Semantics and Planar Geometry for Self-Supervised Road Scene Understanding

Bruce Robert Muller

PhD

University of York

Computer Science

May 2022

Abstract

In this thesis we leverage domain knowledge, specifically of road scenes, to provide a self-supervision signal, reduce the labelling requirements, improve the convergence of training and introduce interpretable parameters based on vastly simplified models. Specifically, we chose to research the value of applying domain knowledge to the popular tasks of semantic segmentation and relative pose estimation towards better understanding road scenes. In particular we leverage semantic and geometric scene understanding separately in the first two contributions and then seek to combine them in the third contribution.

Firstly, we show that hierarchical structure in class labels for training networks for tasks such as semantic segmentation can be useful for boosting performance and accelerating training. Moreover, we present a hierarchical loss implementation which differentiates between minor and serious errors, and evaluate our method on the Vistas road scene dataset.

Secondly, for the task of self-supervised monocular relative pose estimation, we propose a ground-relative formulation for network output which roots our problem in a locally planar geometry. Current self-supervised methods generally require over-parameterised training of both a pose and depth network, and our method entirely replaces the need for depth estimation, while obtaining competitive results on the KITTI visual odometry dataset, dramatically simplifying the problem.

Thirdly, we combine semantics with our geometric formulation by extracting the road plane with semantic segmentation and robustly fitting homographies to fine-scale correspondences between coarsely aligned image pairs. We show that with aid from our geometric knowledge and a known analytical method, we can decompose these homographies into camera-relative pose, providing a self-supervision signal that significantly improves our visual odometry performance at both training and test time. In particular, we form a non-differentiable module which computes real-time pseudo-labels, avoiding training complexity, and additionally allowing for test-time performance boosting, helping tackle bias present with deep learning methods.

Contents

Acknowledgments

I would like to thank my family for their unwavering support. Specifically I would like to thank Sarah Lecinski - your support and fun/crazy times is a blessing. I thank Jamie Williams for supporting me throughout most of my time in York - without your support it would have been so much harder. I would also like to thank my supervisor William Smith for his invaluable guidance and support - our discussions are really appreciated and often fun to take part in.

Declaration of Authorship

I declare that this thesis is a presentation of original work and I am the sole author. This work has not previously been presented for an award at this, or any other, University. All sources are acknowledged as References. The research in this thesis has resulted in the following papers (in collaboration with my supervisor Dr William Smith), corresponding to Chapters 3, 4 and 5 respectively:

- Muller, B. R. and Smith, W. A. P. (2020). A Hierarchical Loss for Semantic Segmentation. **In VISIGRAPP** (4: VISAPP) (pp. 260-267) [159].
- Muller, B. R. and Smith, W. A. P. (2022). Self-Supervised Ground-Relative Pose Estimation. **In ICPR** (pp. 3507-3513) [160].
- Muller, B. R. and Smith, W. A. P. (2023). Self-supervised Relative Pose with Homography Model-fitting in the Loop. **In WACV** (pp. 5705-5714) [161].

Signed, January 12, 2023 Bruce Robert Muller

1 Introduction

Road scene images are an incredibly important part of the future of visual pattern recognition with clearly compelling applications. Largely due to autonomous driving, there is now a focus on machine learning for road scenes but there are many more applications ranging from road surveying [60], mapping [169], orthomosaicing [120], property valuations [208], town and utility planning [157], to the more futuristic potentials of shared augmented reality, and traffic-routing optimisation [131, 168]. Furthermore, the road scene analysis field has grown to the extent that there are now well established conference workshops solely devoted to computer vision for road scenes [239].

Currently publications focus on analysing data for tasks such as lane and pedestrian detection [242], 3D reconstruction [36, 104, 177], semantic segmentation [19, 64, 285], visual odometry [40, 121, 231, 250, 287], depth estimation [263] or completion [127], 3D object detection [242], multiple-object tracking [241], image dehazing [2], optical or scene flow estimation [128, 248], correspondence estimation [238] and view synthesis [220]. The KITTI Vision Benchmark [66] is popular in the field and is used for road scene analysis in tasks such as these (see Fig. 1.1 for an idea of what these tasks entail). Specifically, we note tasks relating to the rich and consistent semantics present within road scenes, with many classes spanning man made and natural objects, to generally static to dynamic objects such as signs, vehicles and even birds. Furthermore, we note that imagery from vehicular scene variation is constrained approximately to planar motion.

The current literature for these techniques do not explicitly exploit the regularities and constraints of road scenes versus what we see in generic scenes [66, 169]. For example, road surfaces tend to be approximately planar and scene contents are limited to a consistent number of classes such as vehicles, pedestrians, road signs and so on [38]. Moreover, from the vehicle perspective, the road surface dominates the view and tends to be centred, with the same classes (e.g. pavements, buildings) clustering the image boundaries. We see the same classes repeatedly appearing in semantic segmentation of road scenes. These classes can be semantically related (e.g. a fence is a type of barrier which itself is a type of construction, and road-markings are also types of construction, but not a type of barrier) which could potentially be used to better train deep learning estimators. Geometric and semantic constraints like these are almost always learnt implicitly from scratch on

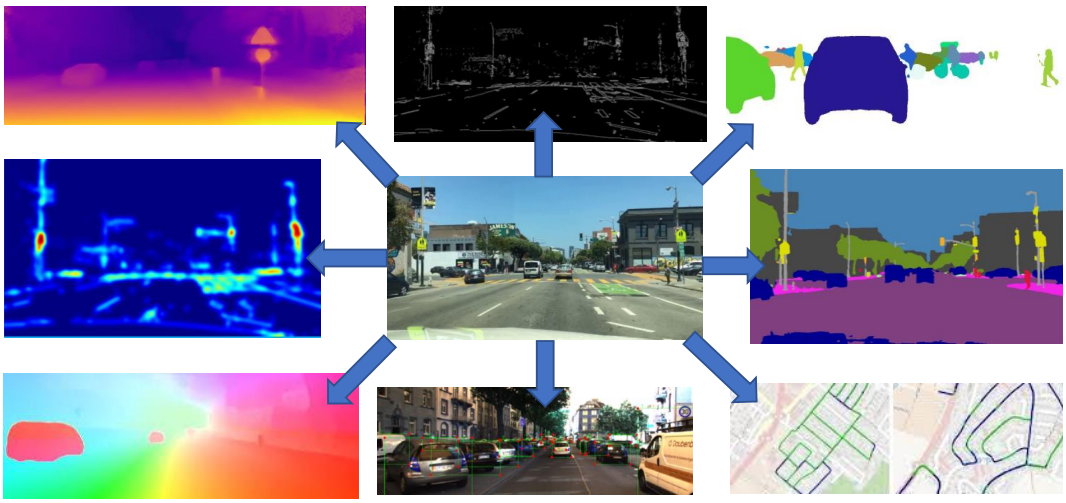

Figure 1.1: Many tasks can be performed with road-scenes such as depth estimation, various levels of semantic segmentation, visual odometry, and saliency or object detection (source: [44, 57, 66, 68, 210, 230]).

data with deep learning systems [13, 68, 250, 287]. For example, the top-left image of Fig. 1.1 shows a result from Monodepth2 [68] depth estimation on road scenes and the right-side shows semantic segmentation of scenes. Here we can see that the planarity of the road creates a regular constraint and yet this is being learnt implicitly, making the task significantly harder to solve.

This thesis asks the question of whether we can explicitly integrate domain knowledge, what we already know about the task being solved, into road scene analysis as applied with deep learning. Can we improve deep learning methods on road scene understanding by integrating prior knowledge into the machine learning process? This improvement could be in terms of performance, speed of training, amount of data required, complexity of network architecture, memory requirements, robustness of network outputs and so on.

In order to investigate this question we focused on two specific sub-tasks: road-scene semantic segmentation and relative pose estimation. We use the definition of semantic segmentation as the classification of each individual pixel in an image into a discrete number of classes. For relative pose estimation, we are predicting the relative translation and rotation between two cameras associated with their given images. We mentioned previously that domain knowledge for road scenes could be geometric constraints or it could be semantic in nature as relating to the classes commonly found in road scenes. For this thesis, we have focused on

1. The hierarchical structure present in road scene classes for semantic segmentation.
2. The planar geometry of the road present in all road scenes for relative pose estimation.

1.1 Contributions

In this thesis, we make the following contributions:

- **Chapter 3**:
  - Show that we can differentiate between serious and minor errors by exploiting semantic knowledge of class hierarchies.
  - Provide an implementation for a novel hierarchical loss as applied to semantic segmentation (see Section 2.1.4 for related literature).
  - Demonstrate that this hierarchical method can provide training benefits over non-hierarchical supervision (see Section 2.1.5 for related literature).
- **Chapter 4**:
  - Put forward a novel ground-relative parameterisation for the pose of two cameras in a locally planar geometry.
  - Show how this parameterisation allows for a homography to cross-project the road scene contents between image pairs for the purposes of an appearance loss, replacing the need for dense depth estimation.
  - Illustrate the effectiveness for motion estimation of a perceptual appearance loss using a pre-trained VGG network [202] for a wide-basin of convergence (see Section 2.2.6 for related literature).
  - Show that a geometric matching network [190] can be used for the task of regressing relative pose effectively (see Section 2.2.5 for related literature).
  - Demonstrate that our local ground-relative pose formulation in combination with the geometric matching network allows for a degree of flexibility towards arbitrary pose estimations, not present in other self-supervised methods (see Section 2.2.2 for related literature).

- **Chapter 5**:
  - Illustrate a method to refine our relative pose estimates (see Section 2.2.2 for related literature) where a pre-trained optical flow estimator can be used in conjunction with RANSAC to estimate homographies for the planar regions extracted by a semantic segmentation network.
  - Show that we can use a known analytical method for decomposing homographies with our knowledge of the geometry for our motion to provide camera-relative pseudo-labels for training our network (see Section 2.2.4 for related literature).
  - Explain how this non-differentiable modeling-fitting based method can be further utilised at inference time to refine relative pose estimation for tasks such as visual odometry (see Section 2.2.7 for related literature).

1.2 Publications

The research in this thesis has resulted in the following papers, corresponding to Chapters 3, 4 and 5 respectively:

- Muller, B. R. and Smith, W. A. P. (2020). A Hierarchical Loss for Semantic Segmentation. **In VISIGRAPP** (4: VISAPP) (pp. 260-267) [159].
- Muller, B. R. and Smith, W. A. P. (2022). Self-Supervised Ground-Relative Pose Estimation. **In ICPR** (pp. 3507-3513) [160].
- Muller, B. R. and Smith, W. A. P. (2023). Self-supervised Relative Pose with Homography Model-fitting in the Loop. **In WACV** (pp. 5705-5714) [161].

2 Related Work

A decade ago, the famous AlexNet [108] neural network implementation unexpectedly won the image classification competition (ImageNet) with significant performance gain compared to the second best method [195]. Since then, there have been many variations on neural architectures which have brought ImageNet accuracy to within human level performance [78, 202, 213, 268]. In the following years the annual competition effectively became a proving ground for neural network implementations, with researchers globally pursuing the next big deep learning techniques. Since then deep learning has since proven itself in numerous fields from image recognition [53, 176, 188, 259] to natural language processing [49, 185, 201, 223] in providing state-of-the-art performance. Whilst the recent revolution in deep learning has seen real performance gains, it has developed a reputation as a one tool fits all technique, often ignoring the principled research and models which many researchers have worked hard to develop [87, 174, 207].

The initial wave of very successful deep learning techniques were essentially black-box methodologies where data is fed into a network which outputs a relevant result for the application at hand [78, 108, 202]. For example, images were input into the neural network which outputs a general classification or pixel-wise segmentation [7, 108]. Many methods were purely black-box and entirely supervised with little real physically rooted modelling under the hood [179].

More recently we have seen a second wave of more principled approaches where better understood models have been re-introduced into learning pipelines. Some of these methods do this by engineering accepted principles directly into the neural architectures. For example, Rocco et al. [190] take the well known classical pipeline stages for feature matching represented as sequential layers in the network pipeline for geometric matching. Other works such as NeRF [152] have introduced understanding of geometry and physics of light into the deep learning pipeline through exploiting image formation knowledge as 3D volumes and using view dependence radiance. Further, inverse rendering methods have integrated knowledge about reflectance and lighting into the deep learning process [198, 261].

In this thesis we research some ways of combining what we already know into the learning process for neural networks as applied to road scenes. Specifically we break it down into two overarching themes: scene understanding and 3D geometry.

In Section 2.1 we overview methods which attempt to integrate knowledge

of scene elements. In particular, we investigate the angle of using hierarchical knowledge present in how we perceive classes present in imagery. Secondly we overview related work for the task of semantic segmentation - setting the scene for our hierarchical loss for semantic segmentation.

In Section 2.2 we review literature which uses geometric understanding to enrich the learning process. We start by looking at methods in generic geometric terms and then hone in on literature which focuses on geometry for synthesising views for supervision signals, and methods which use geometry to aid the pose estimation pipelines, specifically in relation to visual odometry, and briefly overview sources relevant for our optical flow with RANSAC approach.

2.1 Scene Understanding Meets Deep Learning

There are many ways of interpreting what we mean by scene understanding. In this section we isolate and briefly overview some more general examples of what we could mean, before discussing the hierarchical and semantic scene understanding relevant for our work in Chapter 3.

2.1.1 Scene Understanding with Physics

Zheng et al. [278] illustrate an ambitious attempt to understand 3D scenes. From a point cloud they generate cubic models of a scene, group together unstable parts, and attempt to reason about stability using physical mechanics. The novelty in their approach comes from use of disturbance fields and disconnectivity graphs (a way of modelling the scene energy landscape) to reason about the stability of objects. The strengths of this kind of scene understanding are clear in that it provides a way of assessing scenes for potential hazards, but the complexity of their method leaves room for error in cluttered or dynamic scenes where their assumption of static and gravitationally stable objects may break down.

More recently, Mezghanni et al. [150] propose a physical simulation layer for 3D trajectories of objects, taking gravity into account. Hong et al. [83] propose a deep pipeline integrating physical dynamics and simulation for state-of-the-art performance for fixing malformed 3D objects. Christen et al. [37] use reinforcement learning with physical simulation towards physically realistic hand-object interaction. Mezghanni et al. [151] propose an approach of training generative networks such that information about the physical practicality (e.g. connectivity of chair joints) of 3D models are improved and achieve competitive results for shape modelling on the PartNet 3D object understanding dataset [155]. We note that

work remains to model material properties of objects, for example, glass objects compared to wooden ones, within a deep learning setting, particularly with respect to road scenes [54, 91].

2.1.2 Scene Understanding with Aerial Imagery

It is worth noting that scene understanding could also be perceived from an aerial perspective where the coverage is much greater and can differ significantly in nature versus a ground perspective [11]. State-of-the-art methods vary from exploring vision transformers for object detection or semantic segmentation [229], a Convolutional Neural Network (CNN) feature fusion method for scene classification [5] and a path aggregation CNN method for instance segmentation [23, 129, 234].

In terms of combining domain knowledge with road segmentation, Mattyus et al. [145] use a CNN to initially segment roads from aerial images, and then graph theoretic shortest path reasoning to help fill in the gaps. Their approach achieves highly accurate results on complicated road networks. One downside of their work is that it can fail to fill in road paths correctly from the gaps left by the deep learning segmentation, for example suggesting a route which passes over a building. A residual U-Net [191] (see Section 2.1.4) architecture is utilized by Zhang et al. [275] to extract roads from aerial images. In contrast with Mattyus et al. [145], their work uses only three convolutional residual blocks, compared to fifty-five. They also use a Mean Square Error (MSE) loss instead of an Intersection over Union (IoU). Their results show promise in comparison to other attempts but they do not compare to Mattyus et al. [145], and, in contrast, make no attempt to solve for discontinuities in the road segmentation.

Incorporating information of building location could help improve accuracy in cases such as these, perhaps using details of buildings, shadows and other content as priors to the deep learning segmentation pipeline could help improve the initial road segmentation, avoiding the use of error prone shortest path estimates [11, 97, 145]. It is this sort of knowledge incorporation into the deep learning process which is of significant interest to our research.

2.1.3 Scene Understanding with Objects and Events

We could interpret scene understanding in the context of recognizing objects or even events within a scene. For example, Wei et al. [236] construct an algorithm which takes depth video and segments it in a way which simultaneously extract events and objects. Similar to previous approaches, they use a hierarchical tree structure to represent relationships between events and objects. This form of representation

has the strength of being well understood and algorithmically implemented, but could show weakness in the ambiguity between associating an event with the correct object(s). Their approach has clear strength in overcoming problems such as localizing objects under occlusion, by utilizing contextual information of human interaction, but it seems unclear of whether a graph hierarchical representation is sufficient to capture the complexities of temporal variations between scenes. Recently, Singh et al. [203] provide a dataset for recognising events for road scenes towards autonomous driving, extending the Oxford RobotCar Dataset [140]. We observe that state-of-the-art methods [63, 260] utilising deep networks for road scene event recognition tasks tend to lack a training regime which explicitly leverage inter-event class relationships [246].

In terms of integrating deep learning in the object detection field, work such as that by Li et al. [118] use a CNN for salient object detection. With the aim of producing a saliency map of a scene, they highlight that most deep learning methods use a CNN which has too coarse a level of extraction, resulting in blurred saliency maps and boundaries. They aim to remedy this by using two parallel networks to capture coarse and fine details, with sharing between them. Using the convolutional part of a VGG [202] architecture they also attempt to capture multiple scales of the input to produce the desired saliency map. Object detection is a richly populated field spanning many domains [125, 176]. For road scenes the KITTI dataset [66] is the most popular benchmark with state-of-the-art methods [73] including a graph CNN [249], depth completion [242], patch refinement [115], feature-voting convolutional layers [55] and a cascade attention network [240]. We note there there may be a gap in the literature around leveraging training losses which integrate knowledge for pre-existing relationships between object classes [73, 91, 176].

2.1.4 Scene Understanding with Semantic Segmentation

The state-of-the-art in semantic segmentation has advanced rapidly thanks to end-to-end learning with fully convolutional networks [30, 71, 132, 172, 191]. The field is large, so we provide only a brief summary here. The well known U-Net architecture from Ronneberger et al. [191], originally devised for the application of biological cell segmentation, is commonly utilised for many general segmentation domains [8, 122]. The U-Net [191] architecture is characterised by a hourglass shape where we have convolutional layers followed by deconvolutional layers with additional direct data links between symmetric layers in the convolution-deconvolution network [191]. Deconvolutional layers generally aim to upsample the final feature map

from the convolutional layers such that we reform the original resolution of input imagery. In Chapter 3 we utilise a basic U-Net [191] for its simplicity and popularity.

In Chapter 3 we apply semantic segmentation to the Helen facial dataset [112] as a toy example before using it for road scene segmentation. In addition, we argue that segmenting faces could be a not insignificant part of road scene understanding with pedestrians being a common and important class [153]. The state-of-the-art on the Helen dataset [112] is currently achieved by Zheng et al. [279] with their method of concurrently learning edge detection tasks with a graph CNN. A significant facial segmentation method by Lin et al. [124] use a spatial focusing transform and a Mask R-CNN/Resnet-18-FPN [77, 125] region-of-interest network for segmenting facial sub-components into a whole. Güçlü et al. [71] perform facial semantic segmentation by augmenting a CNN with Conditional Random Fields (CRF) and an adversarial loss, while Ning et al. [172] achieve very fast performance using hierarchical dilation units and feature refinement.

The encoder-decoder style of U-Net [191] architectures have been applied extensively to the segmentation of road scenes [74, 91, 179, 275]. Kendall et al. [7] present SegNet, a fully convolutional network in semantic segmentation, for its accurate, fast and practical engineering to semantically label every pixel in an image. They use an enconder-decoder architecture (similar to U-Net [191] but without skip connections transferring entire feature maps). The main novelty with SegNet [7] is using the pooling indices from the enconder pipeline in the corresponding upsample phase in the successive decoder pipeline, which reduces the requirement to learn upsampling. Wu et al. [243] more recently explore extensions of the SegNet [7] architecture.

Most recently, transformer based approaches have become prominent and achieve state-of-the-art performance in semantic segmentation on datasets such as ADE20K [281], but these approaches are generally computationally expensive [9, 33, 35, 117, 237]. Fully convolutional approaches such as Rota Bulò et al. [192] and Verelyst et al. [224] achieve competitive road scene semantic segmentation performance whilst saving on compute resources. Rota Bulò et al. [192] combine a DeepLabv3 [28] head with a wideResNExt [265] body and propose a special form of activated batch normalisation which saves memory and allowing for a larger network throughput. Verelyst et al. [224] uses reinforcement learning to decide on the complexity of image regions in order to process them at higher or lower resolution to save on computation. Chen et al. [29] achieve state-of-the-art for panoptic segmentation (combining semantic and instance segmentation) on the Mapillary Vistas dataset [169] by investigating variants of the Wide-ResNet architecture. Furthermore, Nag et al. [167] achieve impressive night-time segmentation results on the Mapillary Vistas dataset [169]. Ganeshan et al. [64] and Borse et al. [16] achieve state-of-

the-art on the Kitti [66] and Cityscapes [38] datasets respectively with methods utilising convolutional networks, but we note again that there exist transformer methods which are close contenders [33, 264].

Tao et al. [215] achieve highly competitive segmentation performance on the Mapillary Vistas [169] and Cityscapes [38] road scene datasets. They use a hierarchical attention approach towards leveraging multiple scales of images input into deep pipelines to show that some scales are favourable for tackling fail cases [215]. A hierarchical approach for motion segmentation by Bideau et al. [14] combines ridged motion constraints with optical flow and object proposals to group related objects, but do not explicitly utilise the inherent structure in class labels themselves. As we noted in Section 2.1.3, we similarly observe a lack of literature in semantic segmentation for road scenes which leverage known relationships between classes [91, 98, 110].

2.1.5 Scene Understanding with Hierarchical Knowledge

Another rich source of domain knowledge comes in the form of training labels themselves, often encapsulated within the labelling structure [218]. Two obvious ways of utilising hierarchy is by hierarchical structuring of network architecture and leveraging the hierarchical relationship between training labels.

Hierarchical Architectures

Many existing hierarchy-based methods have focused on *hierarchical architectures*, i.e. methods that specifically adapt the architecture of the network to the specific hierarchy for a particular task. This is typified by Branch-CNN [284] and Hierarchical Deep CNN [247] in which a network architecture is constructed to reflect the classification hierarchy. Deng et al. [45] encode class relationships in a Hierarchy and Exclusion (HEX) graph, which enables them to reason probabilistically about label relations using a CRF. While very powerful, this also makes inference on their model more expensive and defining a HEX graph requires rich information.

Yan et al. [247] recognize that little prior work explored utilising the hierarchical structure of categories within a CNN. They take a two-level hierarchy and use coarse predictions (learned from initial layers) to learn fine predictions with the later layers, on a per category basis, which are then combined probabilistically to produce a final prediction. Their work shows encouraging performance on the CIFAR100 [107] and ImageNet [195] datasets, but lacks more theoretical underpinning, and only operates on a two-level categorical hierarchy.

Fan et al. [58] take this idea further by extracting feature maps from various network depths and constructing the hierarchy itself into groups based on visual similarity. This appears to extract a decision tree based classifier from the convolutional layers, which is used to replace the final softmax-layer. Murdock et al. [164] choose to simultaneously learn the model architecture depending on the hierarchy of object categories present within the data. Hu et al. [85] use networks containing multiple levels of activation functions towards brain segmentation.

Popular research by Yang et al. [253] leverages the hierarchical structuring of text documents for document classification. They illustrate that mirroring the hierarchical data structure by the architectural network structure in combination with attention layers helps to learn more informative features for the task. Their insight is that the structure of the document data (e.g. the interaction of different words) is important towards the text meaning, and also to focus on more meaningful parts of the text. The dual idea of attention and data structure seems important when we talk about context: meaning of a string of words could change when implanted into another element of text structure. Perhaps a similar idea could apply to computer vision towards identifying contextual information which helps to split classes of similar appearance. A leading hierarchical classification approach by Wehrmann et al. [235] outputs multiple classifications at varying levels of neural architecture. These methods while quite powerful and potentially useful for imbalanced datasets [95], are somewhat complex and difficult to implement widely for differing tasks [218].

Hierarchical Priors

We briefly note a few works which have attempted to capture relative ordering within machine learning. Examples include ordering regional image brightness by Chai et al [25], metrics within pairs of points in an image by Zoran et al. [286], and age estimation by Niu et al. [173]. The advantages of using losses which capture such ordinality seems to be robustness to noise. Conversely, Chen et al. [31] suggest a better way of capturing ordinal relations using a "ranking CNN".

Various works leverage hierarchy of classes within images. Luo et al. [136] use hierarchical structuring of faces to constrain face segmentation. Srivastava and Salakhutdinov [206] take a probabilistic approach and attempt to learn a hierarchy as part of the training process. Meletis et al. [149] use a restricted set of classifiers in a hierarchical fashion on the output of a standard deep learning architecture to harness differing levels of semantic description. Roy et al. [193] propose a way of hierarchical learning, and their novelty lies in retaining previously learned classes when integrating new classes (termed "lifelong learning"). Methods of learning

hierarchy or growing the neural architecture with prior information is promising but seems potentially unrealistic.

Motivated by the observation that traditional CNN image classifiers treat all classes on an equal footing, Zhu et al. [284] learn to predict multiple levels of the hierarchical categorization (given as prior knowledge) through sections of the standard CNN pipeline. They combine coarse to fine predictions to form a final classification. The strength of their method is evident in using a novel training scheme where the prior structure is integrated and where a separate loss is aligned with the different levels of the hierarchy. This form of adjusting your losses to be in line with the knowledge you want to capture (in this case hierarchical structuring of the data, given as a prior) seems promising to building effective systems which better mine expert domain knowledge.

Furthermore, Redmon et al. [189] use the YOLO model [188] for classifying objects hierarchically. Graham et al. [70] use uncertainty maps at differing hierarchical levels of brain regions towards brain segmentation. Mehdipour et al. [146] use hierarchical soft-max and fusion for 3D segmentation of the human brain.

Much of the literature in deep classification approach training a network with only a flat hierarchy [70, 146, 166] where all classification errors are treated equally. As pointed out by Graham et al. [70], it is important to penalise differently between classifying different brain regions as this could have diagnostic or medical consequences. Similarly for the road scene domain, we want to differentiate training signals between major errors (e.g. mistaking a person for a truck) and minor errors (e.g. mistaking a bus for a truck). In particular we observe that hierarchical information between classes is under-utilised for semantic segmentation for road scenes [1, 3, 91, 98, 156] and in Chapter 3 we propose a method for training with a hierarchical loss which penalises more severe training errors according to a pre-defined hierarchy on the Mapillary Vistas road scene dataset [159, 169].

2.2 Geometry Meets Deep Learning

While semantic understanding of classes is conceptually simpler, geometric understanding can be a more complex but obvious knowledge form to integrate within deep learning methodologies. Tasks such as semantic segmentation are generally solved within the two dimensional domain of images, whereas geometric tasks are often framed in the context of three dimensional space, increasing the variability of the estimation parameters and the amount of data required to train models [106]. Moreover, whereas in semantics we are often classifying for a discrete number of classes, in geometric understanding we usually are concerned with estimating

continuous parameters like pose, which significantly increases the complexity of the task as generally we need to learn more complex features in order to ascertain from data an entire spectrum of solution parameters [196]. Further, due to the increase in solution space complexity, constraints are generally required to achieve a reasonable solution, further challenging geometric learning [13, 67, 282].

It is interesting to note the work being done by Hauser et al. [76] in terms of understanding deep learning with finite differences and dynamical systems. This kind of research focuses on the internal geometric transformation of the underlying data manifold in a theoretical sense, which is present within all neural networks. An advantage of this approach is that we may represent each layer in terms of a high-level class of differentiable finite difference relations, which can then be used to help form architectures manually. Furthermore, they reveal residual type networks in particular to be finite difference versions of Ordinary Differential Equations (ODE) representing dynamical systems.

In this context, the term geometry can be used to mean different concepts: the shape or dimensionality of the underlying data (e.g. graphs and geodesics [17]), spatial volumes associated with parts of the neural architecture, or the physical geometry of objects and surfaces occupying the space captured by images. In this thesis we are concerned about the latter and we want to explore how knowledge of the environmental geometry can be used to help inform the training of deep models.

Firstly, we investigate works around relative pose estimation for its relevance to road scenes and geometry. Secondly, we explore self-supervised relative pose estimation. Thirdly, we explore works around the topic of view-synthesis which can be useful for the purpose of forming a training signal. Fourthly, we focus on literature for deep learning with homographies as a homography is key for plane-to-plane transformations for our planar road geometry. Fifthly, we briefly overview some work relevant to choice of neural architecture and network input. Sixthly, we review works around perceptual loss for training these architectures. Lastly, we discuss the idea of leveraging classical methods within the deep learning pipeline.

2.2.1 Geometry as Relative Pose Estimation

The primary form of geometric deep learning which we explore in this thesis comes in the way of estimating camera poses. In this section we review some early more general approaches, and then in the next section overview more recent literature towards self-supervised relative pose estimation.

A major step forwards towards incorporating geometric understanding into the

deep learning framework came with PoseNet by Kendall et al. [99]. PoseNet directly regresses pose from an image and is trained in an end-to-end manner. Strengths of PoseNet included illustrating viability of simple pose regression from scene imagery, and the ability to generalize to unseen scenes with a little extra training. It is interesting to note that PoseNet seems to learn pose relevant information from an initial phase of pre-training on the pose-invariant task of ImageNet classification. Although their system performs remarkably well to various occlusions and weather, it exhibits significant weakness with motion blurred input.

Vijayanarasimhan et al. [225] propose SfM-Net where they use video to learn Structure from Motion (SfM). Their system is said to be "geometry-aware" in that they take a consecutive pair of frames into a CNN and output depth, motion segmentation and camera/object motions. Subsequently these are combined to generate an optical flow field and a form of view synthesis is used between frames to allow back-propagation. Their network architecture uses a U-Net [191] format and some fully-connected layers are used to extract camera and object motions. This architecture seems somewhat hand-made but results show value in segmenting moving objects. Elements of geometry in this method seems to be output of pose-change, transformation of point clouds and the options of supervising by either depth or camera motion.

Taking inspiration from multi-view geometry to inform the architectural design, Kendall et al. [100] take key steps in classical stereo regression and replaces each step with a distinct differentiable layer in the network. This form of integration of traditional geometric modelling, while somewhat artificial, allows for accurate and fast results on the Scene Flow and KITTI datasets without the need for additional post-processing. The paper claims to leverage the task geometry and context but potential weakness exists in terms of a lack of explicitly spatial geometric ideas.

Classical methods for estimating camera-relative pose often involves finding correspondences between two images, and often this fails where, for example, there are large viewpoint changes between the two images, or textureless regions. Melekhov et al. [148] use a Siamese CNN to regress relative pose from a stereo pair. Their method has the advantage over traditional methods of not requiring camera intrinsics but it does not always improve accuracy.

In data where there is a strong sequential structure (e.g. in natural language processing), it is common for recurrent networks to be employed in the learning process. With this in mind, Wang et al. [231] have used a recurrent CNN to learn pose end-to-end from video. They highlight that traditional visual odometry pipelines need to be manually tuned to work accurately in different environments. The value in their method is bringing recurrent networks, which model sequential dynamics, into the CNN end-to-end pipeline without the need to manually tune to

differing environments. It seems that in some ways it bears trademarks with Kendall et al. [100] where distinct conventional modules are replaced by a differentiable layer in the network. Potential weaknesses of the system is the absence of explicit attention mechanisms within the network. Additionally, they lose camera intrinsics information, which is present within the conventional pipeline. The authors express its usefulness as complementing the traditional pipeline, which better captures geometry; this softly suggests the need for a fuller integration between traditional geometric and deep learning paradigms.

We observe a lack of work around explicitly integrating the geometry of road scenes with the relative pose parameterisation output from deep pipelines [1, 91]. Deep networks generally estimate a 6 DoF (Degrees of Freedom) camera-to-camera pose [68, 287] and do not attempt to model pose relative to surrounding geometry within this parameterisation [3, 73, 98, 179]. In particular, we suggest this may be of specific interest to the self-supervised relative pose community for the purpose of forming a training signal.

2.2.2 Self-supervised Relative Pose Estimation

Many early deep learning pose estimation works focus on directly supervised methods. We now overview related work within the self-supervised arena, with a focus on the evaluation domain of visual odometry. State-of-the-art visual odometry solutions tend to rely on many cameras or expensive sensors such as laser scanners [36, 40, 104]. Others rely heavily on training using manually collected ground truth labels with GPS and inertial sensors [91]. For example, a real-time SLAM system by Min et al. [154] use optical flow residuals with a multi-task probabilistic model but rely on ground truth labels to solve for scale ambiguity.

Methods which combine neural networks with classical pipelines [133, 219, 250, 251, 256, 271], leverage techniques such as photometric bundle adjustment or loop closure for optimisation and various SLAM pipelines. For example, D3VO [250] is the most competitive purely monocular method on the KITTI odometry benchmark [66] which utilises pose-depth networks with illumination transformation with uncertainty maps. Furthermore, they rely on a combination of front-end tracking and back-end non-linear optimisation. The front-end tracking takes estimates from the depth and pose networks towards a constant motion model. The back-end non-linear optimisation takes estimates of depth, pose and uncertainty for use in a photometric bundle adjdustment. Additionally, while D3VO [250] uses an improved self-supervised training loss similar to Monodepth2 [68], they utilise full dense depth network training along side a pose network. In our work we choose to focus

on end-to-end learning for ease of use and compare our results to such methods in Chapters 4 and 5.

Although training labels are valuable it is worth highlighting that much work in the past few years has focused on learning visual representations without the need for millions of labelled images for training [73, 179]. For example, Wang et al. [232] show that we can obtain comparable performance using unsupervised learning of video sequences to supervised methods. Their key contribution is showing we can use visual tracking of objects within a video as the basis for training.

For the purposes of forming an appearance loss, most self-supervised visual odometry methods will parameterise two network outputs as dense depth and 6 DoF camera-relative pose respectively [13, 24, 59, 68, 69, 121, 270, 282], (some of which rely on stereo training [59, 121]). In particular we note that the leading end-to-end self-supervised relative pose estimation methods on the KITTI benchmark [66] are all reliant on training a pose network jointly with dense depth [13, 68, 282, 287] or additionally optical flow [187, 257, 276] networks.

LTMVO [287] achieves one of the best visual odometry results for self-supervised methods by using a recurrent CNN to temporally constrain the trajectory but also rely on a pose-depth network with a 6 DoF camera relative pose. Further, LTMVO's LSTM modules are easy to overfit, sensitive to weight initialisation, memory intensive, and take longer to train [262].

Various works are also highly competitive self-supervised approaches but again are reliant on dense depth estimation [59, 69, 219, 228]. Another competitive method is Towards Better Generalisation (TBG) by Zhao et al. [276] which shows that many methods lack performance when frame separation is too high for relative poses. They replace the pose network, in standard dual pose-depth network approaches, with dense optical flow and then recover relative pose with projection constraints. While this combination of geometric constraining with deep learning is in line with our thesis of modeling what we already know, they still require to train for thousands of depth and optical flow parameters, and in our work we instead propose to replace multiple dense map networks with a ground-relative parameterisation for a single pose network.

Recently methods use optical flow networks with camera-relative pose estimation to form a self-supervised signal [135, 187, 257, 276, 288]. In particular, Ranjan et al. [187] perform competitively as a self-supervised method on the KITTI visual odometry dataset [66] but rely heavily on training jointly various dense estimation networks. Yin et al. [257] is also a well known approach combining optical flow with pose-depth networks but performs poorly [287] on the KITTI visual odometry dataset [66] relative to other self-supervised methods such as LTMVO [287] and TBG [276]. Notably Bian et al. [13] try using a depth warping loss and masking

of dynamic objects but also performs less favourably than other self-supervised methods. Most of these approaches rely heavily on frame-to-frame predictions, and many set pose estimation as a secondary objective [68, 69].

Parameterising as a camera relative pose and dense depth or optical flow estimation task tends to limit estimation to adjacent or temporally close video frames [276]. Further, estimating many thousands of parameters for depth or flow is a demanding and ill-posed task which is tricky to train [68, 187]. Moreover, we see from the Monodepth2 [68] experimental results that it performs very well with depth estimation, but significantly less so with pose estimation. This implies that methods using both a pose and depth network are prone to the issue of one network influencing the accuracy of the other. Work by Tiwari et al. [219] attempt remedying this issue by arguing their complementary nature but rely on classical SLAM and potentially expensive optimisation routines such as bundle adjustment and loop closure.

We note there is a lack of work in self-supervised relative pose approaches where the geometry itself could be leveraged towards a self-supervised training signal [1, 91], rather than relying on over-parameterising training with expensive dense depth and optical flow networks. Instead, in Chapter 4 we propose a deep pipeline which estimates camera pose relative to the local planarity of the road surface and form a self-supervised training signal with view synthesis via this geometry.

2.2.3 View Synthesis

One technique which understanding the geometry of a scene affords us is that of view synthesis. In the context of the work in this thesis view synthesis is where we may transform an image captured from one camera pose into the perspective of another pose by leveraging some geometric knowledge of the scene with the relative pose of the cameras. For example, Zhou et al. [68] and Godard et al. [282] use dense depth estimation in conjunction with relative camera pose to warp imagery into the perspective of target images in order to form a self-supervision signal. In this thesis we will utilise a similar method but instead of attempting to estimate depth we assume a planar geometry for perspective warping ground-plane pixels (see Chapter 4).

A key development related to view synthesis is that of Spatial Transformer Networks (STNs) by Jaderberg et al. [90]. In their work they propose a sub-network which can be inserted between any layers of a network flow. Their sub-network is composed of a localiser network which explicitly learns a spatial transformation of the input to reduce the overall loss. Using this transformation with a grid-sampler of lower resolution than the input data, we are able to learn a spatial transform

which focuses in (and transforms) on the part of the input which helps minimize the loss. This is valuable because we can automatically focus network attention on elements of the image which is most relevant to our task, as well as their spatial transformation. For our work this is relevant as we are also outputting a transformation (in our case a homography) from a deep network to transform a grid of regular points for sampling input imagery to synthesise views.

Intersecting work from Zhou et al. [283] frames view synthesis from the perspective of consecutive views of a scene or object as being highly dependent on each other. They propose a learning scheme that directly predicts vectors which outline which pixels in the input view count towards reconstructing the target view, with their focus being on synthesizing accurate views rather than accurate depth or ego-motion. They argue that this "appearance flow" perspective improves performance but has weaknesses such as difficulty in handling pixel gaps in the input images, difficulty with correlating far apart images, and prior knowledge of object category.

