"Notice how the flowers grow. They do not toil or spin.
But I tell you not even Solomon in all his splendor was dressed like one of them"
(Lk 12:27)

ISBN 978-94-93197-59-6

Outdoor Image Understanding from Multiple Vision Modalities - Hoàng-Ân Lê

Outdoor Image Understanding from Multiple Vision Modalities

Hoàng-Ân Lê

Outdoor Image Understanding from Multiple Vision Modalities

ACADEMIC DISSERTATION

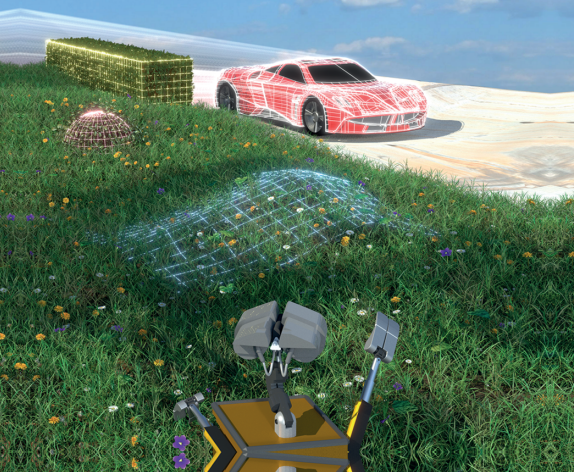
to obtain the Doctoral Degree
at the University of Amsterdam
by order of the Rector Magnificus
prof. dr. ir. K. I. J. Maex
in the presence of a committee appointed by
the Board of Doctoral Examiners,

to be publicly defended
in the Agnietenkapel
on Tuesday May 18, 2021
at 12.00 p.m.

by

Hoàng-Ân Lê

PARANYMPHS
Anıl Sırrı Başlamışlı
Ngô Lê Minh



Outdoor Image Understanding from Multiple Vision Modalities

Hoàng-Ân Lê

This dissertation was typeset by the author using LaTeX 2ε, originally developed by Leslie Lamport and based on Donald Knuth's TeX. The body text is set in 12 point Egenolff-Berner Garamond, a revival of Claude Garamont's humanist typeface.

A template that can be used to format a PhD dissertation with this look & feel has been released under the permissive AGPL license and can be found online from the author's github repository, at github.com/lhoangan/template-uva-thesis, which originates from its lead author, Jordan Suchow, at github.com/suchow/Dissertate.

ISBN 978-94-93197-59-6

Outdoor Image Understanding from Multiple Vision Modalities

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Universiteit van Amsterdam
op gezag van de Rector Magnificus
prof. dr. ir. K.I.J. Maex
ten overstaan van een door het College voor Promoties ingestelde commissie,
in het openbaar te verdedigen in de Agnietenkapel
op dinsdag 18 mei 2021, te 12.00 uur

door

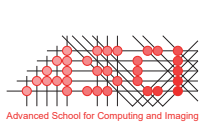
Hoàng-Ân Lê

geboren te Ho Chi Minh

Promotiecommissie

Promotores:	prof. dr. T. Gevers	Universiteit van Amsterdam
	dr. T. E. J. Mensink	Google Research
Overige leden:	prof. dr. R. B. Fisher	University of Edinburgh
	prof. dr. S. Lefèvre	Université Bretagne-Sud
	prof. dr. C. G. M. Snoek	Universiteit van Amsterdam
	dr. A. Visser	Universiteit van Amsterdam
	dr. S. Karaoğlu	Universiteit van Amsterdam

Faculteit der Natuurwetenschappen, Wiskunde en Informatica



The work described in this thesis has been carried out within the ASCI graduate school, dissertation series number 415, at the Computer Vision lab of the University of Amsterdam. The research is supported by the EU Horizon 2020 program, No.688007 (TrimBot2020).

UNIVERSITY OF AMSTERDAM

To my heavenly Father,
for the strength that keeps me standing and the hope that keeps me walking.

***

Kính tặng Ba và Mẹ,
người đã sinh thành, yêu thương và dạy con biết sống làm Người .

Contents

Fecisti nos ad te, Domine,
et inquietum est cor nostrum donec requiescat in te.

Augustine of Hippo

- Do you see anything?
Looking up he replied,
- I see people looking like trees and walking.

Mark 8:23–24

1 Introduction

SEEING IS ESSENTIAL FOR HUMAN BEINGS. Seeing, or light perceiving ability in general, is essential for most living forms. Because light is a rich source of energy and environmental information, perceiving light is advantageous for seeking favorable conditions (like food, shelter, mates, *etc*.), avoiding dangers, and subsequently crucial for survival and reproduction [1]. Thus evolution necessitates that most living organisms develop some form of light perception, from the simplest photo-pigment molecules in the primitive prokaryote bateria [2], and light-sensing cells in multi-cellular leeches, to plant phototropism [3], and visual systems of humans and animals; or even those living under low-light environments like deep-sea fish [4] and bats [5]; *etc*. Vision has become a principal sensory modality for 7 of the 36 main phyla, accounting for 96% of animal species [6], and eyes with resolving power, now exist in 10 fundamental variations [7].

Different from the primitive cells in simple organisms, which only allow recognizing light and shades, animals and especially humans' visual systems allow more sophisticated interpretations. Although an eye and a camera may share some particular traits, seeing, unlike photography, does not mean merely receiving light into eyes' photo-receptors. For humans, seeing, or, in its fullest sense *visual perception*, includes extracting information from objects' emitted light and acquiring knowledge about the environments [1]. An example is the man in the epigraph, who looks with recovering eyes and, thanks to the perceived motion, can differentiate humans from trees despite the similar perception. In other words, vision is a cognitive process for understanding the environment via optical information.

1.1 Optical Information

1

To study human visual perception, Marr's computational approach [8] has been an inspiration for different disciplines such as physiology, psychology and computer science. The approach starts with projected images on eyes' retinae and works backward to find models best describing the scenes' properties. Among the factors, light photometry and scene geometry are the main impacts on retinal images, and subsequently the scene perception [9].

Photometry We see things because there is light coming from object surfaces into our eyes. We see colors as a result of the brains' interpretation of light wavelengths. Not all objects emit light, yet they reflect, absorb, and/or transmit light. Properties of light change after interacting with object surfaces before entering into the eyes, inducing the perception of textures and materials. Red textures do not absorb red wavelengths, allowing them to travel to the eyes; matte surfaces reflect light equally in all directions; and transparent objects transmit light and bend its path; *etc*. In reality, light that comes to our eyes from a certain object is a combination of light from a primary source (*e.g*. the sun, or lamps) and light interacting with other objects. Thus, *photometric* information carries information about objects' properties. In computer vision and graphics, the light and surface interactions can be generally modelled by the bidirectional scattering distribution function [10], or in simpler forms such as the Lambertian and dichromatic reflection model [11, 12].

Geometry At the heart of retinal image formation is *projective geometry*. Light reflected from the 3D world, passing through the eye pupil and lens forms a 2D upside-down image on the retina (Figure 1.1a). Closer objects create larger images, and farther smaller. The similar mechanism has been applied in creating the camera (Figure 1.1b) and the pinhole camera model (Figure 1.1c), which forms the basis of several applications in computer graphics and vision.

Since retinal images are two-dimensional, depth information is lost after perspective projection. Humans perceive depth via the stereopsis process of the two eyes. The projected images on the retinae are close but not exactly the same due to binocular disparity of the eyes. The same point on an environmental object is thus projected on the retinae at two different locations, whose displacement depends on the distance of the objects to the eyes. Images of points that have near-zero disparity end up at very close places in the retinae of the two eyes and get fused as single images, whereas images with large disparity end up further away, creating the depth perception. Finding correspondences between two retinal images are effortless for humans, yet still an open problem in computer vision [1].

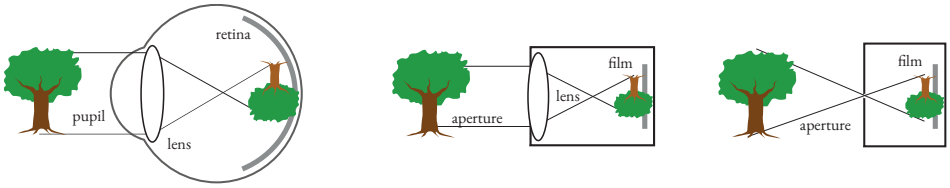


Figure 1.1: Image formation in an eye (a) compared to a camera with lens (b) and pinhole camera (c)

1.2 Cross-Modality Perception

Photometric and geometric information explain the formation of retinal images and how they induce color and distance perceptions. They are the basis from which other scene's aspects, such as motion, shape, and object identities are perceived [1].

