stereoscopic vision system, we can naturally perform tasks such as scene classification (am I in a church, hospital, or school?), depth estimation (which object is closest to me? can I reach it?) and object identification, detection and localization (is this object near me a pen or a pencil?). All of those are examples of innate tasks for humans. In Computer Vision, however, reasoning about scenes in 3D is still an open field of study. Despite the great advances we have seen in the last few decades, there is still a lot of room for improvement. Issues to be addressed include low accuracy and field of view limited to the restricted angle of coverage of the sensor. With this research project, we intend to contribute to enhancing the current computational results on scene understanding, both in accuracy and coverage.

This Chapter contains a brief presentation of our field of study and a statement of the problem we intend to face. It also includes our objectives and the contributions we have achieved, as well as the expected contributions. To conclude the Chapter, an outline of the entire document is presented.

1.1 Scene Understanding: a Brief Presentation

The ability of reasoning about 3D scenes is considered to be one of the fundamental problems in Computer Vision [102]. Despite some remarkable progress that has been achieved in the past few decades, general-purpose computational scene understanding is still considered to be a very challenging problem [144].

The first works on scene understanding date back to the 70s. By that time, researchers were already using intrinsic image properties including range, orientation, reflectance and incident illumination of the surface element visible at each point in the 2D image [8] and the relationship between objects [122]. Given the computational power available, the tasks were very simple and the results were very poor.

After the year 2000, the increase of computational power made possible for data-driven methods like Histograms of Oriented Gradients (HOG) [32], Bags of Visual Words [29], eigenvectors-eigenvalue based algorithms [79] and Cascaded Classification Models [87] to be developed, leading to some improvement. However, in the past few years, two factors were decisive to the achievement of the current state-of-the-art results in computational scene understanding: The large-scale production of inexpensive depth sensors, such as Microsoft Kinect, and the boom of the Convolutional Neural Network (CNN).

The low-cost depth sensors led to great advances in indoor 3D scene understanding, especially because of the public RGB-D datasets that have been created and widely used for many 3D tasks. Among them are the prediction of unobserved voxels without semantic labelling [43], segmentation of visible surface [139, 118, 116, 54], object detection [137] and single object completion [109].

On the other hand, the increasing popularity of the CNN, especially after 2012, came together with enormous advances in general image understanding. Large datasets like ImageNet [33] began to be used to train deep convolutional models obtaining good results in image classification [77]. As occurred with image classification, convolutional networks also started to be successful in segmentation tasks, especially after the introduction of the Fully Convolutional Network (FCN)s [134]. Alongside the use of deep CNN for image classification and segmentation, a technique to leverage knowledge gathered from large datasets to other image domains became very popular: Transfer Learning [27].

In 2016, the joint use of real scene images gathered from depth sensors, 3D FCN, and transfer learning made possible the introduction of a new task in Scene Understanding that comprises semantic segmentation and scene completion in the field of view of the sensor with an end-to-end model, which the authors named Semantic Scene Completion (SSC) [148]. Given a partial 3D scene model acquired from a single RGB-D image, the goal of semantic scene completion is to assign a label to each voxel of the field-of-view of the sensor that indicates which class of object it belongs to, including visible, occluded, and inner voxels, as illustrated on the right part of Figure 1.1. Obtaining detailed labels of the whole 3D space of a set of scenes large enough to train a data-intensive model like a FCN is expensive. To tackle this problem, the authors first trained their model using a large synthetic dataset, then, using transfer learning, fine-tuned the network to make inferences using a small real dataset. Since that seminal paper, this new task became a very active line of research and the state-of-the-art has been pushed further by a sequence of works [167, 51, 96]. All of those works have confirmed that the use of transfer learning from synthetic datasets is useful for improving the accuracy of the models.

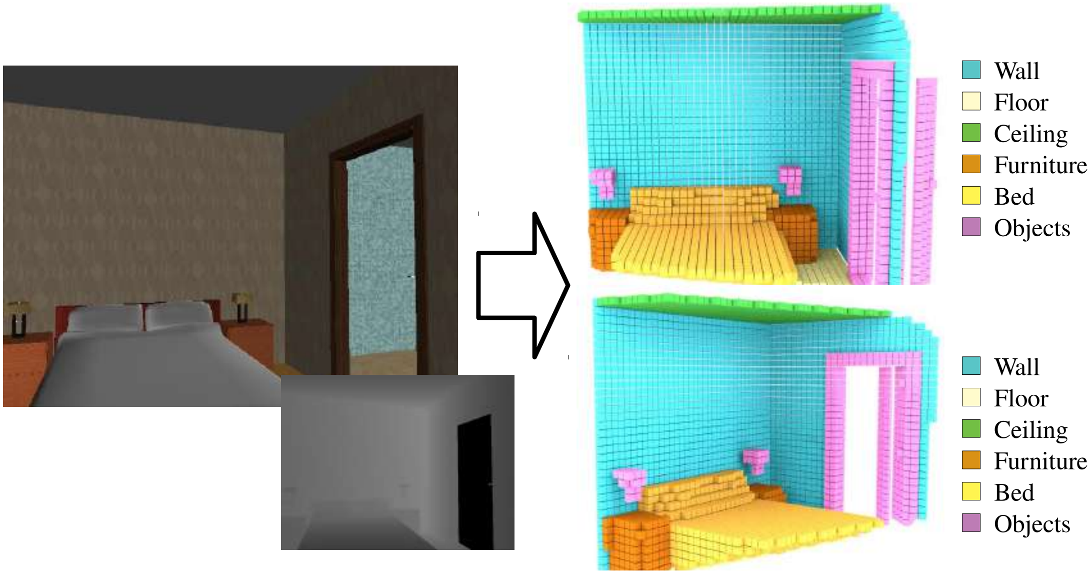


Figure 1.1: Semantic scene completion overview. Given an RGB-D image, the goal is to infer a complete 3D occupancy grid with associated semantic labels.

Given that the main goal of this research is to advance towards a complete understanding of indoor scenes, we focus our work on Semantic Scene Completion because it is the most complete task related to scene reasoning, as it aims to infer semantic labels to all the FOV including surface and inner voxels in the visible and occluded spaces. Knowing the complete 3D geometry of a scene and the semantic labels of each 3D voxel has many practical applications, namely robotics and autonomous navigation in indoor environments, surveillance, assistive computing, augmented reality, immersive spatial audio reproduction, and others.

1.2 Problem Statement

CNN-based deep learning models have reached human accuracy in several Computer Vision tasks, as in Large Scale Image Classification [125]. However, this is not the case for SSC. Despite the advances observed in the past few years, there is still room for improvement. Although some algorithms may be used in some applications as they are, current state-of-the-art model results are still far from ideal. This can be observed in many qualitative results presented in current state-of-the-art works on SSC [148, 166, 168]. Given the input images related to those results, an adult person can easily identify the errors of the networks and point out the correct classes. Figure 1.2, adapted from a cutting-edge work on SSC [96], highlights some of these flaws. We identify four main deficiencies in current approaches:

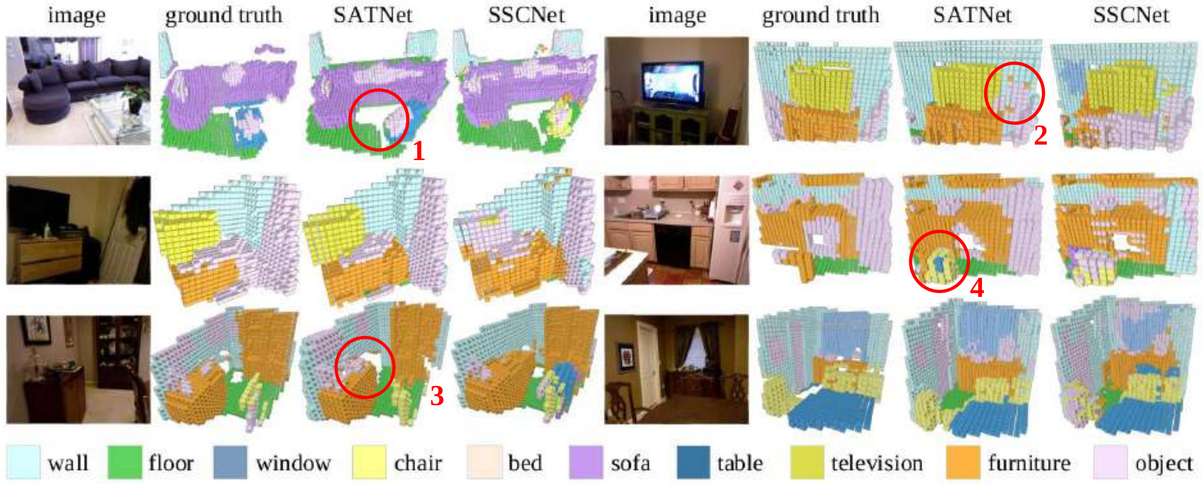


Figure 1.2: Some examples of SSC flaws in current state-of-the-art work. Numbers 1 and 3 highlight completion flaws in occluded regions of the scenes and 2 and 4 point out voxel classification errors. Adapted from qualitative results of SATNet model [96].

- the RGB part and other modes of the RGB-D images are not completely explored by current solutions;
- some techniques widely used in 2D deep CNN training like One Cycle Learning [141] and data augmentation [77], have been neglected in most SSC works;
- none of the identified previous works have explored the use of available unlabelled data to improve the generalization power of the models by domain adaptation and semi-supervised training [85];
- current solutions are limited to the restricted FOV of depth sensors like Kinect, for example.

Regarding the use of the information present in RGB data, we classify approaches into three main groups, based on the type of input of the semantic completion CNN:

1. **Depth maps only**: solutions in this category completely neglect the RGB information present in the RGB-D data. The seminal work of Song *et al.* which is known as SSCNet [148] is one example. To deal with data sparsity after projecting depth maps from 2D to 3D, the authors used a variation of Truncated Signed Distance Function (TSDF) which they called Flipped TSDF (F-TSDF). Other examples are Zhang *et al.* [167] and Guo and Tong [51]. All solutions in this category are end-to-end approaches, in other words, the network is trained as a whole, with no need for extra training stages for specific parts.
2. **Depth maps plus RGB**: Guedes *et al.* [49] reported preliminary results obtained by adding color to an SSCNet-like architecture. In addition to the F-TSDF en-

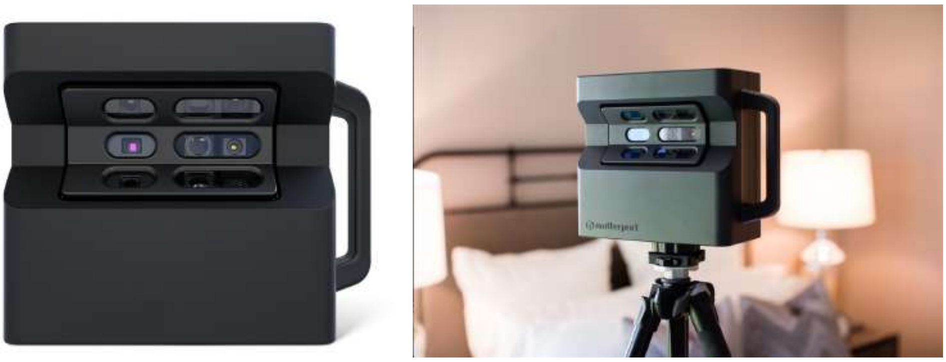


Figure 1.3: Matterport 360° Camera. This tripod-mounted device contains three structured light sensors pointing slightly up, straight ahead, and slightly down. To cover the whole scene, the camera rotates around its vertical axis while capturing multiple high-quality RGB-D images. Source: `www.matterport.com`. ©Matterport, Inc. Reproduced with permission.

coded depth volume, they used three extra projected volumes, corresponding to the channels of the RGB image, with no encoding, resulting in 3 sparse volumetric representations of the partially observed surfaces. The authors reported no significant improvement using the color information in this sparse manner.

3. **Depth maps plus 2D segmentation**: models in this category use a two-step training protocol, in which a 2D segmentation CNN is first trained and then used to generate input to a 3D semantic scene completion CNN. Examples of this category are Garbade *et al.* [45] and Liu *et al.* [96]. Current models differ in the way the generated 2D information is fed into the 3D CNN, but all of them suffer from the same sparsity problem faced by Guedes *et al.* In addition to that, using 2D segmentation maps on 3D SSC increases the complexity of the training phase, requiring training and evaluating the 2D segmentation network prior to the 3D CNN training.

Regarding the limited scene coverage, to the best of our knowledge, all works on SSC are limited to the FOV of the depth sensor. However, there are few recent works on scene understanding that uses 360-degree panoramic images generated by more advanced sensors like the Matterport as input, which focuses on objects' surfaces. Matterport, shown in Figure 1.3, combines multiple structured light sensors and allows 3D datasets that comprise high-quality panoramic RGB images and its corresponding depth maps of indoor scenes [2, 20] for a whole room. The datasets generated with these new sensors allowed the development of several scene understanding works [21, 95, 115], however, these works focus only on the visible surfaces, rather than on the complete understanding of the scene which should include occluded regions and inner parts of the objects.

A key aspect that makes it difficult for the development of effective 360° models is the absence of a large dataset with complete ground truth for SSC that comprises the whole scene. Existing 360° datasets are neither large nor generic enough to train very deep models or their ground truth annotations are not complete for SSC. As mentioned before, a synthetic dataset could be an affordable alternative to this limitation.

1.3 Objectives

The general objective of this research is to propose, implement and evaluate new tools and models that could push SSC solutions towards a complete understating of the whole indoor scene, including enhancing the coverage and quality of the inferences. The specific objectives of this research project are:

1. to assess the benefits of domain adaptation and semi-supervised training techniques in the context of 2D image segmentation aiming to further explore unlabeled data in 3D SSC;
2. to apply current trends on 2D deep CNN training protocols to 3D SSC networks;
3. to propose and evaluate a new SSC model that uses the RGB information present in RGB-D images and overcomes the sparsity problem when projecting features from 2D to 3D;
4. to propose and evaluate a multi-modal deep neural network to explore multiple modes of the RGB-D image and enhance 3D SSC scores;
5. to propose and evaluate the benefits of the use of unlabeled data in 3D SSC through semi-supervised learning.
6. to propose and evaluate a solution to perform 360° SSC using existing limited FOV datasets for training.

1.4 Contributions

This section presents the contributions of our work. We organize our contributions into five main groups according to the publication status of each proposed novelty.

1.4.1 Domain Adaptation and pseudo-labels applied to image segmentation

These contributions are detailed in Chapter 4 and in the paper **Domain Adaptation for Holistic Skin Detection** [36] which was published in the proceedings of the *34th SIBGRAPI Conference on Graphics, Patterns and Images* **(SIBGRAPI 2021)** and comprises:

- the proposal of a new Domain Adaptation strategy that combines Pseudo-Labeling and Transfer Learning for cross-domain training;
- a comparison between holistic and local approaches on in-domain and cross-domain experiments applied to skin segmentation with an extensive set of experiments;
- a comparison of CNN-based approaches with state-of-the-art pixel-based ones; and
- an experimental assessment of the generalization power of different human skin datasets (domains).

1.4.2 Better use of RGB information present in RGB-D images, and better training pipeline for 3D SSC

The contributions listed below can be found in Chapter 5 and in the paper **EdgeNet: Semantic Scene Completion from RGB-D images** [34] which was published in the proceedings of the *International Conference on Pattern Recognition* **(ICPR 2020)**:

- EdgeNet, a new end-to-end CNN architecture that fuses depth and RGB edge information to achieve state-of-the-art performance in semantic scene completion with a much simpler approach than previous works;
- a new 3D volumetric edge representation using flipped signed-distance functions which improves performance and unifies data aggregation for semantic scene completion from RGBD;
- a more efficient end-to-end training pipeline for semantic scene completion with relation to previous approaches.

1.4.3 Multiple modes of RGB-D input and data augmentation for 3D SSC

These contributions are shown in Chapter 6 and in the paper **Data Augmented 3D Semantic Scene Completion With 2D Segmentation Priors** [35] which was published in the proceedings of Proceedings of the *IEEE/CVF Winter Conference on Applications of Computer Vision* **(WACV 2022)**:

- *SPAwN*, a novel lightweight multimodal 3D SSC CNN architecture that uses 2D prior probabilities from a 2D segmentation network, that achieves state-of-the-art results on both real and synthetic data;
- *BN-DDR*, a memory-saving batch-normalized dimensional decomposition residual building block for 3D CNNs that preserves previous approaches' regularization characteristics while consuming much less memory during training;
- a novel strategy to apply a data augmentation technique for 3D semantic scene completion based on three 3D data transformations that operate on batches directly in GPU tensors.

1.4.4 Exploring Unlabelled Data for improving generalization of 3D SSC networks

This yet unpublished contribution is explained in detail in Chapter 7 and consists of *S3P*, a novel 2D-prior-based semi-supervised training approach to the SSC task that explores unlabeled data from the target dataset in a pseudo-label inspired approach and dramatically reduces overfitting when training with a small amount of labeled data. Combining *SPAwN* and *S3P* surpasses all known previous works on SSC on both real and synthetic data with or without pretraining.

1.4.5 360° Semantic Scene Completion

These contributions are shown in Chapter 8 and in the paper **Semantic Scene Completion from a Single 360° Image and Depth Map** [37] which was published in the proceedings of the Conference on Computer Vision Theory and Applications (VISAPP 2020):

- the extension of the SSC task to complete scene understanding using 360° imaging sensors or stereoscopic spherical cameras;
- a novel approach to perform SSC for 360° images taking advantage of existing standard RGB-D datasets for network training;
- a pre-processing method to enhance depth maps estimated from a stereo pair of low-cost 360° cameras.

The 360°semantic scene completion solution introduced in our VISAPP 2020 paper was applied in an audio-visual scene reproduction system that uses the reconstructed scene from our method to generate realistic 3D audio with scene ambiance. This application work was conducted in partnership with Surrey University and generated the paper **Immersive audio-visual scene reproduction using semantic scene reconstruction from 360 cameras**, that was published in the virtual reality journal [71].

1.5 Document Outline

This thesis is structured into 9 chapters. Chapter 1 consists in this introduction. In Chapter 2, we present some general knowledge related to neural networks, CNN, FCN, domain adaptation, stereo images, depth sensors, and 2D to 3D projection. Chapter 3 presents previous works related to the objectives of this research project.

Chapters 4 to 8 describe the contributions achieved so far, as summarised in the last section. Each one of these chapters contains its own sections on methodology, results, and conclusions.

We conclude and present future work on Chapter 9.

Chapter 2

Background and Related Concepts

In this Chapter, we present the background that is relevant to the remaining of this thesis. This knowledge should be useful to help those who are unfamiliar with 3D imaging and deep learning applied to Computer Vision. Some approaches presented here, are inspired by the way image and scene reasoning is biologically performed in nature, especially by humans.

2.1 Image Acquisition and Depth Estimation

In this section, we present some concepts regarding acquiring data and estimating depth for further processing by scene understanding algorithms.

2.1.1 Stereo Vision

Stereo vision is a Computer Vision area that addresses the problem of reconstructing the 3D distribution of the visible points from a pair of images of a scene. This approach relates to the stereo nature of human eyes and the way we subconsciously estimate depth in our daily activities. In the human vision system depicted in Figure 2.1, light from the surrounding environment that passes through the pupil is focused on the retina by the cornea and the lens combination, forming a projected two-dimensional (2D) image, similar to what happens in nowadays camera sensors. In the Retina, light is transformed into an electrical sign that reaches the primary visual cortex of the brain through the optic nerve. Our two eyes form a binocular imaging system, where each eye is responsible for generating a distinct image of the same scene, from two different locations. Our brain estimates the depth from the physical separation of the corresponding points on the retina of each eye (binocular disparity) [11].

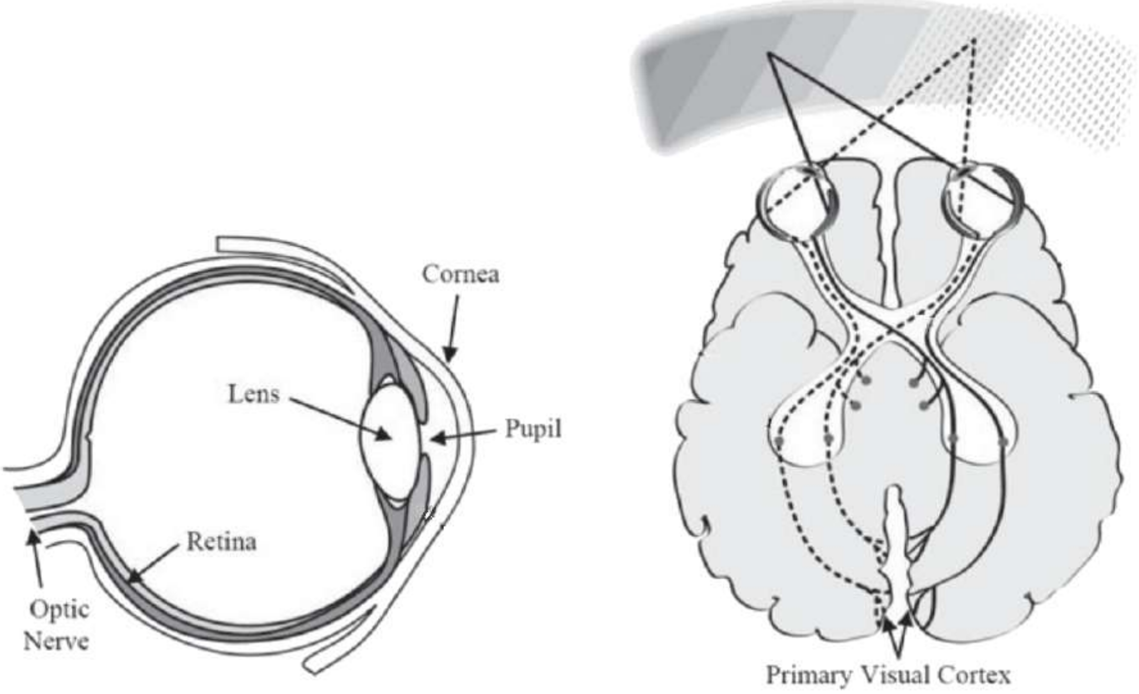
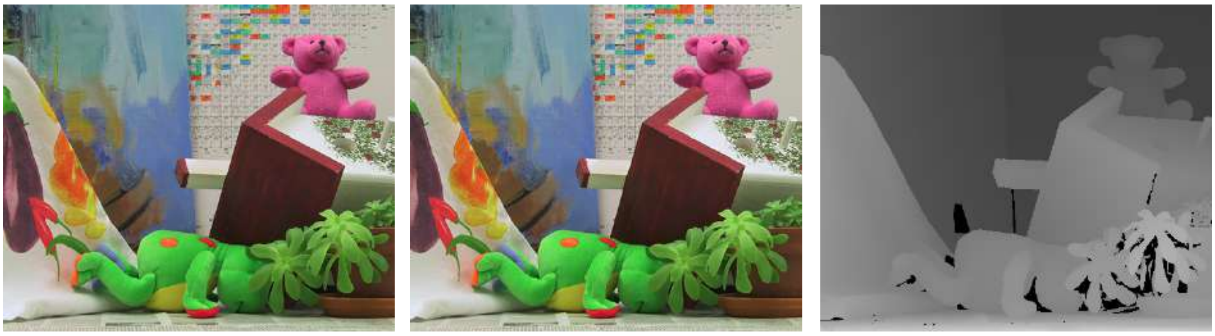


Figure 2.1: Details of the human vision system. Adapted from Sensification of computing: adding natural sensing and perception capabilities to machines [11]. Copyright held by the authors under CC BY 4.0 license.



(a) Left view (b) Right view (c) Depth map

Figure 2.2: A stereo image with corresponding disparity map from the Middlebury 2003 dataset [129]. Permission to use was granted by the authors.

The main component of a stereo computer vision system is a stereo camera which comprises two cameras placed next to each other, normally side-by-side horizontally. The two images captured simultaneously by these cameras are post-processed for the recovery of visual depth information [56] by building a disparity map. Figure 2.2 shows an example of a stereo image with the corresponding disparity map from the popular Middlebury dataset [128]. A disparity map is a monochromatic image in which the intensity of each pixel corresponds to the normalized disparity of the point between the two images (the concept of disparity is further defined in section 2.1.2, equations 2.3, 2.4 and 2.5). Finding a pixel-to-pixel correspondence between the two images is one of the main challenges in stereo computer vision. For instance, the black regions in the disparity map of Figure 2.2 are pixels to which it was not possible to establish the correspondence between the two images, due to occlusion or lack of distinguishing features.

2.1.2 Epipolar Geometry and Stereo Vision

In stereo vision, the set of geometric relations between the 3D world points and their projections onto the 2D image is known as Epipolar Geometry. In this subsection, we follow the notation and definitions of Hartley and Zisserman [58]. According to them, the Epipolar Geometry relations are derived based on the assumption that the cameras can be approximated by the pinhole camera model. Figure 2.3 depicts two cameras focusing at the real world 3-dimensional point $\mathbf{X} = (X, Y, Z)^\top$. For simplification, the two image planes are placed in front of the focal centers O_L and O_R. In real cameras, the image planes defined by the camera sensors are behind the focal centers.

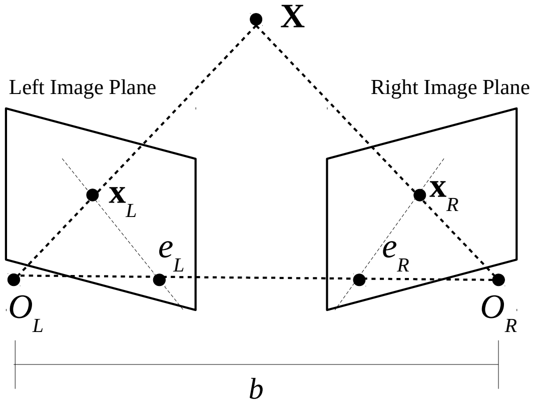


Figure 2.3: Epipolar Geometry based on Hartley and Zisserman [58].

A real world point $\mathbf{X}$ projected to a plane generates a 2-dimensional point $\mathbf{x} = (x, y)^\top$. Points $\mathbf{x}_L$ and $\mathbf{x}_R$ are the projections of point $\mathbf{X}$ onto the left and right image planes. The lines $\mathbf{x}_L - \mathbf{e}_L$ and $x_R - \mathbf{e}_R$ are called epipolar lines. The epipoles $\mathbf{e}_L$ and $\mathbf{e}_R$ are the projection of the other camera focal center onto the image planes. The distance between the two focal centers b is known as baseline and the distance between a focal center and its corresponding projection plane (camera sensor) is known as focal length. Although focal lengths are usually given in millimeters, it is common to use the focal length in pixels to simplify the calculations. The focal length in pixels in the x direction F can be

Figure 2.4: Zed stereo camera. Source: `www.stereolabs.com`. ©Stereolabs Inc. Reproduced with permission.

obtained from the focal length in millimeters F as in equation 2.1. Its counterpart in the y direction αf, where α is the aspect ratio, is given by equation 2.2.

$$f = \frac{F \times SensorWidth_{pixels}}{SensorWidth_{mm}} \tag{2.1}$$

$$\alpha f = \frac{F \times SensorHeight_{pixels}}{SensorHeight_{mm}} \tag{2.2}$$

For most modern Charged Coupled Device (CCD) cameras, $\alpha \sim 1$.

When the two cameras are aligned, as observed in the model in Figure 2.4, the epipolar geometry can be much simplified. In this situation, the world coordinates X, Y, and Z are given by the equations 2.3, 2.4 and 2.5, being (x_L, y_L) e (x_R, y_R) the corresponding coordinates in the left and right images, respectively. In these equations, b indicates the baseline and F is the focal length in pixels, and $(x_L - x_R)$ is known as the disparity of the point.

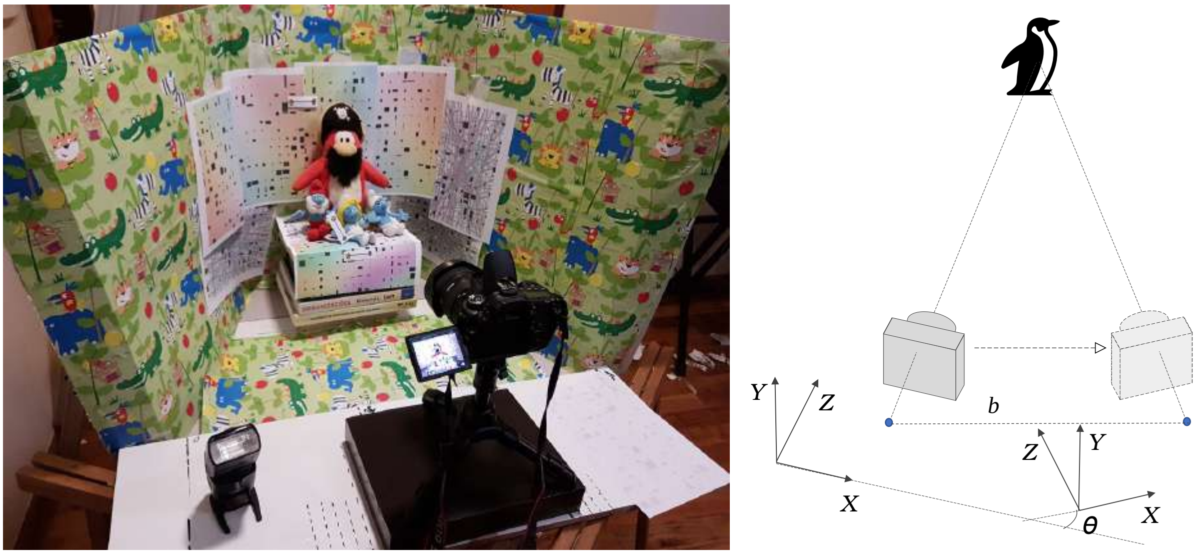
$$X = \frac{b(x_L + x_R)}{2(x_L - x_R)} \tag{2.3}$$

$$Y = \frac{b(y_L + y_R)}{2(x_L - x_R)} \tag{2.4}$$

$$Z = \frac{bf}{(x_L - x_R)} \tag{2.5}$$

When the two cameras are not aligned, the epipolar lines are not horizontal, so it is not trivial to match key points between the two cameras, since they may be at different heights in the images. Within a static scene, it is possible to emulate a non aligned stereo system with one single camera using the setup presented in Figure 2.5. In the example experiment, we used a textured background to make it easy to find distinguishing features and matching points between two images. In order to emulate the stereo capture, we took two pictures of the same scene with the camera slightly shifted and turned. In this case, we used a 20° angle between image planes (θ) and a 98mm baseline (b) as shown in Figure 2.5 .

In stereo systems where cameras are not perfectly aligned, before the stereo matching



(a) Real setup

(b) Schematic representation

Figure 2.5: Details of the setup used in the stereo emulation experiments with only one camera. At first, the left image is captured. Then, the camera is translated to the right and rotated. After that, the right image is captured.

procedure, it is necessary to rectify the image. A widely accepted method for rectification using Epipolar Geometry is described by Hartley and Zisserman [58]. following these steps:

1. camera calibration;
2. image rectification;
3. depth map generation through stereo matching.

We briefly describe these steps in the following subsections.

Camera calibration

The first step of the rectification process is to calibrate the camera, in order to determine its radial distortion and intrinsic parameters.

According to section 3.2 of the work of deCampos [16], the radial distortion may be modeled as equation (2.6), where rd is the displacement of an ideal image point and r_u is the radial distance from the center of distortion. The parameter k_1 indicates the type of distortion, where $k_1 < 0$ models barrelling distortion and $k_1 > 0$ models pin-cushion distortion.

$$r_d = r_u \left(\frac{1}{\sqrt{1 - 2k_1 r_u^2}} \right) \tag{2.6}$$

Equation (2.7) is the backward operation that corrects measured distorted image points back to their ideal position.

$$r_u = r_d \left(\frac{1}{\sqrt{1 - 2k_1 r_d^2}} \right) \tag{2.7}$$

The distortion parameter k may be estimated with Levemberg-Marquardt, through the minimization of the matching error between the undistorted image and the ideal image. The reference points for such a procedure may be obtained with a standard calibration object like the one on in Figure 2.6.

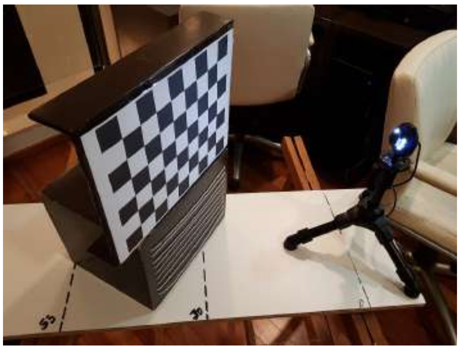


Figure 2.6: Calibration object used in the experiments.

After estimating the distortion parameter, the intrinsic parameters can be estimated. Although the procedure described in this section allows determining both intrinsic and extrinsic parameters, in this step, we are especially interested in the intrinsic parameter matrix K (that will be further explained), since the extrinsic parameters may vary depending on the actual position of the cameras in the next steps. For instance, in the setup described in Figure 2.5, the camera position varies during the acquisition process.

In order to allow the matrix operations necessary for the process, from now on, we will use homogeneous coordinates and represent the world coordinate vector $\mathbf{X}$ as $(X, Y, Z, 1)^\top$ and the image coordinate vector $\mathbf{x}$ as $(x, y, 1)^\top$. If we consider a camera as a device that can map a point from the 3D world to a 2D image, this mapping can be expressed by the equation 2.8, where P is a 3×4 matrix known as camera projection matrix and λ is a scale factor. The camera projection matrix P can be decomposed as shown in equation 2.9, where K is a 3×3 matrix which represents the intrinsic parameters of the camera, R is the 3×3 rotation matrix and $\mathbf{t}$ is the translation vector. R e $\mathbf{t}$ are the extrinsic parameters of the camera with relation to the world [58].

$$\lambda \mathbf{x} = \mathrm{PX} \tag{2.8}$$

$$\mathrm{P} = \mathrm{K}[\mathrm{R}|\mathbf{t}] \tag{2.9}$$

The intrinsic parameter matrix can be represented by equation 2.10, where F and αf represent the focal length in pixels, in x e y directions, respectively, and x_0 e y_0 represent the coordinate of the central point of the sensor, also in pixels[1] [58].

[1] This representation omits the image skew parameter, which is usually close to zero.

$$\mathrm{K} = \begin{bmatrix} f & 0 & x_0 \\ 0 & \alpha f & y_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.10}$$

In order to estimate the camera parameters in the projection matrix P, given that we have a set of real-world points ($\mathbf{X}_i$) for which the corresponding points in the image ($\mathbf{x}_i$) are known, from the equation 2.8, setting the scale factor $\lambda = 1$, it is possible to set up an equation system given by $\mathbf{x}_i = \mathrm{P}\mathbf{X}_i$.

To simplify the calibration process we can use a setup like the one presented in Figure 2.6, with a standard calibration object, to get a set of corresponding points ($\mathbf{X}_i \to \mathbf{x}_i$). The 48 internal corners of the chessboard can easily be detected in the captured image and are used as world reference points since chessboard dimensions are known. In such a setup, given that all the key points in the calibration object are coplanar, it is possible to assume that all points are on the world plane $Z = 0$. With such an assumption, the equation 2.8 can be simplified to equation 2.11.

$$(x_i \; y_i \; 1)^\top = \mathrm{P}_{3\times 3}(X_i \; Y_i \; 1)^\top \tag{2.11}$$

According to section 3.2 of the work of deCampos [16], initial values for the elements of P can be found by a linear transformation and these values can then be refined by the non-linear minimization in equation 2.12, where P′ represents the candidate values for P and $d(\mathbf{x}_i, \mathrm{P}'\mathbf{X}_i)$ is the Euclidean distance between the observation and the estimation.

$$\mathrm{P} = \arg\min_{\mathrm{P}'} \sum_i d(\mathbf{x}_i, \mathrm{P}'\mathbf{X}_i)^2 \tag{2.12}$$

Once P is determined, the rotation and intrinsic parameter matrices R and K can be found by applying QR-decomposition to P_L^{-1}, which is the inverse of the leftmost 3×3 block of the projection matrix P as in equation 2.13, where $\mathrm{R} = \mathcal{Q}^{-1}$ and $\mathrm{K} = \lambda'\mathcal{R}^{-1}$.

$$\lambda'\mathrm{R}^{-1}\mathrm{K}^{-1} = \mathcal{QR} \leftarrow \mathrm{P}_L^{-1} \tag{2.13}$$

The open-source software OpenCV [66] provides utility functions that performs the calibration processes described in this subsection.

Image Rectification

Image rectification is the process of projecting images onto a common image plane used to simplify the problem of finding matching points between images. In rectified images, all epipolar lines are parallel to the horizontal axis and corresponding points have identical vertical coordinates. This is done by computing two projective transformations H and H′ that are applied to the images from the first and second cameras respectively, generating two images that satisfy the rectified images properties. This means that the epipoles on both images should be mapped to infinity.

According to Hartley and Zisserman [58] (Chapter 9), the epipolar geometry is the intrinsic projective geometry between two views, so it is independent of scene structure and only depends on the cameras' internal parameters and relative pose. In a stereo camera setup, if a world point $\mathbf{X}$ is projected as $\mathbf{x}$ in the first camera and as $\mathbf{x}'$ in the second camera, there is a 3×3 matrix F, known as the Fundamental Matrix that satisfies this relation:

$$\mathbf{x}'^{\top}\mathrm{F}\mathbf{x} = 0. \tag{2.14}$$

Given a set of corresponding points in both views, it is possible to estimate the Fundamental Matrix F through minimization of the disparity or least-square difference of corresponding points on the horizontal axis of the rectified image pair.

The usual method to find corresponding points consists of detecting distinguishing features in the two images to further match those key points. Surf [9] and SIFT [99] are usually employed for feature detection, but ORB [123] is an open-source alternative that presents good results. Once the key points were found, it is necessary to match them in the two images. In Figure 2.7, are presented the matching points for our example scene. The distinguishing features (or key points) were detected with the ORB algorithm and the matches between them were detected using an exhaustive search, known as Brute Force algorithm in OpenCV's implementation [66].

Those matched points were then used to estimate the fundamental matrix using the least median of squares (LMedS algorithm). Recall that the fundamental matrix F is a transformation that maps a point in one camera to its correspoding point in the other camera, however, we want to find the transformations H and H′ that maps both epipoles

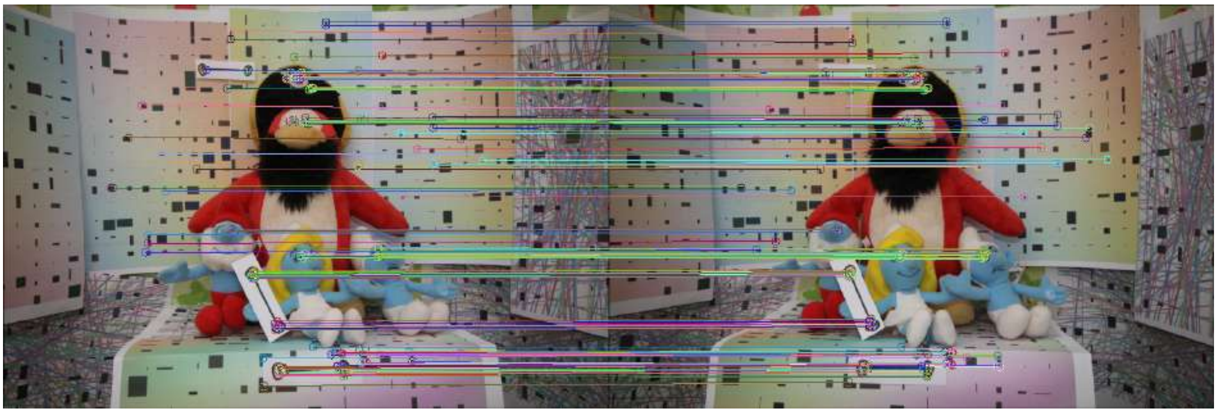


Figure 2.7: ORB key points and matches between both images.

to infinity.

The essential matrix is the specialization of the fundamental matrix to the case of normalized image coordinates. Given that we can obtain K through calibration, from F it is possible to derive the essential matrix E, through equation 2.15.

$$\mathrm{E} = \mathrm{K}^{\top}\mathrm{FK}. \tag{2.15}$$

Now, we can derive the rotation matrix R and the translation vector $\mathbf{t}$ from E through Singular Value Decomposition (SVD). The SVD decomposition of an $m \times n$ complex matrix M is a linear algebra procedure to perform a factorization of the form $\mathrm{M} = \mathrm{U}\Sigma\mathrm{V}^*$, where U is an $m \times m$ complex unitary matrix, Σ is an $m \times n$ rectangular diagonal matrix with non-negative real numbers on the diagonal, and V is an $n \times n$ complex unitary matrix. In our specific application, M is real, so, U and V can also be guaranteed to be real orthogonal matrices. In this case of real matrix applications, the SVD is denoted as $\mathrm{U}\Sigma\mathrm{V}^{\top}$.

Following equations 9.13 and 9.14 from Hartley and Zisserman [58], suppose that

$$SVD(\mathrm{E}) = \mathrm{U}\Sigma\mathrm{V}^{\top} \tag{2.16}$$

and

$$\mathrm{W} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.17}$$

Then

$$\mathrm{R} = \mathrm{UWV}^{\top} \tag{2.18}$$

and $\mathbf{t}$ is the last column of U:

$$\mathbf{t} = [u_{13}; u_{23}; u_{33}]^{\top}. \tag{2.19}$$

Thus, from R, $\mathbf{t}$ and K we can obtain the epipoles $\mathbf{e}$ and $\mathbf{e}'$:

$$\mathbf{e} = \mathrm{KR}^{\top} \tag{2.20}$$

and

$$\mathbf{e}' = \mathrm{K}\mathbf{t}. \tag{2.21}$$

The epipolar lines obtained through this process for our example are shown in Figure 2.8.

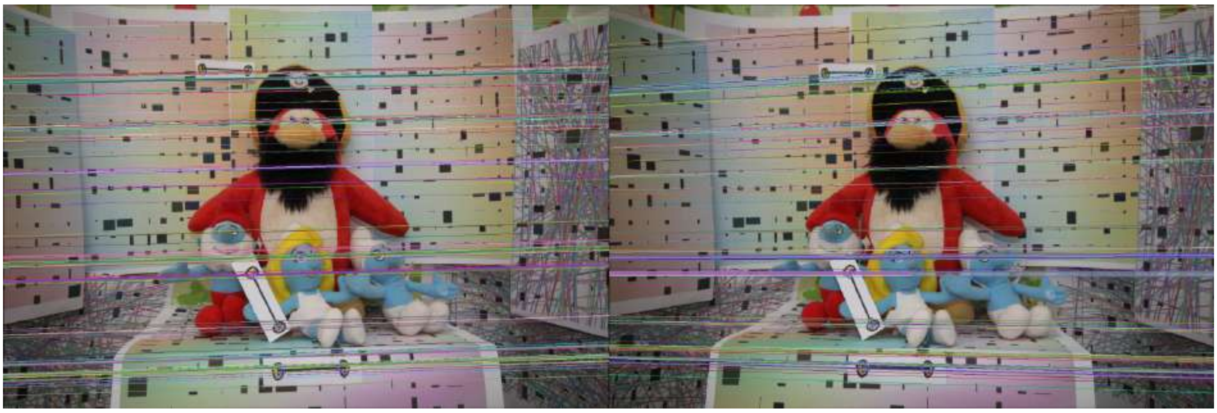


Figure 2.8: Epipolar lines obtained by computing the fundamental matrix F using the matches shown in Figure 2.7.

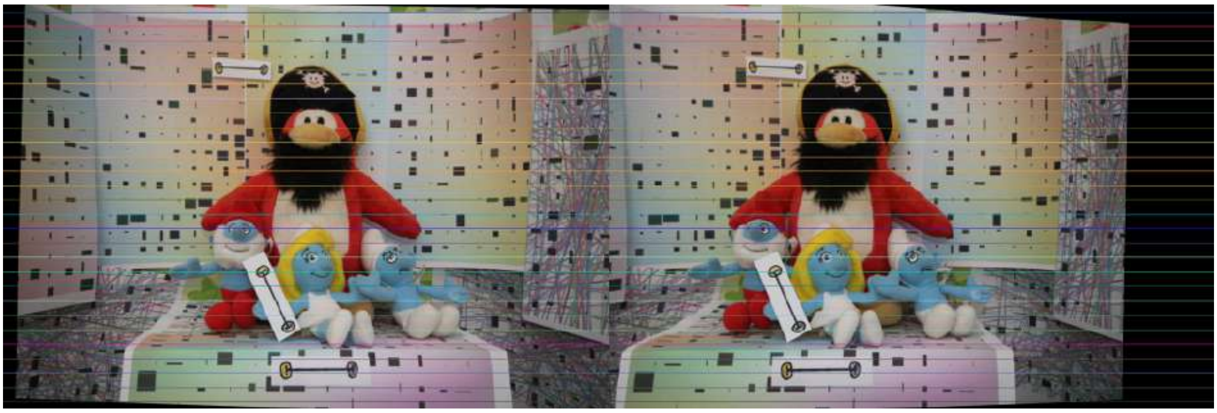


Figure 2.9: Epipolar Geometry rectification.

In order to map the epipoles to infinity, consider the following transformation G, which maps the epipole $[f, 0, 1]^\top$ to the point at infinity $(f, 0, 0)^\top$:

$$\mathrm{G} = \begin{bmatrix} 1 & 0 & x_0 \\ 0 & 1 & 0 \\ -1/f & 0 & 1 \end{bmatrix} \tag{2.22}$$

The transformation $\mathrm{H}' = \mathrm{GR}\mathbf{t}$ maps the epipole e to infinity, where $\mathbf{t}$ is a translation that maps the point x_0 to the origin, R is a rotation about the origin that maps the epipole e' to a point $[f; 0; 1]^\top$ on the x-axis and G is given by equation 2.19. Given the set of matches between both images $\mathbf{x}_i \longleftrightarrow \mathbf{x}'_i$ (see Figure 2.7), the transformation H is the one that minimizes the sum of square distances between the corresponding points of the images as in the equation below:

$$\arg\min_{\mathrm{H}} \sum_i ||\mathrm{H}\mathbf{x}_i - \mathrm{H}'\mathbf{x}'_i||^2 \tag{2.23}$$

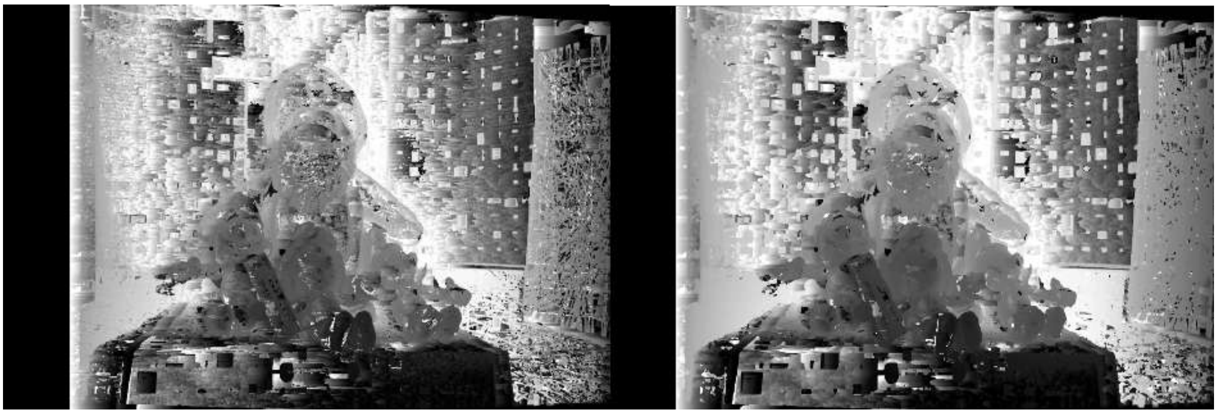


Figure 2.10: Final depth maps. Brighter areas means are far from the camera and darker areas are closer. We show both left and right maps for illustration purposes. Normally, only the left one is used.

Figure 2.9 shows the resulting rectification using H and H′. Note that the epipolar lines are parallel to the x-axis in the rectified images (epipoles mapped to infinity).

Depth map generation through stereo matching

After rectification, the depth map generation is much simplified. The disparity estimation can be done by finding matches at the same horizontal height in both images and depth can be obtained using equation 2.5.

Figure 2.10 presents the final result obtained with a naive sliding window algorithm. However, there are currently available a number of much more robust stereo-matching methods which may lead to better results, including Segment-Based Stereo Matching [75] and Color-Weighted Correlation [162].

2.1.3 360° Stereo Vision

In the previous subsection, we demonstrated the use of Epipolar Geometry to estimate depth using a pair of regular cameras with a limited field of view. To overcome the limited view, there are currently many low-cost consumer-level spherical cameras available, allowing high-resolution 360° RGB image capture. Those cameras can be combined to generate 360° images and corresponding depth maps through stereo matching. We present an example of this procedure in Chapter 8.

For example, in order to get high-resolution spherical images with accurate calibration and matching, Spheron developed a line-scan camera, Spheron VR[2], with a fish-eye lens to capture the full environment as an accurate high resolution and high dynamic range

[2] Spheron, `https://www.spheron.com/products.html`

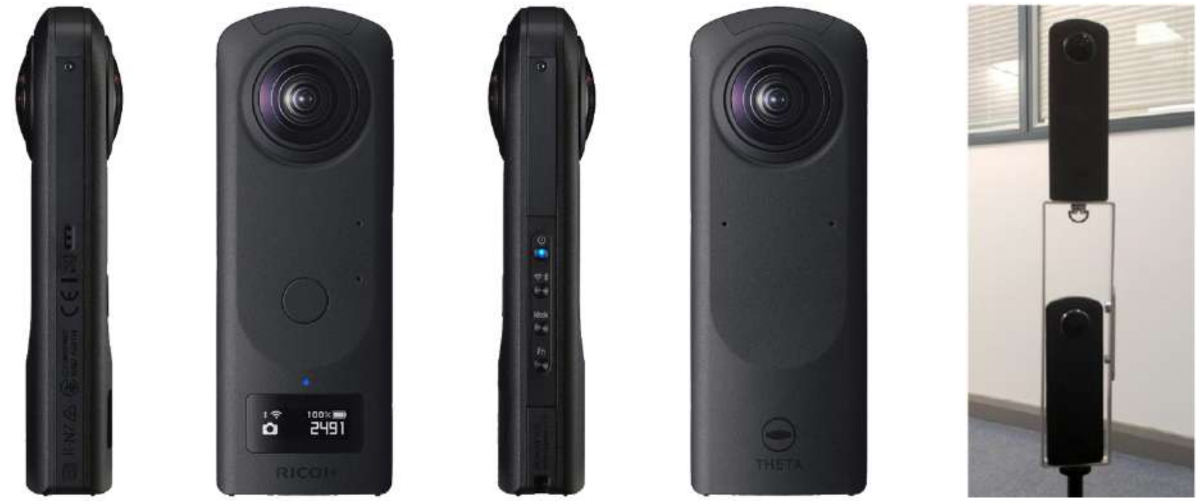


Figure 2.11: Ricoh Theta 360°and the stereo setup used in our experiments. ©Ricoh Company, Ltd. (the first 4 images). The last image was acquired by our collaborator Hansung Kim in the University of Surrey.

image. Li [89] has proposed a spherical image acquisition method using two video cameras with fish-eye lenses pointing in opposite directions. Various inexpensive off-the-shelf 360° cameras with two fish-eye lenses have recently become popular[3,4]. In order to estimate depth using those cameras, the scenes can be captured as a vertical stereo image pair and dense stereo matching with spherical stereo geometry [70]. Figure 2.11 shows the vertical 360° stereo setup used in our experiments presented in Chapter 8. In this setup, we used two 360° Ricoh Theta cameras vertically aligned. Each device has two fish-eye lenses placed back to back.

[3] Insta360, `https://www.insta360.com`

[4] Ricoh Theta, `https://theta360.com/en/`

2.2 Depth Sensors

Figure 2.12: RealSense® . ©Intel Corporation. Reproduced with permission.

During the past decade, 3D imaging technology experienced a boost with the production of inexpensive depth sensors like Kinect®[5], RealSense®[6] (Figure 2.12) and Structure Sensor®[7]. This caused a leap in the development of many successful semantic Computer Vision tasks that use both RGB and depth, especially in data-driven 2.5D and 3D vision [148, 51, 167]. Those structured light sensors usually have a a RGB camera, an infrared (IR) light source that projects invisible light over the subject and an active Infra Red (IR) sensor to capture depth, without the need of distinguishing feature points as in regular stereo cameras (See Figure 2.12). Many datasets have been created using this kind of sensor, such as NYU Depth V2 [140], one of the most widely used.

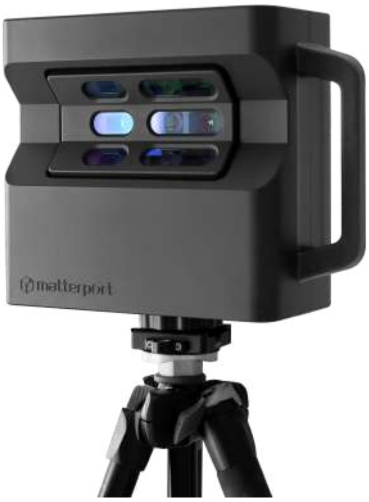


Figure 2.13: Matterport 360° Camera. ©Matterport Inc. Reproduced with permission.

The 3D sensing field has recently had another boost after the public availability of high-quality datasets like Stanford 2D-3D-Semantics Dataset [2] and Matterport3D [20], which comprises point cloud ground truth of whole buildings, 360° RGB panoramas and corresponding depth maps and other features. These datasets are acquired with the Matterport[8] camera, shown in Figure 2.13. The Matterport camera consists of three structured light sensors (color and depth) pointing slightly up, horizontally, and slightly down. For the scene capture, the camera is placed on a tripod. During the scanning process, it rotates and acquires high-quality RGB photos and depth data of the whole room. The resulting 360° RGB-D panoramas are software-generated from this data [20].