View synthesis was incorporated into a deep learning setting by Garg et al. [65]. Motivated by the value of not requiring to manually label a vast quantity of data, they propose a CNN which takes a stereo pair (of known pose shift) and outputs a depth map. The idea of view synthesis is often to use the depth and pose to warp one of the stereo input images unto its partner to synthesize what the view should be. At this stage generally a photometric loss is calculated between the target and synthesized view for training (see Section 2.2.6). An early self-supervised relative pose approach by Zhou et al. [282] take this idea further by not requiring the pose to be known. They use two separate CNN to predict depth from a target image and pose from nearby source images, and achieve comparable results to supervised methods but with an entirely unlabelled video sequence. One down side is that intrinsics of the camera need to be known and, as previously discussed, estimating many thousands more parameters for dense depth adds significant complexity.

Mahjourian et al. [141] go further by combining a 3D-based loss with the 2D based view synthesis loss. Taking up on suggestions by Zhou et al. [282] to learn a 3D representation (as opposed to solely 2D depth maps), they develop a system which now has the strength of using uncalibrated monocular video (without knowing intrinsics) to predict depth and ego-motion. Rather than following a voxel representation they take the estimated 2D depth maps to generate 3D point clouds of both views and use a registration technique, Iterative Closest Point (ICP), to build a loss function which forces consistency between two consecutive frames. The idea is to use ICP loss to generate gradients which better align the depth and relative pose (ego-motion) estimates. The novel ICP loss developed here helps align 3D structures across consecutive frames. Furthermore, Mahjourian et al. [141] highlight that

their method has the strength of not requiring a calibrated camera, rectification, and is strong against lens distortion, and so widely available internet videos can be used. Additionally, they highlight that they do not explicitly model dynamic scenes or objects. Again, a downside here is that the estimation of depth requires many more parameters to be estimated and for standard geometric constraints to be learnt implicitly from data.

In many cases, issues with view synthesis supervision is that it often relies upon a number of assumptions: a static scene, absence of occlusion between views, and Lambertian surfaces [257, 282]. Moreover, these methods fail to model dynamics of the 3D scene, and Zhou et al. [282] suggest using motion segmentation (as illustrated previously by Ranftl et al. [186] and Vijayanarasimhan et al. [225]). Furthermore, camera intrinsics are sometimes assumed but Mahjourian et al. [141] overcome this. Finally in cases where 2D depth maps are predicted, Zhou et al. [283] suggest this is too simplistic and that learning a voxel based representation (as by Tulsiani et al. [222]) could be worth researching further.

Most of the leading road scene self-supervised relative pose approaches (see Section 2.2.2) utilise view synthesis to form a training signal but fail to leverage regularity in the planar road geometry to do so, instead choosing to train additionally for dense depth estimation, which significantly increases task complexity [68]. Recently, Zhao et al. [277] explicitly utilise plane based pose priors as input to their network for depth estimation to show that camera pose is highly significant for this task. This suggests that we could explore leveraging similar prior understanding towards the task of relative pose estimation itself, which is the focus of our work in Chapter 4.

Moreover, Zhao et al. [277] utilise their planar model for homographic transformation for view synthesis towards data augmentation, though their approach is limited to rotational view synthesis. We see a lack of literature which models a similar kind of explicit model based learning within the setting of self-supervised relative pose estimation [98]. In particular, there is an opportunity to leverage the local planarity around cameras overlooking the same scene towards forming homographies for view synthesis and ultimately to form a training signal, without requiring dense depth estimation networks. This brings us to the related field of deep homography estimation, which we discuss next.

2.2.4 Homography Estimation

One way of encoding geometry is with regards to planar geometry between a camera pair and the homography which describes the transformation of pixels belonging to a plane between both images [181]. In this thesis, we are concerned primarily with

imagery of road scenes which are largely planar due to the planar nature of the road and surrounding elements such as walkways. Homography estimation is relevant to our work because it is the transformation which describes the mapping between these planar points as viewed from varying camera perspectives. Furthermore, we will use an estimated homography to transform one image into the perspective of another for the purposes of forming a supervision signal to train a CNN. In addition, we note that the task of homography estimation is tightly coupled with relative pose estimation between cameras capturing the same planar surface [75, 142, 181].

Traditionally homography estimation is generally approached by estimating hand crafted features and robustly fitting a homographic transformation [75]. Related learnt methods include SuperPoint [48], SOSNet [217] and LIFT [255]. However, we will focus our attention on the more recent and relevant deep learning approaches which estimate homographies explicitly.

The first prominent deep learning approach by DeTone et al. [47] uses a VGG-like [202] CNN to directly regress homography from an image pair using Euclidean loss image warping in an end-to-end manner. Results show deeply learned homography estimation is viable and sometimes improves upon traditional methods based upon ORB [194] and RANSAC [61]. Nowruzi et al. [56] show that accuracy can be further improved by using a hierarchy of Siamese networks upon the input pair to sequentially improve the estimate. We note in particular that Nowruzi et al. [56] highlight the potential use of a similar approach towards the task of odometry estimation. Nguyen et al. [170] propose self-supervised homography estimation via a photometric loss on a perspective warped input pair, although this could be better served with a perceptual loss (see Section 2.2.6) and a Siamese architecture (see Section 2.2.5). Le et al. [111] propose a multi-scale deep homography estimation method to handle moving objects by jointly estimating dynamic masks. Similarly, Zhang et al. [272] propose a novel CNN architecture where masks are learnt in parallel to the feature extractors for the purpose of tackling content such as moving objects. Nie et al. [171] propose a multi-scale iterative deep pipeline for learning homography estimation between two images and subsequently an edge-preservation deep network for learning to accurately stitch these images.

Currently, Yoon et al. [258] achieve state-of-the-art performance for homography estimation on the Oxford and Paris dataset [184] by improving contextual description of line features with transformer networks. Zeng et al. [269] achieve state-of-the-art performance on the Synthetic COCO [47] dataset by framing homographies as perspective fields output from a fully convolutional network, avoiding the need for parameter heavy fully connected layers. Koguciuk et al. [103] achieve state-of-the-art performance on the Photometrically Distorted Synthetic COCO dataset [103] by using a perceptual loss to improve illumination robustness and to

allow for larger view point variations. In their work they propose extensions on details surrounding how to formulate the loss for homography estimation when warping source and target images, leveraging the invertibility of homographies for additional constraints. Cao et al. [22] leverage an iterative homography estimation method which uses a two-scale cascaded pipeline (with a Siamese style architecture and correlation volume as discussed in Section 2.2.5) to achieve state-of-the-art performance on the MSCOCO dataset [126].

Recently, Ye et al. [254] formulate unsupervised deep homography estimation as a weighted sum of predefined optical flow bases. These weights are output from a Siamese style CNN, in similar style to Rocco et al. [190] and Zhang et al. [272], with a novel layer to aid extraction of features for dominant motions. We note that they concatenate feature maps from the feature extractor and that they could benefit instead by fusing them with a correlation volume (see Section 2.2.5). Hong et al. [82] build on their work by replacing the homography regression component of the Siamese architecture with a multi-scale transformer pipeline and leverage a Generative Adversarial Network (GAN) to focus training on the dominant plane in the scene to achieve state-of-the-art performance on the natural image dataset from Zhang et al. [272].

Most of these methods focus on the architecture and loss formulation surrounding deep homography estimation between a pair of images overlooking scenes of widely varying types, from aerial images [22] to rich outdoor scenes [272]. Each individual scene could contain various planar surfaces, each modelled by a different homography [181]. For road scenes in particular we have scenes which all contain the same dominant and consistent planar surface, namely the road itself. The current literature lacks an obvious factor: linking homography estimation with relative pose for cameras capturing planar road geometry. Specifically, for two cameras overlooking the same dominant planar road, there lacks work around parameterising network output as a ground-relative translation and rotation, which can subsequently be transformed into a homography mapping road plane points between both images (useful for self-supervision and potentially other tasks such as orthomosaicing [214]), which we explore in Chapter 4. Furthermore, we find that there is a lack of work around decomposing homographies [142] into camera-relative translation and rotation for the purpose of visual odometry and for aiding the training of deep networks, which we explore in Chapter 5. Moreover, we note that much of the deep homography literature focus on architecture and particularly on using Siamese networks for input, which seems lacking in the self-supervised relative pose estimation literature (see Section 2.2.2).

2.2.5 Architecture and Input

In this section we briefly review works around architecture and network input for tasks such as stereo matching, homography estimation and visual odometry.

Zbontar & LeCun [266] successfully train a CNN for stereo matching with errors generally within 5% on the road scene KITTI dataset [66]. Their method takes a stereo pair of image patches and outputs a similarity measure, which is then used in producing the disparity map using well accepted stereo algorithms. The architecture takes both left and right input patches through a series of convolutional layers (with the same parameters - sometimes referred to as a Siamese network), concatenates the resulting feature maps, which then proceeds through a series of fully connected layers before outputting the desired similarity measure. The strength of this method is that they show accurate results, but with the weakness of having slow inference, which is a problem for responsiveness required in applications such as autonomous driving. Further, concatenating feature maps assumes input pair features are similarly localised, whereas perhaps a correlation volume (as used by Rocco et al. [190]) could be more flexible.

Building on their work, Luo et al. [138] propose removing the time expensive fully connected layers in favour of a single inner product layer and brings inference time down from a minute to under a second. Zbontar & LeCun [267] concurrently publish a paper illustrating a similarly fast network using a similar inner-product layer method as Luo et al. [138] but without training using probability distributions. The probability distribution training used by Luo et al. [138] potentially achieves more accurate results than the simple output of a similarity score alone as they attempt to capture disparities between all combinations of pixels.

Many relative pose estimation approaches will concatenate input images to a pose network [187, 190, 282]. A problem with this approach is that if the variation in pose between both views is sufficiently large, the receptive field of the convolutional layers could be insufficient to capture matching features in both concatenated images [137, 200]. Concatenating both input images therefore constrains pose variation between both cameras, which limits the flexibility of the estimated relative pose [56, 101, 276]. Most self-supervised relative pose estimation approaches limit themselves to temporally constrained and adjacent images [67, 68, 187, 287]. Work by Zhao et al. [276] shows that when many of the leading visual odometry methods are tested on image-pairs with a much larger pose separation, their performance degrades very significantly. Moverover, most self-supervised relative pose methods further constrain pose between image frames to be small due to the necessity of a 2D smoothness prior on depth or optical flow network estimates [68, 187, 257, 276,

282, 287]. Recently, Jia et al. [93] propose to use a smoothness prior instead on a 3D point cloud to predict more natural depth maps.

Rocco et al. [190] use a geometric matching architecture for directly estimating a geometric transformation to synthetically warp object instances into a similar perspective. Inspired by traditional feature matching pipelines, their architecture consists of separate feature extraction branches with shared weights, and a novel matching layer, essentially allowing regression based on putative feature matches between both images. In Chapter 4 and 5 of this thesis, we chose the Rocco et al. [190] architecture firstly due to the effectiveness of their results in capturing features and correspondences which accurately convey geometric perspective, which is relevant for our task of pose estimation. Secondly the separate feature extraction branches avoids the use of concatenating the input and so aligns with our thesis of allowing arbitrary pose estimates.

In addition, Rocco et al. [190] fuse the feature maps output from the Siamese feature extractors in a novel network correlation layer operation. This serves to create a correlation volume which captures tentative feature matches between one point in feature space and every other spatial feature point (see Section 4.2.4). Rocco et al. [190] show that this correlation operation outperforms the alternative of simply concatenating feature maps. Because there is a lack of Siamese architectures within the visual odometry literature [1, 3, 98], this correlation volume has never been used within motion estimation directly, although we note that Cao et al. [22] use it for deep homography estimation to obtain state-of-the-art performance on the MSCOCO dataset [126].

Work by Dong et al. [52] uses the geometric matching network by Rocco et al. [190] to estimate a thin plate spline directly for their human-pose system for trying on clothing. To the best of our knowledge, we are the first to use the Rocco et al. [190] geometric matching architecture for the task of 3D relative pose estimation, as opposed to 2D warping based direct transformation functions. Furthermore, our ground-relative output parameterisation of this network allows for an essential part of our work, which is the formation of homographies relevant for ground-plane cross-projection.

Furthermore, we note that many of the various deep homography estimation methods in Section 2.2.4 use a Siamese architecture to independently extract features from an image pair [22, 56, 82, 171, 254, 272]. Homographies are tightly related to relative camera pose [75, 142, 181], and yet there is less obvious deep relative pose literature using Siamese architectures [1, 3, 98]. Our method in this thesis is to allow for estimation of arbitrary poses and therefore we favour a network pipeline which extracts features independently of each image in a paired input [52, 56, 101, 113, 190].

In this section we have discussed literature around the input and architecture of deep approaches towards tasks related to scene geometry. In Chapter 4 we propose utilising a Siamese network to estimate camera pose around a local planar geometry of the road scene towards view synthesis and to form an appearance based self-supervised training signal. In the next section we discuss work around such an appearance loss, and specifically for perceptual loss.

2.2.6 Perceptual Loss

Many methods for self-supervised tasks involving synthesised views will form an appearance loss as a photometric error between two images where a euclidean difference is computed at the pixel-level, usually with a Structured Similarity (SSIM) term to compute image similarity [68, 170, 233, 276, 282, 287]. The loss used for training self-supervised relative pose or depth estimation is most often formed by a summation of appearance loss and a smoothness regularisation term for depth estimation [68, 282, 287]. Nguyun et al. [170] use a L1 photometric loss for the task of self-supervised homography estimation.

Popularised by work in style transfer and image denoising [94, 252], alternatively a perceptual loss can be used instead of a per-pixel loss for the image difference, where both images are passed through a pre-trained feature extractor and a euclidean error is computed between the resulting feature maps, and which provides a wider basin of convergence [183, 199]. Wang et al. [227] use perceptual loss with another network (in addition to pose and depth networks) to estimate a mask to overcome vanishing-gradient issues caused by dynamic objects and non-Lambertian surfaces, but this requires an additional regularisation term, further complicating the training process. Additionally, their focus is on depth estimation and they do not evaluate for visual odometry.

Perceptual loss robustness to illumination issues is leveraged for various works including homography [103] or depth [130, 227] estimation, multispectral image classification [199], cleaning document imagery [50], low light reconstruction [46], dynamic background removal [209], person identification [86], and super-resolution [183]. However, while perceptual loss is used to achieve leading performance for homography estimation [103], there is a lack of work utilising it for the related task of relative pose estimation with a primary focus on visual odometry evaluation [98]. In Chapter 4 we propose to leverage perceptual loss for our self-supervised deep pipeline which integrates both relative pose estimation and homographies.

While perceptual loss has been used successfully as a training signal, we note its reliance on taking a difference within feature space, which may still limit performance fine grained training of deep pipelines [94, 103]. In the next section we

explore literature around the idea of integrating traditional model-fitting into the training loop.

2.2.7 Geometry and Model-fitting In-the-loop

In this section we discuss a few works around the idea of applying traditional model-fitting within the training process of a neural network. Previous works leveraging the idea of combining various classical optimisations with neural networks range from using a CRF with semantic segmentation networks [27], a graphical model with human pose estimation [221], and Markov Random Fields (MRF) with 3D reconstruction [178]. Furthermore, Kolotouros et al. [105] use a classical model-fitting in-the-loop approach for the task of human pose reconstruction. They recognise that coarse initialisation from a deep model can be a good initialisation for model-fitting optimisation methods, which in turn can help supervise the network for refined performance.

Recently, Henning et al. [80] recently leverage the Kolotouros et al. [105] human mesh regressor for jointly optimising a human mesh with the motion of a camera. While various literature for reconstruction and human pose estimation [15, 39, 123, 244] lead on from the work by Kolotouros et al. [105], it appears that this traditional model-fitting in-the-loop idea is not generally used within the self-supervised relative pose estimation literature (see Section 2.2.2) and could apply well to work with our concept of leveraging a planar geometric model for road scenes. In Chapter 5 our approach is to estimate homographies from optical flow generated point-correspondences to help further refine our perceptual loss trained model and allow for inference time refinement. As far as we know, we are the first to apply the concept to the motion estimation setting with homographies [98].

2.3 Conclusions

There is a gap in the literature where the hierarchical relationship between class training labels is used to construct a neural network training loss to explicitly differentiate between serious and minor errors for road scene semantic segmentation: In Section 2.1.5 we discussed literature relating to hierarchical relationships around classes, with varying approaches from constructing hierarchy within the neural architecture, capturing order or learning the hierarchy itself [25, 31, 45, 58, 136, 149, 164, 173, 193, 206, 247, 253, 284, 286]. In Section 2.1.4 we discussed literature around scene understanding with semantic segmentation. In particular, road scenes are rich and regular with semantic information [19, 98,

149] and as far as we know, none of the related literature use the prior knowledge of class hierarchy labels to assist training a deep network for road scene semantic segmentation [1, 3, 91, 153, 218]. In Chapter 3, we propose a loss function which captures differences intuitively when thinking in terms of hierarchical structuring of the input labels, particularly to the task of semantic segmentation for road scenes. Our idea is that by differentiating in the loss function between less and more serious classification errors we can obtain improved performance.

Relative pose estimation literature focus on camera-relative motion: In Section 2.2.1 we discuss camera pose estimation within the deep learning domain. Deep pose approaches generally focus on outputting a 6 DoF camera-relative pose [13, 67, 68, 99, 121, 148, 187, 225, 231, 257, 276, 282, 287]. We propose in Chapter 4 to estimate camera pose relative to the local physical geometry and investigate the viability of utilising this more general parameterisation.

Self-supervised relative pose methods choose to learn scene geometry implicitly with dense depth or optical flow networks: In Section 2.2.2 we discussed leading self-supervised visual odometry approaches [13, 68, 187, 257, 276, 282, 287]. Self-supervisory signals are generally implemented via view synthesis with a photometric error [67, 68, 287]. Some approaches additionally use dense optical flow or motion segmentation networks [187, 257, 276]. Attempting to simultaneously learn robust features for depth or optical flow with pose to estimate tens of thousands of parameters is a challenging task, and potentially liable to overfit known scene regularity [51, 68, 277].

Self-supervised relative pose methods do not explicitly leverage the known planarity of road scenes: Particularly, none of the leading visual odometry approaches [13, 68, 187, 257, 276, 282, 287] utilise the basic known geometry in road scenes: the ground is approximately planar. In Chapter 4 we propose to parameterise with respect to the ground plane, and cross projecting via that known geometry to form the training loss, avoiding the requirement of estimating dense depth with a second network all together.

A majority of relative pose deep learning pipelines limit the extent of camera-relative pose by concatenating network input: In Section 2.2.5 we reviewed types of neural architectures used for the task of relative pose estimation [52, 93, 101, 113, 137, 187, 190, 200, 257, 282, 287]. For tasks like relative pose estimation which often take pairs of images as input [13, 68, 276], concatenation of input assumes that motion is small between each image due to the limited receptive field of convolution layers [101, 276]. We note that more consideration should be given to network input and towards using a Siamese architecture [101, 113, 190], as it could allow for more arbitrary relative poses. Furthermore, this flexibility could

be useful for opening up the relative pose estimation field into applications which may utilise a wider perspective [20, 81, 180].

Perceptual loss has yet to be applied to training neural networks towards visual odometry evaluation: In Section 2.2.6 we discussed literature around using perceptual loss for training a neural network [46, 50, 68, 86, 94, 103, 130, 199, 209, 227, 233, 252, 276, 282, 287]. We note that deep self-supervised relative pose estimation methods tend to always use pixel level losses with strong and restricting regularization terms [68, 233, 276, 282, 287]. It makes sense to operate losses on the feature level as we are less reliant on these terms and additionally convergence and issues of illumination should be less of a concern [46, 50, 86, 103, 130, 199, 209, 227]. While perceptual loss has been applied to depth and homgography estimation [103, 130, 227], to the best of our knowledge we are the first to use a perceptual loss with a primary focus on visual odometry evaluation (see Chapter 4).

Deep homography estimation literature lacks a direct link to relative pose estimation for cameras capturing planar road geometry: In Section 2.2.4 we discussed approaches for deep homography estimation [22, 47, 56, 82, 103, 111, 170, 171, 254, 258, 272]. We note that while there exist various works around the issue of dynamic content [82, 111, 254, 272], architecture refinement [22, 56, 82, 171] and loss formulation [82, 103, 170], there is a lack of application directly to relative pose estimation rooted in the surrounding planar geometry of two or more cameras. We explore the potential of such a parameterisation in Chapters 4 and 5.

There is a lack of work utilising classical model-fitting for in-the-loop supervision of motion estimation networks: In Section 2.2.7 we discussed literature around the idea of utilising traditional model-fitting within the training of a neural network, an approach leveraged most prominently by Kolotouros et al. [105] for fitting human pose models. While other methods for body modelling [15, 39, 123, 244] have followed from Kolotouros et al. [105], as far as we know, we are the first to apply the concept to motion estimation setting with homographies. In Chapter 5 we choose to use a model-fitting in-the-loop approach to help further refine our learned homographic model and allow for inference time refinement.

2.3.1 Summary

Road scenes are rich with semantic information and yet most deep semantic segmentation methods for road scenes do not train with a loss which explicitly captures hierarchical relationships between classes [1, 3, 91, 153, 218]. In Chapter 3 we propose a hierarchical loss which can be utilised with any standard classification pipeline, and which differentiates between minor and major classification errors,

something surprisingly overlooked within the autonomous driving literature [1, 3, 20, 91, 153, 180].

Self-supervised deep pose estimators for road scenes choose to implicitly learn road geometry jointly with either dense depth or optical flow estimation [13, 187, 276, 287]. Dijk et al. [51] show that common road scene depth networks [68] often trivially learn depth of objects by simply using their vertical position in the image, and Zhao et al. [277] show that road plane pose priors as network input have a significant contribution towards road scene depth estimation. Given this and our review of the related literature (see Sections 2.2.1 and 2.2.2), we feel there is a lack of work explicitly modelling planar road geometry towards solving deep motion estimation with only a single pose network, avoiding the gross over-parameterisation from dense depth or optical flow estimation [3, 98].

Network input and training also stood out as potential areas of improvement (see Sections 2.2.5 and 2.2.6 respectively). In particular, while deep homography estimation approaches have leveraged the Siamese architecture to independently extract image features [22, 56, 171, 254], deep pose estimators tend to concatenate input [13, 68, 276], potentially limiting the physical range of relative pose estimators [101]. In addition, while perceptual loss has been applied to deep homography estimation [103] to achieve state-of-the-art performance, and to depth estimation [130, 227], it has yet to be applied to motion estimation with a primary focus on visual odometry evaluation.

Moreover, relative pose literature focuses heavily on camera-relative pose [187, 257, 276, 287], and there is lack of work exploring estimating pose relative to the geometry of the 3D environment itself [98]. Furthermore, relative pose estimation between cameras overlooking the same planar surface is highly related to homography estimation [142]. We see a lack of literature which directly links relative camera pose estimation within approximately planar geometries (such as that captured in road scenes) to homographies in a deep learning pipeline [56, 98].

In Chapter 4 we address these gaps by using a perceptual loss to train a single Siamese CNN to estimate a ground-relative pose parameterisation which can be transformed to compute homographies capturing planar road scene motion. We do this in an entirely self-supervised manner, avoiding any reliance on collecting ground-truth and leveraging the road geometry for homographic planar cross-projection for self-supervision.

In Section 2.1.4 and 2.2.2 we reviewed key literature towards semantic segmentation and self-supervised relative pose estimation respectively. We observe a lack of literature towards combining pre-trained semantic segmentation models with self-supervised visual odometry pipelines for the purpose of filtering out scene

content which does not belong to the geometry used to form a self-supervision signal, which in our case, is the planarity of the local road geometry.

Furthermore, we recognise that fitting a geometric model in a traditional approach has benefited the training or inference of deep pipelines [15, 105, 123], and that there is a lack of application of this technique towards deep pose estimation, specifically with respect to homographic model-fitting [42]. In addition, we note that given homography estimation between two images overlooking the same planar surface is highly related to relative pose [142], there is a lack of work exploring the analytical decomposition of such a homography into relative pose within a deep learning setting [179]. In Chapter 5 we seek to rectify these gaps by combining them into a homography refinement module to further improve our performance from Chapter 4.

3 A Hierarchical Loss for Semantic Segmentation

The visual world is full of structure, from relationships between objects to scenes and objects composed of hierarchical parts. For example, at the most abstract level, a road scene could be segmented into three parts: ground plane, objects on the ground plane and the sky. The next finer level of abstraction might differentiate the ground plane into road and pavement, then the road into lanes, white lines and so on. Human perception exploits this structure in order to reason abstractly without having to cope with the deluge of information when all features and parts are considered simultaneously [226]. Moreover, it is quite easy for a human to describe this structure in a consistent way and to reflect it in annotations or labels [38, 66].

It is therefore surprising that the vast majority of work on learning-based object recognition, object detection, semantic segmentation and many other tasks completely ignores this structure [7, 68, 99, 156, 176]. Classification tasks are usually solved with a flat class hierarchy, but in this thesis we seek to utilise hierarchical class structure for direct supervision [72, 156].

Another motivation is that there is often inhomogeneity between datasets in terms of labelling. For example, the LFW parts label database [96] segments face images into background, hair and skin, while the segment annotations for the Helen [112, 205] dataset define 11 segments. Utilising both datasets to train a single network, while retaining the richness of the labels in the latter one, is not straightforward. With our method of training with a hierarchy, we could enable training with both datasets simultaneously, regardless of the differing classes, boosting the amount of data available or allowing more flexible annotating. Depending on the application, we may also wish to be able to vary the granularity of labels provided by the same network, and a hierarchical relationship between classes could allow this.

We tackle the problem of semantic segmentation and introduce the idea of hierarchical classification losses. The idea is straightforward. For any existing semantic segmentation architecture that outputs one logit per class per pixel, we can compute a loss at each level of abstraction within a provided class hierarchy, and sum these to form an overall loss. The benefit is to differentiate serious errors from less serious. In the toy example shown in Fig. 3.1, a *face*/*hair* error would be penalised less severely than a *background*/*face* error since both *face* and hair

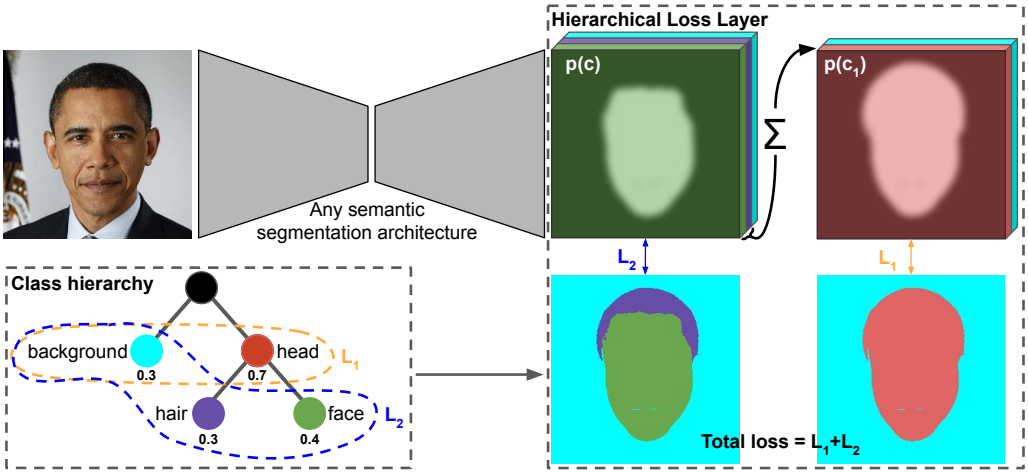

Figure 3.1: Overview of our idea. Given the output of any semantic segmentation architecture and a class hierarchy, we compute losses for each level of abstraction within the hierarchy, inferring probabilities of superclasses from their children.

belong to the superclass *head* and so L_1 would not penalise the error. The network is encouraged to learn visual features that are shared between classes belonging to the same superclass, i.e. the knowledge conveyed by the class hierarchy allows the network to exploit regularity in appearance. Since coarser classification into fewer abstract classes is presumably simpler than finescale classification, it also means that the learning process can naturally proceed in a coarse to fine manner, learning the more abstract classes earlier.

Our contributions for this chapter are:

1. Exploiting semantic knowledge of class hierarchies, we can separate training between serious and minor errors.

2. A novel formulation for a general hierarchical loss for classification tasks, which we evaluate on semantic segmentation.

3. Illustration of the potential training benefits over classification methods which use a flat hierarchy.

4. A formulation for approaching numerical instability challenges during our implementation.

In Section 3.1 we emphasise the hierarchical design and simplicity of the prior knowledge we are proposing to use. In Section 3.2 we detail the mathematics behind

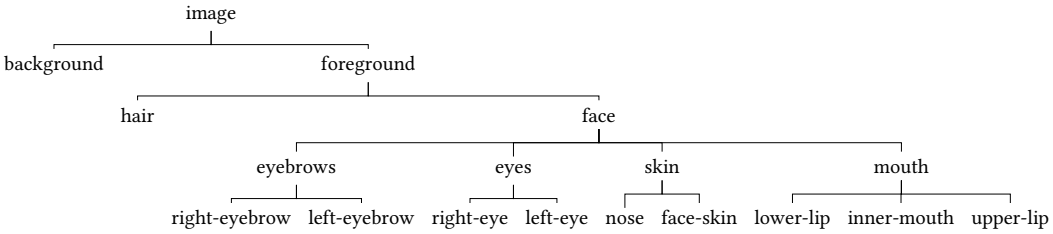

Figure 3.2: Our hierarchy for the Helen [112] segment classes. Note that the classes in the original dataset [205] are the leaves in our hierarchy.

our specific approach for our hierarchical loss. In Section 3.4 we provide details on experiments and results for our two evaluation datasets. We use the Helen [112] facial dataset as a toy example which involves only a few classes with a simple hierarchy, and the Mapillary Vistas [169] road scene dataset as our primary focus for road scene understanding.

3.1 Hierarchy Design

Our approach requires a predefined hierarchy, which we assume is designed based on expert human knowledge. In the case of objects composed of parts, this is straightforward since the parts can naturally be described hierarchically. For more general scenes this may require specific domain knowledge in order to be able to group related objects together into the same superclass. We emphasise two points. First, the practical effort of doing this is extremely low. We do not require any new annotation of the training images, there is simply a one off task to design a hierarchy for the classes already used in the labelling. Second, many existing datasets were annotated with a hierarchical class structure in mind (even if this is rarely used). For example, the COCO-stuff dataset [30] clusters each of the 172 classes into 11 abstract groups, providing a shallow hierarchy.

For the experiments in this thesis we use two datasets that represent each of the cases above. Our goal is to focus our research on how we can use the label hierarchy of road scenes specifically towards improving semantic segmentation in that domain, and in Chapter 5 we will explain our work towards combining semantics with geometric understanding.

In order to experiment with our idea of a hierarchical loss we chose to firstly to use a toy dataset where the number of classes is much lower than road scene datasets [38, 66, 169] and where the hierarchy is simple. For this purpose we use the Helen [112] facial dataset. The segment annotations [205] for the Helen [112]

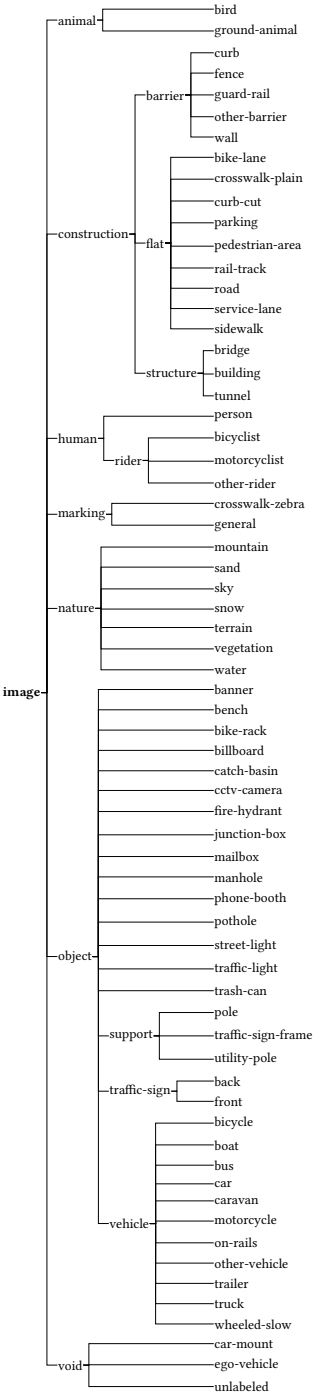

Figure 3.3: Our hierarchy for the Mapillary Vistas [169] segment classes.

dataset are not provided with any hierarchy. However, there is an obvious parts-based partitioning such that the classes used in the dataset correspond to the leaves of a hierarchy tree (see Fig. 3.2). To emphasise again: we do not need to relabel the Helen [112] annotations. We simply use the original annotations in conjunction with the hierarchy.

Our primary focus is on the the second dataset, the Mapillary Vistas [169] road scene dataset. This was originally designed with a hierarchical class structure (see Neuhold et al. [169] for details and Fig. 3.3 for the hierarchy) for which some super-classes are based on clustering related objects (for example, the vehicle superclass).

3.2 Hierarchical loss

Our method is based on computing a sum of classification losses over each level of abstraction within a classification hierarchy. In order to use the approach, one simply needs a class hierarchy defined by a tree and a segmentation architecture that outputs a classification for each of the classes corresponding to leaf nodes in the tree. In this section we describe our representation and the hierarchical losses.

3.2.1 Tree-based representation

We represent our class hierarchy using a tree (V, E), where:

$$V = \{v_1, \ldots, v_n\} \tag{3.1}$$

is the set of vertices and $E \subset V \times V$ the set of ordered edges such that $(v_i, v_j) \in E$ encodes that v_i is a parent of v_j. We assume that the first m nodes correspond to leaves in the tree:

$$\nexists\, v_j \in V, (v_i, v_j) \in E \Rightarrow 1 \leq i \leq m. \tag{3.2}$$

These nodes correspond to the finest scale classes. If $(v_i, v_j) \in E$ then v_i is a more general, more abstract, superclass of v_j. The label for a pixel, c, should be at the finest level of classification, i.e. $c \in \{1, \ldots, m\}$.

We define $\text{depth}(v_i)$ to mean the number of edges between vertex v_i and the root node. Hence, the depth of the tree is given by:

$$D_{\max} = \max_i \text{depth}(v_i). \tag{3.3}$$

We define $\text{ancestor}(v_i, v_j)$ to be true if v_i is an ancestor of v_j, i.e. that v_i is a superclass of v_j and false otherwise.

3.2.2 Inferring coarse classes from fine

We assume that the segmentation CNN outputs one logit, x_i, per pixel per leaf node, i.e. the output of the network is of size $H \times W \times m$, where m is the number of fine-scale classes corresponding to the leaf nodes in the hierarchical tree. In our notation, x represents a single pixel and i represents the logit for class i. Hence, the probability, p_i, associated with class node i is computed by applying the *softmax()* function to x_i:

$$\sigma(x_i) = \frac{\exp(x_i)}{\sum_{j=1}^{m} \exp(x_j)} \tag{3.4}$$

The probability associated with non-leaf nodes is defined recursively by summing the probabilities of its children until leaf nodes are reached:

$$p_i = \begin{cases} \sigma(x_i) & \text{if } 1 \leq i \leq m \\ \sum\limits_{(v_i, v_j) \in E} p_j & \text{otherwise} \end{cases} \tag{3.5}$$

Note that any summation is over a subset of leaf-nodes whose total sum is one so any p_i is ≤ 1.

3.2.3 Depth dependent losses

We define our loss with respect to multiple discrete abstraction levels within our hierarchy tree as depicted in Fig. 3.1 for the simplified example. Our method is to compute a loss for each abstraction level in terms of the deepest branch in the hierarchy of classes. For example, given the hierarchy in Fig. 3.2, $L_{D_{\max}}$ considers all classes equally. For L_3, we consider *background* and *hair* as individual probabilities, but probabilities from the other leaf classes are summed and applied to their immediate parent classes. For L_2, we again retain probabilities for *background* and *hair*, but only further consider the the abstract *face* class by inferring probabilities from its children. Finally, for L_1, we retain the probability for *background* but infer the probability for *foreground* by summing all of the probabilities associated with its children.

The total hierarchical loss is then a summation over all of these abstraction levels $d \in \{1, \ldots, D_{\max}\}$ in the tree (where L_1 and $L_{D_{\max}}$ represent the highest and lowest level of abstraction respectively):

$$L = \sum_{d=1}^{D_{\max}} L_d. \tag{3.6}$$

The classification loss at abstraction level d is computed using the negative log loss:

$$L_d = -\log(p_{c_d}). \tag{3.7}$$

where p_{c_d} is the probability which our network estimates for the correct class node c_d for the given abstraction level d, as given by Eqn. (3.5). The appropriate class node c_d varies depending on the abstraction level d that we are computing. We define the correct class node (either a leaf or abstract class) for abstraction level d to be:

$$c_d = \begin{cases} v_c & \text{if } d = D_{max} \\ v_k : \text{anc}(v_k, v_c) \wedge \text{depth}(v_k) = d & \text{otherwise} \end{cases} \tag{3.8}$$

where v_c is the correct leaf vertex class label given by the relevant ground truth for the non-abstract classes, $\text{anc}(v_k, v_c)$ requires v_k to be a superclass of v_c and $\text{depth}(v_k) = d$ picks a specific superclass given the abstraction level d.

3.3 Numerical stability

In this section we explain how to ensure numerical stability in the computation of the abstraction level losses.

Evaluating cross entropy (log) loss of a probability computed using *softmax()* is numerically unstable and can easily lead to overflow or underflow. In most implementations, this is circumvented using the "log-sum-exp trick" [165] derived from the identity:

$$\exp(x) = \exp(x - b + b) = \exp(x - b)\exp(b)$$

as:

$$L = -\log p_i = -\log\left(\frac{\exp(x_i)}{\sum_{j=1}^{m}\exp(x_j)}\right) = -x_i + b + \log\sum_{j=1}^{m}\exp(x_j - b), \tag{3.9}$$

where:

$$b = \max_{i \in \{1,\ldots,m\}} x_i$$

is chosen so that the maximum exponential has value one and thus avoids overflow, while at least one summand will avoid underflow and hence avoid taking a logarithm of zero.

Our hierarchical classification losses involve computing log losses on internal

nodes in the class hierarchy tree. The probabilities in these nodes are in turn formed by summing probabilities computed by applying *softmax()* to CNN outputs. This leads to evaluation of losses of the form:

$$-\log(\sum_{i\in C} p_i)$$

where C is the set of leaf nodes contributing to the superclass. This can be made numerically stable by double application of the log-sum-exp trick:

$$\begin{aligned} L &= -\log \sum_{i\in C} p_i = -\log \frac{\sum_{i\in C} \exp(x_i)}{\sum_{j=1}^{n} \exp(x_j)} \\ &= b + \log \sum_{j=1}^{n} \exp(x_j - b) - b_C - \log \sum_{i\in C} \exp(x_i - b_C), \end{aligned} \tag{3.10}$$

where b is defined as before while:

$$b_C = \max_{i\in C} x_i.$$

The use of two different shifts for the two logarithm terms is required to avoid underflow when $b_C \ll b$.