Motion Motion perceptibility arises from depth perception, as motion is the displacement of objects over time. Although all that our brains have access are tiny changes of 2D retinal images, we can recognize a car's speed in real-world units, and understand its 3D direction. Humans perceive motion by integrating visual information over space and time: as multiple retinal images are treated as a single coherent description, we can perceive the same object when its retinal images become larger or smaller, inducing the notion of the object moving toward or away [9].

Reversely, motion also creates depth perception. Motion parallax, the difference between pairs of the same points when an object moves, is similar to binocular disparity. How fast an object moves indicates how far it is: *e.g.* airplanes on runways seem to move much faster than when they are in the sky. Computationally, if velocity and direction are known, motion parallax can provide absolute depth [13].

Motion perception is also induced by photometric information, such as color and contrast. An array of light flashing one-by-one appears to be moving, although none of the lights actually move; we see people and objects moving in movies, yet they are only static images displayed at a high-speed frame-rate. The illusory motion of stationary objects is called apparent motion. On the other hand, we tend to look for occlusions and texture boundaries to determine a moving object. Low-contrast between objects and backgrounds might pose difficulty in perceiving their motion.

Shape Similar to motion, the perception of object shape and surface orientation comes from both photometric and geometric cues. Although object shape is often attributed to a 3D representation of the entire object [1], we adopt the term to indicate both the global shape and the local structure such as the visible surface orientation.

Figure 1.2: Michelangelo's Pietà and David: object geometrical structures can be depicted in white marble statues by shading.

Photometrically, scene structures and object shapes can be depicted from shading and cast shadows. Objects' shadings are due to the variations of reflected light off a surface, resulting from the changes of the objects' geometry, while cast shadows provide relative distance between different objects. This is the reason why we can appreciate white marble statues. As shown in Figure 1.2, despite the single color, the objects' details and structures are still manifested clearly from the shading patterns.

Geometrically, as depth is perceived, objects' shapes can be perceived [14]. Objects' shapes are often indicated by their physical boundaries formed by the spatial discontinuity between the objects and the environments, *i.e.* depth edges, whereas inner structures are exhibited by the change in distance and the surface discontinuity, *i.e.* orientation edges.

Computationally, the rate of change of surface depth is defined as surface normals, which can be estimated as the gradient of depth images. The depth images can be computed from various sources, called shape-from-X, such as shading, textures, motions, *etc.* [1]

Object recognition Compared to color, depth and motion, perception of object shapes is of a higher level and provides more useful information. In that sense, object recognition is one of the highest levels of perception, as it provides the functionality and usefulness of the environmental objects. Most of perceptual theorists agree that object recognition is the intermediate step toward functionality perception [1]. In fact, the ultimate goal of the visual evolution is that humans and animals are able to choose useful objects and avoid dangerous situations.

Humans recognize by first *perceiving*, then *connecting* various intrinsic properties of objects and scenes, among which shapes are the most informative, while color, depth, and

motion are the essentials [9]. A toddler learning of the world usually grabs and tosses a toy around, then observes how it behaves in motion. The colors, shapes and the movement help identify the toy, and he can recognize it the next time it comes into sight even before knowing its name and able to speak. Adults are more skillful in recognizing objects and scenes. We look for spilt water on a textured table by tilting and finding where light reflects. Soldiers and hunters use motion cues to identify camouflaged enemies and preys from similar-looking regions. The visual impaired person in the epigraph could have taken humans for trees, if it was not for their walking. Multimodal involvement appears more critical when it comes to complicated tasks like playing sports under weather effects, where recognizing objects, tracking their location and speed, estimating winds and avoiding direct sunlight in the eyes, *etc*. need to be integrated in planning and action.

1.3 Context of the Thesis

Multimodal interaction has been recognized important by computer vision researchers. Malik *et al*. [15] regards vision as a unifying framework of three processes, namely recognition, reconstruction, and reorganization. Multinet [16] and Ubernet [17] propose to learn a universal image representation to tackle multiple vision tasks at once. Interests in multimodality also lead to multimodal datasets [18, 19, 20] being proposed, which, despite being computer-generated imagery, have proven beneficial for training multimodal deep neural networks.

In the context of robotic development, multimodality is especially important for the robot to cope with dynamic surroundings. The research of this thesis is carried in the context of the TrimBot2020 project*, which studies and develops technologies for an autonomous hedge trimmer. With the aim to operate in an outdoor unstructured environment of a garden, the trimmer should be able to identify drivable regions from various terrains and landscapes while avoiding obstacles, recognizing a target bush from similar-looking rose plants or trees, navigating among the objects and position itself at a proper location with respect to a target, as well as assessing the current and reference shape of the bush before starting to trim.

The whole operation requires high understanding of objects and awareness of the environment, thus invoking the usage of divergent vision modalities. The proposed technologies and algorithms span and connect different tasks in computer vision, such as stereo matching using depth and motion [21], semantic-based visual localization [22] and odom-

*http://trimbot2020.webhosting.rug.nl/

etry [23], jointly predicting camera poses and depth [24], as well as intrinsic image decomposition and semantic segmentation [25], *etc*.

Particularly, the outdoor garden context of TrimBot2020 is notable for intricate attributes. Outdoor lighting is a well-known challenge in computer vision for interfering effects such as color variations, specularities, backlighting, cast-shadows, *etc*. Garden scenes are generally different from the popular outdoor driving scenarios, as the scenes are unstructured with deformable and similar-looking objects (bushes, grassy mounds, plants, *etc*.). As such, besides multimodal algorithms, a multimodal large-scale dataset featuring unstructured nature scenes with dedicated semantic labels is desired as generic or popular driving datasets would become inadequate. Within this context, the next section will provide in detail the scope and research questions of the thesis.

1.4 Outline and Research Questions

Human vision primarily comes from visual stimuli, *i.e*. retinal images, captured by our visual receptors when we receive light into our eyes. In computer vision and robotics, sensory stimuli, or information obtained from sensors, are often referred to as sensory modalities, or simply modalities [26]. Different from human vision, where most of the visual perceptions, such as depth, motion, shape, *etc*. are the results of cognitive inference and only appear in our consciousness, artificial cognitive systems have definite representations for such data. In this thesis, we adopt a notion of modality that includes both sensory data, *e.g*. *RGB* and depth images, and their interpretations which cannot be captured by sensors, *e.g*. intrinsic images, surface normals, optical flow, point clouds, *etc*.

The thesis contributions are centered around the main research question:

How can various computer vision modalities be exploited and combined?

In studying multimodal approach, the reverse approach is to try decomposing a primary modality. Photometric information carries not only the color properties of environmental objects, but also geometric information such as surface structures and orientation. In Chapter 2, we study the following question:

How can images be decomposed into different photometric intrinsic components such as reflectance (texture colors) and shading (geometrical structures and lighting)?

We observe that traditional approaches relying on well-established physical models have achieved high qualitative results [27, 28, 29, 30, 31], while contemporary data-driven deep

learning approaches are quantitatively superior. In the chapter, we review the dichromatic reflection model [11], and introduce the image formation loss based on the model to steer the training process of a deep network. The principles of the long-researched Retinex method [31] is then employed to obtain intrinsic image gradient and improve the decomposition quality.

Optical flow, surface normals, and semantics are subsequent perceptions from retinal images. They depict different scene qualities, yet together they bring complementary cues for better scene understanding. In Chapter 3, we study the impact of each modality on the others and their efficiency when used in combination. The governing question is that

How do the subsequent modalities such as optical flow, object semantics, and surface normals complement and impact one another?

We employ a modular approach that separates the subsequent modalities from the primary *RGB* images to study their interactions. To that end, a deep network, pre-trained on *RGB* images for each modality, is refined with various combinations of optical flow, semantics, and surface normals to enforce joint features. The refined networks are tested on different scene types, structured and unstructured, to explore the combinatorial impacts.

There hardly exist any large-scale datasets with dense optical flow of non-rigid motion with real-world imagery as of today. The reason lies mainly in the difficulty of human annotation to generate optical flow ground truth. To circumvent the need for human annotation, in Chapter 4, we propose a framework using object segments to automatically generate optical flow from real-world videos. Chapter 4 attempts to answer the question:

How can object segments be used to generate non-rigid optical flow from real-world movies?