[5] https://developer.microsoft.com/en-us/windows/kinect/develop

[6] https://www.intelrealsense.com/depth-camera-d435/

[7] https://structure.io/

[8] https://matterport.com/

2.3 Image and Scene Reasoning

Performing simple vision-related activities, like identifying a given object in a scene, is considered an easy task for most humans. We have no difficulty identifying an object visible in our field of view, regardless of its size, orientation, and lighting conditions. However, it took fifty years, since the 1960s, for computer vision to achieve human comparable performance in 2D image classification tasks [131]. Regarding 3D scene understanding, computer vision is still very far behind human inference capabilities, as we will explore in Chapters 4 to 8. We believe that there is a lot of room for improvement in this area.

In this section, we present some background related to the evolution of Computer Vision, from the early logics-based approaches until current stage of development achieved with the use of deep Convolutional Neural Network (CNN) for image classification and image segmentation. Image classification is the task of defining a label for the whole image. For example, given a picture of a pet, the goal is to define if it shows a cat or a dog. Image segmentation can be thought of as image classification at pixel level. For example, given a picture that contains two pets, a cat and a dog, the goal is to identify which pixels belong to the dog and which pixels belong to the cat.

2.3.1 Early approaches

In 1956, John McCarthy from Dartmouth College, Marvin Minsky from Harvard University, Nathaniel Rochester from IBM Corporation, and Claude Shannon from Bell Laboratories, promoted the Dartmouth Summer Research Project on Artificial Intelligence. This event was a workshop to discuss computers, natural language processing, neural networks, theory of computation, abstraction, and creativity. This workshop is considered to be the founding event of artificial intelligence as a field of study. From the Dartmouth workshop, two main lines of research gained momentum: the paradigm of symbolic information processing pushed by McCarthy, Allen Newell, and Herbert Simon, which became known as symbolic AI; and the earlier method of brain modeling, a neurophysiological approach related to cybernetics and neural nets defended by Shannon, Minsky, and Rochester [76].

One of the first attempts to computationally solve a vision problem was made by the MIT AI Lab founded in 1959 by Marvin Minsky and Jonh McCarty. In 1960, the Lab received a grant to build a robot to play Ping-Pong. Since performing vision tasks is a simple task for humans, by that time, there was an intuition that it would also be simple to write a computer vision program for that. Following this misleading intuition, the lab assigned the responsibility to write this program to an undergraduate student, Gerald Sussman, as a summer project[9]. Building such a vision program turned out to be a complex task and early AI pioneers noticed that computer vision was a very hard problem [131].

Early computer vision approaches relied on matching a template of a given object with the pixels of the image. For example, consider a bird species classification task from bird pictures based on template matching. Depending on the pose of the bird in the template, pictures from many different species may present a good match if the poses are similar, while a picture of the desired species may have a poor match if the poses are different. The first good results in computer vision were observed when approaches started to focus on features, instead of focusing on pixels. However, developing feature detectors for lots of objects in the world is very labor intensive. Partial occlusions were another difficulty in the early computer vision days. Besides that, in the 1960s, digital computers were inefficient and extremely expensive. Given that digital computers are much more efficient at logical operations than humans, Artificial Intelligence pioneers found much more easy to create programs to prove mathematical theorems using logics than to perform vision and other real word problems like medical diagnosis tasks [131].

Given this scenario, by the 1980s, the way the brain actually works had become irrelevant for most AI researchers. The most common approach for AI was to write programs that were only functionally equivalent to the way the brain functions, which allowed the scientists to ignore the complex biologically details of the brain. This approach was known as functionalism. Only a small group of researchers still believed that a biologically inspired approach known as "neural networks", "connectionism", and "parallel distributed processing" would solve problems not treatable by the logics-based approach [131].

However, as the problems became more and more complex, the logic-based programs created to solve them started to become untreatable. One of the first examples of this difficulty related to computer vision was the Blocks World project, lead in 1965, by Larry Roberts, a Ph.D. student working at Marvin Minsky's MIT AI Lab [119]. The idea of his Ph.D. thesis was to develop a complete scene understanding system for a simplified world of textureless polyhedral shapes. This project further evolved to a robot capable of stacking blocks lead by another Ph.D. student at MIT AI Lab, Patrick Henry Winston, in 1972 [158]. Although it was a very simplified version of the world, the computer vision program behind it was extremely complex. The project was discontinued after Terry Winograd, the student that wrote the code, left the lab [131].

By the early 1980s, these difficulties and the resulting lack of advances in AI lead to a dramatic reduction in research funding and interest, giving origin to the period that

[9] Se details of this project in the MIT Lab document "The Summer Vision Project" by Seymour A. Papert [112]

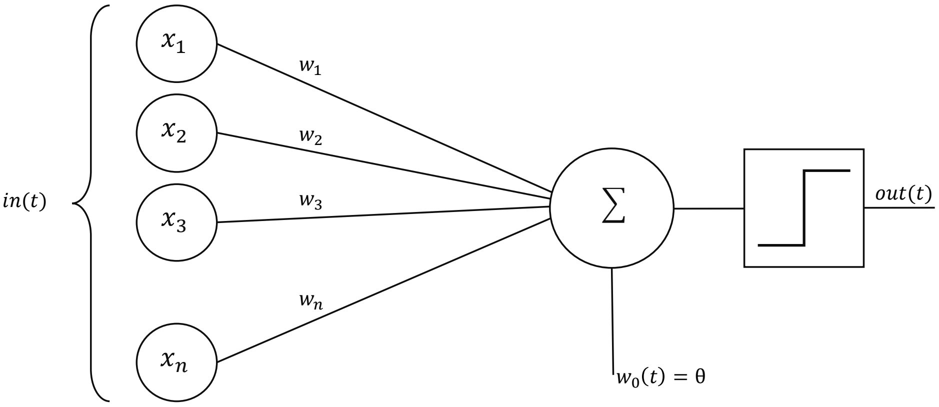


Figure 2.14: Rosenblatt's perceptron [121], a single neuron model that is the predecessor of nowadays deep neural networks. The perceptron is a pattern classification model with binary output. The pattern is fed into the model through the inputs x_1 to x_n. Each input x_i is multiplied by the corresponding weight w_1 to w_n and summed. The result is compared to θ and the step function generates the binary output.

is known as the AI winter [7]. This low-interest period lasted until 2012 when it was observed a dramatic increase in funding and investment was mostly pushed by the field of computer vision and ImageNet database project project [33] and ILSCVR challenge [125].

2.3.2 Rise and fall of biologically inspired vision models

The base for the nowadays successful deep neural networks, the perceptron [121], was conceived by Frank Rosenblatt in 1958. He was one of the first to attempt to mimic brain function with a simple model for pattern recognition that emulates our visual system. A perceptron is a classification model that aims to infer if an input pattern belongs to a given category or not, which learns from examples and functionally resembles a neuron. Consider the perceptron structure presented on Figure 2.14. Each input of the perceptron is associated with a visual stimulus like a photocell signal or the intensity of a pixel in an image. Those inputs (x_i) are multiplied by the weights (w_i), summed and then compared to the threshold θ. The final step function provides a binary output of value “1” when the sum is greater than θ or “0” otherwise, indicating if a set of inputs corresponds to the desired class, or not.

The set of weights are defined by a incremental learning algorithm. During the training phase, for each example in the training set, the output is calculated and compared to the correct answer. If the output is correct no changes are made to the weights. However, if the output is wrong, the weights are slightly changed. When the weights stop changing, it means that the algorithm has found the solution. One interesting fact about the perceptron is that Rosenblatt has proven that, if there is a set of weights that correctly discriminates the two classes and if there are enough training samples, his learning algorithm will find them [121]. The model was first implemented in an IBM 704 from Cornell Aeronautical Laboratory, a huge machine that cost \$2 million at the time (approximately \$20 million nowadays) [131].

The main limitation of this model is that it only can classify classes that are linearly separable. So, there are many real-world situations in which the perceptron model does not apply. In perceptrons: An Introduction to Computational Geometry, first published in 1969, thus, eleven years after the introduction of the perceptron model, Minsky and Papert provided a rigorous analysis of the model's capabilities, demonstrating its strenghts and describing the conditions under which the perceptron is not capable of carrying out the desired results [105]. The book also introduced the idea of combining multiple layers of perceptrons as a way to bypass the limitations, but there was no means at the time to train such a model. After Minsky and Papert's book, many researchers considered the model limitations definitive and the scientific interest in the perceptron model dramatically reduced [131].

2.3.3 Neural networks rebirth

In the second half of the 1980's, some bright new researchers brought some interest back to the field, by exploring the idea of the multilayer perceptron proposed by Minsky and Papert in 1969. In "Learning representations by back-propagating errors (1986)" [124], David Rumelhart and Geoffrey Hinton introduced the backpropagation learning algorithm, a generalization of the training strategy of the perceptron, based on the gradient of the error (or loss) function with respect to the weights of the network. The backpropagation differentiation algorithm for calculating gradients combined with some sort of stochastic gradient descent optimization strategy is the main approach for training nowadays deep neural networks. Figure 2.15 illustrates a typical setup of the multilayer perceptron for handwritten digit recognition from the popular MINST dataset [84].

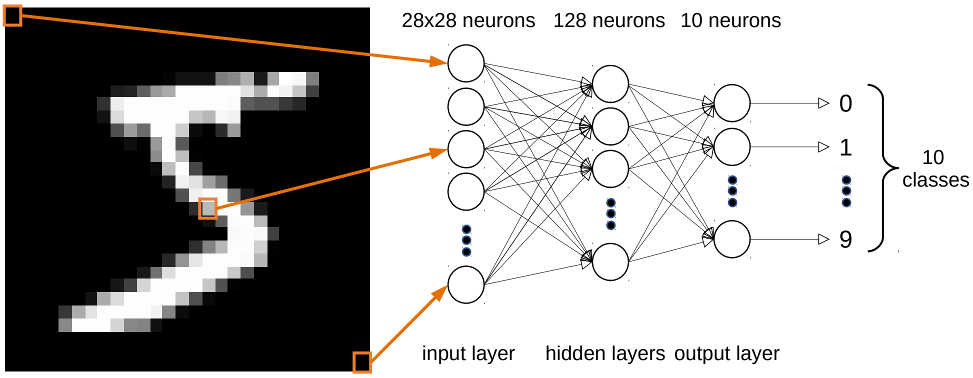


Figure 2.15: Typical fully connected neural network for handwritten digit recognition.

In a multilayer perceptron architecture (also known as fully-connected), each pixel of the input image is connected to a neuron of the input layer, and all neurons of a given layer are connected to all neurons of the next layer. The network also includes one or more hidden layers and one output layer. The output layer contains as many neurons as the number of output classes, and the activation function is usually a softmax function, thus, each one of the output neurons provides, approximately, the probability of the input image belonging to its corresponding class. The softmax function (σ) is defined in equation 2.24, where Z is the input vector and k is the number of classes.

$$\sigma^K : \mathbb{R}^K \to (0,1), \sigma(Z)_i = \frac{e^{z_i}}{\sum_{j=1}^{k} e_j^z} \tag{2.24}$$

However, this fully connected architecture has some limitations. Being (x, y) the dimension of the input images, its asymptotic memory complexity is $O(x \times y)$, thus, it is hard to scale to large input sizes. Most importantly, as the pixels are sequentially treated, this architecture does not take advantage of their 2D spacial organisation and local features are not considered. In "Generalization and network design strategies" (1989) the young Yann LeCunn provides the bases for overcoming this problem. He applied backpropagation to several models and analyzed how different architectures affect the generalization ability of the network [82, 83]. This work also explored the use of prior knowledge about the problem of image classification to constrain the network design and introduced the primitives for convolutional neural networks. His multilayer constrained network was organized in a hierarchical structure with shift-invariant feature detectors to introduce some *a priori* knowledge of the task into the network architecture.

The original architecture is presented in Figure 2.16. The base of the idea was the use of convolutions over the image, with shared weighs. Hidden layers are organized in planes, called feature maps. Each unit in a feature map takes input on a small square region of the previous layer, called filter. All units of a feature map share the same set of weights and the number of weights is determined by the size of the filter. In the original architecture, the convolution implied sub-sampling of the previous layer, so the output feature maps had half the size of the input. The last layer of the network is a regular fully-connected layer with softmax that acts as

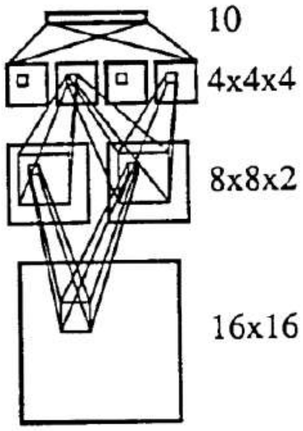


Figure 2.16: Original image of the convolutional network proposed by LeCun in 1989 [82]. ©Elsevier/North Holland, 1989. Reproduced with permission.

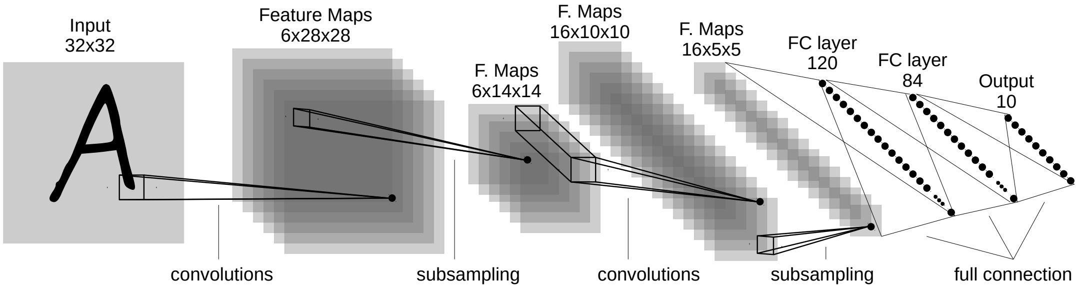


Figure 2.17: LeNet-5 (1998) architecture for check recognition int the bank industry [84], was one of the first commercial convolutional networks. Adapted from the original paper.

a classifier. This architecture has much fewer learnable parameters than its fully-connected counterpart and is capable of capturing local features in multiple resolutions.

Some interesting results of biologically inspired vision models were observed in the 1990s. One example is SexNet [48], a neural network to classify the sex of a person based on the image of the person's face. SexNet had 2 versions: a single layer and a multilayer architecture which achieved 88% and 92% percent accuracy, respectively. In 1998, a CNN called LeNet-5, which architecture is presented in Figure 2.17, was already being used in commercial check recognition systems for the bank industry in the United States [84]. Despite the observed advances, the level of interest in neural networks was much lower than observed in the early days of the field.

2.3.4 The boom of the Convolutional Networks

Since its introduction in 1989, CNNs have been continuously evolving, but a huge boost was observed from 2012 on. In 2012, AlexNet, one of the first deep convolutional networks, was trained to classify 1000 different classes, achieving very impressive results [77]. What enabled the success of AlexNet was its clever way to explore available hardware capabilities. As convolutions and matrix multiplications are operations that could be parallelized, the authors of AlexNet implemented a deep CNN that could run on a Graphics Processing Unit (GPU), a dedicated hardware for large-scale parallel numeric processing, which original purpose was graphical processing.

AlexNet used the off-the-shelf GTX 580 GPU, with 3 GB memory. AlexNet, which architecture is presented in Figure 2.18, consists of 5 convolutional layers followed by max-pooling for downsampling, has 60 million parameters, and was trained using two GTX 580 GPUs. Besides the increase in computer power, especially because of the use of GPUs, the other reason for the current success of the convolutional neural networks was the release of large general-purpose image datasets like Imagenet [33] for training.

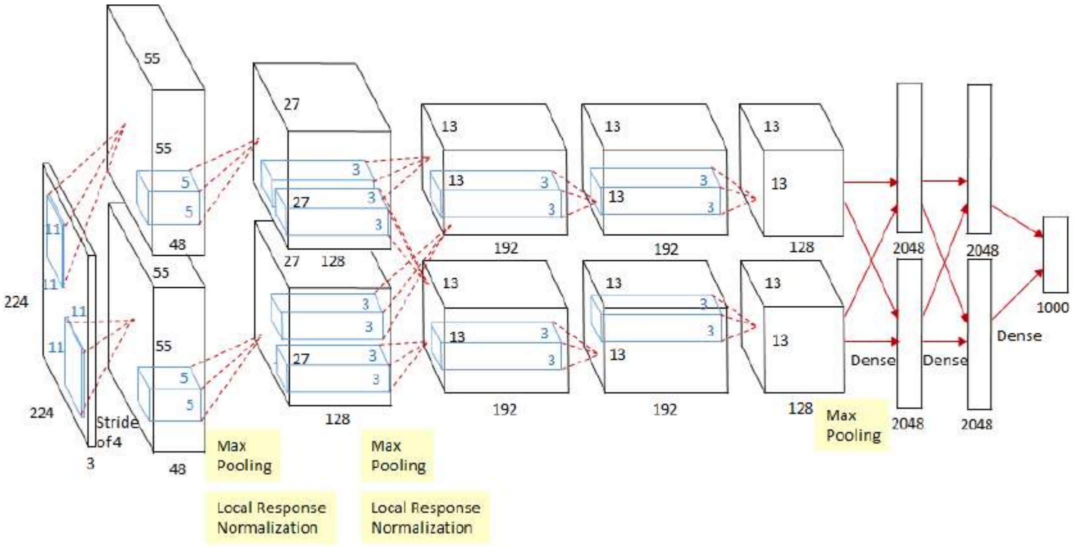


Figure 2.18: AlexNet architecture (2012) [77], designed to be trained in two GTX 590 GPUs with 3GB memory each, which achieved remarkable results in the ImageNet LSVRC-2010 context. Copyright held by the authors. Reproduced with permission.

Another important milestone in the history of the deep CNNs was the introduction of the residual block architecture [60]. As the computational power was increasing, the networks were becoming deeper and unexpectedly more difficult to train.

Researchers noticed that adding more layers to a suitably deep model leads to higher training errors. This phenomenon is known as the degradation problem of deep convolutional networks. Instead of directly connecting a series of convolutional layers, the solution proposed by the authors was the introduction of skip connections between two layers, allowing the following layer to learn a residual mapping, as shown in Figure 2.19. Suppose that the desired mapping from a block of convolutional layers is $\mathcal{H}(x)$ and the actual mapping fit by the block is $\mathcal{F}(x) - x$. So, the desired mapping is $\mathcal{F}(x) + x$. The hypothesis is that it should be easier to learn the desired mapping from the residual $\mathcal{F}(x) + x$ than from $\mathcal{F}(x)$. To the extreme, if the identity mapping x is optimal, it should be easier to push the residual to zero than to fit an identity mapping through a stack of nonlinear layers.

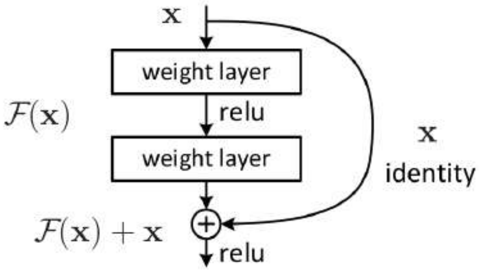

x
weight layer
$\mathcal{F}(\mathbf{x})$
relu
weight layer
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Figure 2.19: Residual block architecture. Source [60]. Adapted from the original paper.

The residual Block Architecture allowed the development of very deep networks like Inception-ResNet [151] which achieved a new state-of-the-art result in the ILSVRC 2016.

2.3.5 FCNs for image segmentation

One of the first solutions to perform image segmentation using deep CNNs was the patch-based approach. It consists of using a patch around each pixel of the image to classify it [24]. This approach has two main problems:

- is highly computationally intensive, as it requires performing a number of inferences that is proportional to the number of pixels to segment the image;
- and it does not consider the context while performing segmentation.

In opposition to patch-based classification, the FCN-based approach for image segmentation introduced by [134] considers the context of the whole image. A Fully Convolutional Network (FCN) is a CNN in which all trainable layers are convolutional. Therefore, they can be quite deep but have a relatively small number of parameters, due to the lack of fully connected layers. Another advantage of FCNs is that, in principle, the dimensionality of the output is variable and it depends on the dimensionality of the input data.

FCNs gave rise to the idea of encoder-decoder architectures, which have upsampling methods, such as unpooling layers and transpose convolutions (so-called deconvolution layers). These methods can perform segmentation taking the whole image as an input signal and generate full image segmentation results in one forward step of the network, without requiring breaking the image into patches. Because of that, FCNs are faster than the patch-based approaches and overcame the state-of-the-art on PASCAL VOC [40], NYUDv2 [139, 140], and SIFT Flow [94] datasets, by using Inductive Transfer Learning[10] from ImageNet.

Following the success of FCNs, Ronneberger *et al.* [120] proposed the U-Net architecture, which consists of an encoder-decoder structure initially used in biomedical 2D image segmentation. In U-Net, the encoder path is a typical CNN, where each down-sampling step doubles the number of feature channels.

What makes this architecture unique is the decoder path, where each up-sampling step concatenates the output of the previous step with the output of the down-sampling with the same image dimensions. This strategy enables precise localization with a simple network that is applied in one shot, rather than using a sliding window. With this strategy, the U-Net is able to model contextual information, which increases its robustness and allows it to generate segmentation results with a much finer level of detail. This strategy is simpler and faster than more sophisticated methods, such as those that combine CNNs with Conditional Random Fields (CRFs) [4]. CRFs are a class of statistical modeling methods often applied in pattern recognition which can take context into account, modeling predictions as a graphical model. The method of Zheng *et al.* [171], which models

[10] See discussion on section 2.5.2.

CRFs as Recurrent Neural Networks (RNNs) (CRF-as-RNN), enables a single end-to-end training/inference process for segmentation, generate sharper edges in the segmentation results in comparison to the standard U-Net. However, CRF-as-RNN is much slower than U-Net due to the nature of RNNs.

The original U-Net architecture does not take advantage of pre-trained classification networks. In order to deal with small amounts of labeled data, the authors made extensive use of Data Augmentation, which has been proven efficient in a many cases [161, 153, 114, 159].

Several variations of U-Net have been proposed since then. For example, the V-Net [104] is also an encoder-decoder network adapted to the segmentation of 3D biomedical images. Nowadays, one of the most used variations consists in replacing the encoder branch with a pre-trained classification network like Inception [152] or ResNet [60], combining the U-Net architecture with the original approach of Fully Convolutional Networks. Another common strategy is the use of short-range residual connections (recall Figure 2.19) in the convolutions blocks of the encoder and decoder branches of U-Net, as in Pandey *et al.* [111].

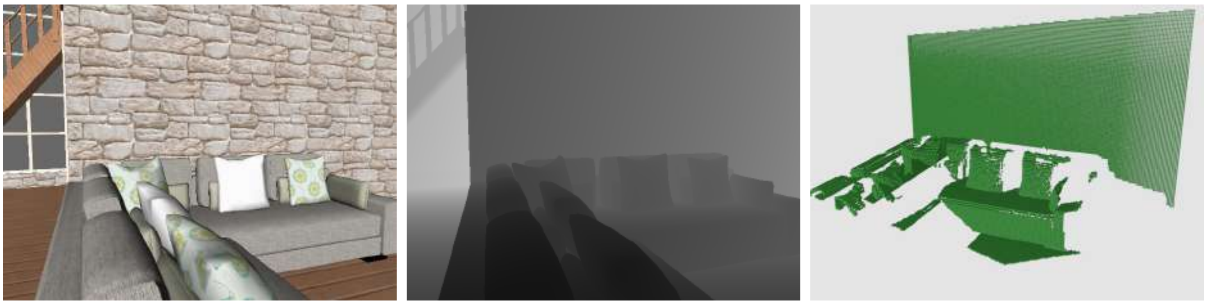
2.4 3D Representation: Voxel Volume Encoding

In Sections 2.3.4 and 2.3.5 we presented how convolution-based networks evolved and achieved impressive results in 2D tasks like image classification and semantic segmentation. In this thesis, our objective is related to 3D scene reasoning which depends on 3D networks. The main difference between 2D and 3D convolution networks is the shape of the convolutional kernel that is used. While in 2D we use plain rectangular kernels like 3×3 kernels, in 3D we use volumetric kernels (3×3×3, for example).

We also showed how one can obtain RGB-D images from stereo cameras or depth sensors in Sections 2.1.2 and 2.2. However, RGB-D images are not a complete 3D representation, since the color channels (RGB) and the depth map are represented as two 2D images of the same dimension in pixels. Fortunately, the process to obtain a pure 3D representation from depth maps is relatively simple. The other topic of this section is the volume encoding process, which is used to feed the 3D with a meaningful signal that makes the training of 3D networks easier.

2.4.1 Lifting from Depth Maps to Voxels

A voxel volume is a regular grid in three-dimensional space. Each position of the 3D grid represents a voxel. The voxel position is inferred from its position relative to other voxels. In a voxel volume that represents the visible part of a scene, each voxel stores



(a) RGB (b) Depth Map (c) Voxel Representation

Figure 2.20: From depth to voxels: lifting the voxel representation of the visible surface of a scene from the synthetic dataset SUNCG [148]. The voxel unit is 0.02m (2cm).

binary information indicating whether or not the voxel belongs to the visible surface. To correctly represent the space in a voxel representation, the size of the voxel should be specified. This is usually done by stating the linear size of the voxels' edges, which is called voxel unit (v_u). Figure 2.20 shows a voxel representation of a scene from the SUNCG dataset [148], using a 2cm voxel unit.

The processes of lifting the voxel representation starts by obtaining the point cloud corresponding to the depth map. A point cloud is a set of 3D points coordinates where each 3D point corresponds to a pixel in the depth map. At first, it is necessary to denormalize the value of depth encoded into the depth map. When working with depth maps generated from depth sensors, it is important to know the normalization method used by the the manufacturer to correctly denormalize the depth data[11]. Given that the denormalized depth of a pixel (x_i, y_i) of the depth map is d_i, and the focal length f and the origin (x_0, y_0) and the α factor of the sensor or camera are known from the matrix K (refer to equation 2.10), the world coordinate (X_i^c, Y_i^c, Z_i^c) relative to the sensor or camera position is given by equations (2.25), (2.26) and (2.27),

$$X_i^c = \frac{(x_i - x_0) \times d_i}{f} \tag{2.25}$$

$$Y_i^c = \frac{(y_i - y_0) \times d_i}{\alpha f} \tag{2.26}$$

$$Z_i^c = d_i \tag{2.27}$$

When the camera position is represented by the rotation matrix R and the translation vector $\mathbf{t}$ is provided, its is possible to obtain the actual world coordinates (X_i, Y_i, Z_i) relative to the scene from the camera coordinates previously obtained, with equation

[11] We provide a denormalization function for the Microsoft Kinect sensor in our CUDA-based preprocessing library, alongside other useful functions to voxel volume manipulation here: `https://gitlab.com/UnBVision/spawn`

(2.28).

$$\begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix} = \begin{bmatrix} X_i^c \\ Y_i^c \\ Z_i^c \end{bmatrix} \times \begin{bmatrix} \mathrm{R} \end{bmatrix} + \mathbf{t}^\top. \tag{2.28}$$

This equation is useful to maintain the resulting point cloud aligned to a reference point of view, independent of the camera movements. It is also useful to generate point clouds aligned according to the Manhattan Principle.

To obtain the final representation, consider that the world coordinate (X_0, Y_0, Z_0) is the origin of the voxel representation. Recalling that v_u is the voxel unit, once the world coordinate of the point is known, the corresponding coordinate in the voxel representation (X_i^v, Y_i^v, Z_i^v) is given by equation (2.29).

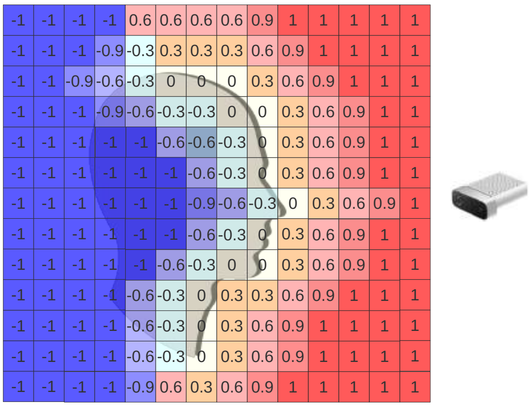
$$\begin{bmatrix} X_i^v \\ Y_i^v \\ Z_i^v \end{bmatrix} = \begin{bmatrix} \mathit{floor}((X_i - X_0)/v_u) \\ \mathit{floor}((Y_i - Y_0)/v_u) \\ \mathit{floor}((Z_i - Z_0)/v_u) \end{bmatrix} \tag{2.29}$$

The voxel grid is usually initialized with zeros and when at least one point of the point cloud falls into a voxel, that voxel is set to 1. When all the points of the point cloud are considered, the voxel representation is complete. With the procedure described in this section, the voxel representation that is obtained is equivalent to that of Figure 2.20c.

2.4.2 TSDF 3D Volume encoding

The voxel grid obtained from a depth map is a very sparse representation of the scene, as can be seen in Figure Figure 2.20c. Only a few voxels contain useful information while all other voxels value zero. Besides that, there is no differentiation between visible and occluded regions. Although this representation can be used as input to a 3D CNN, the convergence of the network may take too long to obtain or even may not be obtained. A much more useful encoding should fill a larger portion of the grid with some information that gently guides the convergence of the parameters of the model in the direction of the visible surface. This representation should be invariant to the view-point projection, and provide a not sparse signal for the network.

The Truncated Signed Distance Function (TSDF) [30] encoding procedure is a very common procedure to tackle the sparsity problem of volumetric representations. In TSDF encoding, every voxel stores the distance to its closest surface, clipped in a given range, normalized to -1 to 1, as illustrated in Figure 2.21. Voxels on the surface are set to zero. Voxels in the occluded part of the scene are set to a negative value ranging from -1 to 0 and voxels in the visible empty space are set to a positive value ranging from 0 to 1.



-1	-1	-1	-1	0.6	0.6	0.6	0.6	0.9	1	1	1	1	1
-1	-1	-1	-0.9	-0.3	0.3	0.3	0.3	0.6	0.9	1	1	1	1
-1	-1	-0.9	-0.6	-0.3	0	0	0	0.3	0.6	0.9	1	1	1
-1	-1	-1	-0.9	-0.6	-0.3	-0.3	0	0	0.3	0.6	0.9	1	1
-1	-1	-1	-1	-1	-0.6	-0.6	-0.3	0	0.3	0.6	0.9	1	1
-1	-1	-1	-1	-1	-1	-0.6	-0.3	0	0.3	0.6	0.9	1	1
-1	-1	-1	-1	-1	-1	-0.9	-0.6	-0.3	0	0.3	0.6	0.9	1
-1	-1	-1	-1	-1	-1	-0.6	-0.3	0	0.3	0.6	0.9	1	1
-1	-1	-1	-1	-1	-0.6	-0.3	0	0	0.3	0.6	0.9	1	1
-1	-1	-1	-1	-0.6	-0.3	0	0.3	0.3	0.6	0.9	1	1	1
-1	-1	-1	-1	-0.6	-0.3	0	0.3	0.6	0.9	1	1	1	1
-1	-1	-1	-1	-0.6	-0.3	0	0.3	0.6	0.9	1	1	1	1
-1	-1	-1	-1	-0.9	0.6	0.3	0.6	0.9	1	1	1	1	1

Figure 2.21: Illustration of the Truncated Signed Distance Function (TSDF) encoding in 2D for better visualization (best seen in color).

Consider that c is the clipping value, e_i is the euclidean distance from the voxel v_i to the closest visible surface voxel and $signal_i$ is an occlusion indicator for the voxel v_i that values -1 for the occluded region and 1 for the visible region. The TSDF encoding value $TSDF_i$ for the voxel v_i is given by equation (2.30).

$$TSDF_i = \frac{min(c, e_i)}{c} \times signal_i \tag{2.30}$$

2.5 Domain Adaptation

As Deep Neural Networks require high amounts of labeled data to be trained, Transfer Learning (TL) and Semi-Supervised Learning methods can be employed to dramatically reduce the cost of acquiring training data. While semi-supervised learning exploits available unlabeled data in the same domain, transfer learning is a family of methods that deal with the change of task or change of domain. Domain Adaption (DA) has been categorized as a particular case of transfer learning [27]. We will discuss these methods in the next subsections.

2.5.1 Transfer Learning Base Concepts

To present the base concepts of TL and Domain Adaptation (DA) we will use the notation of [110].

A domain $\mathcal{D}$ is composed of a d-dimensional feature space $\mathcal{X} \subset \mathbb{R}^d$ with a marginal probability distribution $P(\mathrm{X})$. A task $\mathcal{T}$ is defined by a label space $\mathcal{Y}$ with conditional probability distribution $P(\mathrm{Y}|\mathrm{X})$.

In a conventional supervised machine learning problem, given a sample set $\mathrm{X} = \{\mathrm{x}_1, \cdots, \mathrm{x}_\mathrm{n}\} \in \mathcal{X}$ and the corresponding labels $\mathrm{Y} = \{\mathrm{y}_1, \cdots, \mathrm{y}_\mathrm{n}\} \in \mathcal{Y}$, $P(\mathrm{Y}|\mathrm{X})$ can be learned from feature-label pairs in the domain. Suppose we have a source domain $\mathcal{D}^s = \{\mathcal{X}^s, P(X^s)\}$ with a task $\mathcal{T}^s = \{\mathcal{Y}^s, P(Y^s|X^s)\}$ and a target domain $\mathcal{D}^t = \{\mathcal{X}^t, P(X^t)\}$ with a task $\mathcal{T}^t = \{\mathcal{Y}^t, P(\mathrm{Y}^t|\mathrm{X}^t)\}$. If the two domains correspond ($\mathcal{D}^s = \mathcal{D}^t$) and the two tasks are the same ($\mathcal{T}^s = \mathcal{T}^t$), we can use conventional supervised Machine Learning techniques. Otherwise, adaptation and/or transfer methods are required.

If the source and target domains are represented in the same feature space ($\mathcal{X}^s = \mathcal{X}^t$), but with different probability distributions ($P(\mathrm{X}^{\mathrm{s}}) \neq P(\mathrm{X}^{\mathrm{t}})$) due to domain shift or selection bias, the transfer learning problem is called homogeneous. If $\mathcal{X}^s \neq \mathcal{X}^t$, the problem is heterogeneous TL [27, 110]. In this thesis, we deal with homogeneous transfer learning as we use the same feature space representation for source and target datasets.

Domain Adaptation is the problem where tasks are the same, but data representations are different or their marginal distributions are different (homogeneous). Mathematically, $\mathcal{T}^s = \mathcal{T}^t$ and $\mathcal{Y}^s = \mathcal{Y}^t$, but $P(\mathrm{X}^s) \neq P(\mathrm{X}^t)$.

2.5.2 Inductive Transfer Learning

When the source and target domains are different ($\mathcal{D}^s \neq \mathcal{D}^\top$), models trained on $\mathcal{D}^s$ may not perform well while predicting on $\mathcal{D}^\top$ and if tasks are different ($\mathcal{T}^s \neq \mathcal{T}^\top$), models trained on $\mathcal{D}^s$ may not be directly applicable on $\mathcal{D}^\top$. Nevertheless, when $\mathcal{D}^s$ maintains some kind of relation to $\mathcal{D}^\top$ it is possible to use some information from $\{\mathcal{D}^s, \mathcal{T}^s\}$ to train a model and learn $P(\mathrm{Y}^\top|\mathrm{X}^\top)$ through a process that is called Transfer Learning (TL) [110].

According to Csurka [27], the Transfer Learning approach is called inductive if the target task is not exactly the same as the source task, but the tasks are in some aspects related to each other. For instance, consider an image classification task on ImageNet [125], which has 1000 classes, as the source task and a Cats vs. Dogs classification problem as a target task. If a model is trained on a dataset that is as broad as ImageNet, one can assume that most classification tasks performed on photographies downloaded from the web are subdomains of ImageNet which includes the Cats vs. Dogs problem (i.e. $\mathcal{D}^{\texttt{cats}\times\texttt{dogs}} \subset \mathcal{D}^{\texttt{ImageNet}}$), even though the tasks are different ($\mathcal{Y}^{\texttt{ImageNet}} = \mathbb{R}^{1000}$ and $\mathcal{Y}^{\texttt{cats}\times\texttt{dogs}} = \mathbb{R}^2$). This is the case of a technique to speed up convergence in Deep CNNs that became popularised as *Fine Tuning* for vision applications.

In deep artificial neural networks, fine-tuning is done by taking a pre-trained model, modifying its final layer so that its output dimensionality matches $\mathcal{Y}^\top$ and further training this model with labeled samples in $\mathcal{D}^\top$.

Further to fine-tuning, a wide range of techniques has been proposed for inductive TL [110], particularly using shallow methods, such as SVMs [5], where the source domain is used to regularize the learning process. The traditional fine-tuning process usually requires a relatively large amount of labeled data from the target domain with respect to shallow methods [27]. In spite of that, this technique is very popular with CNNs.

2.5.3 Unsupervised Domain Adaptation

Domain adaptation methods are called unsupervised (also known as transductive TL) when labeled data is available only on source domain samples. Several approaches have been proposed for unsupervised DA, most of them were designed for shallow learning methods [28]. The methods that exploit labeled samples from the source domain follow a similar assumption to that of Semi-Supervised Learning methods, with the difference that test samples come from a new domain. This is the case of Long et al. [98] and Farajidavar et al. [41]. Both methods start with a standard supervised learning method trained on the source domain in order to classify samples from the target domain. The classification results are taken as pseudo-(soft) labels and used to iteratively improve the learning method in a way that works better on the target domain.

When labeled samples are not available at all, it is possible to perform unsupervised transfer learning using methods that perform feature space transformation. Their goal is to align source and target domain samples to minimize the discrepancy between their probability density functions [12]. Style transfer techniques, such as that of Gatys et al. [46], achieve a similar effect, but their training process is much more complex.

2.5.4 Semi-supervised learning

Semi-supervised learning methods deal with the problem in which not all training samples have labels [173, 108]. Most of these methods use a density model in order to propagate labels from the labeled samples to unlabeled training samples. This step is usually combined with a standard supervised learning step in order to strengthen the classifiers [86, 26].

There are several semi-supervised learning approaches for deep neural networks. Methods include training networks using a combined loss of an auto-encoder and a classifier [117], discriminative restricted Boltzmann machines [80] and semi-supervised embeddings [157].

Pseudo-Labelling [85] is a simple yet effective approach, where the network is trained in a semi-supervised way, with labeled and unlabeled data in conjunction. During the training phase, for the unlabeled data, the class with the highest probability (pseudo-label) is taken as it was a true label. To account for the unbalance between true and pseudo labels, the loss function uses a balancing coefficient to adjust the weight of the unlabeled data on each mini-batch. As a result, pseudo-label works as an entropy regularization strategy.

These methods assume that training and test samples belong to the same domain, or at least that they are very similar ($\mathcal{D}^s \approx \mathcal{D}^\top$).

2.6 Chapter Summary

In this chapter, we presented some base knowledge required to better understand the rest of this thesis. Most of the chapter is focused on computer vision approaches that are inspired by human vision and learning systems.

We started with the basis of 3D image acquisition and depth estimation methods showing how similar they are to the human binocular vision system. We also showed how the field boosted after the introduction of more advanced structured light sensors. Then we showed how the image and scene reasoning research field evolved from a limited logics-based approach to the nowadays powerful and biologically inspired deep neural networks, which are influenced by how we learn to understand images. These concepts are important to understand the next chapters.

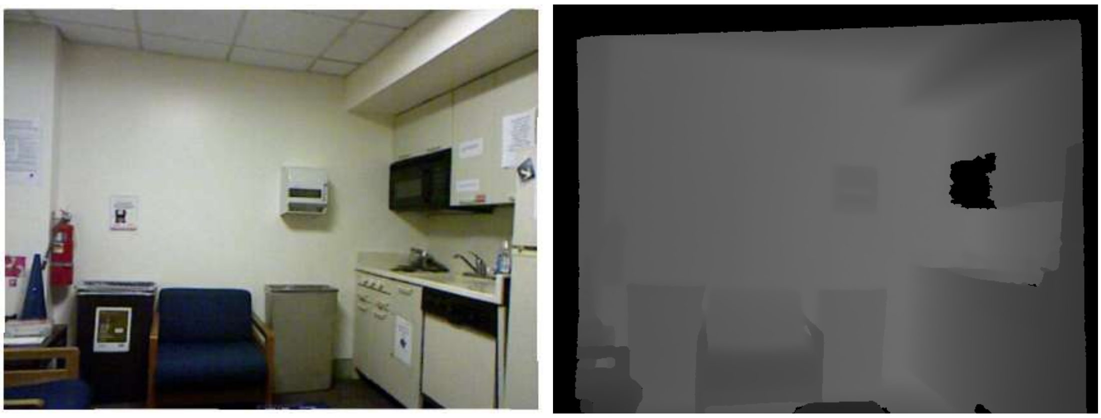
After that, we introduced the concept of transfer learning and domain adaptation, a set of approaches used to enhance the deep neural network learning process, making use of knowledge gathered from domains other than the task in question itself. This concept will be especially useful for Chapters 4 and 7.

Chapter 3

Related Works

In this Chapter, we present some previous works closely related to our research project. Here we focus on depth-enabled inputs for scene understanding like RGB-D sensors, binocular images, and scene scan systems from a single point of view. For example, Figure 3.1 illustrates the output of an off-the-shelf structured light sensor, used as input for many scene reasoning tasks.

Solutions that aim to perform scene reasoning from this kind of input should address two main problems: the completion of the occluded part of the scene, and the identification of the semantic labels of each object. In the next sections, we will explore the evolution of the scene reasoning field from early incomplete approaches to nowadays more complete tasks, that fully address both the problems mentioned before and extend the coverage to 360°from a single point of view.



(a) RGB

(b) Depth map

Figure 3.1: **A RGB-D output generated by a structured light sensor from a single point of view.** The output only covers the visible surface of the scene. For instance, the inner part of the furniture and the part of the trash cans behind the chair are not modeled. Scene from NYUD v2 dataset [140], further described in section 3.5. (Best viewed in color).

3.1 2D RGB-D Semantic Segmentation

The scene understanding study field observed a boost after the availability of low-cost structured light sensors like Microsoft Kinect. Those sensors made possible the task of **2D RGB-D Semantic Segmentation**, which is an extension to regular 2D RGB semantic segmentation with the addition of 3D depth maps obtained from the sensor. This task aims to obtain semantic labels for only the observed pixels without considering the full shape of the objects in the scene.

Early works [118, 54, 140] relied on handcrafted features fed into a classification model. Later, following the boom of CNNs, models started using the depth map as a fourth input channel [38], eventually encoded in HHA (horizontal disparity, height above ground, and angle with gravity) [55], before feeding into the neural network [97, 116, 155]. Lin *et al.* [92] introduced the RefineNet module to fuse skip-connections from different scales of the segmentation encoder, and [113] extended it to deal with multiple modes introducing the Multimodal Fusion (MMF) module. Later, Jiao *et al.* [64] proposed an encoder-decoder network that uses depth maps as ground truth to extract depth embeddings that are later fused to semantic features from the RGB image, thus eliminating the need for depth maps during inference and also achieving the current state-of-the-art in RGB-D semantic segmentation.

Multimodal networks achieved very high levels of accuracy for the semantic segmentation task from RGB-D images. However, the output was a simple 2D segmentation map and occlusions in the image were not addressed. This limitation leads to the development of more complete RGB-D 3D tasks as follows.

3.2 Partial 3D Scene Reasoning from RGB-D

After the establishment of the task of 2D Semantic Segmentation from RGB-D, a number of researchers started to work on tasks that aim to address the full 3D shape, rather than only focusing on the visible surface. However, these first attempts only partially addressed the scene reasoning problem, focusing only on individual objects' geometry or on the completion of the occluded part of the scene, leaving semantics aside. Here we present some of these early tasks and works on 3D domain.

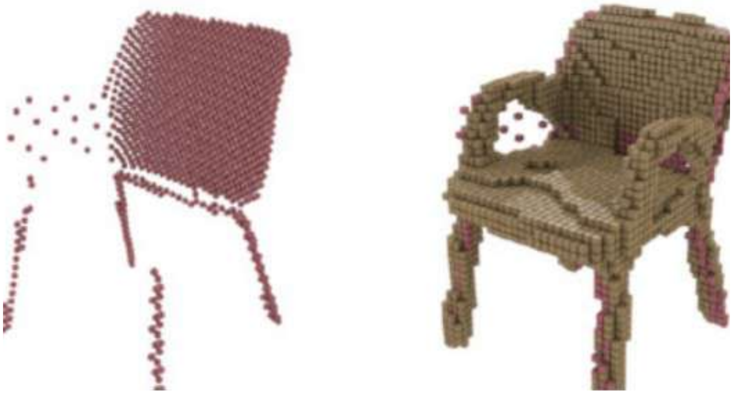


Figure 3.2: Shape completion of a chair. The incomplete point cloud representation (left) is obtained with the 3D projection of the visible surface from the RGB-D image after the segmentation of the object. The output (right) is a representation of the object in voxel space that includes regions not visible in the original image. Adapted from [150]. Copyright held by the authors under CC BY 4.0 license.

3.2.1 Shape Completion

The shape completion task aims to complete the 3D geometry of single objects in a scene (a chair, a table, and so on). To perform this task in real scenes, it is necessary to apply additional instance segmentation methods to isolate the target object from the rest of the scene. Figure 3.2 illustrates the task.

Shape completion methods usually rely on the regularity of the geometry. Such methods comprise plane fitting [106] and object symmetry based approaches [73, 103]. However, these methods often fail when the occluded regions are big or the geometry is not regular. Firman *et al.* [43] proposed another geometry-based approach with promising results, however, as their approach is based purely on geometry without semantics, it produces less accurate results for more complex scenes.

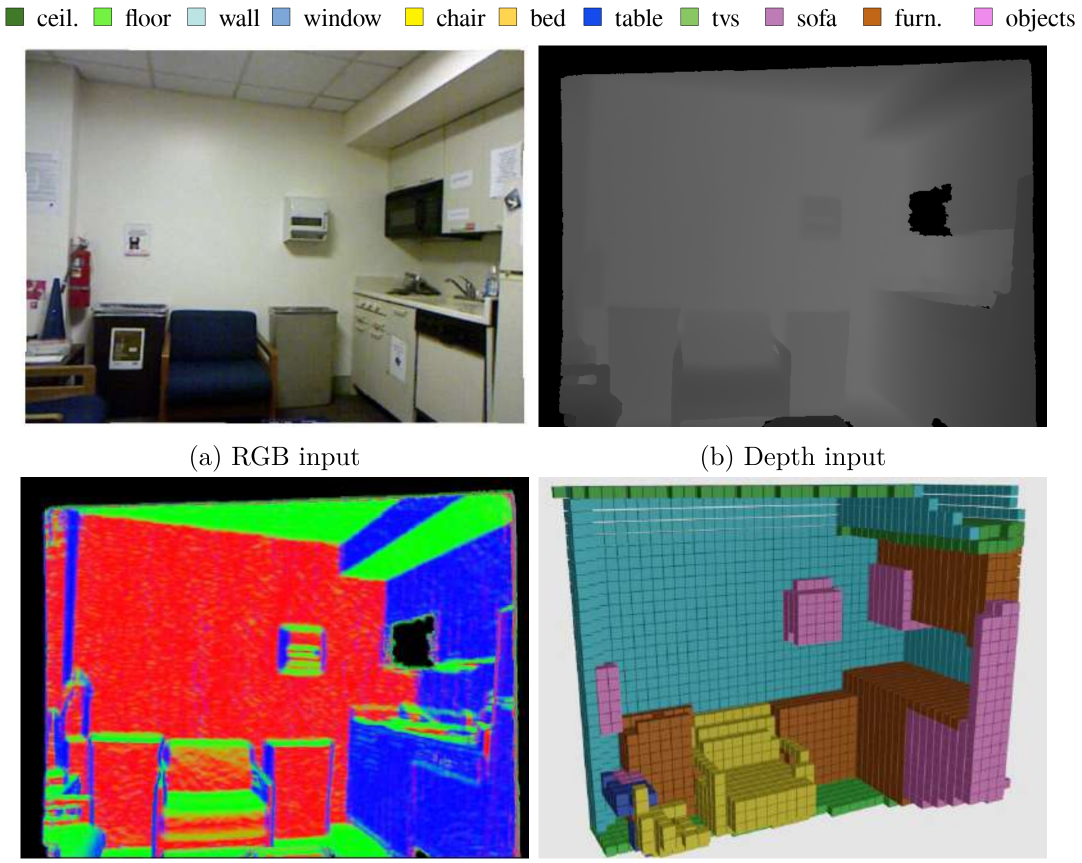
3.2.2 3D model fitting

A common approach to infer the complete geometry and semantic labels for a scene is to fit predefined 3D mesh models to the input depth map [53, 47, 146]. However, the accuracy of these approaches is highly dependent on the quality, quantity, and variety of the 3D mesh models. For real use cases, these solutions are also dependent on an instance segmentation procedure that must be applied previously to the 3D mesh alignment stage. In this type of solution, objects detected by the instance segmentation step that cannot be explained by the available models are expected to be missed by the method. On the other hand, if the available 3D model dataset is large enough to embrace all the possible detected objects, it would rise a difficult problem of retrieval and alignment of the correct model in such a large set of mesh models.

To solve this retrieval and alignment problem, another line of work, instead of using complete 3D mesh models as reference, uses 3D primitives like cuboids to approximately define the complete 3D geometry of detected objects [63, 91, 147]. Obviously, approximating objects to simple 3D primitives leads to severe inaccuracy for complex objects.

3.3 3D Semantic Scene Completion

The term Semantic Scene Completion (SSC) was introduced relatively recently, by Song *et al.* [148]. This new task aims to jointly infer the occupation of the occluded part of the scene (completion) and the semantic labels for all the voxels in the view frustum (semantic labeling). Figure 3.3d illustrates the expected output of the task, extracted from a multimodal SSC solution of ours.



(a) RGB input

(b) Depth input

(c) Surface normals input

(d) SSC output

Figure 3.3: **Multimodal input and expected output for the Semantic Scene Completion Task**. Scene from NYU CAD dataset. (Best viewed in color).

As mentioned before, scenes captured with RGB-D sensors from a single viewing position are subject to occlusion among objects, thus we only get information about the visible surface of the objects. For instance, in the scene depicted in the Figure 3.3, parts of the wall, floor, and furniture are occluded by the chair. There is also self-occlusion: the interior of the chair, its sides, and its rear surfaces are hidden by the visible surface. Because of those characteristics of RGB-D sensors, before the introduction of SSC, most of the work on scene reasoning only partially address the problem, and two scene understanding tasks were common: scene completion and semantic segmentation of visible surface.

In this section, we highlight the contributions introduced in the seminal work by Song *et al.* and also present additional contributions of subsequent works released before our first publication in the field. As SSC is a very active field of study, related works released after our first publication will be addressed in Chapters 5 to 8, to keep the chronological order of release.

3.3.1 The seminal SSC work

Besides being the first work of the task Semantic Scene Completion (SSC), Song *et al.* work [148] introduced several concepts and ideas there are still used. They showed that jointly training for segmentation and completion leads to better results, as both tasks are inherently intertwined.

Prior to the introduction of the SSC task, completion and semantic labeling were seen as two distinct problems. It was common to complete and label 3D scenes with dedicated modules for feature extraction and context modeling [170]. The first work that jointly addressed both problems was SSCNet [148], an end-to-end 3D CNN that simultaneously infers occupancy and semantic labels from depth maps, introduced by Song *et al.* . SSCNet introduced several design and training strategies that are still used nowadays by state-of-the-art solutions. Here we point out the most important ones.

Dilated Convolutions to Enhance Receptive Field

Context is very important to scene reasoning. Relative positions of objects in the scene provide powerful discriminatory contextual information. In a convolutional network, it is instinctive to say that the larger the area of the input image that provides information for a given feature, the greater the amount of context that is considered in that feature. The term Receptive Field (RF) represents this relationship between the features of a given layer and the area of the input image that influences that feature and is defined as the

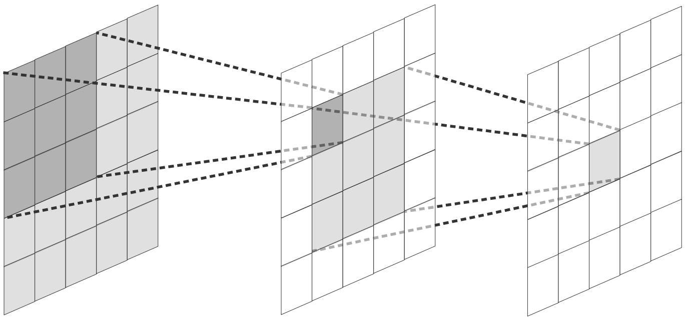


Figure 3.4: Illustration of a 2D CNN's Receptive Field (RF). The RF of a given layer is defined as the size of the region of the input that produces the features of that layer. For a 3D CNN replace the plain regions by volumes.