3.4 Experiments

We seek to investigate the relative performance gain in using the hierarchical loss versus training simply on a flat hierarchy. To this end, in our experiments we train two networks for each task. One is a "vanilla" U-net [191], the other is exactly the same U-net architecture but trained with hierarchical loss (referred to as U-net+HL). We train U-Net/U-Net+HL models simultaneously such that they receive identical data input at each iteration. Note that we do not seek nor achieve state-of-the-art performance. A more complex architecture, problem-specific tuning and so on would lead to improved performance but our goal here is to assess relative performance gain using a simple baseline architecture. Networks use Kaiming uniform initialisation [79] with the same random seed (to equally initialise vanilla and hierarchically trained networks). Pre-training is not utilised. We use Stochastic Gradient Decent with a learning rate of 0.01 and a batch size of 5 (due to memory constraints). During training, images/labels were randomly square-cropped using

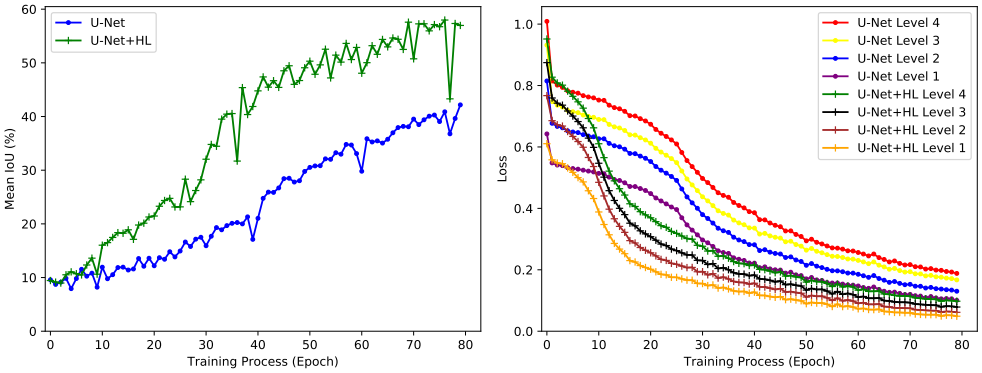

Figure 3.4: Training behaviour versus epoch on the Helen [112] dataset. Left: Mean IOU. In both cases we show results trained with vanilla (U-Net) and hierarchical (U-Net+HL) losses. Right: Classification loss for each depth $D = 1..4$.

the shortest dimension and re-sized to 256^2. The only further data augmentation used was random flipping (p = 0.5).

3.4.1 Datasets

For experimenting with hierarchical losses on segmentation we chose two very different datasets: the Helen [112] facial dataset and the Mapillary Vistas [169] road scene dataset. As a toy example, we chose to use the facial Helen [112] dataset to experiment with our method where the number of classes is low and the hierarchy very simple. The Helen [112] dataset covers a wide variety of facial types (age, ethnicity, colour/grayscale, expression, facial pose), originally built for facial feature localisation [112]. We use an augmented Helen [205] dataset with semantic segmentation labels. Helen [112] contains 2000, 230 and 100 images/annotations for training, validation and testing respectively, for only 11 classes (10 facial and background, see Tab. 3.1(left)). It should be noted that the ground truth annotations are occasionally inaccurate, particularly for hair which makes it challenging to learn.

Our primary focus is on the road scene Mapillary Vistas [169] dataset which is composed of 25000 images/annotations (18000 training, 2000 validation, 5000 testing), with 66 classes. We have chosen a representative subset of Mapillary Vistas [169] classes in Tab. 3.1 (right) which show the most significant differences in performance and have given the mean over all classes. Further, our intention is to indicate the performance improvement by using hierarchical learning, rather

Helen [112] Dataset			Mapillary Vistas [169] Dataset		
Class	**U-Net**	**U-Net+HL**	**Class**	**U-Net**	**U-Net+HL**
Background	92.0	**92.7**	Car	80.4	**80.9**
Face skin	86.5	**87.0**	Terrain	54.2	**56.1**
Left eyebrow	**63.1**	62.7	Lane Marking	49.1	**51.7**
Right eyebrow	63.7	**64.3**	Building	77.3	**79.7**
Left eye	64.0	**67.8**	Road	82.3	**82.7**
Right eye	64.9	**72.7**	Trash Can	5.4	**18.6**
Nose	**84.1**	82.6	Manhole	2.3	**17.0**
Upper lip	52.9	**56.5**	Catch Basin	1.6	**13.6**
Inner mouth	62.2	**67.9**	Snow	57.0	**71.5**
Lower lip	65.6	**67.9**	Person	39.2	**48.3**
Hair	65.4	**66.1**	Water	**29.9**	16.1
Mean	69.49	**71.65**	**Mean**	24.74	**26.51**

Table 3.1: Mean and class IOU (%) on the Helen [112] and Mapillary Vistas [169] (subset selected) datasets at training convergence.

than to compare between datasets. The Mapillary Vistas [169] hierarchy is three levels deep, contains 66 leaf nodes, and 11 internal nodes.

3.4.2 Results

Fig. 3.4 (right) shows losses for each abstraction depth of the class hierarchy for the Helen [112] experiment. Note that the deeper loss is always larger than a shallower one, suggesting that our hierarchically trained method significantly benefits from the hierarchical structure in the class labels, particularly in the early phase of training, learning much faster than the vanilla model. Fig. 3.4 (left) illustrates the mean Intersection over Union (IOU) during training. Performance gain is most significant post epoch 35 and can be observed in the qualitative results from Fig. 3.6. At performance convergence we observe some qualitative differences between the hierarchically trained network and the vanilla. For example, in Fig. 3.6 U-net+HL predictions at epoch 200 have somewhat less hair artefacts, while the 1st example shows improvement over a difficult angled facial pose. Epoch 50 results clearly show faster convergence.

For Mapillary Vistas [169], the IOU performance gain is less notable than on Helen [112], but we show the hierarchically trained model outperforming the vanilla model in both level losses and mean IOU (Fig. 3.5 and Tab. 3.1 (right)). The qualitative results in Fig. 3.7 illustrate predictions for both methods at epoch 1 and

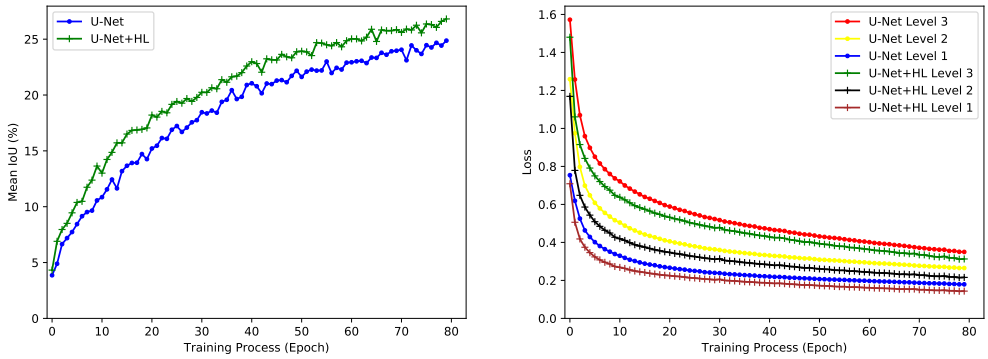

Figure 3.5: Training behaviour on Mapillary Vistas [169]. Left: mean IOU versus epoch. Right: classification loss for each abstraction depth $D = 1..3$ versus epoch. We show results trained with vanilla (U-Net) and hierarchical (U-Net+HL) loss.

80. Most interestingly, after 1 epoch the hierarchically trained model is able to classify correctly a significant proportion of lane-markings whereas the vanilla trained model cannot, showing how quickly our hierarchical model is learning. Relative to the vanilla model, our hierarchically trained model achieves a 3% and 7% relative improvement for Helen [112] and Mapillary Vistas [169] respectively (see Tab. 3.1).

3.5 Conclusions

Our results in this chapter illustrate the potential of using losses that encourage semantically similar classes within a hierarchy to be classified close together, where the model parameters are guided towards a solution not only better quantitatively, but faster in training than using a standard loss implementation.

Training with a semantic hierarchy prior improves performance and accelerates training: Taking advantage of the hierarchical cues readily apparent to us can help train a deep network faster and with greater accuracy. We suggest that the hierarchically trained models perform better due to learning more robust features from visually similar classes which are close within the tree structure. The hierarchy is providing the network with more information (e.g. a pixel belongs to an eye-brow, which belongs to a face and so on), which can be exploited to learn shared and more robust features. Rather than positive and negative class labels, we are provided with classes that can be more or less similar within a hierarchy.

We also contribute a numerically stable formulation for computing log and

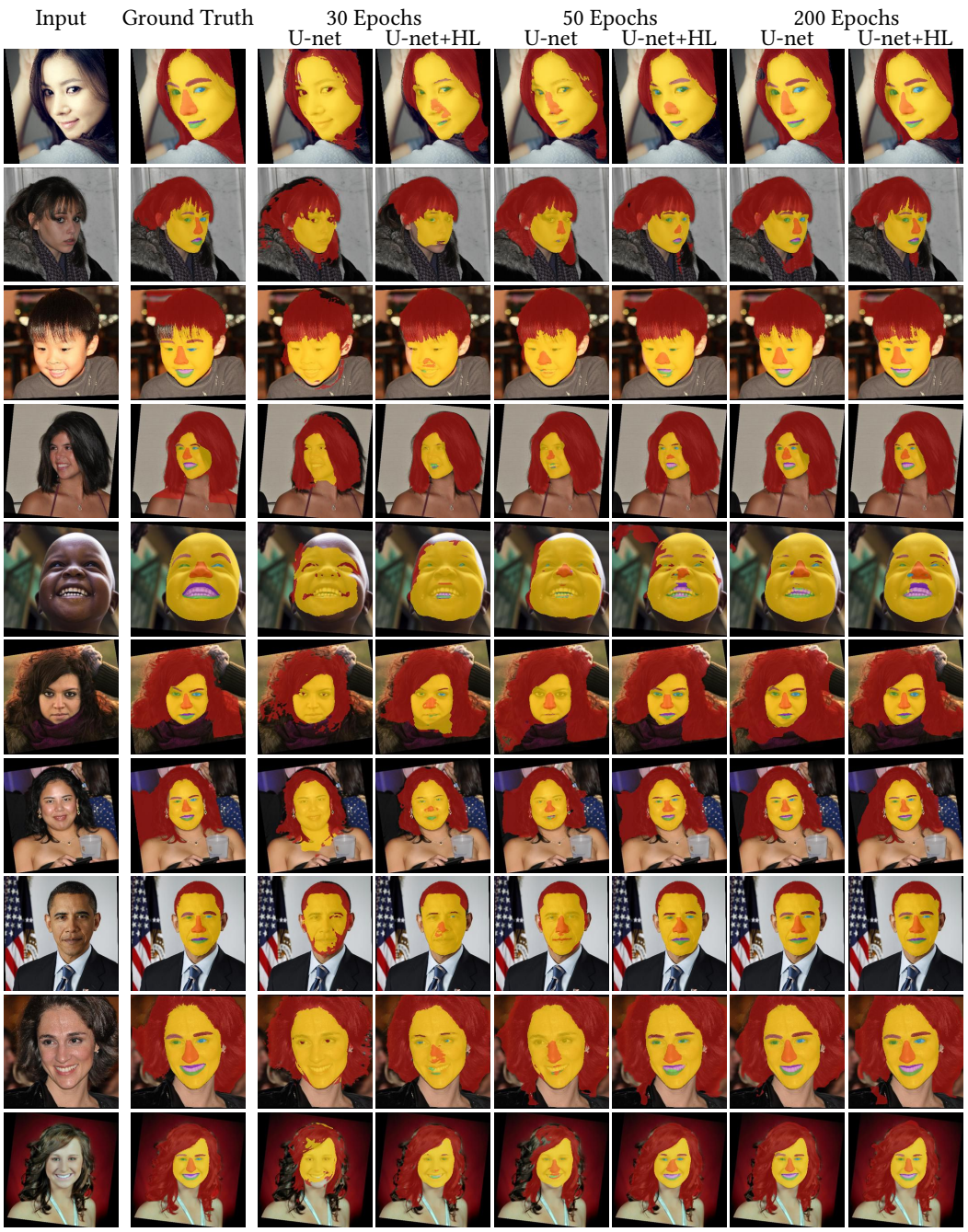

Figure 3.6: Prediction comparisons on the Helen [112] dataset. From left to right: raw input image, ground truth annotation, vanilla trained U-Net prediction at 30 epochs, hierarchically trained U-Net prediction at 30 epochs, vanilla trained U-Net prediction at 50 epochs, hierarchically trained U-Net prediction at 50 epochs, vanilla trained U-Net prediction at 200 epochs, and the hierarchically trained U-Net prediction at 200 epochs.

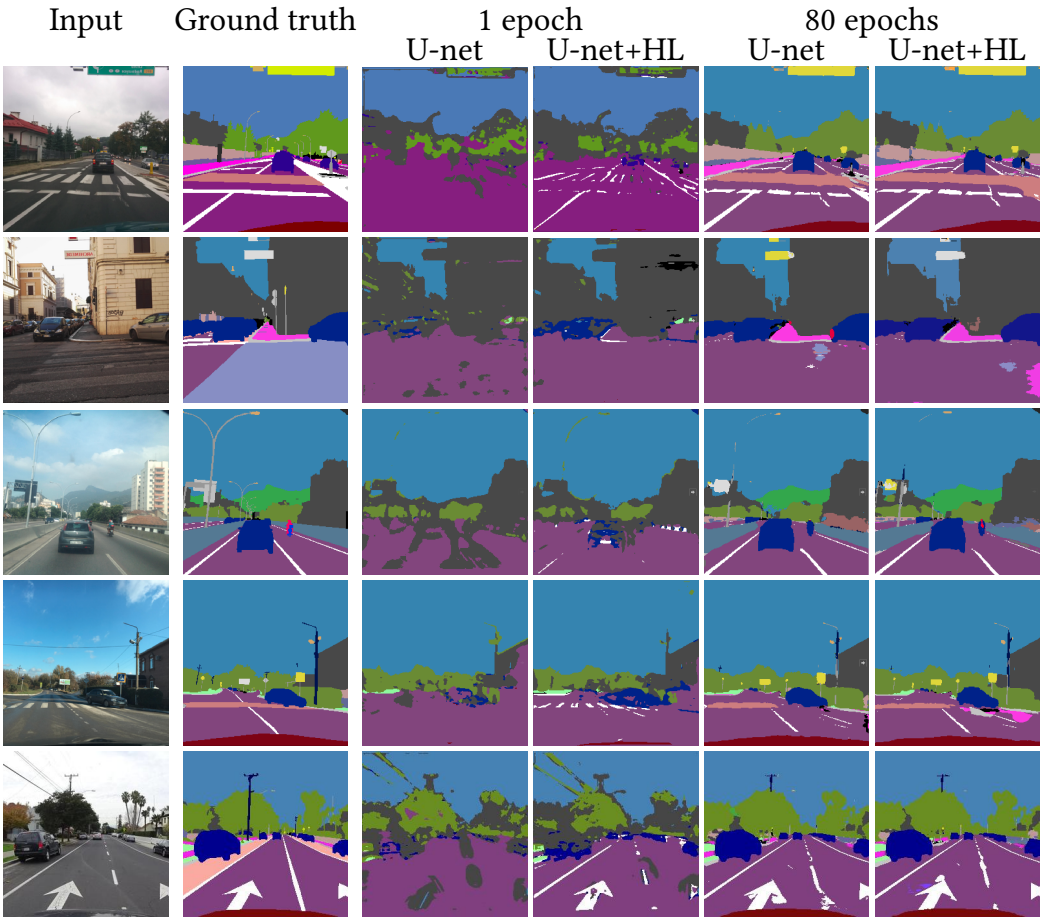

Figure 3.7: Qualitative comparisons on the Mapillary Vistas [169]. From left to right: raw input image, ground truth annotation, vanilla trained U-Net prediction at 1 epoch, hierarchically trained U-Net prediction at 1 epoch, vanilla trained U-Net prediction at 80 epochs, and the hierarchically trained U-Net prediction at 80 epochs.

softmax() of a network output separately, a necessity for summing probabilities according to a hierarchical structure.

Hierarchies can be applied generally: A particular advantage of our method is its generality and self-contained nature allows the possibility of plugging this hierarchical loss on the end of any deep learning architecture. Moreover any hierarchical structure can be provided to help train your model.

Semantic structure is useful. What about geometric regularity? We have shown that we can inform the training of a CNN using a basic understanding of the structural semantics present within images. Specifically we show examples for

road scenes where semantics of classes is always consistent and focused around the road plane itself.

Looking at the imagery present in Fig. 3.7 an obvious question arises of whether we can use the planarity of the road plane itself to help inform training of neural networks or to assist development of methodology for motion related tasks. In Chapter 4 we will show that indeed it is possible to do so, particularly in terms of self-supervised training and focusing on the task of relative pose estimation.

4 Geometry and Pose Estimation with Appearance Loss

Road scenes are highly regular. In the previous chapter we showed that structure within semantic labels is useful for boosting performance for semantic segmentation of road scenes. In this chapter we ask whether geometric domain understanding for road scenes can be used similarly but for the task of motion estimation. Relative pose estimation is important for computer vision applications such as SfM, image stitching and alignment, change detection, visual odometry, augmented and virtual reality, and many others. Particularly, visual odometry is increasingly important for the development of autonomous vehicles.

Many deep learning based visual odometry methods rely on ground truth labels for supervising relative pose models, generally collected by inertial and GPS systems. However, the accuracy is limited, and it suffers from inconsistent coverage. Further, significant effort is required to collect labels and correctly synchronise and geometrically calibrate them relative to camera images. Alternatively, vast amounts of unlabelled driving video is available. Moreover, ground truth labels are noisy and self-supervised methods can potentially achieve higher accuracy. So, there is strong motivation for developing methods that use only image data to learn relative pose in a self-supervised manner.

Self-supervised relative pose methods often use a combination of depth and pose estimation to form a self-supervised signal through cross-projection between source and target images [68, 287]. Other approaches train an optical flow network in addition to depth [187, 276]. While a complete, dense depth model is useful, estimating many thousands of depth values per image is an ill-posed problem, with models liable to overfit to scene contents and adjacent frame-to-frame views with little variation in pose between images in a pair. Furthermore, errors in one task may influence accuracy of the other as, for example, a depth estimator could learn to compensate for the pose estimation being inaccurate. Additionally, many approaches merely use relative pose estimation as a step towards a primary goal of estimating depth.

Moreover, the pose network architecture is less often considered (see Section 2.2.5). Most deep learning based methods tend to concatenate image pairs channel-wise for input to the pose estimation network, which only makes sense when pose changes are small as in that case the receptive field of convolutional layers can observe corresponding points in both images. If the relative pose between these

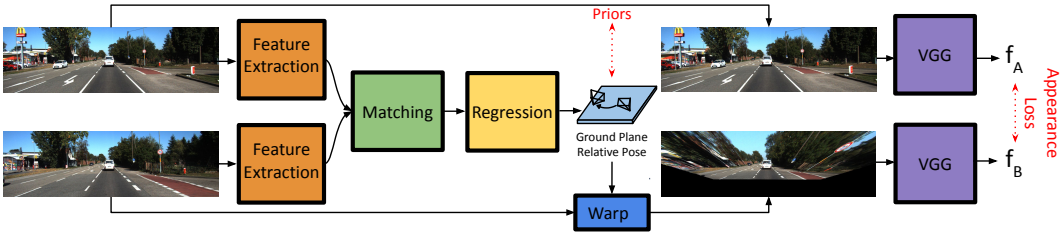

Figure 4.1: Ground-relative pose estimation. We propose to estimate ground-relative pose using a geometric matching network (See Fig. 4.4 for details). Assuming a locally planar environment, we can cross-project one image into the perspective of the other (see Fig. 4.3) using only our 9D ground-relative pose parameterisation (without estimating dense depth). This enables self-supervision via an appearance loss for which we use a perceptual loss based on deep features provided by a pre-trained VGG [202] network. See Section 4.3 and Fig. 4.5 regarding the method for transforming these relative poses into absolute poses.

images is large, the receptive field may not capture matching features, and hence relative motion between both cameras may fail to activate neurons to accurately estimate relative pose [101, 137, 200, 276]. Our approach is to allow for greater flexibility between cameras by separately extracting features from both images in a Siamese style architecture [101, 113] and performing putative feature matching and regression via a Siamese network originally proposed by Rocco et al. [190].

We propose a self-supervised method to learn relative pose estimation *without* the need to estimate depth. Instead, we exploit that road scenes are predominantly planar, which allows for image cross-projection and self-supervision in a similar manner to depth estimation methods but without the need for a depth map. While the planar nature of road scenes might be learnt implicitly by depth estimation networks, here we enforce it. This dramatically simplifies the network task, while retaining the benefits of self-supervision. A benefit of our simplified method is that we can train a single CNN with only 9 parameters for output, versus multiple CNNs with thousands of parameters for output - this is a much easier and faster model to train successfully in practical road odometry. Harnessing the known planarity of road scenes allows us to move away from explicitly modelling the complex and dynamic geometry of buildings, people, cars, signs, and so on. Specifically, we make the following contributions:

1. A physically interpretable and novel 9D ground-relative pose parameterisation for network output, from which scene contents on the ground plane can be cross-projected between images to form a self-supervision signal, without any requirement for a depth network.

2. Using a known geometric matching Siamese network for the purpose of pose regression which can handle arbitrary pose changes on overlapping image-pairs.

3. Application of perceptual loss in the context of motion estimation, which allows for a wide basin of convergence.

Further, we achieve trajectory estimation from relative poses via transformation synchronisation.

In Section 2.2.2 we discussed leading work towards end-to-end self-supervised relative pose estimation as applied to road scenes for the KITTI benchmark [66]. These methods all rely heavily on implicitly learning planar motion from road geometry with dense estimation of depth [13, 68, 282, 287] or additionally include dense optical flow estimation [187, 257, 276]. To the best of our knowledge we are the first self-supervised relative-pose estimation method to use geometric knowledge of road scenes (in our case, that our local road geometry is approximately planar) to facilitate cross-projection for a self-supervision training signal. On the KITTI dataset we show supervising with our simple geometric model achieves competitive performance versus state-of-the-art self-supervised methods that rely on dense depth estimation [13, 68, 187, 276, 287].

In Section 4.1 we detail our ground-relative pose parameterisation, how it allows for camera-relative pose and cross-projection, then discuss scale ambiguity. In Section 4.2 we detail a Siamese network architecture and explain how we train with appearance loss. In Section 4.3 we provide an overview on the method from Arrigoni et al. [6] which optimises a collection of camera-relative poses for absolute pose useful for our visual odometry evaluation. Finally, in Section 4.4 we illustrate our experimental results on the KITTI road scene dataset.

4.1 Two View Ground-Relative Geometry

We propose to predict relative pose between a pair of views and also the *positioning of the two views relative to the local ground plane*. See Fig. 4.2 for an illustration of our novel ground-relative pose parameterisation, which has 9 degrees of freedom: 6 for relative pose and 3 to define the plane relative to one of the cameras. In this section, we describe how we parameterise ground-relative pose, how we extract camera-relative pose from this formulation, how to perform planar cross-projection between views and finally the scale ambiguity inherent in planar cross-projection.

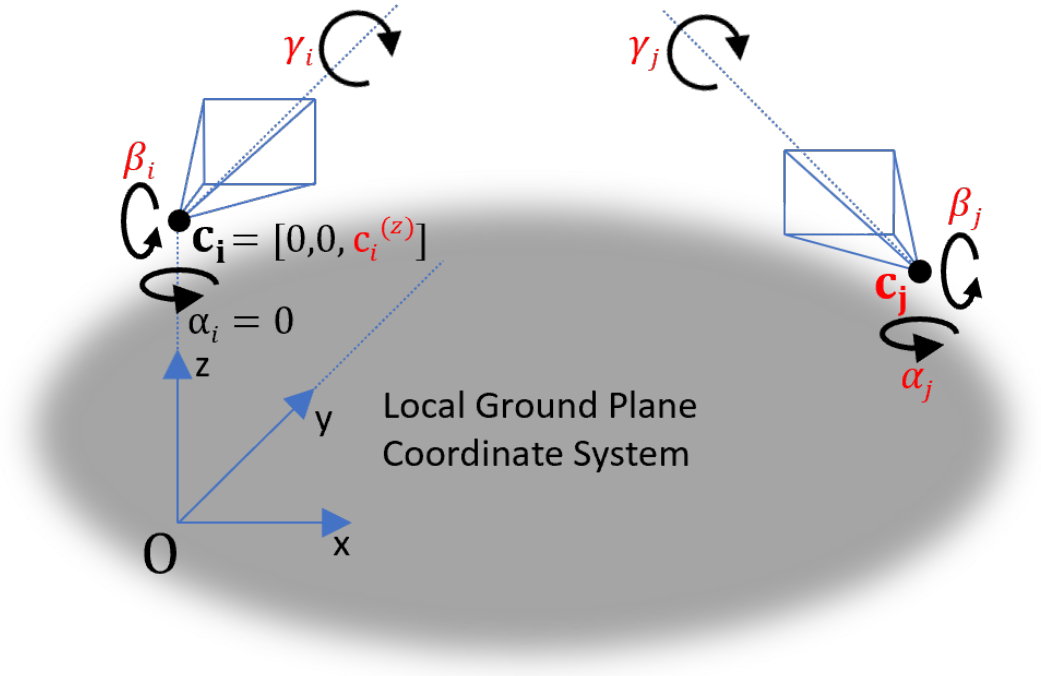

Figure 4.2: Our proposed ground-relative coordinate system. Our Siamese network outputs a 9D representation (shown in red). Specifically, for the input-pair we propose to predict the two camera heights, roll and pitch for camera i, planar position for camera j, and finally roll, pitch and yaw for camera j, all of them relative to an origin defined to be on the road plane directly under camera i.

4.1.1 Parameterisation

We parameterise the 9D ground-relative pose $\theta \in \mathbb{R}^9$ (see Fig. 4.2) of cameras i and j as:

$$\theta = (c_i^{(z)}, \gamma_i, \beta_i, c_j^{(x)}, c_j^{(y)}, c_j^{(z)}, \gamma_j, \beta_j, \alpha_j). \tag{4.1}$$

Defining a local coordinate system in which the ground plane coincides with $z = 0$, camera i is centred above the origin and the projection of its optical axis onto the ground plane is aligned with the y-axis. Hence, the DoF of camera i is three: its height above the ground plane ($c_i^{(z)}$), and its roll (γ_i) and pitch (β_i) relative to the local ground plane orientation. Camera j is specified by its absolute orientation in the local ground plane coordinate system, i.e. a position $\mathbf{c}_j = [c_j^{(x)}, c_j^{(y)}, c_j^{(z)}]$ and rotation parameterised by roll, pitch and yaw (γ_j, β_j and α_j).

We use an angular parameterisation for rotation (specifically Tait–Bryan angles)

as it is a natural representation for vehicle motion and leads to simple priors on each parameter. For example, a forward facing camera mounted with its optical axis parallel to the ground has zero mean pitch and roll and they will be normally distributed over unbiased motion sequences. Note: our parameterisation describes the position of the two cameras relative to the local ground plane. It does not mean that the local ground plane coincides with the global $z = 0$ plane, i.e. the gravity direction is not necessarily aligned with the z axis. Hence, we can describe non-planar motion sequences under the assumption that small motions can be approximated by planar motion.

4.1.2 Relative Pose from Parameterisation

We can extract conventional 6 DoF camera-relative pose from our ground-relative pose. This is important later for computing the optimisation for absolute pose trajectories. World-to-camera rotation and translation are formed from camera angles and centres as:

$$\begin{aligned} \mathbf{R}(\gamma, \beta, \alpha) &= \mathbf{R}_z(\gamma)\mathbf{R}_x(\beta)\mathbf{R}_y(\alpha)\mathbf{R}_x(90^\circ) \\ \mathbf{t}(\mathbf{c}, \gamma, \beta, \alpha) &= -\mathbf{R}(\gamma, \beta, \alpha)\mathbf{c} \end{aligned} \tag{4.2}$$

where, the fixed rotation converts from conventional world coordinates (z up) to conventional camera coordinates (z aligned with the optical axis). Combining Eqns. (4.1) with (4.2) we obtain world-to-camera transformations for the two views:

$$\begin{aligned} \mathbf{R}_i &= \mathbf{R}(\gamma_i, \beta_i, 0), \quad \mathbf{t}_i = \mathbf{t}([0, 0, c_i^{(z)}]^\top, \gamma_i, \beta_i, 0) \\ \mathbf{R}_j &= \mathbf{R}(\gamma_j, \beta_j, \alpha_j), \quad \mathbf{t}_j = \mathbf{t}([c_j^{(x)}, c_j^{(y)}, c_j^{(z)}]^\top, \gamma_j, \beta_j, \alpha_j) \end{aligned} \tag{4.3}$$

where we have utilised the fact that we define camera i to be forward facing ($\alpha_i = 0$) and directly above the local coordinate system ($c_i^{(x,y)} = 0$), as depicted in Fig. 4.2. Finally, the camera-relative pose to transform from the coordinate system of camera i to camera j is given by:

$$\mathbf{R}_{i \to j} = \mathbf{R}_j \mathbf{R}_i^\top, \quad \mathbf{t}_{i \to j} = \mathbf{t}_j - \mathbf{R}_{i \to j}\mathbf{t}_i \tag{4.4}$$

4.1.3 Planar Cross-Projection

For self-supervision via an appearance consistency loss, we require to cross-project one image into the perspective of the other. Due to our assumption of a locally planar world, this is particularly simple. The planar components of the scene can

be accurately cross-projected via a homography derived from our ground-relative parameterisation. First, we write the homography that transforms a point on the local $z = 0$ ground-plane to a given camera k as:

$$\mathbf{H}_k(\mathbf{K}_k, \mathbf{R}_k, \mathbf{t}_k) = \mathbf{K}_k[\mathbf{R}_k\mathbf{S}^\top \quad \mathbf{t}_k], \quad \mathbf{S} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \tag{4.5}$$

where $\mathbf{K}_k$ are the given camera intrinsics and $\mathbf{R}_k$, $\mathbf{t}_k$ are computed from Eqn. (4.3). Combining ground-plane to camera homographies, we write the homography that directly transforms a location in image i corresponding to a point on the ground plane to the corresponding position in image j as:

$$\mathbf{H}_{i \to j} = \mathbf{H}_j \mathbf{H}_i^{-1}. \tag{4.6}$$

4.1.4 Scale Ambiguity

Points lying on the local ground-plane are related by an 8 DoF homography between two views. Our ground-relative pose in Eqn. (4.1) has a 9 DoF form. The additional dimension is explained via a scale ambiguity: scaling the ground-relative translations (or equivalently camera centres) does not change the homography between views. Therefore, from the planar correspondences alone, we cannot estimate the ground-relative camera poses at world scale. However, datasets usually include calibration details for average camera height, which we use as a prior to softly constrain scale to the calibrated value on average, thereby resolving the unknown scale ambiguity. This allows our Siamese network to handle small variations in height above the local ground-plane due to accelerating, cornering, bumps in the road etc.

4.2 Learning Ground-Relative Geometry

We use the geometric matching Siamese network by Rocco et al. [190] outlined in Fig. 4.1 to estimate the ground-relative pose between two cameras overlooking the same scene (see Fig. 4.2). In this section we describe our supervision losses and training details.

4.2.1 Priors

We assume that the calibrated height of the camera above the ground plane, $c^{(z)}_{\text{calib}}$, is known. We assume that variation around these values are normally distributed

and encode these priors via the following loss function:

$$L_{\text{priors}} = (c_i^{(z)} - c_{\text{calib}}^{(z)})^2 + (c_j^{(z)} - c_{\text{calib}}^{(z)})^2 + \gamma_i^2 + \gamma_j^2 + \beta_i^2 + \beta_j^2 \tag{4.7}$$

where $c_{i,j}^{(z)}$, $\gamma_{i,j}$ and $\beta_{i,j}$ are the estimated camera height, roll and pitch respectively for each camera pair (i, j). Our motion model assumes that the camera height ($c_{i,j}^{(z)}$) has a mean of $c_{\text{calib}}^{(z)}$ (1.65m for KITTI [66]) and that the camera pitch and roll ($\beta_{i,j}$ and $\gamma_{i,j}$) have a mean of zero, respectively. These are *soft* priors introduced to help constrain our model to the correct scale and solution space. Our Siamese network will tolerate deviation from these priors in order to reduce our perceptual loss.

4.2.2 Perceptual Loss

At training-time, essentially we re-estimate the road plane for every image pair, rather than maintaining an estimate of ground plane location throughout an entire odometry sequence. When computing our appearance loss, for each pair we assume approximately the road surface is *locally* planar. Assuming local planarity means we can still handle scenes with changes in gradient, and effectively we are approximating a curved surface by a series of planar patches.

Perceptual loss as originally proposed by Johnson et al. [94] uses both a feature and style loss to penalise differences between two images in feature space as opposed to pixel space (see Section 2.2.6). In our approach, we measure appearance loss between one image in an overlapping pair and a cross-projected version of the other input image using only the feature side of the perceptual loss [94].

To form the cross-projected image, we follow the same approach for performing differentiable cross-projection as in Spatial Transformer Networks [90] (see Fig. 4.3). Specifically, we use their method of performing backwards warping via differentiable bilinear sampling as follows. To warp image $\mathbf{I}_i$ from camera i into the perspective of camera j to form an image $\mathbf{I}_{i\to j}$, we first compute the reverse homography $\mathbf{H}_{j\to i}$ using Eqns. (4.5) and (4.6) from the pose parameters of Eqn. (4.1) estimated by the Siamese network. We apply this to every coordinate in a regular grid of size $H \times W$ whose homogeneous coordinates are stacked to form matrix $\mathbf{X} \in \mathbf{R}^{3\times HW}$. Finally, we sample image $\mathbf{I}_i$ at the warped coordinates using differentiable bilinear sampling as $\mathbf{I}_{i\to j} = \text{sample}(\mathbf{I}_i, \mathbf{H}_{j\to i}\mathbf{X})$, where $\text{sample}(\mathbf{I}, \mathbf{X})$ performs differentiable bilinear sampling of image $\mathbf{I}$ at locations $\mathbf{X}$.

To compute the perceptual loss we take the symmetric L2 loss between one input image and the cross-projection of the second input image. For improved

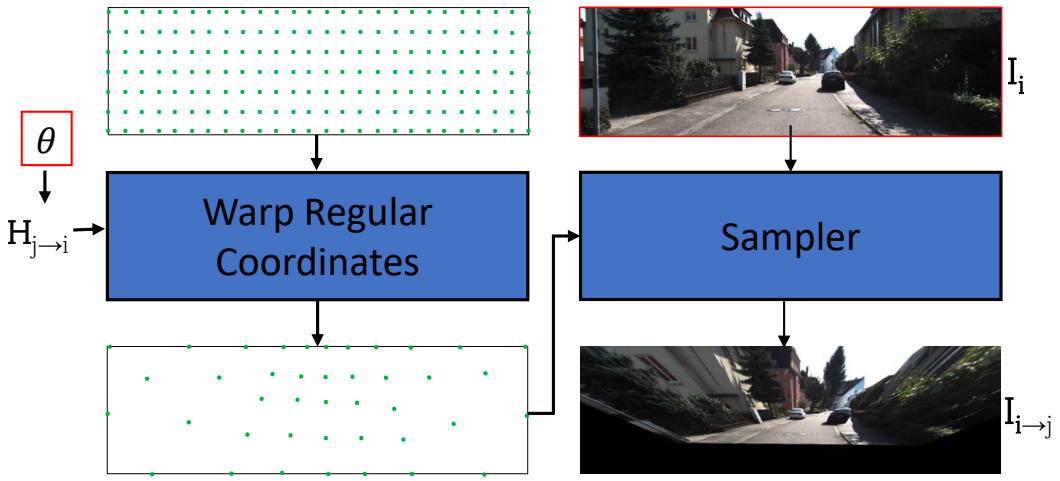

Figure 4.3: Differentiable cross-projection via homography-based backwards warping is made possible by the planar geometry of the local road plane. Ground-relative pose output by our Siamese network θ is used to compute a homography for transforming a regular grid of points, which is used to sample input. Red boundaries indicate the input to the warp function in Fig 4.1

convergence, we sum over 2 scales :

$$L_{\text{pe}} = \sum_{s=1}^{2} \left\| \text{VGG}(\text{ds}(\mathbf{I}_j, s)) - \text{VGG}(\text{ds}(\mathbf{I}_{i\to j}, s)) \right\|_2 \frac{1}{M_j^s} + \left\| \text{VGG}(\text{ds}(\mathbf{I}_i, s)) - \text{VGG}(\text{ds}(\mathbf{I}_{j\to i}, s)) \right\|_2 \frac{1}{M_i^s} \tag{4.8}$$

where VGG denotes extraction of the first seven convolutional layers of an ImageNet [195] pre-trained VGG-16 [202], $\text{ds}(\mathbf{I}, s)$ denotes differentiable downsampling of $\mathbf{I}$ by a factor s and $M_{i,j}^s$ is the number of pixels present in the cross-projected image (for scale s) that are within warped coordinates (e.g. the non-black region of $\mathbf{I}_{i\to j}$ in Fig. 4.3).

4.2.3 Training Details

We train on the weighted sum of the perceptual and prior losses:

$$L_{\text{total}} = w_1 L_{pe} + w_2 L_{\text{priors}}, \tag{4.9}$$

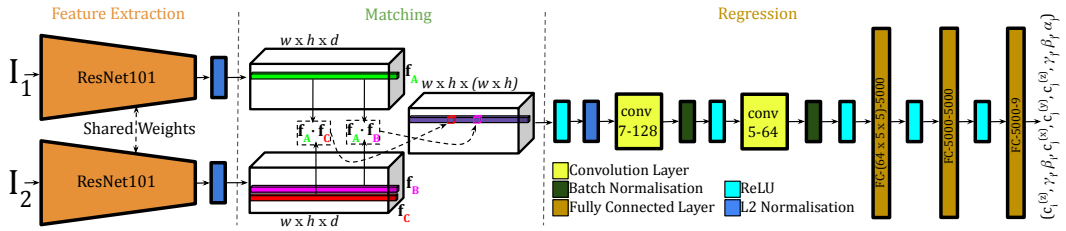

Figure 4.4: Detailed view of the Rocco et al. [190] architecture from Fig. 4.1. Feature maps are extracted separately from images I_1 and I_2. A matching layer combines these to form a correlation volume representing potential matches between both images. A sequence of convolutions and fully connected layers learn to regress our ground-relative pose from the correlation volume. See Section 4.2.4 for a detailed description.

where $w_1 = 1$ and $w_2 = 287000$. These weights were defined as such in order to match the scale of L_{priors} with that of L_{pe} on average. By increasing the scale of the prior term L_{priors} to be in proportion with L_{pe}, we found that the Siamese network would learn the prior values for camera height, roll and pitch more readily than without any weighting. Freezing weights of the pre-trained VGG-16 [202] CNN, we use default SGD with a learning rate of 10^{-4} and use a batch size of 16. The feature extraction and regression network components are initialised with ImageNet and default weights respectively. In Eqn. 4.8 we use $120^2, 240^2$ for $s = 1, 2$ image scale respectively and 240^2 for Siamese network input. Note that each image in a pair is fed individually to the Siamese feature extraction backbone and fusion only happens in the correlation volume. We do not concatenate image channel-wise as in most pose estimators.