Object segments signify the extension of an object in a scene. By focusing on individual objects and their correspondences in each video frame, the object's motion patterns are extracted and used as constraints for motion generation. We show that dense optical flow fields, although synthetically generated, retain the objects' appearances and useful for pre-training deep neural networks for optical flow prediction.

Surface geometry defines objects' shapes and the interaction with light, hence impacts their appearances. In Chapter 5, we study the use of objects' 3D point clouds for the novel-view synthesis problem. The following question is posed for the chapter:

How does geometric and photometric information from a single view help predict the other views of an object?

We observe that object's geometry provides the basis to obtain a coarse novel view in a straightforward manner. From a partial point cloud constructed by monocular depth estimation, the pixels in the current view can be re-located or removed depending on their visibility in the target view using point cloud transformation and projection. The coarse-view completion process used to obtain the final dense view, and self-supervised training of monocular depth prediction can be formulated as backward and forward warping of input and target view, thus can be employed in an end-to-end system. The benefit of using point clouds as an explicit 3D shape for novel view synthesis is experimentally validated on the 3D ShapeNet benchmark.

Multimodal large-scale datasets for outdoor scenes are mostly designed for driving problems. The common urban scenes are highly structured and semantically different from scenarios seen in nature-centered scenes such as gardens or parks. To accommodate machine learning applications for nature-oriented scenes, in Chapter 6, we seek to address the following question:

How can a large scale dataset with multiple modalities help for unstructured outdoor scenes understanding?

We propose a synthetic dataset of Enclosed garDEN scenes (EDEN) containing more than 300K images captured from more than 100 garden models. The use of virtual garden models allows annotated data for various low- and high-level computer vision tasks, including semantic segmentation, depth, surface normal, intrinsic images, and optical flow. The dataset is used to benchmark computer vision state-of-the-art methods and show competitive results on the dataset.

1.5 Origins

This thesis is based on the following publications, in which all authors contributed to the writing process. The detailed contributions are presented following each paper:

Chapter 2 Anil S. Baslamisli, **Hoàng-Ân Lê**, Theo Gevers, "CNN based Learning using Reflection and Retinex Models for Intrinsic Image Decomposition". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (CVPR), 2018.

Anil S. Baslamisli	Analysis, methodology, implementation, and experiments
Hoàng-Ân Lê	Analysis, data generation, and experiments
Theo Gevers	Supervision, idea, and insight

Chapter 3 **Hoàng-Ân Lê**, Anil S. Baslamisli, Thomas Mensink, Theo Gevers, "Three for one and one for three: Flow, Segmentation, and Surface Normals." In: *Proceedings of the British Machine Vision Conference* (BMVC), 2018.

Hoàng-Ân Lê	Analysis, methodology, implementation, and experiments
Anil S. Baslamisli	Analysis and helping with experiment running
Thomas Mensink	Guidance and technical advice
Theo Gevers	Supervision, idea, and insight

Chapter 4 **Hoàng-Ân Lê**, Tushar Nimbhorkar, Thomas Mensink, Anil S. Baslamisli, Sezer Karaoglu, Theo Gevers, "Automatic Generation of Dense Non-Rigid Optical Flows", Under submission to *Computer Vision and Image Understanding* (CVIU), 2020.

Hoàng-Ân Lê	Analysis, methodology, implementation, and experiments
Tushar Nimbhorkar	Helping with experiment running
Thomas Mensink	Guidance and technical advice
Anil S. Baslamisli	Helping with experiment running
Sezer Karaoglu	Helping with method design
Theo Gevers	Supervision, idea, and insight

Chapter 5 **Hoàng-Ân Lê**, Thomas Mensink, Partha Das, Theo Gevers, "Novel View Synthesis from a Single Image via Point Cloud Transformation". In: *Proceedings of the British Machine Vision Conference* (BMVC), 2020.

Hoàng-Ân Lê	Analysis, methodology, implementation, and experiments
Thomas Mensink	Guidance and technical advice
Partha Das	Helping with experiment running
Theo Gevers	Supervision, idea, and insight

Chapter 6 **Hoàng-Ân Lê**, Thomas Mensink, Partha Das, Sezer Karaoglu, Theo Gevers, "EDEN: Multimodal Synthetic Dataset of Enclosed GarDEN Scenes". In: *Proceedings of the IEEE/CVF Winter Conference of Applications on Computer Vision* (WACV), 2021.

Hoàng-Ân Lê	Analysis, methodology, implementation, and experiments
Thomas Mensink	Guidance and technical advice
Partha Das	Helping with experiment running
Sezer Karaoglu	Helping with method design
Theo Gevers	Supervision, idea, and insight

The author has further contributed to the following publications:

Anil S. Baslamisli, Thomas Tiel Groenestege, Partha Das, **Hoàng-Ân Lê**, Sezer Karaoglu, Theo Gevers, "Joint Learning of Intrinsic Images and Semantic Segmentation". In: *Proceedings of the European Conference in Computer Vision* (ECCV), 2018.

Radim Tylecek, Torsten Sattler, **Hoàng-Ân Lê**, Thomas Brox, Marc Pollefeys, Robert B. Fisher, Theo Gevers, "The Second Workshop on 3D Reconstruction Meets Semantics: Challenge Results Discussion". In: *Proceedings of the European Conference in Computer Vision Workshops* (ECCVw), 2018.

Anil S. Baslamisli, Partha Das, **Hoàng-Ân Lê**, Sezer Karaoglu, Theo Gevers, "ShadingNet: Image Intrinsics by Fine-Grained Shading Decomposition". Under submission to the *International Journal of Computer Vision* (IJCV), 2020.

Jian Han, Sezer Karaoglu, **Hoàng-Ân Lê**, Theo Gevers, "Improving Face Detection Performance with 3D-Rendered Synthetic Data". In: *International Conference on Pattern Recognition* (ICPR), 2020.

2

Physics-based Deep Architectures for Intrinsic Image Decomposition

Traditional work on intrinsic image decomposition relying on physical characteristics produce high qualitative images, while deep-learning-based models dominate quantitative results. In this chapter, we propose a deep-learning-empowered method steered by the physics-based reflection models, thus achieving the best of the two worlds. The network architecture, coined RetiNet, exploits reflectance and shading gradients to obtain intrinsic images as inspired by the well-established Retinex model. The proposed approach allows for the integration of all intrinsic components. To train the new model, an object centered large-scale datasets with intrinsic ground-truth images are created. The experimental evaluations show that the new model outperforms existing methods. Visual inspection shows that the image formation loss function augments color reproduction and the use of gradient information produces sharper edges.

2.1 Introduction

Intrinsic image decomposition is the process of separating an image into its formation components such as reflectance (albedo) and shading (illumination) [27]. Reflectance is the color of the object, invariant to camera viewpoint and illumination conditions, whereas shading, dependent on camera viewpoint and object geometry, consists of different illumination effects, such as shadows, shading and inter-reflections. Using intrinsic images,

instead of the original images, can be beneficial for many computer vision algorithms. For instance, for shape-from-shading algorithms, the shading images contain important visual cues to recover geometry, while for segmentation and detection algorithms, reflectance images can be beneficial as they are independent of confounding illumination effects. Furthermore, intrinsic images are used in a wide range of computational photography applications, such as material recoloring [32, 33], relighting [34, 35], and retexturing [36, 37].

Most of the pioneering work on intrinsic image decomposition, such as [27, 28, 29, 30], rely on deriving priors about scene characteristics to understand the physical interactions of objects and lighting in a scene. In general, an optimization approach is taken imposing constraints on reflectance and shading intrinsics for a pixel-wise decomposition. [31] introduces the well-known Retinex algorithm which is based on the assumption that larger gradients in an image usually correspond to reflectance changes, whereas smaller gradients are more likely to correspond to illumination changes. In addition to the traditional work, more recent research focuses on using deep learning models [38, 39]. However, these deep learning-based methods do not consider the well-established, traditional image formation process as the basis of their intrinsic learning process. Deep learning is used as in-and-out black box, which may lead to inadequate or restricted results. Furthermore, the contribution and physical interpretation of what the network learned is often difficult to interpret. As a consequence, although current deep learning approaches show superior performance when considering quantitative benchmark results, traditional approaches are still dominant in achieving high qualitative results. Therefore, the goal of this chapter is to exploit the best of the two worlds. A method is proposed that (1) is empowered by deep learning capabilities, (2) considers a physics-based *reflection model* to steer the learning process, and (3) exploits the traditional approach to obtain intrinsic images by exploiting reflectance and shading *gradient* information.