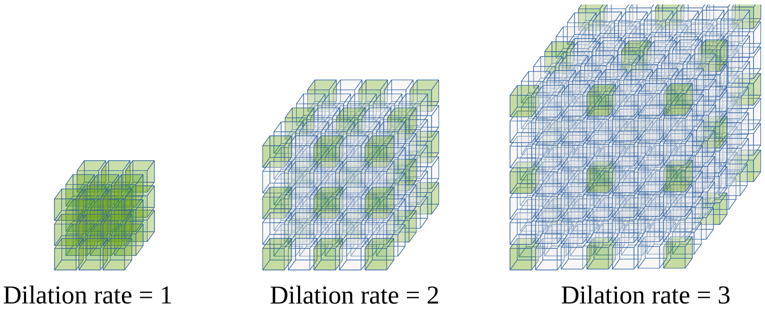


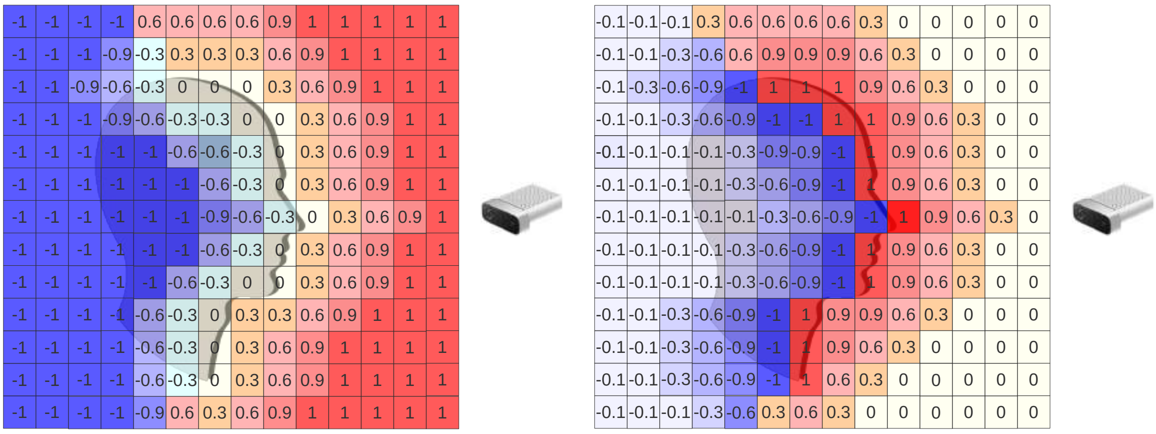
Figure 3.5: Dilated 3D convolution kernels. Green voxels represent active parameters. From left to right: standard 3D convolution kernel (dilation rate = 1); 3D dilated convolution with a dilation rate of 2; and one with a dilation rate of 3. All convolutions have a 3 × 3 × 3 kernel size and the same number of parameters, however, dilated kernels present larger Receptive Field (RF).

size of the region of the input that produces the features of a layer. Figure 3.4 illustrates this concept for a 2D CNN. The RF for a 3D CNN can be easily extrapolated.

To learn the relations between objects in a scene it is necessary to keep a big enough receptive field in the deeper layers of the network. The usual way to achieve a larger receptive field is to enlarge the filters of the convolutions, however, it would consume too much memory. To solve this problem, Song *et al.* extended the dilated convolution [164] to 3D, as shown in Figure 3.5. Dilated convolutions keep the same number of parameters as regular convolutions, but enlarge the region covered by the filter, depending on the dilation rate.

Better 3D volume encoding.

The ideal encoding for this task should represent the RGB-D input into the same representation of the expected output. As seen in Section 2.4.2, TSDF [30] is a very common



-1	-1	-1	-1	0.6	0.6	0.6	0.6	0.9	1	1	1	1	1
-1	-1	-1	-0.9	-0.3	0.3	0.3	0.3	0.6	0.9	1	1	1	1
-1	-1	-0.9	-0.6	-0.3	0	0	0	0.3	0.6	0.9	1	1	1
-1	-1	-1	-0.9	-0.6	-0.3	-0.3	0	0	0.3	0.6	0.9	1	1
-1	-1	-1	-1	-1	-0.6	-0.6	-0.3	0	0.3	0.6	0.9	1	1
-1	-1	-1	-1	-1	-1	-0.6	-0.3	0	0.3	0.6	0.9	1	1
-1	-1	-1	-1	-1	-1	-0.9	-0.6	-0.3	0	0.3	0.6	0.9	1
-1	-1	-1	-1	-1	-1	-0.6	-0.3	0	0.3	0.6	0.9	1	1
-1	-1	-1	-1	-1	-0.6	-0.3	0	0	0.3	0.6	0.9	1	1
-1	-1	-1	-1	-0.6	-0.3	0	0.3	0.3	0.6	0.9	1	1	1
-1	-1	-1	-1	-0.6	-0.3	0	0.3	0.6	0.9	1	1	1	1
-1	-1	-1	-1	-0.6	-0.3	0	0.3	0.6	0.9	1	1	1	1
-1	-1	-1	-1	-0.9	0.6	0.3	0.6	0.9	1	1	1	1	1

(a) TSDF

-0.1	-0.1	-0.1	0.3	0.6	0.6	0.6	0.6	0.3	0	0	0	0	0
-0.1	-0.1	-0.3	-0.6	0.6	0.9	0.9	0.9	0.6	0.3	0	0	0	0
-0.1	-0.3	-0.6	-0.9	-1	1	1	1	0.9	0.6	0.3	0	0	0
-0.1	-0.1	-0.3	-0.6	-0.9	-1	-1	1	1	0.9	0.6	0.3	0	0
-0.1	-0.1	-0.1	-0.1	-0.3	-0.9	-0.9	-1	1	0.9	0.6	0.3	0	0
-0.1	-0.1	-0.1	-0.1	-0.3	-0.6	-0.9	-1	1	0.9	0.6	0.3	0	0
-0.1	-0.1	-0.1	-0.1	-0.1	-0.3	-0.6	-0.9	-1	1	0.9	0.6	0.3	0
-0.1	-0.1	-0.1	-0.1	-0.3	-0.6	-0.9	-1	1	0.9	0.6	0.3	0	0
-0.1	-0.1	-0.1	-0.1	-0.3	-0.6	-0.9	-1	1	0.9	0.6	0.3	0	0
-0.1	-0.1	-0.3	-0.6	-0.9	-1	1	0.9	0.9	0.6	0.3	0	0	0
-0.1	-0.1	-0.3	-0.6	-0.9	-1	1	0.9	0.6	0.3	0	0	0	0
-0.1	-0.1	-0.3	-0.6	-0.9	-1	1	0.6	0.3	0	0	0	0	0
-0.1	-0.1	-0.1	-0.3	-0.6	0.3	0.6	0.3	0	0	0	0	0	0

(b) F-TSDF

Figure 3.6: Comparing TSDF and F-TSDF in 2D. While TSDF provides a smooth transition from visible to ocluded regions, F-TSDF provides a strong change in the surface, highlighting the regions of higher interest for the SSC network. (Best viewed in color).

encoding procedure in RGB-D related tasks to address the sparsity problem, providing a signal invariant to the view-point. However, the transition from the visible region to the occluded region is very smooth. Consider the TSDF encoding illustrated in Figure 3.6a. Note that, on the surface, the encoded value changes from a very small positive value to a very small negative value.

To provide a stronger signal in the surface, Song *et al.* proposed the Flipped Truncated Signed Distance Function (F-TSDF) encoding, given by equation 3.1. In F-TSDF the transition from the visible region to the occluded region goes from 1 to -1 providing a high gradient and thus, a stronger signal in the surface, as Figure 3.6b highlights.

$$F - TSDF = sign(TSDF) \times (1 - |TSDF|)) \tag{3.1}$$

Training on synthetic data.

Song *et al.* introduced SUNCG [148], a large dataset of synthetic indoors scenes, used to pre-train a SSC neural networks before fine-tuning then to real datasets like NYUDv2 [140]. The use of this fine-tuning strategy improves generalization and reduces overfitting. Those datasets are further presented in detail in Section 3.5.

In the following subsections, we present the most important works on Semantic Scene Completion since the introduction of the task in 2017, until our first contribution to the field in 2020. We classify the existing approaches on Semantic Scene Completion according to the modes of the image used to feed the 3D neural network and the previous approaches are shown according to this classification. Further approaches that were introduced during our research project are shown cronologicaly as we present our contributions in Chapters 5 to 8.

3.3.2 Depth maps only

Solutions in this category only use depth information, ignoring all information from RGB channels available in the RGB-D input. SSCNet from Song *et al.* [148] itself is an example. They used depth maps from the SUNCG synthetic dataset to train a typical contracting fully convolutional CNN with 3D dilated convolutions, called SSCNet. To compensate for the data imbalance between empty vs. occupied voxels, SSCNet uses a weighted softmax loss function. As mentioned before, after training on synthetic data, SSCNet was fine-tuned on depth maps from real scenes of the NYUDv2 dataset.

Although SSCNet introduced many contributions, its training pipeline was inefficient. Preprocessing was performed as a network layer, running in the GPU, overloading the deep learning framework. After SSCNet, other works achieved better results with more efficient training pipelines and improved network architectures. Zhang *et al.* [167] used dense conditional random field to enhance SSCNet results. Guo and Tong [51] applied a sequence of 2D convolutions to the depth maps, used a projection layer to project the features to 3D, and feed the output to a 3D CNN.

3.3.3 Depth maps plus RGB

Color information is expected to be useful to distinguish objects that approximately share the same plane in the 3D space, and thus, are hard to be distinguished using only depth. Examples of such instances are flat objects attached to the wall, such as posters, paintings and flat TVs fitted on walls. Some types of closed doors and windows are also problematic for depth-only approaches.

From 2018 on, works on SSC started to explore color information from RGB-D images to improve semantic scene completion scores. Some methods project color information to 3D in a naive way, leading to a problem of data sparsity in the voxelized data that is fed to the 3D CNN. For example, Guedes *et al.* [49] reported preliminary results obtained by adding color to an SSCNet-like architecture. In addition to the F-TSDF encoded depth volume, they used three extra projected volumes, corresponding to the channels of the RGB image, with no encoding, resulting in 3 sparse volumetric representations of the partially observed surfaces. The authors reported no significant improvement using the color information in this sparse manner.

3.3.4 Depth maps plus 2D segmentation

Models in this category use a two-step training protocol, where a 2D segmentation CNN is first trained, and then it is used to generate input to a 3D semantic scene completion CNN. Current models differ in the way the generated 2D information is fed into the 3D CNN.

Garbade *et al.* [45] used a pre-trained 2D segmentation CNN with a fully connected CRF [22] to generate a segmentation map, which, after post-processing, was projected to 3D. Liu *et al.* [96] used depth maps and RGB information as input to an encoder-decoder 2D segmentation CNN. The encoder branch of the 2D CNN is a ResNet-101 [60] and the decoder branch contains a series of dense upsampling convolutions. The generated features from the 2D CNN are then reprojected to 3D using camera parameters, before being fed into a 3D CNN. The paper shows results using 2 different strategies to fuse depth and RGB: SNetFusion performs fusion just after the 2D segmentation network, while TNetFusion only performs fusion after the 3D convolutional network. TNetFusion achieves higher performance, with a much higher computational cost. The 2D CNN is also pre-trained offline.

Using 2D segmentation maps on 3D SSC brings additional complexity to the training phase which is training and evaluating the 2D segmentation network prior to the 3D CNN training. In Chapter 5, we present our alternate end-to-end approach to fuse information from depth and color, where the network can be trained and evaluated as a whole, and still achieves state-of-the-art performance, without the extra cost of training a 2D network.

3.4 360°Scene Understanding

Current Semantic Scene Completion approaches are restricted to the limited field of view of the regular RGB-D sensors. However, there are sensors and cameras that could enable 360°scene reasoning tasks. One goal of this thesis is to extend SSC to a 360°coverage. In this section we present some previous works that explore the full scene coverage for other tasks than SSC, but, in some way, relate to our objectives.

3.4.1 Scene Understanding from Large Scale Scans

The Scene Understanding research field observed a boost after the public availability of high-quality datasets like Stanford 2D-3D-Semantics Dataset [2] and Matterport3D [20], acquired with the Matterport® camera, which comprises point cloud ground truth of the whole buildings, 360° RGB panoramas and corresponding depth maps and other features. The scanning process uses a tripod-mounted sensor that comprises three color and three depth cameras pointing slightly up, horizontally, and slightly down. It rotates and acquires RGB photos and depth data, which are combined to generate 360° RGB-D panoramas [20]. These datasets allowed the development of several scene understanding works [21, 95, 115]. Most of these works focus only on the visible surfaces, rather than on the full understanding of the scene including occluded regions and inner parts of the objects.

In a different line of work, Im2Pano3D [149] uses data from large-scale scans to train a CNN that generates a surface-only prediction of a full 360° view of an indoor scene from a given partial view of the scene corresponding to a regular RGB-D image. The work that is most related to our goals is ScanComplete [31]. Using data from synthetic or real large-scale datasets and a generative 3D CNN, it tries to complete the scene and classify all surface points. However, unlike SSC, it takes inputs from multiple viewpoints.

Although large-scale scans provide a workaround to surpass the FOV limitations of popular RGB-D sensors, they have the significant drawback that multiple captures of the scene are required to cover a complete scene layout. In addition, each acquisition is a slow scanning process that can only work if the scene remains static for the duration of all captures. Therefore it may be unfeasible to apply them for dynamic scene understanding.

3.4.2 Scene Understanding using 360° Stereo Images

Spherical imaging provides a solution to overcome the drawbacks inherent to large-scale scans. Schoenbein *et al.* proposed a high-quality omnidirectional 3D reconstruction pipeline that works from catadioptric stereo video cameras [130]. However, these catadioptric omnidirectional cameras have a large number of systematic parameters that need to be set, including the camera and mirror calibration.

In order to get high-resolution spherical images with accurate calibration and matching, Spheron developed a line-scan camera, Spheron VR[1], with a fish-eye lens to capture the full environment as an accurate high resolution / high dynamic range image. Li [89] has proposed a spherical image acquisition method that uses two video cameras with fish-eye lenses pointing in opposite directions. Various inexpensive off-the-shelf 360° cameras with two fish-eye lenses have recently become popular[2,3,4]. However, 360° RGB-D cameras which automatically generate depth maps are not yet available. Kim and Hilton proposed depth estimation and scene reconstruction methods using a pair of 360° images from various types of 360° cameras [69, 72], shown in Figure 2.11. We applied this

[1] Spheron, `https://www.spheron.com/products.html`

[2] Insta360, `https://www.insta360.com`

[3] GoPro Fusion, `https://shop.gopro.com/EMEA/cameras/fusion/CHDHZ-103-master.html`

[4] Ricoh Theta, `https://theta360.com/en/`

stereo-based method to acquire depth maps from images captured with 360° cameras in the experiments presented on Chapter 8.

3.5 Densely annotated RGB-D Datasets

In the past decade, RGB-D sensors have enabled major breakthroughs for several computer vision tasks through the public availability of several large-scale sparsely annotated datasets [78, 145]. However, regarding the SSC task, which requires densely annotated datasets for training, there are not so many labeled data available. Given the high cost of acquiring dense labels, the use of synthetic data is an affordable and reliable approach [148]. In this section, we present the most used real and synthetic SSC benchmark datasets for training 3D CNN.

3.5.1 NYUD v2

NYUD v2 [140] is a popular dataset containing RGB-D images captured using MS Kinect on real indoor scenes. Scenes are gathered from commercial and residential buildings, comprising 464 different indoor scenes, divided into 795 samples for training and 654 for testing. Although not included in the original dataset, dense ground truth is usually generated by voxelizing the 3D mesh annotations from [50] and mapped object categories based on [57] to label occupied voxels with semantic classes. Song *et al.* [148], made this generated ground truth publicly available.

NYUD v2 is a challenging dataset due to the small number of images for training and the misalignment between ground truth generated from 3D object meshes and the depth maps.

3.5.2 NYUCAD

NYUCAD [43] is a dataset generated from NYUD v2 where the depth maps are synthesized from the 3D objects meshes, eliminating the misalignment between the depth map and the 3D ground truth. The RGB images and 2D segmentation ground truth are the original ones from NYUD v2.

NYUCAD is a less challenging dataset than the original NYUD v2, but it is still a small dataset for training deep neural networks.

3.5.3 SUNCG

SUNCG dataset [148] consists of about 45K synthetic scenes generated from manually created architectural 3D house models with realistic room and furniture layouts. From the scenes, more than 130K 3D scenes were rendered with corresponding depth maps and ground truth, emulating images collected from an RGB-D sensor. The dataset provides standard training and test sets for benchmark purposes.

As Song *et al.* [148] did not use the RGB images in their work, they did not include the rendered RGB images of the scenes in the original dataset. As we believe that the RGB channels should provide important extra information to the SSC task, we extracted the camera poses from the provided ground truth and rendered a new set of depth and RGB images from the SUNCG synthetic scenes. To avoid misalignments, the ground truth volumes were regenerated from the scene meshes.

Training on SUNCG and then fine-tuning to NYUD v2 or NYUCAD is a reliable approach to overcome the difficulty of only having a small amount of densely labeled real RGB-D images for training.

3.6 Chapter Summary

In this chapter, we presented some works closely related to our research project, focusing on depth-enabled inputs for scene understanding like RGB-D sensors and binocular images from a single point of view.

We showed important works that illustrated how scene understanding tasks evolved from simple 2D image segmentation to more complete 3D semantic scene completion. We further showed the next step on semantic scene completion that corresponds to replacing the input gathered with a limited view of most common 3D sensors with a full 360°coverage input image from sensors like Matterport or 360°binocular cameras. We also introduced the main benchmark datasets for the SSC task. The works and datasets presented here will be particularly useful for Chapters 5 to 8.

Chapter 4

Fully Convolutional Networks, Domain Adaptation and Semantic Segmentation

The goal of this Chapter is to confirm the effectiveness of Fully Convolutional Network (FCN) approaches while performing Semantic Segmentation in comparison to more traditional approaches. We also want to assess the benefits of Domain Adaptation techniques applied to deep learning models. For this chapter, we choose to work in the 2D domain, as 3D tasks are much more complex and require much more computational power. This choice is also motivated by a wish to broaden the scope of our evaluations, as the 2D image segmentation literature is much wider than its counterpart in 3D. Another reason to explore the use of 2D FCN for Segmentation is the possibility to use the output of such networks, when fed with RGB-D images, as an extra input mode for a 3D CFN.

More specifically, we choose to investigate the human skin detection problem as a study case because it is binary, simplifying the analysis, and is a widely studied topic of Computer Vision for which it is commonly accepted that analysis of pixel color or local patches may suffice and the use of deep FCN is not necessary. Besides that, in our literature review, we have not found studies related to the benefits of Domain Adaptation in this kind of application.

The content of this Chapter was mainly extracted from our paper **Domain adaptation for holistic skin detection** [36] which was published at the 34th SIBGRAPI Conference on Graphics, Patterns, and Images (SIBGRAPI 2021).

4.1 2D Image Segmentation

In 2017, Brancati *et al.* [14] achieved state-of-the-art results in skin segmentation using correlation rules between the YCb and YCr subspaces to identify skin pixels on images. Faria proposed a variation of that method and Hirata [42], claimed to have achieved a new state-of-the-art plateau on rule-based skin segmentation based on neighborhood operations. Lumini and Nanni [100] compared different color-based and CNN-based skin detection approaches on several public datasets and proposed an ensemble method.

In contrast to Domain Adaptation for image classification, it is difficult to find literature focused on domain adaptation methods for image segmentation [27], especially for the skin detection problem. San Miguel and Suja [127] use agreement of two detectors based on skin color thresholding, applied to selected images from several manually labeled public datasets for human activity recognition, but do not explore their use in cross-domain setups. Conaire *et al.* [25] also use two independent detectors, with their parameters selected by maximizing agreement on correct detections and false positives to dynamically change a classifier on new data automatically without any user annotation. Kamnistas [68] use unsupervised domain adaptation to improve brain lesion detection in MR images. Bousmalis *et al.* [13] developed a generative adversarial network model which adapts source-domain images to appear as if drawn from the target domain, a technique that enables dataset augmentation for several computer vision tasks.

4.2 The Skin Detection Problem and the Use of Domain Adaptation

Human skin detection is the task of identifying which pixels of an image correspond to skin. It has several applications: video surveillance, people tracking, human-computer interaction, face detection and recognition, and gesture detection, among many others [133, 101].

Before the boom of Convolutional Neural Networks (CNNs), most approaches were based on skin-color separation or texture features, as in [61] and [138]. By that time, there were other approaches for image segmentation in general, like Texton Forest [136] and Random Forest [135]. As occurred with image classification from 2012 on, convolutional networks have become very successful in segmentation tasks. One of the first approaches using deep learning was patch-based classification [24], where each pixel is classified using a patch of the original image that surrounds it. This is a local approach that does not consider the context of the image as a whole. Later, Shelhamer *et al.* [134] introduced the Fully Convolutional Networks (FCNs), a global approach, where image segmentation is done holistically, achieving state-of-the-art results in several reference datasets.

Despite all the advances that deep fully convolutional neural networks have brought for image segmentation, some common criticism is made to argue that pixel-based approaches are still more suitable for skin detection. Namely,

1. the need for large training datasets [67]; one may not know in advance the domain of the images that will be used, therefore, no amount of labeled training data may be enough;
2. the specificity or lack of generalization of neural nets; and
3. their inference time [14]; especially for video applications where the frame rate is around 30 or 60 frames-per-second, allowing a maximum total processing time of 17 to 33ms per image.

Those arguments seem to ignore several proposed approaches that exploit unlabeled data of the domain of interest (unsupervised domain adaptation) or labeled data and models from other domains (inductive transfer learning) to solve the lack of labeled data. Amid the fast evolution of CNNs and domain adaptation techniques, we ask ourselves: *Do those criticisms still hold for the skin detection problem?*

In this chapter, to address the first criticism (on the need for large training datasets), we propose a new Domain Adaptation strategy that combines Transfer Learning and Pseudo-Labeling [85] in a cross-domain scenario that works under several levels of target domain label availability. We evaluate the proposed strategy on several cross-domain situations on four well-known skin datasets. We also address the other criticisms with a series of comprehensive in-domain and cross-domain experiments. Our experiments show the effectiveness of the proposed strategy and confirm the superiority of FCN approaches over local approaches for skin segmentation. With the proposed strategy we can improve the F_1 score on skin segmentation using little or no labeled data from the target domain.

4.3 Methods

In this Chapter, we compare two CNN approaches (patch-based and fully convolutional) with above mentioned state-of-the-art pixel-based methods for in-domain skin detection. We also compare the two CNN approaches to each other in cross-domain setups, even in the absence of target-domain labeled data. Previous state-of-the-art pixel-based skin segmentation papers do not present results on cross-domain setups. We also propose to combine the strengths of both inductive transfer learning and unsupervised or semi-supervised domain adaptation using Pseudo-Labeling to address the lack of training data

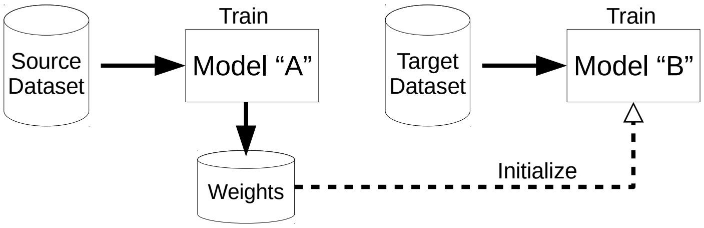


Figure 4.1: Inductive Transfer Learning by fine-tuning parameters of a model to a new domain. Model "A" parameters are trained on the source dataset. Model "B" parameters are initialized from Model "A" parameters. Model "B" is then fine-tuned to the new domain by progressively unfreezing layers.

issue using cross-domain setups. In this section, we present details of the training approaches, models, and experimental protocols.

4.3.1 Cross-domain learning approaches

To exploit domain adaptation techniques to address training data availability problem for skin segmentation, we evaluate conventional inductive transfer learning using fine-tuning, our cross-domain extension applied to the Pseudo-Labeling approach of [85] and our proposed combined approach that uses both inductive transfer learning and unsupervised or semi-supervised DA. Here we present each one of these approaches.

Inductive Transfer Learning approach

For inductive transfer learning with deep networks, we use the learned parameters from the source domain as a starting point for optimization of the parameters of the network on the target domain. The optimization first focuses on the modified output layer, which is intimately linked to the classification task. Other layers are initially frozen, working as a feature extraction method. Next, all parameters are unfrozen and optimization carries on until convergence. This can be seen as a way to regularise the learned parameters on the source domain and avoid catastrophic forgetting. Figure 4.1 illustrates this process, which is widely used and known as fine-tuning [28].

Cross-domain Pseudo-Labeling approach

In this work, we propose a method that is related to the pseudo-labeling approach of Lee [85], but instead of using the same model and domain for final prediction and pseudo-label generation, we use a model trained in a different domain to generate pseudo labels for the target domain. These pseudo-labels are then used to fine-tune the original model or to train another model from scratch in a semi-supervised manner. We call this technique **cross-domain pseudo-labeling**.

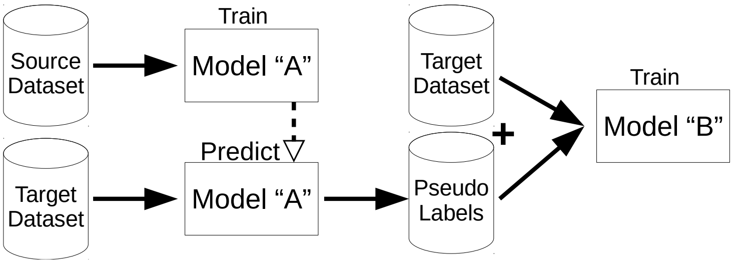


Figure 4.2: Semi-supervised and unsupervised Domain Adaptation by cross-domain pseudo-labeling. Model "A" is trained on the source dataset and it is used to predict labels on the target dataset. Then, the target dataset and previously predicted labels are used to train Model "B". When no labels are available on the target dataset, the process is unsupervised.

Figure 4.2 illustrates this procedure. This approach allows us to train the final model with very few labeled data of the target domain. In the worst-case scenario, the model can be trained with no true label at all, in a fully unsupervised fashion. This still takes advantage of entropy regularization of the pseudo-label technique.

Combined approach

Our last approach consists in combining fine-tuning and pseudo-labeling approaches to improve the final model performance. Figure 4.3 illustrates this procedure. We use weights obtained from a cross-domain pseudo-label model (Model "B") to fine-tune a model that will be used to generate a more accurate set of pseudo-labels. These new pseudo-labels are then used in one in-domain pseudo-label training round to get the final model ("Model C"). The intuition behind this approach is that using a more accurate set of labels jointly with weights of a better model should lead to better results. Because of the fine-tuning step, which requires at least some labels from the target dataset, this approach is semi-supervised.

4.3.2 Models

We evaluated two approaches for skin segmentation, a local (patch-based) convolutional classification method and a holistic (FCN) segmentation method. Here we describe these methods.

Patch-based CNN

The patch-based approach uses the raw values of a small region of the image to classify each pixel position based on its neighborhood. The architecture of the CNN is presented on Figure 4.4 Inspired by the architecture described by Ciresan *et al.* [24], we use a 3 convolutional layer network with max-pooling between convolutions, but, in the inner

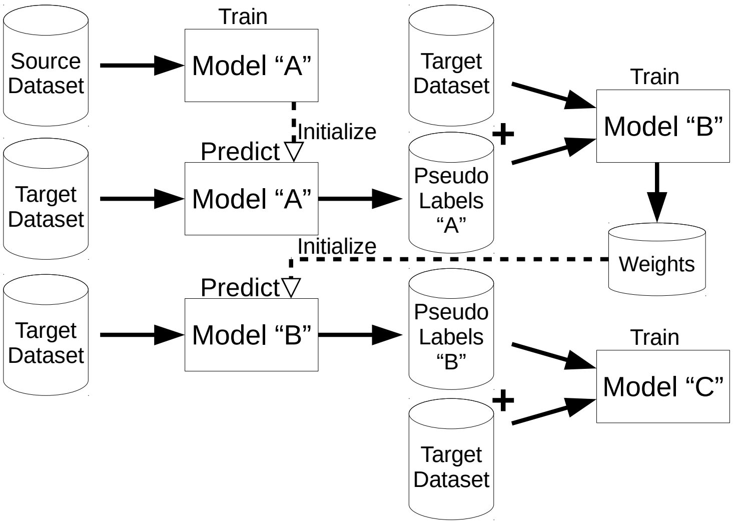


Figure 4.3: Combined transfer learning and domain adaptation approach. Model "A" is trained on the source dataset and it is used to predict labels on the target dataset. Then, the target dataset and previously predicted labels are used to train Model "B" which is fine-tuned on the target dataset before being used to generate a new set of more accurate pseudo-labels.

layers, used ReLU instead of a nonlinear activation function. As input, we use a patch of 35 × 35 pixels and 3 channels, to allow the network to capture the surroundings of the pixel. This patch size is similar to that used by Ciresan *et al.* [24] (32 × 32), but we chose an odd number to focus the prediction in the center of the patch. The output of the network consists of two fully connected layers and a sigmoid final activation for binary classification. For this approach, the images are not resized. To reduce the cost of training while maintaining data diversity, data subsampling is used so that only 512 patches are randomly selected from each image. For inference, all patches are extracted in a sliding window fashion, making one prediction per pixel. Due to the path size, the prediction process generates a 17-pixel-wide border where this method does not predict an output, so zero padding is applied. This does not cause much harm to the predictions, since the presence of skin near the borders is rare in all datasets used.

Holistic segmentation FCN

Due to its simplicity and performance, we choose to use the U-Net as the holistic segmentation method to be evaluated in this Chapter. Our model follows the general design proposed by Ronneberger *et al.* [120], but we use a 7-level structure with the addition of batch normalization between the convolutional layers, as shown in figure 4.5. We also use an input frame of 768 × 768 pixels and 3 channels to fit most images and the same size output.

Smaller images are framed in the center of the input and larger ones are resized in a way that their larger dimension fits the input frame. For evaluation purposes, predictions are done over the images restored to their original sizes.

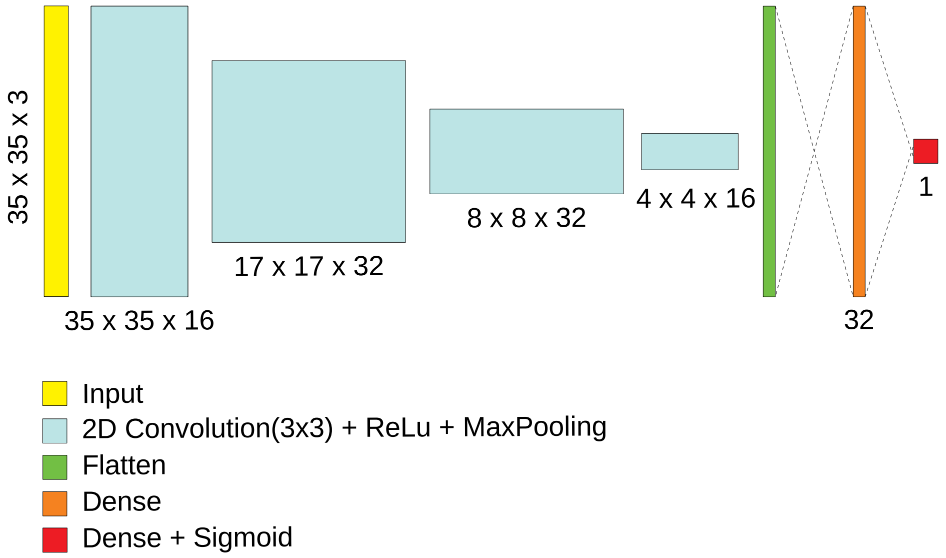


Figure 4.4: Our patch-based model.

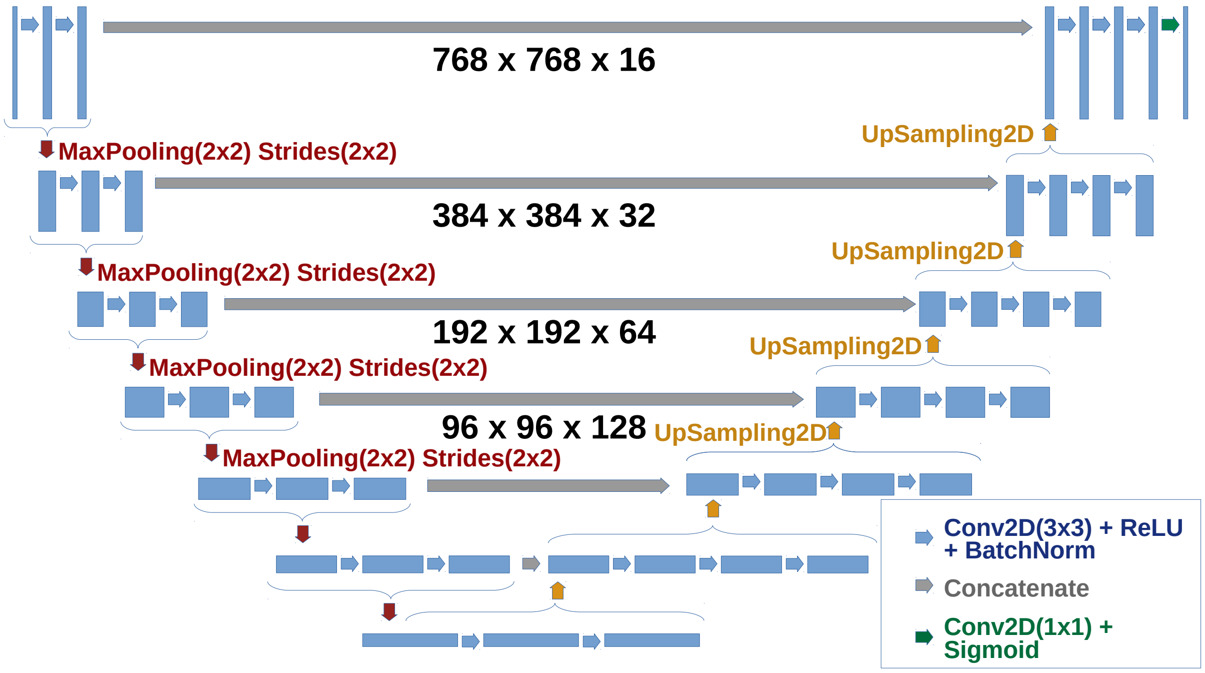

768 x 768 x 16
MaxPooling(2x2) Strides(2x2)
UpSampling2D
384 x 384 x 32
MaxPooling(2x2) Strides(2x2)
UpSampling2D
192 x 192 x 64
UpSampling2D
MaxPooling(2x2) Strides(2x2)
96 x 96 x 128
UpSampling2D
MaxPooling(2x2) Strides(2x2)
Conv2D(3x3) + ReLU + BatchNorm
Concatenate
Conv2D(1x1) + Sigmoid

Figure 4.5: Our variation of the U-Net architecture for holistic image segmentation.

4.3.3 Evaluation measures and loss function

From the literature, we have identified that the most popular evaluation criteria for image segmentation are Accuracy (Acc), Jaccard Index (a.k.a. Intersection over Union, IoU), Precision, Recall, and F_1 Score (a.k.a. Sørensen–Dice Coefficient or Dice Similarity Coefficient). In this section, we revise them following a notation that helps to compare them. For each given class label, let $\vec{p} \in [0,1]^{\mathcal{I}}$ be the vector of predicted probabilities for each pixel (where $\mathcal{I}$ is the number of pixels in each image), $\vec{q} \in \{0,1\}^{\mathcal{I}}$ be the binary vector that indicates, for each pixel, if that class has been detected, based on $\vec{p}$, and $\vec{g}$ be the ground truth binary vector that indicates the presence of that label on each pixel. We have the following definitions:

$$\begin{aligned}
\mathrm{Acc} &= \frac{\sum_i^{\mathcal{I}} \mathbb{1}_{g_i}(q_i)}{\mathcal{I}} = \frac{\vec{q}\cdot\vec{g} + (\vec{1}-\vec{q})\cdot(\vec{1}-\vec{g})}{\mathcal{I}} && (4.1)\\
\mathrm{IoU} &= \frac{|\vec{q}\cap\vec{g}|}{|\vec{q}\cup\vec{g}|} = \frac{\vec{q}\cdot\vec{g}}{\sum_i^{\mathcal{I}} \max\left(p_c, g_c\right)} = \frac{\vec{q}\cdot\vec{g}}{|\vec{q}|+|\vec{g}|-\vec{q}\cdot\vec{g}} && (4.2)\\
\mathrm{Prec} &= \frac{\vec{q}\cdot\vec{g}}{|\vec{q}|} && (4.3)\\
\mathrm{Rec} &= \frac{\vec{q}\cdot\vec{g}}{|\vec{g}|} && (4.4)\\
\mathrm{F}_1 &= \left(\frac{\mathrm{Prec}^{-1}+\mathrm{Rec}^{-1}}{2}\right)^{-1} = 2\cdot\frac{\vec{q}\cdot\vec{g}}{|\vec{p}|+|\vec{g}|} && (4.5)
\end{aligned}$$

Also, from 4.2 and 4.5, we can derive that the Jaccard index and F_1 score are monotonic in one another:

$$\mathrm{IoU} = \frac{\mathrm{F}_1}{2-\mathrm{F}_1} \qquad \therefore \mathrm{F}_1 = \frac{2\cdot\mathrm{IoU}}{1+\mathrm{IoU}} \tag{4.6}$$

As such, there is no quantitative argument to prefer one over the other. Qualitatively, though, we recommend using F_1 score as it is a more prevalent metric in other fields. Although accuracy has been widely used, we consider that not to be a good metric, as its numerator not only considers true positives, but also true negatives, and a null hypothesis gives high accuracy on imbalanced datasets.

In most cases, we evaluate results using Precision (Prec), Recall, and F_1 scores, because they are the most popular metrics for skin detection and additionally provide Accuracy (Acc) and Intersection over Union (IoU) scores. However, in dense tables, to save space, we just present results in terms of F_1 score, because it is the most used score.

As for the loss function, the training objective and evaluation metric should be as close as possible, but the F_1 score is not differentiable. Therefore, we use a modified (and differentiable) Sørensen–Dice coefficient, given by equation 4.7, where s is the smoothness parameter that was set to $s = 10^{-5}$. The derived loss function is given by equation 4.8.

$$softDiceCoef(\vec{p}, \vec{g}) = \frac{s + 2\vec{p} \cdot \vec{g}}{s + |\vec{p}| + |\vec{g}|} \tag{4.7}$$

$$DiceLoss(P, G) = 1 - softDiceCoef(P, G) \tag{4.8}$$

4.3.4 Data augmentation

In both local and holistic models, the image pixels are normalized to 0 to 1 and the sigmoid activation function is applied to the output. In both models, we used data augmentation, randomly varying pixel values in the HSV color space (uniform probability from -100 to $+100$ in each channel). For the U-Net model, we also used random shift (uniform probability from -9% to $+9\%$) and flip (uniform probability 50%).

4.4 Experiments and results

The main goal of our experiments is to evaluate the performance of homogeneous transductive fine-tuning, cross-domain pseudo-labeling, and a combined approach in several domains and under different availability of labeled data in the target domain. To achieve this goal, we used four well-known datasets dedicated to skin segmentation (described in Section 4.4.1) and permuted them as source and target domains. The first set of experiments (Section 4.4.2) was conducted to compare the CNN approaches to the state-of-the-art pixel-based works. The second set of experiments (Section 4.4.3) was designed to evaluate the generalization power and the amount of bias in each dataset. Next, to evaluate the cross-domain approaches, for each pair of datasets and for each approach we performed a range of experiments using different amounts of labeled training data from the target domain (Section 4.4.4).

4.4.1 Datasets

The datasets we used were Compaq [65] – a widely used skin dataset with 4,670 images of several levels of quality; SFA [18] – a set of 1,118 face images obtained from two distinct datasets, most of them with white background; Pratheepan [163] – 78 family and face photos, randomly downloaded using Google; and VPU [127] – 290 images extracted from video surveillance cameras.

To evaluate the methods, SanMiguel and Suja [127] proposed a pixel-based split of training and testing samples (not image-based) for the VPU dataset, making it impossible to evaluate holistic methods. The other datasets do not have a standard split of samples.

Table 4.1: Same domain results on the SFA dataset (in %).

Model	Acc	IoU	Prec	Recall	F_1
Faria and Hirata (2018) [42]	-	-	92.88	39.58	55.51
Our patch-based	91.14	82.17	89.71	91.00	90.35
Our U-Net	**97.94**	**92.80**	**96.65**	**95.89**	**96.27**

Table 4.2: Same domain results on the Compaq dataset (in %).

Model	Acc	IoU	Prec	Recall	F_1
Brancati *et al.* (2017) [14]	-	-	43.54	**80.46**	56.50
Our patch-based	90.18	46.00	58.92	73.59	65.45
Our U-Net	**92.62**	**54.47**	**68.49**	71.64	**70.03**

For this reason, we adopted the same test split reported by the authors of SFA [18], which uses 15% of the images for testing and the remaining for training on all these datasets.

4.4.2 In-domain evaluations

The same-domain training evaluation results are shown on tables 4.1, 4.2, 4.3 and 4.4. Our fully convolutional U-Net model surpassed all recent works on skin segmentation available for the datasets in study, and, in most cases, our patch-based CNN model stands in second, confirming the superiority of the deep learning approaches over feature engineering methods. The results also show that the datasets have different levels of difficulty, being VPU the most challenging one and SFA being the least challenging one. The best accuracy was obtained on VPU, but this is because this is a heavily unbalanced dataset where most pixels belong to the background. As for all remaining criteria, the best results occurred on SFA, which confirms our expectation, as SFA is a dataset of frontal mugshot-style photos.

4.4.3 Cross-domain baseline results

The cross-domain capabilities of our models and generalization power of domains are shown in table 4.5, which presents source-only mean F_1 scores results without any transfer or adaptation to the target dataset. As we can see, the source dataset Compaq in conjunction with the U-Net Model presented the best generalization power on targets SFA and Pratheepan. Source dataset Pratheepan also in conjunction with the U-Net Model did better on targets Compaq and VPU. These source-only setups surpassed the respective color-based approaches shown in previous tables, except for the VPU dataset.

Note that the patch-based model surpassed U-Net when using source domains with low generalization power like SFA and VPU. For example, using VPU as the source domain

Table 4.3: Same domain results on the Pratheepan dataset (in %).

Model	Acc	IoU	Prec	Recall	F_1
Brancati *et al.* (2017) [14]	-	-	55.13	81.99	65.92
Faria and Hirata (2018) [42]	-	-	66.81	66.83	66.82
Our patch-based	87.12	55.57	59.83	**82.49**	69.36
Our U-Net	**91.75**	**60.43**	**72.91**	74.51	**73.70**

Table 4.4: Same domain results on the VPU dataset (in %).

Model	Acc	IoU	Prec	Recall	F_1
SanMiguel and Suja (2013) [127]	-	-	45.60	**73.90**	56.40
Our patch-based	93.48	14.14	46.34	42.82	44.51
Our U-Net	**99.04**	**45.29**	**57.86**	71.33	**63.90**

and SFA as the target, patch-based reached a mean F_1 score of 82.63%, while U-Net only got 14.83%. Using SFA as the source and Compaq as the target, patch-based also surpassed U-Net (54.80% vs. 18.92%). These results are expected since SFA and VPU are datasets of very specific domains with little variation in the type of scenes between their images (SFA images are close-ups of faces and VPU images are typical views from conference rooms or surveillance cameras). On the other hand, Compaq and Pratheepan include images with a wide range of layouts. Therefore, SFA and VPU only offer relevant information at a patch level for skin detection, their contexts are very specific, which hinders their generalization ability. If the goal is to design a robust skin detector and avoid negative transfer, our results show that it is better to use Compaq or Prateepan as source samples.

Table 4.5: Cross-domain mean F_1 scores (%) obtained without transfer or adaptation.

Model	Source Domain	Target Domain			
		SFA	Compaq	Prathee.	VPU
U-Net	SFA	-	18.92	44.98	11.52
	Compaq	**86.14**	-	**75.30**	23.67
	Prathee.	80.66	**63.49**	-	**36.68**
	VPU	14.83	44.71	48.02	-
Patch	SFA	-	54.80	62.92	21.60
	Compaq	71.28	-	72.59	19.94
	Prathee.	80.04	62.68	-	13.74
	VPU	82.63	51.48	58.34	-

Table 4.6: U-Net mean F_1 scores under different scenarios and domain adaptation approaches.

Source	Target	Approach	Target Training Label Usage				
			0%	5%	10%	50%	100%
Target only	SFA	Target only	-	93.49	94.50	95.72	**96.27**
	Compaq		-	**66.84**	**67.78**	**69.37**	70.03
	Pratheepan		-	46.36	59.86	69.04	73.70
	VPU		-	41.27	53.44	63.18	63.90
Compaq	SFA	Source only	86.14	-	-	-	-
		Fine-tuning only	-	92.89	94.04	**95.86**	95.98
		Cross-domain pseudo-label only	88.80	88.90	89.69	93.22	-
		Combined approach	**89.24**	90.05	90.36	94.57	-
	Pratheepan	Source only	75.30	-	-	-	-
		Fine-tuning only	-	72.52	74.69	76.47	**77.16**
		Cross-domain pseudo-label only	75.58	75.52	77.18	**80.08**	-
		Combined approach	**76.80**	**75.67**	**77.84**	79.87	-
	VPU	Source only	**23.67**	-	-	-	-
		Fine-tuning only	-	**51.51**	46.50	67.47	**69.62**
		Cross-domain pseudo-label only	02.67	02.86	02.68	02.77	-
		Combined approach	02.66	02.68	02.67	02.66	-
Pratheepan	SFA	Source only	80.66	-	-	-	-
		Fine-tuning only	-	**93.68**	**94.70**	**95.69**	95.99
		Cross-domain pseudo-label only	82.50	83.36	83.63	90.60	-
		Combined approach	**82.96**	84.12	84.47	92.93	-
	Compaq	Source only	**63.49**	-	-	-	-
		Fine-tuning only	-	64.88	66.10	68.97	**70.52**
		Cross-domain pseudo-label only	39.50	41.26	44.69	62.39	-
		Combined approach	34.72	36.22	39.05	57.06	-
	VPU	Source only	**36.68**	-	-	-	-
		Fine-tuning only	-	**51.61**	**60.19**	**68.15**	**69.44**
		Cross-domain pseudo-label only	02.66	02.66	02.67	02.77	-
		Combined approach	02.65	02.66	02.67	02.74	-

4.4.4 Domain Adaptation Results

Following the recommendation in the previous section, we performed domain adaptation experiments using Compaq and Pratheepan as source datasets. Table 4.6 presents the F_1 scores obtained by the methods and settings we evaluated. For each source→target pair, we indicate in boldface which result was better than the target-only method. We evaluated the effect of the amount labeled target samples given and present results ranging from no labels (0%), i.e. an unsupervised domain adaptation setting, to all labels (100%) given in the target training set, i.e., an inductive transfer setup. Target-only results are provided for comparison purposes, i.e, within domain experiments with the number of training labels ranging from 5 to 100%. The target-only results are expected to be an upper bound in performance when 100% of the training labels are used because there is no domain change, but they may suffer from the reduced training set size in comparison to the domain adaptation settings.

Compaq has confirmed our expectations of being the most generalizable source dataset,

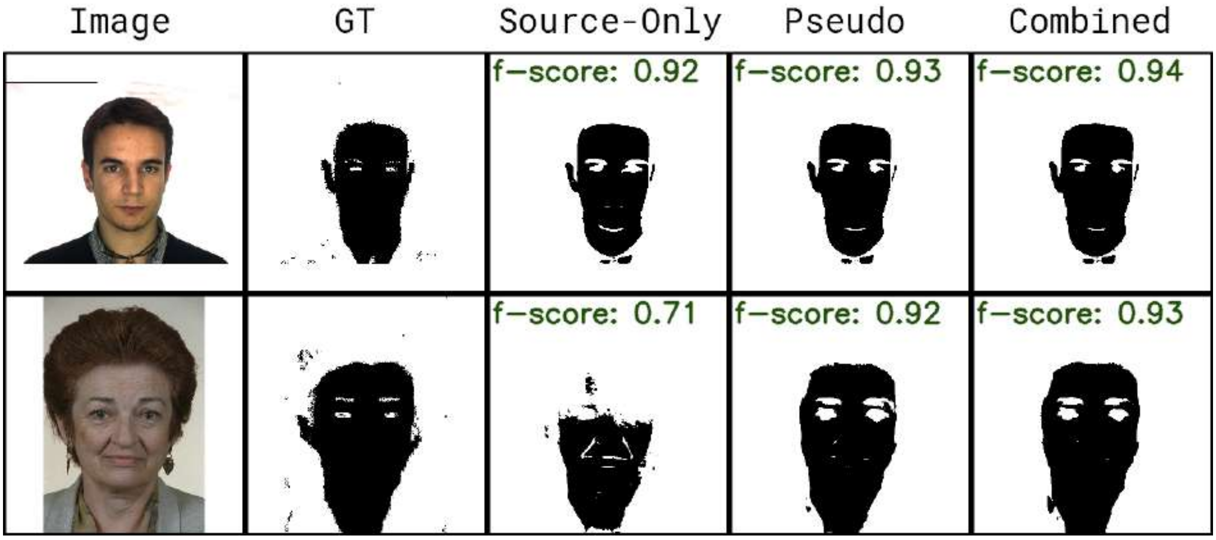


Figure 4.6: Domain adaptation from Compaq to SFA using no real labels from the target. From left to right: target test image, ground truth, and results with source only, domain adaptation based on cross-domain pseudo-labels, and the combined domain adaptation + transfer learning approach.

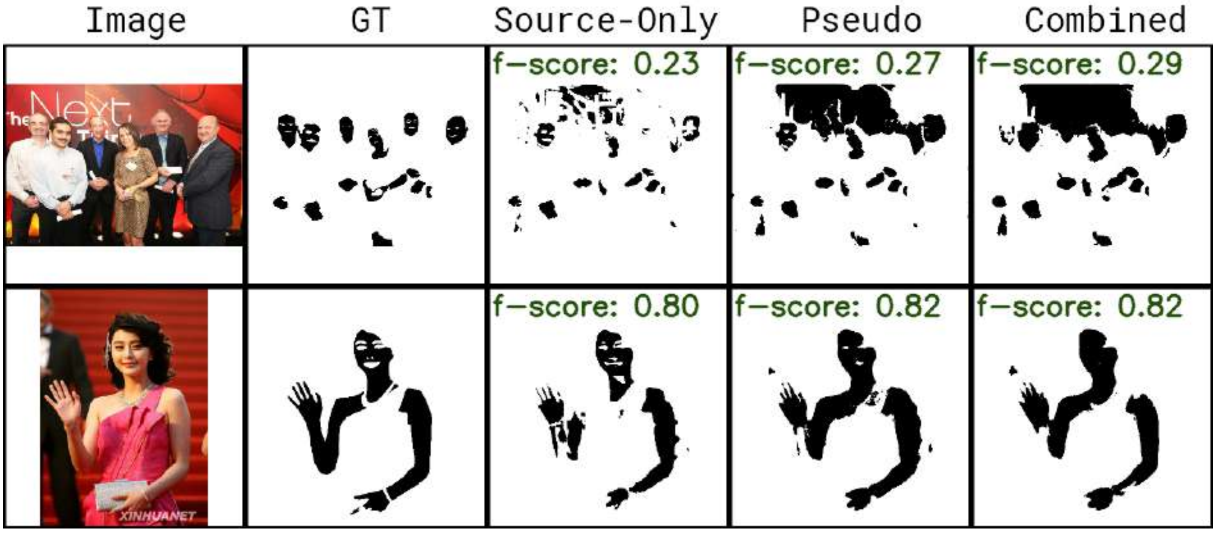


Figure 4.7: Domain adaptation from Compaq to Pratheepan using no real labels from the target (same setting as Figure 4.6).

not only for being the most numerous in terms of sample images but also due to their diversity in appearance. The use of Compaq as the source leads to very good results on SFA and Pratheepan as targets. These results are illustrated in figures 4.6 and 4.7, respectively, which show the effects of using different domain adaptation methods with no labels from the target dataset. Note that when using Compaq as the source and Pratheepan as the target, the gain of the domain adaptation approaches is very expressive when compared to target-only training. Domain adaptation methods got better results using any amount of labels on the target training set, being the combined approach the best option in most cases. Using 50% of training data our cross-domain pseudo-label approach was better than regular supervised training with 100% of training data. Besides that, all the results of domain adaptation methods with no labels were better than the state-of-the-art results of color-based approaches presented in Section 4.4.2.

When VPU is the target dataset, Pratheepan outperformed Compaq as the source dataset. However, the pseudo-labels caused negative transfer, leading to very bad results when domain adaptation was used. The results with fine-tuning were better than regular supervised training with all evaluated amounts of training labels. In this scenario, the reference color-based approach by [127] was beaten starting from 10% of training label usage. Results with 5, 10, and 50% are shown for two sample images in Figure 4.8.

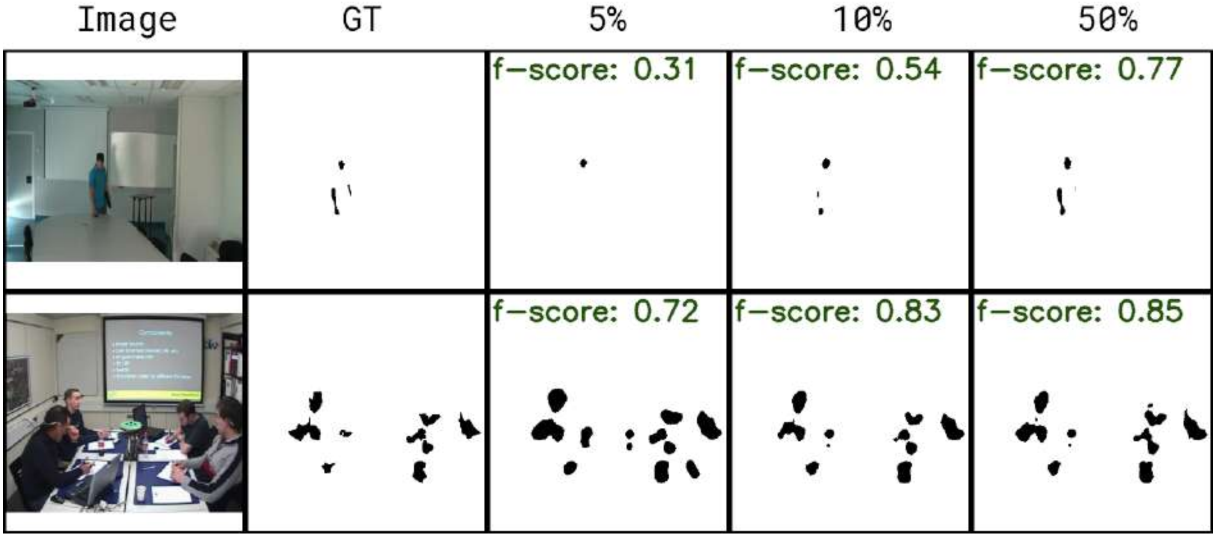


Figure 4.8: Adaptation from Pratheepan to VPU with fine-tuning TL. From left to right: target test image, ground truth, and results with 5, 10, and 50% of labels on the target training set.