4.2.4 Network Architecture

To estimate ground-relative pose from a pair of images both overlooking the same scene, we use a geometric matching architecture by Rocco et al. [190] that mimics the classical steps of feature extraction, matching, pruning and model-fitting. To the best of our knowledge, we are the first to use this architecture for the task of 3D relative pose estimation, as opposed to 2D warping based transformation estimation (see Section 2.2.5). We illustrate the high level and detailed structure of this architecture in Figs. 4.1 and 4.4 respectively. The architecture is composed of three parts: feature extraction, feature matching and regression.

For the Siamese feature extraction component (see the left part of Fig. 4.4) we chose to use ResNet101 [78] (without the final residual block, average pooling and linear layers) to compute a feature map for each of the two input images individually.

This differs from the original Rocco et al. [190] model which uses VGG-16 [202] cropped at the *pool4* layer. We note that this is a Siamese architecture where both images are fed separately into the same feature extractor which allows us to learn shared features across both input images, and where fusion only occurs in the subsequent matching component [101, 113, 190]. Finally, Siamese feature extraction is followed by L2 normalisation over the feature channels, in preparation for input to the matching layer.

The matching component of the architecture (see the middle part of Fig. 4.4) involves a correlation operation between both feature maps from the preceding Siamese feature extraction component and contains no trainable weights. Specifically, as described by Rocco et al. [190], this layer computes a correlation volume c_{AB} as:

$$c_{AB}(i,j,k) = \mathbf{f_A}(i,j) \cdot \mathbf{f_B}(i_k, j_k) \tag{4.10}$$

where $\mathbf{f_A}$ and $\mathbf{f_B}$ are individual feature vectors in the two input feature maps, (i,j) and (i_k, j_k) are positional indices for the $w \times h \times d$ feature maps, and $k = h(j_k - 1) + i_k$ [190]. Each spatially reduced feature vector of the correlation volume represents a vector of similarity scores (the purple vector of the correlation volume in Fig. 4.4), which is a measure of how similar a spatial neighbourhood of the first input image (in feature space) is to every other spatial neighbourhood for the second input image. The resulting correlation volume represents a spatial mapping of potential feature matches between both of the input images [190], from which we can learn to regress relative pose. The correlation volume is followed by ReLU and L2 normalisation (depicted in Fig. 4.4 by the turquoise and blue blocks respectively), which has the effect of highlighting potential matches and softening ambiguous ones [190].

The purpose of the subsequent regression component (see the right part of Fig. 4.4) is to regress pose based on the tentative similarity scores of the correlation volume. As illustrated in Fig. 4.4, the initial regression stage is composed of two sequential 2D convolutional layers, both followed with batch normalisation and ReLU layers (illustrated in Fig. 4.4 by the yellow, green and turquoise blocks respectively). As with the archtitecture in Rocco et al. [190], we note that the convolutional layers are without padding and have a stride of one. Similarly, the first and second convolutional layers have 128 and 64 kernels with a size of 7 and 5 respectively. Rocco et al. [190] only use one fully connected layer to regress a final transformation, but we chose to use three fully connected layers (with ReLU activation for the first two) with a size of 5000 for the internal input/output feature dimensions to compute our 9 dimensional ground-relative pose. The general idea for the regression component of the architecture is to emulate traditional approaches of

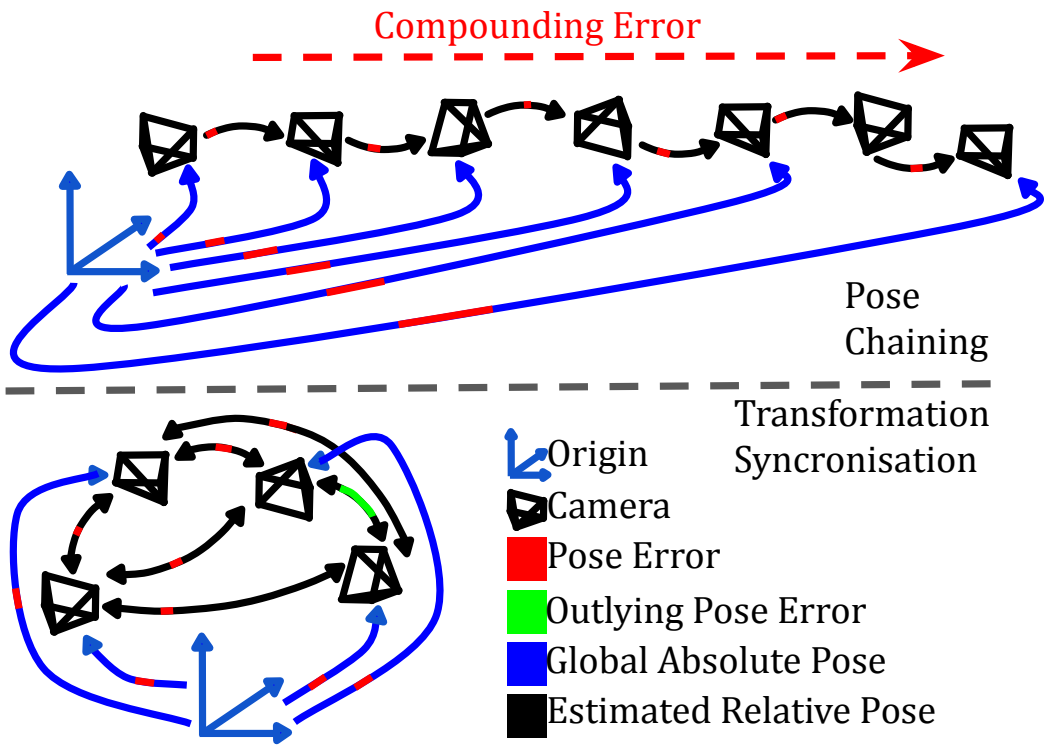

Figure 4.5: Estimating absolute poses from camera-relative poses. Top: Chaining poses results in compounding error (red). Bottom: Transformation syncronisation can optimise a set of camera-relative poses such that the absolute pose error is minimised. Further, the method we use [6] is robust to outliers in our relative pose estimates (green).

pruning potential feature correspondences in a local neighbourhood [190, 197, 204] and robustly fitting a geometric transformation (e.g. RANSAC [61]).

4.3 Transformation Synchronisation for Visual Odometry

So far we have shown how we train for estimating relative pose between image pairs. In order to perform visual odometry, we must compute the absolute poses for each sequence of cameras in order to compute a trajectory in world coordinates. In Fig. 4.5 we illustrate the general idea behind two methods for computing these absolute poses from a collection of relative poses. The most obvious way to convert relative pose estimates to a world-space trajectory is to apply relative pose estimation

between consecutive frames and chain together the relative poses (see Fig. 4.5 (top)). However, this is susceptible to compounding error and is not robust to individual estimates with high error, thus it will lead to highly inaccurate results.

Instead, we exploit the fact that our Siamese network can compute relative pose between arbitrary pairs (see the relative pose output of our Siamese network in Fig. 4.1) and apply it to pairs with multiple different frame offsets. Specifically, each image in a sequence is compared with the following five images temporally. This provides redundancy in that we have multiple ways of combining transformations in order to estimate the relative pose between image pairs. The task of finding the absolute poses that best fit these redundant sets of poses is known as transformation synchronisation which is far more accurate than chaining, and can be adapted to handle large outlying errors in our relative pose estimates (see Fig. 4.5 (bottom)). Note the pairwise planarity assumption we use for training no longer applies here for estimating a global trajectory, hence we can handle non-planar trajectories.

In our context, transformation synchronisation is the optimisation of a collection of camera-relative poses into a single frame of reference, or absolute pose. Specifically we use the method by Arrigoni et al. [6] for spectral motion synchronisation, which we only summarise briefly here. From a high-level perspective we take our Siamese network output to form camera-relative poses for n cameras as:

$$\mathbf{X} = \begin{pmatrix} \mathbf{I}_4 & \mathbf{M}_{12} & \dots & \mathbf{M}_{1n} \\ \mathbf{M}_{21} & \mathbf{I}_4 & \dots & \mathbf{M}_{2n} \\ \dots & \dots & \dots & \dots \\ \mathbf{M}_{n1} & \mathbf{M}_{n2} & \dots & \mathbf{I}_4 \end{pmatrix}, \text{ where } \mathbf{M}_{ij} = \begin{pmatrix} \mathbf{R}_{ij} & \mathbf{t}_{ij} \\ \mathbf{0} & 1 \end{pmatrix} \tag{4.11}$$

where M_{ij} is the camera relative pose between frames i and j in a video sequence of length n. For our toy example in Fig. 4.5 (bottom) we have 4 cameras and therefore $\mathbf{X}$ would consist of a 16 by 16 matrix of camera-relative poses. In the case of our training regime where we are inputting pairs of images into the CNN, we have chosen to separate both images by one to five frames. This was chosen so that there would always be scene-overlap even on video with faster motion, while providing a spread of relative pose variations in the image pairs. As a result of this choice, we can populate the matrix $\mathbf{X}$ with one to five blocks of M_{ij} above and below the main diagonal. As we do not estimate camera relative poses for frames beyond this range, the rest of the entries in $\mathbf{X}$ are missing and defined to be zero. In the case where we have missing relative poses in $\mathbf{X}$ we have that:

$$\mathbf{L} = ((\mathbf{D} - \mathbf{A}) \otimes 1_{4\times 4}) \circ \mathbf{X} \tag{4.12}$$

where $\mathbf{A}$ and $\mathbf{D}$ is the degree and adjacency matrix of $\mathbf{X}$, $1_{4\times4}$ is a matrix composed of ones, and $\otimes$ and $\circ$ denote the Kronecker and Hadamard products respectively [6]. Furthermore $\mathbf{U}$ is defined as a basis for null-space $null(\mathbf{L})$ [6].

Optimal absolute poses are found by solving:

$$\min_{\mathbf{U}^T\mathbf{U}=n\mathbf{I}_4} \left\| \widehat{\mathbf{L}}\mathbf{U} \right\|_F^2, \tag{4.13}$$

where F refers to the Frobenius norm and m are the eigenvalues for corresponding eigenvectors of $\mathbf{X}$. This amounts to solving $\widehat{\mathbf{L}}\mathbf{U} = \mathbf{0}$ in least-squares and furthermore an Iteratively Reweighted Least Squares method is utilised to deal with outliers as described by Arrigoni et al. [6]. We write the resulting absolute poses as:

$$\widehat{\mathbf{R}}_i = [\widehat{\mathbf{R}}_1, \widehat{\mathbf{R}}_2, ..., \widehat{\mathbf{R}}_n] \tag{4.14}$$

$$\widehat{\mathbf{t}}_i = [\widehat{\mathbf{t}}_1^T, \widehat{\mathbf{t}}_2^T, ..., \widehat{\mathbf{t}}_n^T]. \tag{4.15}$$

4.4 Experiments

In this section we present our experimental evaluation. Firstly, we provide some experimental details relating to the data and training process. Secondly, we convey our quantitative and qualitative results. Finally, we illustrate results for our visual odometry trajectories and path length.

4.4.1 Experimental Details

We evaluate our pipeline using the KITTI visual odometry dataset [66]. In general we chose to train on the raw dataset but omitting visual odometry sequences 09 and 10 which are used for testing. Note sequence 03 is not in the raw dataset. For trajectories 11, 12 and 14 in Fig. 4.10 we trained with sequences 09 and 10.

Training pairs are shuffled over all sequences. For each sequence, our training and testing pairs consisted of target $\mathbf{I}_t$ and source $\mathbf{I}_s$ images which are separated by zero to four adjacent frames. Using the self-supervised loss and priors outlined in Section 4.2 we trained models consisting of approximately 65 million parameters to estimate camera-relative pose. Having obtained relative poses on test sequences we use the transformation syncronisation outlined in Section 4.3 to obtain absolute poses $\widehat{\mathbf{R}}_i$ and $\widehat{\mathbf{t}}_i$ for visual odometry evaluation. As our proposed method does not rely on any direct supervision we focus our comparison on leading methods which are fully self-supervised and only rely on a single camera.

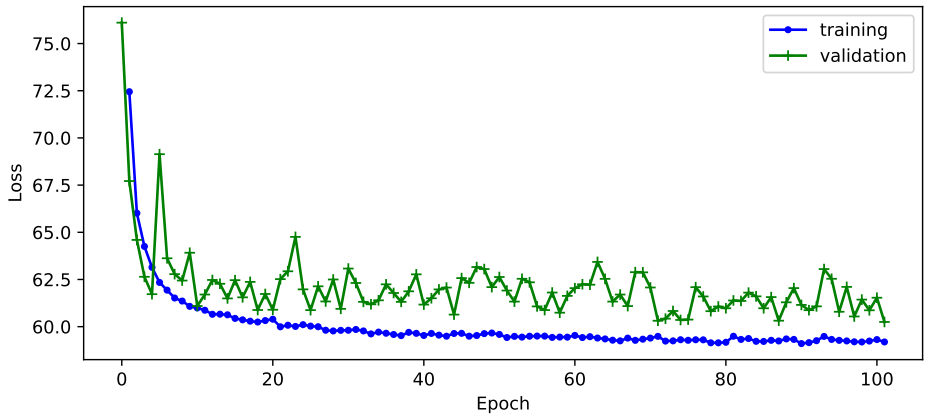

Figure 4.6: Appearance loss versus training epoch for training and validation data.

Convergence In Fig. 4.6 we show a typical loss curve (specifically the left-hand side appearance loss) with the training and validation datasets to illustrate that our model converged. We observe that the validation set loss values are significantly less smooth relative to the training set loss. We suspect this is because the validation set is much smaller than the training set, which introduces noise into the training convergence. It is clear that most of the gain occurs over the first 40 epochs and hence we stopped training at approximately 100 epochs, where generalisation error converges.

4.4.2 Quantitative Results

In Table 4.1 we provide visual odometry scores on sequences 09 and 10 of the KITTI visual odometry dataset [66]. It is standard practice to focus evaluation on these two sequences within the visual odometry community [13, 68, 187, 257, 276, 282, 287]. We use the KITTI benchmark translation and rotation error as metrics, which is measured as an average positional or rotational error over all possible subsequences of length (100, 200, ..., 800 metres) in units of % and $^\circ/m$ respectively (see Geiger et al. [66]). Using these metrics is standard practice in the visual odometry community and they originate from the KITTI benchmark paper by Geiger et al. [66], where they are defined formally as:

$$r_{err}(\mathcal{F}) = \frac{1}{\mathcal{F}} \sum_{(i,j)\in\mathcal{F}} \angle[(\hat{p}_j \ominus \hat{p}_i) \ominus (p_j \ominus p_i)] \tag{4.16}$$

Method	Depth	6DoF	S.Inp	Net+	Staged	Seq. 9			Seq. 10		
						t_{err}	r_{err}	ATE	t_{err}	r_{err}	ATE
LTMVO [287]	✓	✓	✓	✓	✓	**3.49**	*0.010*	**11.30**	*5.81*	*0.018*	**11.80**
TBG [276]	✓	✓	✗	✓	✓	6.93	**0.004**	-	**4.66**	**0.006**	-
CC [187]	✓	✓	✓	✓	✗	*6.92*	0.018	29.0	7.97	0.031	13.77
GeoNet [257]	✓	✓	✓	✓	✗	28.72	0.098	158.4	23.90	0.090	43.04
SfM [282]	✓	✓	✓	✓	✗	8.28	0.031	*24.31*	12.20	0.030	20.87
SC-SfM [13]	✓	✓	✗	✗	✗	11.20	0.034	-	10.10	0.050	-
Mono2 [68]	✓	✓	✗	✗	✗	11.47	0.032	55.47	7.73	0.034	20.46
Ours (PLoss fixed priors)	✗	✗	✗	✗	✗	10.42	0.033	29.45	9.55	0.048	*13.73*
Ours (PLoss mono2-net)	✗	✗	✗	✗	✗	16.69	0.058	58.88	16.72	0.071	32.0
Ours (PLoss)	✗	✗	✗	✗	✗	11.30	0.043	28.68	11.66	0.060	16.48

Table 4.1: Visual odometry results on KITTI for our perceptual loss (PLoss). Metrics t_{err} (%), r_{err} (deg/m) and ATE (m) are translation, rotation error over a set of subsequences, and the absolute trajectory RMSE respectively.

$$t_{err}(\mathcal{F}) = \frac{1}{\mathcal{F}} \sum_{(i,j)\in\mathcal{F}} ||(\hat{p}_j \ominus \hat{p}_i) \ominus (p_j \ominus p_i)||_2 \tag{4.17}$$

where $\ominus$ is the inverse compositional operator as given by Kuemmerle et al. [109], $\angle$ represents angle, p and $\hat{p}$ are the ground truth and estimated camera poses respectively, and $\mathcal{F}$ is a set of camera frames (i, j). Additionally, we provide Absolute Trajectory Error (ATE) in units of metres. The ATE measures the standard deviation between the ground truth and estimated trajectory aligned to the ground truth [287].

We show key differences between leading methods which encapsulate various levels of constraint and complexity. From left to right, methods are split between: usage of a dense depth network, estimating only a 6 DoF camera-relative pose, requiring adjacent or sequential input for inference or training, using additional network(s) for dense estimation (e.g. optical flow, explainability mask), and requiring a staged training process. Drawbacks of training for dense networks include requiring to capture features relevant for thousands of output parameters, versus our 9 geometry aware parameters, increasing complexity drastically, and requiring to learn complex scene contents. A drawback of only using 6 DoF pose is that we lose the richer information given by our 9D ground-relative pose, with which we

may calculate 6 DoF pose (see Eqn. 4.4). Requiring adjacent or sequential input constrains the network to learning limited relative poses, whereas our method can handle more arbitrary poses due to the flexibility of the homographic transformation. Lastly, a staged training process increases the complexity of acquiring an accurate solution in practical applications, whereas our method is easy to train.

While LTMVO [287] and TBG [276] perform most accurately, they are more restrictive and complex in their approach , as discussed in Sections 2.2.2 and 2.3. Both of these methods concatenate sequential frames of input images into the feature extractor parts of their deep pipelines and, as discussed in Section 2.2.5, this can restrict the range of relative pose between images, whereas we choose to use a Siamese network for independently extracting features from both images. Furthermore, these methods estimate only a 6 DoF camera-relative pose, whereas we estimate a 9 DoF ground-relative pose, which allows for the flexibility of estimating both the 6 DoF camera-relative pose and inter-image planar homographies. Further still, LTMVO [287] and TBG [276] use a staged training process and our method in Chapter 4 is simply a single stage of training, reducing the complexity of our approach. Moreover, we only train a single neural network with a 9D output whereas all of the competing approaches in Tab. 4.1 train a dense depth network outputting tens of thousands of parameters, increasing the complexity of their approach. Additionally, TBG [276], CC [187], GeoNet [257] train for dense optical flow estimation on top of dense depth estimation, greatly increasing the number of network and output parameters. Lastly, we note that LTMVO [287] use LSTM modules which can be memory and time intensive, easy to overfit and sensitive to various weight initialisations [262].

Our method is competitive with leading self-supervised approaches, while remaining the most flexible and unconstrained. Crucially, our method does not rely on training networks for estimating thousands of parameters for dense maps, simplifying relative pose estimation with a CNN significantly. Additionally, we show our method with the Monodepth2 [68] pose network is notably worse than the matching Siamese network we use. Further, we obtain boosted performance by fixing at test-time the camera heights, pitches, and rolls to their prior values.

4.4.3 Qualitative Results

Visual Perspective Warps We show qualitative results in Figs. 4.7 and 4.8. The first example in Fig. 4.7 shows a case where our model learns a pose which aligns features such as road lines (green ellipse) but fails to align more localised features like the manhole covers (red ellipse); likely due to cases where our method converges to a local minimum. Moreover, ground truth does not perform very well, likely as

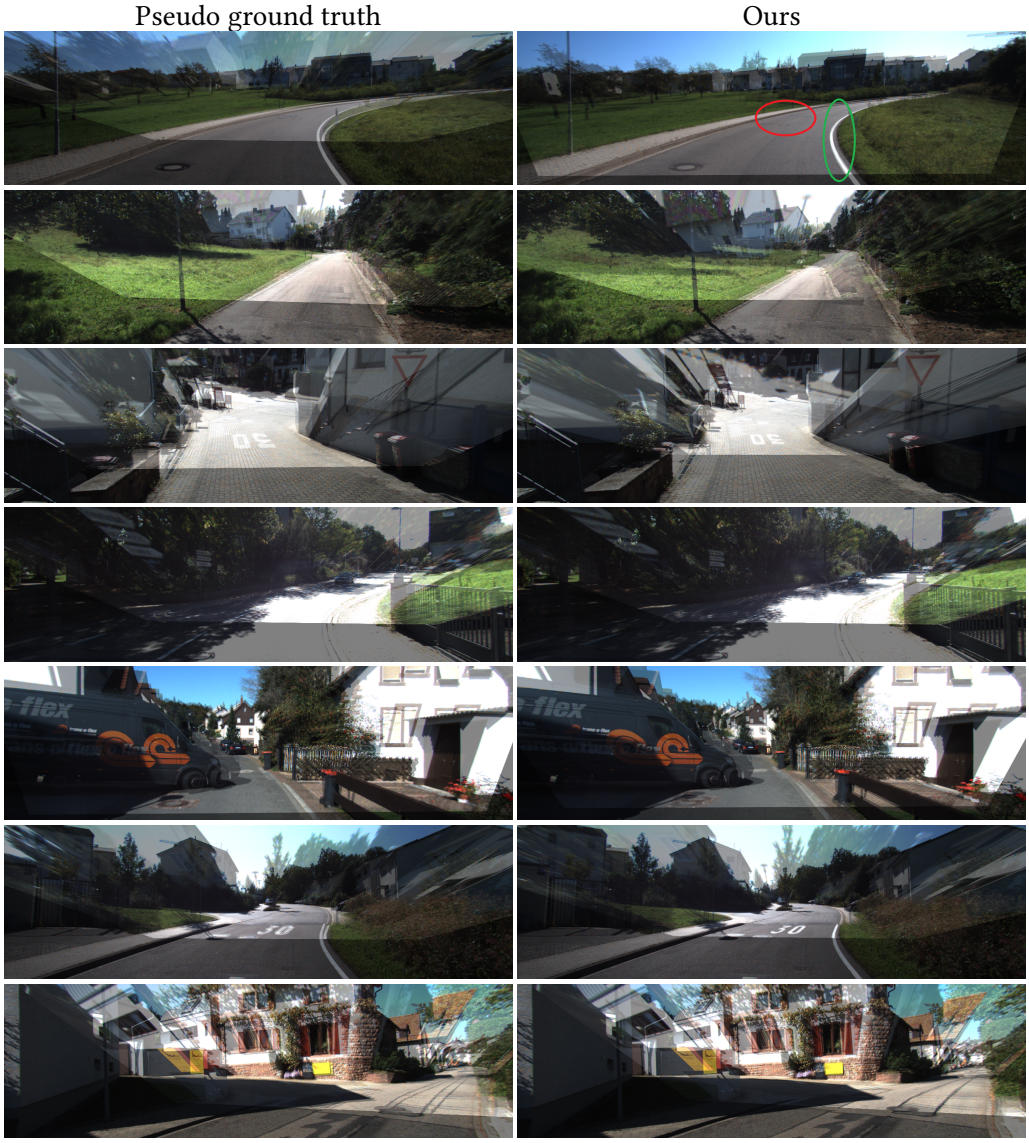

Figure 4.7: KITTI qualitative results where images are a composition of one Siamese network input with its warped counterpart. Left: Pseudo ground truth where we assume fixed prior values for the ground plane. Right: our ground-relative pose result.

Figure 4.8: Further KITTI qualitative results: images are a composition of one Siamese network input with its warped counterpart. Left: Pseudo ground truth where we assume fixed prior values for the ground plane. Right: our ground-relative pose result.

we are using camera-relative ground truth and transforming it to ground-relative using assumed fixed priors for camera height and rotations (which we call pseudo ground truth). The fourth example shows accurate performance in the presence of multiple challenging elements: cornering, gradient change, an oncoming vehicle, shadows and glare. Examples two and three show partial failure modes where the road bends more sharply than usual in the vertical dimension, challenging our planarity assumption.

In the fifth example performance is likely hampered by a reversing vehicle. We attempted filtering out non-road pixels for input to the Siamese network or appearance loss with semantic segmentation, but had difficulty with successful convergence. We suggest that this be investigated further for future work. Subsequent examples show our method outperforming the assumed ground-relative positioning with highly accurate alignment.

To further highlight weaknesses and strengths of our method we provide perspective warps in Fig. 4.9 for test sequences 13, 15 and 16 where absolute ground truth is unavailable. For these we provide predicted perspective warp compositions at two different times where t_n and t_{n+x} denotes image-pair n and then temporally subsequent pair $n+x$ in the motion sequence. The first example shows a case where little road is present in one of the images going around a corner, and yet our method is able to estimate gross relative pose accurately (In Fig. 4.11 this is the second right-hand corner from the top-left, for sequence 13). Second and third examples show where our planarity assumption is challenged by sudden changes in the road gradient. Specifically, the second row example error may be due to the vehicle front and back spanning a slight change in road height, while the third example shows an outlying case where the vehicle transitions between planes sharply. Our method performs well for pairs at either side of these outliers and the overall trajectories appear grossly robust in these regions.

The fourth example shows a fail case perhaps due to three factors: mismatching from very similar repetitive features such as lines, dynamic shadows from trees, and faster speeds. For the last point here we note that our model sometimes performs less well on faster sections of road. Most of the training data is captured for slower urban speeds and thus dataset bias could be an issue. However, visual separation of matching features is likely greater at higher speeds, hence the risk of finding local minima, as described previously, is more pronounced. Subsequent examples show robust pose estimation with cornering, cluttered scenes with narrow roads and dynamic vehicles.

Figure 4.9: Strengths and weaknesses: images are a composition of one Siamese network input with its warped counterpart at for temporally close image pairs n and $n+x$.

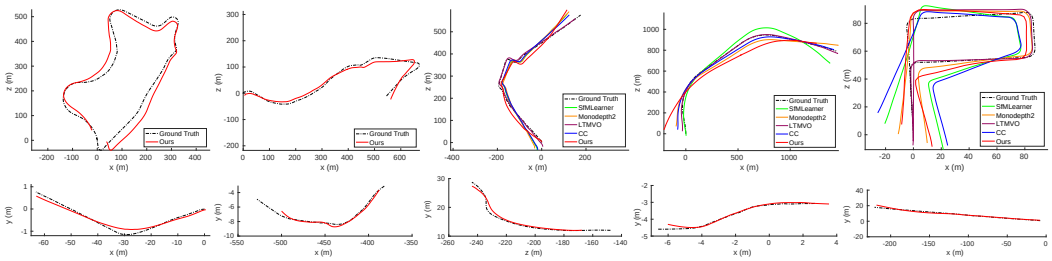

Figure 4.10: Top: Trajectories for sequences 09, 10, 11, 12 and 14 of the KITTI visual odometry dataset. For sequences 11, 12 and 14 we compare against leading fully self-supervised methods. Bottom: We evaluate trajectories along sub-sections of sequences 10, 09 and 03 where the gradients change more rapidl.

4.4.4 Trajectories and Path Length

Fig. 4.10 (top) displays trajectories for visual odometry sequences 09, 10, 11, 12 and 14. For sequences 11, 12 and 14 we compare with leading methods from Table 4.1. While LTMVO [287] generally achieves the best result, our trajectories are visually close to competing methods, with a less complex approach (see Section 4.4.2). Fig. 4.10 (bottom) shows trajectories for vertical cross-sections where gradient change is sharper except for the last plot where we show good performance on a constant uphill gradient. The first two plots relate to the partial failure modes shown in the second and third example of Fig. 4.7. Our planarity assumption is challenged in these sections and while error is higher, we still achieve trajectories close to ground truth. In Fig. 4.7 and 4.10 we see a significant change in gradient does negatively impact results. In practice, guidance[1] for safe construction of roads with adequate camber for drainage seem to be *for the most part* a road will slope smoothly, without exceeding a maximum gradient of 1 in 12. So, these rapid gradient related errors should usually occur as outliers due to the outlying nature of sharply curving road sub-sections. The SE(3) method in Section 4.3 is robust to outliers in Eqn. 4.11, which should reduce the overall impact of these errors on visual odometry. Note we use unequal axes on the bottom row to help visualise the vertical variation.

We note that the trajectories provided in Fig. 4.10 are computed over entire sequences. The error or drift from visual odometry can accumulate significantly over long sequences, even with techniques such as transformation syncronisation (see Section 4.3). Therefore, in Fig. 4.11 we evaluate over smaller sub-sections of length approximately 200 frames each, to observe how our method performs

1 https://www.hse.gov.uk/comah/sragtech/techmeastraffic.htm

more locally. We evaluate over test sequences 03, 09, 10, 11, 12, 13, 14 and 15 respectively. The full 3D ground truth poses in Fig. 4.11 for sequences 03, 09 and 10 are provided by the KITTI dataset [66]. We note that the full 3D ground truth poses for sequences 11, 12, 13, 14 and 15 are not provided directly as a part of the KITTI dataset [66] as these are official test set sequences where ground truth is withheld. As a compromise, in Fig. 4.11 we used the 2D ground truth poses (i.e. the aerial view trajectory) from the KITTI visual odometry benchmark results page [102], which lacks the vertical component, and hence is somewhat pseudo ground truth. For the results in Fig. 4.11 we align our estimated trajectories to the ground truth using a Procrustes fitting function [144]. For the 2D pseudo ground truth the trajectories are essentially a projection of the real 3D ground truth unto a plane, and therefore aligning our estimated 3D trajectories unto these pseudo ground truth trajectories is an approximation. However, as the vertical versus horizontal variation in trajectory is generally small, the trajectories of small sub-sections as illustrated in Fig. 4.11 for sequences 11, 12, 13, 14 and 15 will appear similar to those with full ground truth as seen for sequences 03, 09 and 10. Aside from errors present with sequence 12, we note that our method performs visually very well on smaller sub-sections of these sequences.

In Fig. 4.12 we show how translation and rotation errors vary with trajectory path length on sequences 09 and 10, with and without fixing priors at test-time for parameters with priors. Interestingly we observe that the translation error on sequence 10 departs from the trend while fixing priors.

4.5 Conclusions

road scene visual odometry can be at least approximately solved by self-supervising a *single* pose network via modelling the road surface as a series of planar patches.

Parameterising network output as two cameras posed around a locally planar patch is a useful representation: We proposed learning a novel ground-relative parameterisation for relative pose where local planarity is leveraged to allow cross-projection via a homography to form a self-supervisory training signal. It is shown to be entirely possible to leverage known geometry to train a neural network to perform at least coarsely with the KITTI road scene dataset.

At this stage it should be clear that we now have a method which can replace the usual solution of training for *10s of thousands* of parameters for dense depth (dramatically simplifying training), removing the need to implicitly learn complex scenes by explicitly providing a known geometric model.

An architecture useful for generic geometric transformations was found to be significantly helpful: We showed that the geometric matching network by Rocco et al. [190] can be applied to the task of 3D pose estimation. We found this architecture to be particularly useful compared to more generic architectures such as that used by Monodepth2 [68].

Perceptual loss can be leveraged for successfully training a self-supervised relative pose network: The perceptual loss, as popularised by Johnson et al. [94] for Style Transfer, proved very effective for converging our network training to a viable solution. As far as we know, all other self-supervised relative pose approaches use pixel-level appearance losses (see Section 2.2.2). These generally require regularization, such as a smoothing term, to allow for successful training convergence [68, 287]. In our case we find that a two-scale symmetric perceptual loss was able to train a Siamese network (see Fig. 4.4), only initialised with simple ImageNet [195] weights, to comparable performance with leading approaches (see Tab. 4.1).

A ground-relative parameterisation is useful : We have demonstrated a parameterisation which is more powerful than most 6 DoF camera-relative pose formulations, as our ground-relative pose is more general. From our ground-relative formulation we may compute 6 DoF camera-relative poses, for tasks such as visual odometry and we will see in Chapter 5 it is useful for further training to gain a more accurate model. Further we may use ground-relative pose to compute homographies between image and ground planes, for supervision purposes, or for other tasks, such as mosaicing for mapping applications.

Limitations: In many places along the KITTI driving sequences we observe a lack of performance, perhaps where we become stuck in local minima due to features such as road lines. In our approach towards using transformation syncronisation for visual odometry, trajectory accuracy can be severely hampered if a particular stretch of road is particularly lacking in relative pose accuracy. This can especially be the case in places where we have illumination issues or more unusual motion which is not commonly observed in the rest of the training data.

In Chapter 3, the utility of semantic knowledge was shown to be useful for the task of semantic segmentation. To help address some of the performance issues we observe with our appearance loss for relative pose estimation, we now propose to combine semantics with a new refinement-stage training and inference step. In particular, we take inspiration from Kolotouros et al. [105] to use *model-fitting in-the-loop* for a more direct and classical approach with the estimation of homographies, and use a semantic segmentation model to help filter out noisy scene content and isolate the geometry most relevant for our geometric model representation.

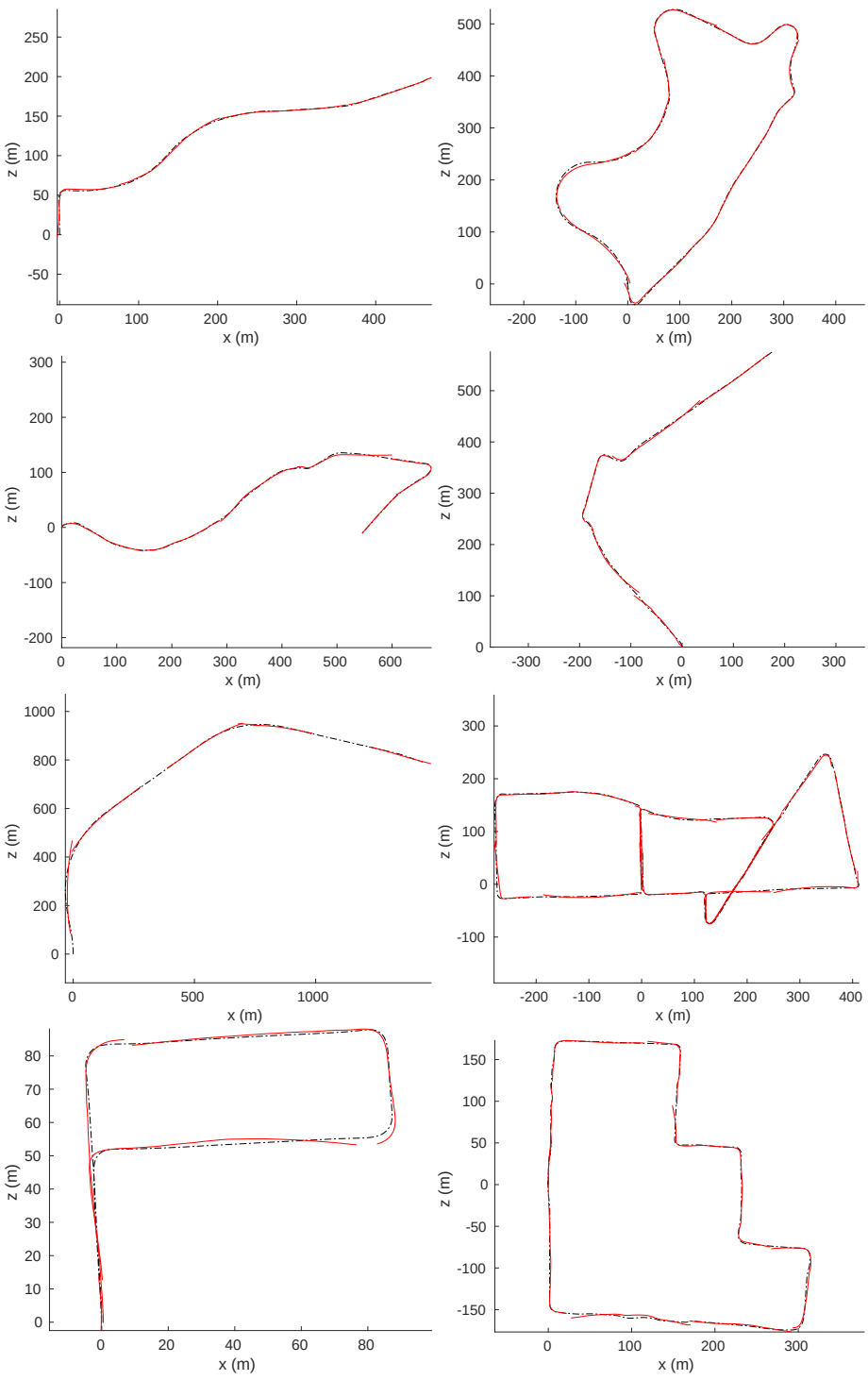

Figure 4.11: Localised trajectory evaluation for testing sequences 03, 09, 10, 11, 12, 13, 14 and 15 respectively on the KITTI visual odometry dataset [66]. Each sub-sequence is roughly 200 frames in length.