To this end, a physics-based convolutional neural network (CNN), *IntrinsicNet*, is proposed first. A standard CNN architecture is chosen to exploit the dichromatic reflection model [11] as a standard reflection model to steer the training process by introducing a physics-based loss function called the *image formation loss*, which takes into account the reconstructed image of the predicted reflectance and shading images. The goal is to analyze the contribution of exploiting the image formation process as a constraining factor in a standard CNN architecture for intrinsic image decomposition. Then, we propose the *RetiNet*, which is a two-stage Retinex-inspired convolutional neural network which first learns to decompose (color) image gradients into intrinsic image gradients i.e. reflectance and shading gradients. Then, these intrinsic gradients are used to learn the CNN to de-

compose, at the pixel, the full image into its corresponding reflectance and shading images.

The availability of annotated large-scale datasets is key to the success of supervised deep learning methods. However, the largest publicly available dataset with intrinsic image ground-truth has around a thousand of redundant images taken from an animated cartoon-like short film [40]. Therefore, to train our CNN's, we introduce a large-scale dataset with intrinsic ground-truth images: a synthetic dataset with man-made objects. The dataset consists of around 20,000 images. Rendered with different environment maps and viewpoints, the dataset provides a variety of possible images in indoor and outdoor scenes.

In summary, our contributions are: (1) a standard CNN architecture *IntrinsicNet* incorporating the *image formation loss* derived by a physics-based reflection model, (2) a new two-stage Retinex-inspired convolutional neural network *RetiNet* exploiting *intrinsic gradients* for image decomposition at the pixel, (3) gradient (re)integration (inverse problem) where images are integrated based on intrinsic gradients by a set of simple convolutions rather than complex computations (e.g. Poisson), and (4) a large-scale synthetic object-centered dataset with intrinsic ground-truth images.

2.2 Related Work

As intrinsic image decomposition is an ill-posed problem [41, 42], an important line of research is to study scene characteristics and derive priors for reflectance and shading. An optimization procedure is used to enforce imaging constraints for pixel-wise decomposition. One of the earliest and most successful methods is the Retinex algorithm [31]. Retinex considers that the reflectance image is piece-wise constant and that the shading image varies smoothly. The algorithm assumes that larger derivatives in an image correspond to reflectance changes, and that the smaller ones correspond to illumination changes. This approach is extended to color images [43] by exploiting the chromaticity information, which is invariant to shading cues. Since then, most of the (traditional) related work continued to focus on understanding the physical interactions, geometries of the objects, and lighting cues by inferring priors. Priors that are used to constrain the inference problem are based on texture cues [44, 45], sparsity of reflectance [41, 42], user in the loop [36, 46], and depth cues [28, 47, 48]. Other methods use multiple images [30, 49, 50], where reflectance is considered as the constant factor and illumination the changing one. These methods produce promising results as they disambiguate the decomposition. However, their applicability is limited by the use of priors.

Supervised Deep Learning Deep convolutional neural networks are very success-

ful for various computer vision tasks, such as image classification [51] and object detection [52]. The success of supervised deep learning depends on the availability of annotated large-scale datasets [53, 54]. The data collection is expensive for pixel-wise annotation and more challenging for low-level such as optical flow, surface normal, intrinsic image decomposition, *etc*. For such tasks, synthetic data have proven to produce competitive performance [55, 19]. Real-world data collection for ground-truth intrinsic images is only possible in controlled laboratory settings, which require excessive effort and time. As a results, the only existing real-world imagery dataset,the MIT intrinsic benchmark [56], contains as few as 20 object-centered images.

For supervised learning, intrinsic image researchers mostly rely on synthetic datasets. The MPI-Sintel dataset [40] provides a scene-level 3D animated cartoon-like short film with intrinsic image ground-truths. Although the dataset has only around a thousand images, [38, 57] show that the synthetic images and ground-truth are useful for training deep networks. The large-scale synthetic dataset of non-Lambertian objects [58] achieves the state-of-the-art results by training an encoder-decoder CNN. Their dataset has yet been published. Relative reflectance comparison of point pairs are proposed to obtain intrinsic color for real-world indoor scenes by crowd-sourcing [59]. The dataset does not provide ground-truth intrinsic images, yet it is shown to be effective in learning priors and relationships in a data-driven manner [60, 61, 62].

Supervised deep learning, trained on large scale datasets, achieves state-of-the-art results on different benchmarks. However, they ignore physics-based characteristics of the intrinsic image formation process. Traditional methods rely on reflection models, yet do not exploit the learning power of CNNs. [38] argues that the learning model should consider both patch level information and the overall gist of the scene, while [58] assumes that the intrinsic components are highly correlated. The training data of such methods are generated in a physics-based manner, including a specular component, yet they do not explicitly embed a physics-based image formation loss. Another recent work [57] uses an image formation component in their unary term for CRF (for the optimization process, not in the learning process itself), but their training data (Sintel) was not created in a physics-based manner. Nonetheless, none of proposed deep learning methods consider the image formation process for consistent decomposition during training, nor a Retinex driven gradient separation approach [29, 41, 44, 56, 63, 64]. As Retinex has a solid background in intrinsic image decomposition, this chapter seeks to combine the best of the two worlds: supervised deep learning based on reflection and Retinex models.

2.3 Reflection-model-based Architectures

In this section, we first describe the image formation model. Then, the IntrinsicNet architecture, an encoder-decoder CNN based on the reflection model is presented with the image formation loss. Finally, we propose a Retinex-inspired new CNN architecture, RetiNet, which exploits image gradients in combination with the image formation loss.

2.3.1 Image Formation Model

The dichromatic reflection model [11] describes a surface as a composition of the body I_b (diffuse) and specular I_s (interface) reflectance:

$$I = I_b + I_s. \tag{2.1}$$

Then, the pixel value, measured over the visible spectrum ω, is expressed by:

$$I = m_b(\mathbf{n}, \mathbf{s}) \int_\omega f_c(\lambda)\, e(\lambda)\, \rho_b(\lambda)\, \mathrm{d}\lambda + m_s(\mathbf{n}, \mathbf{s}, \mathbf{v}) \int_\omega f_c(\lambda) e(\lambda) \rho_s(\lambda) \mathrm{d}\lambda, \tag{2.2}$$

where $\mathbf{n}$ is the surface normal, $\mathbf{s}$ is the light source direction, $\mathbf{v}$ is the viewing direction, m is a function of the geometric dependencies, λ is the wavelength, $f_c(\lambda)$ is the camera spectral sensitivity, $e(\lambda)$ defines the spectral power distribution of the illuminant, ρ_b characterizes the diffuse surface reflectance, and ρ_s is the specular reflectance with Fresnel reflection. Assuming a linear sensor response and narrow band filters (λ_I), Equation 2.2 can be re-written as follows:

$$I = m_b(\mathbf{n}, \mathbf{s})\ e(\lambda_I)\ \rho_b(\lambda_I)\ + m_s(\mathbf{n}, \mathbf{s}, \mathbf{v})\ e(\lambda_I)\ \rho_s(\lambda_I)\,. \tag{2.3}$$

The decomposition of an observed image $I(\mathbf{x})$ at a position $\mathbf{x}$ can be approximated from the component intrinsic images. Under the assumption of body (diffuse) reflection, Equation 2.3 can be re-written as the pixel-wise products of its reflectance $R(\mathbf{x})$ and shading $S(\mathbf{x})$ images for different light source models $e(\lambda_I)$ as follows:

$$I(\mathbf{x}) = \begin{cases} R(\mathbf{x})\ S(\mathbf{x}), & \text{for single canonical light source} \quad (2.4) \\ R(\mathbf{x})\ S(\mathbf{x})\ E(\mathbf{x}), & \text{for single non-canonical light source} \quad (2.5) \\ R(\mathbf{x})\ S(\mathbf{x})\ E, & \text{for global non-canonical light source} \quad (2.6) \end{cases}$$