Still, with Pratheepan as the source dataset, but with Compaq as the target, the "source-only" result was reasonable and surpassed the color-based approach. However, we observed that domain adaptation methods did not remarkably improve the results from regular supervised training. Figure 4.9 shows the results of fine-tuning from Pratheepan to Compaq.

4.4.5 Discussion

Although most approaches for skin detection in the past have assumed that skin regions are nearly textureless [19, 61, 143, 133, 14], our results give the unintuitive conclusion that texture and context play an important role. A holistic segmentation approach like fully convolutional networks, taking the whole image as input, in conjunction with ade-

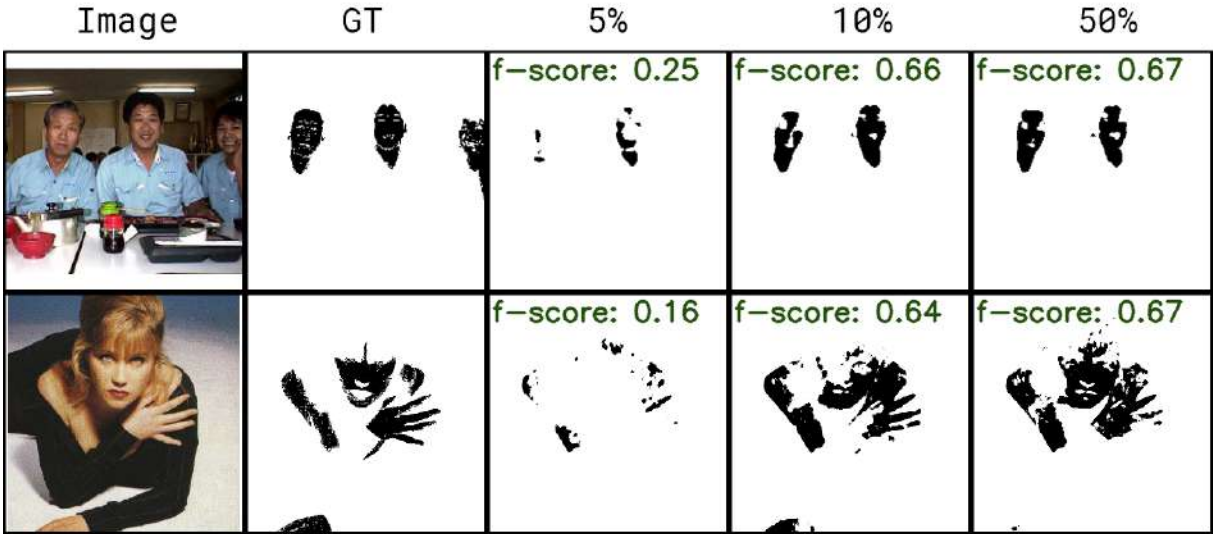


Figure 4.9: Adaptation from Pratheepan to Compaq with fine-tuning using different amounts of labels on the target training set (following the same setting as Figure 4.8).

quate domain adaptation methods, has more generalization power than local approaches like color and patch-based. The improvement level and best domain adaptation approach vary depending on how close the target and source domains are and on the diversity of the samples in the source dataset. The closer the domains and the higher the source variety, the higher the improvement. For example, a very positive transfer from Compaq→SFA was observed because Compaq is more diverse and includes samples whose appearance is somewhat similar to those of SFA. This is intuitive, as these approaches depend on the quality of the pseudo-labels. When the transition between domains goes from specific to diverse datasets, the pseudo-labels are expected to be of low quality, thus, not contributing to the target model training. In these situations, fine-tuning has shown to be more effective, although with the drawback of requiring at least a few labeled images for training.

Figure 4.10, on the other hand, shows the comparison of regular supervised training versus the fine-tune approach in the Pratheepan → VPU scenario. As Pratheepan does not cover scenes that occur on VPU, the fine-tune approach performs better than cross-domain pseudo-labels in this scenario.

Domain Adaptation methods have also shown improvements when compared to regular supervised training in cases where the target has few images, like Pratheepan and VPU. The level of improvement depends on the amount of labeled target training data and on the similarity of source and target domains. The higher the amount, the lower the improvement, and the higher the similarity, the higher the improvement. Figure 4.11 shows a comparison of regular supervised training versus the combined approach in the Compaq→Pratheepan scenario with 5, 10, and 50% of the target training samples with

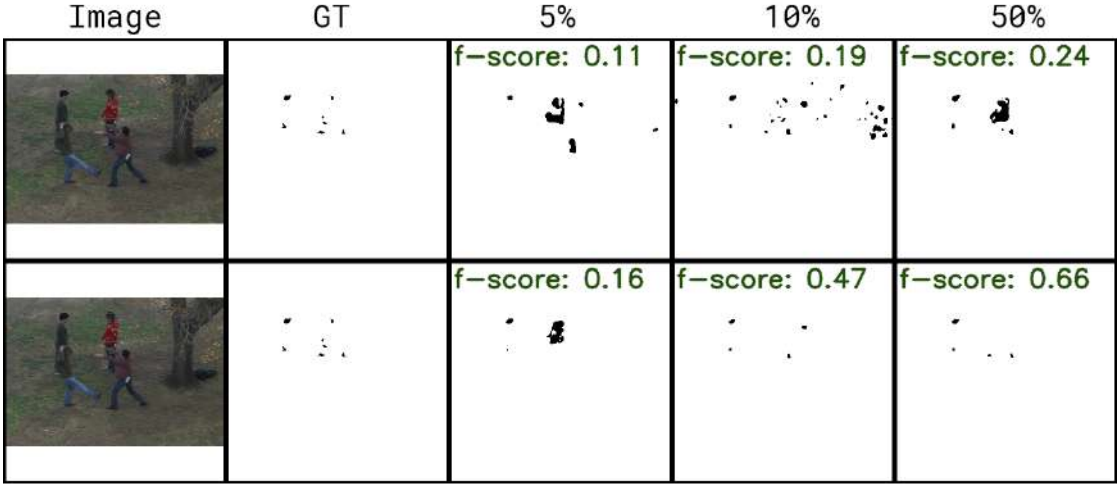


Figure 4.10: Comparison of source only vs. fine-tune in the Pratheepan → VPU scenario with different proportions of labeled target training samples. For each target test image, the first row is regular supervised training and the second is the fine-tuning approach.

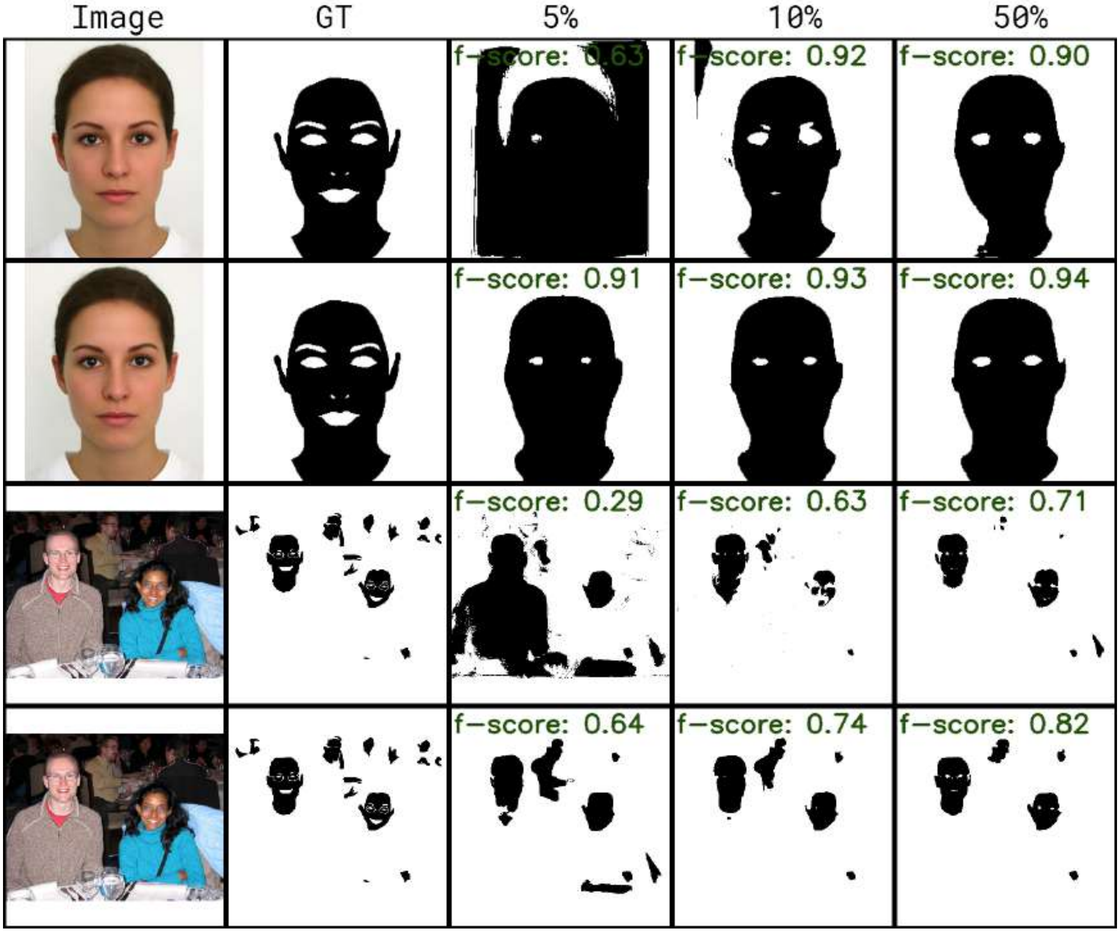


Figure 4.11: Comparison of source only vs. domain adaptation combined approach in the Compaq→Pratheepan scenario with different proportions of labeled target training samples. For each target test image, the first row is regular supervised training and the second is the combined domain adaptation approach.

labels. This scenario is good for the pseudo-label approach since Compaq has more diversity than Pratheepan. Note the superiority of the combined approach at all levels of the target labels availability.

Another important aspect to be addressed is the criticism of the applicability of CNN approaches to real-time applications. The criticism is probably valid for patch-based CNN approaches but does not hold for our FCN holistic approach. The average prediction time of our patch-based CNN, using a simple NVIDIA GTX-1080Ti, with a frame size of 768 × 768 pixels, is 7 seconds per image which is indeed not suitable for real-time applications. However, our U-Net prediction time is 80 ms per frame for the same setup, i.e., 12.5 images are processed per second (without parallel processing). [14] has reported a prediction time of about 10ms per frame with a frame size of 300 × 400 pixels (8× faster on images that are 5× smaller), which is indeed a bit faster, at a penalty of producing worse results.

4.5 Chapter Summary

In this Chapter, we refuted some common criticisms regarding the use of Deep Convolutional Networks for skin segmentation. We compared two CNN approaches (patch-based and holistic) to the state-of-the-art pixel-based solutions for skin detection in in-domain situations. As our main contribution, we proposed novel approaches for semi-supervised and unsupervised domain adaptation applied to skin segmentation using CNNs and evaluated them with an extensive set of experiments.

Our evaluation of in-domain skin detection approaches on different domains/datasets showed the expected and incontestable superiority of CNN-based approaches over color-based ones. Our U-Net model obtained F_1 scores which were on average 30% better than the state-of-the-art recently published color-based results. In more homogeneous and clean datasets, like SFA, our F_1 score was 73% better. Even in more difficult and heterogeneous datasets, like Prathepaan and VPU, our U-Net CNN was more than 10% better.

More importantly, we experimentally came to the unintuitive conclusion that a holistic approach like U-Net, besides being much faster, gives better results than a patch-based local approach.

We also concluded that the common critique of the lack of generalization of CNNs does not hold against our experimental data. With no labeled data on the target domain, our domain adaptation method's F_1 score is an improvement of 60% over color-based results for homogeneous target datasets like SFA and 13% in heterogeneous datasets like Pratheepan.

Note that the approaches for both inductive transfer learning (TL) and unsupervised domain adaptation (DA) are baseline methods. More sophisticated approaches have been proposed for both problems, such as [98, 41, 27, 46]. Our study shows that, despite the simplicity of the chosen methods, they greatly contribute to the improvement in the performance of skin segmentation across different datasets, showing that even better results are expected with more sophisticated methods. For example, our results were in general better than the individual methods gathered in [100] and on par with their proposed ensemble method.

Chapter 5

Using RGB Edges to improve Semantic Scene Completion from RGB-D Images

As we saw in Chapter 3, Semantic Scene Completion (SSC) is the task of predicting a complete 3D representation of volumetric occupancy with corresponding semantic labels for a scene given a single RGB-D image. It was established fairly recently [148] and consists of, given a single RGB-D image, classifying the semantic labels of all the voxels within the voxel space of the field-of-view, including occluded and non-surface regions. The seminal work used a large synthetic dataset (SUNCG) to generate approximately 140 thousand depth maps that were used to train a typical contracting fully convolutional neural network with 3D dilated convolutions.

Before the beginning of our research project, most works on SSC used either depth-only or depth with color by projecting 2D semantic labels generated by a 2D segmentation network into the 3D volume, requiring a two-step training process and suffering from the sparsity problem when projecting features from 2D to 3D. In this Chapter, we present our proposed EdgeNet, a new end-to-end fully convolutional neural network architecture that fuses information from depth and RGB, explicitly representing RGB edges encoded in 3D space using F-TSDF, thus solving the sparsity problem. Our proposed network is a FCN that improves semantic scene completion scores, especially in hard-to-detect classes. We achieved state-of-the-art scores on both synthetic and real datasets with a simpler and more computationally efficient training pipeline.

The work described in this Chapter focuses on enhancing semantic scene segmentation scores using information from both the depth and color of RGB-D images in an end-to-end manner. To address the RGB data sparsity issue, we introduce a new strategy for encoding information extracted from RGB images in 3D space. We also present a new end-to-end 3D CNN architecture to combine and represent the features from color and depth. Comprehensive experiments were conducted to evaluate the main aspects of the proposed solution. Results show that our fusion approach is superior to depth-only solutions and that EdgeNet achieves equivalent performance to the current state-of-the-art fusion approach, with a much simpler training protocol.

The content of this Chapter was mainly extracted from our paper **EdgeNet: Semantic Scene Completion from a single RGB-D image** [34] which was published in the proceedings of the 25th International Conference on Pattern Recognition (ICPR 2020). This work was developed in collaboration with the Centre of Vision, Speech and Signal Processing (CVSSP) of the University of Surrey, UK.

5.1 Our solution: EdgeNet

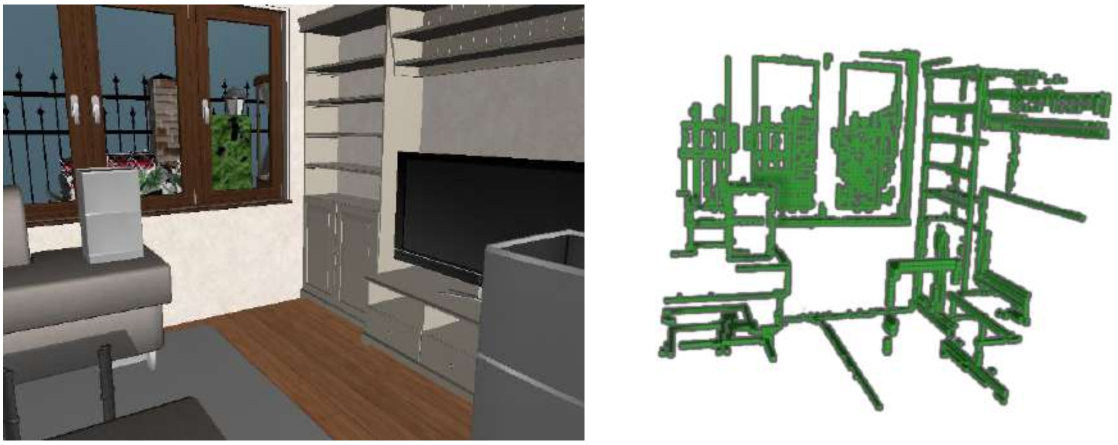
Our proposed solution is the first end-to-end approach that successfully uses information from RGB to improve semantic scene completion performance over depth only. It consists of a novel approach to encoding information from RGB edges and depth maps and a new 3D CNN architecture to fuse both modalities. We call it EdgeNet.

5.1.1 Encoding edges in 3D

As discussed earlier, color information should complement depth maps for 3D semantic scene completion. However, the combination of these modalities in a meaningful representation of learning is not trivial. Guedes *et al.* [49] naively added 3 channels to each voxel to insert R, G, and B color information into the representation, with no encoding. In this way, the vast majority of voxels have no color data while only those on the visible surface have a color value. This explains why they do not improve on the previous approach using depth only.

As described in Sections 2.4.2 and 3.3.1 respectively, TSDF is a common procedure to tackle the sparsity problem in 3D volumes projected from depth maps and F-TSDF is a flipped version of it. Song *et al.* [148] demonstrate that F-TSDF encoding plays an important role in feeding a projected depth map to a 3D CNN and produces better results than TSDF or no encoding at all. With F-TSDF, a discontinuity near the occupied surface (from -1 to 1) occurs and the first derivative tends to infinity. This type of signal helps the convergence of the network.

F-TSDF encoding of volumetric data can be easily applied to depth maps after 3D projection because each voxel carries binary information: occupied or free. On the other hand, F-TSDF can not straightforwardly be applied to RGB or semantic segmentation



(a) (b)

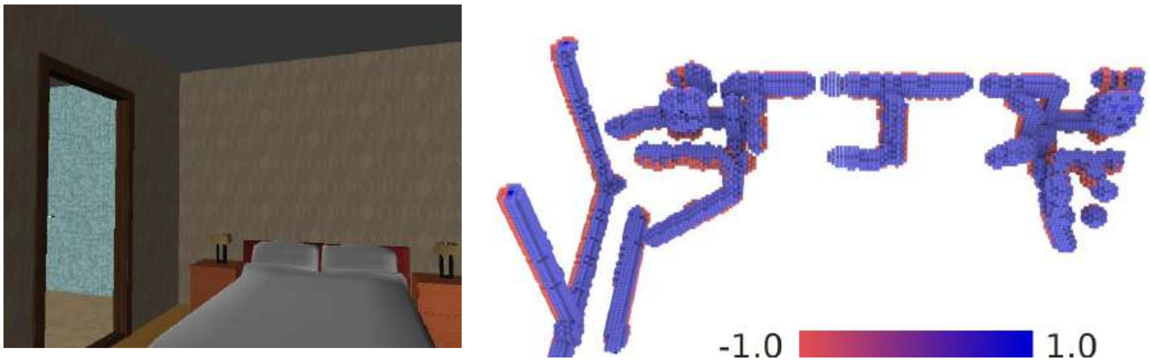
Figure 5.1: Projection of Edges to 3D: (a) original RGB image, (b) voxelized edges after projection.

maps[1], because they are not binary. To deal with this problem, we introduce a new strategy to fuse color appearance and depth information for 3D semantic scene completion. Our approach exploits edge detection in the image, which gives a 2D binary representation of the scene that can highlight objects that are hard to detect in depth maps. For instance, a poster on a wall is expected to be invisible in a depth map, especially after down-sampling. On the other hand, RGB edges highlight the presence of that object.

The main advantage of extracting edges and projecting them to 3D is the possibility to apply F-TSDF on both edges and surface volumes, as they are both binary, providing two complementary and strong input signals to the 3D CNNs. Another advantage is that due to their simplicity, edges are more transferable, removing the need for the application of a domain adaptation method when learning from synthetic images and applying them to real images.

We apply F-TSDF to 3D edges, similarly to F-TSDF applied to 3D surfaces: for each voxel in the edge volume, our method looks for the nearest edge to calculate the Euclidean distance. Visible and occluded voxels are related only to edges, not to surfaces. We use the standard Canny edge detector [17] and each edge location is projected to a point in the 3D space using its depth information and the camera calibration matrix. The resulting point cloud is voxelized in the same way as the depth point cloud, resulting in a sparse volume of $240 \times 144 \times 240$ voxels. Figure 5.1 shows a scene from the SUNCG dataset and its corresponding edges projected to 3D. Figure 5.2b shows in detail a region of the

[1] Theoretically, it is possible to apply F-TSDF to segmentation maps, however, it would be necessary to apply one-hot-encoding to the input segmentation map and the resulting number of channels of the input would be the number of classes. Each channel would generate a binary 3D volume and F-TSDF would be applied to each one of the volumes. However, this comes with a prohibitive GPU memory footprint.



(a)

(b)

Figure 5.2: (a) original scene. (b) F-TSDF of edges in 3D. The edge image is a horizontal cut of the scene, taken just above the bed. Only F-TSDF values with absolute values greater than 0.8 are shown (best viewed in color).

projected edges of 5.2a after F-TSDF encoding. Note that a sharp change occurs along the edges.

5.1.2 EdgeNet architecture

To combine depth and edge modalities, we propose a new 3D semantic segmentation CNN architecture that we call **EdgeNet**. Our proposed solution is a 3D CNN inspired by the U-Net design [120] which has successfully been used in many 2D semantic segmentation problems (see Chapter 4) and is presented in Figure 5.3. We address the degradation problem of deeper networks [59], by replacing simple convolutional blocks of U-Net with ResNet modules [60]. In lower resolutions, the ResNet modules use dilated convolutions to improve the receptive field. To match the resolution of the output, the input branch reduces the resolution to 1/4 of the input. The next blocks follow an encoder-decoder design and the last stage of the decoding branch is responsible for reducing the number of channels to match the desired number of output classes and loss calculations.

Depth and Edges Fusion Schemes. The encoder-decoder structure of EdgeNet allows us to evaluate three fusion schemes: Early Fusion (EdgeNet-EF), Middle-level Fusion (EdgeNet-MF), and Late Fusion (EdgeNet-LF). In EdgeNet-EF, just after F-TSDF encoding, both input volumes are concatenated and fed into the main network. In EdgeNet-MF, the input branch is divided into two parts while in EdgeNet-LF, both input and encoding branches are divided. To keep the same memory requirement in all fusion schemes, the total quantity of channels in all schemes is always the same.

Data balancing and loss function. In volumetric data, occluded and occupied voxels are highly unbalanced, so we use a weighted version of categorical cross entropy as the loss function to train our models. To obtain the weights, for each training batch, we

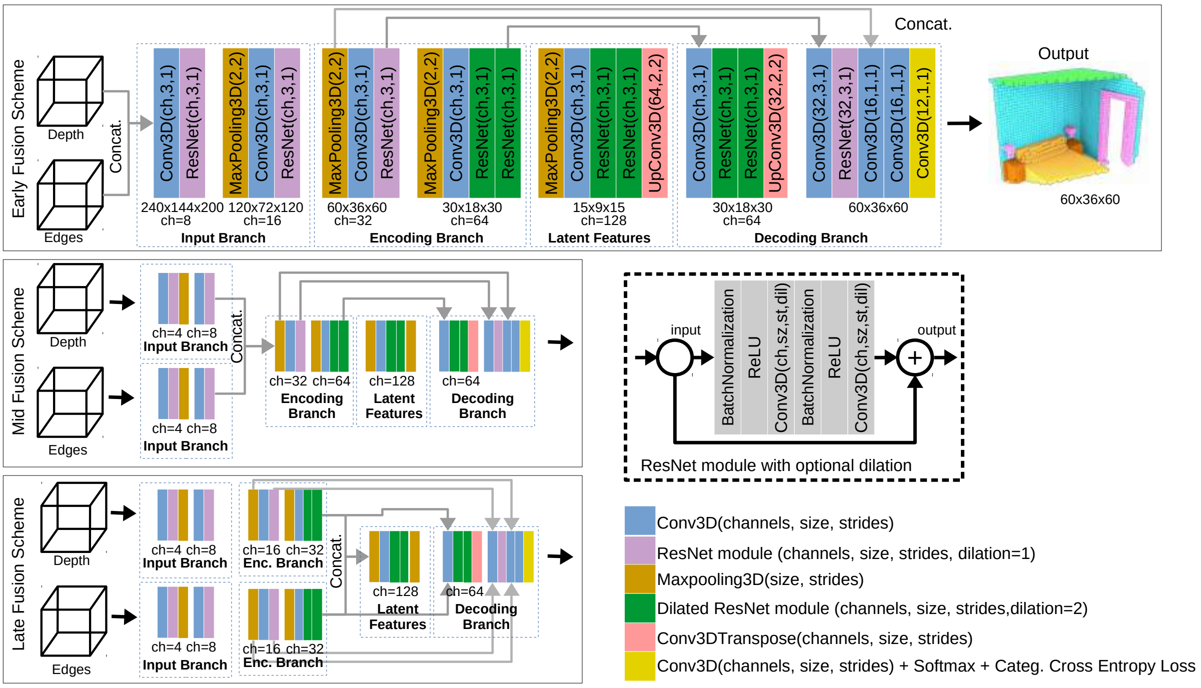


Figure 5.3: Our EdgeNet proposed architecture and fusion schemes (best viewed in color).

randomly initialize a tensor $rand_{occl}$ of the same shape as the batch with ones and zeroes using the ratio $r = (2\sum occu / \sum occl)$, where $occu$ and $occl$ are two tensors obtained from the offline calculated occupancy grid relative to occupied and occluded voxels (See Section 5.1.3). The final weight tensor is $w = occu + occl \odot rand_{occl}$, where $\odot$ denotes the Hadamard product. Let p be the predicted probabilities of the 12 classes for each voxel and y be the one hot encoded ground truth tensor. The categorical cross-entropy loss function is then given by

$$L_{cce}(p, y) = -\sum(w \odot y \odot \log p). \tag{5.1}$$

5.1.3 Training pipeline with offline data preparation

As F-TSDF calculation is computationally intensive, to reduce overall training time, the F-TSDF volumes that feed the models are preprocessed offline once. The preprocessed dataset is then stored, and may be used as many times as needed, including by different models. Following previous works, we rotate the 3D Scene to align it with gravity and have room orientation based on the Manhattan assumption. We fixed the dimensions of the 3D space to 4.8 m horizontally, 2.88 m vertically, and 4.8 m in depth. Voxel grid size is 0.02 m, resulting in a $240 \times 144 \times 240$ 3D volume. The TSDF truncation value is 0.24 m. Surface and edge projection as well as F-TSDF encoding of all volumes are done in this stage. During preprocessing, we also calculate an occupancy grid where we distinguish occupied voxels inside the room and FOV; non-occupied occluded voxels inside the room and FOV; and all other voxels. This occupancy grid will be further used to balance the dataset during training time.

5.2 Experiments

In this section, we describe the datasets and the evaluation protocol we used.

5.2.1 Datasets

We train and validate our proposed approach on SUNCG [148] and NYUDv2 [140] datasets (Refer to Section 3.5 for a complete description). Recall that SUNCG's original training and test sets did not include RGB images. As we need the RGB images for EdgeNet, we extracted the camera poses from the provided ground truth and rendered a new set of depth and RGB images from the SUNCG synthetic scenes (Details on Section 3.5).

5.2.2 Training protocols

Our experiments consist in training our models from scratch on SUNCG and NYUDv2, and also fine-tuning models trained from SUNCG to NYUDv2. For experiments in which we trained our models from scratch, we use the technique known as One Cycle Learning [141], which is a combination of Curriculum Learning [10] and Simulated Annealing [1]. After some preliminary evaluations, we found 0.01 to be a good base learning rate. We use a maximum of 30 epochs, to maintain total training time within an acceptable limit. Following Smith [141], we start with the base learning rate and linearly increase the effective learning until 0.1 in the 10th epoch, then linearly decrease the learning rate until reaches the start-up level in the 20th epoch. During the annealing phase, we linearly go from 0.01 to 0.0005 in a further 10 epochs. Due to GPU memory size constraints, we use batches of 3 samples. We use the SGD optimizer with a momentum of 0.9 and decay of 0.0005 in all experiments, as used in most previous works. For SUNCG, each epoch consists of 30,540 scenes randomly selected from the whole training set. For NYUDv2, each epoch comprises the whole training set. For fine-tuning, we initialize the network with parameters trained on SUNCG and use the standard training policy with SGD with a fixed learning rate of 0.01 and 0.0005 of weight decay.

Thanks to our lightweight training pipeline with offline F-TSDF preprocessing, our training time is only 4 days on SUNCG and 6 hours on NYUDv2, using a GTX 1080 TI. In contrast, Song *et al.* took 7 days on SUNCG and 30 hours on NYUDv2.

input	model	scene completion			semantic scene completion (IoU, in percentages)											
		prec.	rec.	IoU	ceil.	floor	wall	win.	chair	bed	sofa	table	tvs	furn.	objs.	avg.
d	SSCNet[148]	76.3	**95.2**	73.5	96.3	84.9	56.8	28.2	21.3	56.0	52.7	33.7	10.9	44.3	25.4	46.4
	SSCNet*	92.7	89.7	83.8	97.0	94.6	74.3	51.1	43.7	78.2	70.9	49.5	45.2	61.0	51.3	65.2
	DCRF [167]	–	–	–	95.4	84.3	57.7	24.5	28.2	63.4	55.3	34.5	19.6	45.8	28.7	48.8
	VVNetR-120 [51]	90.8	91.7	84.0	**98.4**	87.0	61.0	54.8	49.3	83.0	**75.5**	55.1	43.5	68.8	57.7	66.7
	EdgeNet-D	93.1	90.4	84.8	97.2	94.4	78.4	56.1	50.4	80.5	73.8	54.5	49.8	69.5	59.2	69.5
d+s	SNetFuse[96]	56.7	91.7	53.9	65.5	60.7	50.3	56.4	26.1	47.3	43.7	30.6	37.2	44.9	30.0	44.8
	TNetFuse[96]	53.9	95.2	52.6	60.6	57.3	53.2	52.7	27.4	46.8	53.3	28.6	41.1	44.1	29.0	44.9
d+e	SSCNet-E	92.8	89.6	83.8	97.0	94.5	74.6	51.8	43.9	77.0	70.8	49.3	49.2	62.1	52.0	65.7
	EdgeNet-EF(Ours)	**93.7**	90.3	**85.1**	97.2	94.9	**78.6**	57.4	49.5	80.5	74.4	**55.8**	51.9	70.1	**62.5**	**70.3**
	EdgeNet-MF(Ours)	93.3	90.6	**85.1**	97.2	**95.3**	78.2	**57.5**	**51.4**	**80.7**	74.1	54.5	**52.6**	**70.3**	60.1	70.2
	EdgeNet-LF(Ours)	93.0	89.6	83.9	97.0	94.6	76.4	52.0	44.6	79.8	71.5	48.9	48.3	66.1	55.9	66.8

Table 5.1: **Results and ablation studies on SUNCG test set**. We took SSCNet as a baseline and show the effect of each one of the main aspects of our proposed approach. Column 'input' indicates the type of input: d = depth only; d+e = depth + edges. SSCNet* is our implementation of the original SSCNet, with our training pipeline. EdgeNet-D has the same architecture as the other versions of EdgeNet, but the edge volume is not fed into the network. EdgeNet-EF achieves the best overall scores and surpassed VVNetR-120 by 3.3% on average IoU for semantic scene completion.

5.2.3 Evaluation

For the semantic scene completion task, we report the Intersection over Union (IoU) of each object class on both the observed and occluded voxels. For the scene completion task, all non-empty object classes are considered as one category, and we report Precision, Recall, and IoU of the binary predictions on occluded voxels[2]. Voxels outside the view or the room are not considered.

5.2.4 Experimental results

We compare our results to semantic scene completion approaches that use depth-only [51, 148, 167], depth plus RGB [49] and depth plus 2D segmentation maps [45, 96]. We also investigate the effects of the main aspects of our proposed solution on SUNCG. Comparative results were extracted from the original papers.

Ablation Studies and results on SUNCG

In Table 5.1, investigate the effects of the main aspects of our proposed solution. First, we analyze the effect of our training pipeline. We took SSCNet as a baseline and retrain it, using our lightweight training framework, which allows a batch size of 3 samples in comparison to the 1 sample batch size of the original SSCNet. The results of that exper-

[2] Despite what is said in Chapter 4, section 4.3.3 regarding the evaluation of 2D semantic segmentation, F1 scores has not been used for 3D semantic scene completion. Therefore we use IoU as most of the previous works.

train	input	model	scene completion			semantic scene completion (IoU, in percentages)											
			prec.	rec.	IoU	ceil.	floor	wall	win.	chair	bed	sofa	table	tvs	furn.	objs.	avg.
SUNCG	d	SSCNet[148]	55.6	91.9	53.2	5.8	81.8	19.6	5.4	12.9	34.4	26	13.6	6.1	9.4	7.4	20.2
	d+e	EdgeNet-EF(Ours)	**61.9**	80.0	**53.6**	9.1	**92.9**	18.3	5.7	15.8	40.4	30.7	9.2	3.3	13.7	11.6	22.8
		EdgeNet-MF(Ours)	60.7	80.3	52.8	**11.0**	92.3	**20.5**	7.2	**16.3**	42.8	**32.8**	**10.5**	**6.0**	**15.7**	**11.8**	**24.3**
		EdgeNet-LF(Ours)	59.9	**80.5**	52.3	3.2	87.1	19.9	**8.6**	15.4	**43.5**	32.3	8.8	4.3	13.7	10.0	22.4
NYU	d	SSCNet[148]	57.0	**94.5**	55.1	15.1	94.7	24.4	0.0	**12.6**	32.1	35.0	**13.0**	**7.8**	27.1	10.1	24.7
	d+e	EdgeNet-EF(Ours)	**78.1**	65.1	55.1	**21.8**	**95.0**	27.3	**8.4**	6.8	**53.1**	38.6	7.5	0.0	30.4	**13.3**	27.5
		EdgeNet-MF(Ours)	76.0	68.3	**56.1**	17.9	94.0	**27.8**	2.1	9.5	51.8	**44.3**	9.4	3.6	**32.5**	12.7	**27.8**
		EdgeNet-LF(Ours)	75.5	67.5	55.4	19.8	94.9	24.4	5.7	7.2	50.3	38.8	10.0	0.0	33.2	12.2	27.0
SUNCG + NYU	d	SSCNet[148]	59.3	92.9	56.6	15.1	94.6	24.7	10.8	17.3	53.2	45.9	15.9	13.9	31.1	12.6	30.5
		DCRF[167]	-	-	-	18.1	92.6	27.1	10.8	18.8	54.3	47.9	17.1	15.1	34.7	13.0	31.8
		VVNetR-120[51]	69.8	83.1	61.1	19.3	94.8	28.0	12.2	19.6	57.0	50.5	17.6	11.9	35.6	15.3	32.9
	d+c	Guedes *et al.* [49]	-	-	56.6	-	-	-	-	-	-	-	-	-	-	-	30.5
	d+s	Garbade *et al.* [45]	69.5	82.7	**60.7**	12.9	92.5	25.3	20.1	16.1	56.3	43.4	17.2	10.4	33.0	14.3	31.0
		SNetFuse[96]	67.6	**85.9**	**60.7**	22.2	91.0	28.6	**18.2**	19.2	56.2	51.2	16.2	12.2	37.0	17.4	33.6
		TNetFuse[96]	67.3	85.8	**60.7**	17.3	92.1	28.0	16.6	19.3	**57.5**	**53.8**	**17.7**	**18.5**	**38.4**	**18.9**	**34.4**
	d+e	EdgeNet-EF(Ours)	77.0	70.0	57.9	16.3	**95.0**	27.9	14.2	17.9	55.4	50.8	16.5	6.8	37.3	15.3	32.1
		EdgeNet-MF(Ours)	**79.1**	66.6	56.7	**22.4**	**95.0**	**29.7**	15.5	**20.9**	54.1	53.0	15.6	14.9	35.0	14.8	33.7
		EdgeNet-LF(Ours)	77.6	69.5	57.9	20.6	94.9	29.5	9.8	18.1	56.2	50.5	11.4	5.2	35.9	15.3	31.6

Table 5.2: **Semantic scene completion results on NYUDv2 test set**. Column input indicates the type of input: d=depth only; d+s=depth and segmentation maps; d+e=depth and edges. Column train indicates the dataset used for training the models. SUNCG + NYU means trained on SUNCG and fine-tuned on NYUDv2.

iment are shown as SSCNet*. We observed a large improvement in SSC scores just using our pipeline.

After isolating the effect of our training protocol, we investigate the effect of our encoder-decoder architecture, with dilated ResNet modules. To accomplish this, we used EdgeNet-D, that is the Ednet architecture fed only with depth, without edges. Once again we observed a high level of improvement, compared to SSCNet*. EdgeNet-D also got the best overall scores amongst the depth-only approaches. The next experiment evaluates the effect of adding edges to an existing depth-only architecture. We took SSCNet and fed it with both depth and edges after F-TSDF encoding (SSCNet-E). We observed improvements compared to SSCNet* on overall scores and especially on hard-to-detect classes like TVs and objects.

Finally, we evaluate the benefits of adding Edges to our architecture in three fusion schemes: EdgeNet-EF, EdgeNet-MF, and EdgeNet-LF. Performance gains from EdgeNet-D show, once again, that adding edges is useful. A discussion about fusion schemes is provided in Section 5.3.

We also compare EdgeNet results to previous approaches. Overall, our proposed solutions achieve the best performance by a large margin. EdgeNet-EF achieves the best average scores, while EdgeNet-MF achieves the best score in some classes. EdgeNet-EF surpassed VVNetR120, the best previous approach on average SSC, by 3.3%. As expected, the highest improvements are observed in hard-to-detect classes, like objects and TVs. Although SUNCG is synthetic, evaluation on this dataset is quite important because of the poor quality of the ground truth in NYU, which impacts negatively more accurate models like EdgeNet.

5.2.5 Results on NYUDv2

Table 5.2 shows the results of EdgeNet on the NYUDv2 dataset and compares it with previous approaches. We compare results for models trained only on synthetic data, only on NYUDv2, and on both synthetic and NYUDv2 using fine-tuning.

On SUNCG-only and NYUDv2-only training scenarios, EdgeNet-MF achieved the best overall scores on Scene Completion and Semantic Scene Completion. However, in the SUNCG+NYU training scenario, TNetFuse presented the best result. EdgeNet-MF achieved the best scores on structural elements and chairs. It is worth mentioning that the NYUDv2 dataset has severe ground truth errors and misalignment, so results are not precise, and small differences in results may be questioned (see Section 5.2.6).

Despite these problems on NYU ground truth, EdgeNet achieves state-of-the-art level results with a much simpler and more computationally efficient training pipeline. EdgeNet is an end-to-end approach, and its memory consumption allows a batch size of 3 samples in a GTX 1080TI GPU, while TNetFuse requires a complex two-step training procedure and uses a batch size of only 1 sample, in the same GPU.

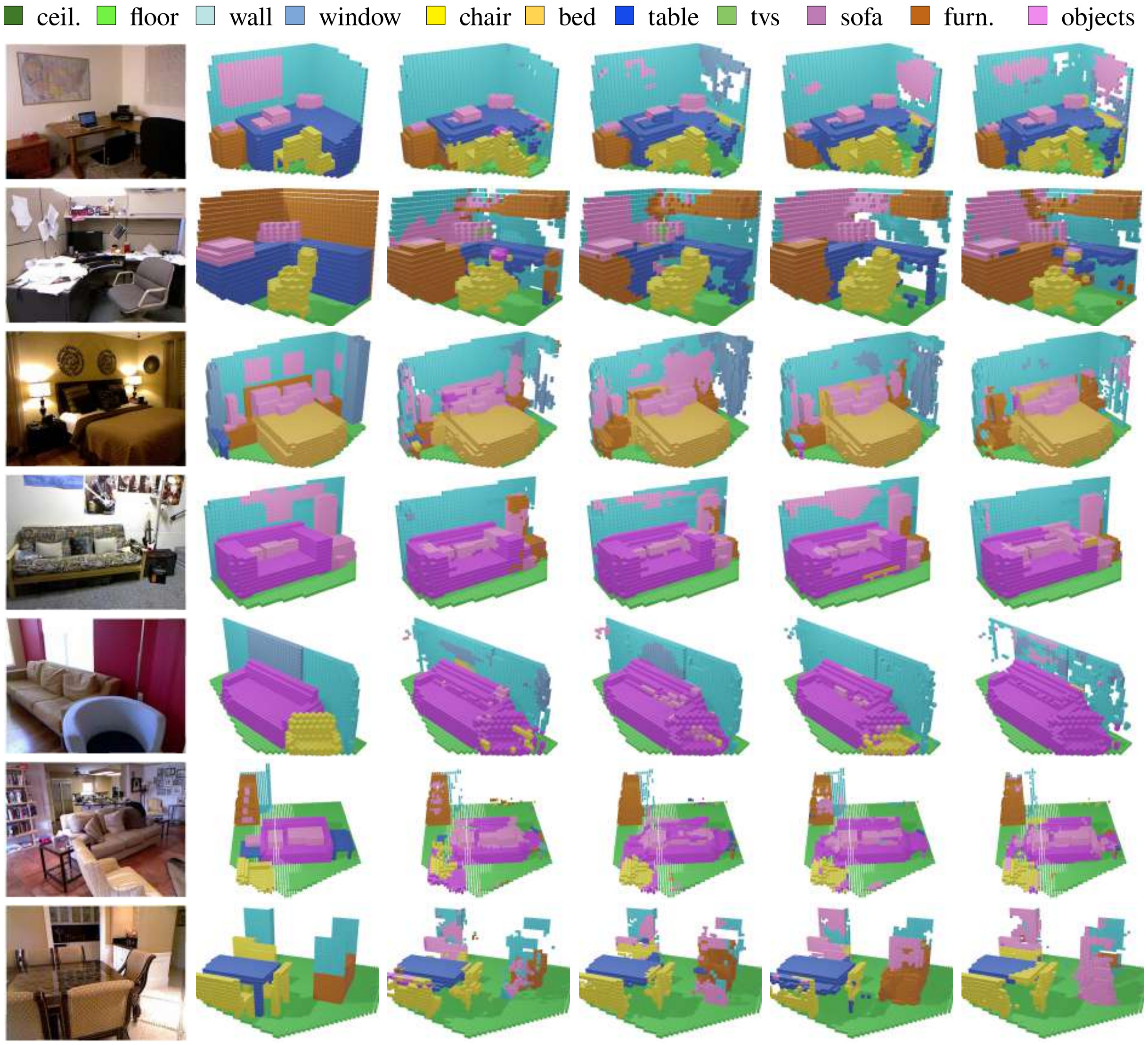
5.2.6 Qualitative Results

Qualitative results on NYUDv2 are shown in Figure 5.4. Models used to generate the inferences were trained on SUNCG and fine-tuned on NYUDv2. We compare the results of SSCNet* to our three models. It is visually perceptible that EdgeNet presents more accurate results.

In the first row of images of Figure 5.4, note the presence of a picture and a window, and observe that the ground truth misses the window. SCCNet* did not detect the picture and the window while EdgeNet-MF detects the window and some parts of the picture. This ground truth mislabelling affects negatively the performance of EdgeNet.

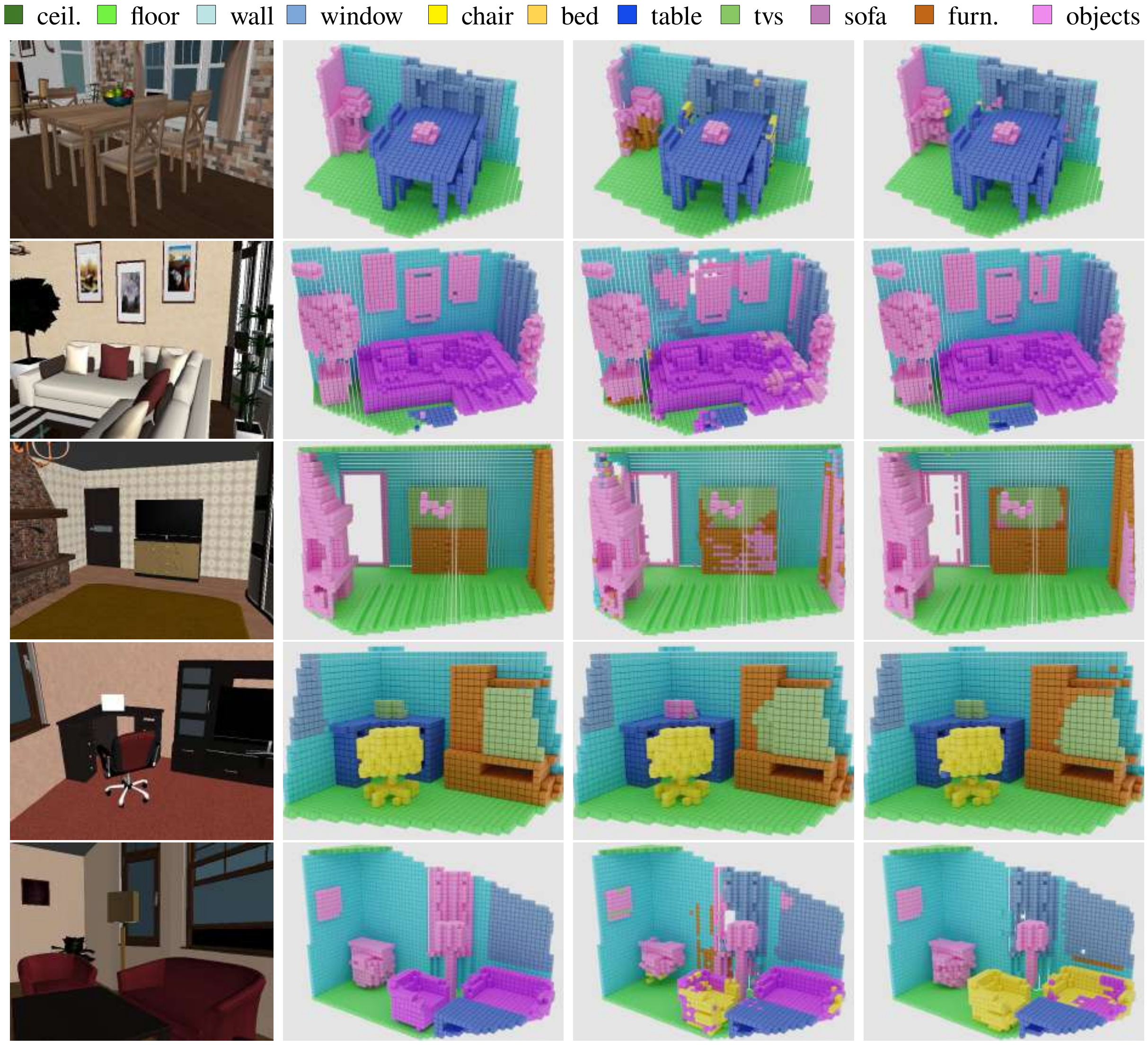
The second row of Figure 5.4 also depicts some problems related to Ground Truth annotations on the NYUDv2 dataset. Note that neither the papers fixed on the wall nor the shelf appears in the Ground Truth. All models captured the shelf, but only EdgetNet inferred the presence of objects fixed on the wall. When quantitative results are computed, these ground truth annotation flaws unfairly benefit the less precise models and harm more precise models like ours.

Figure 5.5 shows the qualitative results of our models on SUNCG. We compare SSCNET* to our Mid Fusion model EdgeNet-MF, as it presented better generalization capabilities from SUNCG to NYU. As SUNCG is a much larger dataset than NYU and does not have the noise and depth flaws of scenes captured with Kinect sensors, the results are remarkably better. Rows 1 to 3 show how EdgeNet presents much more accurate predictions than SSCNet. Note in the second row, how EdgeNet almost reached a perfect score, while SSCNet presented several points of errors. Row 4 exemplifies how EdgeNet is capable of correctly classifying hard-to-detect objects. Note that SSCNet labeled as "object" the small TV on the table while EdgeNet correctly classified it as "tv". Also, note that EdgeNet delimited the border of the large TV much better than SSCNet. Although it is not as common as in NYU, SunCG also presents some ground truth errors, as can be seen in row 5. Note that the window behind the lamp is labeled as "object" in the ground truth. Also, note that there is a chair that is incorrectly labeled as "sofa". EdgeNet correctly classified both objects and was penalized by ground truth errors.



(a) RGB image (b) G.T. (c) SSCNet* (d) EdgeNet-EF (e) EdgeNet-MF (f) EdgeNet-LF

Figure 5.4: **Qualitative Results on NYUDv2**. We compare EdgeNet results using SSCNet* as a baseline on NYUDv2. Overall, EdgeNet gives more accurate voxel predictions, especially for hard-to-detect classes (best viewed in color).



(a) RGB image (b) Ground Truth (c) SSCNet* (d) EdgeNet-MF (ours)

Figure 5.5: **Qualitative results on SUNCG**. Here we compare the results of SSCNet* (our implementation of Song et al.'s method [24], with the proposed training strategies) with EdgeNet-MF (our mid-level fusion method that combines depth and RGB edge information). Overall, EdgeNet-MF gives more accurate voxel predictions (best viewed in color).

5.3 Discussion

In this section, we discuss key aspects and contributions of our proposed approach.

5.3.1 Has the new training pipeline any influence over results?

We compared the results originally achieved by SSCNet to the results of the version of it trained with our pipeline (SSCNet*). On SUNCG we observed an improvement of almost 20% on semantic scene completion and more than 10% on scene completion. Besides the improvements in model performance, the more computationally efficient pipeline also contributed to reducing training time from 7 days to 4 days when training on SUNCG and from 30 hours to 6 hours when training on NYUDv2, with a batch of size 3, whereas the original framework only allowed a batch size of 1 sample on an NVIDIA® GTX 1080Ti (which has 11GB of memory). Besides reducing training time, larger batch sizes enhance training stability, acting as a regularizer [142].

5.3.2 Is a deeper U-shaped CNN with dilated ResNet modules helpful?

We investigated the effects of our architecture with and without aggregating edges. In both scenarios, our proposed architecture outperformed the shallower network, confirming that our network architecture is helpful.

5.3.3 Is aggregating edges helpful? May Other 3D CNN architectures benefit from aggregating edges?

We compared the original SSCNet architecture trained with our pipeline to a modified version of it that aggregates edges encoded with F-TSDF (SSCNet-E). SSCNet-E presented better results on SUNCG, demonstrating that the aggregation of edge information is helpful. We also observed improvements using a deeper depth-only network (EdgeNet-D). These experiments demonstrate that the proposed 3D volumetric representation of color edges can improve the performance of other previous depth-only approaches.

5.3.4 What is the best fusion strategy?

The later the fusion, the higher the memory requirement, due to the duplication of convolutional branches. Higher memory may imply in smaller batch sizes which may negatively impact learning. Liu *et al.* [96] observed better results using late fusion, but they faced the problem of higher memory consumption. Our choice was to fix the memory footprint, reducing the number of channels of duplicated branches without compromising the training time and stability. However, very late fusion schemes may suffer from accuracy degradation due to the reduced number of parameters in deeper layers. Taking those aspects into account, we found that a mid-level fusion strategy works and generalizes better for EdgeNet considering both synthetic and real datasets.

5.3.5 How does EdgeNet compare to other RGB + depth approaches?

We have compared EdgeNet with other RGB + depth approaches on SUNCG (Table 5.1) and NYUDv2 (Table 5.2. On SUNCG, EdgeNet versions surpassed previous approaches by a large margin. On NYU, EdgeNet got similar results as the solutions from TNetFuse [96], with less than a 1% difference. It is important to observe that NYU ground truth annotations are not precise, which impacts negatively more accurate models. Another aspect that is worth mentioning is that TNetFuse needs a complex and less computationally efficient two-step training protocol, while EdgeNet and the previous depth-only solutions cited in this paper are end-to-end networks, with a much simpler and more efficient training pipeline.

5.4 Chapter Summary

In this chapter, we presented a new approach to fuse depth and color into a CNN for semantic scene completion. We introduced the use of F-TSDF encoded 3D projected edges extracted from RGB images. We also presented a new end-to-end network architecture capable of properly aggregating edges and depth, extracting useful information from both sources, without requiring previous 2D semantic segmentation training as is the case of previous approaches that combine depth and color. Experiments with alternate models showed that both aggregating edges and the new proposed architecture have a positive impact on semantic scene completion, especially for hard-to-detect objects. Qualitative results show significant improvement for objects such as pictures, which cannot be differentiated by depth only. On SUNCG, we have achieved the best overall result, and on NYU, we have achieved the state-of-the-art results of other approaches that use a more complex training protocol.

Experiments showed that our proposed approach of aggregating Edges may be applied to other existing solutions, opening room for further improvements.

We also developed a lightweight training pipeline for the task, which reduced the memory footprint in comparison to other solutions and reduced the training time on SUNCG from 7 to 4 days and on NYUDv2 from 30 to 6 hours. All the code and weights necessary to reproduce the results presented in this chapter are publicly available in our GitLab repository: `https://gitlab.com/UnBVision/edgenet-v2`.

Chapter 6

Multimodal 3D SSC with 2D Segmentation Priors and Data Augmentation

The main motivation for the edge-based approach introduced by EdgeNet (Chapter 5) was the ability to use the RGB information encoded with F-TSDF to avoid the sparsity problem, faced by some previous works like Guedes *et al.* [49] (refer to Sections 2.4.2 and 3.3.1). By the time of the release of EdgeNet, another line of work was gaining momentum: exploiting RGB information by projecting inner features of 2D segmentation networks to 3D. Despite the good results observed at that time, the inner features of the 2D network do not use the full discrimination power of the last layers of the model. Prior to EdgeNet, one of the first works that tried to fully exploit the 2D segmentation network by using its 12-channel segmentation map output was Garbade *et al.* [45]. However, to feed the 3D network with the segmentation map, they had to apply an encoding strategy to reduce the number of input channels due to memory constraints. This reduction in the number of channels represents a significant loss of semantics.

In this Chapter, we bring back the idea of using a 2D segmentation network, but we tackle the semantics and input size problems mentioned before. Here we present a new approach for using the full 2D segmentation output, in the form of a 2D prior probabilities map (explained in section 6.2), as shown in Figure 6.1. The proposed solution explores multiple modes of the input in a new, semantic-rich encoding strategy from 2D networks. The solution uses 2D prior probabilities from a bimodal 2D segmentation network as semantic guidance to the depth map's structural data. The proposed multimodal 3D network, *SPAwN*, uses a new memory-saving batch-normalized dimensional decomposition residual building block (BN-DDR) and can be trained on a single 10GB GPU with a 4-scene mini-batch.