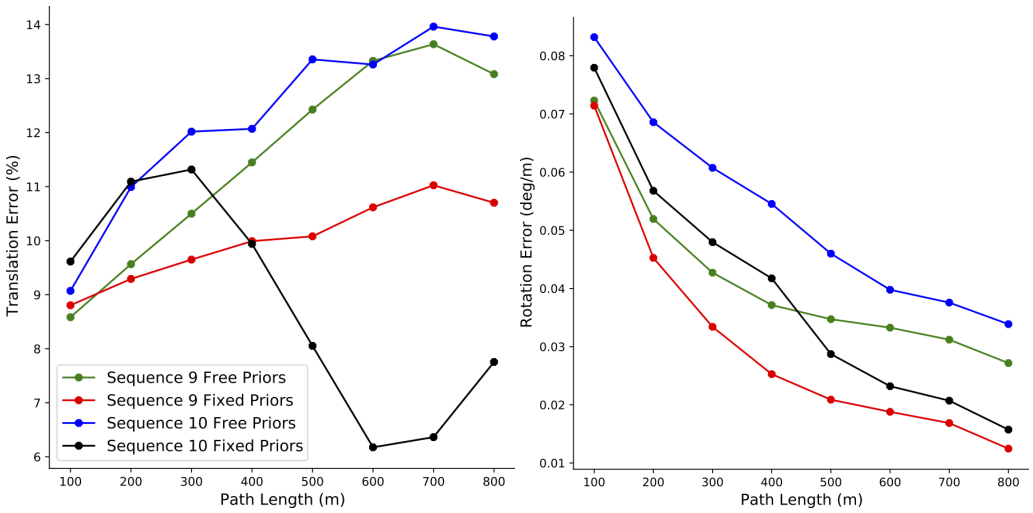

Figure 4.12: Translation and rotation errors by path length on sequences 09 and 10. We compare errors with and without fixing parameters with known priors.

5 Geometry and Pose Estimation with Homographic Model-fitting

While a classical problem in vision, those visual odometry methods based on local feature extraction and matching can be fragile (e.g. failing for larger ego-motion or purely rotational motion), fail for textureless scenes and slow [18, 162, 163, 276]. Deep learning methods have proven themselves to be robust and to provide fast inference, however, these methods are only trained to be optimal in aggregate over a training set, and therefore do not necessarily provide the optimal solution for a given image pair [78, 87, 202, 207, 213]. Further, as we have illustrated in Chapter 4, appearance losses can be prone to local minima where, for example, road features such as lane markings could be misaligned along the direction of motion. In contrast, a classical method could exactly align features which are correctly matched, which raises the question of whether we can combine more classically orientated techniques with deep learning approaches.

In Chapter 3 we showed that domain understanding for semantics can be used to improve the accuracy of a semantic segmentation network. In Chapter 4 we switched to the task of motion estimation and, by explicitly enforcing the planar nature of road scenes, we dramatically simplify the task that a CNN must solve, while retaining the benefits of self-supervision.

road scenes are highly regular with much of the motion proceeding in a locally planar fashion. Further, a problem with the appearance loss of Chapter 4 was that much of the scene did not relate to our planar model (e.g. cars, lamposts, sky etc.). Therefore, it was natural to ask whether semantics could be helpful to filter out

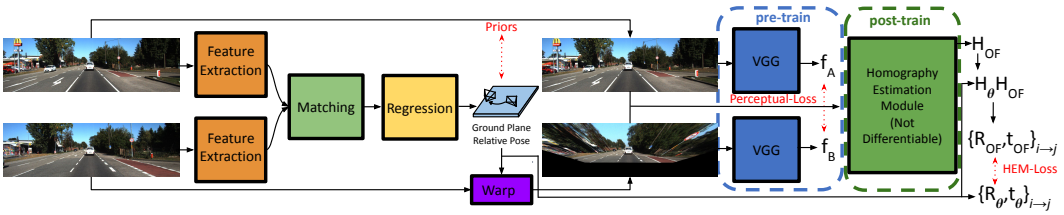

Figure 5.1: We train in two phases: 1. using a perceptual loss based on deep features as described in Chapter 4, 2. via our Homography Estimation Module (HEM) that fits a refined homography, which we decompose to camera-relative pose for supervision and visual odometry.

scene content which did not relate to the base geometry we were using as a prior. In this chapter we combine semantic and geometric scene understanding by using it to isolate the planar geometry of the road. In particular, we propose a method for improving the relative pose estimation of Chapter 4, by fitting and decomposing refined homographies for the camera-to-camera motion model shown in Fig. 4.2.

As illustrated in Fig. 5.1 we use a pre-training phase as described in Chapter 4 with appearance loss. Subsequently in Chapter 5, we describe our Homography Estimation Module (HEM) which is a post-training and inference time refinement model-fitting approach. In contrast to the appearance loss, the refinement loss (HEM-Loss) is a non-differentiable pseudo-label generation technique where we fit a homographic model to road plane correspondences.

Specifically, we make the following contributions:

1. We compute a refinement of the regressed homography from Chapter 4 by applying a non-differentiable optical flow plus RANSAC [61] procedure to regions of the image labelled as ground plane by a semantic segmentation network.

2. Subsequently we show this homography can be decomposed using a known analytical method, and how with knowledge of our motion geometry problem, camera-relative poses can be approximately obtained. We provide a module that generates pseudo-labels and, hence, another source of self-supervision which improves visual odometry performance.

3. At inference time, the model-fitting refinement module and homography decomposition method can be applied on top of network output for improved accuracy for tasks such as visual odometry.

As shown with our appearance loss in the previous chapter, the estimated relative poses from our method can be used for trajectory estimation by applying transformation synchronisation from Section 4.3. Self-supervision provided by our simple geometric model and model-fitting based refinement is highly competitive with state-of-the-art self-supervised methods that require dense depth estimation. To the best of our knowledge we are the first to apply this model-fitting in-the-loop idea to the application of motion estimation and visual odometry specifically.

In Section 5.1 we explain the details behind how we refine the homography from the Siamese network in Chapter 4 by using our Homography Estimation Module. In Section 5.2 we explain how this homography is decomposed into approximate camera-relative pose. In Section 5.3 we explain how these poses are used to refine the training of our pose Siamese network and boost visual odometry performance at inference time. In Section 5.4 we describe our experimental results.

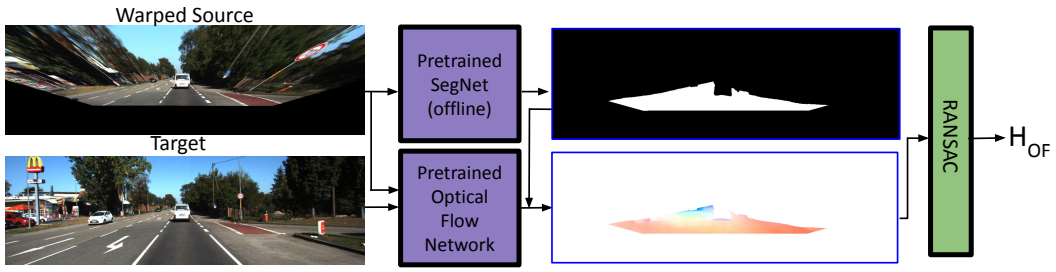

Figure 5.2: Non-Differentiable Homography Estimation Module (see Fig. 5.1): we use a pre-trained optical flow network to estimate point correspondences between one network input and the corresponding input transformed via the ground-relative pose output. A pre-trained semantic segmentation network isolates the road plane points so that RANSAC [61] can be used to robustly estimate the road plane homography.

5.1 Homography Estimation Module

In Section 4.1 we explained our 9D ground-relative parameterisation, and in Section 4.2 we detailed how we use it to form an appearance loss to train our Siamese network. Specifically, we relied on the Siamese network to learn an image to homography function based on a perceptual loss where backward gradients must pass coherently through a bilinear sampler. In Fig. 5.1 this is shown as our pre-training phase in the blue-dashed box. In this section we show that we can extract a homography directly from an image-pair (green-dashed box in Fig. 5.1) and in Section 5.2 we will show how to decompose this to form relative poses useful for training (see HEM-Loss in Fig. 5.1).

5.1.1 Optical Flow

Fig. 5.2 illustrates our approach where we use a direct matching method in a non-differentiable module for estimating a homography between $\mathbf{I}_{i\to j}$ and $\mathbf{I}_j$. To compute $\mathbf{I}_{i\to j}$ we form a homography $\mathbf{H}_\theta$ from the Siamese network output θ by using Eqn. 4.6, which is used to warp a source image for camera i into the perspective of its corresponding target image for camera j.

Therefore, given our initial training from Chapter 4, $\mathbf{I}_{i\to j}$ and $\mathbf{I}_j$ are aligned sufficiently for our refinement step to perform accurately, and additionally concatenation of these images for input into a convolutional network for optical flow is sensible. For simplicity we chose to use a pre-trained optical flow network (FlowNet2 [89]) to estimate the flow between $\mathbf{I}_j$ and $\mathbf{I}_{i\to j}$, but it is worth noting that other methods for feature matching could be employed. We compute pixel

destination points $\mathbf{P}_d$ from a regular grid of source points $\mathbf{P}_s$ as:

$$\mathbf{P}_d = \mathbf{P}_s + OF(\mathbf{I}_{i\to j}, \mathbf{I}_j), \tag{5.1}$$

where OF denotes inference with FlowNet2 [89].

Optical Flow Performance In Figs. 5.3, 5.4 and 5.5 we show the performance of FlowNet2 [89] on the output of our Siamese network. On the left column we illustrate the input $(\mathbf{I}_{i\to j}, \mathbf{I}_j)$ and on the right column we show the optical flow results for the road plane region. We note that the flow in this region is largely coherent. In Fig. 5.3 the final three examples show potential issues where the optical flow is focused on regions with dynamic shadows. Shadows from moving vehicles will not be correctly cross-projected and represents a limitation in our modelling. However, our Siamese network is still able to produce close alignments in the presence of such noise and we suggest that automatic identification of features such as dynamic shadows could be a useful side-affect of our method. Additionally, we note that we generally see more misalignment in the background road plane, which is reflected in many of the optical flow visualisations.

5.1.2 Semantic Segmentation

Many parts of road scenes could contain planar surfaces, for example, sides of structures, billboards, and sides of trucks. For dynamic planes such as sides of trucks, our homography model assumes a static scene between images in a pair, therefore we do not want to consider these points. More generally, as demonstrated in Chapter 4, the road plane itself is the consistent geometric feature with which we want our deep learning model to generalise relative pose, and as such, we do not propose to use other planar surfaces to form training signals. More importantly, we aim to refine $\mathbf{H}_\theta$, which is relevant for the road plane parameterisation utilised in Chapter 4 (see Fig. 4.2). Building on our thesis from Chapter 3 of utilising semantics for scene understanding, we explicitly isolate the road plane by filtering non-road pixels using a pre-trained semantic segmentation network [285]:

$$\mathbf{P}_s^{(road)} = mask_{road}(\mathbf{P}_s), \quad \mathbf{P}_d^{(road)} = mask_{road}(\mathbf{P}_d) \tag{5.2}$$

where $mask_{road}$ denotes filtering out non-road pixels. By doing so, we have now isolated estimated point correspondences between $\mathbf{I}_j$ and $\mathbf{I}_{i\to j}$ for the road plane, and our goal is to use these to compute a transformation between these images (i.e.

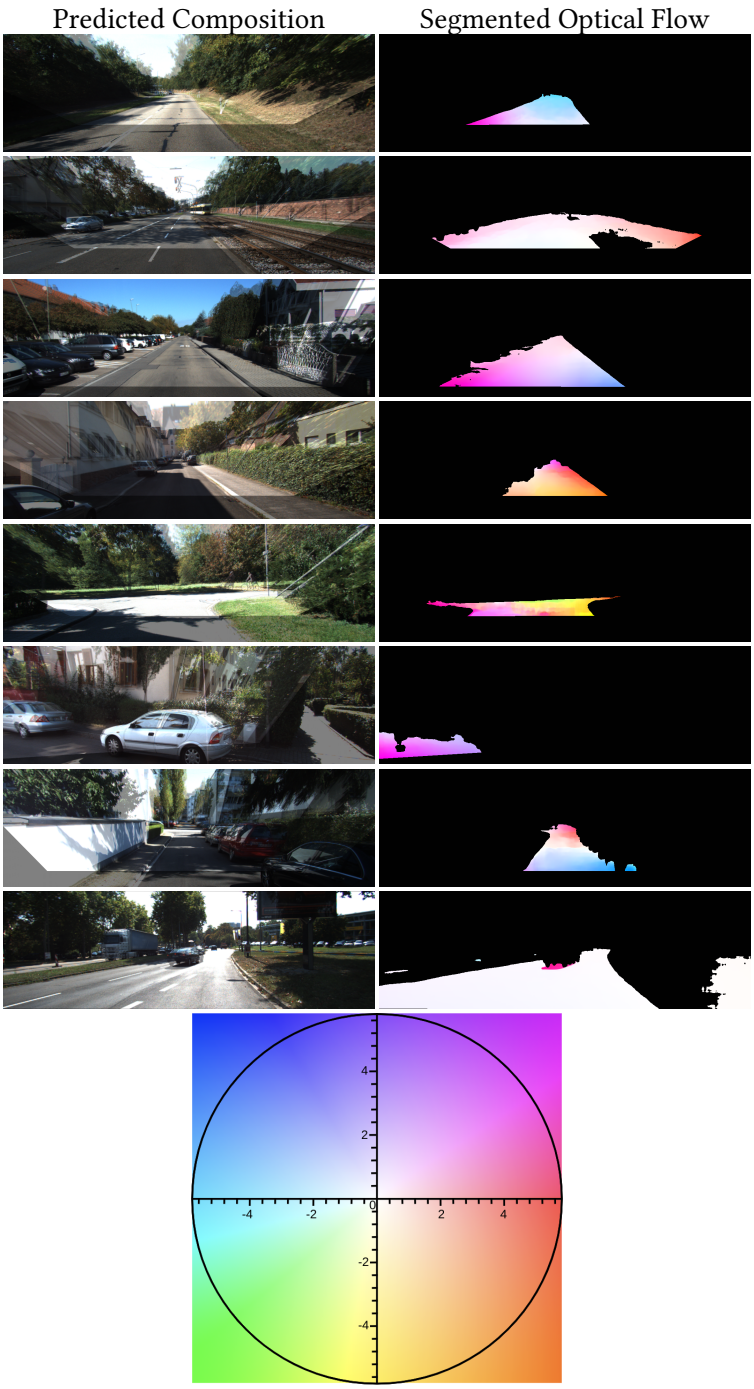

Figure 5.3: Qualitative performance of pre-trained optical flow and segmentation networks. Left: Composition of target image with perspective-warped source image. Right: Optical flow result of stacked target and perspective-warped source images, segmented for road pixels. Note that in the last three examples dynamic shadows are misaligned due to our homographic model being limited to static features on the road surface, which causes optical flow visuals to focus mostly on these features. Bottom: Optical flow key.

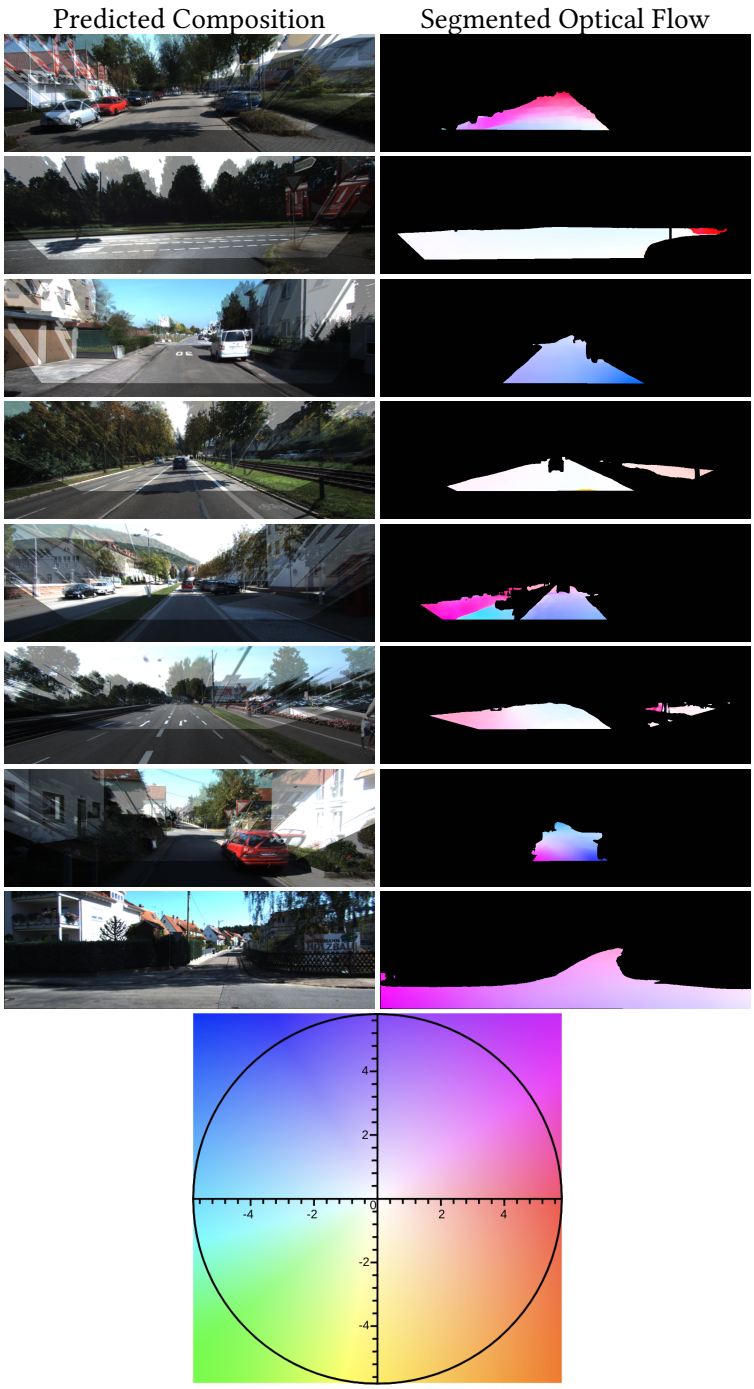

Figure 5.4: Qualitative performance of pre-trained optical flow and segmentation networks. Left: Composition of target image with perspective-warped source image. Right: Optical flow result of stacked target and perspective-warped source images, segmented for road pixels. Bottom: Optical flow key.

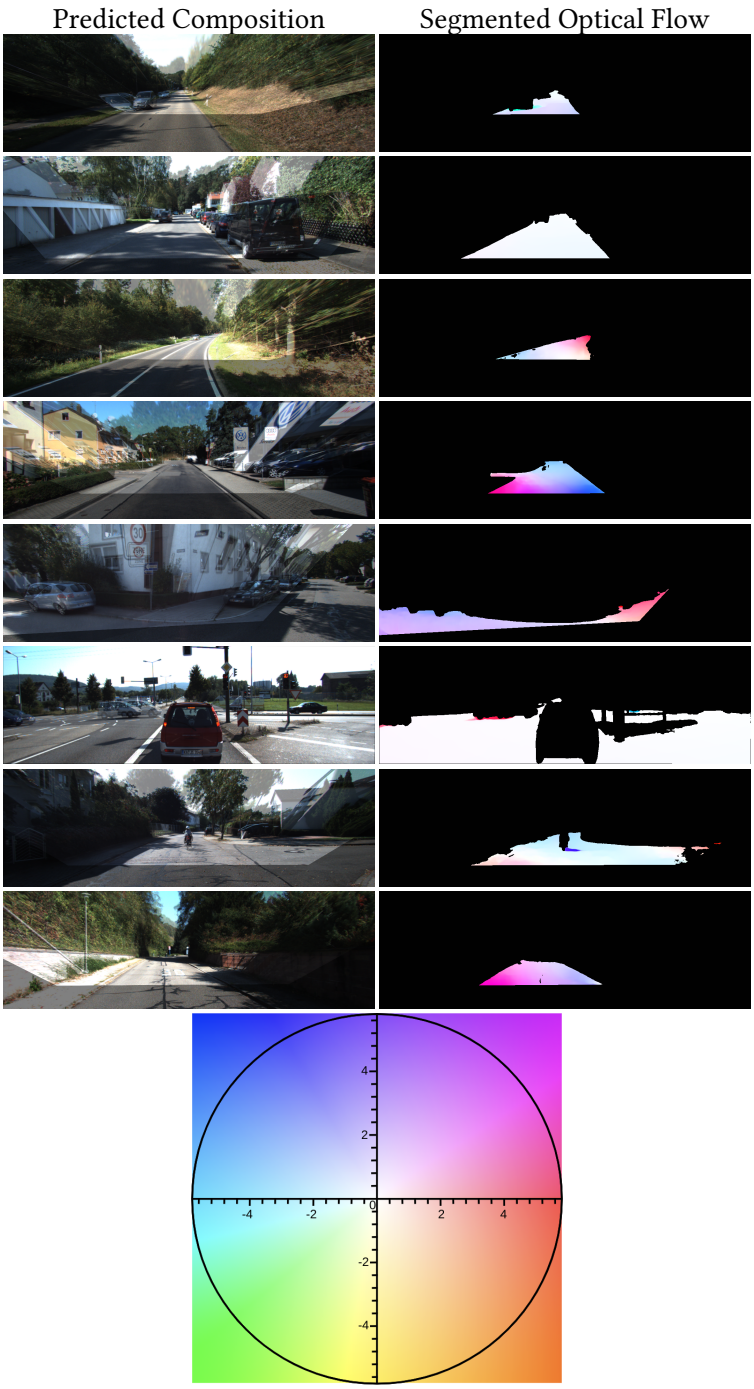

Figure 5.5: Qualitative performance of pre-trained optical flow and segmentation networks. Left: Composition of target image with perspective-warped source image. Right: Optical flow result of stacked target and perspective-warped source images, segmented for road pixels. Bottom: Optical flow key.

a homography in this case), which can be used to update $\mathbf{H}_\theta$ into a more accurate transformation.

Road Segmentation Performance In Figs. 5.3, 5.4 and 5.5, in addition to optical flow, we illustrate the performance of the pre-trained segmentation network for our scenes. These were computed offline for the source and target images and combined with the warped source to obtain the visualised results. The road segmentation is of a high quality, though we occasionally observe some errors. For example, for the second example in Fig. 5.3 we see much of the train tracks have been classed as road, and the last example show a grassy area also mis-classified. However, these regions are generally still mostly planar and hence can still be harnessed positively by our method.

5.1.3 Homography Fitting with RANSAC

Optical flow with road plane semantic segmentation allows for estimating dense correspondences across the whole road plane region. Our goal is to use these correspondences to improve the accuracy of the current parameter estimates by fitting a homography. However, these correspondences may be noisy or may include regions that were incorrectly segmented as belonging to the ground plane. For this reason, we require a robust means to fit a homography to the road plane pixelwise correspondences estimated by optical flow. For this purpose we use RANSAC [61].

A homographic transformation between source points $(x_k^{'}, y_k^{'})$ of $\mathbf{P}_s^{(road)}$ and destination points (x_k, y_k) of $\mathbf{P}_d^{(road)}$ in homogeneous coordinates can be written as [175]:

$$s_k \begin{bmatrix} x_k^{'} \\ y_k^{'} \\ 1 \end{bmatrix} \sim \mathbf{H} \begin{bmatrix} x_k \\ y_k \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x_k \\ y_k \\ 1 \end{bmatrix} \tag{5.3}$$

where k denotes each point correspondence and s_k are scale factors. We use an OpenCV [175] RANSAC [61] implementation to robustly fit a homographic model to the corresponding road plane points in a pair of coarsely aligned training or test images. Specifically, this implementation functions in two steps. Firstly, it estimates an initial homography $\mathbf{H}$ where a simple least-squares method is used to minimise the reprojection error in homogeneous coordinates as:

$$\sum_k \left(x_k^{'} - \frac{h_{11}x_k + h_{12}y_k + h_{13}}{h_{31}x_k + h_{32}y_k + h_{33}} \right)^2 + \left(y_k^{'} - \frac{h_{21}x_k + h_{22}y_k + h_{23}}{h_{31}x_k + h_{32}y_k + h_{33}} \right)^2 \tag{5.4}$$

However, as we have outliers present within our point correspondences we use RANSAC [61] to robustly fit a homography to our set of points k as:

$$\mathbf{H}_{OF,RANSAC} = RANSAC(\mathbf{P}_s^{(road)}, \mathbf{P}_d^{(road)}). \tag{5.5}$$

and the reprojection error in Eqn. (5.4) is used within each randomly sampled subset of correspondences in a standard least-squares routine. Secondly, this homography is refined with only the chosen inliers from RANSAC [61] using the Levenberg-Marquardt method [116, 143] to further decrease the reprojection error of $\mathbf{H}_{OF,RANSAC}$ to finally form $\mathbf{H}_{OF}$ [175].

We represent the homography computed from the estimated ground-relative pose via Eqns. (4.3), (4.5) and (4.6) (and as illustrated in Fig. 5.1) by $\mathbf{H}_\theta$. We update the homography $\mathbf{H}_\theta$ computed from our network output for ground-relative pose as:

$$\mathbf{H}_{i\to j}^{(OF)} = \mathbf{H}_\theta \mathbf{H}_{OF}. \tag{5.6}$$

While the homography itself could be useful for tasks such as mosaicing or image stitching [214], having obtained a refined homography $\mathbf{H}_{i\to j}^{(OF)}$, we ask the question of whether we can decompose it back into camera-relative pose for the purpose of training our Siamese network and as input to transformation synchronisation (see Section 4.3) for visual odometry.

5.2 Homographic Decomposition

Our refined homography $\mathbf{H}_{i\to j}^{(OF)}$ is a transformation which encapsulates information relevant to the relative motion between two cameras overlooking a planar surface [142]. In this section we explain how we decompose this homography into camera-relative pose and subsequently use it.

5.2.1 Choosing Between Four Solutions

While it would be possible to compute a loss between $\mathbf{H}_\theta$ and $\mathbf{H}_{i\to j}^{(OF)}$ in order to provide a self-supervision signal to the Siamese network, our experience is that it is ineffective. Specifically, we found that the loss function given by:

$$\mathbf{L}_\mathbf{H} = \|\mathbf{H}_\theta - \mathbf{H}_{i\to j}^{(OF)}\|_2 \tag{5.7}$$

failed to converge while training. As the scale of each homography is arbitrary, we suspect that this could be part of the reason of why training with this loss function

fails to consistently reduce the loss. Instead, we find that we achieve improved performance by decomposing $\mathbf{H}_{i\to j}^{(OF)}$ into ground-relative pose parameters that can be used to directly supervise the Siamese network output.

In general, any homography can be decomposed into four possible plane-relative poses via a closed form solution using the analytical method of Malis & Vargas [142] as:

$$\mathbf{H}_{i\to j}^{(OF)} \to \{\mathbf{R}_{i\to j}^{(OF)}, \mathbf{t}_{i\to j}^{(OF)}, \mathbf{n}\}_k \text{ where } k = 0, 1, 2, 3 \tag{5.8}$$

where we have camera-relative rotation and translation $\mathbf{R}_{i\to j}$ and $\mathbf{t}_{i\to j}$ respectively, plane normals $\mathbf{n}$ relevant for the homography $\mathbf{H}_{i\to j}^{(OF)}$, and k which denotes the possible solutions. In practice, we obtain these four possible solutions using the OpenCV [43] implementation of this procedure [142].

We need to use knowledge about our problem to discount three of these four possibilities. We know that our cameras in the KITTI dataset are always travelling approximately perpendicular to the road surface. Therefore, the normal $\mathbf{n}$ we require should be close to $(\mathbf{0}, \mathbf{1}, \mathbf{0})^T$. We find that generally two of these normals tend to be negative for the y-component, a physical impossibility. To choose between the remaining two normals we simply select the normal closest to $(\mathbf{0}, \mathbf{1}, \mathbf{0})^T$, and take the associated camera-relative poses as our solution.

5.2.2 Scale Ambiguity

Another challenge with this method is that relative translations are only defined up to an unknown scale, and so the scale for the chosen relative translation in Eqn. 5.8 is ambiguous. A problem we highlighted with the appearance loss method outlined in Chapter 4 is that of scale ambiguity (see Section 4.1.4). In that approach we simply introduced soft priors to let the Siamese network learn the scaling such that the camera height matched the known calibration on average. In contrast, our method in this case is to enforce the correct scaling from the known camera calibration in the KITTI dataset as a hard constraint. Specifically, we know that the cameras in KITTI are 1.65 metres from the ground, discounting the effects of vehicle motion. Therefore, we simply enforce this constraint by multiplying the camera-relative translation by 1.65 (the closed form solution [142] handles unknown scale by normalising the height of the camera above the plane). Malis & Vargas [142] provide the analytical method for decomposing a homography into the possible relative translation, rotation and planar normals. In their work, the relative translation between both cameras is normalised with respect to the distance from the desired camera to the object plane. Note that in our refinement post-training

stage we continue the soft enforcement of priors as in Section 4.2.1 but only for the height for camera i to help constrain our solution to the correct scale.

5.3 Model-fitting in-the-loop with HEM-Loss

We take inspiration from Kolotourous et al. [105] who use model-fitting in the training loop to reconstruct human pose and shape. In their work they recognise the benefit of using an approximate estimate from the deep network as a good initialisation for fitting a parametric body model to a collection of data points, and additionally that the resulting fitted solution, provides a good training signal to the network.

This type of thinking where we can combine more classical model fitting paradigms with deep learning approaches is part of the underlying thesis for our research and works well with our focus on semantic and geometric scene understanding. We find that without semantically isolating the road plane, model-fitting is much more challenging and lacking in performance. As far as we know, we are the first to apply this model-fitting in-the-loop idea to the application of motion estimation (see Sections 2.2.1, 2.2.2 and 2.2.7 for related work).

The application of this idea to our problem is implemented as follows and illustrated in Figs. 5.1 and 5.2. We treat the regressed Siamese network parameters from Chapter 4 as a coarse estimate which we use for warping one input image into the perspective of the other. These two images are stacked and fed into the optical flow network in order to estimate fine-scale disparities. Semantic segmentation and model-fitting is used as described in Section 5.1 to compute a refined homography between network input images. Subsequently, we decompose this homography into camera-relative pose using the method in Section 5.2.

This refined set of camera-relative pose parameters $\mathbf{R}^{(OF)}_{i\rightarrow j}, \mathbf{t}^{(OF)}_{i\rightarrow j}$ are used as pseudo-labels to supervise those camera-relative poses from the Siamese network $\mathbf{R}_{\theta,i\rightarrow j}, \mathbf{t}_{\theta,i\rightarrow j}$ (see Eqn. 4.4) to form the HEM-loss as:

$$L_{HEM} = \|\mathbf{R}^{(OF)}_{i\rightarrow j}\mathbf{R}^{T}_{\theta,i\rightarrow j} - \mathbf{I}\|_2 \; + \; \|\mathbf{t}^{(OF)}_{i\rightarrow j} - \mathbf{t}_{\theta,i\rightarrow j}\|_2 \tag{5.9}$$

Crucially, we note that in contrast to our appearance loss given in Chapter 4, we are not back propagating gradients through our homography estimation module. We are using our model-fitting approach with pre-trained optical flow and semantic segmentation networks to refine our Siamese network homography, and decomposing that to generate more accurate relative poses to supervise those poses previously output from our Siamese network.

This is a non-differentiable approach. As such, we can use the camera-relative pose labels themselves not only for supervision, but at inference time, to help boost accuracy. In our case, we use these relative poses as input to the transformation synchronisation overviewed in Section 4.3 for visual odometry evaluation. It is worth emphasising the benefit of this approach of utilising a refinement module such as this at inference time as it is uncommonly found in the literature. Often neural network pipelines fully rely on the generalisation power of the network, which is largely limited by data quality and quantity. Hence, we note the benefit our non-differentiable approach for avoiding total reliance on network generalisation, and boosting experimental performance.

5.4 Experiments

We evaluate our method quantitatively and qualitatively, and we compare the perceptual loss method outlined in Chapter 4 with our homographic model-fitting approach.

5.4.1 Experimental Details

As outlined in Fig. 5.1, we train a geometric matching Siamese network in two sequential stages. Firstly, we pre-train using the perceptual loss outlined in Section 4.2 and refer to that model in results as PLoss. Secondly, we refine the PLoss model with the Homography Estimation Module (HEM) loss described in Section 5.3 which we refer to as HEM-Train. Thirdly, we apply the HEM to the PLoss model at test time (HEM-Test). Lastly, we also evaluate by applying the HEM to the HEM-Train model at test time (HEM-Train+Test).

Similarly to Chapter 4, we use the transformation syncronisation method by Arrigoni et al. [6] (see Section 4.3) to compute absolute poses from the camera-relative poses derived from our Siamese network output estimation. Specifically, we compute camera-relative poses from Eqn. (4.4) to form the matrix $\mathbf{X}$ in Eqn. (4.11), which is then used to compute absolute poses with the optimisation process outlined in Section 4.3. As previously described in Section 4.4, these absolute poses are used to evaluate quantitatively and to generate visual odometry trajectory visualisations. Again, as our proposed method does not rely on any direct supervision we focus our comparison on leading methods which are fully self-supervised and only rely on a single camera (see Section 2.2.2).

In Fig. 5.6 we show the rotation label loss for the training and validation datasets to illustrate that our model converged. It is clear that most of the gain occurs over

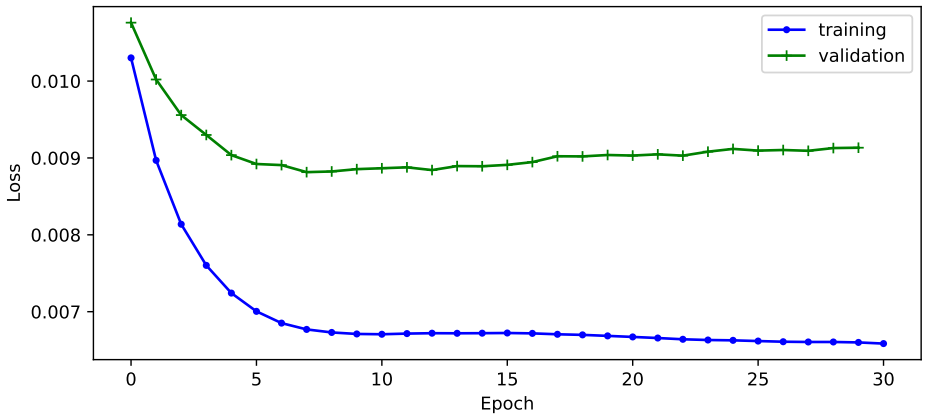

Figure 5.6: Rotation label loss versus training epoch for training and validation data.

the first 5 epochs and hence we stopped training at 30 epochs, where generalisation error diverges.

5.4.2 Quantitative Results

In Table 5.1 we provide visual odometry scores on sequences 09 and 10 of the KITTI benchmark [66], expanding on Tab. 4.1 for our initial stage of training. The first three rows for our results are the same results from Tab. 4.1 which we use here as a baseline. As described in Section 4.4.2, we use the KITTI benchmark translation (%), rotation error (deg/m) and absolute trajectory RMSE (m) for metrics (as in LTMVO [287]).

We use the same table split with differing levels of complexity as described previously in Section 4.4.2. Leading self-supervised monocular methods tend to evaluate quantitatively by using sequences 9 and 10 of the KITTI visual odometry dataset [68, 276, 287]. Most of these values for comparing to other methods are referenced from the LTMVO results [287]. Note that while we do use pre-trained networks for optical flow [89] and road plane semantic segmentation [285], these are only for inference and are not trained.

Results indicate that training with our Homography Estimation Module can significantly improve performance of our Siamese network from Chapter 4. In particular, we note that fine-tuning with our HEM (HEM-Train) provides a state-of-the-art score for ATE on sequence 10. However, performance of HEM-Train is comparable to that for HEM-Test and HEM-Train+Test. Given that inference time will be faster without the additional computation of the HEM, we suggest utilising

Method	Depth	6DoF	S.Inp	Net+	Staged	Seq. 9			Seq. 10		
						t_{err}	r_{err}	ATE	t_{err}	r_{err}	ATE
LTMVO [287]	✓	✓	✓	✓	✓	**3.49**	*0.010*	**11.30**	*5.81*	*0.018*	*11.80*
TBG [276]	✓	✓	✗	✓	✓	6.93	**0.004**	-	**4.66**	**0.006**	-
CC [187]	✓	✓	✓	✓	✗	6.92	0.018	29.0	7.97	0.031	13.77
GeoNet [257]	✓	✓	✓	✓	✗	28.72	0.098	158.4	23.90	0.090	43.04
SfM [282]	✓	✓	✓	✓	✗	8.28	0.031	24.31	12.20	0.030	20.87
SC-SfM [13]	✓	✓	✗	✗	✗	11.20	0.034	-	10.10	0.050	-
Mono2 [68]	✓	✓	✗	✗	✗	11.47	0.032	55.47	7.73	0.034	20.46
Ours (PLoss fixed priors)	✗	✗	✗	✗	✗	10.42	0.033	29.45	9.55	0.048	13.73
Ours (PLoss mono2-net)	✗	✗	✗	✗	✗	16.69	0.058	58.88	16.72	0.071	32.0
Ours (PLoss)	✗	✗	✗	✗	✗	11.30	0.043	28.68	11.66	0.060	16.48
Ours (HEM-Test)	✗	✓	✗	✗	✗	*6.13*	0.017	*15.73*	7.38	0.033	*11.80*
Ours (HEM-Train)	✗	✗	✗	✗	✓	7.14	0.023	16.27	8.58	0.031	**11.72**
Ours (HEM-Train+Test)	✗	✓	✗	✗	✓	6.53	0.018	19.65	7.19	0.037	12.77

Table 5.1: Extending Tab. 4.1 visual odometry results on KITTI for our perceptual loss (PLoss), HEM-Loss post-training and HEM-Loss post-training with test-time refinement. Metrics t_{err} (%), r_{err} (deg/m) and ATE are translation, rotation error over a set of subsequences, and the absolute trajectory RMSE (m) respectively.

the HEM-Train model for applications where inference speed is of significance. Nevertheless, our performance is refined with HEM application at inference time on PLoss (HEM-Test), which helps to remove reliance on the generalisation power of the Siamese network.

The competing methods always involve training an additional network for a dense map (depth [13, 282, 287] and optical flow [187, 257, 276]). While they are able to obtain similar performance without the need of a scene assumption, their reliance on estimating thousands of additional parameters can make them harder to train. Using our ground-relative modelling and model-fitting in-the-loop methods, our results are highly competitive with leading self-supervised approaches (see Section 2.2.2), while remaining flexible and unconstrained, as we discussed in Section 4.4.2.