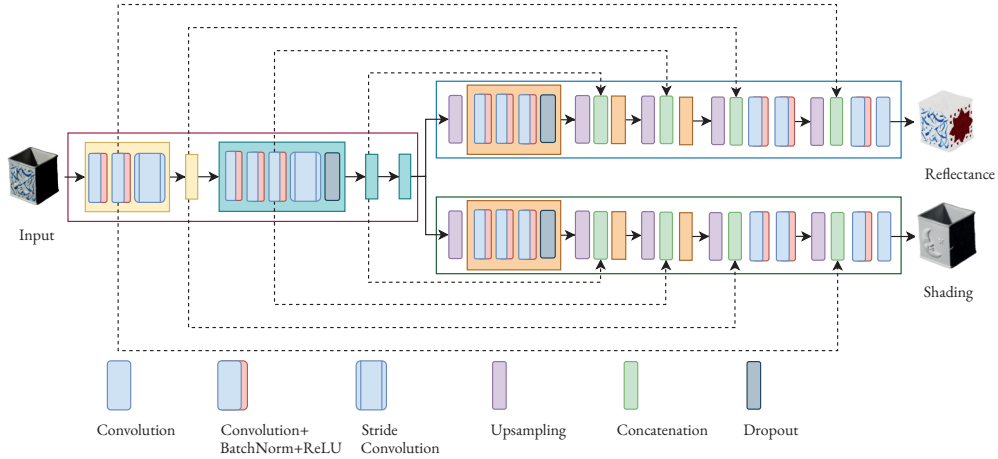


Figure 2.1: IntrinsicNet model architecture with one shared encoder and two separate decoders: one for shading and one for reflectance prediction. Encoder part contains both shading and reflectance characteristics. The decoder parts aim to disentangle those features.

where $E(\mathbf{x})$ describes the color of the light source at position $\mathbf{x}$. Equation 2.4–2.6 are extended to non-diffuse reflection by adding the specular (surface) term $H(\mathbf{x})$ as follows:

$$I(\mathbf{x}) = \begin{cases} R(\mathbf{x})\, S(\mathbf{x}) + H(\mathbf{x}), & \text{for local canonical light sources} \quad (2.7) \\ R(\mathbf{x})\, S(\mathbf{x})\, E(\mathbf{x}) + H(\mathbf{x})\, E(\mathbf{x}), & \text{for local non-canonical light sources} \quad (2.8) \\ R(\mathbf{x})\, S(\mathbf{x})\, E + H(\mathbf{x})\, E, & \text{for global non-canonical light sources} \quad (2.9) \end{cases}$$

In the next section, the reflection model is considered to introduce different image formation losses within an encoder-decoder CNN model for intrinsic image decomposition.

2.3.2 IntrinsicNet

In this section, a physics-based deep learning network, IntrinsicNet, is proposed. We use a standard CNN architecture to constrain the training process by introducing a physics-based loss and verify the benefit of constraining CNNs with the reflection model. An architecture is adopted with one shared encoder and two separate decoders, one for shading and the other for reflectance prediction. The features learned by the encoder contain both shading and reflectance cues which are disentangled by the decoder. Figure 2.1 illustrates our model. The architecture can be extended to consider more formation factors (*e.g.* light source or highlights) by adding the corresponding decoders.

To train the model, we use the standard L_2 reconstruction loss. Let $\hat{J}$ be the ground-

truth intrinsic image and J be the prediction of the network. Then, the reconstruction loss $\mathcal{L}_{RL}$ is given by:

$$\mathcal{L}_{RL}\left(J,\hat{J}\right)=\frac{1}{n}\sum_{\mathbf{x},c}\left\|\hat{J}-J\right\|_2^2, \tag{2.10}$$

where $\mathbf{x}$ denotes the image pixel, c the channel index and n is the total number of evaluated pixels. In our case, the final, combined loss $\mathcal{L}_{CL}$ is composed of 2 distinct loss functions, one for reflectance reconstruction $\mathcal{L}_{RL_R}$ and one for shading reconstruction $\mathcal{L}_{RL_S}$:

$$\mathcal{L}_{CL}\left(R,\,\hat{R},\,S,\,\hat{S}\right)=\gamma_R\,\mathcal{L}_{RL_R}\left(R,\,\hat{R}\right)\,+\gamma_S\,\mathcal{L}_{RL_S}\left(S,\,\hat{S}\right), \tag{2.11}$$

where the γs are the corresponding weights. In general, this type of network may generate color artifacts and blurry reflectance maps [39, 21]. The goal of the image formation loss is to increase the color reproduction quality because of the physics constraint.

More precisely, the image formation loss $\mathcal{L}_{IMF}$ takes into account the reconstructed image of the predicted reflectance and shading images. That is in addition to the *RGB* input image. Hence, this loss imposes the reflection model constraint of Equation 2.4:

$$\mathcal{L}_{IMF}\left(R,\,S,\,I\right)=\gamma_{IMF}\,\mathcal{L}_{RL_{IMF}}\left(\left(R\times S\right),\,I\right) \tag{2.12}$$

where I is the input image. Thus, the final loss of the IntrinsicNet becomes:

$$\mathcal{L}_{FL}\left(I,\,R,\,\hat{R},\,S,\,\hat{S}\right)=\mathcal{L}_{CL}\left(R,\,\hat{R},\,S,\,\hat{S}\right)\,+\mathcal{L}_{IMF}\left(R,\,S,\,I\right). \tag{2.13}$$

Note that the image formation loss is not limited to Equation 2.4. Any intrinsic image Equation 2.4–2.9 can be used depending on the intrinsic problem at hand. For example, the loss function for the full reflection model $\mathcal{L}_{FRM}$ is as follows:

$$\begin{aligned}\mathcal{L}_{FRM}(*)=&\gamma_R\,\mathcal{L}_{RL_R}\left(R,\hat{R}\right)\,+\gamma_S\,\mathcal{L}_{RL_S}\left(S,\hat{S}\right)\,+\gamma_H\,\mathcal{L}_{RL_H}\left(H,\hat{H}\right)\,+\gamma_E\,\mathcal{L}_{RL_E}\left(E,\hat{E}\right)\\&+\gamma_{IMF}\,\mathcal{L}_{RL_{IMF}}\left(\left(R\times S\times E+H\times E\right),\,I\right).\end{aligned} \tag{2.14}$$

The image formation loss function is designed to augment the color reproduction. To augment both *color reproduction* and *edge sharpness*, in the next section, a two-stage Retinex-inspired CNN architecture is described which uses intrinsic gradients (for edge sharpness) and the image formation loss (for color reproduction).

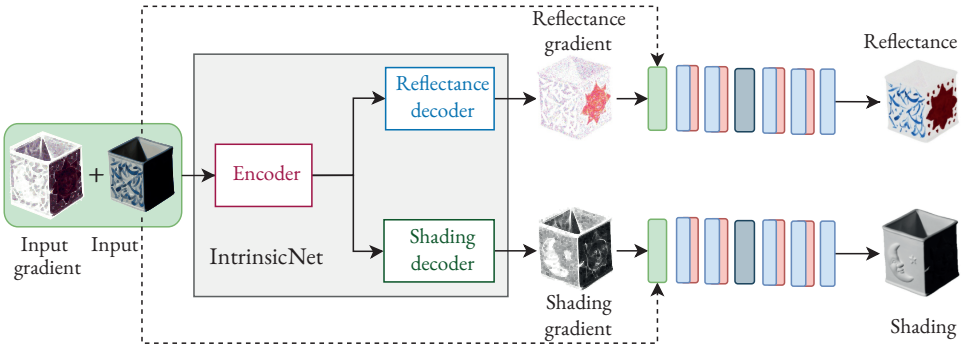


Figure 2.2: RetiNet model architecture (layer types and sub-network details as in Figure 2.1). Instead of generating intrinsic image pixel values, the encoder-decoder network is trained to separate (color) image gradients into intrinsic image gradients. Then, for gradient re-integration part, the input image is concatenated with predicted intrinsic gradients and forwarded to a fully convolutional sub-network to perform the actual pixel-wise intrinsic image decomposition.

2.3.3 RetiNet

In this section, we employ the principles of the well-established Retinex model to steer the CNN for intrinsic image decomposition. To that end, the 2-stage RetiNet model is proposed, which combines gradient information with the image formation loss.