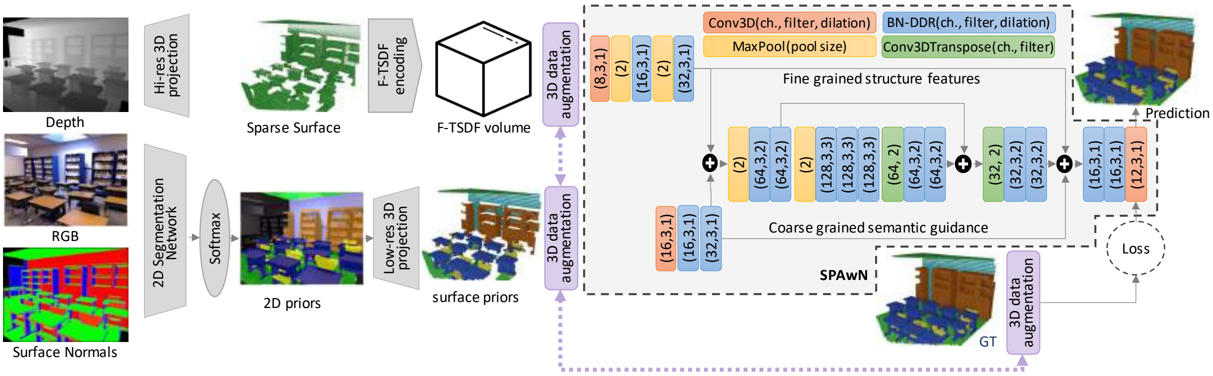


Figure 6.1: **Overview of our solution**. Our system comprises SPAwN, a 3D CNN that uses 2D priors as semantic guidance, and a novel 3D data augmentation approach for regularization and overfitting reduction. The 2D segmentation network is multimodal, combining RGB and surface normals. (Best viewed in color.)

To overcome the limitations imposed by the lack of sizeable real-world datasets, we are the first to apply 3D data augmentation for the SSC task. Data augmentation is widely used in the training of 2D deep CNNs [77, 60] and its goal is to reduce overfitting by artificially increasing the variety of samples in the training dataset using transformations like flipping, cropping, rotation and color transforms. However, those transformations can not naïvely be used in 3D applications like semantic Scene completion because of the difference in the number of dimensions of the input (2D) and output (3D). Our approach is to apply data augmentation to inner 3D volumes of the solution with three fast 3D transformations in voxel space that preserve the main characteristics of the scene. Our proposed data augmentation approach reduces overfitting and achieves unprecedented levels of semantic completion when compared to previous works of similar memory footprint and complexity.

We evaluated our contributions with and without pretraining on synthetic data and observed that our method surpasses, by far, all previous state-of-the-art results in both scenarios. We demonstrate the benefits of the proposed architecture and the data augmentation approach separately, with several experiments in a comprehensive and reproducible ablation study. Regarding the proposed augmentation scheme, we evaluate it for training (regular data augmentation) and testing (test-time data augmentation).

The contributions presented in this Chapter are listed below.

- *SPAwN*, a novel lightweight multimodal 3D SSC CNN architecture that uses 2D prior probabilities from a 2D segmentation network. These priors are used as semantic guidance to the structural data from the depth part of the RGB-D input. This architecture can be efficiently trained on a single 10GB GPU and achieves state-of-the-art results on both real and synthetic data.

- *BN-DDR*, a memory-saving batch-normalized dimensional decomposition residual building block for 3D CNNs. It preserves previous approaches' regularization characteristics while consuming much less memory during training.
- We are the first to apply a data augmentation technique for 3D semantic scene completion. Our method uses three 3D data transformations which operate on batches directly in GPU tensors.

The content of this Chapter was mainly extracted from our paper **Data Augmented 3D Semantic Scene Completion with 2D Segmentation Priors** [35] which was published in the proceedings of the *IEEE/CVF Winter Conference on Applications of Computer Vision* **(WACV 2022)**.

6.1 Evolution of the Field since EdgeNet

EdgeNet [34] introduced the use of RGB edges to capture details of the RGB-D images not visible on the depth map. After EdgeNet, AMFNet [90] explored an alternate mode of the input, namely HHA (horizontal disparity, height above ground, and angle with gravity) that can be generated from the depth map, alongside the RGB image in a bimodal network. However, the boost in the results was not expressive.

On the other hand, two works achieved better results: CCPNet [168] introduced multi-scale context information fusion, and ForkNet [156] introduced the use of multiple separate generators. After CCPNet and ForkNet, Sketch-Aware [23] enhanced the idea of using edges introduced by EdgeNet, combining a Conditional Variational Autoencoder (CVAE) that generates the sketch of the scene with semi-supervision from depth maps. By the time of its release, Sketch-Aware achieved very impressive results establishing new state-of-the-art scores on most benchmark datasets.

All previous methods, including EdgeNet and the work presented in this Chapter, use a straightforward training pipeline, where the input flows through the network in a single direction, without any loops. In 2021, SISNet [15] introduced the scene-instance-scene pipeline, which includes a sequence of semantic segmentation and instance completion networks. This pipeline is iteratively executed multiple times, surpassing Sketch-Aware results at the cost of requiring much more computational power.

In this Chapter, we present SPAwN, a multimodal, lightweight, and straightforward solution that, like some other methods, explores 2D segmentation. However, we propose a completely novel way of extracting knowledge from the RGB-D channels. We go further exploring new modes of the input in a trimodal network, that uses depth, RGB, and surface normals, as seen in figure 6.1. Our results represent a significant improvement

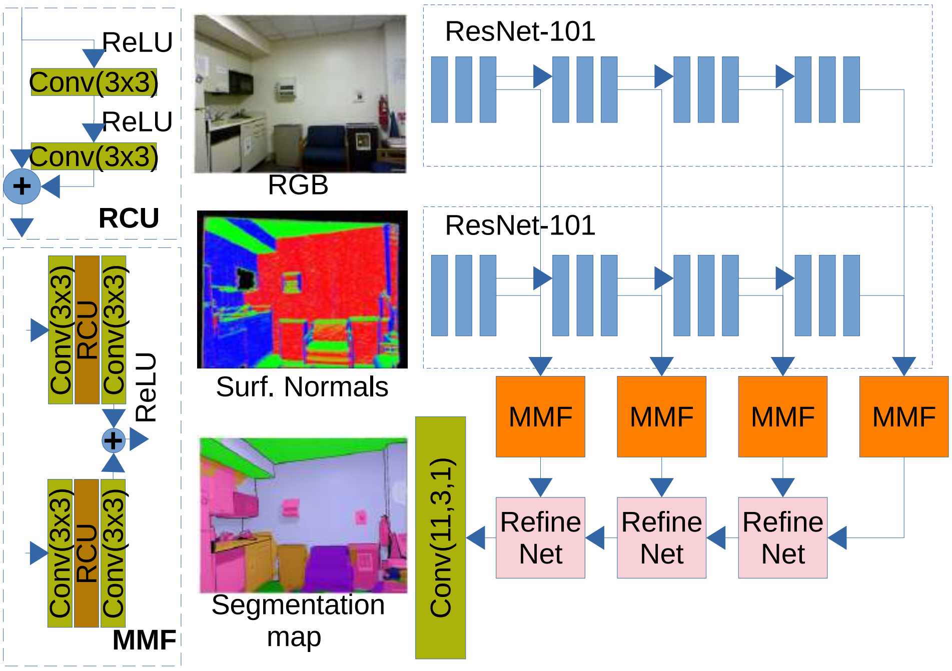


Figure 6.2: **2D bimodal segmentation network architecture.** The Residual Convolution Unit (RCU) and the RefineNet module were first defined in [92]. Here, we use a simplified MMF block [113]. (Best viewed in color.)

over Sketch-Aware and are comparable with the much more expensive iterative SISNet solution. We are also the first to explore Data Augmentation in the SSC task.

6.2 Proposed Solution

The overall multimodal solution is shown in Figure 6.1. Initially, we feed two modes of the input RGB-D image (RGB and surface normals) into a 2D segmentation CNN. Then, we submit the output of the 2D network to a Softmax function and obtain the prior probabilities that will be further projected to a low-resolution 3D voxel volume. The depth map, a third input mode, is projected to a high-resolution 3D volume and encoded with F-TSDF [148]. Data augmentation is applied directly to the 3D volumes, including ground truth, before feeding our SPAwN CNN. The input branches of SPAwN match the scale and the volumes are fused with an early/late fusion network to produce the final predictions.

6.2.1 2D Segmentation Network

To acquire high-quality 2D priors while keeping the memory footprint low, we crafted a new architecture based on RDFNet [113]. We use a bimodal encoder-decoder 2D RGB-D segmentation network with two ImageNet pre-trained ResNet-101 [60] backbones, one for each input mode, as presented in Figure 6.2[1]. The main adjustments to the original RDFNet are simplifications to the MMF module reducing the number of convolution layers; the usage of 3 RefineNet [92] modules, instead of 4; and the modification of the number of channels of the last layer since we need a classifier for 11 classes (in 2D, the empty or void class is ignored and the original RDFNet was trained for 40 classes).

This customized RDFNet takes two input images: the color channels of the RGB-D input, as in the original version, and surface normals, instead of the HHA encoded image. The surface normals are obtained from the depth map after aligning the scene to the Cartesian axes following the Manhattan assumption, as in [140]. Each axis is mapped to one RGB channel, and the absolute surface normal values are normalized from 0 to 255. The orientation of the surface normals is not encoded in our representation. Our goal with this simplified 2D network is only to test the hypothesis that 2D segmentation priors would improve the overall result. More details of the 2D network are provided in Appendix A.

6.2.2 SPAwN Semantic Scene Completion Network

Our Segmentation Priors Aware Network (*SPAwN*) is a novel 3D CNN architecture that uses 2D prior probabilities from a semantic segmentation network to guide the depth map's structural information.

Depth map projection and encoding. We use the same projection and encoding strategy described in Section 5.1.3. The dimensions of the 3D space are also the same. Examples of a projected depth map are shown in Figure 6.5b.

2D prior probabilities projection and ensemble. Projecting the output of the 2D network to 3D is a delicate task. One could be tempted to project the features to 3D at the same resolution as the structural volume. However, it would lead to 11 volumes of $240 \times 144 \times 240$ voxels, consuming too much GPU memory. Our solution, instead, consists in projecting the data from 2D at a lower resolution. We use $60 \times 36 \times 60$, the same as the output of the network. We compensate for the reduction in details, which come from the structural branch, with a boost in accuracy during downsampling by using a simple yet effective classifier ensemble method known as the "sum rule" [74], as follows. Firstly we obtain the probabilities for the 11-class output by applying a Softmax function, resulting in 11 planes with the exact resolution as the input image. Then, we project each pixel

[1] In Chapter 4, we used a different architecture for 2D semantic segmentation. We upgraded the network here for two main reasons: the experiments on Chapter 4 were the first we conducted in our research project, so the architecture got dated, and here we need an architecture capable of fusing two input modes (bimodal). Besides that, in Chapter 4 we were more interested in assessing the benefits of the FCN architecture, compared to more traditional approaches, and the gains we could observe using Domain Adaptation and Semi-supervised approaches. Those experiments were extremely useful and guided us to the path that lead us to this point of the work.

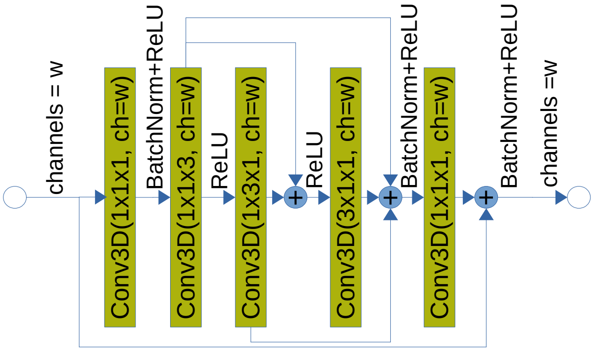


Figure 6.3: **Proposed *BN-DDR* module**. Our arrangement presents good discrimination and regularization properties while keeping memory consumption manageable.

to 3D at low resolution (voxel size = 0,08m). When more than one pixel falls into the same voxel, we sum the probabilities of each class and, to normalize the resulting priors, we divide the probabilities by the number of pixels that fell into the voxel.

The previous approach only provides information for the surface voxels. To provide information to non-surface voxels, we add an extra channel for the empty class, and for all non-surface voxels, we set the probability 1 for voxels belonging to the class “empty” and 0 to the other 11 classes. An example of projected prior volume compared to the ground truth is shown in Figures 6.5c and 6.5f.

Batch-normalized DDR. The fundamental building block of our 3D network is a batch normalized version of the Dimensional Decomposition Residual (DDR) [88] block, named *BN-DDR*. The DDR block is a lightweight alternative to the ResNet block [60] to avoid the vanishing gradient problem in deep neural networks. Our preliminary experiments showed that adding batch normalization layers to the DDR block produces better results. However, adding a batch normalization layer after each convolutional layer of the block as in [90], consumes too much memory, making the network difficult to train with larger mini-batches, reducing the overall benefits of the batch normalization, and making the training slower. Our solution eliminates the batch normalization layers between the dimensional decomposition layers resulting in our proposed *BN-DDR* block as presented in Figure 6.3. Due to its reduced memory footprint, we keep the same number of channels of the outer layers in the inner layers, while keeping the mini-batch with 4 scenes. For example, previous DDR-based solutions like DDR-Net [88] and AMFNet [90] use a batch size of 2 and 1, respectively. SketchAware [23] uses a batch size of 4, as well, however, it requires 2 11GB GPUs for training.

Fusing structure and semantics. We use an early/late fusion strategy to fuse the detailed structure information from the F-TSDF encoded high-resolution volume to the semantic information from the surface prior volume, as presented in Figure 5.3. The two initial branches are used to match the resolution and number of channels of both inputs. Then, both signals are early fused in an encoder-decoder 3D CNN with a skip connection in the mid-resolution stage, inspired by the U-Net architecture [120]. This fusion helps to preserve the details of the higher resolution level. The outer skip connections provide additional structural and semantic guidance, and the output branch performs the final late fusion and classification into the desired number of classes.

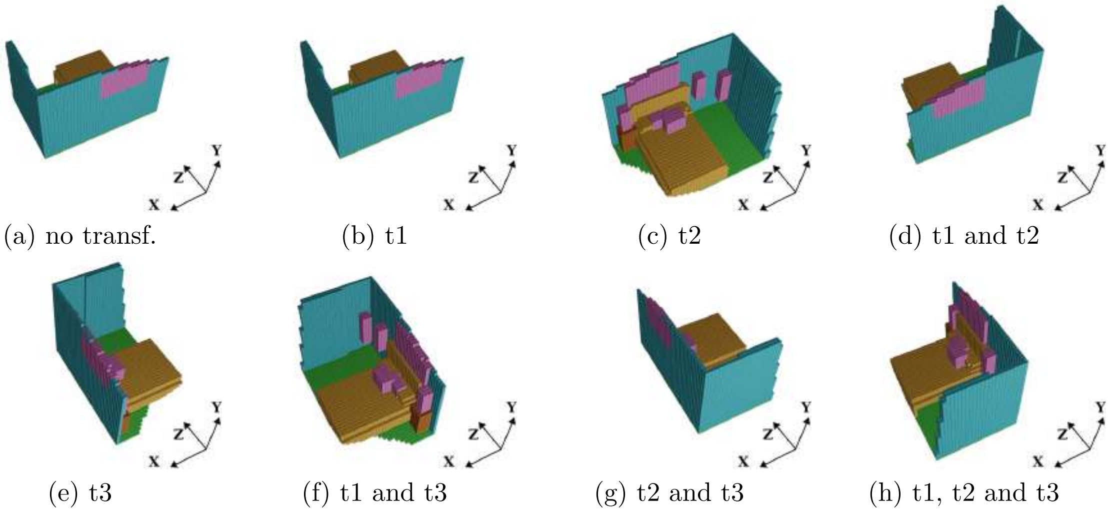
Data balancing and loss function. There are two main sources of data unbalancing in the 3D SSC problem: the first is the unbalance between occluded empty and occupied voxels; the second is the unbalance between the several classes. In this work, we face both unbalances in our loss functions. For the first one, we follow [34] and, for each training mini-batch, we randomly sample the occluded voxels to balance occupied and empty voxels, while ignoring empty voxels in the visible space. For the class unbalance, we use a class-weighted cross-entropy loss function, where the weight $w_c = 2$ for less frequent classes like TVs, objects, tables, and chairs, and $w_c = 1$ for all other classes. Being V the set of voxels of the mini-batch (lmb) selected for evaluation, $v \in V$, n the number of classes, $y_{v,c}$ a binary indicator if the voxel v belongs to class c and $P_{v,c}$ the predicted probability of the voxel v related to class c, the loss L is given by equation 6.1.

$$L = \frac{1}{|V|} \sum_{v} \left(- \frac{\sum_{c=1}^{n} [w_c \cdot y_{v,c} \cdot \log(P_{v,c})]}{\sum_{c=1}^{n} w_c} \right) \tag{6.1}$$

6.2.3 Data Augmentation for SSC

Data augmentation is a regularization technique that is vastly used in for training neural networks for 2D computer vision [77, 60, 92, 64]. Regularization and consequent reduction in overfitting are achieved by artificially enlarging the training dataset by randomly applying transformations like flipping and cropping to the input images [77]. More recently introduced data augmentation strategies include blocking-based approaches like Blockout [107] and Random Erasing [172].

Applying the transformations to the input images is enough for image classification tasks since they do not affect the corresponding labels. However, for semantic segmentation tasks, if the transformations affect objects' position in the input image, it is necessary to apply the same transformations to the ground truth maps to preserve the pixel-to-pixel correspondence [77]. The need for applying the same transformations to the input and the ground truth is an extra difficulty for using data augmentation in RGB-D to voxel segmentation tasks like semantic scene completion. Keeping the correspondence between the input and ground truth representation is necessary, but there is no direct pixel-to-voxel mapping from the input to the output. When the solution uses some kind of Truncated Sign Distance Function like TSDF or F-TSDF [148] the problem is exacerbated because



(a) no transf.
(b) t1
(c) t2
(d) t1 and t2
(e) t3
(f) t1 and t3
(g) t2 and t3
(h) t1, t2 and t3

Figure 6.4: **All augmented volumes generated from a single scene**. Each image caption indicates which transformations were applied. The transformations are: t1 (X-axis flipping), t2 (Z-axis flipping) and t3 (X ↔ Z axes swapping).

changing one pixel of the input would propagate in 3D space through the TSDF-encoded volume in a pyramid shape, affecting a large region.

To overcome the lack of pixel-to-voxel correspondence in those applications, we introduce the use of data augmentation applied directly to the projected 3D volumes of the SSC solutions. We randomly rotate the scene in 90°steps and randomly flip along the horizontal axes (X and Z), to avoid generating upside-down scenes. This precaution is usually taken in 2D domains. For instance, vertical flipping is usually avoided in 2D. We achieve this augmentation with 3 simple and fast 3D transformations. The first two transformations are random X-axis flipping and random Z-axis flipping. The third transformation is random X ↔ Z axes swapping. The use of this strategy during training is equivalent to augmenting eight times the size of the dataset. Figure 6.4 illustrates all possible augmented volumes from a single scene. All operations are done in the GPU tensors and can be easily executed in parallel, with almost no impact on training time. The proposed data augmentation strategy is mode and resolution agnostic, thus, it may be readily applied to multi-modal or to multi-resolution SSC setups. For instance, SPAwN 3D mini-batches contain the F-TSDF encoded high-resolution surface volume and the 12 channels of low-resolution priors.

Training-time data augmentation. During training, for each training mini-batch, we randomly choose to apply or not each one of the 3 transformations. The chosen transformations are then applied to the whole 3D input mini-batch at once and also to the ground truth.

Test-time data augmentation. We also evaluate the use of the transformations in test-time. In this case, for each test mini-batch, we apply all eight possible combinations of transformation to each input volume, generate the predicted volumes and, unlike in training time, we apply the inverse transformation to the output volumes instead of the same transformation. In this way, the eight output volumes share the same orientation as the original mini-batch. The aligned predictions are then ensembled. For that, we apply the “sum rule” [74], generating a single and more accurate output.

6.3 Experiments

6.3.1 Datasets

We executed our experiments on the three most important SSC benchmark datasets: NYUDv2 [140], NYUCAD [43], and SUNCG [148]. Refer to Section 3.5 for a detailed description.

SUNCG dataset neither includes the RGB images nor the surface normals, we render the RGB images using the provided camera positions for each snapshot as specified in [34] and extract the surface normals from the depth map.

6.3.2 Training Details

All models of the experiments conducted in this Chapter, including 2D networks, were optimized with SGD using the *one cycle learning rate policy* [141] with the maximum and minimum learning rates (LR) multipliers set to 25 and 1×10^{-4}, respectively, cosine annealing, 1×10^{-5} weight decay, and mini-batches with four scenes.

2D network training. 2D models were trained in two data-augmented stages. In the first stage, the ImageNet pretrained ResNet-101 backbone weights were frozen, and the base LR was set to 2×10^{-4}. In the second stage, the base LR was set to 8×10^{-5}, and all weights were unfrozen, but the ResNet backbones LR was set to 1/10 of the running LR. For NYUDv2 and NYUCAD, each stage comprises 150 epochs, and for SUNCG, 10 epochs. The data augmentation transforms were the conventional 2D random resize, random crop, and random horizontal flip.

3D network training. 3D models were trained in a single stage, with base LR set to 4×10^{-4}. Our 3D data augmentation strategy reduces overfitting and thus, allows us to train for more epochs. For NYUDv2 and NYUCAD, the models were trained for 80 epochs when not using data augmentation and for 120 epochs when augmented. As SUNCG is a very large dataset, the benefits of data augmentation are expected to be inexpressive, so we do not apply data augmentation on SUNCG and train for only 10 epochs.

Fine-tuning from SUNCG. When fine-tuning from SUNCG, both 2D and 3D models were first trained on SUNCG and then fine-tuned to the desired target dataset. The same protocols described previously were applied.

Metrics. We follow the previous works and report scores for the completion task and the semantic scene completion task. For completion, we report precision, recall, and Intersection over Union (IoU) considering the prediction of occupancy for each occluded voxel. For the semantic scene completion task, we report the IoU of each object class on both the observed and occluded voxels and the averaged result over all classes except the void class (mIoU). Voxels outside the view and visible empty voxels are ignored.

6.3.3 Ablation Study

We evaluate the importance of each aspect of the proposed solution with a comprehensive set of experiments using only real images from NYUDv2 without pretraining on synthetic images. Table 6.1 presents the progressive contribution of the proposed *BN-DDR* module, the use of the proposed class balancing strategy, and the data augmentation training and test approaches, considering one, two, and three input modes. All models were trained and evaluated under the same protocols except for the number of epochs when using data augmentation, as explained in the previous section.

We observed positive contributions for each of the evaluated aspects. Using RGB and surface normals as inputs consistently produced positive impacts. The model itself surpassed state-of-the-art results with regular not augmented training. Moreover, combining the proposed network with the data augmentation approaches enhances results to unprecedented levels.

Table 6.1 also evaluates the model's theoretical upper bound limit in an Oracle Test, supposing we have predicted perfect semantic 2D priors. To this matter, we replace the output of the 2D network with the 2D ground truth. The Oracle experiment shows there is still room for improvement by enhancing 2D predictions. Future works can exploit this.

6.3.4 Comparison to the State-of-the-Art

In tables 6.2, 6.3, and 6.4, for each training scenario, the best scores for straightforward solutions are presented in bold, while the second-best scores are underlined. We only show the best two or three competing models in each category. Further results are presented in Appendix A.

SUNCG. Table 6.2 shows the results on SUNCG synthetic dataset. As the SUNCG training set is large, the benefits of data augmentation are not expected to be significant. Therefore, we only evaluate the standard training approach. Our proposed 3D CNN

input modes	DDR type	class bal.	DA	TTDA	comp. IoU	SSC mIoU
depth	Regular	no	no	no	55.5	24.5
	BN-DDR	no	no	no	60.8	31.8
	BN-DDR	yes	no	no	60.8	32.2
depth rgb	Regular	no	no	no	60.9	38.6
	BN-DDR	no	no	no	63.0	41.0
	BN-DDR	yes	no	no	64.4	42.2
depth rgb sn	Regular	no	no	no	61.3	39.2
	BN-DDR	no	no	no	63.4	41.4
	BN-DDR	yes	no	no	63.8	43.4
	BN-DDR	yes	yes	no	65.7	47.7
	BN-DDR	yes	yes	yes	66.2	48.0
oracle test	*BN-DDR*	yes	no	no	76.7	67.9

Table 6.1: **Progressive impact of SPAwN components on NYUDv2.** No pretraining was performed. "sn" means surface normals, DA means data augmentation, and TTDA means test-time data augmentation.

model	pipeline type	scene completion			semantic scene completion (IoU, in percentages)											
		prec.	rec.	IoU	ceil.	floor	wall	win.	chair	bed	sofa	table	tvs	furn.	objs.	avg.
SISNet-BSN[15]	iterative	93.3	96.1	89.9	85.2	90.0	83.7	80.8	60.0	83.5	80.8	68.6	77.3	86.7	70.1	78.8
SISNet-DL3[15]		92.6	96.3	89.3	85.4	90.6	82.6	80.9	62.9	84.5	82.6	71.6	72.6	85.6	69.7	79.0
EdgeNet[34]	straight-forward	93.3	90.6	85.1	97.2	95.3	78.2	57.5,	51.4	80.7	74.1	54.5	52.6	70.3	60.1	70.2
ESSC[166]		92.6	90.4	84.5	96.6	83.7	74.9	59.0	55.1	83.3	78.0	61.5	47.4	73.5	62.9	70.5
CCPNet[168]		**98.2**	**96.8**	**91.4**	99.2	89.3	76.2	63.3	58.2	**86.1**	**82.6**	65.6	53.2	**76.8**	65.2	74.2
SPAwN (ours)		91.9	88.7	82.3	**99.3**	**96.1**	**84.4**	**75.1**	**59.2**	81.5	78.1	**67.3**	**80.1**	76.3	**70.4**	**78.9**

Table 6.2: **Results on SUNCG test set**. Our SPAwN semantic scene completion overall results surpass by far all known previous straight-forward solutions on SUNCG synthetic images, and are comparable to both SISNet models, even though they have a much higher parameter count and operate with a complext iterative pipeline for both training and inference.

surpassed CCPNet, the previous best straightforward solution, by a 6.3% margin (4.7p.p.) and got similar results to the iterative models.

NYUDv2. Table 6.3 presents the results on real images. We evaluated two training scenarios: training from scratch on NYUDv2 and training on SUNCG, then fine-tuning on NYUDv2. Data augmented *SPAwN* presented the best overall results in all scenarios. Without fine-tuning, the boost over SketchAware was 16.7% (6.9p.p.). With fine-tuning, the bost over CCPNet was 20.8% (8.6p.p.) and the result is comparable to the more expensive SISNet models.

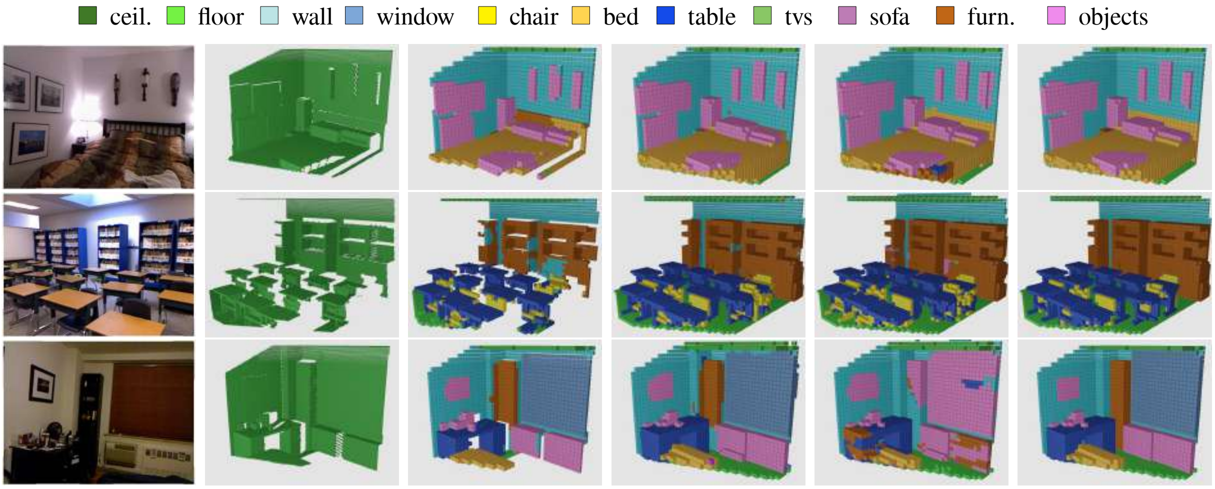
NYUCAD. Table 6.4 confirms our expectation of good results due to the better quality of the ground truth related to surface volume and 2D priors. The observed mIoU boost of our model over SketchAware and CCPNet, the best previous solution in each

model	pipeline type	train	scene compl.			semantic scene completion (IoU, in percentages)											
			prec.	rec.	IoU	ceil.	floor	wall	win.	chair	bed	sofa	table	tvs	furn.	objs.	avg.
SISNet-BSN[15]	iterative	NYU	90.7	84.6	77.8	53.9	93.2	51.3	38.0	38.7	65.0	56.3	37.8	25.9	51.3	36.0	49.8
SISNet-DL3[15]			92.1	83.8	78.2	54.7	93.8	53.2	41.9	43.6	66.2	61.4	38.1	29.8	53.9	40.3	52.4
TS3D[45]	straight-forward	NYU	-	-	60.0	9.7	93.4	25.5	21.0	17.4	55.9	49.2	17.0	27.5	39.4	19.3	34.1
SketchAware[23]			**85.0**	**81.6**	**71.3**	**43.1**	93.6	**40.5**	24.3	30.0	57.1	49.3	29.2	14.3	42.5	28.6	41.1
SPAwN (ours)			82.3	77.2	66.2	41.5	**94.3**	38.2	**30.3**	**41.0**	**70.6**	**57.7**	**29.7**	**40.9**	**49.2**	**34.6**	**48.0**
TNetFuse[96]	straight-forward	NYU + SUNCG	67.3	85.8	60.6	17.3	92.1	28.0	16.6	19.3	57.5	53.8	17.7	18.5	38.4	18.9	34.4
ForkNet[156]			-	-	63.4	36.2	93.8	29.2	18.9	17.7	61.6	52.9	23.3	19.5	45.4	20.0	37.1
CCPNet[168]			**91.3**	**92.6**	**82.4**	25.5	**98.5**	38.8	27.1	27.3	64.8	**58.4**	21.5	30.1	38.4	23.8	41.3
SPAwN (ours)			81.2	80.4	67.8	**44.2**	94.2	**40.9**	**33.5**	**42.5**	**69.3**	**58.4**	**32.4**	**44.3**	**53.4**	**36.3**	**49.9**

Table 6.3: **Results on NYUDv2 test set**. SUNCG + NYU means trained on SUNCG and fine-tuned on NYUDv2. Our SPAwN models hold the best and second-best overall semantic scene completion results for real-world images, on both training scenarios, when compared to previous straight-forward solutions.

model	pipeline type	train	scene compl.			semantic scene completion (IoU, in percentages)											
			prec.	rec.	IoU	ceil.	floor	wall	win.	chair	bed	sofa	table	tvs	furn.	objs.	avg.
SISNet-BSN[15]	iterative	NYUCAD	94.2	91.3	86.5	65.6	94.4	67.1	45.2	57.2	75.5	66.4	50.9	31.1	62.5	42.9	59.9
SISNet-DL3[15]			94.1	91.2	86.3	63.4	94.4	67.2	52.4	59.2	77.9	71.1	58.1	46.2	65.8	48.8	63.5
CCPNet[168]	straight-forward	NYUCAD	**91.3**	**92.6**	82.4	56.2	**96.6**	58.7	35.1	44.8	68.6	65.3	37.6	35.5	53.1	35.2	53.2
SketchAware[23]			90.6	92.2	**84.2**	59.7	94.3	**64.3**	32.6	51.7	72.0	68.7	45.9	19.0	60.5	38.5	55.2
SPAwN (ours)			84.5	87.8	75.6	**65.3**	94.7	61.9	**36.9**	**69.6**	**82.2**	**72.8**	**49.1**	**43.6**	**63.4**	**44.4**	**62.2**
SSCNet[148]	straight-forward	NYUCAD + SUNCG	75.4	**96.3**	73.2	32.5	92.6	40.2	8.9	40.0	60.0	62.5	34.0	9.4	49.2	26.5	40.0
CCPNet[168]			**93.4**	91.2	**85.1**	58.1	**95.1**	60.5	36.8	47.2	69.3	67.7	39.8	37.6	55.4	37.6	55.0
SPAwN (ours)			86.3	90.1	78.9	**77.6**	95.0	**68.0**	**38.1**	**67.9**	**82.2**	**77.1**	**56.8**	**50.0**	**65.7**	**46.5**	**65.9**

Table 6.4: **Results on NYUDCAD**. Our SPAwN models hold the best and second-best overall results on both training scenarios, when compared to previous straight-forward solutions. When fine-tuned from SUNCG, SPAwN surpasses both SISNet models, which are much more complex than ours.



(a) RGB (b) Visible surf. (c) 2D priors (d) SPAwN (e) SSCNet (f) GT

Figure 6.5: **SPAwN qualitative results on NYUCAD.** 2D segmentation priors projected to 3D provide good semantic guidance while SPAwN complete and refine the predictions, achieving results visually close to perfection. Compared to baseline SSCNet [148], results are much more accurate. (Best viewed in color).

training scenario, is 12.7% (7.0p.p.) and 19.8% (10.9p.p.), respectively. Our fine-tuned model even surpassed both expensive SISNet models.

6.3.5 Qualitative Analysis

Figure 6.5 presents a qualitative analysis on NYUCAD due to its better alignment between depth map and ground truth compared to NYUDv2, making it easier to perceive our predictions' high quality visually. We generated predictions with *SPAwN* trained on SUNCG and fine-tuned on NYUCAD, using both training-time and test-time data augmentation. We also qualitatively compare our results to a baseline SSCNet model, pretrained on SUNCG and fine-tuned to NYUCAD. This SSCNet model was trained with our one-cycle learning rate protocol and achieves better results than the presented in the original paper [148] (53.3% vs 40.0% avg. IoU, respectively). *SPAwN* overall results are perceptible better than SSCNet. In column (c), it is possible to see that our projection and ensemble methods provide first-rate priors to the visible surface, with minimal prediction errors. *SPAwN* fusion strategy can complete the predictions and fix errors from priors, achieving remarkable final results. Flat objects on flat surfaces are difficult to be detected by depth-only approaches like SSCNet. Notice in the third row of figure 6.5 that SSCNet was unable to identify the window, while *SPAwN* predicitions are almost perfect. The supplementary material presents qualitative results on the other datasets.

6.4 Chapter Summary

In this Chapter, we presented *SPAwN*, a novel 3D SSC network that explicitly fuses semantic priors with high-resolution structural information from depth maps. *SPAwN* uses as a fundamental building block a novel lightweight batch normalized DDR module with higher discrimination power than its predecessors. We also introduce the use of 3D data augmentation to the SSC task. The proposed data augmentation strategy is mode and resolution agnostic and may be applied to other SSC solutions. An ablation study with a comprehensive set of experiments shows the effectiveness of each one of our contributions. That study also includes an oracle test, which showed that the proposed solution can be further enhanced using better sources of semantic priors.

Data augmented *SPAwN* surpasses by far all previous state-of-the-art solutions with similar complexity in SSC benchmarks, in all training scenarios, achieving a boost of 19.8% (10.9p.p.) over the best previously reported result on real images. Compared to the recently introduced and much more expensive iterative solution, the improvement is 3.8% (2.4p.p.).

Supplementary graphs and data regarding all experiments are provided in Appendix A. All models and training code necessary to reproduce our results and the ablation experiments are publicly available[2] .

[2] Source code: `https://gitlab.com/UnBVision/spawn`

Chapter 7

Exployting unlabeled data to enhance SSC scores

A crucial difficulty in Semantic Scene Completion is the lack of fully labeled real-world 3D datasets which are large enough to train the current data-hungry deep 3D convolutional networks. Semi-Supervised Learning is an approach that aims to improve the performance of regular supervised learning methods by using information customarily associated with unsupervised training. Such methods are beneficial in situations where labeled data is expensive or challenging to obtain [39]. Many semi-supervised approaches have been recently proposed to perform computer vision tasks [93, 165], achieving good results in low-labeled data scenarios.

In this Chapter, we present S3P, a semi-supervised training procedure that exploits unlabeled 2D priors to tackle this problem. We were influenced by the Pseudo-Label method [85], a simple yet effective approach that uses the same model being trained to generate (pseudo-)labels for the available unlabeled data. Since its release, it has influenced many others [62, 126, 132, 169, 36] (See Chapter 4 for a detailed use case of pseudo-labels). In our case, instead of using the same model as the ground-truth generator, we used 2D priors and achieved better results. We apply S3P to our SPAwN network presented in Chapter 6 and demonstrate the efficacy of the training procedure with a comprehensive and reproducible ablation study.

7.1 Semi-Supervision via Segmentation Priors

Our **S**emi-**S**upervision via **S**egmentation **P**riors approach (S3P) is partially inspired by the Pseudo-Label method [85]. The original method trains the network in a supervised manner using labeled and unlabeled mini-batches simultaneously. The same model is used to generate predictions and “pseudo”-ground-truth for unlabeled data. The quality of the

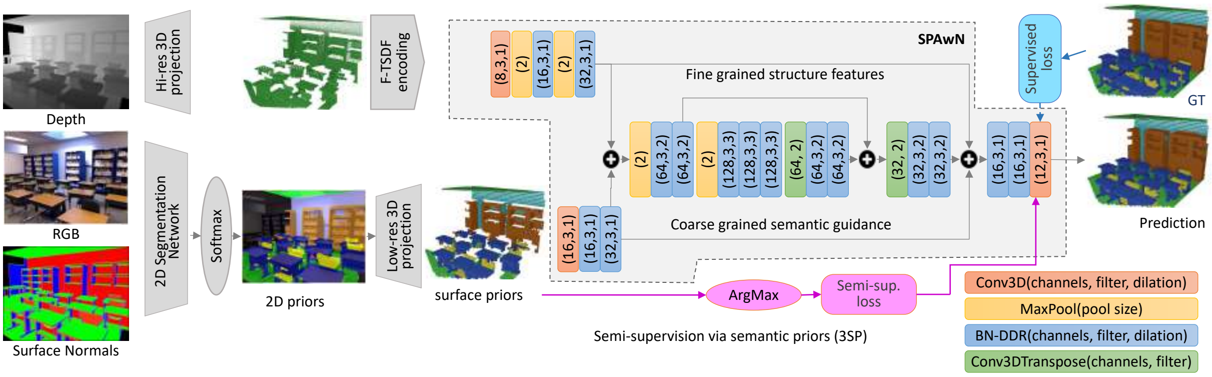


Figure 7.1: **Overview of our solution**. Our system comprises SPAwN, a 3D CNN that uses 2D priors as semantic guidance, and S3P, a semi-supervised training approach for regularization and overfitting prevention. The 2D segmentation network is multimodal, combining RGB and surface normals. (Best viewed in color).

generated pseudo-labels starts low but improves as the training iterates. This method has been largely used, and it has been shown that it acts as an entropy regularization strategy and favors low-density separation between classes [85].

In our case, instead of using the same model to generate ground truth for unlabeled mini-batches, we propose to use the pre-trained 2D segmentation network. The quality of generated labels starts high and does not change during training. For unlabeled data, the 2D network provides good predictions for the visible surfaces. The predicted probabilities for each class and each 2D pixel are projected to 3D, generating prior probabilities for the surface. The ground truth for unlabeled data is generated by picking the highest probability class for each surface voxel. Figure 7.1 shows our SPAwN adapted to S3P. For clarity, we did not include the data augmentation steps.

7.1.1 Semi-supervised training hyperparameters

We control the amount of unlabeled data used during training with the hyperparameter γ, which defines the number of labeled mini-batches performed before an unlabeled step. To constrain the unlabeled loss's effect, we apply a weight factor α to the loss before the backward propagation.

7.1.2 Supervised and Semi-supervised Loss Functions

Being V the set of voxels of the labeled mini-batch (lmb) selected for evaluation, $v \in V$, n the number of classes, $y_{v,c}$ a binary indicator if the voxel v belongs to class c and $P_{v,c}$ the predicted probability of the voxel v related to class c, the loss L_l for labeled data is given by equation 7.1.

$$L_l(\text{lmb}, P) = \frac{1}{|V|} \sum_{v \in V}^{|V|} \left(- \frac{\sum_{c=1}^{n} [w_c \cdot y_{v,c} \cdot \log(P_{v,c})]}{\sum_{c=1}^{n} w_c} \right) \tag{7.1}$$

The generated ground truth (pseudo-labels) only provides information for the visible surface. Therefore, when using unlabeled mini-batches, we only compute the loss for visible occupied voxels of the scene, preventing the degradation of the prediction of the occluded part of the scene. Being α the unlabeled weight factor hyperparameter and P the predicted probabilities, the semi-supervised loss L_u for an unlabeled mini-batch umb is similar to $\alpha L_l(\text{umb}, P)$ in Equation 7.1, though with a different voxel selection criterion. In L_u, the set of evaluated voxels V only contains the voxels of the visible scene surface in the mini-batch.

7.1.3 S3P Training Procedure.

During training, both labeled and unlabeled data are used. Each epoch comprises all labeled data ($\mathcal{D}_{\text{labeled}}$) randomly grouped into mini-batches. All unlabeled data are also grouped into mini-batches as a circular list ($\mathcal{D}_{\text{unlabeled}}$). The unlabeled mini-batches are used as required according to the γ hyperparameter as described in Algorithm 1.

Algorithm 1 S3P Semi-supervised training

```
for all epochs do
    $\mathcal{D}_{\text{labeled}} \leftarrow \text{shuffle}(\mathcal{D}_{\text{labeled}});\ i \leftarrow 0$
    for all lmb $\in \mathcal{D}_{\text{labeled}}$ do
        $P \leftarrow \text{net.forward(lmb)};\ i \leftarrow i + 1$
        net.backward($L_l(\text{lmb}, P)$)
        if $i \bmod \gamma = 0$ then
            umb $\leftarrow \mathcal{D}_{\text{unlabeled}}$.pull()
            $P \leftarrow$ net.forward(umb)
            net.backward($\alpha L_u(\text{umb}, P)$)
        end if
    end for
end for
```

7.2 Experiments

As in Chapter 6, we executed our experiments on the three most important SSC benchmark datasets: NYUDv2 [140], NYUCAD [43] and SUNCG [148]. Refer to subsection 3.5 for detailed information on the datasets. As with most of the previous works, we evaluate our models using the Intersection over Union (IoU) of each object class on both the observed and occluded voxels and the averaged result (mIoU). Voxels outside the view and visible empty voxels are ignored.

7.2.1 Training Details

All models of the experiments presented in this Chapter were optimized with SGD using the *one cycle learning rate policy* [141] with the maximum and minimum learning rates (LR) multipliers set to 25 and 1×10^{-4}, respectively, cosine annealing, 1×10^{-5} weight decay, and mini-batches with four scenes. No model selection was done using the test/validation sets to avoid biased results. All results are reported using the model after the last epoch of training, even though better results had been observed after previous epochs.

2D network training. 2D models were trained in two data-augmented stages. In the first stage, the ImageNet pretrained ResNet-101 backbones weights were frozen, and the base LR was set to 2×10^{-4}. In the second stage, the base LR was set to 8×10^{-5}, and all weights were unfrozen, but the ResNet backbones LR was set to 1/10 of the running LR. For NYUDv2 and NYUCAD, each stage comprises 150 epochs, and for SUNCG, 10 epochs. The data augmentation transforms were *random resize, random crop, and random horizontal flip.*

3D network training. 3D models were trained in a single stage, with base LR set to 4×10^{-4}. For NYUDv2 and NYUCAD, the models were trained for 80 epochs, and for SUNCG, 10 epochs.

Fine tuning from SUNCG. When fine-tuning from SUNCG, both 2D and 3D models were first trained on SUNCG and then fine-tuned to the desired target dataset. The same protocols described previously were applied.

7.2.2 Ablation Study

We evaluate the contribution of the semi-supervised training approach *S3P* over several training and architectural scenarios using only real images from NYUDv2 without pretraining on synthetic images and without 3D data augmentation. Table 7.1 presents the contribution of the proposed semi-supervised training approach, considering one, two, and three input modes. All models were trained and evaluated under the same protocols except for the semi-supervised training approach when indicated. When not specified differently, semi-supervised models were trained with hyper-parameters α and γ set to 0.6 and 5, respectively.

We observed a positive contribution to each of the evaluated scenarios. The semi-supervised training approach enhances baseline SPAwN network results to levels never

input modes	DDR type	class balancing	train type	SSC IoU
depth	Regular	no	Sup.	21.6
	BN-DDR	no	Sup.	28.4
	BN-DDR	yes	Sup.	30.1
	BN-DDR	yes	S-Sup.	39.1
depth+rgb	Regular	no	Sup.	34.9
	BN-DDR	no	Sup.	38.4
	BN-DDR	yes	Sup.	39.4
	BN-DDR	yes	S-Sup.	<u>43.5</u>
depth+rgb+sn	Regular	no	Sup.	35.2
	BN-DDR	no	Sup.	39.2
	BN-DDR	yes	Sup.	41.4
	BN-DDR	yes	S-Sup.	**45.1**
oracle test	*BN-DDR*	yes	Sup.	67.9
	BN-DDR	yes	S-Sup.	67.9

Table 7.1: **Impact of *SP3* training procedure under several training scenarios on NYUDv2.** No pretraining or 3D data augmentation was performed. "sn" means surface normals. "Sup." and "S-Sup." mean supervised and semi-supervised training respectively.

seen before. Table 7.1 also evaluates the model's theoretical upper bound limit in an Oracle Test, supposing we have predicted perfect semantic 2D priors. To this matter, we replace the output of the 2D network with the 2D ground truth. The Oracle experiment shows there is still room for improvement by enhancing 2D predictions. As in Chapter 6, future works can exploit this opportunity. Moreover, the experiments show that the closer the 2D prior gets to perfection, the lower the benefit of semi-supervision. In those situations, lowering α would help to avoid adverse effects.

We further investigate the benefits of semi-supervised training with real images in more detail. Figure 7.2 shows the impact of the approach on model regularization and overfitting reduction. The semi-supervised validation curve consistently shows better results and is more stable than the regular supervised training one. Regarding overfitting, the standard supervised training curve ended up with a higher value than the semi-supervised one, but the semi-supervised validation curve was better than the supervised one, indicating a reduction in overfitting.

In Figure 7.3, we show the effect of changing hyper-parameters α and γ in the semi-supervised final result. The higher the γ hyper-parameter, the faster the training due to the decrease in unlabeled steps. For each value of γ there is an optimum value of $\alpha =$. The optimum value of $\alpha =$ for $\gamma = 7$ is probably outside the graph range. The graph indicates that $\alpha = .6$ and $\gamma = 5$ give the best mIoU results on NYUDv2. With this setup,

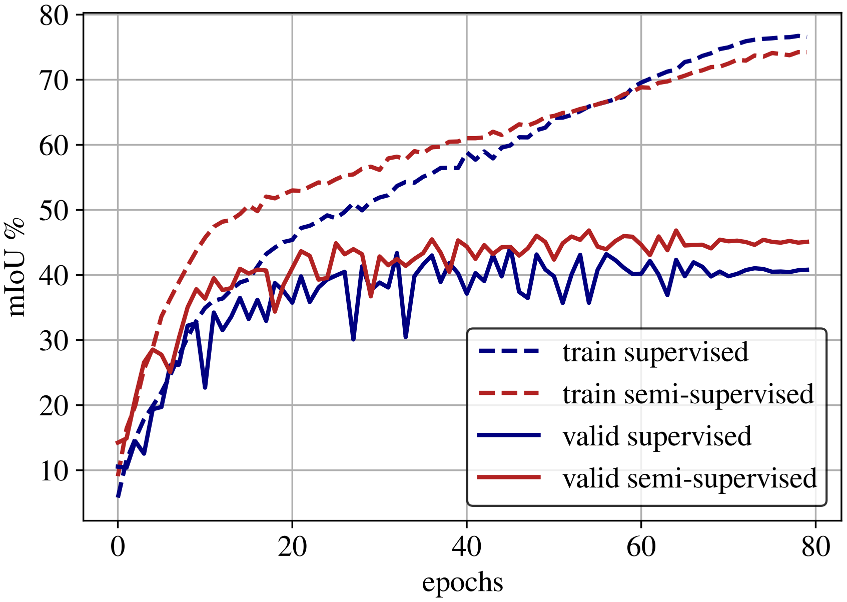


Figure 7.2: **Effect of the semi-supervised training** over model overfitting and regularization on NYDv2.

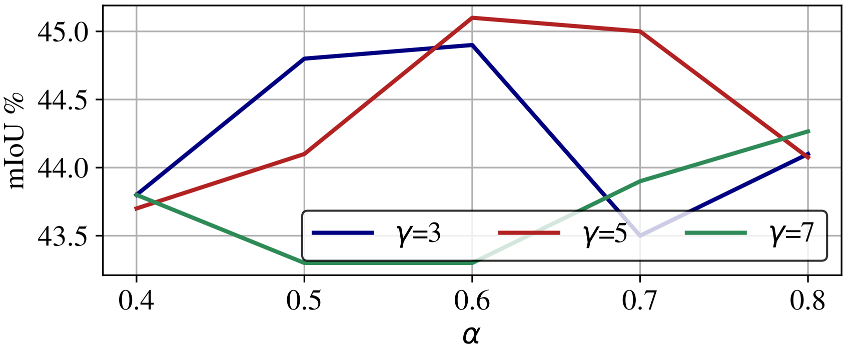


Figure 7.3: **Effect of the hyperparameters** α (unlabeled loss weight) and γ (number labeled steps for each unlabeled step) in semi-supervised training.

the impact on execution time is not significant, since the additional unsupervised training step only occurs after 5 regular steps.

7.2.3 Comparison to the State-of-the-Art

Here we compare our results to some of the best straightforward methods available by the time the experiments were run. For each training scenario, the best scores are presented in bold, while the second-best scores are underlined. We only show the best two or three competing models in each category. Further results are available in Appendix B.

SUNCG. As the SUNCG training set is large, the benefits of semi-supervised training are not expected to be significant. Therefore, we only evaluate the standard supervised

train	model	semantic scene completion (IoU, in percentages)											
		ceil.	floor	wall	win.	chair	bed	sofa	table	tvs	furn.	objs.	avg.
NYUDv2	TS3D[45]	9.7	93.4	25.5	21.0	17.4	55.9	49.2	17.0	27.5	39.4	19.3	34.1
	CCPNet[168]	23.5	**96.3**	35.7	20.2	25.8	61.4	56.1	18.1	28.1	37.8	20.1	38.5
	SketchAware[23]	**43.1**	93.6	**40.5**	24.3	30.0	57.1	49.3	**29.2**	14.3	42.5	28.6	41.1
	SPAwN (sup.)	22.9	94.8	35.8	25.4	33.2	65.6	54.4	20.0	33.5	44.2	25.7	41.4
	SPAwN+S3P (s-sup.)	35.6	94.4	37.0	**30.4**	**36.8**	**68.5**	**58.9**	23.4	**32.3**	**47.9**	**30.6**	**45.1**
SUNCG + NYUDv2	TNetFuse[96]	17.3	92.1	28.0	16.6	19.3	57.5	53.8	17.7	18.5	38.4	18.9	34.4
	ForkNet[156]	36.2	93.8	29.2	18.9	17.7	61.6	52.9	23.3	19.5	45.4	20.0	37.1
	CCPNet[168]	25.5	**98.5**	**38.8**	27.1	27.3	64.8	**58.4**	21.5	30.1	38.4	23.8	41.3
	SPAwN (sup.)	31.5	94.5	38.7	27.0	32.8	67.6	57.2	20.9	**30.7**	47.5	27.2	43.2
	SPAwN+S3P (s-sup.)	**37.5**	93.6	37.8	**35.0**	**39.4**	**71.9**	58.2	**23.4**	29.7	**50.7**	**34.2**	**46.5**

Table 7.2: **Results on NYUDv2 test set**. The column “train” indicates datasets used for training the models. SUNCG + NYU means trained on SUNCG and fine-tuned on NYUDv2. Our SPAwN semi-supervised and supervised models hold the best and second-best overall semantic scene completion results for real-world images, on both training scenarios.

train	model	semantic scene completion (IoU, in percentages)											
		ceil.	floor	wall	win.	chair	bed	sofa	table	tvs	furn.	objs.	avg.
NYUCAD	CCPNet[168]	56.2	**96.6**	58.7	**35.1**	44.8	68.6	65.3	37.6	35.5	53.1	35.2	53.2
	SketchAware[23]	**59.7**	94.3	**64.3**	32.6	51.7	72.0	68.7	45.9	19.0	60.5	38.5	55.2
	SPAwN (sup.)	54.0	94.7	61.6	33.4	62.8	80.7	68.9	**47.6**	**41.4**	61.5	**42.4**	**59.0**
	SPAwN+S3P (s-sup.)	57.4	94.5	60.7	33.5	**63.6**	**81.0**	**69.0**	44.0	40.9	**61.8**	41.6	58.9
SUNCG + NYUCAD	SSCNet[148]	32.5	92.6	40.2	8.9	33.9	57.0	59.5	28.3	8.1	44.8	21.1	40.0
	CCPNet[168]	58.1	**95.1**	60.5	36.8	47.2	69.3	67.7	39.8	37.6	55.4	37.6	55.0
	SPAwN (sup.)	62.4	94.7	65.3	38.3	**70.0**	82.4	**78.2**	**50.7**	40.2	**64.9**	**42.4**	62.7
	SPAwN+S3P (s-sup.)	**66.6**	94.7	**65.9**	**39.2**	69.6	**83.3**	78.0	50.4	**41.6**	64.4	42.2	**63.3**

Table 7.3: **Results on NYUDCAD**. Once again, our SPAwN semi-supervised and supervised models hold the best and second best overall semantic scene completion results on both training scenarios.

approach. The results of the regular supervised training were shown in Table 6.2, in Chapter 6.