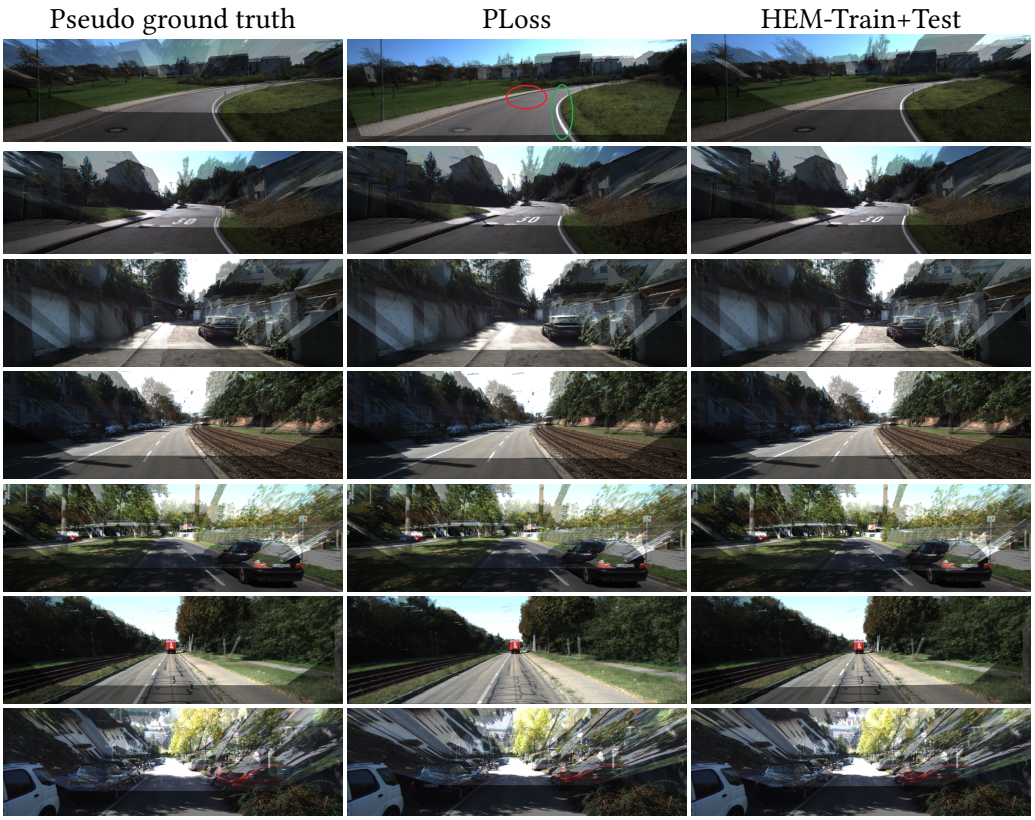

Figure 5.7: KITTI qualitative results, best viewed zoomed in for alignment detail. Images are a composition of one Siamese network input with its warped counterpart input. Left: Ground truth where we assume fixed prior values for ground plane cross-projection. Middle: Our full ground-relative pose result with perceptual loss pre-training. Right: Our Homography Estimation Module (HEM) applied at training and test time.

5.4.3 Qualitative Results

Qualitative results are shown in Fig. 5.7. As described in Section 4.4, we use pseudo ground truth. In the first example the ground truth performs poorly (perhaps due to the unknown roll relative to the ground) and in our PLoss version, features such as road lines (green) align but other features misalign globally (red), though our HEM method significantly corrects these errors. Examples display increasing refinement, particularly in the penultimate example where though our PLoss has found a suitable rotation and failed with estimating an accurate translation, our HEM is able to correctly recover an accurate transformation. The last example shows a fail case where our HEM method is unable to achieve alignment (see

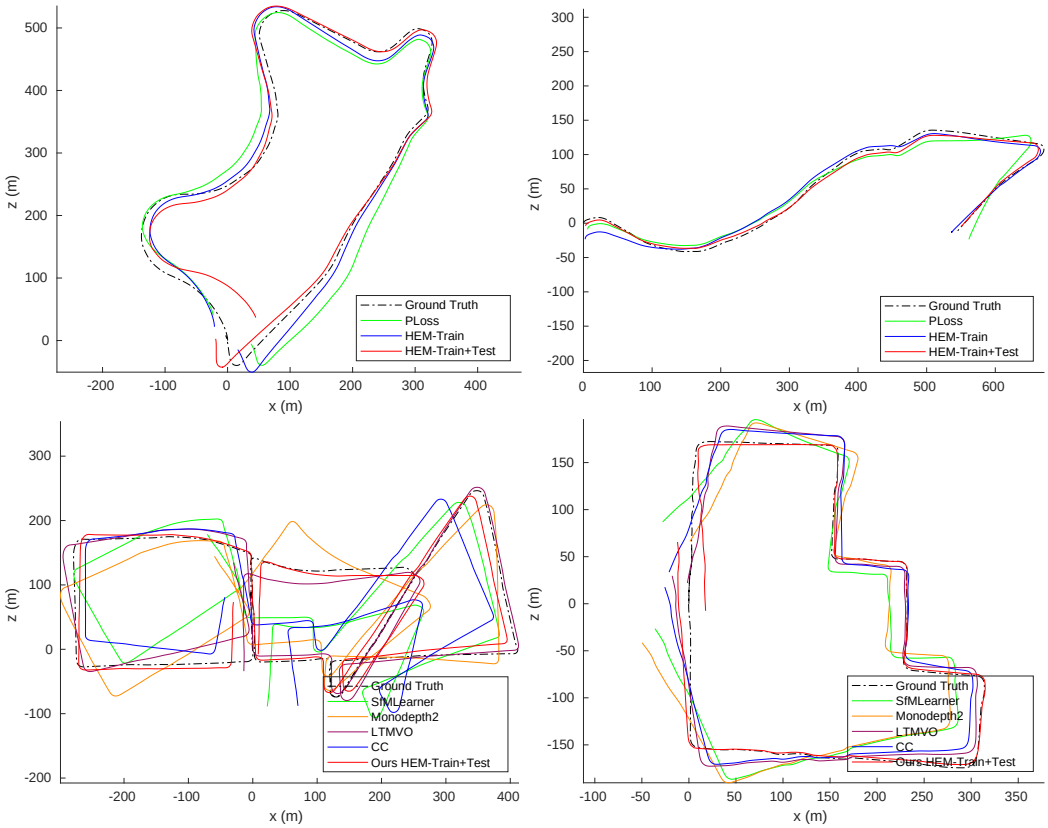

Figure 5.8: KITTI visual odometry trajectories for seqs. 9, 10, 13 and 15 respectively. For seq. 9 and 10 we compare between our training methods: pre-trained perceptual loss alone (PLoss), and post-training with our HEM at training and test time.

manhole cover), possibly due to excessive glare in the road plane, resulting in high translational error. In summary, the PLoss model performs visually very well but is inclined to misaligning features in one direction, which is likely due to cases where it converges to a local minimum. The HEM refinement is able to correct these errors but can be prone to illumination issues.

5.4.4 Trajectories and Path Length

In Fig. 5.8 and 5.9 we show predictions for trajectories on sequences 09 and 10 and also four trajectories on the benchmark test set. For the benchmark sequences we compare with leading self-supervised methods. While LTMVO [287] generally achieves the best trajectories, we can see that our method remains competitive,

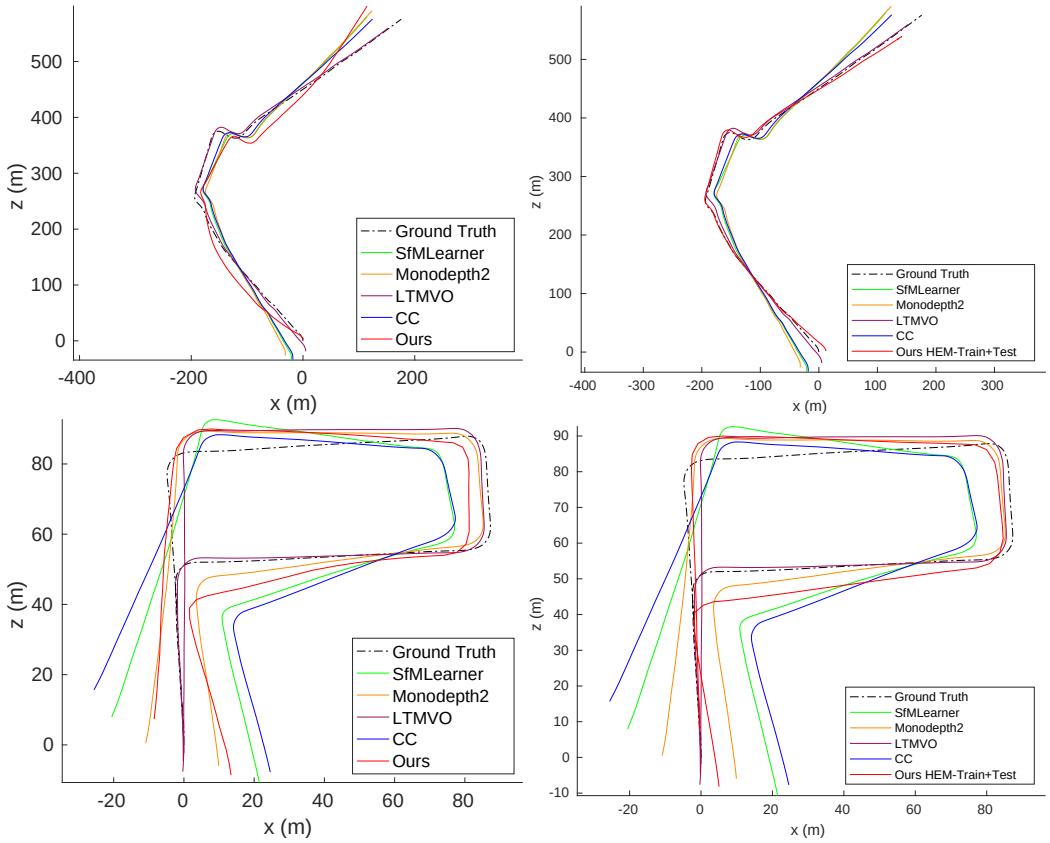

Figure 5.9: KITTI visual odometry trajectories for seq. 11 and 14 respectively. We compare with leading self-supervised methods with applying our HEM at training and test time (HEM-Train+Test).

again noting the effectiveness particularly on Sequence 14 which contains imagery very different to the rest of the test sequences. For sequences 09 and 10 we observe significant improvement in visual trajectory from training only with perceptual loss to using the homography estimation and decomposition approach. Similarly, in Fig. 5.9 comparing with Fig. 4.10 we observe significant improvement in accuracy on sequences 11 and 14 with our HEM applied at training and test time.

For sequences 13 and 15 perceptual loss performed badly and we only show our improved result with competing methods. We note that for sequence 15 HEM-Train+Test we achieve visually comparable results to competing methods. Similarly, for sequence 13 we found an overall trajectory using appearance loss alone failed to achieve a comparable result, but with our HEM-Train+Test method we are able to

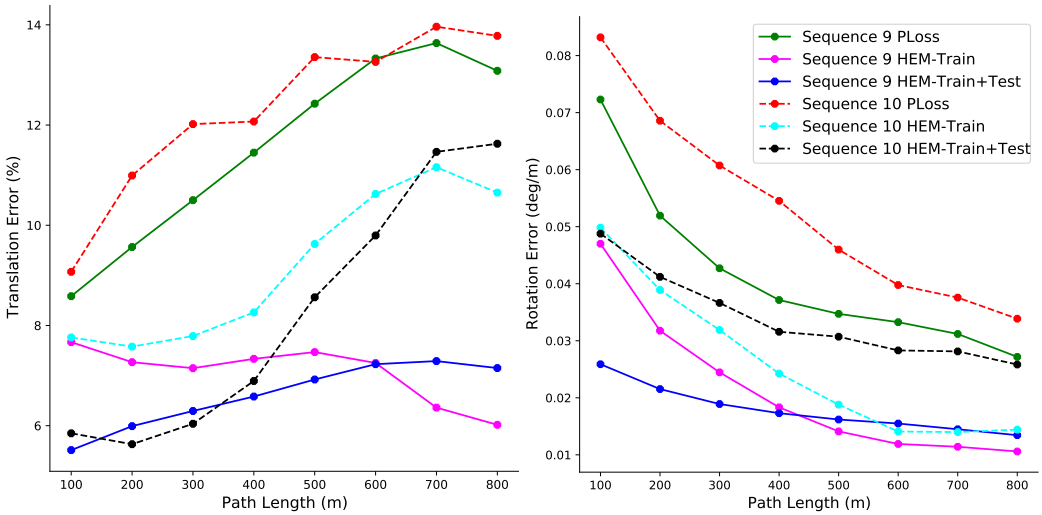

Figure 5.10: Translation and rotation errors by path length on sequences 09 and 10. We compare errors with pre-training with perceptual loss (PLoss), post-training with our homography estimation module (HEM-Train) and additionally applied at test-time (HEM-Train+Test).

recover error and improve results to be visually comparable to competing methods on this sequence.

In Fig. 5.10 we show how translation and rotation errors vary with trajectory path length on sequences 09 and 10 and compare with our previous results from Fig. 4.12. Generally errors are refined with each method but interestingly we observe that the rotation error on sequence 10 is higher after applying the HEM at test-time.

5.5 Conclusions

In this chapter we have illustrated a method of combining geometric and semantic scene understanding with classical model-fitting to refine the accuracy of a geometric neural network for relative pose estimation at both training and inference time. In Section 2.2.7 we discussed work that explicitly leverages traditional model-fitting within deep pipelines. We observed a lack of literature applying this idea to relative pose estimation and in this chapter we show it be useful in bringing our approximate solution from Chapter 4 to be highly competitive with the leading self-supervised relative pose estimators discussed in Section 2.2.2.

Camera-to-camera ground-relative homographies can be refined by robustly fitting a homographic model to correspondences: In Section 5.1.1 we show that correspondences can be computed between image pairs which are already nearly aligned. The exact method which is used to compute these correspondences could vary, but we chose to use an optical flow network for its accuracy and fast-inference. These correspondences contain information which is directly helpful to our goal of improving relative pose accuracy. We have shown that classical homograpic model-fitting [175] to hundreds of these correspondences with RANSAC [61] provides significant performance boost to our self-supervised model from Chapter 4. Furthermore, with this boost in accuracy, our method is highly competitive with state-of-the-art [13, 68, 187, 257, 276, 282, 287] end-to-end self-supervised monocular visual odometry approaches (see Section 2.2.2), with a leading score in ATE on Sequence 10 (see Tab. 5.1).

Pseudo-labels are useful for improving accuracy at both training or inference time: Our refined camera-relative poses from the HEM are more accurate than those obtainable from the appearance loss method of Chapter 4 alone, and bring our performance to be comparable to state-of-the-art self-supervised deep visual odometry approaches (see Section 2.2.2). The utility of these poses are two-fold: 1. we use them to train the network further to increase its generalisation performance. 2. they are useful in themselves at inference time, because they can be input directly into the transformation syncronisation routine (see Section 4.3) for significant performance boost to our original model from Chapter 4 (see Tab. 5.1). This inference time application allows us to be less restricted to generalisation errors inherent with training on biased datasets with neural networks [87].

Semantic scene understanding can be combined with ground-relative pose estimation in road scenes via non-road plane scene filtering: The correspondences we use to fit a homographic model to are only relevant for those scene points which lie on the road plane. By leveraging semantic scene understanding in the form of a state-of-the-art semantic segmentation network [285], we can filter

noisy correspondences to make model-fitting a viable tool within deep relative pose estimation.

We find that semantic scene understanding can be helpful within the context of geometric scene understanding as a way of filtering out parts of the scene which do not apply to the geometric regularity we are attempting to utilise and capture within the learnt features of the network. Furthermore, we note that applying thoroughly understood model-fitting methods such as RANSAC [61] and state-of-the-art models in optical flow [89] and semantic segmentation [285] can have a place within deep learning pipelines for tasks such as visual odometry.

Homographic decomposition can be used within a deep pipeline to obtain camera poses for visual odometry: In Chapter 4 we utilise a ground-relative parameterisation of two cameras to form a homography between both cameras for the purposes of forming a self-supervision signal. In Section 2.2.4 we noted a lack of literature explicitly linking relative pose estimation with homographies in a deep pipeline, particularly within the road scene domain. Yet within planar scenes such as these, homographies are highly related with camera motion [75, 142, 181].

In Section 2.2.4 we go further and noted a particular gap where there is a lack of work exploring the use of the method by Malis & Vargas [142], an approach for analytically decomposing a homography to camera-relative motion, within deep pipelines. In Chapter 5 we show how we can utilise this method in such a pipeline by decomposing our refined homographies back into relative pose. While this decomposition allows us to achieve highly competitive visual odometry performance with leading self-supervised methods, we suggest that homography decomposition could be further researched within current state-of-the-art deep homography estimation literature (see Section 2.2.4) as a potential benefit to relative pose or homography estimation applications and to help further bridge these two fields of research.

6 Conclusions & Outlook

Road scene images are rich with semantic and geometric information which the human mind models and understands intuitively. It seems likely that our perception of the world and how we move through it is informed by our basic understanding of the structure of scenes. In this thesis we have sought to move away from black-box deep learning methodology and to better inform networks with what we already know about the scene contents. Specifically we have chosen to focus research on harnessing two sources of structure: semantics and geometry.

We emphasise that other self-supervised monocular visual-odometry techniques on the KITTI dataset use a combination of two networks: a dense depth or optical flow estimator, and a 6 DoF camera-relative pose estimator. Firstly, this attempts to implicitly learn the geometry of entire complex 3D scenes simultaneously with relative-pose, training for thousands of parameters, which is a very challenging proposition. Secondly, most works estimate only 6 DoF relative pose and we show that by re-framing the output into a geometry aware context we obtain a more generally powerful representation of relative pose.

6.1 Summary of contributions

In Chapter 3 we focused on exploiting structure within class hierarchies for distinguishing between minor and serious errors for the task of semantic segmentation. In contrast to segmenting without prior knowledge, we discovered that it is promising to train a network using a pre-formulated hierarchy of classes in a way which boosts performance or reduces the amount of training required to achieve a desired level of performance. Specifically we provided a novel implementation for a hierarchical loss which we evaluated on two very different types of dataset - a facial [112, 205] and road scene [169] dataset. For both datsets we found an improvement in segmentation accuracy overall, while requiring less training epochs.

In Section 2.1.5 we discussed literature around leveraging hierarchical knowledge for scene understanding, particularly with respect to deep pipelines, and in Section 2.1.4 we discussed literature around semantic segmentation. We observed a lack of work where class hierarchies are used to construct training losses for differentiating between the severity of the class errors in the training process for road scene

semantic segmentation [3, 45, 58, 91, 164, 253]. Ideally we would implement leading architectures for semantic segmentation (see Section 2.1.4) to place our work into a greater context as, for example, achieved by Koguciuk et al. [103] for their perceptual loss applied to deep homography estimation. However, as the architectures for leading semantic segmentation approaches vary widely from transformers [9, 35] to CNN [16, 29, 64], we chose to keep the core architecture simple with a popular U-Net [191] such that we could focus purely on comparison between training with a vanilla cross-entropy loss and our hierarchical loss.

In Chapter 4 we explored how we can leverage geometric understanding of road scenes into training a Siamese network for estimating motion. The obvious geometry of road scenes is that vehicles move approximately around a planar field, at least locally where two images capture the same section of road, and it is within this setting we explored relative pose estimation. We provide a novel 9D parameterisation of network output which is physically interpretable and rooted within a locally planar model of the scene.

In Section 2.2.2 we identified leading approaches for end-to-end self-supervised relative pose estimation on the KITTI benchmark [66]. We observed that these approaches generally have three limitations: 1. reliance on training for tens of thousands of parameters with dense depth, optical flow or additional networks in conjunction with a pose network and 2. limiting relative pose magnitude by concatenating network input, and 3. limiting relative pose to camera-to-camera translation and rotation.

We discover that it is possible to entirely replace dense depth estimation which other leading self-supervised approaches take [187, 276, 287], with our ground-relative parameterisation, to provide a self-supervision signal in the form of an appearance loss. As a result we greatly reduce the number of parameters we need to learn towards solving for relative pose. Specifically, we discover that perceptual loss [94] is effective in self-supervising a deep network to an approximate solution for the task of visual odometry. Furthermore, we find that the Siamese architecture proposed by Rocco et al. [190] to be particularly effective for our task of relative pose estimation.

Leading self-supervised visual odometry methods for road scenes phrase relative pose with respect to 6D camera-to-camera pose [13, 68, 187, 257, 276, 282, 287]. Instead, we find that by phrasing relative pose of two cameras with respect to our novel 9D ground-relative network output, we may include the road geometry within our estimation process, effectively fitting a series of planar patches to visual odometry sequences. This allows for clear benefits: 1. a more general and hence powerful representation of pose, 2. less restricted relative pose versus leading

methods which rely on sequential image framing [276], and 3. self-supervision via ground-plane homographic cross-projection.

In Section 2.2.4 we discussed leading deep homography estimation methods and observed a lack of literature combining homographies and relative camera pose with respect to translation and rotation tensors. In Chapter 4 we contribute a relative pose estimation approach where camera-to-camera homographies relating to the road plane geometry can be computed in conjunction with camera pose. While we use these homographies to form a self-supervision signal, they could be utilised for applications such as orthomosaicing [214].

We have the basis of a method where we can estimate relative poses from sequences of images through training a single deep pose network without using any direct supervision with annotations, and where the local planarity of the road is leveraged and modeled within our parameterisation and training approach. We obtain competitive performance on the KITTI road scene dataset [66], but found that our model was liable to errors due to illumination issues, similar features forcing training into local minima, and issues with strongly non-planar regions (see Section 4.4).

In Chapter 5, in order to address some of these limitations we further utilise our homographic integration and additionally combine semantic knowledge into the learning pipeline while still leveraging the approximate performance of our model from Chapter 4. In Section 2.2.7 we discussed explicit approaches of using traditional model-fitting techniques within a deep learning pipeline. We recognised that we could fine-tune our performance by taking a model-fitting in-the-loop approach which so far seems not to have been applied to motion estimation tasks [15, 105]. Leading self-supervised relative pose estimation literature generally rely wholly on the models fitted from their deep pipelines [13, 187, 257, 276, 287], and we recognised that traditional model-fitting could be applied at inference time to tackle the limitation caused by the bias present when utilising neural networks [87].

We combine semantics and geometric understanding with the more classical methodology of model-fitting to a collection of points. Specifically, we discovered that by isolating the planar geometry of the road plane with a semantic segmentation network, we can refine the homography between an image-pair by fitting a homographic model to a collection of road plane correspondences generated by an optical flow estimator (see Section 5.1.1). While the pre-trained semantic segmentation [285] and optical flow [89] networks are highly competitive implementations, they still produce error and hence we discovered that using RANSAC [61] as our model-fitting routine to be particularly effective.

Subsequently, we discovered that by using knowledge of the motion geometry it is possible to decompose the refined homography into a camera-relative pose for

the purposes of further training (see Section 5.2). To the best of our knowledge we are the first to apply model-fitting in-the-loop (see Section 2.2.7) and the analytical decomposition of homographies by Malis & Vargas [142] (see Section 2.2.4) to the task of monocular self-supervised relative pose. Moreover, we find that our model-fitting method has the added benefit of boosting performance at inference time for tasks which require camera-relative pose.

Neural networks are only as good as the data they are trained on, and are prone to generalisation error and dataset bias issues. Therefore, we emphasise and utilise the fact that our pseudo-label generation method we propose in Chapter 5 has this dual-utility. We improve significantly upon our result in Chapter 4 (see Section 5.4), and achieve competitive visual odometry results versus other leading self-supervised monocular methods (see Section 2.2.2) on the KITTI dataset [66].

6.2 Overarching Conclusions

We make the following overarching conclusions from our work on integrating hierarchical semantics and scene geometric relative pose:

- **Differentiating between classes in a hierarchical structure helps with training semantic segmentation networks**: In Chapter 3 we show that semantic segmentation performance is improved by leveraging the hierarchical structure of labels for directly supervised training (see Tab. 3.1). Furthermore, training convergence is improved as fewer epochs are required to achieve the same level of performance compared to a vanilla training loss where errors between all classes are treated equally (see Section 3.4.2). In Section 2.1.5 we discussed literature utilising hierarchical approaches [25, 173, 286] within deep learning in general and in Section 2.1.4 we reviewed state-of-the-art semantic segmentation approaches [9, 33, 35, 117, 237, 279], particularly for road scene evaluation [16, 29, 64]. While we note that replicating these methods may help to put our hierarchical loss into a greater context, our approach is specifically to show improvement in performance between a vanilla loss and our hierarchical loss, which is applicable to any semantic segmentation network in general.

- **Choice of network output parameterisation matters for end-to-end self-supervised relative pose estimation:** Road scenes are highly regular, however, the state-of-the-art for self-supervised pipelines for visual odometry [13, 68, 187, 257, 276, 287] attempt to explicitly model the depth of every pixel with a separate neural network. Dijk et al. [51] show that common road

scene depth networks utilise the vertical image position of objects, rather than their overall size. Moreover, they show that generalised depth accuracy can depend on the presence of accompanying features for objects (e.g. shadows). By modelling the regularity of the road plane with our ground-relative parameterisation, we can keep network output parameter numbers low and avoid potential errors caused by additional networks, simplify the training process. Furthermore, keeping a geometric parameterisation general is more powerful from an application perspective, as we can extract from it multiple useful transformations such as camera-relative poses or homographies.

- **Domain knowledge is useful for decomposing homographies between cameras:** State-of-the-art deep relative pose methods [13, 250, 276, 282, 287] for road scenes fail to leverage homographies (see Section 2.2.2) while leading deep homography estimation methods [22, 103, 258, 269] avoid linking homographies to relative camera pose altogether (see Section 2.2.4). In our case, knowledge of our planar road motion aided an analytical method [142] for decomposing homographies into a camera-relative pose useful for training our network and boosting performance at inference. However, decomposing a geometric transformation into camera poses could be more widely useful in other applications or pipelines where we are transforming between camera poses and geometric mappings.

- **Choice of architecture appears to matter in geometric problems:** In Tab. 4.1 we show that using a geometric matching Siamese network [190] helped significantly versus a simpler more generic architecture [68] which concatenates a pair of input images. State-of-the-art deep homography estimation approaches (see Section 2.2.4) use this Siamese style of architecture [22, 272], which is not leveraged within the self-supervised relative pose literature for the main focus of visual odometry evaluation (see Section 2.2.2). As discussed in Section 2.2.5, we suggest that input concatenation should be avoided in favour of Siamese networks where each image is fed separately through a feature extractor. Moreover, this allows for more flexibility towards an arbitrary relative pose between the input image pair.

- **Perceptual loss can be used for self-supervised relative pose estimation:** In Section 2.2.6 we discussed literature around using perceptual loss for training neural networks. While a state-of-the-art accuracy has been achieved with deep homography estimation by simply leveraging a perceptual loss [103], we find a lack of literature leveraging a perceptual loss with a focus on estimating relative pose. We found that learning from a perceptual

loss (as popularised by Johnson et al. [94]) allows for competitive visual odometry versus other state-of-the-art end-to-end self-supervised relative pose estimation approaches (see Section 2.2.2).

- **Model-fitting is helpful at both training ("in-the-loop") and inference time:** In Section 2.2.7 we discussed literature explicitly leveraging traditional model-fitting within the training loop. As far as we know, methods in the literature for the tasks in this thesis of semantic segmentation (see Section 2.1.4) and self-supervised relative pose estimation (see Section 2.2.2) do not utilise model-fitting in-the-loop. Our results in Tab. 5.1 indicate that model-fitting at training and inference time can be a useful tool. Self-supervised methods rarely apply model-fitting at inference time, and inherent generalisation error from deep models limits accuracy of solutions. We find it helpful to refine solutions beyond these limitations by applying model-fitting at inference time. In particular, we note that this can be made possible by letting your model-fitting process compute pseudo-labels, such that they can be used for both directly supervised training and as a refined solution.

- **Semantic scene understanding is useful for filtering out scene contents which do not conform to your geometric model:** In Section 2.1.4 and 2.2.2 we discussed literature including the leading work around the tasks of semantic segmentation and self-supervised pose estimation respectively. While certain competing self-supervised relative pose methods utilise motion segmentation [13, 187], this involves training for a dense map (which we explicitly avoid), and additionally they do not leverage semantic segmentation explicitly. In our case, fitting a homographic model to scene correspondences between planar regions aided the model-fitting solution (see Chapter 5). Generally, we conclude that semantic scene understanding becomes particularly useful in combination with geometric scene understanding when it is used to filter out scene points which do not readily apply to your geometric regularity.

6.3 Critical Analysis

In this section we provide critique relating to weaknesses and limitations in our work. We split our analysis by chapter but we note that an overarching limitation is that we do not test for speed as we do not code for a fast implementation.

6.3.1 Chapter 3:

Practical Issues We would ideally compare our hierarchical method against other methods in the literature but this would require reproducing many methods where often code was unavailable and thus we decided to keep experiments limited to with and without our hierarchical loss using a popular architecture. Our primary aim at this stage was not to produce competitive performance but simply to show a benefit from leveraging a class hierarchy within labels for supervising on the task of semantic segmentation.

A further practical limitation is that the class hierarchy which we utilised for the Vistas dataset is somewhat shallow. While ideally we would have some deeper branches, we felt that keeping a balanced tree was sensible and an intuitive hierarchy was already presented from the publication by Neuhold et al. [169] which introduced the Mapillary Vistas dataset [169].

Theoretical Weaknesses While we chose to use a shallow hierarchy of classes for the road scene dataset, we recognise that many variations on tree depth per branch is possible. In particular, if we were to use an unbalanced tree where one of the branches is significantly deeper than the rest, our loss could be biased towards classes in that branch. This remains a theoretical weakness where we could investigate ways of tackling this scenario.

Another weakness is that currently our theory is limited to simple trees and not those containing cycles or directed edges. Further, we do not account for ordering in classes and leave this to future work.

Moreover, we note that our approach is somewhat limited as we only exploit hierarchies within the training signal. While this is beneficial for the flexibility of using our loss to train any network, performance could possibly be improved by integrating a hierarchy into the network itself. For example, coarse classes could be estimated from earlier network layers, while finer classes are estimated by later layers.

6.3.2 Chapter 4:

Practical Issues For evaluating our method for relative pose estimation we only use the KITTI dataset [66], where we could have also used other road scene datasets such as Cityscapes (though this dataset lacks ground truth poses). We compare to other fully self-supervised approaches and note that we could have compared to more methods.

Some practical issues exist in the fairness of comparisons. Our results in Table. 4.1 are referenced from the LTMVO paper [287] where SfM, GeoNet, CC and MonoDepth2 results where obtained by running pre-trained models on sequences 9 and 10. We note that we do not know whether some of these might have been trained with other datasets. For example, CC contains a variation where they have trained on both KITTI [66] and Cityscapes [38]. While we feel that evaluating quantitatively on only two sequences is limiting, we note that this is common practice with the competing approaches.

Another practical limitation is that the VGG [202] network for the perceptual loss in Section 4.2 only uses ImageNet for pre-trained weights. While this is also a common practice, our concern would be if another feature space, perhaps one trained with KITTI, would be more suitable.

A general weakness of appearance loss methods is that of illumination. Specifically, we commonly see oversaturation of pixels where there is significant glare on the road surface, which may appear in one or both of the images in a pair. In these cases it can be very difficult to perform any kind of feature matching.

Finally, we note that the code we use for Transformation Syncronisation by Arrigoni et al. [6] is only available as MATLAB code currently, which limits our ability to investigate speed and efficiency of our method with regards to estimating absolute poses.

Theoretical Weaknesses The most obvious theoretical weakness for our ground-relative cross-projection appearance loss is our assumption that the road-surface is planar. We note that while we can approximate the road surface globally by a series of planar patches, this can be a significant issue where the road sometimes sharply bends (e.g. at the crest of a hill) and is a source of error. We recognise that more in depth analysis of the impact of the planar assumption could have been investigated, which we note in future work.

As shown in our results, occasionally we have an image-pair where little road-surface is observed (e.g. around some corners) and these present a weakness as our model will not be able to cross-project meaningfully in these outlying cases.

Another weakness presents itself in the form of the construction of the road surface itself. For example, if an image-pair is captured where the vehicle moves over a speed bump in one image, then our planar model will be challenged.

Moreover, illumination can cause various weaknesses within our approach. For example, shadows may be cast on the road surface by dynamic objects (e.g. cars, people, trees perturbed by wind or time of day) and our modelling does not currently account for this.

Furthermore, in Chapter 4 we did not integrate any form of segmentation. It would make more sense to only use the road-surface in our appearance loss and we argue that we are relying too heavily on the Siamese network ability to ignore feature matching of the non-planar scene. In practice, we had some practical training issues with training with segmented input and we leave this for future work (although we did successfully utilise segmentation with our method illustrated in Chapter 5). We note that these issues may be due to the void regions and that filling them with appropriate substitutes or avoiding their contribution to losses, could help, as noted by Zhao et al. [277].

6.3.3 Chapter 5:

Practical Issues Originally we intended to research combining semantic knowledge of Chapter 3 with the geometric parameterisation proposed in Chapter 4 in a deeper and more direct fashion. For example, we would have preferred to train the semantic segmentation network outlined in Section 5.1.2 with our hierarchical loss, but decided that this would limit the performance of our results as our segmentation results were not state-of-the-art. Therefore, we used a leading pre-trained semantic segmentation model.

Furthermore, the semantic segmentation model we use is trained using sparse annotations [285] and therefore we could argue that our method is not entirely self-supervised as it has benefited from this pre-trained model. We also note that we are limited by the performance of the optical flow and semantic segmentation accuracy.

We have chosen to focus primarily on performance and concept, rather than efficiency, and we did not make a fast implementation. It is worth noting that the HEM efficiency could be limited if requiring inference on large optical flow and segmentation architectures. Lowering the input image resolution would help to resolve this issue but could limit performance of the segmentation and optical flow results.

Theoretical Weaknesses Our HEM is somewhat reliant on the pose Siamese network providing a coarse initialisation so that the warped source and target images (see Fig. 5.2) are reasonably well aligned to accurately perform optical flow inference. Ideally we could like to make our HEM method work from a random initialisation, which we leave to future work.

We note that RANSAC in Section 5.1.3 requires setting a threshold to determine inliers from outliers, which presents a minor weakness with determining a reasonable value.

In Section 5.2.2 we resolve scale ambiguity for the relative camera translation obtained from decomposing the refined homography. We do this by simply multiplying by the known average camera height from the road surface but in reality the camera height will vary due to road variations and vehicular motions (e.g. due to the suspension). Therefore the relative camera translations will contain some error and this will propagate through to our visual odometry results. However, we are still able to achieve significantly better performance over utilising our appearance loss method in Chapter 4 alone.

Finally, for the method of computing correspondences, it could be that local feature points might be beneficial versus our choice of using dense optical flow.

6.4 Personal reflections

At the beginning of our research the field of deep learning with tasks of scene understanding was very focused still on the idea of giving the neural networks complete freedom over learning, with many approaches training additional networks, increasing the complexity and challenge greatly. While many high quality data sources to train deep networks exist, we believe that we can only progress so far with this type of catch-all methodology because our datasets will never be perfect and complete, our networks never fully-adept function approximators, and it is therefore important to focus efforts on training with prior scene understanding in mind. In this research we have entirely replaced the need for depth estimation and re-doubled our efforts on utilising the most obvious ques for the motion of a vehicle moving along a road-surface: the road is locally planar in geometry and consistent in appearance. Over the past few years we now recognise that scene understanding is progressing towards the idea of integrating knowledge of domains within a deep learning arena, and combining more of classical modelling, as we reach limits in black-box deep learning. We recognise that there is still a major focus on implicit learning across machine learning and related fields, and a lack of well understood modelling. Specifically, our concern is that there is still an over reliance on implicit learning towards achieving state-of-the-art results in well accepted benchmarks. We suggest that the field will develop towards datasets and evaluation methodologies which incentivise competition towards learning more robust features.

6.5 Future Work

In this section we highlight areas where there is a potential gap in research which our work could be expanded into.

Further Semantics

- Utilising Semantic Segmentation: In Chapter 4 we described an appearance loss based on cross-projecting planar scene points. While there exist recent works around combining semantic segmentation with appearance losses in general [12, 32, 182, 183, 245], we would like to investigate the benefit of utilising semantics to specifically filtering out non-planar points of either the input of our Siamese network or the input to the VGG [202] network of our perceptual loss.

- There are various contemporary works around the issue of class and data imbalance for classification tasks [95, 114, 139, 216]. Regarding our hierarchical semantics research from Chapter 3, we would like to investigate potential issues of unbalanced trees and the possibility of handling more complex relationships with other types of trees. Furthermore, we suggest that semantic hierarchies could be learnt themselves in a way which optimises performance for tasks such as semantic segmentation.

- Additionally, further work could include the ability to extract segmentations at multiple levels in a hierarchy describing your data. This would be quite useful, intuitive and is not something commonly achieved by semantic segmentation solvers in the current literature [74, 110, 156].

Combining Semantics / Alternate Forms of Supervision

- In Chapter 5 we used leading pre-trained segmentation and optical flow models [89, 285], but future work could combine the hierarchical training from Chapter 3 into this setting. Moreover, we see that an additional loss could be used to help guide the supervision where we cross-project the road labels into the corresponding views and operate a classification loss on the pixel level.

- In our work we have specifically avoided training for dense depth or optical flow estimation for the task of visual odometry, although we see potential benefits with regards to training for semantic segmentation, specifically in

terms of training networks for pose and semantic segmentation simultaneously. Multi-task learning is a popular and multi-faceted field with many different approaches [274]. In our case we might expand our research and consider learning camera-relative pose estimation with semantic segmentation, in a similar way to Zhang et al. [273], but without the requirement for depth estimation. If the segmentation network is performing well, then the same road plane scene points in each view should be given the same class, and any misalignment here should help the geometric network to learn. Further, if image-pairs have no semantic labels, the planar cross-projection helps to constrain the solution of scene points between views which should be identical. Hence, while we use a pre-trained network, there is possibly a benefit of letting pose and segmentation networks learn from each other, perhaps in helping to reduce supervision by guiding the networks into the most physically sensible solution space.

Alternate Architectures

- For the optical flow and semantic segmentation networks [89, 285] of Chapter 5 we suggest experimenting with different architectures and competing models to see whether there is a significant improvement utilising another perhaps more accurate pre-trained model [19, 64, 92, 128, 211].

Planarity and Beyond

- While we use homographies for modeling the local road geometry, there are more complex geometric modeling possibilities. For example, Rocco et al. [190] include thin-plate splines as a transformation output from their geometric CNN, Friji et al. [62] use non-ridged transforms for human action recognition, and Chen et al. [26] propose predicting more arbitrary transforms for image registration. Our planarity assumption is the primary theoretical weakness in our work and we suggest investigating possibilities of modelling non-linear road geometries.

- While we attempted to analyse how the planarity assumption impacts results by looking at sections of sequences with significant vertical variation, we recognise that more in depth analysis of the impact of assuming planarity could be researched. For instance, we could potentially apply aerial imagery unto a ground-plane, and generate a pair of images from virtual camera positions. Gradually flexing the ground plane towards a quadratic curve in this virtual setting could help analyse cross-projection errors as we move

further away from a planar surface. Though there are a great many recent works around aerial imagery [11, 23, 97], as far as we know this kind of analysis for cross-projecting aerial images into new views with a perturbed plane would be novel.