In the first stage, the IntrinsicNet architecture is employed to separate color-image gradient into intrinsic-image gradients. The gradients $\nabla f = [f_x, f_y]$ of an image f are approximated by the channel-wise finite difference, where f_x, f_y are the horizontal and vertical components, respectively. For the sake of simplicity, in RetiNet, the channel-wise gradient magnitudes $\|\nabla f\|$ are used, where:

$$\|\nabla f\| = \sqrt{{f_x}^2 + {f_y}^2} \tag{2.15}$$

To assist the network with image gradients, the input RGB image is concatenated to its per-channel gradient magnitudes before being fed into the network, which results in a 6-channel input. The combined loss function in Equation 2.11 is used to enforce the intrinsic-image gradients of the first stage, as follows:

$$\mathcal{L}_{S1} = \mathcal{L}_{CL}\left(\|\nabla R\|, \left\|\nabla \hat{R}\right\|, \|\nabla S\|, \left\|\nabla \hat{S}\right\|\right), \tag{2.16}$$

For the second stage, the input image is concatenated with the predicted intrinsic gradients. The newly formed input is provided to a fully convolutional sub-network to perform

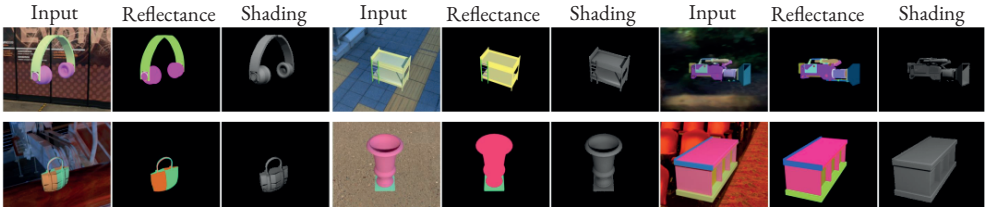


Figure 2.3: Overview of the synthetic dataset with input images and corresponding reflectance and shading ground truths. Different environment maps are used to render the models for realistic appearance.

the actual decomposition by using Equation 2.13 with the intrinsic loss. Figure 2.2 illustrates our RetiNet model.

Our network differs considerably from the threshold-driven In contrast to threshold-driven gradient separation, our network learns intrinsic gradients directly from the data without using hard-coded thresholds. For re-integration, a series of simple convolutions is proposed to separately compute the intrinsic images. This is different from other methods which try to find, by complex computations, the pseudo-inverse of an unconstrained system of derivatives, or to solve the Poisson equation.

Despite the similarities to the model of [65], our method differs in several ways. Instead of using edges, our model seeks to separate image gradient to different intrinsic components, while their method predicts a single target component. In addition, re-integration is done by a series of simple convolutions in our method, but while they use an encoder-decoder based network with de-convolutions.

2.4 Experiments

2.4.1 Physics-based Synthetic Dataset

For our experiments, large scale datasets are needed to train the networks. Due to the unavailability of the synthesis dataset [58], for fair comparison, we generate a similar dataset following the paper description. We randomly sample around 20K 3D models obtained from the ShapeNet dataset [39] for training. To increase variation and decouple the shape and texture correlation, the models' textures are replaced by random colors. The rendering is performed by the physics-based Blender Cycles*. The engine traces a light ray from each pixel back to a light source to determine the pixel color. The process depends on the properties of surfaces with which light rays interact following physically based reflection models.

*https://www.blender.org/

In the dataset, the objects' materials are modelled by a diffuse bidirectional scattering distribution function (BSDF) with random roughness. Different environment maps are used to simulate ambient light. The objects are rendered at random viewpoints sampled from the upper hemisphere as conducted in [58]. To guarantee the relationship between reflectance and shading, the Cycles rendering pipeline is modified to obtain the reflectance and shading maps corresponding to each rendered image. All images are in high-dynamic range without gamma-correction. The generated dataset contains around 20K object-centered images. An overview of the datasets is given in Figure 2.3.

2.4.2 Error Metrics

For evaluation, the common metrics are chosen, including the mean squared error (MSE), the local mean squared error (LMSE), and the structural dissimilarity index (DSSIM) [47]. The image absolute brightness is adjusted to minimize the errors. Following [56], LMSE is computed by aggregating the MSEs over $k \times k$-size regions with steps of $k/2$ ($k = 20$), normalized to $[0, 1]$. DSSIM measures the perceptual visual quality of the predicted images.

2.4.3 Implementation Details

For the encoder network, the VGG16 architecture [51] without fully-connected layers is used. Moreover, for dimensionality reduction, the max-pooling layers are replaced by convolutional layers with stride 2. In this way, our model learns its customized spatial down-sampling and is fully convolutional. For the decoder network, the encoder part is mirrored. The strided convolutional layers are inverted by a 4×4 deconvolution with stride 2. We follow [66] and use skip-layer connections to pass image details to the top layers. The connections are linked between the convolutional layers before down-sampling of encoder blocks, and the corresponding deconvolutional layers of the decoder part, except between the last block of the encoder and the first block of the decoder. Moreover, batch normalization [67] is applied after each convolutional layer, except for the last layer of the decoders and the inference net of RetiNet (prediction results) to speed up the convergence and to maintain the gradient flow. The inference net has convolution kernels of 3×3 and the layers have $[64, 128, 128, 64]$ feature maps, respectively. Our models are implemented using the stochastic gradient descent optimizer with learning rate of 10^{-5} and momentum of 0.9. A polynomial decay is applied to the learning rate to reach a final learning rate of 10^{-7}. Convolution weights are initialized by using [68] with a weight decay of 0.0005, whereas deconvolution weights are initialized randomly from a normal distribution with

	MSE		LMSE		DSSIM	
	Albedo	Shading	Albedo	Shading	Albedo	Shading
*Without $\mathcal{L}_{IMF}$	**0.0045**	0.0062	0.0309	0.0326	0.0940	0.0704
*With $\mathcal{L}_{IMF}$	0.0051	**0.0029**	**0.0295**	**0.0157**	**0.0926**	**0.0441**
+Without $\mathcal{L}_{IMF}$	**0.0005**	**0.0007**	0.0300	**0.0498**	0.0075	**0.0082**
+With $\mathcal{L}_{IMF}$	**0.0005**	**0.0007**	**0.0297**	0.0505	**0.0072**	0.0084

Table 2.1: Evaluation results of the IntrinsicNet with and without image formation loss on the MIT intrinsic benchmark (∗) and the ShapeNet test set (+). The image formation loss constrains the model to obtain better DSSIM performance. At the same time, it outperforms other models considering the MSE and LMSE metrics on real world images.

mean of 0 and standard deviation of 1. Moreover, the input size is fixed to 120×160 pixels and the batch size is fixed at 16 for all experiments. Throughout all experiments, we randomly flip, vertical or horizontal, and shift images by a random factor of $[-20, 20]$ pixels horizontally and vertically to generate additional training samples (data augmentation).

2.5 Evaluation

2.5.1 Image Formation Loss

The image formation loss benefit is shown in Figure 2.4 and Table 2.1. The results show that the image formation loss better constrains the model, leading to less halo effects, improved color reproduction, and lower perceptual dissimilarity. The model with the image formation loss obtains lower MSEs and LMSEs on average, while the model with the image formation loss achieves similar performance for MSE and LMSE on the ShapeNet set. Considering the generalization ability and the effect on a unseen real-world dataset, the network with image formation loss achieves best performance for all metrics. Employing image formation has thus proven positive impact in constraining CNNs for intrinsic image decomposition.

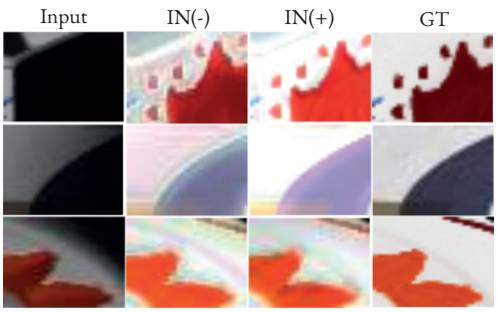


Figure 2.4: Close-ups of reflectance prediction on images from the MIT intrinsic benchmark [56]. IN(+/-) denotes the IntrinsicNet with/without the image formation loss. The image formation loss suppresses color artifacts and halo effects.

	MSE		LMSE		DSSIM	
	Albedo	Shading	Albedo	Shading	Albedo	Shading
DirectIntrinsics [38]	0.1487	0.0505	0.6868	0.3386	0.0475	0.0361
ShapeNet [58]	0.0023	0.0037	0.0349	0.0608	0.0186	0.0171
IntrinsicNet	0.0005	0.0007	0.0297	0.0505	0.0072	0.0084
RetiNet	**0.0003**	**0.0004**	**0.0205**	**0.0253**	**0.0052**	**0.0064**

Table 2.2: Evaluation results on ShapeNet. Our proposed methods yield better results on the test set. Moreover, our RetiNet model outperforms all by a large margin.