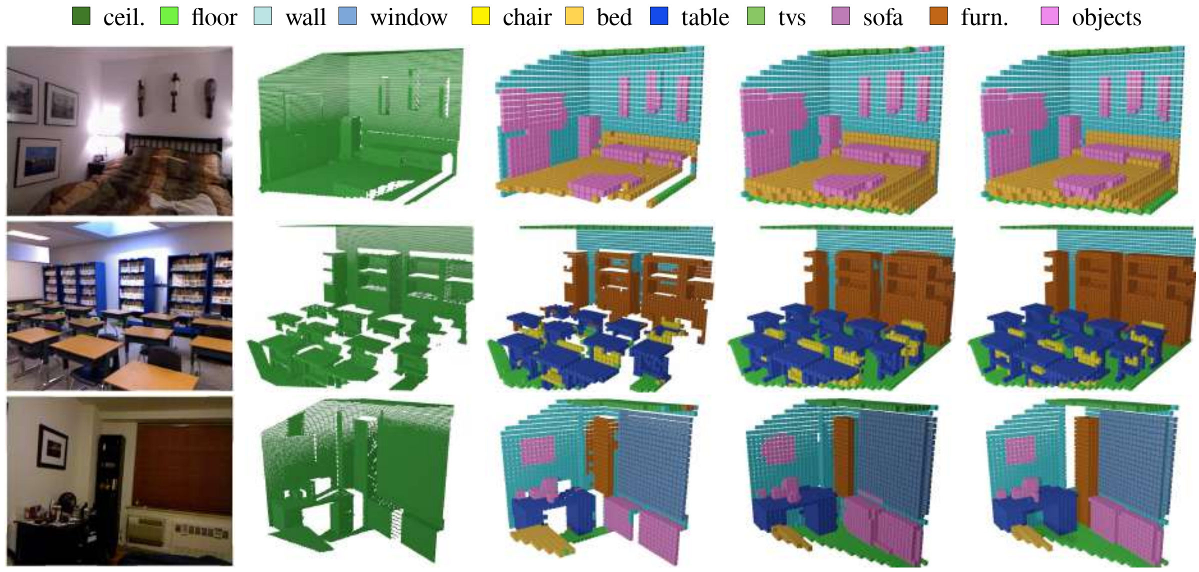
NYUDv2. Table 7.2 presents the results on real images. We evaluated two training scenarios: training from scratch on NYUDv2 and training on SUNCG, then fine-tuning to NYU. We also compared results using regular supervised training and with our proposed semi-supervised approach. *SPAwN* presented the best overall results in all scenarios. Our semi-supervised training approach confirmed the expectation of presenting a very representative boost over regular training, with a 12.6% margin over CCPNet (5.2p.p.).

NYUCAD. Table 7.3 confirms our expectation of good results due to the better quality of the ground truth related to surface volume and 2D priors. The observed mIoU boost of our supervised model over SketchAware and CCPNet, the best previous solution in each training scenario, is 6.9% (3.7p.p.) and 15% (8.3p.p.), respectively. These scenarios also confirmed the expectation of a lower impact of semi-supervision due to better quality of the 2D priors, as anticipated by the oracle test.

7.2.4 Qualitative Analysis

Figure 7.4 presents a qualitative analysis on NYUCAD due to its better alignment between depth map and ground truth compared to NYUDv2, making it easier to perceive our predictions' high quality visually. We generated predictions with *SPAwN* trained on SUNCG and fine-tuned on NYUCAD, using our *S3P* semi-supervised training approach with $\alpha = .2$ and $\gamma = 5$.

In column (c), it is possible to see that our projection and ensemble methods provide first-rate priors to the visible surface, with minimal prediction errors. *SPAwN* fusion strategy with *S3P* can complete the predictions and fix errors from priors, achieving a visually perceptible remarkable final prediction. Notice in the third row of figure 7.4 that the scene has a cornerstone that is not labeled as "wall" by the provided ground truth but was correctly labeled by our solution. Appendix B presents qualitative results on the other evaluation datasets.



(a) RGB (b) Visible surface (c) Semantic priors (d) Prediction (e) GT

Figure 7.4: **SPAwN & S3P qualitative results on NYUCAD.** 2D segmentation priors projected to 3D provide good semantic guidance. However, the resulting volume is incomplete and still presents some errors. SPAwN & S3P together complete and refine the predictions, and final results are visually close to perfection. (Best viewed in color).

7.3 Chapter Summary

In this Chapter, we presented your yet unpublished semi-supervised training approach for 3D SSC named *S3P* which is a semi-supervised training approach that uses unlabeled data in the form of semantic priors to regularize the output and reduce overfitting. *S3P* is independent of both 2D and 3D network architectures and may be combined with other SSC solutions. Here it was applied to *SPAwN*, our state-of-the-art SSC network architecture presented on Chapter 6 An ablation study with a comprehensive set of experiments shows the effectiveness of our semi-supervised approach. That study also includes an oracle test, which showed that the proposed solution can be further enhanced using better sources of semantic priors.

SPAwN with *S3P* surpasses by far all previous state-of-the-art solutions in SSC benchmarks, in all training scenarios, achieving a boost of 12.6% (5.2p.p.) over the best previously reported result on real images.

Chapter 8

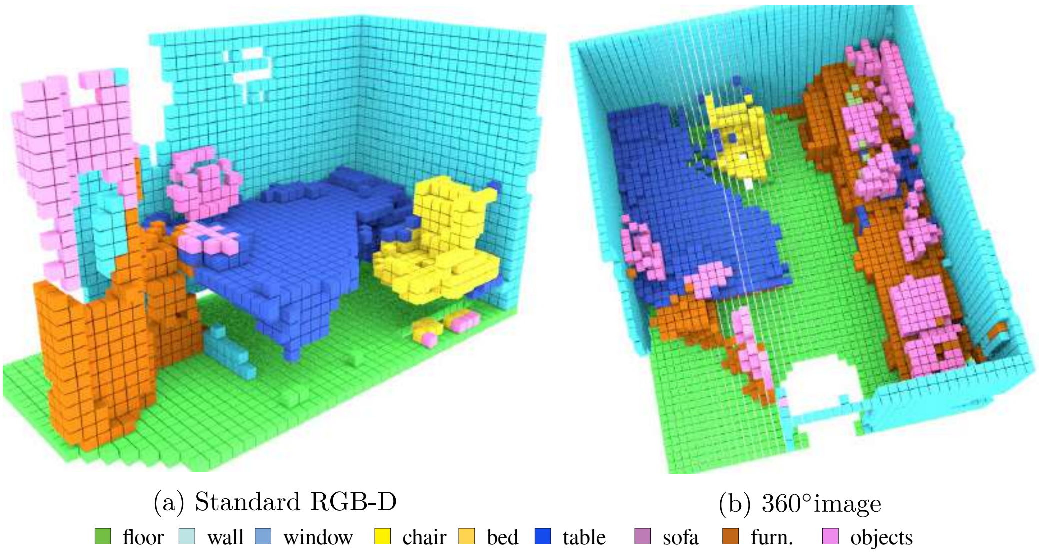
Extending Semantic Scene Completion for 360° Coverage

Previous works on SSC only perform occupancy prediction of small regions of the room covered by the field-of-view of regular RGB-D sensor in use. It would be possible to enlarge the coverage of the solution by using multiple shots to cover the whole scene, however, this approach is inappropriate for dynamic scenes[1]. One key aspect that favors limited angle approaches based on a single shot from regular RGB-D sensors is the abundance and variety of available datasets with densely annotated ground truth which enables the training of deep CNNs. On the other hand, their main drawback is the limited FOV of the sensor, as depicted in Figure 8.1.

As seen in Section 3.4, there are other sources of 360°RGB plus depth images, however, the datasets available for that kind of images are not densely annotated or are not big enough. In this Chapter, we present a method for Semantic Scene Completion (SSC) of complete indoor scenes from a single 360° RGB image and corresponding depth map using a Deep Convolution Neural Network that takes advantage of existing datasets of synthetic and real regular RGB-D images for training. Our approach uses a single 360° image with its corresponding depth map to infer the occupancy and semantic labels of the whole room. However, the SSC network is trained using regular RGB-D datasets. The use of a single image is important to allow predictions with no previous knowledge of the scene and enable the extension to dynamic scene applications.

We evaluated our method on two 360° image datasets: a high-quality 360° RGB-D dataset gathered with a Matterport® sensor and low-quality 360° RGB-D images generated with a pair of commercial 360° cameras and stereo matching. The experiments show

[1] To perform 360° SSC using images from regular RGBD sensors in dynamic scenes, it would be necessary to use multiple synchronized devices. This approach requires a complex setup which may restrict the number of possible applications.



(a) Standard RGB-D

(b) 360°image

Figure 8.1: SSC prediction from a regular RGB-D image in (a) covers only a small part of the Scene, while the result from panoramic RGB-D images in (b) covers the whole scene.

that the proposed pipeline performs SSC not only with Matterport® cameras but also with more affordable 360° cameras, which adds a great number of potential applications, including immersive spatial audio reproduction, augmented reality, assistive computing, and robotics.

The content of this Chapter was mainly extracted from our paper **Semantic Scene Completion from a Single 360° Image and Depth Map** [37] which was published in the proceedings of the Conference on Computer Vision Theory and Applications (VISAPP 2020). This work was developed in collaboration with the Centre of Vision, Speech and Signal Processing (CVSSP) of the University of Surrey, UK.

8.1 From Limited to Full 360° Scene Coverage

Due to the limited field-of-view (FOV) of regular RGB-D sensors like Microsoft Kinect®, current methods for Semantic Scene Completion (SSC) only predict semantic labels for a small part of the room and at least four images are required to understand the whole scene.

This scenario may change with the use of more advanced technology for large-scale 3D scannings, such as Light Detection and Ranging (LIDAR) sensors and Matterport® cameras. LIDAR is one of the most accurate depth-ranging devices using a light pulse signal but it acquires only a point cloud set without color or connectivity. Some recent LIDAR devices provide colored 3D structure by mapping photos taken during the scan[2],

[2] FARO LiDAR, `https://www.faro.com/products/construction-bim-cim/faro-focus/`

but it does not provide full texture maps. The Matterport® camera[3], using structured light sensors, allows the acquisition of 3D datasets that comprise high-quality panoramic RGB images and its corresponding depth maps of indoor scenes [2, 20] for a whole room. Figure 8.1 depicts the difference in SSC results from normal RGB-D and 360° RGB-D images.

Alongside the advanced sensors like Matterport, there are currently many consumer-level spherical cameras, allowing high-resolution 360° RGB image capture, which made widely possible the generation of 360° images and corresponding depth maps using two cameras through stereo matching. A system created to perform SSC for high-quality 360° images should be also able to work on images generated from low-cost cameras, widening the possibilities of applications.

Despite the interesting features of the new large-scale 3D datasets, the lack of variety in the type of scenes is an important drawback. For instance, while NYUDv2 regular RGB-D dataset [140] comprises a wide range of commercial and residential environments in three different cities across 26 scene classes, Stanford 2D-3D-Semantics large-scale dataset [3] only comprises 6 academic buildings and Matterport® 3D [20] dataset covers only 90 private homes. As most of the SSC solutions are data-driven and CNN-based, a dataset containing a large variety of scene types and object compositions is important to train generalized models. Another limitation of recent scene completion or segmentation methods that use large scans is that they usually take, as input, a point cloud generated from multiple points of view, implying pre-processing and some level of prior knowledge of the scene. Besides that, most of those new datasets are not densely annotated, instead, only bring annotations for the visible surface, which is not suitable for SSC tasks.

To overcome these limitations of both previous approaches, we propose an SSC method for a single 360° RGB image and its corresponding depth map image that uses 3D CNN trained on standard synthetic RGB-D data and fine-tuned on real RGB-D scenes. The overview of our proposed approach is presented in Figure 8.2. The proposed method decomposes a single 360° scene into several overlapping partitions so that each one simulates a single view of a regular RGB-D sensor, and submits to our pre-trained network. The final result is obtained by aligning and ensembling the partial inferences.

We evaluated our method on two datasets: the Stanford 2D-3D-Semantics Dataset (2D-3D-S) [2] gathered with the Matterport® sensor; and a set of stereo 360° images captured by a pair of low cost 360° cameras by ourselves. Both datasets are further detailed in section 3.5. For the experiments with low-cost cameras, we propose a pre-processing method to enhance noisy 360° depth maps before submitting the images to the

[3] Matterport, `https://matterport.com/pro2-3d-camera/`

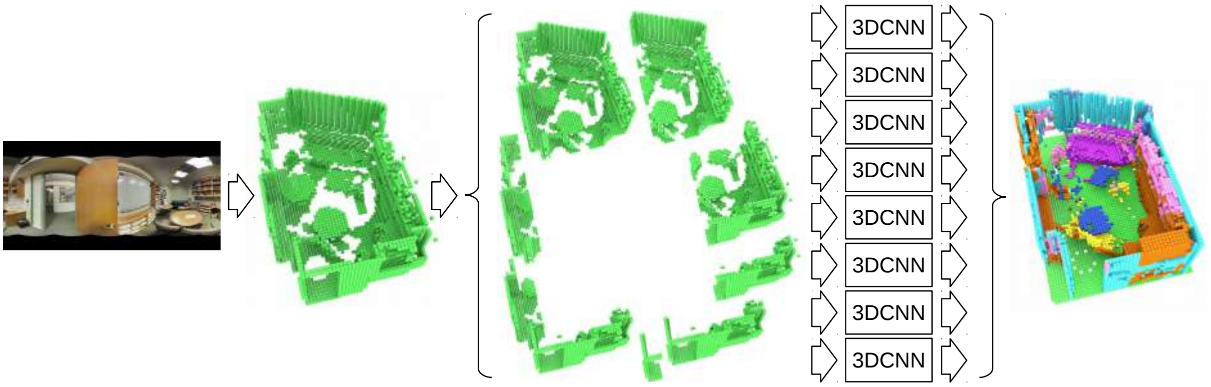


Figure 8.2: **Overview of our proposed approach**. The incomplete voxel grid generated from the input panoramic depth map is automatically partitioned into 8 overlapping views that are individually submitted to our 3D CNN. The resulting prediction is generated from an automatic ensemble of the 8 individual predictions. The result is a complete 3D voxel volume with corresponding semantic labels for occluded surfaces and objects' interiors.

network for prediction. Our qualitative analysis shows that the proposed method achieves reliable results with the low-cost 360° cameras.

8.2 Proposed Approach

Our proposed approach, illustrated in Figure 8.2, is described in detail in the next subsections.

8.2.1 Input Partitioning

From the 360° panoramic depth map, we generate a voxel grid of all the visible surfaces from the camera position, resulting in an incomplete and sparse 3D volume ($480\times144\times480$ voxels). The preferred voxel size throughout this work is 0.02m which gives a coverage of $9.6\times2.8\times9.6m^3$. The resulting volume is then automatically partitioned into 8 views using a 45° step, each of them emulating the field of view of one standard RGB-D sensor. The emulated sensor is positioned at 1.7m back from the original position of the 360° sensor, in order to get a better-overlapped coverage, especially when the camera is placed near a wall, as is the case of a scene from Figure 8.2 (in that scene, the camera is placed in the bottom left corner of the room). The reason for taking overlapping partitions is to improve the final prediction in the borders of the emulated sensors FOV, by ensembling multiple SSC estimates. Voxels behind the original sensor position are not included in the partition. Each partition size is $240 \times 144 \times 240$ voxels.

8.2.2 Semantic Scene Completion Network

In our experiments, we used our FCN EdgeNet-MF, which was presented in Chapter 5[4]. After the input partitioning, the resulting partitions are individually submitted to EdgeNet for prediction. The partition scheme for the edge volume is the same as that used for the surface volume. As the final activation function of EdgeNet is a Softmax, each voxel of the output volume contains the predicted probabilities of the 12 classes used for training. The output resolution for each partition is 60 × 36 × 60 voxels.

Our EdgeNet was pretrained on standard RGB-D images extracted from the SUNCG training set and fine-tuned on NYUDv2 following the training protocol described in Chapter 5.

8.2.3 Prediction Ensemble

Each partition of the input data is processed by our CNN, generating 8 predicted 3D volumes. There are significant overlaps between the FOV of each CNN (some voxels are even captured from 3 different viewpoints), and their predictions need to be combined. We use a simple yet effective strategy of summing the *a posteriori* probability for each class over all classifier outputs, i.e., we apply the "sum rule", demonstrated by Kittler *et al.* [74]. Firstly, the prediction of each partition is aligned according to its position in the final voxel volume. If a given voxel is not covered by a given partition, then the corresponding classifier *a posteriori* probabilities for all classes for that voxel and that partition will be 0, i.e., the softmax result is overruled in voxels outside the field of view of a given partition. Otherwise, the sum of the *a posteriori* probabilities for all classes for that voxel and that classifier will be 1. Given that, for any arbitrary voxel, being n the number of partitions and P_{ij} the *a posteriori* probability of the class i predicted by the classifier j, then, the sum of the probabilities for class i over all classifiers is given by

$$S_i = \sum_{j=1}^{n} P_{ij} \tag{8.1}$$

and the winning class C for that voxel is

$$C = \arg\max_i(S_i)\ . \tag{8.2}$$

[4] By the time we started working on the 360°extension, our EdgeNet architecture was one of the state-of-the-art methods available for SSC

8.2.4 Depth Map Enhancement

Since an affordable single-shot 360° RGB-D system is not available in the market, stereo capture using commercial 360° cameras is a realistic approach. The problem is that depth estimation from stereoscopic images is subject to errors due to occlusions between two camera views and correspondence matching errors. These depth errors would lead to noisy and incomplete predictions in SSC. We propose a preprocessing step to enhance this erroneous depth map by taking into account two characteristics of most of the indoor scenes:

1. their alignment to the Cartesian axis, following the Manhattan principle [52];
2. the edges present in the RGB images are usually distinguishing features for stereo matching, providing good depth estimates in their neighborhood.

The Canny edge detector [17], with low and high thresholds of 30 and 70, is applied to the RGB image and the edges are dilated to 3 pixels width. We observed that those parameters work well for a wide range of RGB images. Using the dilated edges as a mask, we extract the most reliable depth estimations from the original depth map. Vertical edges are removed from the mask as they do not contribute to the stereo matching procedure in the given vertical stereo camera setup. Using the thin edges as a border delimitation, coherent regions with similar colors are searched by a flood-fill approach in the RGB image. With this procedure, we expect to get featureless planar surfaces like single-colored walls and table tops whose depth surfaces are hard to be estimated by stereo matching. RANSAC [44] is used to fit a plane over those regions eliminating outliers from false stereo matching. If the normal vector of the fitted plane is close to one of the principal axes, we replace the original depth information of the region with the back-projected depths estimated from the plane. Discarding non-orthogonal planes is important to avoid planes estimated from non-planar regions, like wall corners, where the contrast is not enough to produce an edge between two walls. We keep the original depth estimations from the regions where we were not able to fit good planes. We also re-estimate the depths of the south and north poles of the image, as they usually have bad depth estimations as proved by Kim and Hilton [69]. Good depth estimations from the outer neighborhood of the poles are used as a source for the RANSAC plane fitting.

8.3 Datasets

We take advantage of existing diverse RGB-D training datasets to train our networks for general semantic scene completion. After training, we evaluate the performance of our model on datasets never seen before by the networks. This section describes the datasets used for training and evaluation.

As in Chapter 5, we trained our 3D CNN on RGB-D depth maps from the training set of SUNCG [148] and fine-tuned the networks on the train set of NYUv2 dataset [140]. Refer to subsection 3.5 for detailed information on the datasets.

Two distinct datasets are used for evaluation: Stanford 2D-3D-Semantics [2] (large-scale scan) and a dataset created by low-cost 360° cameras.

8.3.1 Stanford 2D-3D-Semantics

Stanford 2D-3D-Semantics is a large-scale scan dataset gathered with a Materpport camera in academic indoor spaces (refer to Figure 2.13). The dataset covers over 6,000 m^2 from 7 distinct building areas. For each room of the building area, two or more 360° scans containing several RGB-D images were taken. The images from the scans are aligned, combined, and post-processed to generate one large-scale point cloud file for each building area. The point cloud is then annotated with 13 class labels, to be used as ground truth. Each point of the point cloud is also annotated with the room to which it belongs. The dataset also provides a complete RGB 360° panorama, with corresponding depths for each room scan, camera rotation/translation information, and other features useful for 3D understanding tasks. Depth maps are provided as 16 bits PNG images, with a sensibility of 1/512 m. The value $2^{16} - 1$ is used for pixels without a valid depth measurement.

8.3.2 General 360°cameras

In order to show general applications of the proposed pipeline, we also used an in-house dataset called Surrey's Spherical Stereo images, captured by two 360° cameras in a vertical stereo setup (refer to Figure 2.11 and Section 2.1.3). The image sets consist of 3 360° scenes: Meeting Room, Usability Lab, and Kitchen. The Meeting Room is similar to a normal living room environment in our daily lives including various objects such as sofas, tables, bookcases, etc. The Usability Lab is similar to the Meeting Room in its size but includes more challenging objects for scene understanding such as large windows and a big mirror on the walls. The Kitchen is a small and narrow room with various kitchen utensils. The scenes are captured as a vertical stereo image pair and dense stereo matching with spherical stereo geometry [70] is used to recover depth information. This dataset is available to download from the S3A AV dataset page from CVSSP web site [154].

8.4 Evaluation

We quantitatively evaluated our approach on the Stanford 2D-3D-Semantics dataset. We also provide a qualitative evaluation on that dataset and on our stereoscopic images. In this section, we describe the experiments and discuss the results.

8.4.1 Evaluation Metric

As in previous works on SSC, we evaluate our proposed approach using Interception over Union (IoU) for each class, on visible occupied and occluded voxels inside the room. However, unlike RGB-D works that only evaluate voxels inside the field of view of the sensor, we evaluate the whole scene. Unfortunately, Stanford 2D-3D does not provide ground truth for the interior of the objects nor for areas that are not visible from at least one of the scanning points, so we limit our quantitative evaluation to the areas to which ground is provided. We kept the predictions not covered by the ground truth for qualitative evaluation purposes.

8.4.2 Experiments on Stanford 2D-3D-Semantics Dataset

In order to feed our SSC network with aligned volumes, we rotated the provided 360° RGB panoramas and depth maps using the camera rotation matrix before generating a corresponding input point cloud. Using the room dimensions provided by the dataset, we discarded depth estimations outside the room and generated the voxel volume by placing the camera in the center of the X and Z axis and keeping the capture height so that the floor level is at the voxel plane y=0.

Quantitative Evaluation

For quantitative evaluation, we extracted only the points belonging to the room from the provided ground-truth (GT) point cloud and translated them to the camera position. In order to align the GT to our input volumes, we voxelized the point cloud using the same voxel size as our input volumes.

Stanford 2D-3D-Semantics dataset classifies each point in 13 classes, while the ground truth extracted from the datasets used to train our network (SUNCG and NYU) classifies the voxels in 12 classes. We mapped the classes *board* and *bookcase* from the Stanford 2D-3D-Semantics dataset to classes $objects$ and $furniture$; and both classes *beam* and *door* to $wall$. Predictions of the classes *bed* and *tv* from SUNCG that have no correspondence in the Stanford 2D-3D-Semantics dataset were remapped to *table* and $objects$, respectively. We evaluated all the panoramas from all rooms of types office, conference room, pantry,

evaluation	model	scene	semantic scene completion (IoU, in percentages)											
dataset		coverage	ceil.	floor	wall	win.	chair	bed	sofa	table	tvs	furn.	objs.	avg.
NYUDv2	SSCNet [148]	partial	15.1	94.6	24.7	10.8	17.3	53.2	45.9	15.9	13.9	31.1	12.6	30.5
	SGC [166]		17.5	75.4	25.8	6.7	15.3	53.8	42.4	11.2	0.0	33.4	11.8	26.7
	EdgeNet-MF		**22.4**	**95.0**	29.7	**15.5**	20.9	**54.1**	**53.0**	15.6	**14.9**	**35.0**	14.8	33.7
Stanford 2D-3D-S	**Ours**	360°	15.6	92.8	**50.6**	6.6	**26.7**	-	35.4	**33.6**	-	32.2	**15.4**	**34.3**

Table 8.1: **Quantitative results.** We compare our 360° semantic scene completion results on the Stanford 2D-3D-S dataset to partial view state-of-the-art approaches in a standard RGB-D dataset (NYUDv2), available by the time we performed the experiments. Our network was trained on SUNCG and NYUDv2 train sets and had no previous knowledge of the evaluation dataset and predicts results for the whole scene. Previous approaches were fine-tuned on the target dataset and only give partial predictions. Even so, our proposed solution achieved better overall results.

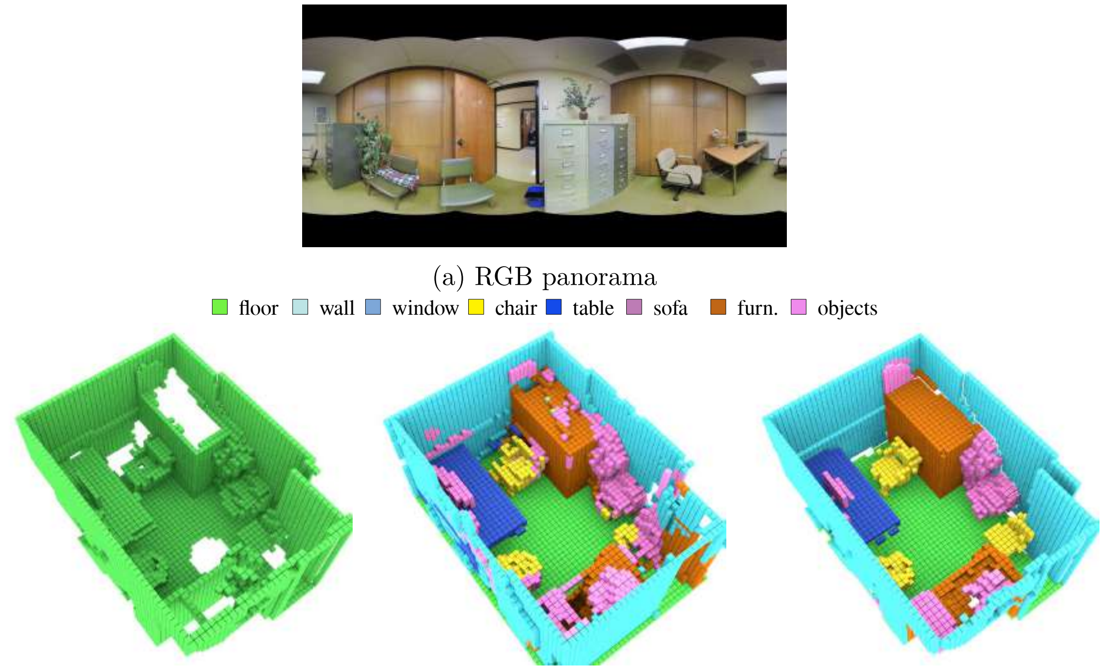
copy room, and storage. We discarded room types open space, lounge, hallway, and WC. We evaluated 669 pairs of 360° RGB images and depth maps from the Stanford 2D-3D-Semantics dataset.

Quantitative results for the Stanford 2D-3D-Semantics dataset are provided in Table 8.1. As a baseline, we compare our results to previous works on SSC evaluated on the NYUDv2 dataset. It is worth mentioning that, as those results are from different datasets and tasks (our work is the only one that covers the whole scene), they cannot be taken as a direct comparison of the model's performance.

Our 360° EdgeNet-based ensemble achieved very good overall results and a high level of semantic segmentation accuracy was observed on structural elements floor and wall. Good results were also observed on common scene objects like chairs, sofas, tables, and furniture, as well. On the other hand, the same level of performance was not observed on the ceiling, due to domain shift [27] between training and evaluation datasets. The ceiling in the Stanford dataset is on average higher than that in the NYU dataset where the network was trained. Even so, given that our model had no previous knowledge of the dataset being evaluated, results show that the proposed model has a good generalization power.

Qualitative Evaluation

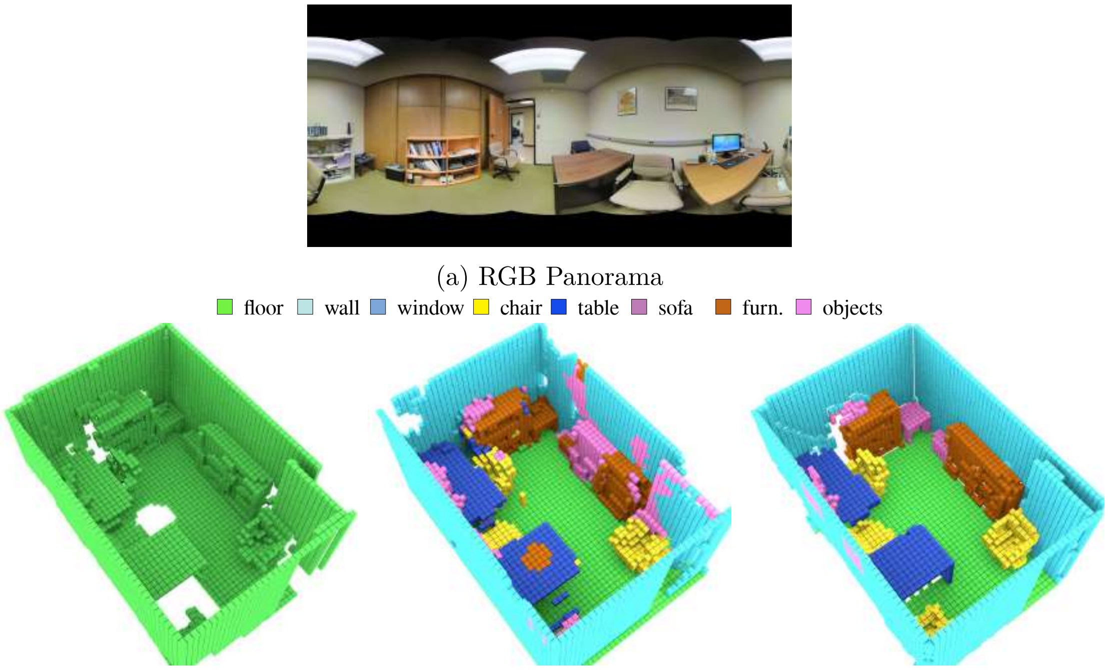
Sample results presented in this subsection (Figures 8.3 to 8.8) depict the high level of completion achieved by our approach with high-quality input, as seen by comparing the input volume (green) to the prediction. The level of completion is even higher than the ground truth models, which were manually composed and labeled by the authors of the dataset using the surface gathered from multiple viewpoints. Note that the missing and occluded regions in the ground truth of scenes were completed in their corresponding predicted volumes.



(a) RGB panorama

(b) Incomplete input volume (c) Predicted volume (d) Ground Truth (GT)

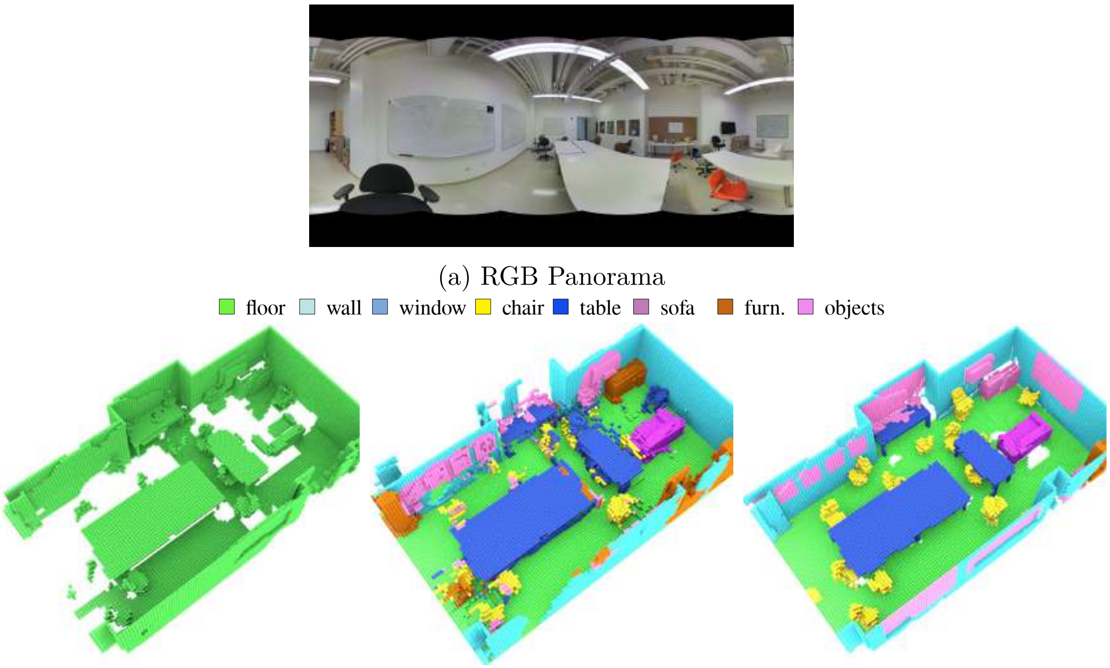
Figure 8.3: **Qualitative evaluation on Stanford 2D-3D - area 4, office 1** (best viewed in colour).



(a) RGB Panorama

(b) Incomplete input volume (c) Predicted volume (d) Ground Truth (GT)

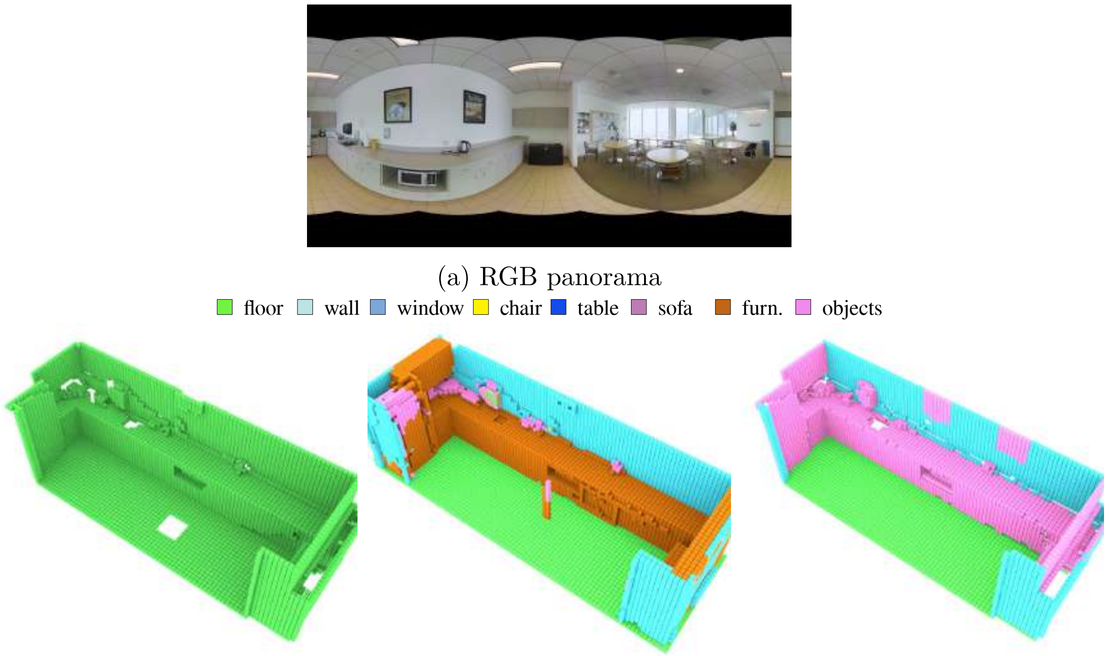
Figure 8.4: **Qualitative evaluation on Stanford 2D-3D - area 4, office 3** (best viewed in colour).



(a) RGB Panorama

(b) Incomplete input volume (c) Predicted volume (d) Ground Truth (GT)

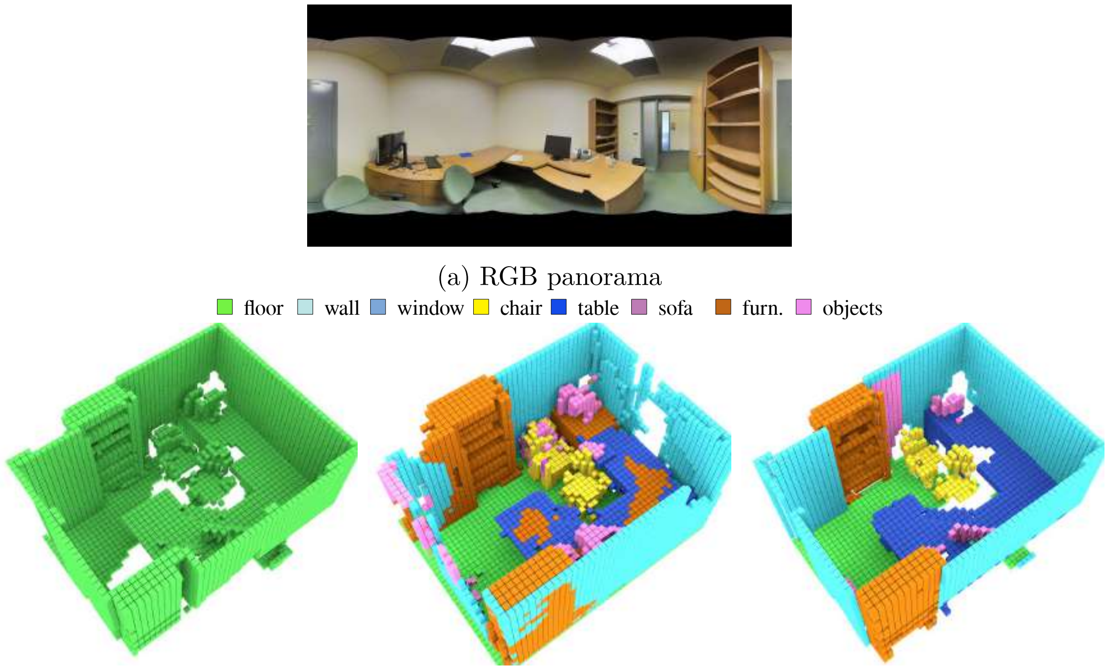
Figure 8.5: **Qualitative evaluation on Stanford 2D-3D - area 5b, conference room 2** (best viewed in colour).



(a) RGB panorama

(b) Incomplete input volume (c) Predicted volume (d) Ground Truth (GT)

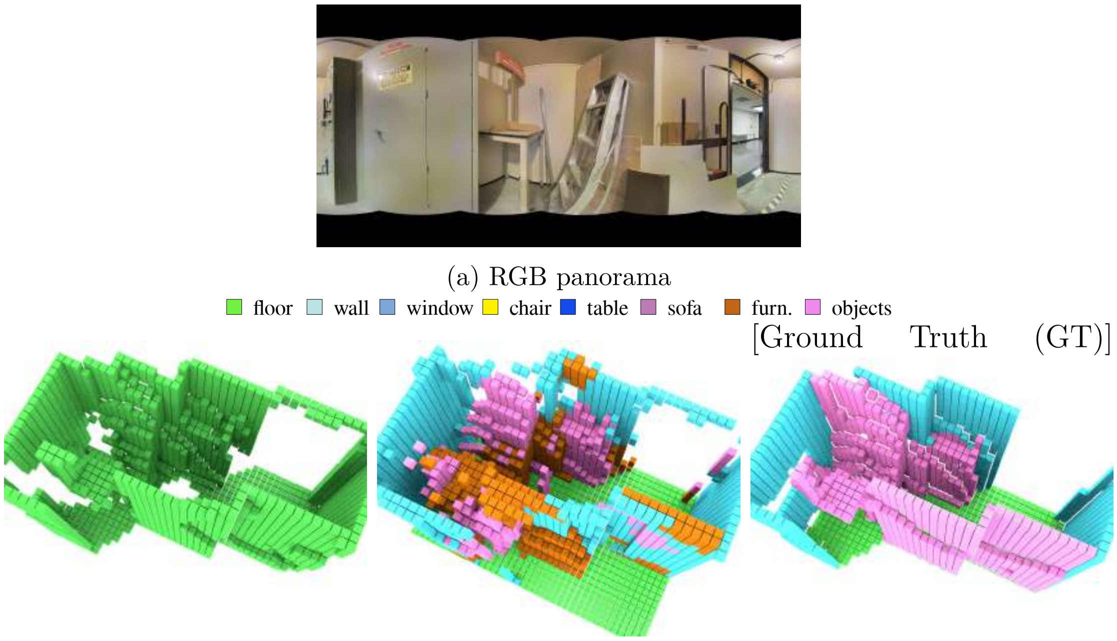
Figure 8.6: **Qualitative evaluation on Stanford 2D-3D - area 5b, pantry 1** (best viewed in colour).



(a) RGB panorama

(b) Incomplete input volume (c) Predicted volume (d) Ground Truth (GT)

Figure 8.7: **Qualitative evaluation on Stanford 2D-3D - area 6, office 17** (best viewed in colour).



(a) RGB panorama

[Ground Truth (GT)]

(b) Incomplete input volume (c) Predicted volume (d) Ground Truth (GT)

Figure 8.8: **Qualitative evaluation on Stanford 2D-3D - area 2, storage 6** (best viewed in colour).

Completion Capabilities. Figures 8.3 (area 4, office 1), 8.4 (area 4, office 3), 8.5 (area 5b, conference room 2) and 8.7 highlight the completion capabilities of our approach. Note how the input volume (c) is completed on the prediction (d). Furthermore, note how some incomplete regions in the ground truth (e) have been completed by the proposed method.

Labeling accuracy. The labeling accuracy is high in most of the scenes. This is well illustrated in Figures 8.3 and 8.4.

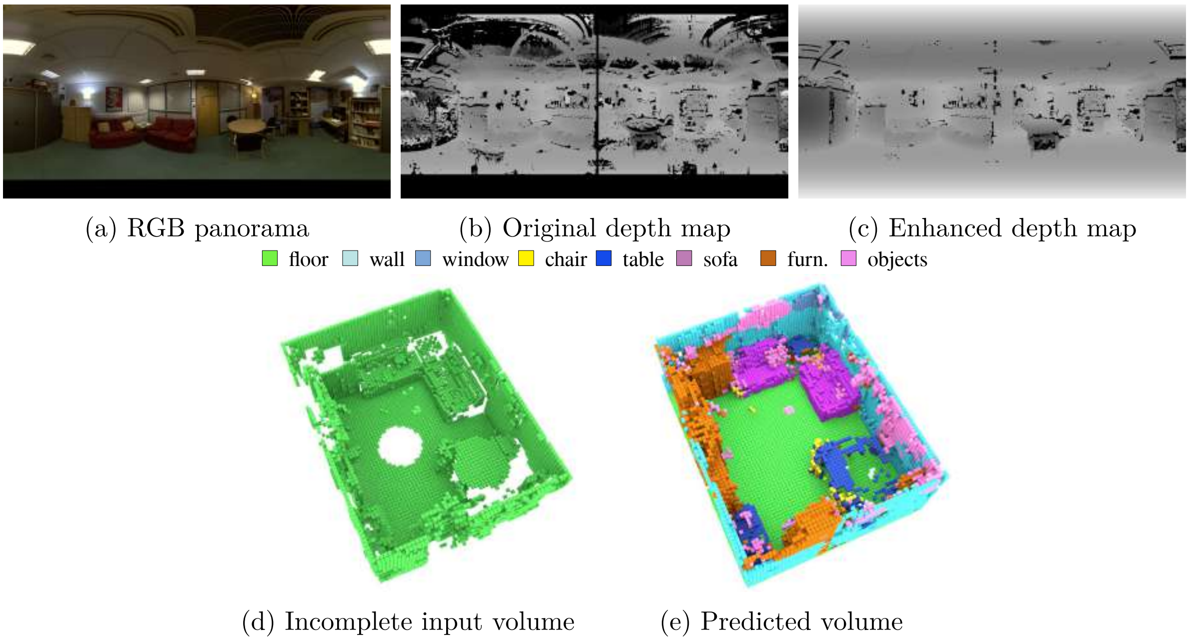
Performance on challenging scenes. The scene presented in figure 8.5 (area 5b, conference room 2) is quite challenging due to its size and the presence of hard-to-detect objects. Note that the scene did not fit on the input voxel space. Despite that issue, the part of the scene that fits was well predicted. Note how the pictures on the wall are invisible in the surface input volume (c). This was not a problem for our edge-based model, those pictures were correctly detected as *objects* in the predicted volume (d).

Ground truth mislabelling. Figure 8.6 illustrates one case of ground truth mislabelling. The provided ground truth label for the kitchen cabinet is *clutter* and we mapped it to SUNCG generic class *objects*. However, our network classified the cabinet as furniture, which seems to be more adequate. Although this scene is labeled as a pantry, it appears to be an open-plan area that combines a kitchen and an office dining room. However, the ground truth only included the kitchen area, so half of the visible volume was disregarded. Since we obtain the room dimensions from the same source, our model also disregarded the dining room area. Object mislabelling also happened in Figure 8.8 (area 2, storage 6). The equipment cabinets were labeled as class *object* and our model predicted as *furniture*.

8.4.3 Experiments on Surrey's Spherical Stereo Images

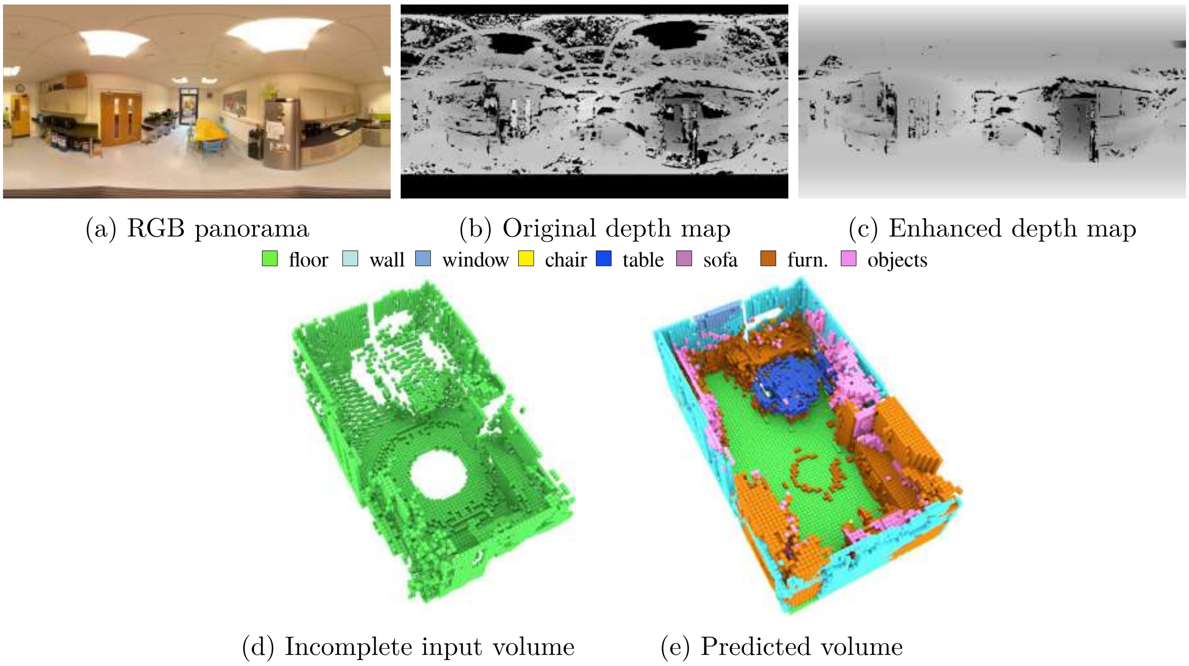
For spherical stereo images, we first rotated them to align to the Cartesian axis and applied the enhancement procedure described in Section 8.2.4. From the resulting images, we generated a point cloud and voxelized the surface and edges with a voxel size of 0.02m, before encoding the volumes with F-TSDF and submitting them to the neural networks. Room dimensions are inferred from the point clouds.

Some results are shown in Figures 8.9 to 8.11 for a qualitative evaluation. Most of the stereo-matching errors of the estimated depth maps are fixed by our enhancement approach. The cabinet in the extreme left part of the Meeting Room (first scene) originally had several depth estimation errors due to the vertical striped patterns, but most errors were eliminated by the enhancement step, though some errors still remained in dark regions where borders are not clear. The lower border of the leftmost sofa in the second scene (Usability Lab) was not detected, and some part of its original depth was replaced


(a) RGB panorama (b) Original depth map (c) Enhanced depth map

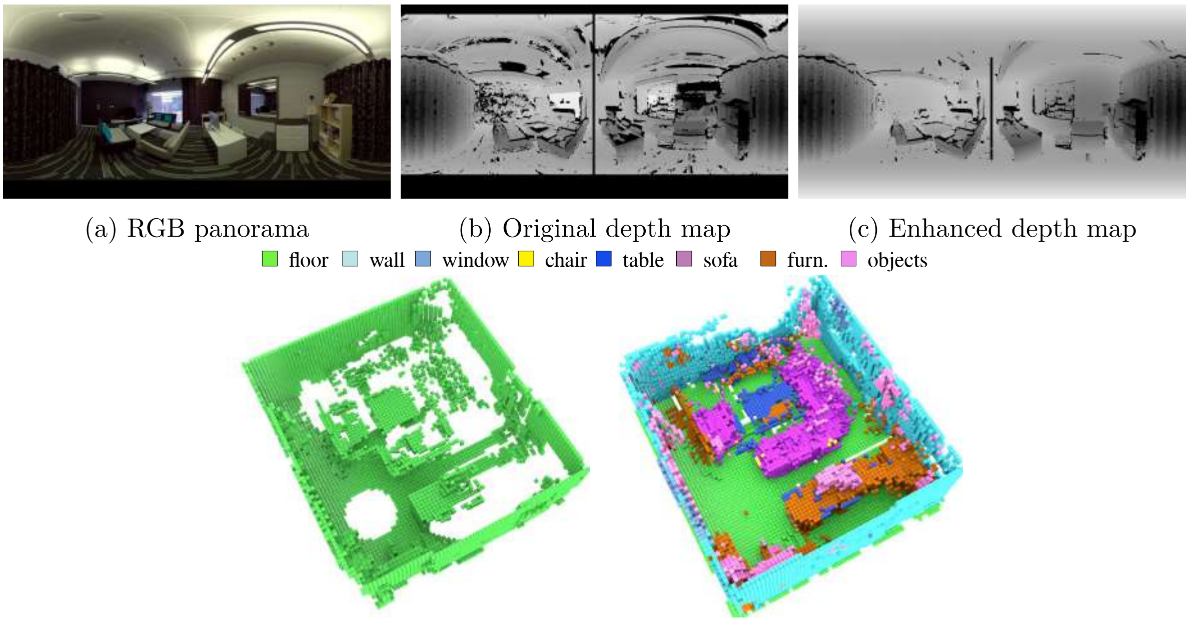
(d) Incomplete input volume (e) Predicted volume

Figure 8.9: **Qualitative evaluation on stereoscopic image - Meeting Room** (best viewed in colour).


(a) RGB panorama (b) Original depth map (c) Enhanced depth map

(d) Incomplete input volume (e) Predicted volume

Figure 8.10: **Qualitative evaluation on stereoscopic image - Kitchen** (best viewed in colour).



(a) RGB panorama (b) Original depth map (c) Enhanced depth map

(d) Incomplete input volume (e) Predicted volume

Figure 8.11: **Qualitative evaluation on stereoscopic image - Usability Lab** (best viewed in color).

by the depth of the floor. However, the proposed depth enhancement step improved the erroneous depth maps estimated by stereo matching over the entire regions.

The SSC results with the enhanced depth maps were also satisfactory. As in the large-scale dataset, the levels of scene completion and semantic labeling were high. Although the input images still carry some depth errors, the final predictions were generally clear enough. Comparing the final predictions from the stereo 360° dataset to the ones from the Stanford 2D-3D-Semantics dataset, the results of spherical stereo ones are noisier than those of the scanned counterparts, but they are still accurate. Results demonstrate that the use of a pair of 360° images gives an inexpensive alternative to perform 360° SSC for dynamic scenes, where large-scale depth scans are not applicable.

8.5 Application: audio-visual VR system

In audio-visual reproduction for virtual reality (VR) systems, personalized audio-visual experiences pose an important issue to improve the sense of presence. Human ambience perception relies on both audio and visual cues to understand and interact with the environment [81]. A full 3D reconstruction of a real space, such as provided by our 360°EdgeNet, which enables a precise association of the acoustic properties of each type of material of the predicted classes, allows users to realistically experience the space remotely.

In collaboration with the University of Surrey, our 360°EdgeNet model and our depth improvement approach were used in an immersive audio-visual virtual reality system, with great results. In this system, our SSC solution was responsible to generate a full representation of scenes captured with a pair of 360°cameras. Each class of this representation was associated with a material that allowed a full reconstruction of the whole scene structure and acoustic properties. Finally, the reconstructed audio-visual VR scene is rendered by setting sound source and player models on a Unity VR platform[5]. Figure 8.12 presents an overview of the audio-visual reconstruction pipeline. This method obtained much faster semantic scene reconstruction with geometric details and achieved better agreement between the real and simulated acoustics than the state-of-the-art algorithm [72] through objective and subjective evaluations.

The proposed audio-visual reproduction system was first presented in the paper **Immersive audio-visual scene reproduction using semantic scene reconstruction from 360 cameras**, published in the Virtual Reality Journal [71].