- Currently, there appears to be little research with regards to learning unusual camera-relative poses from images captured at significantly varied translation and pose [1, 3, 91, 98]. We would like to explore in detail how our local-camera parameterisation and planarity assumptions can help with estimating arbitrary poses with much greater pose displacement. For example, we could cross-project images taken by vehicles at opposite ends of a junction with our planarity model, to learn relative pose between different vehicles overlooking the same scene. Such work could contribute towards solving for tasks which could become more relevant as autonomous vehicles become more prevalent and accepted (e.g. traffic optimisation, routing and planning, or shared augmented reality between vehicles or pedestrians [20, 81, 180]).

Illumination Limitations

- We find that the performance of our training signals are still somewhat limited by visual issues like illumination extremes (e.g. glare) which limit feature matching. There are various recent works towards illumination-invariant pipelines for road scenes [4, 34, 88, 119, 280]. For future work we suggest investigating ways of detecting image pairs where illumination issues within our visual odometry approach are prevalent and adjusting training regimes to help tackle this issue.

Further Post-Processing

- In Section 4.3 we described how we utilised transformation synchronisation to form a collection of relative poses into absolute poses. There are various contemporary works which use additional optimisation routines (e.g. bundle adjustment) in order to improve the performance of road scene related tasks [34, 41, 163, 230, 250]. There may be benefit of performing a non-linear post-optimisation step with the absolute pose result, formed using the Arrigoni et al. [6] transformation syncronisation approach, as an intialisation, which we leave to future work.

Data Quantity

- Various recent works investigate the impact of data quantity on performance of deep models in computer vision tasks [10, 21, 84, 134, 147, 158, 212]. For future work we would like to firstly research how our hierarchical semantic segmentation could be applied to multiple datasets, and potentially to reduce data labelling requirements. For example, if we had two datasets of faces, but one of them is labelled with less classes, we could potentially still train on both datasets by forming an appropriate class tree spanning these datasets. Secondly, for relative pose estimation, we propose to investigate if our method of removing the requirement of training dedicated depth or optical flow estimation networks actually reduces data quantity requirements.

Bibliography

[1] Lucas R Agostinho, Nuno M Ricardo, Maria I Pereira, Antoine Hiolle, and Andry M Pinto. **A Practical Survey on Visual Odometry for Autonomous Driving in Challenging Scenarios and Conditions**. *IEEE Access* 10 (2022), 72182–72205 (see pages 12, 15, 17, 23, 26–28, 103).

[2] Subhash Chand Agrawal and Anand Singh Jalal. **A Comprehensive Review on Analysis and Implementation of Recent Image Dehazing Methods**. *Archives of Computational Methods in Engineering* (2022), 1–52 (see page 1).

[3] M Nadeem Ahangar, Qasim Z Ahmed, Fahd A Khan, and Maryam Hafeez. **A Survey of Autonomous Vehicles: Enabling Communication Technologies and Challenges**. *Sensors* 21:3 (2021), 706 (see pages 12, 15, 23, 26–28, 92, 103).

[4] Naif Alshammari, Samet Akcay, and Toby P Breckon. **On the Impact of Illumination-invariant Image Pre-transformation for Contemporary Automotive Semantic Scene Understanding**. In: *Intelligent Vehicles Symposium*. IEEE. 2018, 1027–1032 (see page 103).

[5] Md Adnan Arefeen, Sumaiya Tabassum Nimi, Md Yusuf Sarwar Uddin, and Zhu Li. **A Lightweight ReLU-based Feature Fusion for Aerial Scene Classification**. In: *International Conference on Image Processing*. IEEE. 2021, 3857–3861 (see page 7).

[6] Federica Arrigoni, Beatrice Rossi, and Andrea Fusiello. **Spectral Synchronization of Multiple Views in SE(3)**. *Society for Industrial and Applied Mathematics Journal on Imaging Sciences* 9:4 (2016), 1963–1990 (see pages 47, 55–57, 82, 98, 103).

[7] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. **SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation**. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2017) (see pages 5, 9, 31).

[8] Bhakti Baheti, Shubham Innani, Suhas Gajre, and Sanjay Talbar. **Eff-UNet: A Novel Architecture for Semantic Segmentation in Unstructured Environment**. In: *Computer Vision and Pattern Recognition Conference Workshops*. 2020, 358–359 (see page 8).

[9] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. **BEiT: BERT Pre-Training of Image Transformers**. In: *International Conference on Learning Representations*. 2022 (see pages 9, 92, 94).

[10] Ana M Barragán-Montero, Melissa Thomas, Gilles Defraene, Steven Michiels, Karin Haustermans, John A Lee, and Edmond Sterpin. **Deep Learning Dose Prediction for IMRT of Esophageal Cancer: The Effect of Data Quality and Quantity on Model Performance**. *Physica Medica* 83 (2021), 52–63 (see page 104).

[11] Lidia María Belmonte, Rafael Morales, and Antonio Fernández-Caballero. **Computer Vision in Autonomous Unmanned Aerial Vehicles—A Systematic Mapping Study**. *Applied Sciences* 9:15 (2019), 3196 (see pages 7, 103).

[12] Paul Bergmann, Sindy Löwe, Michael Fauser, David Sattlegger, and Carsten Steger. **Improving Unsupervised Defect Segmentation by Applying Structural Similarity to Autoencoders**. In: *14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*. INSTICC. SciTePress, 2019, 372–380. ISBN: 978-989-758-354-4. DOI: 10.5220/0007364503720380 (see page 101).

[13] Jiawang Bian, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Shen, Ming-Ming Cheng, and Ian Reid. **Unsupervised Scale-consistent Depth and Ego-motion Learning from Monocular Video**. *Advances in Neural Information Processing Systems* 32 (2019), 35–45 (see pages 2, 13, 16, 26, 28, 47, 58, 59, 84, 89, 92–96).

[14] Pia Bideau, Aruni RoyChowdhury, Rakesh R Menon, and Erik Learned-Miller. **The Best of Both Worlds: Combining CNNs and Geometric Constraints for Hierarchical Motion Segmentation**. In: *Computer Vision and Pattern Recognition Conference*. 2018, 508–517 (see page 10).

[15] Benjamin Biggs, Oliver Boyne, James Charles, Andrew Fitzgibbon, and Roberto Cipolla. **Who Left the Dogs Out? 3D Animal Reconstruction with Expectation Maximization in the Loop**. In: *European Conference on Computer Vision*. Springer. 2020, 195–211 (see pages 25, 27, 29, 93).

[16] Shubhankar Borse, Ying Wang, Yizhe Zhang, and Fatih Porikli. **Inverseform: A Loss Function for Structured Boundary-aware Segmentation**. In: *Computer Vision and Pattern Recognition Conference*. 2021, 5901–5911 (see pages 9, 92, 94).

[17] Michael M Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. **Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges**. *arXiv preprint arXiv:2104.13478* (2021) (see page 13).

[18] Alvaro Parra Bustos, Tat-Jun Chin, Anders Eriksson, and Ian Reid. **Visual SLAM: Why Bundle Adjust?** In: *International Conference on Robotics and Automation*. IEEE. 2019, 2385–2391 (see page 71).

[19] Yingfeng Cai, Lei Dai, Hai Wang, and Zhixiong Li. **Multi-target Pan-class Intrinsic Relevance Driven Model for Improving Semantic Segmentation in Autonomous Driving**. *IEEE Transactions on Image Processing* 30 (2021), 9069–9084 (see pages 1, 25, 102).

[20] Tiziana Campisi, Alessandro Severino, Muhammad Ahmad Al-Rashid, and Giovanni Pau. **The Development of the Smart Cities in the Connected and Autonomous Vehicles (CAVs) Era: From Mobility Patterns to Scaling in Cities**. *Infrastructures* 6:7 (2021), 100 (see pages 27, 28, 103).

[21] Jie Cao, Luanxuan Hou, Ming-Hsuan Yang, Ran He, and Zhenan Sun. **ReMix: Towards Image-to-Image Translation With Limited Data**. In: *Computer Vision and Pattern Recognition Conference*. June 2021, 15018–15027 (see page 104).

[22] Si-Yuan Cao, Jianxin Hu, Zehua Sheng, and Hui-Liang Shen. **Iterative Deep Homography Estimation**. In: *Computer Vision and Pattern Recognition Conference*. 2022, 1879–1888 (see pages 21, 23, 27, 28, 95).

[23] Juan Carrillo and Katherine Borda. **Recent Advances in Artificial Intelligence and Computer Vision for Unmanned Aerial Vehicles**. In: *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*. IEEE. 2021, 7959–7962 (see pages 7, 103).

[24] Vincent Casser, Soeren Pirk, Reza Mahjourian, and Anelia Angelova. **Depth Prediction Without the Sensors: Leveraging Structure for Unsupervised Learning from Monocular Videos**. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 01. 2019, 8001–8008 (see page 16).

[25] Zhenhua Chai, Zhenan Sun, Heydi Mendez-Vazquez, Ran He, and Tieniu Tan. **Gabor Ordinal Measures for Face Recognition**. *IEEE Transactions on Information Forensics and Security* 9:1 (2013), 14–26 (see pages 11, 25, 94).

[26] Jianchun Chen, Lingjing Wang, Xiang Li, and Yi Fang. **Arbicon-Net: Arbitrary Continuous Geometric Transformation Networks for Image Registration**. *Advances in Neural Information Processing Systems* 32 (2019) (see page 102).

[27] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. **Deeplab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs**. *Transactions on Pattern Analysis and Machine Intelligence* 40:4 (2017), 834–848 (see page 25).

[28] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. **Rethinking Atrous Convolution for Semantic Image Segmentation**. *arXiv preprint arXiv:1706.05587* (2017) (see page 9).

[29] Liang-Chieh Chen, Huiyu Wang, and Siyuan Qiao. **Scaling Wide Residual Networks for Panoptic Segmentation**. *arXiv preprint arXiv:2011.11675* (2020) (see pages 9, 92, 94).

[30] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. **Encoder-decoder with Atrous Separable Convolution for Semantic Image Segmentation**. In: *European Conference on Computer Vision*. 2018, 801–818 (see pages 8, 33).

[31] Shixing Chen, Caojin Zhang, Ming Dong, Jialiang Le, and Mike Rao. **Using Ranking-CNN for Age Estimation**. In: *Computer Vision and Pattern Recognition Conference*. 2017, 5183–5192 (see pages 11, 25).

[32] Yifu Chen, Arnaud Dapogny, and Matthieu Cord. **SEMEDA: Enhancing Segmentation Precision with Semantic Edge Aware Loss**. *Pattern Recognition* 108 (2020), 107557 (see page 101).

[33] Zhe Chen, Yuchen Duan, Wenhai Wang, Junjun He, Tong Lu, Jifeng Dai, and Yu Qiao. **Vision Transformer Adapter for Dense Predictions**. *arXiv preprint arXiv:2205.08534* (2022) (see pages 9, 10, 94).

[34] Zhe Chen, Xiaofeng Zhang, Yaojun Ou, and Mei Wang. **Loop Closure Detection Based on Siamese ConvNet Features and Geometrical Verification for Visual SLAM**. In: *International Conference on Artificial Neural Networks*. Springer. 2022, 719–730 (see page 103).

[35] Bowen Cheng, Alex Schwing, and Alexander Kirillov. **Per-pixel Classification is Not All You Need for Semantic Segmentation**. *Advances in Neural Information Processing Systems* 34 (2021), 17864–17875 (see pages 9, 92, 94).

[36] Chih-Chung Chou and Cheng-Fu Chou. **Efficient and Accurate Tightly-Coupled Visual-Lidar SLAM**. *IEEE Transactions on Intelligent Transportation Systems* (2021) (see pages 1, 15).

[37] Sammy Christen, Muhammed Kocabas, Emre Aksan, Jemin Hwangbo, Jie Song, and Otmar Hilliges. **D-Grasp: Physically Plausible Dynamic Grasp Synthesis for Hand-Object Interactions**. In: *Computer Vision and Pattern Recognition Conference*. 2022, 20577–20586 (see page 6).

[38] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. **The Cityscapes Dataset for Semantic Urban Scene Understanding**. In: *Computer Vision and Pattern Recognition Conference*. 2016, 3213–3223 (see pages 1, 10, 31, 33, 98).

[39] Enric Corona, Gerard Pons-Moll, Guillem Alenyà, and Francesc Moreno-Noguer. **Learned Vertex Descent: A New Direction for 3D Human Model Fitting**. *arXiv preprint arXiv:2205.06254* (2022) (see pages 25, 27).

[40] Igor Cvišić, Ivan Marković, and Ivan Petrović. **SOFT2: Stereo Visual Odometry for Road Vehicles Based on a Point-to-Epipolar-Line Metric**. *IEEE Transactions on Robotics* (2022) (see pages 1, 15).

[41] Igor Cvišić, Ivan Marković, and Ivan Petrović. **SOFT2: Stereo Visual Odometry for Road Vehicles Based on a Point-to-Epipolar-Line Metric**. *Transactions on Robotics* (2022) (see page 103).

[42] Qi Dang, Jianqin Yin, Bin Wang, and Wenqing Zheng. **Deep Learning based 2D Human Pose Estimation: A Survey**. *Tsinghua Science and Technology* 24:6 (2019), 663–676 (see page 29).

[43] *decomposeHomographyMat*. https://docs.opencv.org/3.4/d9/d0c/group__calib3d.html#ga7f60bdff78833d1e3fd6d9d0fd538d92. Accessed: 03-07-2022 (see page 80).

[44] *Deep Learning Is Not Good Enough, We Need Bayesian Deep Learning for Safe AI*. https://alexgkendall.com/computer_vision/bayesian_deep_learning_for_safe_ai/. Accessed: 03-10-22 (see page 2).

[45] Jia Deng, Nan Ding, Yangqing Jia, Andrea Frome, Kevin Murphy, Samy Bengio, Yuan Li, Hartmut Neven, and Hartwig Adam. **Large-scale Object Classification Using Label Relation Graphs**. In: *European Conference on Computer Vision*. 2014, 48–64 (see pages 10, 25, 92).

[46] Mo Deng, Alexandre Goy, Shuai Li, Kwabena Arthur, and George Barbastathis. **Probing Shallower: Perceptual Loss Trained Phase Extraction Neural Network (PLT-PhENN) for Artifact-free Reconstruction at Low Photon Budget**. *Optics Express* 28:2 (2020), 2511–2535 (see pages 24, 27).

[47] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. **Deep Image Homography Estimation**. *arXiv preprint arXiv:1606.03798* (2016) (see pages 20, 27).

[48] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. **Superpoint: Self-supervised Interest Point Detection and Description**. In: *Computer Vision and Pattern Recognition Workshops*. 2018, 224–236 (see page 20).

[49] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. **Bert: Pre-training of Deep Bidirectional Transformers for Language Understanding**. *arXiv preprint arXiv:1810.04805* (2018) (see page 5).

[50] Soumyadeep Dey and Pratik Jawanpuria. **Light-Weight Document Image Cleanup Using Perceptual Loss**. In: *International Conference on Document Analysis and Recognition*. Springer. 2021, 238–253 (see pages 24, 27).

[51] Tom van Dijk and Guido de Croon. **How Do Neural Networks See Depth in Single Images?** In: *International Conference on Computer Vision*. 2019, 2183–2191 (see pages 26, 28, 94).

[52] Haoye Dong, Xiaodan Liang, Xiaohui Shen, Bochao Wang, Hanjiang Lai, Jia Zhu, Zhiting Hu, and Jian Yin. **Towards Multi-pose Guided Virtual Try-on Network**. In: *International Conference on Computer Vision*. 2019, 9026–9035 (see pages 23, 26).

[53] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. **An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale**. In: *International Conference on Learning Representations*. 2021 (see page 5).

[54] Jiafei Duan, Arijit Dasgupta, Jason Fischer, and Cheston Tan. **A Survey on Machine Learning Approaches for Modelling Intuitive Physics**. *arXiv preprint arXiv:2202.06481* (2022) (see page 7).

[55] Martin Engelcke, Dushyant Rao, Dominic Zeng Wang, Chi Hay Tong, and Ingmar Posner. **Vote3deep: Fast Object Detection in 3D Point Clouds using Efficient Convolutional Neural Networks**. In: *International Conference on Robotics and Automation*. IEEE. 2017, 1355–1361 (see page 8).

[56] Farzan Erlik Nowruzi, Robert Laganiere, and Nathalie Japkowicz. **Homography Estimation from Image Pairs with Hierarchical Convolutional Networks**. In: *International Conference on Computer Vision Workshops*. 2017, 913–920 (see pages 20, 22, 23, 27, 28).

[57] *Exploring the KITTI 3D Object Detection Dataset*. https://medium.com/@desjoerdhaan/kitti-3d-object-detection-data-set-ef8ee6409574. Accessed: 03-10-22 (see page 2).

[58] Jianping Fan, Tianyi Zhao, Zhenzhong Kuang, Yu Zheng, Ji Zhang, Jun Yu, and Jinye Peng. **HD-MTL: Hierarchical Deep Multi-task Learning for Large-scale Visual Recognition**. *IEEE Transactions on Image Processing* 26:4 (2017), 1923–1938 (see pages 11, 25, 92).

[59] Tuo Feng and Dongbing Gu. **SGANVO: Unsupervised Deep Visual Odometry and Depth Estimation with Stacked Generative Adversarial Networks**. *IEEE Robotics and Automation Letters* 4:4 (2019), 4431–4437 (see page 16).

[60] Nicholas Fiorentini, Mehdi Maboudi, Pietro Leandri, and Massimo Losa. **Can Machine Learning and PS-InSAR Reliably Stand In for Road Profilometric Surveys?** *Sensors* 21:10 (2021), 3377 (see page 1).

[61] Martin A Fischler and Robert C Bolles. **Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography**. *Communications of the ACM* 24:6 (1981), 381–395 (see pages 20, 55, 72, 73, 78, 79, 89, 90, 93).

[62] Rasha Friji, Hassen Drira, Faten Chaieb, Hamza Kchok, and Sebastian Kurtek. **Geometric Deep Neural Network using Rigid and Non-Rigid Transformations for Human Action Recognition**. In: *International Conference on Computer Vision*. 2021, 12611–12620 (see page 102).

[63] Guillermo Gallego, Tobi Delbrück, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew J Davison, Jörg Conradt, Kostas Daniilidis, et al. **Event-based Vision: A Survey**. *Transactions on Pattern Analysis and Machine Intelligence* 44:1 (2020), 154–180 (see page 8).

[64] Aditya Ganeshan, Alexis Vallet, Yasunori Kudo, Shin-ichi Maeda, Tommi Kerola, Rares Ambrus, Dennis Park, and Adrien Gaidon. **Warp-Refine Propagation: Semi-Supervised Auto-labeling via Cycle-consistency**. In: *International Conference on Computer Vision*. 2021, 15499–15509 (see pages 1, 9, 92, 94, 102).

[65] Ravi Garg, Vijay Kumar Bg, Gustavo Carneiro, and Ian Reid. **Unsupervised CNN for Single View Depth Estimation: Geometry to the Rescue**. In: *European Conference on Computer Vision*. Springer. 2016, 740–756 (see page 18).

[66] Andreas Geiger, Philip Lenz, and Raquel Urtasun. **Are We Ready for Autonomous Driving? the KITTI Vision Benchmark Suite**. In: *Computer Vision and Pattern Recognition Conference*. IEEE. 2012, 3354–3361 (see pages 1, 2, 8, 10, 15, 16, 22, 31, 33, 47, 51, 57, 58, 66, 68, 83, 92–94, 97, 98).

[67] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. **Unsupervised Monocular Depth Estimation With Left-right Consistency**. In: *Computer Vision and Pattern Recognition Conference*. 2017, 270–279 (see pages 13, 22, 26).

[68] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J Brostow. **Digging Into Self-supervised Monocular Depth Estimation**. In: *Computer Vision and Pattern Recognition Conference*. 2019, 3828–3838 (see pages 2, 15–17, 19, 22, 24, 26–28, 31, 45, 47, 58–60, 67, 83, 84, 89, 92, 94, 95).

[69] Ariel Gordon, Hanhan Li, Rico Jonschkowski, and Anelia Angelova. **Depth From Videos in the Wild: Unsupervised Monocular Depth Learning from Unknown Cameras**. In: *International Conference on Computer Vision*. 2019, 8977–8986 (see pages 16, 17).

[70] Mark S Graham, Carole H Sudre, Thomas Varsavsky, Petru-Daniel Tudosiu, Parashkev Nachev, Sebastien Ourselin, and M Jorge Cardoso. "Hierarchical Brain Parcellation with Uncertainty." In: *Uncertainty for Safe Utilization of Machine Learning in Medical Imaging, and Graphs in Biomedical Image Analysis*. Springer, 2020, 23–31 (see page 12).

[71] Umut Güçlü, Yağmur Güçlütürk, Meysam Madadi, Sergio Escalera, Xavier Baró, Jordi González, Rob van Lier, and Marcel AJ van Gerven. **End-to-end Semantic Face Segmentation With Conditional Random Fields as Convolutional, Recurrent and Adversarial Networks**. *arXiv preprint arXiv:1703.03305* (2017) (see pages 8, 9).

[72] Yanming Guo, Yu Liu, Ard Oerlemans, Songyang Lao, Song Wu, and Michael S Lew. **Deep Learning for Visual Understanding: A Review**. *Neurocomputing* 187 (2016), 27–48 (see page 31).

[73] Abhishek Gupta, Alagan Anpalagan, Ling Guan, and Ahmed Shaharyar Khwaja. **Deep Learning for Object Detection and Scene Perception in Self-driving Cars: Survey, Challenges, and Open Issues**. *Array* 10 (2021), 100057 (see pages 8, 15, 16).

[74] Shijie Hao, Yuan Zhou, and Yanrong Guo. **A Brief survey on Semantic Segmentation with Deep Learning**. *Neurocomputing* 406 (2020), 302–321 (see pages 9, 101).

[75] Richard Hartley and Andrew Zisserman. **Multiple View Geometry in Computer Vision**. Cambridge university press, 2003 (see pages 20, 23, 90).

[76] Michael Hauser and Asok Ray. **Principles of Riemannian Geometry in Neural Networks**. *Advances in Neural Information Processing Systems* 30 (2017) (see page 13).

[77] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. **Mask R-CNN**. In: *International Conference on Computer Vision*. 2017, 2961–2969 (see page 9).

[78] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. **Deep Residual Learning for Image Recognition**. In: *Computer Vision and Pattern Recognition Conference*. 2016, 770–778 (see pages 5, 53, 71).

[79] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. **Delving Deep into Rectifiers: Surpassing Human-level Performance on Imagenet Classification**. In: *International Conference on Computer Vision*. 2015, 1026–1034 (see page 38).

[80] Dorian F Henning, Tristan Laidlow, and Stefan Leutenegger. **BodySLAM: Joint Camera Localisation, Mapping, and Human Motion Tracking**. In: *European Conference on Computer Vision*. Springer. 2022, 656–673 (see page 25).

[81] Marc Hesenius, Ingo Börsting, Ole Meyer, and Volker Gruhn. **Don't Panic! Guiding Pedestrians in Autonomous Traffic with Augmented Reality**. In: *International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct*. 2018, 261–268 (see pages 27, 103).

[82] Mingbo Hong, Yuhang Lu, Nianjin Ye, Chunyu Lin, Qijun Zhao, and Shuaicheng Liu. **Unsupervised Homography Estimation with Coplanarity-Aware GAN**. In: *Computer Vision and Pattern Recognition Conference*. 2022, 17663–17672 (see pages 21, 23, 27).

[83] Yining Hong, Kaichun Mo, Li Yi, Leonidas J Guibas, Antonio Torralba, Joshua B Tenenbaum, and Chuang Gan. **Fixing Malfunctional Objects with Learned Physical Simulation and Functional Prediction**. In: *Computer Vision and Pattern Recognition Conference*. 2022, 1413–1423 (see page 6).

[84] Shell Xu Hu, Da Li, Jan Stühmer, Minyoung Kim, and Timothy M. Hospedales. **Pushing the Limits of Simple Pipelines for Few-Shot Learning: External Data and Fine-Tuning Make a Difference**. In: *Computer Vision and Pattern Recognition Conference*. June 2022, 9068–9077 (see page 104).

[85] Xiaobin Hu, Hongwei Li, Yu Zhao, Chao Dong, Bjoern H Menze, and Marie Piraud. **Hierarchical Multi-class Segmentation of Glioma Images using Networks with Multi-level Activation Function**. In: *International MICCAI Brainlesion Workshop*. Springer. 2018, 116–127 (see page 11).

[86] Yukun Huang, Zheng-Jun Zha, Xueyang Fu, and Wei Zhang. **Illumination-invariant Person Re-identification**. In: *ACM International Conference on Multimedia*. 2019, 365–373 (see pages 24, 27).

[87] Jessica Hullman, Sayash Kapoor, Priyanka Nanayakkara, Andrew Gelman, and Arvind Narayanan. **The Worst of Both Worlds: A Comparative Analysis of Errors in Learning from Data in Psychology and Machine Learning**. *arXiv preprint arXiv:2203.06498* (2022) (see pages 5, 71, 89, 93).

[88] Manuel José Ibarra-Arenado, Tardi Tjahjadi, and Juan Pérez-Oria. **Shadow Detection in Still Road Images Using Chrominance Properties of Shadows and Spectral Power Distribution of the Illumination**. *Sensors* 20:4 (2020), 1012 (see page 103).

[89] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. **Flownet 2.0: Evolution of Optical Flow Estimation with Deep Networks**. In: *Computer Vision and Pattern Recognition Conference*. 2017, 2462–2470 (see pages 73, 74, 83, 90, 93, 101, 102).

[90] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. **Spatial Transformer Networks**. *Advances in Neural Information Processing Systems* 28 (2015), 2017–2025 (see pages 17, 51).

[91] Joel Janai, Fatma Güney, Aseem Behl, Andreas Geiger, et al. **Computer Vision for Autonomous Vehicles: Problems, Datasets and State of the Art**. *Foundations and Trends® in Computer Graphics and Vision* 12:1–3 (2020), 1–308 (see pages 7–10, 12, 15, 17, 26–28, 92, 103).

[92] Jisoo Jeong, Jamie Menjay Lin, Fatih Porikli, and Nojun Kwak. **Imposing Consistency for Optical Flow Estimation**. In: *Computer Vision and Pattern Recognition Conference*. 2022, 3181–3191 (see page 102).

[93] Shaocheng Jia, Xin Pei, Wei Yao, and SC Wong. **Self-supervised Depth Estimation Leveraging Global Perception and Geometric Smoothness using On-board Videos**. *arXiv preprint arXiv:2106.03505* (2021) (see pages 23, 26).

[94] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. **Perceptual Losses for Real-time Style Transfer and Super-resolution**. In: *European Conference on Computer Vision*. Springer. 2016, 694–711 (see pages 24, 27, 51, 67, 92, 96).

[95] Justin M Johnson and Taghi M Khoshgoftaar. **Survey on Deep Learning with Class Imbalance**. *Journal of Big Data* 6:1 (2019), 1–54 (see pages 11, 101).

[96] Andrew Kae, Kihyuk Sohn, Honglak Lee, and Erik Learned-Miller. **Augmenting CRFs with Boltzmann Machine Shape Priors for Image Labeling**. In: *Computer Vision and Pattern Recognition Conference*. 2013, 2019–2026 (see page 31).

[97] Abdulla Al-Kaff, David Martin, Fernando Garcia, Arturo de la Escalera, and José María Armingol. **Survey of Computer Vision Algorithms and Applications for Unmanned Aerial Vehicles**. *Expert Systems with Applications* 92 (2018), 447–463 (see pages 7, 103).

[98] Iman Abaspur Kazerouni, Luke Fitzgerald, Gerard Dooly, and Daniel Toal. **A Survey of State-of-the-Art on Visual SLAM**. *Expert Systems with Applications* (2022), 117734 (see pages 10, 12, 15, 19, 23–25, 28, 103).

[99] Alex Kendall, Matthew Grimes, and Roberto Cipolla. **PoseNet: A Convolutional Network for Real-time 6-DOF Camera Relocalization**. In: *International Conference on Computer Vision*. 2015, 2938–2946 (see pages 14, 26, 31).

[100] Alex Kendall, Hayk Martirosyan, Saumitro Dasgupta, Peter Henry, Ryan Kennedy, Abraham Bachrach, and Adam Bry. **End-to-end Learning of Geometry and Context for Deep Stereo Regression**. In: *International Conference on Computer Vision*. 2017, 66–75 (see pages 14, 15).

[101] Daewoon Kim and Kwanghee Ko. **Camera Localization with Siamese Neural Networks using Iterative Relative Pose Estimation**. *Journal of Computational Design and Engineering* 9:4 (2022), 1482–1497 (see pages 22, 23, 26, 28, 46, 54).

[102] *KITTI Visual Odometry Benchmark*. https://www.cvlibs.net/datasets/kitti/eval_odometry.php. Accessed: 17-10-2022 (see page 66).

[103] Daniel Koguciuk, Elahe Arani, and Bahram Zonooz. **Perceptual Loss for Robust Unsupervised Homography Estimation**. In: *Computer Vision and Pattern Recognition Conference*. 2021, 4274–4283 (see pages 20, 24, 27, 28, 92, 95).

[104] Kenji Koide, Masashi Yokozuka, Shuji Oishi, and Atsuhiko Banno. **Globally Consistent 3D LiDAR Mapping with GPU-accelerated GICP Matching Cost Factors**. *IEEE Robotics and Automation Letters* 6:4 (2021), 8591–8598 (see pages 1, 15).

[105] Nikos Kolotouros, Georgios Pavlakos, Michael J Black, and Kostas Daniilidis. **Learning to Reconstruct 3D Human Pose and Shape via Model-fitting in the Loop**. In: *International Conference on Computer Vision*. 2019, 2252–2261 (see pages 25, 27, 29, 67, 81, 93).

[106] Mario Köppen. **The Curse of Dimensionality**. In: *5th Online World Conference on Soft Computing in Industrial Applications*. Vol. 1. 2000, 4–8 (see page 12).

[107] Alex Krizhevsky, Geoffrey Hinton, et al. **Learning Multiple Layers of Features from Tiny Images** (2009) (see page 10).

[108] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. **Imagenet Classification with Deep Convolutional Neural Networks**. *Advances in Neural Information Processing Systems* 25 (2012) (see page 5).

[109] Rainer Kümmerle, Bastian Steder, Christian Dornhege, Michael Ruhnke, Giorgio Grisetti, Cyrill Stachniss, and Alexander Kleiner. **On Measuring the Accuracy of SLAM Algorithms**. *Autonomous Robots* 27:4 (2009), 387–407 (see page 59).

[110] Fahad Lateef and Yassine Ruichek. **Survey on Semantic Segmentation using Deep Learning Techniques**. *Neurocomputing* 338 (2019), 321–348 (see pages 10, 101).

[111] Hoang Le, Feng Liu, Shu Zhang, and Aseem Agarwala. **Deep Homography Estimation for Dynamic Scenes**. In: *Computer Vision and Pattern Recognition Conference*. 2020, 7652–7661 (see pages 20, 27).

[112] Vuong Le, Jonathan Brandt, Zhe Lin, Lubomir Bourdev, and Thomas S Huang. **Interactive Facial Feature Localization**. In: *European Conference on Computer Vision*. 2012, 679–692 (see pages 9, 31, 33, 35, 39–42, 91).

[113] Laura Leal-Taixé, Cristian Canton-Ferrer, and Konrad Schindler. **Learning by Tracking: Siamese CNN for Robust Target Association**. In: *Computer Vision and Pattern Recognition Conference Workshops*. 2016, 33–40 (see pages 23, 26, 46, 54).

[114] Joffrey L Leevy, Taghi M Khoshgoftaar, Richard A Bauder, and Naeem Seliya. **A Survey on Addressing High-class Imbalance in Big Data**. *Journal of Big Data* 5:1 (2018), 1–30 (see page 101).

[115] Johannes Lehner, Andreas Mitterecker, Thomas Adler, Markus Hofmarcher, Bernhard Nessler, and Sepp Hochreiter. **Patch Refinement–Localized 3D Object Detection**. *arXiv preprint arXiv:1910.04093* (2019) (see page 8).

[116] Kenneth Levenberg. **A Method for the Solution of Certain Non-linear Problems in Least Squares**. *Quarterly of Applied Mathematics* 2:2 (1944), 164–168 (see page 79).

[117] Feng Li, Hao Zhang, Shilong Liu, Lei Zhang, Lionel M Ni, Heung-Yeung Shum, et al. **Mask DINO: Towards A Unified Transformer-based Framework for Object Detection and Segmentation**. *arXiv preprint arXiv:2206.02777* (2022) (see pages 9, 94).

[118] Guanbin Li and Yizhou Yu. **Deep Contrast Learning for Salient Object Detection**. In: *Computer Vision and Pattern Recognition Conference*. 2016, 478–487 (see page 8).

[119] Ning Li, Yongqiang Zhao, Quan Pan, Seong G Kong, and Jonathan Cheung-Wai Chan. **Illumination-invariant Road Detection and Tracking using LWIR Polarization Characteristics**. *Journal of Photogrammetry and Remote Sensing* 180 (2021), 357–369 (see page 103).

[120] Peiran Li, Haoran Zhang, Zhiling Guo, Suxing Lyu, Jinyu Chen, Wenjing Li, Xuan Song, Ryosuke Shibasaki, and Jinyue Yan. **Understanding Rooftop PV Panel Semantic Segmentation of Satellite and Aerial Images for Better using Machine Learning**. *Advances in Applied Energy* 4 (2021), 100057 (see page 1).

[121] Ruihao Li, Sen Wang, Zhiqiang Long, and Dongbing Gu. **UnDeepVO: Monocular Visual Odometry through Unsupervised Deep Learning**. In: *IEEE International Conference on Robotics and Automation*. IEEE. 2018, 7286–7291 (see pages 1, 16, 26).

[122] Xiaomeng Li, Hao Chen, Xiaojuan Qi, Qi Dou, Chi-Wing Fu, and Pheng-Ann Heng. **H-DenseUNet: Hybrid Densely Connected UNet for Liver and Tumor Segmentation from CT Volumes**. *IEEE Transactions on Medical Imaging* 37:12 (2018), 2663–2674 (see page 8).

[123] Zhongguo Li, Magnus Oskarsson, and Anders Heyden. **3D Human Pose and Shape Estimation through Collaborative Learning and Multi-view Model-fitting**. In: *Winter Conference on Applications of Computer Vision*. 2021, 1888–1897 (see pages 25, 27, 29).

[124] Jinpeng Lin, Hao Yang, Dong Chen, Ming Zeng, Fang Wen, and Lu Yuan. **Face Parsing with RoI Tanh-Warping**. In: *Computer Vision and Pattern Recognition Conference*. 2019, 5654–5663 (see page 9).

[125] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. **Feature Pyramid Networks for Object Detection**. In: *Computer Vision and Pattern Recognition Conference*. 2017, 2117–2125 (see pages 8, 9).

[126] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. **Microsoft COCO: Common Objects in Context**. In: *European Conference on Computer Vision*. Springer. 2014, 740–755 (see pages 21, 23).

[127] Yuankai Lin, Tao Cheng, Qi Zhong, Wending Zhou, and Hua Yang. **Dynamic Spatial Propagation Network for Depth Completion**. *arXiv preprint arXiv:2202.09769* (2022) (see page 1).

[128] Haisong Liu, Tao Lu, Yihui Xu, Jia Liu, Wenjie Li, and Lijun Chen. **CamLiFlow: Bidirectional Camera-LiDAR Fusion for Joint Optical Flow and Scene Flow Estimation**. In: *Computer Vision and Pattern Recognition Conference*. 2022, 5791–5801 (see pages 1, 102).

[129] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. **Path Aggregation Network for Instance Segmentation**. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, 8759–8768 (see page 7).

[130] Xiaoyu Liu, Huachen Gao, and Xiaowen Ma. **Perceptual Losses for Self-supervised Depth Estimation**. In: *Journal of Physics: Conference Series*. Vol. 1952. 2. IOP Publishing. 2021, 022040 (see pages 24, 27, 28).

[131] Oliver Lock, Tomasz Bednarz, and Christopher Pettit. **HoloCity–Exploring the use of Augmented Reality Cityscapes for Collaborative Understanding of High-volume Urban Sensor Data**. In: *The 17th International Conference on Virtual-Reality Continuum and its Applications in Industry*. 2019, 1–2 (see page 1).

[132] Jonathan Long, Evan Shelhamer, and Trevor Darrell. **Fully Convolutional Networks for Semantic Segmentation**. In: *Computer Vision and Pattern Recognition Conference*. 2015, 3431–3440 (see page 8).

[133] Shing Yan Loo, Ali Jahani Amiri, Syamsiah Mashohor, Sai Hong Tang, and Hong Zhang. **CNN-SVO: Improving the mapping in semi-direct visual odometry using single-image depth prediction**. In: *International Conference on Robotics and Automation*. IEEE. 2019, 5218–5223 (see page 15).

[134] Andreea Roxana Luca, Tudor Florin Ursuleanu, Liliana Gheorghe, Roxana Grigorovici, Stefan Iancu, Maria Hlusneac, and Alexandru Grigorovici. **Impact of Quality, Type and Volume of Data used by Deep Learning Models in the Analysis of Medical Images**. *Informatics in Medicine Unlocked* (2022), 100911 (see page 104).

[135] Chenxu Luo, Zhenheng Yang, Peng Wang, Yang Wang, Wei Xu, Ram Nevatia, and Alan Yuille. **Every Pixel Counts++: Joint Learning of Geometry and Motion with 3D Holistic Understanding**. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42:10 (2019), 2624–2641 (see page 16).

[136] Ping Luo, Xiaogang Wang, and Xiaoou Tang. **Hierarchical Face Parsing via Deep Learning**. In: *Computer Vision and Pattern Recognition Conference*. IEEE. 2012, 2480–2487 (see pages 11, 25).

[137] Wenjie Luo, Yujia Li, Raquel Urtasun, and Richard Zemel. **Understanding the Effective Receptive Field in Deep Convolutional Neural Networks**. *Advances in Neural Information Processing Systems* 29 (2016) (see pages 22, 26, 46).

[138] Wenjie Luo, Alexander G Schwing, and Raquel Urtasun. **Efficient Deep Learning for Stereo Matching**. In: *Computer Vision and Pattern Recognition Conference*. 2016, 5695–5703 (see page 22).

[139] Amalia Luque, Alejandro Carrasco, Alejandro Martín, and Ana de Las Heras. **The Impact of Class Imbalance in Classification Performance Metrics Based on the Binary Confusion Matrix**. *Pattern Recognition* 91 (2019), 216–231 (see page 101).

[140] Will Maddern, Geoffrey Pascoe, Chris Linegar, and Paul Newman. **1 year, 1000 km: The Oxford RobotCar dataset**. *The International Journal of Robotics Research* 36:1 (2017), 3–15 (see page 8).