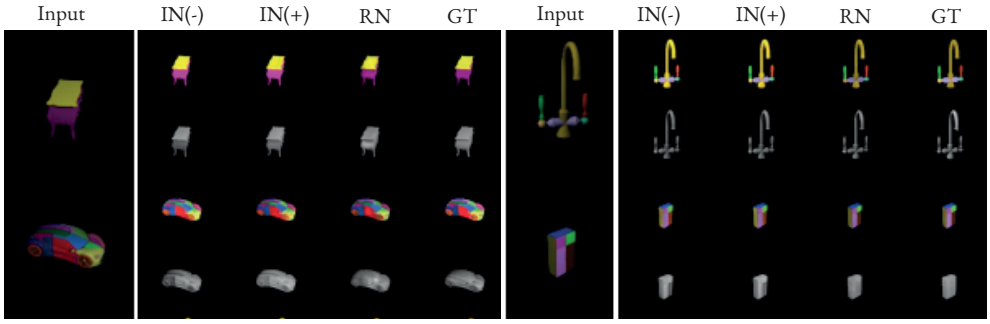


Figure 2.5: Evaluation results on the synthetic test set. All proposed models produce decent intrinsic image compositions. IN(+/-) denotes the IntrinsicNet with/without the image formation loss, and RN denotes the RetiNet model.

2.5.2 ShapeNet Dataset

We now test our models on the ShapeNet test partition. We follow the approach of [58] and randomly pick 1 image per test model, resulting in 7K test images. For all experiments, the same test set is used. Table 2.2 shows the quantitative evaluation results of the synthetic test set of man-made objects. Figure 2.5 displays (visual) comparison results. Our proposed methods yield better results on the test set. Moreover, our RetiNet model outperforms all by a large margin. Visual comparison results show that all of our proposed models are capable of producing decent intrinsic image compositions on the test set.

2.5.3 MIT Intrinsic Benchmark

The MIT dataset [56] consists of 20 object-centered real-world images with a single canonical light source. The quantitative and qualitative comparison to state-of-the-art methods are shown in Table 2.3 and Figure 2.6, respectively. Our proposed methods yield better results compared with the ShapeNet [58] and DirectIntrinsics [38] models. Visually,

	MSE		LMSE		DSSIM	
	Albedo	Shading	Albedo	Shading	Albedo	Shading
Retinex [56]	**0.0032**	0.0348	0.0353	0.1027	0.1825	0.3987
DirectIntrinsics [38]	0.0277	0.0154	0.0585	0.0295	0.1526	0.1328
ShapeNet [58]	0.0468	0.0194	0.0752	0.0318	0.1825	0.1667
IntrinsicNet	0.0051	**0.0029**	**0.0295**	**0.0157**	0.0926	**0.0441**
RetiNet	0.0128	0.0107	0.0652	0.0746	0.0909	0.1054
RetiNet + GT∇	0.0072	0.0034	0.0429	0.0224	**0.0550**	0.0443

Table 2.3: Evaluation on the MIT intrinsic benchmark [56]. Our proposed methods yield better results, while experiment with intrinsic gradient ground-truths shows the benefits of exploiting them.

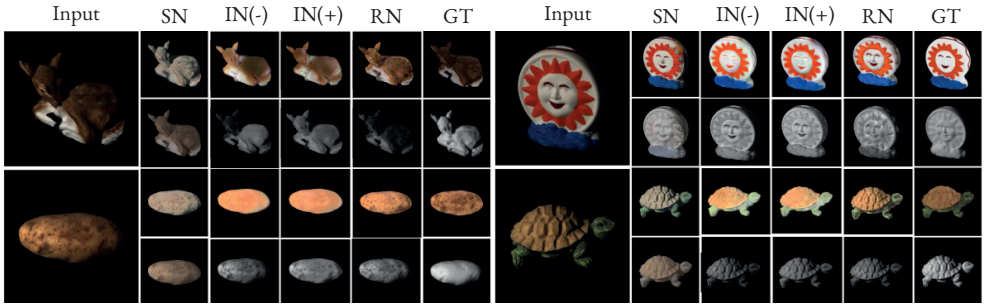


Figure 2.6: Qualitative results on the MIT intrinsic benchmark [56]. SN is the ShapeNet model [58], IN(+/-) are the IntrinsicNet with/without the image formation loss. RN is the RetiNet model (with the image formation loss). The proposed models properly recover the reflectance and shading information. IN(+/-) create blurry results compared with RetiNet, while IN(-) also suffers from color artifacts.

our proposed models properly recover the reflectance and shading information, in which RetiNet's results have more vivid colors, with sharper edges, and less color artifacts, despite the strong shadow cast, *c.f.* the *deer* image. IntrinsicNet trained without image formation loss produces even more artifacts. Figure 2.7 displays a detailed analysis of RetiNet.

2.5.4 In-the-wild Images

We also evaluate our RetiNet algorithm on real and in-the-wild images. Figure 2.8 shows the performance of our method (RetiNet) for a number of images. The results show that it can capture proper reflectance image, free of shadings caused by geometry. Finally, we present the reconstructed input from its albedo and shading prediction to show that the decomposition is consistent.

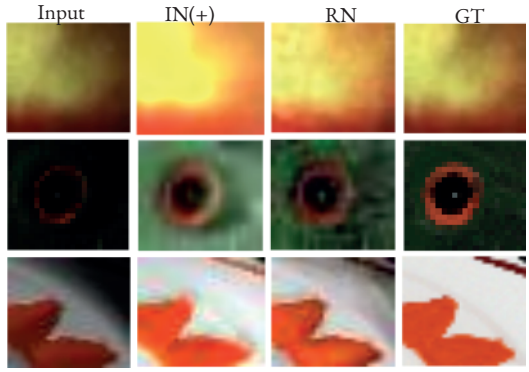


Figure 2.7: Close-ups of reflectance prediction on the MIT intrinsic benchmarks [56]. IN(+) is the IntrinsicNet with the image formation loss, and RN the RetiNet model with image formation loss. Reflectance images appear with more vivid color for RetiNet, while IN(+) has color artifacts and blurriness.

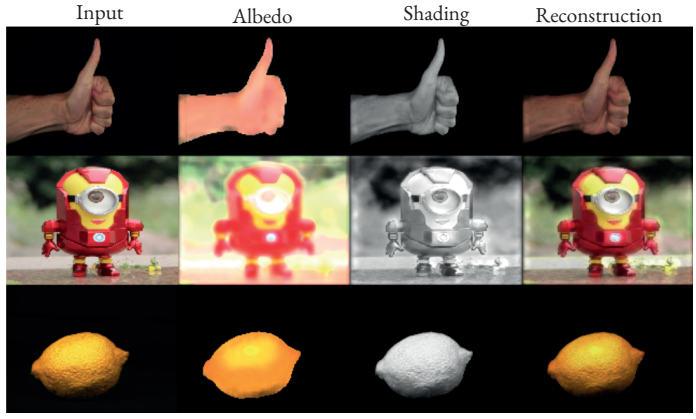


Figure 2.8: Inference RetiNet using real in-the-wild images shows proper shading-free reflectance.

2.6 Conclusions

We propose two deep learning models considering a physics-based reflection model and gradient information to steer the learning process. To train the models, an object centered large-scale synthetic dataset based on physical lighting models with intrinsic are generated. The proposed models are evaluated on synthetic, real world and in-the-wild images. The evaluation results demonstrate that the new model outperforms existing methods. Furthermore, visual inspection show that the image formation loss function augments color reproduction and the use of gradient information produces sharper edges.

3

Three for One and One for Three: Flow, Semantics, and Surface Normals

Optical flow, semantic segmentation, and surface normals depict different traits, yet together they bring complementary cues for scene understanding. In this chapter, we study the impact of one modality on the others and their efficiency in combination. A convolutional refinement network is trained with multimodal input and apart from RGB images to enforce joint modality features. The experimental results on both structured and unstructured datasets show positive influence among the three modalities, especially for objects' boundaries, and region consistency.

3.1 Introduction

Optical flow, semantic segmentation, and surface normals represent different aspects of objects in a scene, i.e. motion, category, and geometry (Figure 3.1). While they are often approached as single-task problems, their combinations are of importance for general scene understanding as humans also rarely perceive objects in a single modality. As different information sources provide different cues to understand the world, they could also become complementary to each other. For example, certain objects have specific motion patterns (flow and semantics), an object's geometry provides specific cues about its category (surface normals and semantics), and object's boundary curves provide cues about motion boundaries (flow and surface normals).