[5] Unit is a platform for creating and operating interactive, real-time 3D content (`https://unity.com/`)

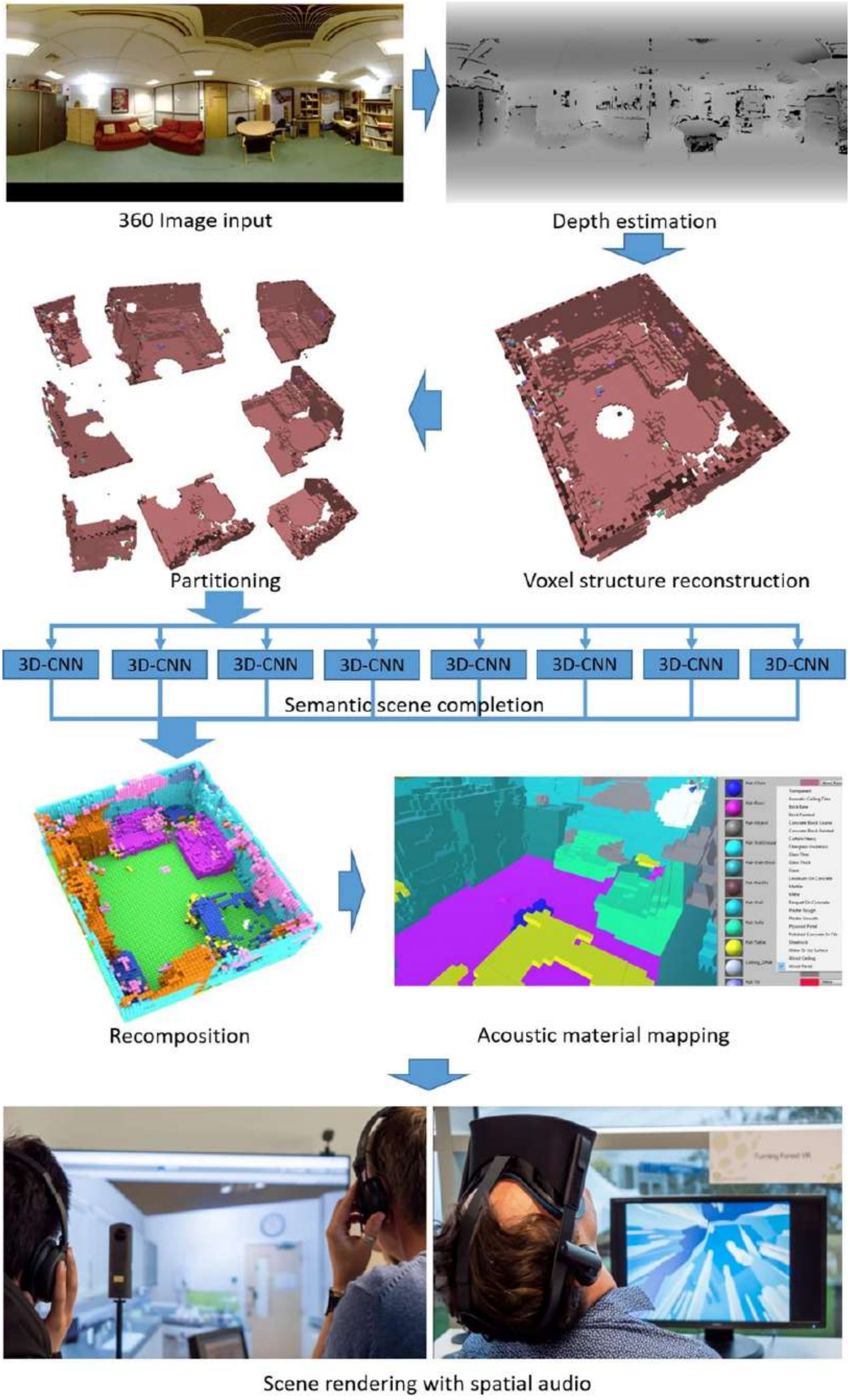

360 Image input
Depth estimation
Voxel structure reconstruction
Partitioning
3D-CNN
3D-CNN
3D-CNN
3D-CNN
3D-CNN
3D-CNN
3D-CNN
3D-CNN
Semantic scene completion
Recomposition
Acoustic material mapping
Scene rendering with spatial audio

Figure 8.12: **Overview of the immersive audio-visual VR system powered by our EdgeNet360.** Reproduced from our paper in collaboration with the University of Surrey [71] (best viewed in color).

8.6 Chapter Summary

In this Chapter, we introduced the task of 360°Semantic Scene Completion from a panoramic image and corresponding depth map. Our proposed method to predict 3D voxel occupancy and its semantic labels for a whole scene from a single point of view can be applied to various ranges of images acquired from high-end sensors like Matterport® to off-the-shelf 360° cameras. The proposed method is based on a CNN which relies on existing diverse RGB-D datasets for training. For images from spherical cameras, we also presented an effective method to enhance stereoscopic 360° depth maps to be used prior to submitting the images to the SSC network.

Our method was evaluated on two distinct datasets: the publicly available Stanford 2D-3D-Semantics high-quality large-scale scan dataset and a collection of 360° stereo images gathered with off-the-shelf spherical cameras from the University of Surrey. Our SSC network requires no previous knowledge of the datasets to perform the evaluation. Even so, when we compare our results to previous approaches using RGB-D images that only give results for part of the scene and were trained on the target datasets, the proposed method achieved better overall results with full coverage. Qualitative analysis shows high levels of completion of occluded regions on both Matteport and spherical images. On the large-scale scan dataset, completion levels achieved from a single point of view were superior to the ones of the ground truth obtained from multiple points of view.

The results show that our approach can be extended to applications that require a complete understanding of 3D dynamic scenes from images gathered using off-the-shelf stereo cameras.

In partnership with Surrey University, the solution presented in this chapter was applied to an audio-visual scene reproduction system that generates realistic 3D audio. The system uses scenes reconstructed with our 360°EdgeNet to reproduce scene sound ambience. This work generated the paper **Immersive audio-visual scene reproduction using semantic scene reconstruction from 360 cameras**, that was published in the virtual reality journal [71].

All source code and pretrained models required to reproduce the experiments presented in this Chapter are publicly available in `https://gitlab.com/UnBVision/edgenet360`.

Chapter 9

Conclusion

In this thesis, we presented the contributions that our research project brought to the computer vision community in the sense of advancing towards a complete 3D indoor scene understanding from a single point of view. We choose to work with the SSC task, which we believe to be one of the most complete tasks in the scene understanding research field.

Our work was fruitful, generating several contributions and publications. Our ultimate network architecture, *SPAwN*, with our data augmentation strategy still holds the state-of-the-art for SSC with straightforward pipelines. Our 360°SSC approach was used in a VR system also achieving state-of-the-art results for realistic immersive audio-visual reproduction.

9.1 Research Objectives Achievement

We achieved our general objective, which is to propose, implement and evaluate new tools and models that could push SSC solutions towards a complete understanding of the whole indoor scene, including enhancing the coverage and quality of the inferences. This goal was achieved by:

1. accessing the benefits of domain adaptation and semi-supervised training techniques in the context of 2D image segmentation (Chapter 4) which was useful to further explore unlabeled data in 3D SSC (Chapter 7);
2. applying current trends on 2D deep CNN training protocols to 3D SSC networks, specifically data augmentation (Chapter 6);
3. proposing and evaluating new SSC models that use the RGB information present in RGB-D images and overcome the sparsity problem when projecting features from 2D to 3D (Chapters 5 to 6);

4. to propose and evaluate a multi-modal deep neural network to explore multiple modes of the RGB-D image and enhance 3D SSC scores(Chapter 6);
5. to propose and access the benefits of the use of unlabeled data in 3D SSC through semi-supervised learning(Chapter 7);
6. proposing and evaluating a solution to perform 360° SSC using existing limited FOV datasets for training (Chapter 8).

9.2 Contributions

Here we summarize the main contributions presented in this thesis.

1. a new Domain Adaptation strategy that combines Pseudo-Labeling and Transfer Learning for cross-domain training (Chapter 4);
2. EdgeNet, a new end-to-end CNN architecture that fuses depth and RGB edge information to achieve good performance in semantic scene completion with a much simpler approach than previous works (Chapter 5);
3. a new 3D volumetric edge representation using flipped signed-distance functions which improves performance and unifies data aggregation for semantic scene completion from RGBD (Chapter 5);
4. a more efficient end-to-end training pipeline for semantic scene completion with relation to previous approaches (Chapter 5);
5. *SPAwN*, a novel lightweight multimodal 3D SSC CNN architecture that uses 2D prior probabilities from a 2D segmentation network, that holds current state-of-the-art results on both real and synthetic data (Chapter 6);
6. *BN-DDR*, a memory-saving batch-normalized dimensional decomposition residual building block for 3D CNNs that preserves previous approaches' regularization characteristics while consuming much less memory during training (Chapter 6);
7. a novel strategy to apply a data augmentation technique for 3D semantic scene completion based on three 3D data transformations that operate on batches directly in GPU tensors(Chapter 6);
8. *S3P*, a novel 2D-prior-based semi-supervised training approach to the SSC task that explores unlabeled data from the target dataset in a pseudo-label inspired approach and dramatically reduces overfitting when training with a small amount of labeled data (Chapter 7);

9. a novel approach to perform SSC for 360° images taking advantage of existing standard RGB-D datasets for network training (Chapter 8);

10. a pre-processing method to enhance depth maps estimated from a stereo pair of low-cost 360° cameras (Chapter 8).

9.3 Publications

Here we list the publications resulting from our research project:

1. **Domain Adaptation for Holistic Skin Detection** [36] - published in the proceedings of the *34th SIBGRAPI Conference on Graphics, Patterns and Images* **(SIBGRAPI 2021)**;

2. **EdgeNet: Semantic Scene Completion from RGB-D images** [34] - published in the proceedings of the *International Conference on Pattern Recognition* **(ICPR 2020)**;

3. **Data Augmented 3D Semantic Scene Completion With 2D Segmentation Priors** [35] - published in the proceedings of Proceedings of the *IEEE/CVF Winter Conference on Applications of Computer Vision* **(WACV 2022)**

4. **Semantic Scene Completion from a Single 360° Image and Depth Map** [37] - published in the proceedings of the Conference on Computer Vision Theory and Applications **(VISAPP 2020)**;

5. **Immersive audio-visual scene reproduction using semantic scene reconstruction from 360 cameras** [71] - Virtual Reality Journal **(VIRE)**.

9.4 Future Work

Even though we believe we have contributed to the evolution of the Semantic Scene Completion (SSC) field of study towards a more complete understating of the 3D scene, there is still space for advancing much further. Some of these advances can be achieved by a direct extension of our work as we explain in the following paragraphs.

In Chapters 6 to 7 we introduced the use of two techniques in the SSC task: 3D data augmentation and semi-supervision. These techniques can be combined into a single model, hopefully achieving even better results, in future work.

Going a little further than a simple combination of techniques, the semi-supervision approach (Chapter 7) itself can be extended to explore large-scale real 3D datasets without dense 3D labels, but with 2D labels like Stanford 2D-3D-Semantics Dataset [2] and Matterport3D [20] (refer to Section 8.3.1). Those are panoramic datasets, however, the partitioning technique presented on Chapter 8 could be used to generate scenes similar to those from regular RGB-D sensors.

Regarding 360° SSC, the resulting model trained on a semi-supervised fashion with large-scale datasets could be used to replace EdgeNet as the base model for the 360° SSC approach presented in Chapter 8. Using a model trained with large-scale datasets is expected to generate much more accurate 360° full scene predictions than we saw on Chapter 8. Besides that, most of the techniques introduced in this thesis may be applied to other state-of-the-art network architectures, probably improving their results.

Another line that could be studied is the use of the new tendency of Visual Convolutional Transformers [160] as a candidate network architecture applied to 3D convolutions for Semantic Scene Completion (SSC).

References

[1] Aarts, E. and Korst, J.: *Simulated Annealing and Boltzmann Machines: A Stochastic Approach to Combinatorial Optimization and Neural Computing.* John Wiley & Sons, Inc., New York, NY, USA, 1989, ISBN 0-471-92146-7. 73

[2] Armeni, I., Sax, S., Zamir, A.R., and Savarese, S.: *Joint2D-3D-semantic data for indoor scene understanding.* Tech. Rep. arXiv:1702.01105, Cornell University Library, 2017. `http://arxiv.org/abs/1702.01105`. 5, 22, 46, 108, 112, 127

[3] Armeni, I., Sener, O., Zamir, A.R., Jiang, H., Brilakis, I., Fischer, M., and Savarese, S.: *3D semantic parsing of large-scale indoor spaces.* In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, June 26 - July 1*, pp. 1534–1543, Piscataway, NJ, June 2016. IEEE. `https://doi.org/10.1109/CVPR.2016.170`. 108

[4] Arnab, A., Zheng, S., Jayasumana, S., Romera-Paredes, B., Larsson, M., Kirillov, A., Savchynskyy, B., Rother, C., Kahl, F., and Torr, P.H.: *Conditional random fields meet deep neural networks for semantic segmentation: Combining probabilistic graphical models with deep learning for structured prediction.* IEEE Signal Processing Magazine, 35(1):37–52, 2018. 30

[5] Aytar, Y. and Zisserman, A.: *Tabula rasa: Model transfer for object category detection.* In *Proceedings of 13th International Conference on Computer Vision, Barcelona, Spain*, pp. 2252–2259, 2011. 35

[6] Baars, B.J. and Gage, N.M.: *Chapter 6 - vision.* In Baars, B.J. and Gage, N.M. (eds.): *Cognition, Brain, and Consciousness (Second Edition)*, pp. 156 – 193. Academic Press, London, second edition ed., 2010, ISBN 978-0-12-375070-9. `http://www.sciencedirect.com/science/article/pii/B9780123750709000061`. 1

[7] Bainbridge, W.S.: *Ai: The tumultuous history of the search for artificial intelligence.* Science, 261(5125):1186–1187, 1993. 25

[8] Barrow, H.G.: *Recovering intrinsic scene characteristics from images.* Computer Vision Systems, 1978. `https://ci.nii.ac.jp/naid/10011460027/en/`. 1

[9] Bay, H., Tuytelaars, T., and Van Gool, L.: *SURF: Speeded up robust features.* In *European conference on computer vision*, pp. 404–417. Springer, 2006. 17

[10] Bengio, Y., Louradour, J., Collobert, R., and Weston, J.: *Curriculum learning.* In *Proceedings of the 26th Annual International Conference on Machine Learning*,

ICML, pp. 41–48, New York, NY, USA, 2009. ACM, ISBN 978-1-60558-516-1. `http://doi.acm.org/10.1145/1553374.1553380`. 73

[11] Bhowmik, A.K.: *Sensification of computing: adding natural sensing and perception capabilities to machines.* APSIPA Transactions on Signal and Information Processing, 6:e1, 2017. 10, 11

[12] Borgwardt, K.M., Gretton, A., Rasch, M.J., Kriegel, H.P., Schölkopf, B., and Smola, A.J.: *Integrating structured biological data by kernel maximum mean discrepancy.* Bioinformatics, 22(14):e49–e57, 2006. 36

[13] Bousmalis, K., Silberman, N., Dohan, D., Erhan, D., and Krishnan, D.: *Unsupervised pixel-level domain adaptation with generative adversarial networks.* IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 95–104, 2017. 51

[14] Brancati, N., Pietro, G.D., Frucci, M., and Gallo, L.: *Human skin detection through correlation rules between the YCb and YCr subspaces based on dynamic color clustering.* Computer Vision and Image Understanding, 155:33 – 42, 2017, ISSN 1077-3142. 51, 52, 59, 60, 63, 66

[15] Cai, Y., Chen, X., Zhang, C., Lin, K.Y., Wang, X., and Li, H.: *Semantic scene completion via integrating instances and scene in-the-loop.* In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 324–333, June 2021. 85, 93, 94, 147, 148

[16] Campos, T. de: *3D Visual Tracking of Articulated Objects and Hands.* PhD thesis, University of Oxford, 2006. 14, 16

[17] Canny, J.: *A computational approach to edge detection.* IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 8(6):679–698, Nov 1986, ISSN 0162-8828. 70, 111

[18] Casati, J.P.B., Moraes, D.R., and Rodrigues, E.L.L.: *SFA: A Human Skin Image Database based on FERET and AR Facial Images.* In *IX Workshop de Visão Computacional*, p. 5, 2013. 58, 59

[19] Chai, D. and Ngan, K.N.: *Face segmentation using skin-color map in videophone applications.* IEEE Transactions on Circuits and Systems for Video Technology, 9(4):551–564, June 1999. 63

[20] Chang, A.X., Dai, A., Funkhouser, T.A., Halber, M., Nießner, M., Savva, M., Song, S., Zeng, A., and Zhang, Y.: *Matterport3D: Learning from RGB-D data in indoor environments.* Tech. Rep. arXiv:1709.06158, Cornell University Library, 2017. `http://arxiv.org/abs/1709.06158`. 5, 22, 46, 47, 108, 127

[21] Charles, R.Q., Su, H., Kaichun, M., and Guibas, L.J.: *PointNet: Deep learning on point sets for 3D classification and segmentation.* In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, Hawaii, July 21-26*, pp. 77–85, Piscataway, NJ, July 2017. IEEE. 5, 47

[22] Chen, L., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A.L.: *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, April 2018, ISSN 0162-8828. 46

[23] Chen, X., Lin, K.Y., Qian, C., Zeng, G., and Li, H.: *3D Sketch-Aware Semantic Scene Completion via Semi-Supervised Structure Prior.* In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 85, 88, 94, 103, 148, 154

[24] Ciresan, D., Giusti, A., Gambardella, L.M., and Schmidhuber, J.: *Deep neural networks segment neuronal membranes in electron microscopy images.* In *Advances in neural information processing systems*, pp. 2843–2851, 2012. 30, 51, 54, 55

[25] Conaire, C.O., O'Connor, N.E., and Smeaton, A.F.: *Detector adaptation by maximising agreement between independent data sources.* In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–6, June 2007. 51

[26] Criminisi, A. and Shotton, J.: *Semi-supervised classification forests.* In *Decision Forests for Computer Vision and Medical Image Analysis*, ch. 8, pp. 95–107. Springer, 2013. `https://doi.org/10.1007/978-1-4471-4929-3_8`. 36

[27] Csurka, G.: *A comprehensive survey on domain adaptation for visual applications.* In Csurka, G. (ed.): *Domain Adaptation in Computer Vision Applications*, pp. 1–35. Springer International Publishing, Cham, 2017, ISBN 978-3-319-58347-1. 2, 34, 35, 36, 51, 67, 114

[28] Csurka, G.: *Domain adaptation in computer vision applications.* Springer, 2017. 36, 53

[29] Csurka, G., Dance, C.R., Fan, L., Willamowski, J., and Cédric: *Visual categorization with bags of keypoints.* In *Workshop on Statistical Learning in Computer Vision, European Conference on Computer Vision (ECCV)*, pp. 1534–1543, Piscataway, NJ, June 2004. IEEE. 2

[30] Curless, B. and Levoy, M.: *A volumetric method for building complex models from range images.* In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pp. 303–312, 1996. 33, 43

[31] Dai, A., Ritchie, D., Bokeloh, M., Reed, S., Sturm, J., and Niessner, M.: *ScanComplete: Large-Scale Scene Completion and Semantic Segmentation for 3D Scans.* In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, June 18-22*, pp. 4578–4587, Piscataway, NJ, 2018. IEEE. 47

[32] Dalal, N. and Triggs, B.: *Histograms of oriented gradients for human detection.* In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, pp. 886–893 vol. 1, 2005. 2

[33] Deng, J., Dong, W., Socher, R., Li, L., Kai Li, and Li Fei-Fei: *Imagenet: A large-scale hierarchical image database.* In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, June 2009. 2, 25, 28

[34] Dourado, A., De Campos, T.E., Kim, H., and Hilton, A.: *Edgenet: Semantic scene completion from a single rgb-d image.* In *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 503–510. IEEE, 2021. 7, 69, 85, 89, 91, 93, 126, 147, 148, 153, 154

[35] Dourado, A., Guth, F., and Campos, T. de: *Data augmented 3d semantic scene completion with 2d segmentation priors.* In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 3781–3790, January 2022. 7, 85, 126

[36] Dourado, A., Guth, F., Campos, T. de, and Weigang, L.: *Domain adaptation for holistic skin detection.* In *34th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI 2021)*, pp. 362–369, 2021. 7, 50, 97, 126

[37] Dourado, A., Kim, H., de Campos, T.E., and Hilton, A.: *Semantic scene completion from a single 360-degree image and depth map.* In *Proceedings of the 15th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2020)*, vol. 5: VISAPP, pp. 36–46. INSTICC, SciTePress, 2020, ISBN 978-989-758-402-2. `https://doi.org/10.5220/0008877700360046`. 8, 107, 126

[38] Eigen, D. and Fergus, R.: *Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture.* In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, December 2015. 39

[39] Engelen, J.E. van and Hoos, H.H.: *A survey on semi-supervised learning.* Machine Learning, 109(2):373–440, Feb. 2020, ISSN 1573-0565. `https://doi.org/10.1007/s10994-019-05855-6`. 97

[40] Everingham, M., Eslami, S., Van Gool, L., Williams, C.K., Winn, J., and Zisserman, A.: *Assessing the significance of performance differences on the pascal voc challenges via bootstrapping.* Technical note, pp. 1–4, 2013. 30

[41] FarajiDavar, N., de Campos, T., and Kittler, J.: *Adaptive transductive transfer machines: A pipeline for unsupervised domain adaptation.* In *Domain Adaptation in Computer Vision Applications*, Advances in Computer Vision and Pattern Recognition, pp. 115–132. Springer International, 2017. 36, 67

[42] Faria, R.A.D. and Hirata Jr., R.: *Combined correlation rules to detect skin based on dynamic color clustering.* In *Proceedings of the 13th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISAPP)*, vol. 5, pp. 309–316. INSTICC, SciTePress, 2018, ISBN 978-989-758-290-5. 51, 59, 60

[43] Firman, M., Aodha, O.M., Julier, S., and Brostow, G.J.: *Structured prediction of unobserved voxels from a single depth image.* In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, June 26 - July 1*, pp. 5431–5440, Piscataway, NJ, June 2016. IEEE. 2, 40, 48, 91, 99

[44] Fischler, M.A. and Bolles, R.C.: *Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography.* Commun. ACM, 24(6):381–395, June 1981, ISSN 0001-0782. `http://doi.acm.org/10.1145/358669.358692`. 111

[45] Garbade, M., Chen, Y.T., Sawatzky, J., and Gall, J.: *Two Stream 3D Semantic Scene Completion.* In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019. 5, 46, 74, 75, 83, 94, 103, 148, 154

[46] Gatys, L.A., Ecker, A.S., and Bethge, M.: *Image style transfer using convolutional neural networks.* In *Proceedings of of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2414–2423, June 2016. 36, 67

[47] Geiger, A. and Wang, C.: *Joint 3d object and layout inference from a single rgb-d image.* In *German Conference on Pattern Recognition*, pp. 183–195. Springer, 2015. 40

[48] Golomb, B.A., Lawrence, D.T., and Sejnowski, T.J.: *Sexnet: A neural network identifies sex from human faces.* In *NIPS*, vol. 1, p. 2, 1990. 28

[49] Guedes, A.B.S., de Campos, T.E., and Hilton, A.: *Semantic scene completion combining colour and depth: preliminary experiments.* In *ICCV workshop on 3D Reconstruction Meets Semantics (3DRMS)*, Venice, Italy, October 2017. Event webpage: `http://trimbot2020.webhosting.rug.nl/events/events-2017/3drms/`. Also published at arXiv:1802.04735. 4, 45, 69, 74, 75, 83, 148, 154

[50] Guo, R., Zou, C., and Hoiem, D.: *Predicting complete 3D models of indoor scenes.* Tech. Rep. arXiv:1504.02437, Cornell University Library, 2015. `http://arxiv.org/abs/1504.02437`. 48

[51] Guo, Y. and Tong, X.: *View-Volume Network for Semantic Scene Completion from a Single Depth Image.* In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, pp. 726–732, Stockholm, Sweden, July 2018. International Joint Conferences on Artificial Intelligence Organization, ISBN 978-0-9992411-2-7. `https://doi.org/10.24963/ijcai.2018/101`. 2, 4, 22, 45, 74, 75, 147, 148, 153, 154

[52] Gupta, A., Efros, A.A., and Hebert, M.: *Blocks world revisited: Image understanding using qualitative geometry and mechanics.* In *Proceedings of 11th European Conference on Computer Vision (ECCV), Crete, Greece, September 5-11*, pp. 482–496, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg. 111

[53] Gupta, S., Arbeláez, P., Girshick, R., and Malik, J.: *Aligning 3d models to rgb-d images of cluttered scenes.* In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4731–4740, 2015. 40

[54] Gupta, S., Arbeláez, P., and Malik, J.: *Perceptual organization and recognition of indoor scenes from rgb-d images.* In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Portland, OR, June 23-28*, pp. 564–571, Piscataway, NJ, June 2013. IEEE. 2, 39

[55] Gupta, S., Girshick, R., Arbeláez, P., and Malik, J.: *Learning rich features from RGB-D images for object detection and segmentation.* In Fleet, D., Pajdla, T., Schiele, B., and Tuytelaars, T. (eds.): *Computer Vision – (ECCV)*, pp. 345–360, Cham, 2014. Springer International Publishing, ISBN 978-3-319-10584-0. 39

[56] Hamzah, R.A. and Ibrahim, H.: *Literature survey on stereo vision disparity map algorithms.* Journal of Sensors, 2016, Nov 2016. 12

[57] Handa, A., Patraucean, V., Badrinarayanan, V., Stent, S., and Cipolla, R.: *SceneNet: Understanding real world indoor scenes with synthetic data.* Tech. Rep. arXiv:1511.07041, Cornell University Library, 2015. `http://arxiv.org/abs/1511.07041`. 48

[58] Hartley, R. and Zisserman, A.: *Multiple view geometry in computer vision.* Cambridge University Press, 2004, ISBN 978-0-511-18618-9. `https://doi.org/10.1017/CBO9780511811685`, OCLC: 804793563. 12, 14, 15, 17, 18

[59] He, K. and Sun, J.: *Convolutional neural networks at constrained time cost.* In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. 71

[60] He, K., Zhang, X., Ren, S., and Sun, J.: *Deep residual learning for image recognition.* In *Proceedings of of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, June 2016. 29, 31, 46, 71, 84, 86, 88, 89

[61] Huynh-Thu, Q., Meguro, M., and Kaneko, M.: *Skin-Color-Based Image Segmentation and Its Application in Face Detection.* In *MVA*, pp. 48–51, 2002. 51, 63

[62] Iscen, A., Tolias, G., Avrithis, Y., and Chum, O.: *Label propagation for deep semi-supervised learning.* In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 97

[63] Jiang, H. and Xiao, J.: *A linear approach to matching cuboids in rgbd images.* In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2171–2178, 2013. 41

[64] Jiao, J., Wei, Y., Jie, Z., Shi, H., Lau, R.W., and Huang, T.S.: *Geometry-aware distillation for indoor semantic segmentation.* In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 39, 89

[65] Jones, M.J. and Rehg, J.M.: *Statistical color models with application to skin detection.* In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Fort Collins CO, June*, vol. 1, pp. 274–280 Vol. 1, 1999. 58

[66] Kaehler, A. and Bradski, G.: *Learning OpenCV 3: Computer Vision in C++ with the OpenCV Library.* O'Reilly Media, 2016, ISBN 9781491937969. `https://books.google.com.br/books?id=LPm3DQAAQBAJ`. 16, 17

[67] Kakumanu, P., Makrogiannis, S., and Bourbakis, N.: *A survey of skin-color modeling and detection methods.* Pattern Recognition, 40(3):1106 – 1122, 2007, ISSN 0031-3203. 52

[68] Kamnitsas, K., Baumgartner, C., Ledig, C., Newcombe, V., Simpson, J., Kane, A., Menon, D., Nori, A., Criminisi, A., Rueckert, D., and Glocker, B.: *Unsupervised domain adaptation in brain lesion segmentation with adversarial networks.* In Niethammer, M., Styner, M., Aylward, S., Zhu, H., Oguz, I., Yap, P.T., and Shen, D. (eds.): *Information Processing in Medical Imaging*, pp. 597–609, Cham, 2017. Springer International Publishing, ISBN 978-3-319-59050-9. 51

[69] Kim, H. and Hilton, A.: *3D scene reconstruction from multiple spherical stereo pairs.* Int Journal of Computer Vision (IJCV), 104(1):94–116, Aug 2013, ISSN 1573-1405. `https://doi.org/10.1007/s11263-013-0616-1`. 47, 111

[70] Kim, H. and Hilton, A.: *Block world reconstruction from spherical stereo image pairs.* Computer Vision and Image Understanding (CVIU), 139(C):104–121, Oct. 2015, ISSN 1077-3142. `http://dx.doi.org/10.1016/j.cviu.2015.04.001`. 21, 112

[71] Kim, H., Remaggi, L., Dourado, A., Campos, T.d., Jackson, P.J., and Hilton, A.: *Immersive audio-visual scene reproduction using semantic scene reconstruction from 360 cameras.* Virtual Reality, pp. 1–16, 2021. 9, 121, 122, 123, 126

[72] Kim, H., Remaggi, L., Jackson, P.J., and Hilton, A.: *Immersive spatial audio reproduction for VR/AR using room acoustic modelling from 360 images.* In *Proceedings of 26th IEEE Conference on Virtual Reality and 3D User Interfaces, Osaka Japan*, Piscataway, NJ, 2019. IEEE. 47, 121

[73] Kim, Y.M., Mitra, N.J., Yan, D.M., and Guibas, L.: *Acquiring 3d indoor environments with variability and repetition.* ACM Transactions on Graphics (TOG), 31(6):1–11, 2012. 40

[74] Kittler, J., Hatef, M., Duin, R.P.W., and Matas, J.: *On combining classifiers.* IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 20(3):226–239, Mar. 1998, ISSN 0162-8828. `https://doi.org/10.1109/34.667881`. 87, 91, 110

[75] Klaus, A., Sormann, M., and Karner, K.: *Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure.* In *18th International Conference on Pattern Recognition (ICPR'06)*, vol. 3, pp. 15–18, 2006. 20

[76] Kline, R.: *Cybernetics, automata studies, and the dartmouth conference on artificial intelligence.* IEEE Annals of the History of Computing, 33(4):5–16, 2010. 23

[77] Krizhevsky, A., Sutskever, I., and Hinton, G.E.: *Imagenet classification with deep convolutional neural networks.* In Pereira, F., Burges, C.J.C., Bottou, L., and Weinberger, K.Q. (eds.): *Advances in Neural Information Processing Systems*, vol. 25. Curran Associates, Inc., 2012. `https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf`. 2, 4, 28, 29, 84, 89

[78] Lai, K., Bo, L., Ren, X., and Fox, D.: *A large-scale hierarchical multi-view rgb-d object dataset.* In *2011 IEEE International Conference on Robotics and Automation*, pp. 1817–1824, 2011. 48

[79] Lalonde, J.F., Vandapel, N., Huber, D.F., and Hebert, M.: *Natural terrain classification using three-dimensional lidar data for ground robot mobility.* Journal of Field Robotics, 23(10):839–861, 2006. 2

[80] Larochelle, H. and Bengio, Y.: *Classification using discriminative restricted Boltzmann machines.* In *Proceedings of the 25th international conference on Machine learning - ICML*, pp. 536–543, Helsinki, Finland, 2008. ACM Press, ISBN 978-1-60558-205-4. `https://doi.org/10.1145/1390156.1390224`. 36

[81] Larsson, P., Väljamäe, A., Västfjäll, D., Tajadura-Jiménez, A., and Kleiner, M.: *Auditory-induced presence in mixed reality environments and related technology.* In *The engineering of mixed reality systems*, pp. 143–163. Springer, 2010. 120

[82] Lecun, Y.: *Generalization and network design strategies.* Connectionism in perspective, 1989. `https://ci.nii.ac.jp/naid/10008946620/en/`. 27

[83] Lecun, Y., Boser, B., Denker, J., Henderson, D., Howard, R., Hubbard, W., and Jackel, L.: *Handwritten digit recognition with a back-propagation network.* In Touretzky, D. (ed.): *Advances in Neural Information Processing Systems (NIPS 1989), Denver, CO*, vol. 2. Morgan Kaufmann, 1990. 27

[84] Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P.: *Gradient-based learning applied to document recognition.* Proceedings of the IEEE, 86(11):2278–2324, Nov 1998, ISSN 1558-2256. 26, 28

[85] Lee, D.H.: *Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks.* In *ICML Workshop on Challenges in Representation Learning (WREPL)*, pp. 1–6, July 2013. 4, 36, 52, 53, 97, 98

[86] Leistner, C., Saffari, A., Santner, J., and Bischof, H.: *Semi-supervised random forests.* In *Proceedings of 12th International Conference on Computer Vision, Kyoto, Japan, Sept 27 - Oct 4*, pp. 506–513. IEEE, 2009. 36

[87] Li, C., Kowdle, A., Saxena, A., and Chen, T.: *Towards holistic scene understanding: Feedback enabled cascaded classification models.* In Lafferty, J.D., Williams, C.K.I., Shawe-Taylor, J., Zemel, R.S., and Culotta, A. (eds.): *Advances in Neural Information Processing Systems 23*, pp. 1351–1359. Curran Associates, Inc., 2010. `http://papers.nips.cc/paper/4003-towards-holistic-scene-understanding-feedback-enabled-cascaded-classification-models`. 2

[88] Li, J., Liu, Y., Gong, D., Shi, Q., Yuan, X., Zhao, C., and Reid, I.: *RGB-D based dimensional decomposition residual network for 3D Semantic Scene Completion.* In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 88, 148, 153, 154

[89] Li, S.: *Real-time spherical stereo.* In *Proceedings of International Conference on Pattern Recognition (ICPR)*, pp. 1046–1049, Piscataway, NJ, 2006. IEEE. 21, 47

[90] Li, S., Zou, C., Li, Y., Zhao, X., and Gao, Y.: *Attention-based Multi-Modal Fusion Network for Semantic Scene Completion.* In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 11402–11409, Apr. 2020. `https://ojs.aaai.org/index.php/AAAI/article/view/6803`. 85, 88

[91] Lin, D., Fidler, S., and Urtasun, R.: *Holistic scene understanding for 3d object detection with rgbd cameras.* In *Proceedings of the IEEE international conference on computer vision*, pp. 1417–1424, 2013. 41

[92] Lin, G., Milan, A., Shen, C., and Reid, I.: *RefineNet: Multi-Path Refinement Networks for High-Resolution Semantic Segmentation.* In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 39, 86, 87, 89, 144, 145

[93] Lin, W., Gao, Z., and Li, B.: *Shoestring: Graph-based semi-supervised classification with severely limited labeled data.* In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 97

[94] Liu, C., Yuen, J., Torralba, A., Sivic, J., and Freeman, W.T.: *Sift flow: Dense correspondence across different scenes.* In Forsyth, D., Torr, P., and Zisserman, A. (eds.): *Computer Vision – ECCV 2008*, pp. 28–42, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg, ISBN 978-3-540-88690-7. 30

[95] Liu, F., Li, S., Zhang, L., Zhou, C., Ye, R., Wang, Y., and Lu, J.: *3DCNN-DQN-RNN: A deep reinforcement learning framework for semantic parsing of large-scale 3D point clouds.* In *Proceedings of 16th International Conference on Computer Vision (ICCV), Venice, Italy*, pp. 5679–5688, Piscataway, NJ, Oct 2017. IEEE. 5, 47

[96] Liu, S., HU, Y., Zeng, Y., Tang, Q., Jin, B., Han, Y., and Li, X.: *See and think: Disentangling semantic scene completion.* In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.): *Procedings of Conference on Neural Information Processing Systems 31 (NIPS)*, pp. 263–274, Reed Hook, NY, 2018. Curran Associates, Inc. `http://papers.nips.cc/paper/7310-see-and-think-disentangling-semantic-scene-completion`. 2, 3, 4, 5, 46, 74, 75, 80, 81, 94, 103, 147, 148, 153, 154

[97] Long, J., Shelhamer, E., and Darrell, T.: *Fully convolutional networks for semantic segmentation.* In *Proceedings of of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3431–3440, June 2015. 39

[98] Long, M., Wang, J., Ding, G., and Yu, P.: *Transfer learning with joint distribution adaptation.* In *Proceedings of 14th International Conference on Computer Vision, Sydney, Australia*, pp. 2200–2207, 2013. 36, 67

[99] Lowe, D.G.: *Distinctive image features from scale-invariant keypoints.* International journal of computer vision, 60(2):91–110, 2004. 17

[100] Lumini, A. and Nanni, L.: *Fair comparison of skin detection approaches on publicly available datasets.* Techn. rep., Cornell University Library, CoRR/cs.CV, August 2019. arXiv:1802.02531 (v3). 51, 67

[101] Mahmoodi, M.R. and Sayedi, S.M.: *A comprehensive survey on human skin detection.* International Journal of Image, Graphics & Signal Processing, 8(5):1–35, 2016. 51

[102] Marr, D.: *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information.* MIT Press, 1982, ISBN 978-0-262-51462-0. 1

[103] Mattausch, O., Panozzo, D., Mura, C., Sorkine-Hornung, O., and Pajarola, R.: *Object detection and classification from large-scale cluttered indoor scans.* In *Computer Graphics Forum*, vol. 33, pp. 11–21. Wiley Online Library, 2014. 40

[104] Milletari, F., Navab, N., and Ahmadi, S.A.: *V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation.* In *Fourth International Conference on 3D Vision (3DV)*, pp. 565–571, 2016. 31

[105] Minsky, M. and Papert, S.A.: *Perceptrons: An Introduction to Computational Geometry.* The MIT Press, Sept. 2017, ISBN 9780262343930. `https://doi.org/10.7551/mitpress/11301.001.0001`. 26

[106] Monszpart, A., Mellado, N., Brostow, G.J., and Mitra, N.J.: *Rapter: Rebuilding man-made scenes with regular arrangements of planes.* ACM Trans. Graph., 34(4), jul 2015, ISSN 0730-0301. `https://doi.org/10.1145/2766995`. 40

[107] Murdock, C., Li, Z., Zhou, H., and Duerig, T.: *Blockout: Dynamic model selection for hierarchical deep networks.* In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2583–2591, 2016. 89

[108] Murphy, K.: *Machine learning: a probabilistic perspective.* MIT press, Cambridge, Massachusetts, 2012, ISBN 978-0-262-01802-9. 36

[109] Nguyen, D.T., Hua, B., Tran, M., Pham, Q., and Yeung, S.: *A field model for repairing 3D shapes.* In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, June 26 - July 1*, pp. 5676–5684, Piscataway, NJ, June 2016. IEEE. 2

[110] Pan, S.J. and Yang, Q.: *A Survey on Transfer Learning.* IEEE Transactions on Knowledge and Data Engineering, 22(10):1345–1359, Oct. 2010, ISSN 1041-4347. `https://doi.org/10.1109/TKDE.2009.191`. 34, 35

[111] Pandey, R.K., Vasan, A., and Ramakrishnan, A.G.: *Segmentation of liver lesions with reduced complexity deep models.* Techn. rep., Cornell University Library, CoRR/cs.CV, 2018. `http://arxiv.org/abs/1805.09233`, arXiv:1805.09233. 31

[112] Papert, S.: *The Summer Vision Project.* AI memo. Massachusetts Institute of Technology, Project MAC, 1966. `https://books.google.com.br/books?id=qOh7NwAACAAJ`. 24

[113] Park, S.J., Hong, K.S., and Lee, S.: *RDFNet: RGB-D multi-level residual feature fusion for indoor semantic segmentation.* In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct 2017. 39, 86, 145

[114] Perez, L. and Wang, J.: *The effectiveness of data augmentation in image classification using deep learning.* Techn. rep., Cornell University Library, CoRR/cs.CV, 2017. `http://arxiv.org/abs/1712.04621`, arXiv:1712.04621. 31

[115] Qi, C.R., Yi, L., Su, H., and Guibas, L.J.: *PointNet++: Deep hierarchical feature learning on point sets in a metric space.* In Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.): *Procedings of Conference on Neural Information Processing Systems 30 (NIPS)*, pp. 5099–5108. Curran Associates, Inc., Reed Hook, NY, 2017. `http://papers.nips.cc/paper/7095-pointnet-deep-hierarchical-feature-learning-on-point-sets-in-a-metric-space`. 5, 47

[116] Qi, X., Liao, R., Jia, J., Fidler, S., and Urtasun, R.: *3D graph neural networks for RGBD semantic segmentation.* In *Proceedings of 16th International Conference on Computer Vision (ICCV), Venice, Italy*, pp. 5209–5218, Piscataway, NJ, Oct. 2017. IEEE. 2, 39

[117] Ranzato, M. and Szummer, M.: *Semi-supervised learning of compact document representations with deep networks.* In *Proceedings of the 25th international conference on Machine learning - ICML*, pp. 792–799, Helsinki, Finland, 2008. ACM Press, ISBN 978-1-60558-205-4. `https://doi.org/10.1145/1390156.1390256`. 36

[118] Ren, X., Bo, L., and Fox, D.: *RGB-(D) scene labeling: Features and algorithms.* In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI, June 16-21*, pp. 2759–2766, Piscataway, NJ, June 2012. IEEE. 2, 39

[119] Roberts, L.G.: *Machine perception of three-dimensional solids.* PhD thesis, Massachusetts Institute of Technology, 1963. 24

[120] Ronneberger, O., Fischer, P., and Brox, T.: *U-Net: Convolutional networks for biomedical image segmentation.* In Navab, N., Hornegger, J., Wells, W.M., and Frangi, A.F. (eds.): *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pp. 234–241, Cham, 2015. Springer International Publishing, ISBN 978-3-319-24574-4. 30, 55, 71, 89

[121] Rosenblatt, F.: *The perceptron: a probabilistic model for information storage and organization in the brain.* Psychological Review, 65(6):19S8, 1958. 25, 26

[122] Rosenfeld, A., Hummel, R.A., and Zucker, S.W.: *Scene labeling by relaxation operations.* IEEE Transactions on Systems, Man, and Cybernetics, SMC-6(6):420–433, June 1976, ISSN 2168-2909. 2

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

[124] Rumelhart, D.E., Hinton, G.E., and Williams, R.J.: *Learning representations by back-propagating errors.* nature, 323(6088):533–536, 1986. 26

[125] 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.: *ImageNet large scale visual recognition challenge.* Int Journal of Computer Vision (IJCV), 115(3):211–252, 2015. `https://doi.org/10.1007/s11263-015-0816-y`. 3, 25, 35

[126] Saito, K., Ushiku, Y., and Harada, T.: *Asymmetric tri-training for unsupervised domain adaptation.* In Precup, D. and Teh, Y.W. (eds.): *Proceedings of the 34th International Conference on Machine Learning*, vol. 70 of *Proceedings of Machine Learning Research*, pp. 2988–2997, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR. `http://proceedings.mlr.press/v70/saito17a.html`. 97

[127] San Miguel, J.C. and Suja, S.: *Skin detection by dual maximization of detectors agreement for video monitoring.* Pattern Recognition Letters, 34(16):2102 – 2109, 2013, ISSN 0167-8655. 51, 58, 60, 63

[128] Scharstein, D., Hirschmüller, H., Kitajima, Y., Krathwohl, G., Nešić, N., Wang, X., and Westling, P.: *High-resolution stereo datasets with subpixel-accurate ground truth.* In *German conference on pattern recognition*, pp. 31–42. Springer, 2014. 12

[129] Scharstein, D. and Pal, C.: *Learning conditional random fields for stereo.* In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2007. 11

[130] Schoenbein, M. and Geiger, A.: *Omnidirectional 3D reconstruction in augmented manhattan worlds.* In *Proceedings of IEEE/RSJ Conference on Intelligent Robots and Systems IROS*, pp. 716 – 723, Piscataway, NJ, 2014. IEEE. 47

[131] Sejnowski, T.J.: *The deep learning revolution.* MIT press, 2018. 23, 24, 26

[132] Sener, O., Song, H.O., Saxena, A., and Savarese, S.: *Learning transferrable representations for unsupervised domain adaptation.* In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, p. 2118–2126, Red Hook, NY, USA, 2016. Curran Associates Inc., ISBN 9781510838819. 97

[133] Shaik, K.B., Ganesan, P., Kalist, V., Sathish, B., and Jenitha, J.M.M.: *Comparative Study of Skin Color Detection and Segmentation in HSV and YCbCr Color Space.* Procedia Computer Science, 57:41–48, 2015, ISSN 18770509. `https://doi.org/10.1016/j.procs.2015.07.362`. 51, 63

[134] Shelhamer, E., Long, J., and Darrell, T.: *Fully convolutional networks for semantic segmentation.* IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 39(4):640–651, 2017, ISSN 0162-8828, 2160-9292. `https://doi.org/10.1109/TPAMI.2016.2572683`, First appeared as a preprint in 2014 at arXiv:1411.4038. 2, 30, 51

[135] Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., and Blake, A.: *Real-time human pose recognition in parts from single depth images.* In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Colorado Springs, CO, June 20-25*, pp. 1297–1304, 2011. 51

[136] Shotton, J., Johnson, M., and Cipolla, R.: *Semantic texton forests for image categorization and segmentation.* In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Anchorage, AK, June 24-26*, pp. 1–8, 2008. 51

[137] Shrivastava, A. and Mulam, H.: *Building part-based object detectors via 3D geometry.* In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Portland, OR, June 23-28*, pp. 1745–1752, Piscataway, NJ, Dec. 2013. IEEE. 2

[138] Shrivastava, V.K., Londhe, N.D., Sonawane, R.S., and Suri, J.S.: *Computer-aided diagnosis of psoriasis skin images with HOS, texture and color features.* Comput. Methods Prog. Biomed., 126(C):98–109, Apr. 2016, ISSN 0169-2607. 51

[139] Silberman, N. and Fergus, R.: *Indoor scene segmentation using a structured light sensor.* In *IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pp. 601–608, Piscataway, NJ, Nov 2011. IEEE. 2, 30

[140] Silberman, N., Hoiem, D., Kohli, P., and Fergus, R.: *Indoor segmentation and support inference from RGBD images.* In Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., and Schmid, C. (eds.): *Proceedings of 12th European Conference on Computer Vision (ECCV), Florence, Italy, October 7-13*, pp. 746–760, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg, ISBN 978-3-642-33715-4. 22, 30, 38, 39, 44, 48, 73, 87, 91, 99, 108, 112

[141] Smith, L.N.: *A disciplined approach to neural network hyper-parameters: Part 1 - learning rate, batch size, momentum, and weight decay.* Tech. Rep. arXiv:1803.09820, Cornell University Library, 2018. `http://arxiv.org/abs/1803.09820`. 4, 73, 91, 100

[142] Smith, S., Kindermans, P. jan, Ying, C., and Le, Q.V.: *Don't decay the learning rate, increase the batch size.* In *Proceedings of Int Conf Learning Representations (ICLR)*, 2018. 80

[143] Son Lam Phung, Bouzerdoum, A., and Chai, D.: *A novel skin color model in ycbcr color space and its application to human face detection.* In *Proceedings. International Conference on Image Processing*, vol. 1, pp. I–I, Sep. 2002. 63

[144] Song, S., Lichtenberg, S.P., and Xiao, J.: *Sun rgb-d: A rgb-d scene understanding benchmark suite.* In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. 1

[145] Song, S., Lichtenberg, S.P., and Xiao, J.: *Sun rgb-d: A rgb-d scene understanding benchmark suite.* In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 567–576, 2015. 48

[146] Song, S. and Xiao, J.: *Sliding shapes for 3d object detection in depth images.* In *European conference on computer vision*, pp. 634–651. Springer, 2014. 40

[147] Song, S. and Xiao, J.: *Deep sliding shapes for amodal 3d object detection in rgb-d images.* In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 808–816, 2016. 41

[148] Song, S., Yu, F., Zeng, A., Chang, A.X., Savva, M., and Funkhouser, T.: *Semantic Scene Completion from a Single Depth Image.* In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, Hawaii, July 21-26*, pp. 190–198, Piscataway, NJ, July 2017. IEEE. 2, 3, 4, 22, 32, 41, 42, 44, 45, 48, 49, 68, 69, 73, 74, 75, 86, 89, 91, 94, 95, 99, 103, 112, 114, 147, 148, 153, 154

[149] Song, S., Zeng, A., Chang, A.X., Savva, M., Savarese, S., and Funkhouser, T.: *Im2Pano3D: Extrapolating* 360° *structure and semantics beyond the field of view.* In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, June 18-22*, pp. 3847–3856, Piscataway, NJ, June 2018. IEEE. 47

[150] Stutz, D. and Geiger, A.: *Learning 3d shape completion under weak supervision.* International Journal of Computer Vision, 128(5):1162–1181, 2020. 40

[151] Szegedy, C., Ioffe, S., and Vanhoucke, V.: *Inception-v4, inception-resnet and the impact of residual connections on learning.* CoRR, abs/1602.07261, 2016. `http://arxiv.org/abs/1602.07261`. 29

[152] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z.: *Rethinking the Inception Architecture for Computer Vision.* In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, June 26 - July 1*, pp. 2818 – 2826, 2016. 31

[153] Vasconcelos, C.N. and Vasconcelos, B.N.: *Increasing deep learning melanoma classification by classical and expert knowledge based image transforms.* Techn. rep., Cornell University Library, CoRR/cs.CV, 2017. `http://arxiv.org/abs/1702.07025`, arXiv:/1711.03954. 31

[154] Vision, S. Centre for and Surrey, S.P.U. of: *S3a audio-visual scene analysis datasets and resources.* `https://cvssp.org/data/s3a/public/AV_Analysis/index.html`, 2018. Acessed: 2020-06-19. 112

[155] Wang, W. and Neumann, U.: *Depth-aware CNN for RGB-D segmentation.* In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018. 39

[156] Wang, Y., Tan, D.J., Navab, N., and Tombari, F.: *Forknet: Multi-branch volumetric semantic completion from a single depth image.* In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8607–8616, 2019. 85, 94, 103, 147, 148, 153, 154

[157] Weston, J., Ratle, F., and Collobert, R.: *Deep learning via semi-supervised embedding.* In *Proceedings of the 25th International Conference on Machine Learning*, ICML, pp. 1168–1175, New York, NY, USA, 2008. ACM, ISBN 978-1-60558-205-4. 36

[158] Winston, P.: *The mit robot'in machine intelligence vol. d. michie and b. meltzer*, 1972. 24

[159] Wong, S.C., Gatt, A., Stamatescu, V., and McDonnell, M.D.: *Understanding data augmentation for classification: when to warp?* In *International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, pp. 1–6. IEEE, 2016. 31

[160] Wu, H., Xiao, B., Codella, N., Liu, M., Dai, X., Yuan, L., and Zhang, L.: *Cvt: Introducing convolutions to vision transformers.* In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22–31, October 2021. 127

[161] Xu, Y., Jia, R., Mou, L., Li, G., Chen, Y., Lu, Y., and Jin, Z.: *Improved relation classification by deep recurrent neural networks with data augmentation.* In *26th International Conference on Computational Linguistics (COLING)*, pp. 1461–1470, 2016. Preprint available at arXiv:1601.03651. 31

[162] Yang, Q., Wang, L., Yang, R., Stewénius, H., and Nistér, D.: *Stereo matching with color-weighted correlation, hierarchical belief propagation, and occlusion handling.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 31(3):492–504, 2009. 20

[163] Yogarajah, P., Condell, J., Curran, K., Cheddad, A., and McKevitt, P.: *A dynamic threshold approach for skin segmentation in color images.* In *Proceedings of International Conference on Image Processing, Hong Kong, September 26-29*, pp. 2225–2228, Sept 2010. 58

[164] Yu, F. and Koltun, V.: *Multi-scale context aggregation by dilated convolutions.* In Bengio, Y. and LeCun, Y. (eds.): *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016. `http://arxiv.org/abs/1511.07122`. 43

[165] Zhai, X., Oliver, A., Kolesnikov, A., and Beyer, L.: *S4l: Self-supervised semi-supervised learning.* In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 97

[166] Zhang, J., Zhao, H., Yao, A., Chen, Y., Zhang, L., and Liao, H.: *Efficient semantic scene completion network with spatial group convolution.* In *Proceedings of 15th*

European Conference on Computer Vision (ECCV), Munich, Germany, September 8-14, pp. 749–765, Cham, September 2018. Springer International Publishing, ISBN 978-3-030-01258-8. 3, 93, 114, 147, 148, 153, 154

[167] Zhang, L., Wang, L., Zhang, X., Shen, P., Bennamoun, M., Zhu, G., Shah, S.A.A., and Song, J.: *Semantic scene completion with dense CRF from a single depth image.* Neurocomputing, 318:182–195, Nov. 2018, ISSN 09252312. `https://doi.org/10.1016/j.neucom.2018.08.052`. 2, 4, 22, 45, 74, 75, 147, 148, 154

[168] Zhang, P., Liu, W., Lei, Y., Lu, H., and Yang, X.: *Cascaded context pyramid for full-resolution 3d semantic scene completion.* In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7800–7809, 2019. 3, 85, 93, 94, 103, 147, 148, 153, 154

[169] Zhang, W., Ouyang, W., Li, W., and Xu, D.: *Collaborative and adversarial network for unsupervised domain adaptation.* In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 97

[170] Zheng, B., Zhao, Y., Yu, J.C., Ikeuchi, K., and Zhu, S.C.: *Beyond point clouds: Scene understanding by reasoning geometry and physics.* In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3127–3134, 2013. 42

[171] Zheng, S., Jayasumana, S., Romera-Paredes, B., Vineet, V., Su, Z., Du, D., Huang, C., and Torr, P.H.: *Conditional random fields as recurrent neural networks.* In *Proceedings of 15th International Conference on Computer Vision, Santiago, Chile*, pp. 1529–1537, 2015. 30

[172] Zhong, Z., Zheng, L., Kang, G., Li, S., and Yang, Y.: *Random erasing data augmentation.* In *AAAI Conference on Artificial Intelligence*, vol. 34, pp. 13001–13008, April 2020. `https://ojs.aaai.org/index.php/AAAI/article/view/7000`. 89

[173] Zhu, X.: *Semi-supervised learning literature survey.* Tech. Rep. 1530, Computer Sciences, University of Wisconsin-Madison, 2005. 36

Appendix A

SPAwN Supplementary Details

This appendix contains extra information and resources about SPAwN presented on Chapter 6, as listed below.