[141] Reza Mahjourian, Martin Wicke, and Anelia Angelova. **Unsupervised Learning of Depth and Ego-motion from Monocular Video using 3D Geometric Constraints**. In: *Computer Vision and Pattern Recognition Conference*. 2018, 5667–5675 (see pages 18, 19).

[142] Ezio Malis and Manuel Vargas. **Deeper Understanding of the Homography Decomposition for Vision-based Control**. PhD thesis. The National Institute for Research in Digital Science and Technology, 2007 (see pages 20, 21, 23, 28, 29, 79, 80, 90, 94, 95).

[143] Donald W Marquardt. **An Algorithm for Least-squares Estimation of Nonlinear Parameters**. *Journal of the Society for Industrial and Applied Mathematics* 11:2 (1963), 431–441 (see page 79).

[144] *MATLAB: Procrustes Analysis*. https://uk.mathworks.com/help/stats/procrustes.html. Accessed: 17-10-2022 (see page 66).

[145] Gellért Máttyus, Wenjie Luo, and Raquel Urtasun. **DeepRoadMapper: Extracting Road Topology from Aerial Images**. In: *International Conference on Computer Vision*. 2017, 3438–3446 (see page 7).

[146] Mostafa Mehdipour Ghazi and Mads Nielsen. **FAST-AID Brain: Fast and Accurate Segmentation Tool using Artificial Intelligence Developed for Brain**. *arXiv e-prints* (2022), arXiv–2208 (see page 12).

[147] Yiqun Mei, Pengfei Guo, and Vishal M. Patel. **Escaping Data Scarcity for High-Resolution Heterogeneous Face Hallucination**. In: *Computer Vision and Pattern Recognition Conference*. June 2022, 18676–18686 (see page 104).

[148] Iaroslav Melekhov, Juha Ylioinas, Juho Kannala, and Esa Rahtu. **Relative Camera Pose Estimation using Convolutional Neural Networks**. In: *International Conference on Advanced Concepts for Intelligent Vision Systems*. Springer. 2017, 675–687 (see pages 14, 26).

[149] Panagiotis Meletis and Gijs Dubbelman. **Training of Convolutional Networks on Multiple Heterogeneous Datasets for Street Scene Semantic Segmentation**. In: *Intelligent Vehicles*. 2018 (see pages 11, 25, 26).

[150] Mariem Mezghanni, Théo Bodrito, Malika Boulkenafed, and Maks Ovsjanikov. **Physical Simulation Layer for Accurate 3D Modeling**. In: *Computer Vision and Pattern Recognition Conference*. 2022, 13514–13523 (see page 6).

[151] Mariem Mezghanni, Malika Boulkenafed, Andre Lieutier, and Maks Ovsjanikov. **Physically-aware Generative Network for 3D Shape Modeling**. In: *Computer Vision and Pattern Recognition Conference*. 2021, 9330–9341 (see page 6).

[152] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. **NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis**. In: *European Conference on Computer Vision*. Springer. 2020, 405–421 (see page 5).

[153] Adam Millard-Ball. **Pedestrians, Autonomous Vehicles, and Cities**. *Journal of Planning Education and Research* 38:1 (2018), 6–12 (see pages 9, 26–28).

[154] Zhixiang Min, Yiding Yang, and Enrique Dunn. **Voldor: Visual Odometry from Log-logistic Dense Optical Flow Residuals**. In: *Computer Vision and Pattern Recognition Conference*. 2020, 4898–4909 (see page 15).

[155] Kaichun Mo, Shilin Zhu, Angel X Chang, Li Yi, Subarna Tripathi, Leonidas J Guibas, and Hao Su. **PartNet: A Large-scale Benchmark for Fine-grained and Hierarchical Part-level 3D Object Understanding**. In: *Computer Vision and Pattern Recognition Conference*. 2019, 909–918 (see page 6).

[156] Yujian Mo, Yan Wu, Xinneng Yang, Feilin Liu, and Yujun Liao. **Review the State-of-the-art Technologies of Semantic Segmentation Based on Deep Learning**. *Neurocomputing* 493 (2022), 626–646 (see pages 12, 31, 101).

[157] Mehdi Mohammadi and Ala Al-Fuqaha. **Enabling Cognitive Smart Cities using Big Data and Machine Learning: Approaches and Challenges**. *IEEE Communications Magazine* 56:2 (2018), 94–101 (see page 1).

[158] Taewon Moon and Jung Eek Son. **Knowledge Transfer for Adapting Pre-trained Deep Neural Models to Predict Different Greenhouse Environments based on a Low Quantity of Data**. *Computers and Electronics in Agriculture* 185 (2021), 106136 (see page 104).

[159] Bruce R Muller and William A P Smith. **A Hierarchical Loss for Semantic Segmentation**. In: *The 18th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*. 2020, 260–267 (see pages xi, 4, 12).

[160] Bruce R Muller and William A P Smith. **Self-Supervised Ground-Relative Pose Estimation**. In: *26th International Conference on Pattern Recognition*. IEEE. 2022, 3507–3513 (see pages xi, 4).

[161] Bruce R Muller and William A P Smith. **Self-Supervised Relative Pose With Homography Model-Fitting in the Loop**. In: *Winter Conference on Applications of Computer Vision*. IEEE/CVF. 2023, 5705–5714 (see pages xi, 4).

[162] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. **ORB-SLAM: A Versatile and Accurate Monocular SLAM System**. *IEEE Transactions on Robotics* 31:5 (2015), 1147–1163 (see page 71).

[163] Raul Mur-Artal and Juan D Tardós. **ORB-SLAM2: An Open-source SLAM System for Monocular, Stereo, and RGB-D Cameras**. *IEEE Transactions on Robotics* 33:5 (2017), 1255–1262 (see pages 71, 103).

[164] Calvin Murdock, Zhen Li, Howard Zhou, and Tom Duerig. **Blockout: Dynamic Model Selection for Hierarchical Deep Networks**. In: *Computer Vision and Pattern Recognition Conference*. 2016, 2583–2591 (see pages 11, 25, 92).

[165] Kevin P Murphy. **Naive Bayes Classifiers**. Tech. rep. 18. University of British Columbia, 2006 (see page 37).

[166] Shikhar Murty, Patrick Verga, Luke Vilnis, Irena Radovanovic, and Andrew McCallum. **Hierarchical Losses and New Resources for Fine-grained Entity Typing and Linking**. In: *Association for Computational Linguistics (Volume 1: Long Papers)*. 2018, 97–109 (see page 12).

[167] Sauradip Nag, Saptakatha Adak, and Sukhendu Das. **What's There in The Dark?** In: *International Conference on Image Processing*. IEEE. 2019, 2996–3000 (see page 9).

[168] Mahima Nama, Ankita Nath, Nancy Bechra, Jitendra Bhatia, Sudeep Tanwar, Manish Chaturvedi, and Balqies Sadoun. **Machine Learning-based Traffic Scheduling Techniques for Intelligent Transportation System: Opportunities and Challenges**. *International Journal of Communication Systems* 34:9 (2021), e4814 (see page 1).

[169] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulo, and Peter Kontschieder. **The Mapillary Vistas Dataset for Semantic Understanding of Street Scenes**. In: *International Conference on Computer Vision*. 2017, 4990–4999 (see pages 1, 9, 10, 12, 33–35, 39–41, 43, 91, 97).

[170] Ty Nguyen, Steven W Chen, Shreyas S Shivakumar, Camillo Jose Taylor, and Vijay Kumar. **Unsupervised Deep Homography: A Fast and Robust Homography Estimation Model**. *Robotics and Automation Letters* 3:3 (2018), 2346–2353 (see pages 20, 24, 27).

[171] Lang Nie, Chunyu Lin, Kang Liao, and Yao Zhao. **Learning Edge-Preserved Image Stitching from Multi-Scale Deep Homography**. *Neurocomputing* (2021) (see pages 20, 23, 27, 28).

[172] Qingqun Ning, Jianke Zhu, and Chun Chen. **Very Fast Semantic Image Segmentation using Hierarchical Dilation and Feature Refining**. *Cognitive Computation* 10:1 (2018), 62–72 (see pages 8, 9).

[173] Zhenxing Niu, Mo Zhou, Le Wang, Xinbo Gao, and Gang Hua. **Ordinal Regression with Multiple Output CNN for Age Estimation**. In: *Computer Vision and Pattern Recognition Conference*. 2016, 4920–4928 (see pages 11, 25, 94).

[174] Niall O'Mahony, Sean Campbell, Anderson Carvalho, Suman Harapanahalli, Gustavo Velasco Hernandez, Lenka Krpalkova, Daniel Riordan, and Joseph Walsh. **Deep Learning vs. Traditional Computer Vision**. In: *Science and Information Conference*. Springer. 2019, 128–144 (see page 5).

[175] *OpenCV: findHomography*. https://docs.opencv.org/3.4/d9/d0c/group__calib3d.html#ga4abc2ece9fab9398f2e560d53c8c9780. Accessed: 26-10-2022 (see pages 78, 79, 89).

[176] Sankar K Pal, Anima Pramanik, Jhareswar Maiti, and Pabitra Mitra. **Deep Learning in Multi-object Detection and Tracking: State of the Art**. *Applied Intelligence* 51:9 (2021), 6400–6429 (see pages 5, 8, 31).

[177] Yue Pan, Pengchuan Xiao, Yujie He, Zhenlei Shao, and Zesong Li. **MULLS: Versatile LiDAR SLAM via Multi-metric Linear Least Square**. In: *International Conference on Robotics and Automation*. IEEE. 2021, 11633–11640 (see page 1).

[178] Despoina Paschalidou, Osman Ulusoy, Carolin Schmitt, Luc Van Gool, and Andreas Geiger. **Raynet: Learning Volumetric 3D Reconstruction with Ray Potentials**. In: *Computer Vision and Pattern Recognition Conference*. 2018, 3897–3906 (see page 25).

[179] Samira Pouyanfar, Saad Sadiq, Yilin Yan, Haiman Tian, Yudong Tao, Maria Presa Reyes, Mei-Ling Shyu, Shu-Ching Chen, and Sundaraja S Iyengar. **A Survey on Deep Learning: Algorithms, Techniques, and Applications**. *ACM Computing Surveys* 51:5 (2018), 1–36 (see pages 5, 9, 15, 16, 29).

[180] F Gabriele Pratticò, Fabrizio Lamberti, Alberto Cannavò, Lia Morra, and Paolo Montuschi. **Comparing State-of-the-art and Emerging Augmented Reality Interfaces for Autonomous Vehicle-to-pedestrian Communication**. *Transactions on Vehicular Technology* 70:2 (2021), 1157–1168 (see pages 27, 28, 103).

[181] Simon JD Prince. **Computer Vision: Models, Learning, and Inference**. Cambridge University Press, 2012 (see pages 19–21, 23, 90).

[182] Hui Qu, Gregory Riedlinger, Pengxiang Wu, Qiaoying Huang, Jingru Yi, Subhajyoti De, and Dimitris Metaxas. **Joint Segmentation and Fine-grained Classification of Nuclei in Histopathology Images**. In: *16th International Symposium on Biomedical Imaging*. IEEE. 2019, 900–904 (see page 101).

[183] Mohammad Saeed Rad, Behzad Bozorgtabar, Urs-Viktor Marti, Max Basler, Hazim Kemal Ekenel, and Jean-Philippe Thiran. **SROBB: Targeted Perceptual Loss for Single Image Super-resolution**. In: *International Conference on Computer Vision*. 2019, 2710–2719 (see pages 24, 101).

[184] Filip Radenović, Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondřej Chum. **Revisiting Oxford and Paris: Large-scale Image Retrieval Benchmarking**. In: *Computer Vision and Pattern Recognition Conference*. 2018, 5706–5715 (see page 20).

[185] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. **Learning Transferable Visual Models from Natural Language Supervision**. In: *International Conference on Machine Learning*. PMLR. 2021, 8748–8763 (see page 5).

[186] Rene Ranftl, Vibhav Vineet, Qifeng Chen, and Vladlen Koltun. **Dense Monocular Depth Estimation in Complex Dynamic Scenes**. In: *Computer Vision and Pattern Recognition Conference*. 2016, 4058–4066 (see page 19).

[187] Anurag Ranjan, Varun Jampani, Lukas Balles, Kihwan Kim, Deqing Sun, Jonas Wulff, and Michael J Black. **Competitive Collaboration: Joint Unsupervised Learning of Depth, Camera Motion, Optical Flow and Motion Segmentation**. In: *Computer Vision and Pattern Recognition Conference*. 2019, 12240–12249 (see pages 16, 17, 22, 26, 28, 45, 47, 58–60, 84, 89, 92–94, 96).

[188] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. **You Only Look Once: Unified, Real-time Object Detection**. In: *Computer Vision and Pattern Recognition Conference*. 2016, 779–788 (see pages 5, 12).

[189] Joseph Redmon and Ali Farhadi. **YOLO9000: Better, Faster, Stronger**. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, 7263–7271 (see page 12).

[190] Ignacio Rocco, Relja Arandjelovic, and Josef Sivic. **Convolutional Neural Network Architecture for Geometric Matching**. In: *Computer Vision and Pattern Recognition Conference*. 2017, 6148–6157 (see pages 3, 5, 21–23, 26, 46, 50, 53–55, 67, 92, 95, 102).

[191] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. **U-Net: Convolutional Networks for Biomedical Image Segmentation**. In: *Medical Image Computing and Computer Assisted Interventions*. 2015, 234–241 (see pages 7–9, 14, 38, 92).

[192] Samuel Rota Bulò, Lorenzo Porzi, and Peter Kontschieder. **In-place Activated Batchnorm for Memory-optimized Training of DNNs**. In: *Computer Vision and Pattern Recognition Conference*. 2018, 5639–5647 (see page 9).

[193] Deboleena Roy, Priyadarshini Panda, and Kaushik Roy. **Tree-CNN: A Hierarchical Deep Convolutional Neural Network for Incremental Learning**. *Neural Networks* 121 (2020), 148–160 (see pages 11, 25).

[194] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. **ORB: An Efficient Alternative to SIFT or SURF**. In: *International Conference on Computer Vision*. Ieee. 2011, 2564–2571 (see page 20).

[195] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. **ImageNet Large Scale Visual Recognition Challenge**. *International Journal of Computer Vision* 115:3 (2015), 211–252. DOI: 10.1007/s11263-015-0816-y (see pages 5, 10, 52, 67).

[196] Paul-Edouard Sarlin, Ajaykumar Unagar, Mans Larsson, Hugo Germain, Carl Toft, Viktor Larsson, Marc Pollefeys, Vincent Lepetit, Lars Hammarstrand, Fredrik Kahl, et al. **Back to the Feature: Learning Robust Camera Localization from Pixels to Pose**. In: *Computer Vision and Pattern Recognition Conference*. 2021, 3247–3257 (see page 13).

[197] Cordelia Schmid and Roger Mohr. **Local Grayvalue Invariants for Image Retrieval**. *Transactions on Pattern Analysis and Machine Intelligence* 19:5 (1997), 530–535 (see page 55).

[198] Soumyadip Sengupta, Angjoo Kanazawa, Carlos D Castillo, and David W Jacobs. **SfSNet: Learning Shape, Reflectance and Illuminance of Faces in the Wild'**. In: *Computer Vision and Pattern Recognition Conference*. 2018, 6296–6305 (see page 5).

[199] Cheng Shi and Chi-Man Pun. **Adaptive Multi-scale Deep Neural Networks with Perceptual Loss for Panchromatic and Multispectral Images Classification**. *Information Sciences* 490 (2019), 1–17 (see pages 24, 27).

[200] Kuan-Hung Shih, Ching-Te Chiu, Jiou-Ai Lin, and Yen-Yu Bu. **Real-time Object Detection with Reduced Region Proposal Network via Multi-feature Concatenation**. *IEEE Transactions on Neural Networks and Learning Systems* 31:6 (2019), 2164–2173 (see pages 22, 26, 46).

[201] Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. **Megatron-lm: Training Multi-billion Parameter Language Models using Model Parallelism**. *arXiv preprint arXiv:1909.08053* (2019) (see page 5).

[202] Karen Simonyan and Andrew Zisserman. **Very Deep Convolutional Networks for Large-scale Image Recognition**. *arXiv preprint arXiv:1409.1556* (2014) (see pages 3, 5, 8, 20, 46, 52–54, 71, 98, 101).

[203] Gurkirt Singh, Stephen Akrigg, Manuele Di Maio, Valentina Fontana, Reza Javanmard alitappeh, Salman Khan, Suman Saha, Kossar Jeddisaravi, Farzad Yousefi, Jacob Culley, Thomas Nicholson, Jordan Omokeowa, Stanislao Grazioso, Andrew Bradley, Giuseppe Di Gironimo, and Fabio Cuzzolin. **ROAD: The Road Event Awareness Dataset for Autonomous Driving**. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2022), 1–1. DOI: 10.1109/TPAMI.2022.3150906 (see page 8).

[204] Josef Sivic and Andrew Zisserman. **Video Google: A text Retrieval Approach to Object Matching in Videos**. In: *International Conference on Computer Vision*. Vol. 3. IEEE Computer Society. 2003, 1470–1470 (see page 55).

[205] Brandon M Smith, Li Zhang, Jonathan Brandt, Zhe Lin, and Jianchao Yang. **Exemplar-based Face Parsing**. In: *Computer Vision and Pattern Recognition Conference*. 2013, 3484–3491 (see pages 31, 33, 39, 91).

[206] Nitish Srivastava and Ruslan R Salakhutdinov. **Discriminative Transfer Learning with Tree-based Priors**. In: *Advances in Neural Information Processing Systems*. 2013, 2094–2102 (see pages 11, 25).

[207] Carolin Strobl and Friedrich Leisch. **Against the "One Method Fits All Data Sets" Philosophy for Comparison Studies in Methodological Research**. *Biometrical Journal* (2022) (see pages 5, 71).

[208] Tengxiang Su, Haijiang Li, and Yi An. **A BIM and Machine Learning Integration Framework for Automated Property Valuation**. *Journal of Building Engineering* 44 (2021), 102636 (see page 1).

[209] Maryam Sultana, Arif Mahmood, Thierry Bouwmans, and Soon Ki Jung. **Dynamic Background Subtraction using Least Square Adversarial Learning**. In: *International Conference on Image Processing*. IEEE. 2020, 3204–3208 (see pages 24, 27).

[210] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. **Models Matter, so does Training: An Empirical Study of CNNs for Optical Flow Estimation**. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42:6 (2019), 1408–1423 (see page 2).

[211] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. **PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume**. In: *Computer Vision and Pattern Recognition Conference*. 2018 (see page 102).

[212] Jennifer J. Sun, Ann Kennedy, Eric Zhan, David J. Anderson, Yisong Yue, and Pietro Perona. **Task Programming: Learning Data Efficient Behavior Representations**. In: *Computer Vision and Pattern Recognition Conference*. June 2021, 2876–2885 (see page 104).

[213] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. **Going Deeper with Convolutions**. In: *Computer Vision and Pattern Recognition Conference*. 2015, 1–9 (see pages 5, 71).

[214] Supannee Tanathong, William AP Smith, and Stephen Remde. **SurfaceView: Seamless and Tile-based Orthomosaics using Millions of Street-level Images from Vehicle-mounted Cameras**. *Transactions on Intelligent Transportation Systems* (2020) (see pages 21, 79, 93).

[215] Andrew Tao, Karan Sapra, and Bryan Catanzaro. **Hierarchical Multi-scale Attention for Semantic Segmentation**. *arXiv preprint arXiv:2005.10821* (2020) (see page 10).

[216] Fadi Thabtah, Suhel Hammoud, Firuz Kamalov, and Amanda Gonsalves. **Data Imbalance in Classification: Experimental Evaluation**. *Information Sciences* 513 (2020), 429–441 (see page 101).

[217] Yurun Tian, Xin Yu, Bin Fan, Fuchao Wu, Huub Heijnen, and Vassileios Balntas. **SOSNet: Second Order Similarity Regularization for Local Descriptor Learning**. In: *Computer Vision and Pattern Recognition Conference*. 2019, 11016–11025 (see page 20).

[218] Eduardo Tieppo, Roger Robson dos Santos, Jean Paul Barddal, and Júlio Cesar Nievola. **Hierarchical Classification of Data Streams: A Systematic Literature Review**. *Artificial Intelligence Review* (2021), 1–40 (see pages 10, 11, 26, 27).

[219] Lokender Tiwari, Pan Ji, Quoc-Huy Tran, Bingbing Zhuang, Saket Anand, and Manmohan Chandraker. **Pseudo RGB-D for Self-improving Monocular Slam and Depth Prediction**. In: *European Conference on Computer Vision*. Springer. 2020, 437–455 (see pages 15–17).

[220] Aysim Toker, Qunjie Zhou, Maxim Maximov, and Laura Leal-Taixé. **Coming Down to Earth: Satellite-to-street View Synthesis for Geo-localization**. In: *Computer Vision and Pattern Recognition Conference*. 2021, 6488–6497 (see page 1).

[221] Jonathan J Tompson, Arjun Jain, Yann LeCun, and Christoph Bregler. **Joint Training of a Convolutional Network and a Graphical Model for Human Pose Estimation**. *Advances in Neural Information Processing Systems* 27 (2014) (see page 25).

[222] Shubham Tulsiani, Tinghui Zhou, Alexei A Efros, and Jitendra Malik. **Multi-view Supervision for Single-view Reconstruction via Differentiable Ray Consistency**. In: *Computer Vision and Pattern Recognition Conference*. 2017, 2626–2634 (see page 19).

[223] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. **Attention is all You Need**. *Advances in Neural Information Processing Systems* 30 (2017) (see page 5).

[224] Thomas Verelst and Tinne Tuytelaars. **SegBlocks: Block-based Dynamic Resolution Networks for Real-time Segmentation**. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2022) (see page 9).

[225] Sudheendra Vijayanarasimhan, Susanna Ricco, Cordelia Schmid, Rahul Sukthankar, and Katerina Fragkiadaki. **SfM-Net: Learning of Structure and Motion from Video**. *arXiv preprint arXiv:1704.07804* (2017) (see pages 14, 19, 26).

[226] Melissa Le-Hoa Võ. **The Meaning and Structure of Scenes**. *Vision Research* 181 (2021), 10–20 (see page 31).

[227] Anjie Wang, Zhijun Fang, Yongbin Gao, Xiaoyan Jiang, and Siwei Ma. **Depth Estimation of Video Sequences with Perceptual Losses**. *IEEE Access* 6 (2018), 30536–30546 (see pages 24, 27, 28).

[228] Anjie Wang, Zhijun Fang, Yongbin Gao, Songchao Tan, Shanshe Wang, Siwei Ma, and Jenq-Neng Hwang. **Adversarial Learning for Joint Optimization of Depth and Ego-motion**. *IEEE Transactions on Image Processing* 29 (2020), 4130–4142 (see page 16).

[229] Di Wang, Qiming Zhang, Yufei Xu, Jing Zhang, Bo Du, Dacheng Tao, and Liangpei Zhang. **Advancing Plain Vision Transformer Towards Remote Sensing Foundation Model**. *arXiv preprint arXiv:2208.03987* (2022) (see page 7).

[230] Ke Wang, Sai Ma, Fan Ren, and Jianbo Lu. **SBAS: Salient Bundle Adjustment for Visual SLAM**. *IEEE Transactions on Instrumentation and Measurement* 70 (2021), 1–9 (see pages 2, 103).

[231] Sen Wang, Ronald Clark, Hongkai Wen, and Niki Trigoni. **DeepVO: Towards End-to-end Visual Odometry with Deep Recurrent Convolutional Neural Networks**. In: *International Conference on Robotics and Automation*. IEEE. 2017, 2043–2050 (see pages 1, 14, 26).

[232] Xiaolong Wang and Abhinav Gupta. **Unsupervised Learning of Visual Representations using Videos**. In: *International Conference on Computer Vision*. 2015, 2794–2802 (see page 16).

[233] Xintao Wang, Yu Li, Honglun Zhang, and Ying Shan. **Towards Real-world Blind Face Restoration with Generative Facial Prior**. In: *Computer Vision and Pattern Recognition Conference*. 2021, 9168–9178 (see pages 24, 27).

[234] Syed Waqas Zamir, Aditya Arora, Akshita Gupta, Salman Khan, Guolei Sun, Fahad Shahbaz Khan, Fan Zhu, Ling Shao, Gui-Song Xia, and Xiang Bai. **iSAID: A Large-scale Dataset for Instance Segmentation in Aerial Images**. In: *Computer Vision and Pattern Recognition Conference Workshops*. 2019, 28–37 (see page 7).

[235] Jonatas Wehrmann, Ricardo Cerri, and Rodrigo Barros. **Hierarchical Multi-label Classification Networks**. In: *International conference on machine learning*. PMLR. 2018, 5075–5084 (see page 11).

[236] Ping Wei, Yibiao Zhao, Nanning Zheng, and Song-Chun Zhu. **Modeling 4D Human-object Interactions for Event and Object Recognition**. In: *International Conference on Computer Vision*. 2013, 3272–3279 (see page 7).

[237] Yixuan Wei, Han Hu, Zhenda Xie, Zheng Zhang, Yue Cao, Jianmin Bao, Dong Chen, and Baining Guo. **Contrastive Learning Rivals Masked Image Modeling in Fine-tuning via Feature Distillation**. *arXiv preprint arXiv:2205.14141* (2022) (see pages 9, 94).

[238] Hui Wenbin and Kamata Sei-Ichiro. **Unsupervised Learning for Stereo Depth Estimation using Efficient Correspondence Matching**. In: *5th International Conference on Advances in Image Processing*. 2021, 30–34 (see page 1).

[239] *Workshop on Autonomous Driving*. https://cvpr2022.wad.vision/. Accessed: 02-10-22 (see page 1).

[240] Hai Wu, Jinhao Deng, Chenglu Wen, Xin Li, Cheng Wang, and Jonathan Li. **CasA: A Cascade Attention Network for 3-D Object Detection From LiDAR Point Clouds**. *IEEE Transactions on Geoscience and Remote Sensing* 60 (2022), 1–11 (see page 8).

[241] Hai Wu, Qing Li, Chenglu Wen, Xin Li, Xiaoliang Fan, and Cheng Wang. **Tracklet Proposal Network for Multi-Object Tracking on Point Clouds.** In: *International Joint Conference on Artificial Intelligence*. 2021, 1165–1171 (see page 1).

[242] Xiaopei Wu, Liang Peng, Honghui Yang, Liang Xie, Chenxi Huang, Chengqi Deng, Haifeng Liu, and Deng Cai. **Sparse Fuse Dense: Towards High Quality 3D Detection with Depth Completion**. In: *Computer Vision and Pattern Recognition Conference*. 2022, 5418–5427 (see pages 1, 8).

[243] Zifeng Wu, Chunhua Shen, and Anton Van Den Hengel. **Wider or Deeper: Revisiting the Resnet Model for Visual Recognition**. *Pattern Recognition* 90 (2019), 119–133 (see page 9).

[244] Hongyi Xu, Eduard Gabriel Bazavan, Andrei Zanfir, William T Freeman, Rahul Sukthankar, and Cristian Sminchisescu. **GHUM & GHUML: Generative 3D Human Shape and Articulated Pose Models**. In: *Computer Vision and Pattern Recognition Conference*. 2020, 6184–6193 (see pages 25, 27).

[245] Weijin Xu, Huihua Yang, Mingying Zhang, Zhiwei Cao, Xipeng Pan, and Wentao Liu. **Brain Tumor Segmentation with Corner Attention and High-dimensional Perceptual Loss**. *Biomedical Signal Processing and Control* 73 (2022), 103438 (see page 101).

[246] Jian-Ru Xue, Jian-Wu Fang, and Pu Zhang. **A Survey of Scene Understanding by Event Reasoning in Autonomous Driving**. *International Journal of Automation and Computing* 15:3 (2018), 249–266 (see page 8).

[247] Zhicheng Yan, Hao Zhang, Robinson Piramuthu, Vignesh Jagadeesh, Dennis DeCoste, Wei Di, and Yizhou Yu. **HD-CNN: hierarchical deep convolutional neural networks for large scale visual recognition**. In: *International Conference on Computer Vision*. 2015, 2740–2748 (see pages 10, 25).

[248] Gengshan Yang and Deva Ramanan. **Learning to Segment Rigid Motions from Two Frames**. In: *Computer Vision and Pattern Recognition Conference*. 2021, 1266–1275 (see page 1).

[249] Honghui Yang, Zili Liu, Xiaopei Wu, Wenxiao Wang, Wei Qian, Xiaofei He, and Deng Cai. **Graph R-CNN: Towards Accurate 3D Object Detection with Semantic-Decorated Local Graph**. In: *European Conference on Computer Vision*. Springer. 2022, 662–679 (see page 8).

[250] Nan Yang, Lukas von Stumberg, Rui Wang, and Daniel Cremers. **D3VO: Deep Depth, Deep Pose and Deep Uncertainty for Monocular Visual Odometry**. In: *Computer Vision and Pattern Recognition Conference*. 2020, 1281–1292 (see pages 1, 2, 15, 95, 103).

[251] Nan Yang, Rui Wang, Jorg Stuckler, and Daniel Cremers. **Deep Virtual Stereo Odometry: Leveraging Deep Depth Prediction for Monocular Direct Sparse Odometry**. In: *European Conference on Computer Vision*. 2018, 817–833 (see page 15).

[252] Qingsong Yang, Pingkun Yan, Yanbo Zhang, Hengyong Yu, Yongyi Shi, Xuanqin Mou, Mannudeep K Kalra, Yi Zhang, Ling Sun, and Ge Wang. **Low-dose CT Image Denoising using a Generative Adversarial Network with Wasserstein Distance and Perceptual Loss**. *IEEE Transactions on Medical Imaging* 37:6 (2018), 1348–1357 (see pages 24, 27).

[253] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. **Hierarchical Attention Networks for Document Classification**. In: *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2016, 1480–1489 (see pages 11, 25, 92).

[254] Nianjin Ye, Chuan Wang, Haoqiang Fan, and Shuaicheng Liu. **Motion Basis Learning for Unsupervised Deep Homography Estimation with Subspace Projection**. In: *Computer Vision and Pattern Recognition Conference*. 2021, 13117–13125 (see pages 21, 23, 27, 28).

[255] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. **LIFT: Learned Invariant Feature Transform**. In: *European Conference on Computer Vision*. Springer. 2016, 467–483 (see page 20).

[256] Xiaochuan Yin, Xiangwei Wang, Xiaoguo Du, and Qijun Chen. **Scale Recovery for Monocular Visual Odometry using Depth Estimated with Deep Convolutional Neural Fields**. In: *International Conference on Computer Vision*. 2017, 5870–5878 (see page 15).

[257] Zhichao Yin and Jianping Shi. **GeoNet: Unsupervised Learning of Dense Depth, Optical Flow and Camera Pose**. In: *Computer Vision and Pattern Recognition Conference*. 2018, 1983–1992 (see pages 16, 19, 22, 26, 28, 47, 58–60, 84, 89, 92–94).

[258] Sungho Yoon and Ayoung Kim. **Line as a Visual Sentence: Context-Aware Line Descriptor for Visual Localization**. *Robotics and Automation Letters* 6:4 (2021), 8726–8733 (see pages 20, 27, 95).

[259] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. **Coca: Contrastive Captioners are Image-text Foundation Models**. *arXiv preprint arXiv:2205.01917* (2022) (see page 5).

[260] Lijun Yu, Yijun Qian, Wenhe Liu, and Alexander G Hauptmann. **Argus++: Robust Real-time Activity Detection for Unconstrained Video Streams with Overlapping Cube Proposals**. In: *Winter Conference on Applications of Computer Vision*. 2022, 112–121 (see page 8).

[261] Ye Yu and William AP Smith. **InverseRenderNet: Learning Single Image Inverse Rendering**. In: *Computer Vision and Pattern Recognition Conference*. 2019, 3155–3164 (see page 5).

[262] Yong Yu, Xiaosheng Si, Changhua Hu, and Jianxun Zhang. **A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures**. *Neural Computation* 31:7 (2019), 1235–1270 (see pages 16, 60).

[263] Weihao Yuan, Xiaodong Gu, Zuozhuo Dai, Siyu Zhu, and Ping Tan. **New CRFs: Neural Window Fully-connected CRFs for Monocular Depth Estimation**. *arXiv preprint arXiv:2203.01502* (2022) (see page 1).

[264] Yuhui Yuan, Xiaokang Chen, Xilin Chen, and Jingdong Wang. **Segmentation Transformer: Object-contextual Representations for Semantic Segmentation**. *European Conference on Computer Vision* (2020) (see page 10).

[265] Sergey Zagoruyko and Nikos Komodakis. **Wide Residual Networks**. *arXiv preprint arXiv:1605.07146* (2016) (see page 9).

[266] Jure Zbontar and Yann LeCun. **Computing the Stereo Matching Cost with a Convolutional Neural Network**. In: *Computer Vision and Pattern Recognition Conference*. 2015, 1592–1599 (see page 22).

[267] Jure Zbontar, Yann LeCun, et al. **Stereo Matching by Training a Convolutional Neural Network to Compare Image Patches**. *Journal of Machine Learning Research* 17:1 (2016), 2287–2318 (see page 22).

[268] Matthew D Zeiler and Rob Fergus. **Visualizing and Understanding Convolutional Networks**. In: *European Conference on Computer Vision*. Springer. 2014, 818–833 (see page 5).

[269] Rui Zeng, Simon Denman, Sridha Sridharan, and Clinton Fookes. **Rethinking Planar Homography Estimation using Perspective Fields**. In: *Asian Conference on Computer Vision*. Springer. 2018, 571–586 (see pages 20, 95).

[270] Huangying Zhan, Ravi Garg, Chamara Saroj Weerasekera, Kejie Li, Harsh Agarwal, and Ian Reid. **Unsupervised Learning of Monocular Depth Estimation and Visual Odometry with Deep Feature Reconstruction**. In: *Computer Vision and Pattern Recognition Conference*. 2018, 340–349 (see page 16).

[271] Huangying Zhan, Chamara Saroj Weerasekera, Jia-Wang Bian, and Ian Reid. **Visual Odometry Revisited: What Should be Learnt?** In: *International Conference on Robotics and Automation*. IEEE. 2020, 4203–4210 (see page 15).

[272] Jirong Zhang, Chuan Wang, Shuaicheng Liu, Lanpeng Jia, Nianjin Ye, Jue Wang, Ji Zhou, and Jian Sun. **Content-aware Unsupervised Deep Homography Estimation**. In: *European Conference on Computer Vision*. Springer. 2020, 653–669 (see pages 20, 21, 23, 27, 95).

[273] Junning Zhang, Qunxing Su, Bo Tang, Cheng Wang, and Yining Li. **DPSNet: Multitask Learning Using Geometry Reasoning for Scene Depth and Semantics**. *IEEE Transactions on Neural Networks and Learning Systems* (2021) (see page 102).

[274] Yu Zhang and Qiang Yang. **A Survey on Multi-task Learning**. *IEEE Transactions on Knowledge and Data Engineering* (2021) (see page 102).

[275] Zhengxin Zhang, Qingjie Liu, and Yunhong Wang. **Road Extraction by Deep Residual U-Net**. *Geoscience and Remote Sensing Letters* 15:5 (2018), 749–753 (see pages 7, 9).

[276] Wang Zhao, Shaohui Liu, Yezhi Shu, and Yong-Jin Liu. **Towards Better Generalization: Joint Depth-Pose Learning without PoseNet**. In: *Computer Vision and Pattern Recognition Conference*. 2020, 9151–9161 (see pages 16, 17, 22, 24, 26–28, 45–47, 58–60, 71, 83, 84, 89, 92–95).

[277] Yunhan Zhao, Shu Kong, and Charless Fowlkes. **Camera Pose Matters: Improving Depth Prediction by Mitigating Pose Distribution Bias**. In: *Computer Vision and Pattern Recognition Conference*. 2021, 15759–15768 (see pages 19, 26, 28, 99).

[278] Bo Zheng, Yibiao Zhao, Joey Yu, Katsushi Ikeuchi, and Song-Chun Zhu. **Scene Understanding by Reasoning Stability and Safety**. *International Journal of Computer Vision* 112:2 (2015), 221–238 (see page 6).

[279] Qingping Zheng, Jiankang Deng, Zheng Zhu, Ying Li, and Stefanos Zafeiriou. **Decoupled Multi-task Learning with Cyclical Self-Regulation for Face Parsing**. In: *Computer Vision and Pattern Recognition Conference*. 2022, 4156–4165 (see pages 9, 94).

[280] Wang Zhiyu, Ding Weili, and Wang Mingkui. **Illumination Invariance Adaptive Sidewalk Detection Based on Unsupervised Feature Learning**. *International Journal of Image and Graphics* (2022), 2350027 (see page 103).

[281] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. **Semantic Understanding of Scenes through the ADE20k Dataset**. *International Journal of Computer Vision* 127:3 (2019), 302–321 (see page 9).

[282] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. **Unsupervised Learning of Depth and Ego-motion from Video**. In: *Computer Vision and Pattern Recognition Conference*. 2017, 1851–1858 (see pages 13, 16–19, 22–24, 26, 27, 47, 58, 59, 84, 89, 92, 95).

[283] Tinghui Zhou, Shubham Tulsiani, Weilun Sun, Jitendra Malik, and Alexei A Efros. **View Synthesis by Appearance Flow**. In: *European Conference on Computer Vision*. Springer. 2016, 286–301 (see pages 18, 19).

[284] Xinqi Zhu and Michael Bain. **B-CNN: Branch Convolutional Neural Network for Hierarchical Classification**. *arXiv preprint arXiv:1709.09890* (2017) (see pages 10, 12, 25).

[285] Yi Zhu, Karan Sapra, Fitsum A Reda, Kevin J Shih, Shawn Newsam, Andrew Tao, and Bryan Catanzaro. **Improving Semantic Segmentation via Video Propagation and Label Relaxation**. In: *Computer Vision and Pattern Recognition Conference*. 2019, 8856–8865 (see pages 1, 74, 83, 89, 90, 93, 99, 101, 102).

[286] Daniel Zoran, Phillip Isola, Dilip Krishnan, and William T Freeman. **Learning Ordinal Relationships for Mid-level Vision**. In: *International Conference on Computer Vision*. 2015, 388–396 (see pages 11, 25, 94).

[287] Yuliang Zou, Pan Ji, Quoc-Huy Tran, Jia-Bin Huang, and Manmohan Chandraker. **Learning Monocular Visual Odometry via Self-supervised Long-term Modeling**. In: *European Conference on Computer Vision*. Springer. 2020, 710–727 (see pages 1, 2, 15, 16, 22–24, 26–28, 45, 47, 58–60, 65, 67, 83, 84, 86, 89, 92–95, 98).

[288] Yuliang Zou, Zelun Luo, and Jia-Bin Huang. **DF-Net: Unsupervised Joint Learning of Depth and Flow using Cross-task Consistency**. In: *European Conference on Computer Vision*. 2018, 36–53 (see page 16).