- Additional information regarding the experiments presented in the main document.
- Complete comparison tables including all previous Semantic Scene Completion (SSC) approaches that we are aware of.
- Additional figures with images for a qualitative evaluation of the results on the three evaluated datasets.

A.1 Additional experimental details about the semantic segmentation method

To produce the results presented in the main document, we trained two different 2D semantic segmentation networks. The results of the *"depth + RGB"* models presented in the ablation study (Table 6.1) were produced using a simplified version of RefineNet [92] (single-mode) which architecture is shown in Figure A.1. The results of the *"depth + RGB + surface normals"* models presented in the ablation study and in the result tables were produced using a bimodal 2D segmentation network. Its architecture was presented in the main document. For convenience, we show it here again in Figure A.2.

The main difference between these two networks is the addition of surface normals as a second input mode and a corresponding second pre-trained ResNet-101 backbone. MMF modules are also added to combine the features from the two modes.

Figure A.3 presents the learning curves of the two models regarding the fine-tuning stage of training. In that stage, the ResNet backbones weights are unfrozen. Note that the bimodal model takes a little longer to stabilize compared to the single-mode one. This

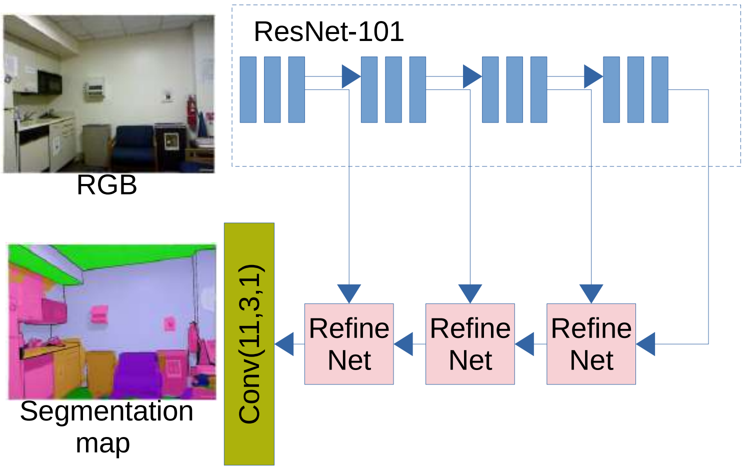


Figure A.1: **2D single-mode segmentation network architecture.** This is a simplified version of RefineNet [92].

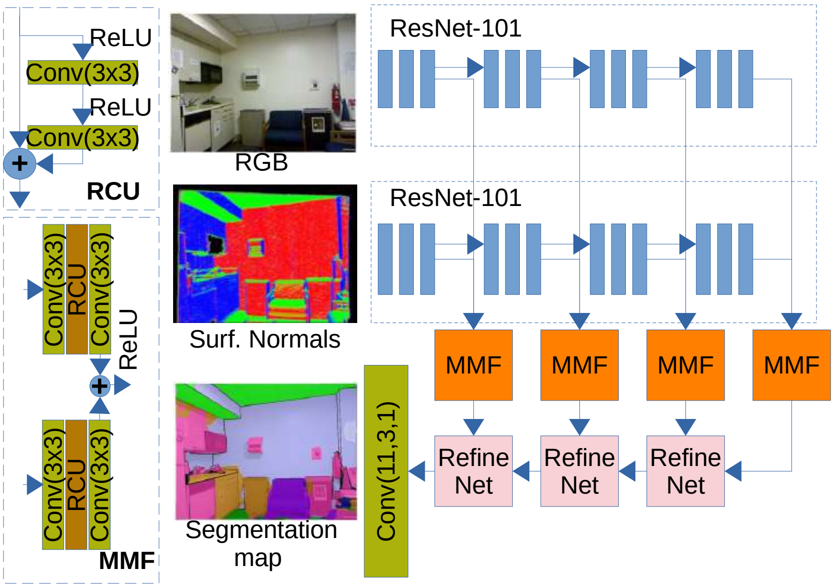


Figure A.2: **2D bimodal segmentation network architecture.** It is shown here again to facilitate comparison with the unimodal network of Figure A.1. The Residual Convolution Unit (RCU) and the RefineNet module were first defined in [92]. Here, we use a simplified MMF block [113].

is somewhat expected, since the ResNet-101 backbones are pre-trained on RGB images, and the surfaces normals represent a completely different domain. However, the bimodal model achieves a better validation final score, even though the train mIoU of the single-mode model is better. This indicates that adding surface normals as input helps reduce model overfitting.

In Table A.1 we present the semantic segmentation results of the 2D models. As expected, the better learning curve of the bimodal network leads to better per class results.

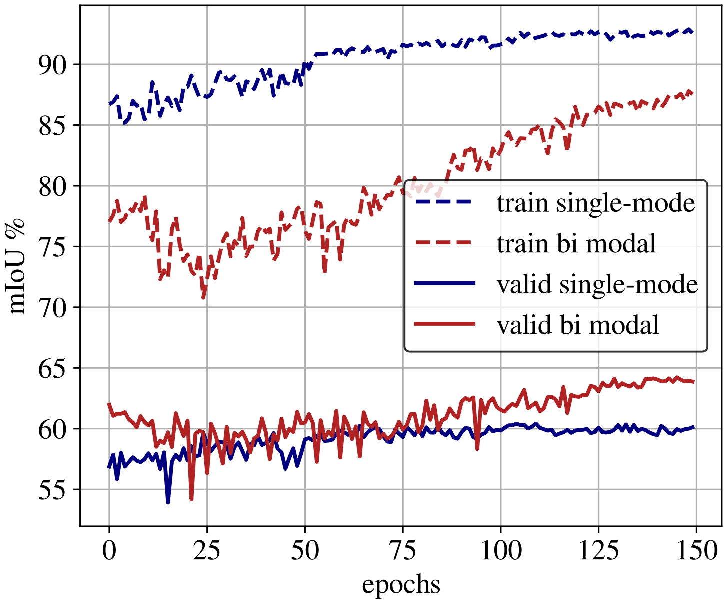


Figure A.3: **Learning curves** of the fine-tune stage of 2D segmentation networks on NYDv2 (no pre-training on SUNCG).

training set	model	ceil.	floor	wall	win.	chair	bed	sofa	table	tvs	furn.	objs.	avg.
NYUDv2	Single-mode	64.1	83.8	75.7	62.5	**75.0**	62.8	58.3	38.9	52.2	57.1	54.4	60.6
	Bimodal	**73.3**	**89.2**	**76.7**	**62.9**	63.4	**67.6**	**62.1**	**40.3**	**56.7**	**58.7**	**55.5**	**64.2**
SUNCG → NYUDv2	Single-mode	-	-	-	-	-	-	-	-	-	-	-	-
	Bimodal	**74.6**	**90.6**	**77.4**	**64.7**	**64.2**	**72.0**	**62.9**	**43.8**	**54.5**	**58.7**	**56.7**	**65.5**

(Columns ceil. through avg.: 2D RGB-D semantic segmentation (IoU, in percentages))

Table A.1: **Semantic segmentation results of 2D models on NYUDv2 test set**. For each model we show per class segmentation IoU and the average score. The bimodal network is superior in average and in most of the classes. We did not test fine-tuning from SUNCG in single-mode setup.

A.2 Overfitting reduction using the proposed data augmentation approach

Figure A.4 presents the training and validation learning curves of SPAwN on NYUDv2, with and without data augmentation. Note the inversion of the positions of the red and blue curves when data augmentation is used. Although the regular train curve (blue/dashed) reaches a higher score at the end of the training when compared to the data augmented curve (red/dashed), the final data augmented score in validation (red/solid) is higher than the regular curve (blue/solid). This indicates overfitting reduction due to data augmentation.

Also note that regular training starts overfitting around the 76th epoch, while the data augmented validation score keeps raising until the 118th epoch. This indicates that training can go on for more epochs to reach better results using our data augmented models.

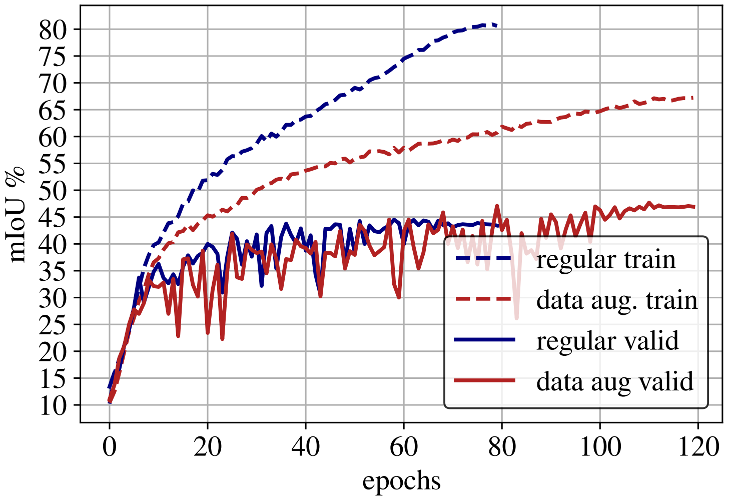


Figure A.4: **Learning curves** of the training on NYUDv2 with and without data augmentation (no pre-training on SUNCG).

A.3 Complete comparison tables

Tables A.2, A.3, and A.4 complement Tables 2, 3 and 4 of the main document (respectively) and compare our results to all the competing SSC solutions that we are aware of. Note the superiority of our method among all straight-forward methods. It is worth mentioning that our data augmentation strategies can be applied with other methods, such as the SISNet, certainly leading to further improvement.

model	pipeline type	scene completion			semantic scene completion (IoU, in percentages)											
		prec.	rec.	IoU	ceil.	floor	wall	win.	chair	bed	sofa	table	tvs	furn.	objs.	avg.
SISNet-BiSeNet [15]	iterative	93.3	96.1	89.9	85.2	90.0	83.7	80.8	60.0	83.5	80.8	68.6	77.3	86.7	70.1	78.8
SISNet-DeepLabv3 [15]		92.6	96.3	89.3	85.4	90.6	82.6	80.9	62.9	84.5	82.6	71.6	72.6	85.6	69.7	79.0
SSCNet[148]	straight-forward	76.3	95.2	73.5	96.3	84.9	56.8	28.2	21.3	56.0	52.7	33.7	10.9	44.3	25.4	46.4
TNetFuse[96]		53.9	95.2	52.6	60.6	57.3	53.2	52.7	27.4	46.8	53.3	28.6	41.1	44.1	29.0	44.9
DCRF[167]		-	-	-	95.4	84.3	57.7	24.5	28.2	63.4	55.3	34.5	19.6	45.8	28.7	48.8
ForkNet[156]		-	-	-	95.0	85.9	73.2	54.5	46.0	81.3	74.2	42.8	31.9	63.1	49.3	63.6
VVNet[51]		90.8		91.7	84.0	61.0	54.8	49.3	83.0	75.5	55.1	43.5	68.8	57.7	66.7	
EdgeNet[34]		<u>93.3</u>	90.6	<u>85.1</u>	97.2	<u>95.3</u>	<u>78.2</u>	57.5,	51.4	80.7	74.1	54.5	52.6	70.3	60.1	70.2
ESSC[166]		92.6	90.4	84.5	96.6	83.7	74.9	59.0	55.1	<u>83.3</u>	78.0	61.5	47.4	73.5	62.9	70.5
CCPNet[168]		**98.2**	**96.8**	**91.4**	<u>99.2</u>	89.3	76.2	<u>63.3</u>	<u>58.2</u>	**86.1**	**82.6**	<u>65.6</u>	<u>53.2</u>	**76.8**	<u>65.2</u>	<u>74.2</u>
SPAwN (ours)		91.9	88.7	82.3	**99.3**	**96.1**	**84.4**	**75.1**	**59.2**	81.5	<u>78.1</u>	**67.3**	**80.1**	<u>76.3</u>	**70.4**	**78.9**

Table A.2: **Results on SUNCG test set**. Our SPAwN semantic scene completion overall results with regular training (not augmented) surpasses by far all known previous solutions on SUNCG synthetic images with straightforward pipeline and gets close to much more complex SISNet models (complementing Table 2 of the main paper).

model	pipeline type	train	scene compl.			semantic scene completion (IoU, in percentages)											
			prec.	rec.	IoU	ceil.	floor	wall	win.	chair	bed	sofa	table	tvs	furn.	objs.	avg.
SISNet-BiSeNet[15]	iterative	NYU	90.7	84.6	77.8	53.9	93.2	51.3	38.0	38.7	65.0	56.3	37.8	25.9	51.3	36.0	49.8
SISNet-DLabv3[15]			92.1	83.8	78.2	54.7	93.8	53.2	41.9	43.6	66.2	61.4	38.1	29.8	53.9	40.3	52.4
SSCNet[148]	straight-forward	NYU	57.0	**94.5**	55.1	15.1	94.7	24.4	0.0	12.6	32.1	35.0	13.0	7.8	27.1	10.1	24.7
ESSCNet[166]			71.9	71.9	56.2	17.5	75.4	25.8	6.7	15.3	53.8	42.4	11.2	0.0	33.4	11.8	26.7
EdgeNet[34]			76.0	68.3	65.1	17.9	94.0	27.8	2.1	9.5	51.8	44.3	9.4	3.6	32.5	12.7	27.8
DDRNet[88]			71.5	80.8	61.0	21.1	92.2	33.5	6.8	14.8	48.3	42.3	13.2	13.9	35.3	13.2	30.4
DCRF[167]			-	-	-	18.1	92.6	27.1	10.8	18.8	54.3	47.9	17.1	15.1	34.7	13.0	31.8
TS3D[45]			-	-	60.0	9.7	93.4	25.5	21.0	17.4	55.9	49.2	17.0	27.5	39.4	19.3	34.1
CCPNet[168]			**91.3**	92.6	**82.4**	23.5	**96.3**	35.7	20.2	25.8	61.4	56.1	18.1	28.1	37.8	20.1	38.5
SketchAware[23]			85.0	81.6	71.3	**43.1**	93.6	**40.5**	24.3	30.0	57.1	49.3	29.2	14.3	42.5	28.6	41.1
SPAwN			82.0	74.2	63.8	36.3	94.0	38.3	26.1	33.7	61.2	54.8	25.1	35.0	43.5	29.6	43.4
SPAwN+DA			80.8	77.8	65.7	41.5	94.2	38.0	**30.4**	40.3	69.6	57.2	29.4	**41.4**	48.8	34.1	47.7
SPAwN+DA+TTDA			82.3	77.2	66.2	41.5	94.3	38.2	30.3	**41.0**	**70.6**	**57.7**	**29.7**	40.9	**49.2**	**34.6**	**48.0**
SSCNet[148]	straight-forward	SUNCG + NYU	59.3	92.9	56.6	15.1	94.6	24.7	10.8	17.3	53.2	45.9	15.9	13.9	31.1	12.6	30.5
CSSCNet[49]			62.5	82.3	54.3	-	-	-	-	-	-	-	-	-	-	-	30.5
DCRF[167]			-	-	-	18.1	92.6	27.1	10.8	18.8	54.3	47.9	17.1	15.1	34.7	13.0	31.8
VVNet[51]			**86.4**	92.0	**80.3**	19.3	94.8	28.0	12.2	19.6	57.0	50.5	17.6	11.9	35.6	15.3	32.9
TNetFuse[96]			67.3	85.8	60.6	17.3	92.1	28.0	16.6	19.3	57.5	53.8	17.7	18.5	38.4	18.9	34.4
ForkNet[156]			-	-	-	36.2	93.8	29.2	18.9	17.7	61.6	52.9	23.3	19.5	45.4	20.0	37.1
CCPNet[168]			78.8	**94.3**	67.1	25.5	**98.5**	38.8	27.1	27.3	64.8	**58.4**	21.5	30.1	38.4	23.8	41.3
SPAwN			77.6	82.6	66.7	**47.3**	93.4	**41.3**	28.9	41.6	**69.5**	57.1	**33.1**	30.9	50.9	35.0	48.1
SPAwN+DA			79.8	80.8	67.1	44.1	94.0	39.9	31.5	41.6	67.4	57.3	32.5	42.8	52.5	35.0	49.0
SPAwN+DA+TTDA			81.2	80.4	67.8	44.2	94.2	40.9	**33.5**	**42.5**	69.3	**58.4**	32.4	**44.3**	**53.4**	**36.3**	**49.9**

Table A.3: **Results on NYUDv2 test set** . The column "train" indicates datasets used for training the models. SUNCG + NYU means trained on SUNCG and fine-tuned on NYUDv2. In both scenarios, our SPAwN semantic scene completion overall surpasses all known previous solutions with straightforward pipeline and gets close to much more complex SISNet models (complementing Table 3 of the main paper).

model	pipeline type	train	scene compl.			semantic scene completion (IoU, in percentages)											
			prec.	rec.	IoU	ceil.	floor	wall	win.	chair	bed	sofa	table	tvs	furn.	objs.	avg.
SISNet-BiSeNet[15]	iterative	NYUCAD	94.2	91.3	86.5	65.6	94.4	67.1	45.2	57.2	75.5	66.4	50.9	31.1	62.5	42.9	59.9
SISNet-DLabv3[15]			94.1	91.2	86.3	63.4	94.4	67.2	52.4	59.2	77.9	71.1	58.1	46.2	65.8	48.8	63.5
DCRF[167]	straight-forward	NYUCAD	-	-	-	35.5	92.6	52.4	10.7	40.0	60.0	62.5	34.0	9.4	49.2	26.5	43.0
TS3D[45]			80.2	94.4	76.5	34.4	93.6	47.7	31.8	32.2	65.2	54.2	30.7	32.5	50,1	30.7	45.7
DDRNet[88]			88.7	88.5	79.4	54.1	91.5	56.4	14.9	37.0	55.7	51.0	28.8	9.2	44.1	27.8	42.8
CCPNet[168]			**91.3**	**92.6**	82.4	56.2	96.6	58.7	35.1	44.8	68.6	65.3	37.6	35.5	53.1	35.2	53.2
SketchAware[23]			90.6	92.2	**84.2**	59.7	94.3	**64.3**	32.6	51.7	72.0	68.7	45.9	19.0	60.5	38.5	55.2
SPAwN			83.7	87.2	74.6	64.0	94.6	61.4	33.3	63.1	80.4	**72.8**	47.6	**44.0**	**64.0**	42.7	60.7
SPAwN+DA			82.9	88.0	74.5	65.2	**94.7**	60.9	36.4	69.1	82.0	72.1	48.3	41.4	63.4	43.9	61.6
SPAwN+DA+TTDA			84.5	87.8	75.6	**65.3**	**94.7**	61.9	**36.9**	**69.6**	**82.2**	**72.8**	**49.1**	43.6	63.4	**44.4**	**62.2**
SSCNet[148]	straight-forward	NYUCAD + SUNCG	75.4	**96.3**	73.2	32.5	92.6	40.2	8.9	40.0	60.0	62.5	34.0	9.4	49.2	26.5	40.0
CCPNet[168]			**93.4**	91.2	**85.1**	58.1	**95.1**	60.5	36.8	47.2	69.3	67.7	39.8	37.6	55.4	37.6	55.0
SPAwN			87.7	88.4	78.7	69.9	94.9	67.6	35.0	**68.8**	**82.8**	76.0	53.2	42.4	64.0	45.8	63.7
SPAwN+DA			84.8	90.0	77.6	76.1	94.9	67.2	37.8	67.2	81.7	76.8	55.7	49.9	65.3	46.1	65.3
SPAwN+DA+TTDA			86.3	90.1	78.9	**77.6**	95.0	**68.0**	**38.1**	67.9	82.2	**77.1**	**56.8**	**50.0**	**65.7**	**46.5**	**65.9**

Table A.4: **Results on NYUDCAD**. Our SPAwN models hold the best and second-best overall results on both training scenarios, when compared to previous straight-forward solutions. When fine-tuned from SUNCG, SPAwN surpasses both SISNet models, which are much more complex than ours (complementing Table 4 of the main paper).

A.4 Additional Qualitative Analysis

Figures A.5, A.6, and A.7 complement Figure 6.5 by presenting additional results for a qualitative analysis on SUNCG, NYUDv2, and NYUCAD, respectively. Each row of the figures corresponds to one scene. From top to bottom, we present images of RGB image, depth map, surface normals, 2D predictions (obtained with our bimodal semantic segmentation network), projected visible surface, projected semantic priors, SSC predictions (obtained by our method), and 3D ground truth.

Figures A.5 and A.7 show that our method benefits from the excellent semantic segmentation results, guiding SPAwN to generate outstanding semantic scene completion results, filling the gaps left by simply projecting semantic priors to 3D. In SUNCG, both RGB and depth maps come from synthetic 3D models of the scenes, so the 2D segmentation results approach perfection. Segmentation results are also excellent in NYUCAD, even though the RGB images come from real scenes and there is a level of mismatch between depth maps and RGB textures (see the chairs and the bookshelf on the fourth column of Figure A.7). The surface normals have certainly played an essential role in the quality of our bimodal segmentation CNN.

In Figure A.6, the first column shows that specular surfaces give poor surface normals, and saturation corrupts RGB information. These specular surfaces have perturbed the segmentation priors and generated a poor 3D prediction for the whiteboard. On the other hand, the fourth column of that figure shows that our method has correctly identified the blackboard surface as "objects", even though that region was incorrectly labeled as "wall" in the ground truth.

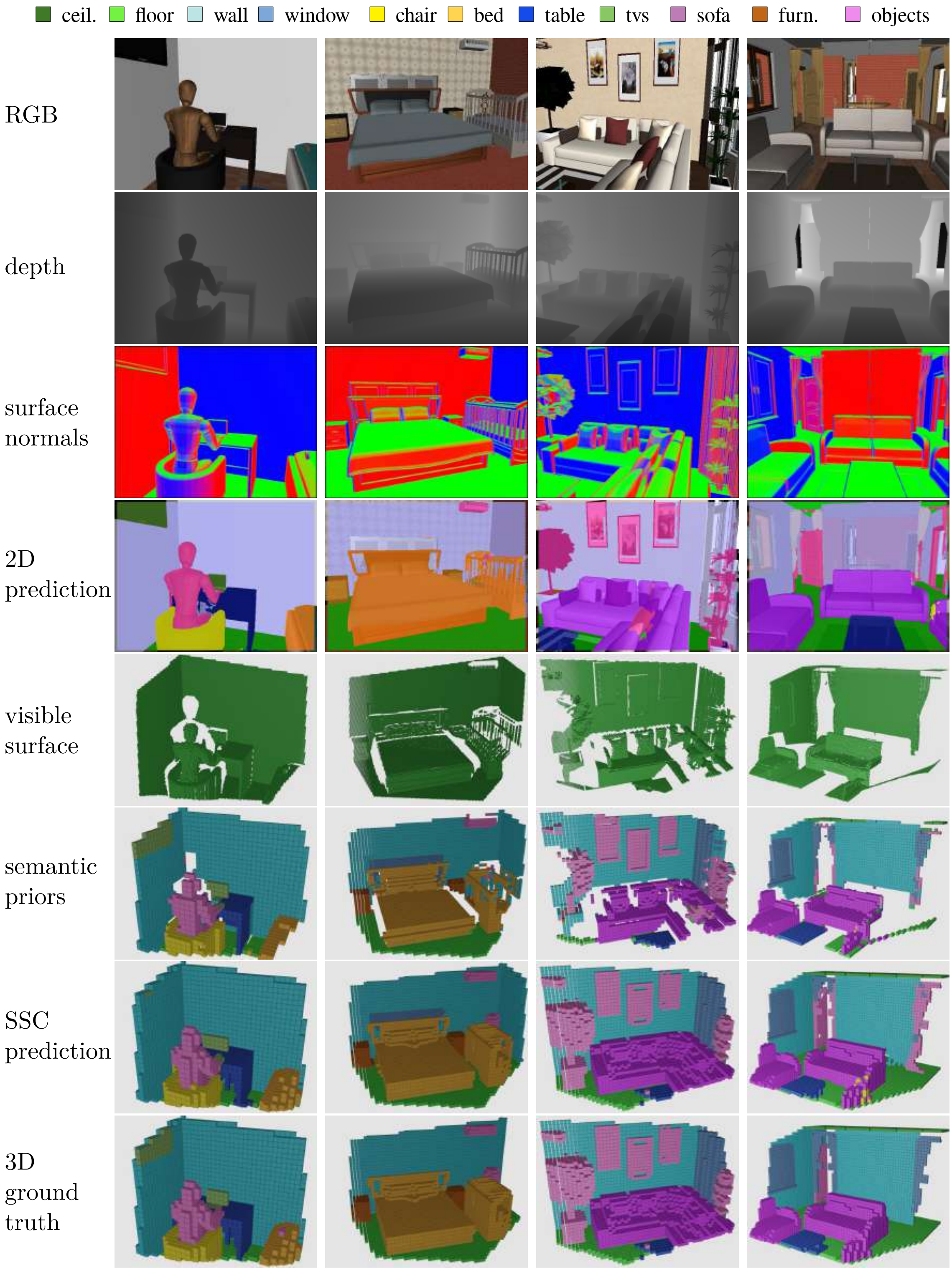


Figure A.5: **Qualitative results on SUNCG.** From top to bottom: RGB, depth, surface normals, 2D segmentation with our bimodal CNN, visible surface, 3D priors, prediction with our SPAwN architecture, ground truth. (Best viewed in color.)

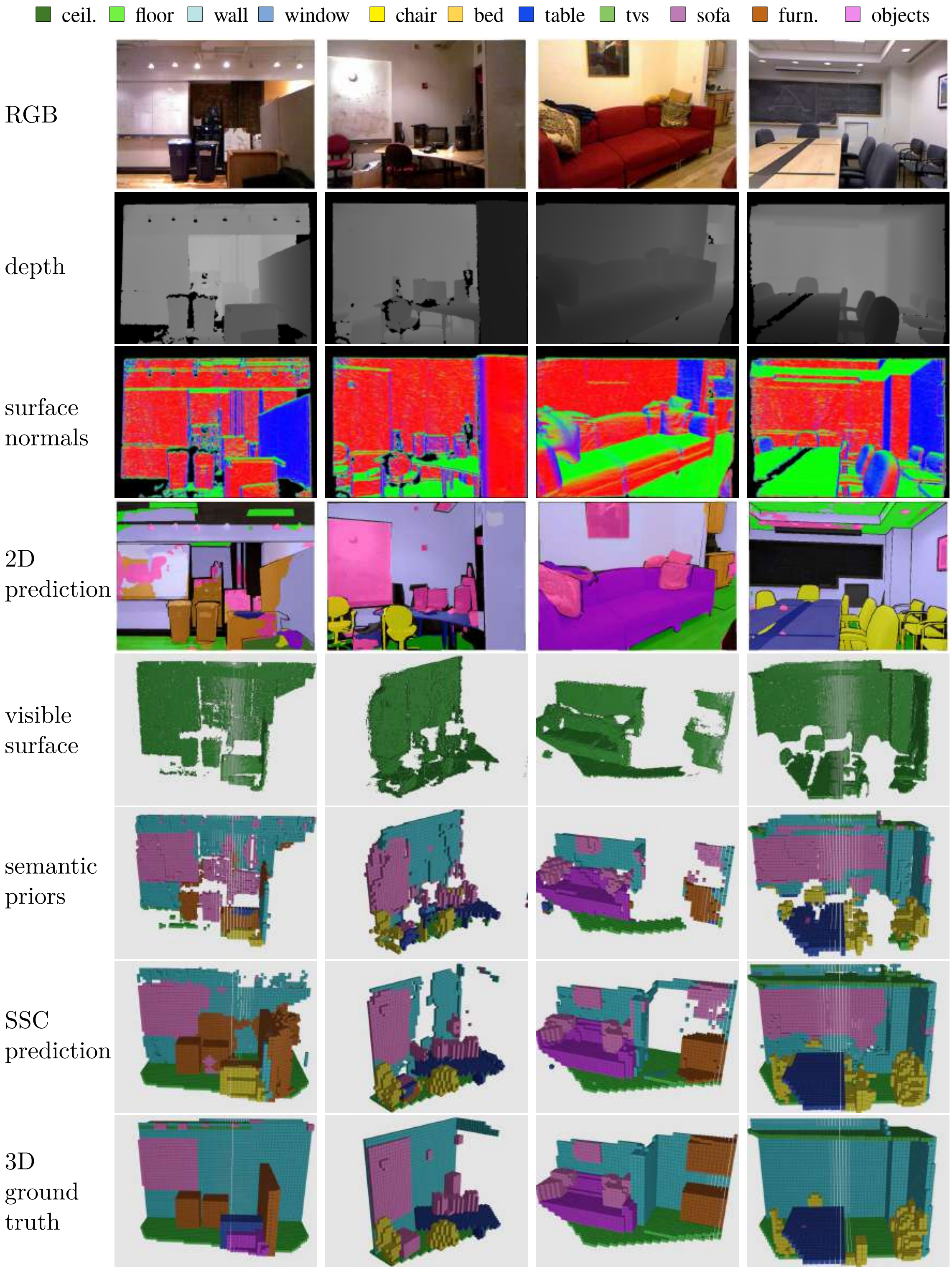


Figure A.6: **Qualitative results on NYUDv2.** From top to bottom: RGB, depth, surface normals, 2D segmentation with our bimodal CNN, visible surface, 3D priors, prediction with SPAwN+S3P, Ground Truth. (Best viewed in color).

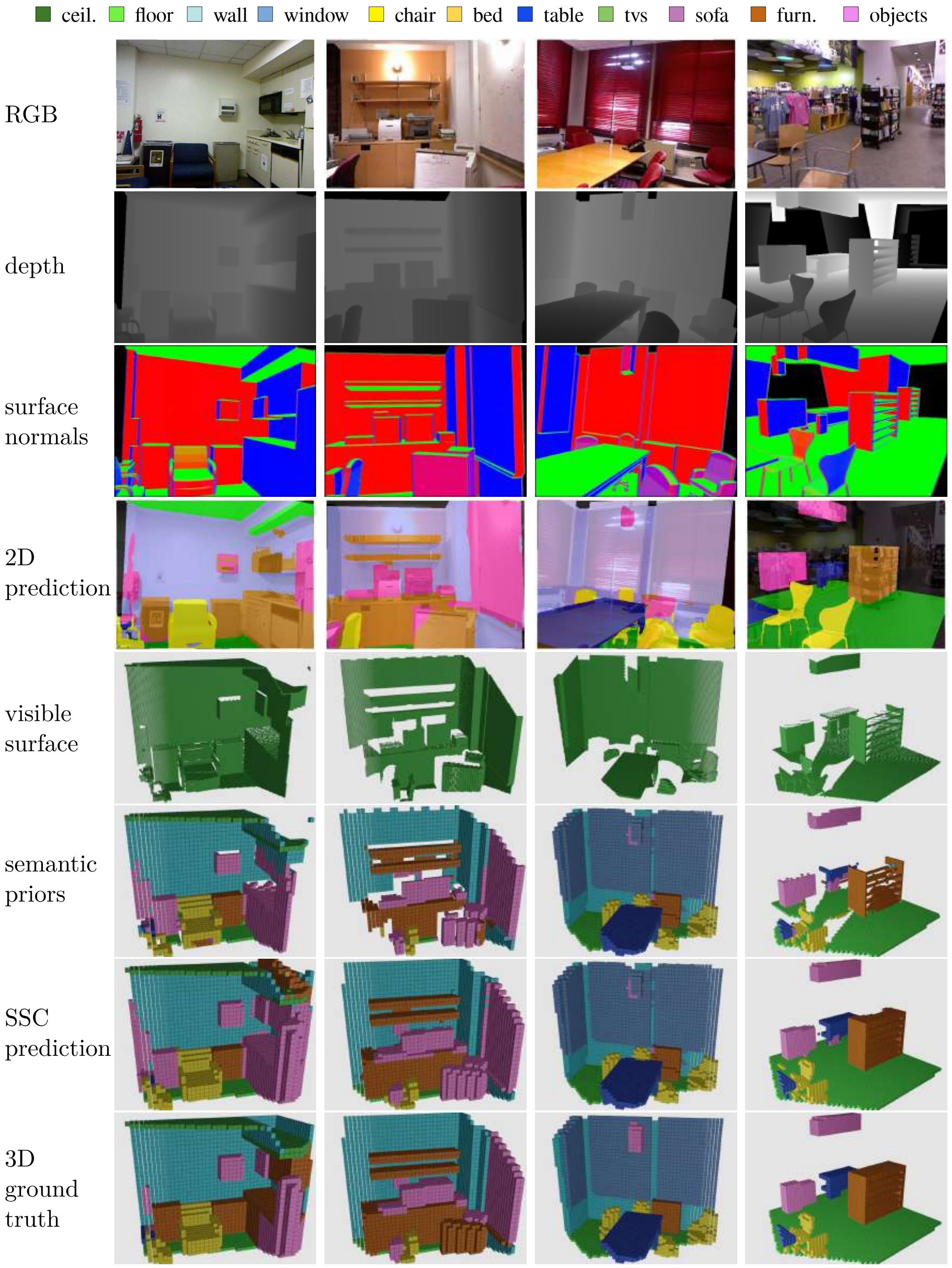


Figure A.7: **Qualitative results on NYUCAD.** From top to bottom: RGB, depth, surface normals, 2D segmentation with our bimodal CNN, visible surface, 3D priors, prediction with our SPAwN+S3P, Ground Truth. (Best viewed in color.)

Appendix B

S3P Supplementary Results

This appendix contains extra information regarding *S3P*, our semi-supervised training approach for SSC, preented on Chapter 7 as listed below.

- Complete comparison tables including all previous Semantic Scene Completion (SSC) approaches that we are aware of, by the time the experiments where made.
- Additional figures with images for a qualitative evaluation of the results on the three evaluated datasets.

B.1 Complete comparison tables

Tables B.1, B.2 and B.3 complement result tables of Chapter 7 and compare our approach to all the competing SSC solutions that we are aware of by the time of the execution of the experiments.

model	semantic scene completion (IoU, in percentages)											
	ceil.	floor	wall	win.	chair	bed	sofa	table	tvs	furn.	objs.	avg.
SSCNet[148]	96.3	84.9	56.8	28.2	21.3	56.0	52.7	33.7	10.9	44.3	25.4	46.4
TNetFuse[96]	60.6	57.3	53.2	52.7	27.4	46.8	53.3	28.6	41.1	44.1	29.0	44.9
DDRNet[88]	95.4	84.3	57.7	24.5	28.2	63.4	55.3	34.5	19.6	45.8	28.7	48.8
ForkNet[156]	95.0	85.9	73.2	54.5	46.0	81.3	74.2	42.8	31.9	63.1	49.3	63.6
VVNet[51]	98.4	87.0	61.0	54.8	49.3	83.0	**75.5**	55.1	43.5	68.8	57.7	66.7
EdgeNet[34]	97.2	95.3	78.2	57.5	51.4	80.7	74.1	54.5	52.6	70.3	60.1	70.2
ESSC[166]	96.6	83.7	74.9	59.0	55.1	83.3	78.0	61.5	47.4	73.5	62.9	70.5
CCPNet[168]	**99.2**	89.3	76.2	63.3	58.2	**86.1**	**82.6**	65.6	53.2	76.8	65.2	74.2
SPAwN (sup.)	95.8	**96.1**	**79.3**	**73.1**	**62.4**	85.6	80.5	**68.4**	**79.3**	**78.7**	**75.7**	**79.5**

Table B.1: **Results on SUNCG test set**. Our SPAwN semantic scene completion overall results with regular supervised training surpasses by far all known previous solutions on SUNCG synthetic images.

train	model	semantic scene completion (IoU, in percentages)											
		ceil.	floor	wall	win.	chair	bed	sofa	table	tvs	furn.	objs.	avg.
NYUDv2	SSCNet[148]	15.1	94.7	24.4	0.0	12.6	32.1	35.0	13.0	7.8	27.1	10.1	24.7
	ESSCNet[166]	17.5	75.4	25.8	6.7	15.3	53.8	42.4	11.2	0.0	33.4	11.8	26.7
	EdgeNet[34]	17.9	94.0	27.8	2.1	9.5	51.8	44.3	9.4	3.6	32.5	12.7	27.8
	DDRNet[88]	21.1	92.2	33.5	6.8	14.8	48.3	42.3	13.2	13.9	35.3	13.2	30.4
	DCRF[167]	18.1	92.6	27.1	10.8	18.8	54.3	47.9	17.1	15.1	34.7	13.0	31.8
	TS3D[45]	9.7	93.4	25.5	21.0	17.4	55.9	49.2	17.0	27.5	39.4	19.3	34.1
	CCPNet[168]	23.5	**96.3**	35.7	20.2	25.8	61.4	56.1	18.1	28.1	37.8	20.1	38.5
	SketchAware[23]	**43.1**	93.6	**40.5**	24.3	30.0	57.1	49.3	**29.2**	14.3	42.5	28.6	41.1
	SPAwN (sup.)	22.9	94.8	35.8	25.4	33.2	65.6	54.4	20.0	33.5	44.2	25.7	41.4
	SPAwN+S3P (s-sup.)	35.6	94.4	37.0	**30.4**	**36.8**	**68.5**	**58.9**	23.4	**32.3**	**47.9**	**30.6**	**45.1**
SUNCG + NYUDv2	SSCNet[148]	15.1	94.6	24.7	10.8	17.3	53.2	45.9	15.9	13.9	31.1	12.6	30.5
	CSSCNet[49]	-	-	-	-	-	-	-	-	-	-	-	30.5
	DCRF[167]	18.1	92.6	27.1	10.8	18.8	54.3	47.9	17.1	15.1	34.7	13.0	31.8
	VVNet[51]	19.3	94.8	28.0	12.2	19.6	57.0	50.5	17.6	11.9	35.6	15.3	32.9
	TNetFuse[96]	17.3	92.1	28.0	16.6	19.3	57.5	53.8	17.7	18.5	38.4	18.9	34.4
	ForkNet[156]	36.2	93.8	29.2	18.9	17.7	61.6	52.9	23.3	19.5	45.4	20.0	37.1
	CCPNet[168]	25.5	**98.5**	**38.8**	27.1	27.3	64.8	**58.4**	21.5	30.1	38.4	23.8	41.3
	SPAwN (sup.)	31.5	94.5	38.7	27.0	32.8	67.6	57.2	20.9	**30.7**	47.5	27.2	43.2
	SPAwN+S3P (s-sup.)	**37.5**	93.6	37.8	**35.0**	**39.4**	**71.9**	58.2	**23.4**	29.7	**50.7**	**34.2**	**46.5**

Table B.2: **Results on NYUDv2 test set**. The column "train" indicates datasets used for training the models. SUNCG + NYU means trained on SUNCG and fine-tuned on NYUDv2. Our SPAwN semi-supervised and supervised models hold the best and second-best overall semantic scene completion results for real-world images, on both training scenarios.

train	model	semantic scene completion (IoU, in percentages)											
		ceil.	floor	wall	win.	chair	bed	sofa	table	tvs	furn.	objs.	avg.
NYUCAD	DCRF[167]	35.5	92.6	52.4	10.7	40.0	60.0	62.5	34.0	9.4	49.2	26.5	43.0
	TS3D[45]	34.4	93.6	47.7	31.8	32.2	65.2	54.2	30.7	32.5	50,1	30.7	45.7
	DDRNet[88]	54,1	91.5	56.4	14.9	37.0	55.7	51.0	28.8	9.2	44.1	27.8	42.8
	CCPNet[168]	56.2	**96.6**	58.7	**35.1**	44.8	68.6	65.3	37.6	35.5	53.1	35.2	53.2
	SketchAware[23]	**59.7**	94.3	**64.3**	32.6	51.7	72.0	68.7	45.9	19.0	60.5	38.5	55.2
	SPAwN (sup.)	54.0	94.7	61.6	33.4	62.8	80.7	68.9	**47.6**	**41.4**	61.5	**42.4**	**59.0**
	SPAwN+S3P (s-sup.)	57.4	94.5	60.7	33.5	**63.6**	**81.0**	**69.0**	44.0	40.9	**61.8**	41.6	58.9
SUNCG + NYUCAD	SSCNet[148]	32.5	92.6	40.2	8.9	33.9	57.0	59.5	28.3	8.1	44.8	21.1	40.0
	CCPNet[168]	58.1	**95.1**	60.5	36.8	47.2	69.3	67.7	39.8	37.6	55.4	37.6	55.0
	SPAwN (sup.)	62.4	94.7	65.3	38.3	**70.0**	82.4	**78.2**	**50.7**	40.2	**64.9**	**42.4**	62.7
	SPAwN+S3P (s-sup.)	**66.6**	94.7	**65.9**	**39.2**	69.6	**83.3**	78.0	50.4	**41.6**	64.4	42.2	**63.3**

Table B.3: **Results on NYUDCAD**. Once again, our SPAwN semi-supervised and supervised models hold the best and second best overall semantic scene completion results on both training scenarios.

B.2 Additional Qualitative Analysis

Figures B.1, B.2 and B.3 complement Chapter 7 by presenting additional results for a qualitative analysis on SUNCG, NYUDv2 and NYUCAD, respectively. Each row of the figures corresponds to one scene. From top to bottom, we present images of RGB image, depth map, surface normals, 2D predictions (obtained with our bimodal semantic segmentation network), projected visible surface, projected semantic priors, SSC predictions (obtained by our method), and 3D ground truth.

Figures B.1 and B.3 show that our method benefits from the excellent semantic segmentation results, guiding SPAwN to generate outstanding semantic scene completion results, filling the gaps left by simply projecting semantic priors to 3D. In SUNCG, both RGB and depth maps come from 3D models of the scenes, so the 2D segmentation results approach perfection. Segmentation results are also excellent in NYUCAD, even though the RGB images come from real scenes and there is a level of mismatch between depth maps and RGB textures (see the chairs and the bookshelf on the fourth column of Figure B.3). The surface normals have certainly played an essential role in the quality of our bimodal segmentation CNN.

In Figure B.2, the first column shows that specular surfaces give poor surface normals, and saturation corrupts RGB information. These specular surfaces have perturbed the segmentation priors and generated a poor 3D prediction for the whiteboard. On the other hand, the fourth column of that figure shows that our method has correctly identified the blackboard surface as “objects”, even though that region was incorrectly labeled as “wall” in the ground truth.

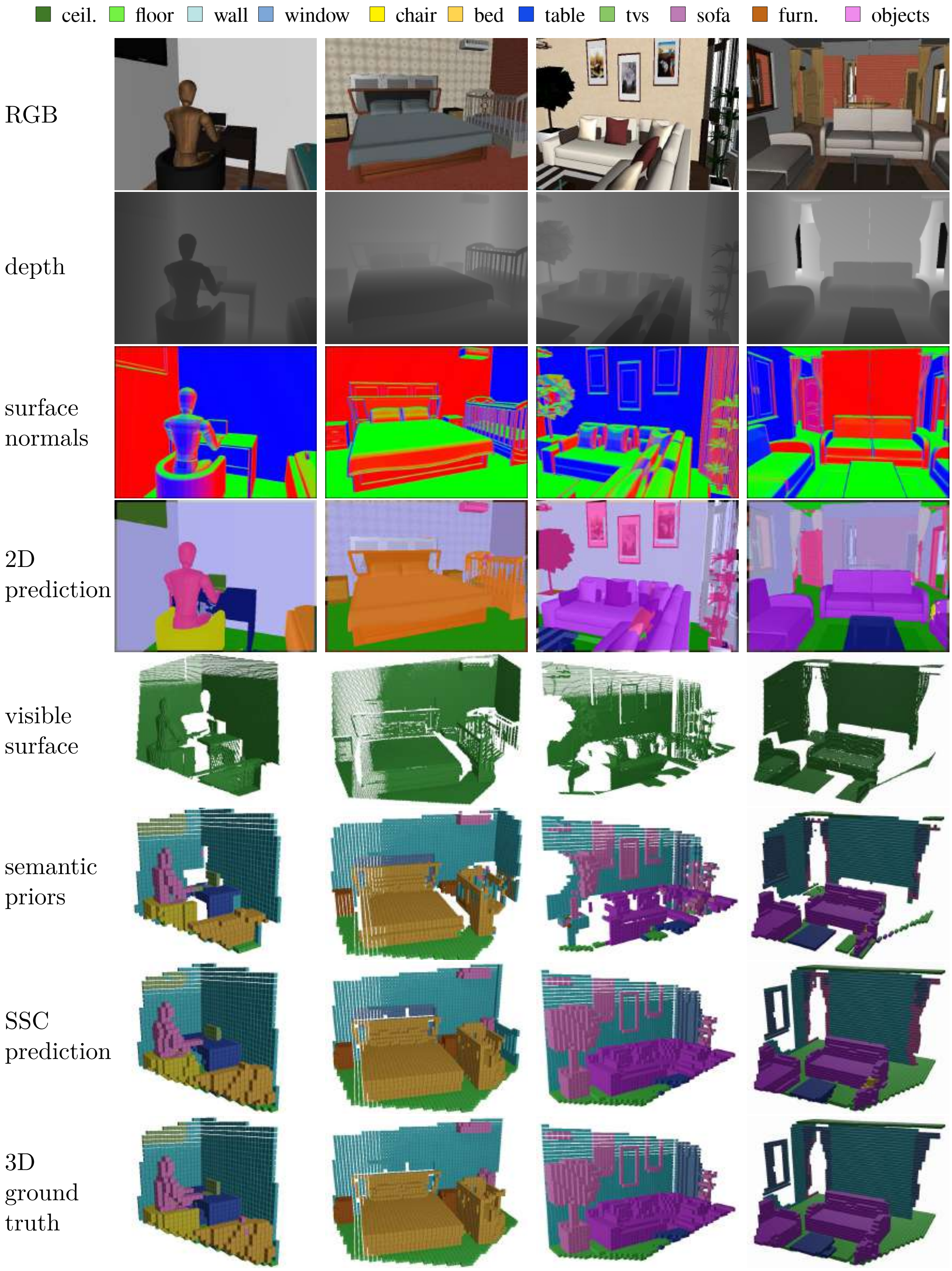


Figure B.1: **Qualitative results on SUNCG.** From top to bottom: RGB, depth, surface normals, 2D segmentation with our bimodal CNN, visible surface, 3D priors, prediction with our SPAwN architecture, ground truth. (Best viewed in color).

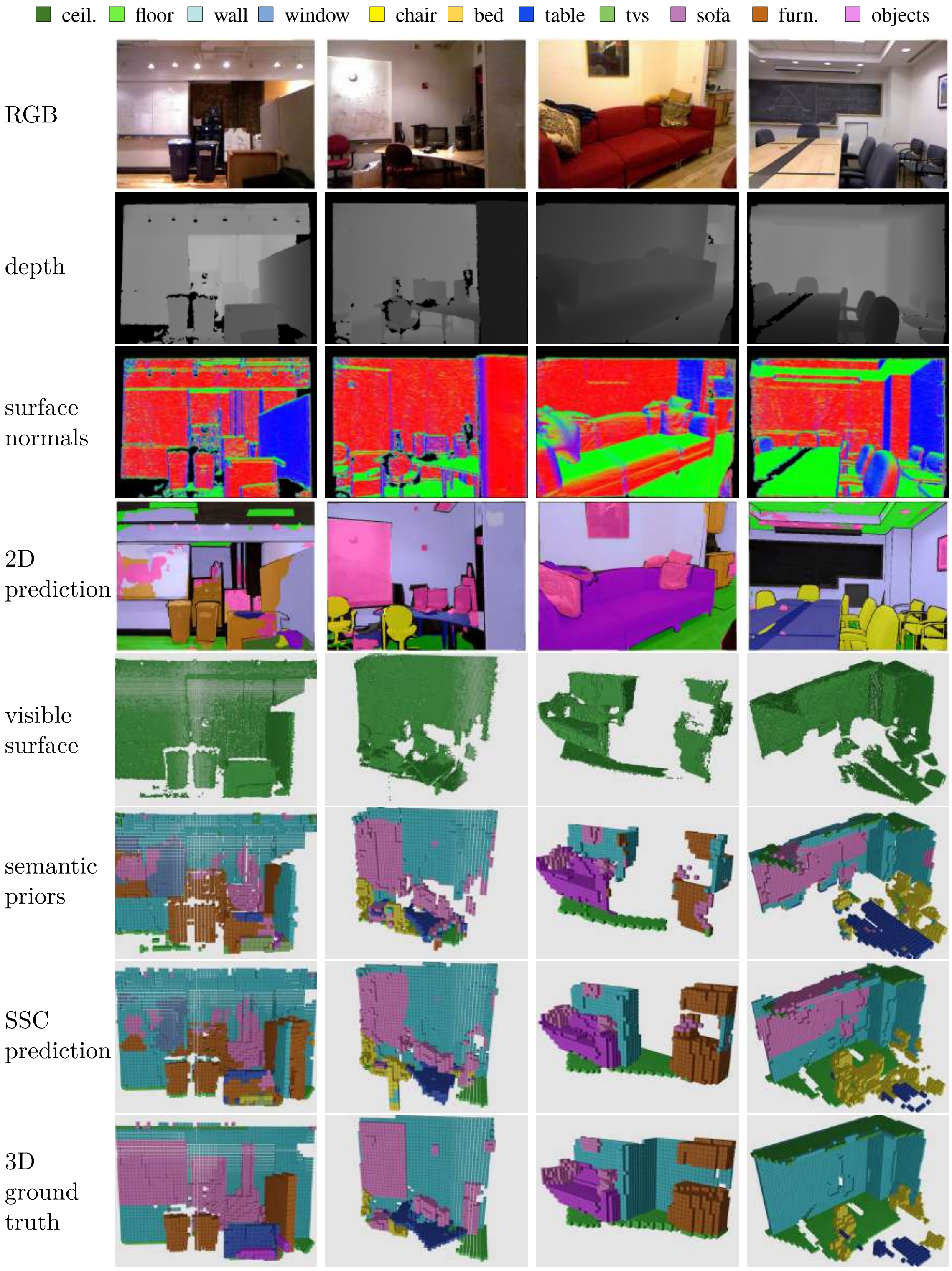


Figure B.2: **Qualitative results on NYUDv2.** From top to bottom: RGB, depth, surface normals, 2D segmentation with our bimodal CNN, visible surface, 3D priors, prediction with SPAwN+S3P, Ground Truth. (Best viewed in color).

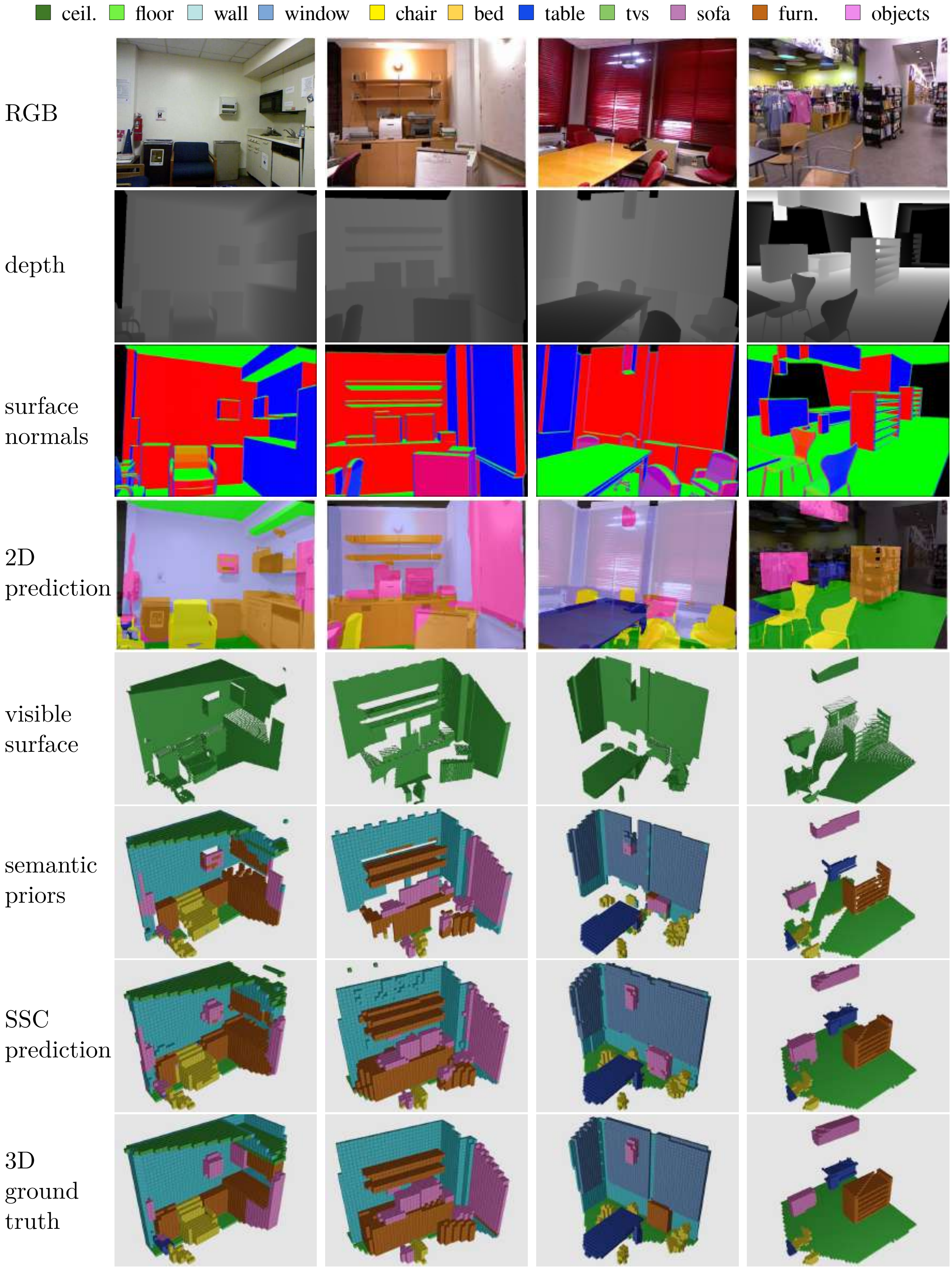


Figure B.3: **Qualitative results on NYUCAD.** From top to bottom: RGB, depth, surface normals, 2D segmentation with our bimodal CNN, visible surface, 3D priors, prediction with our SPAwN+S3P, Ground Truth. (Best viewed in color).