(a) (b) (c) (d) (e)

Figure 3.1: Multimodal data from the Virtual KITTI [18] (*top*) and EDEN (Chapter 6, *bottom*) datasets: (a) RGB image, (b) semantic segmentation, (c) optical flow, (d) optical flow magnitude, (e) surface normals.

Scene-based optical flow estimation is a challenging problem because of complicated scene variations such as texture-less regions, large displacements, strong illumination changes, cast shadows, and specularities. As a result, optical flow estimation tends to perform poorly in homogeneous areas or around objects' boundaries. Another hindrance for many optical flow estimators is the common assumption of spatial homogeneity in the flow structure across an image [69]. That assumption poses difficulties as different objects have different motion patterns: objects closer to the viewer have stronger optical flow; independent objects have their own flow fields, while static objects follow the camera's motion (Figure 3.1). Thus, by modeling optical flow based on image segmentation, one could improve flow accuracy, especially at objects' boundaries [69, 70].

The goal of image segmentation is to partition an image into different parts that share common properties. In particular, semantic segmentation assigns to each image pixel the object category to which the pixels belong. It is a challenging task especially in videos, due to inherent video artifacts such as motion blur, frame-to-frame object-to-object occlusions and object deformations. As optical flow encodes temporal-visual information of image sequences, it is often exploited to relate scene changes over time [71, 72, 73]. Yet, optical flow, as an approximation to object *motion field* [74], also encodes the 3D structure of a viewed scene. If the camera's translation is known beforehand, an optical flow image can be used to recover the scene depth [75]. Considering a moving camera, closer objects appear with stronger motion fields than distant ones, independent moving objects have prominent motion patterns compared to the background, and different object shapes generate different motion fields because of depth discontinuities. Therefore, temporal and structural information provided by optical flow can guide semantic segmentation by providing cues about scene ambiguities.

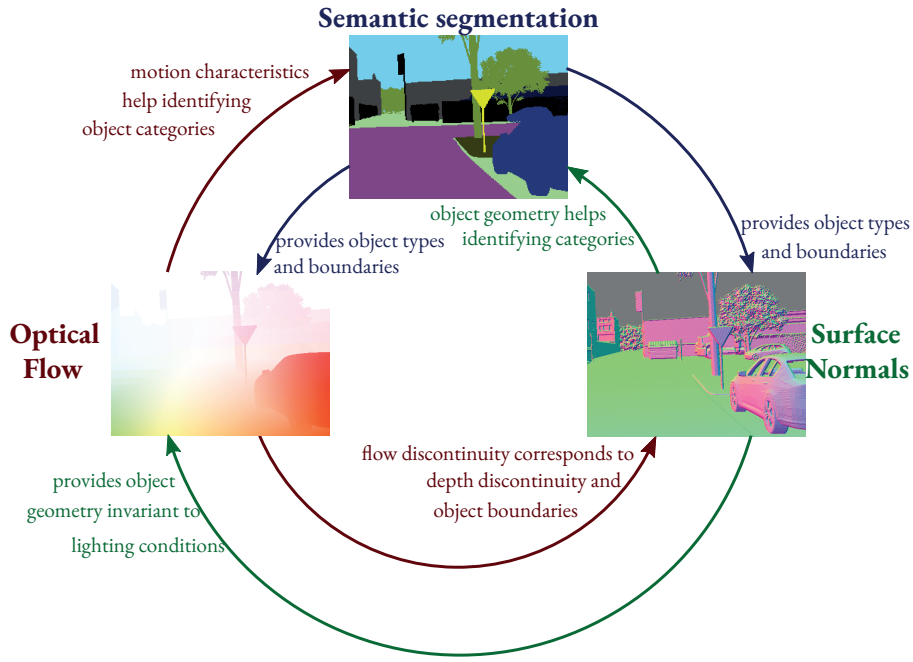


Figure 3.2: The relationship among the three modalities: optical flow, semantics, and surface normals; and their impacts on one another.

Surface normals, on the other hand, represent changes of depth, i.e. the orientation of object surfaces in 3D space. Thus, they are independent of illumination effects or object textures. That information is particular helpful for texture-less objects, regions of strong cast shadows, or scenes of less visibility. Additionally, object boundaries provide cues about both motion and semantic boundaries. Therefore, surface normals are expected to assist the optical flow estimation process by providing various cues. Figure 3.2 illustrates the relationship of object boundaries depicted in segmentation and 3D structure cues in optical flow and surface normal images.

In this chapter, we study the mutual interaction of optical flow, semantic segmentation, and surface normals and analyze their contribution to each other. We employ a modular approach and adapt a convolution based supervised refinement network to examine the efficiency of joint features from the different modalities.

In summary, our contributions are: (1) the connection among the three modalities (optical flow, semantic segmentation and surface normals), (2) adapting a convolutional based supervised refinement network to improve one of the three using the other two, (3) an experimental study to estimate all three in a joint fashion.

3.2 Optical Flow, Semantics, and Surface Normals

3.2.1 Related Work

3

In this section, we review the work on optical flow, semantic segmentation, and surface normals and how they are mostly targeted as single tasks.

Optical flow is defined as the apparent *motion field* resulted from an intensity displacement in a time-ordered sequence of images. It is an approximation to image motion, because estimating optical flow is an ill-posed problem [74]. To model the displacement of image intensities that are caused solely by the objects' motion in the physical world, several priors are derived to constrain the problem. Two of the most exploited ones are the assumptions of brightness constancy and Lambertian surface reflectance [74, 76, 77]. Besides, [78] makes use of robust statistics to promote discontinuity-preservation. Many popular methods also apply coarse-to-fine strategies [79, 80, 81]. On the other hand, deep convolutional neural networks (CNNs) are dominating the field more recently. For instance, [82] applies a coarse-to-fine strategy with the help of a CNN framework. Then, Dosovitskiy *et al*. [83] proposes an end-to-end CNN called FlowNet, which is later improved by Ilg *et al*. [84] to perform state-of-the-art optical flow estimations.

Semantic Segmentation is vital for robot vision and scene understanding tasks as it provides pixel-wise annotations to scene properties. Traditional methods approach the problem by engineering hand-crafted features and perform pixel-wise classification with the help of a classifier [85, 86]. Other works try to group semantically similar pixels [87, 88]. Like most of the computer vision tasks, semantic segmentation also benefits from powerful CNN models. After the pioneering work of [89], many other deep learning based methods are proposed such as [90, 91].

Surface Normals provide information about an object's surface geometry. Traditional methods that infer 3D scene layout from single images rely on primitives detection such as oriented 3D surfaces [92] or volumetric primitives [93]. Their performance depends on the discriminative appearance of the primitives [94]. In the context of deep learning, Wang *et al*. [95] proposes a method to predict surface normals from a single color image by employing a scene understanding network architecture with physical constraints; Bansal *et al*. [96] predicts surface normals and use them as an intermediate representation for 3D volumetric objects for model retrieval. Eigen and Fergus [97] designs an architecture that can be used for predicting each of the three modalities, including depth, surface normals, and semantic segmentation, in a separating manner.

3.2.2 Inter-modal Influences

Semantics and surface normals on optical flow prediction

Object motion field, although varies across image regions, is often treated in the same way by many optical flow methods. Semantic segmentation provides a way to partition an image into different groups of predefined semantic classes. Hence it provides optical flow with information of object boundaries, and helps to enforce motion consistency within similar object regions. Similar ideas employed by [70] with object instance or [69] with 3 semantic classes (things, planes, and stuff) have shown to improve optical flow accuracy.

On the other hand, surface normals represent the orientation of objects' surfaces in 3D space. They contain geometry information that is invariant to scene lighting and objects' appearance, which is rendered useful for optical flow in case of intricate lighting such as cast shadows, texture-less regions, specularities, etc. Additionally, surface normals can be beneficial for optical flow for depth order reasoning, and improve occlusion boundaries.

Optical flow and surface normals on semantic segmentation

Optical flow is often exploited for its ability in relating scene changes along time-axis: He *et al*. [98] aggregates information from multiple views using optical flow to perform semantic segmentation for a single frame, while Zhu *et al*. [73] uses optical flow to propagate image features from keyframe images to nearby frames, speeding up the segmentation process. Several methods exploit motion information to segment images into foreground objects from a moving background, such as SegFlow [99] or FusionSeg [100]. However, they only rely on the motion properties of objects and do not take into account the objects' identities.

In this chapter, we leverage the notion of segmentation into a more semantic meaning of a scene, i.e. we do not limit the segmentation to just foreground/background [99] or coarse general classes (things, planes, stuff) [69], but rather adhere to the current segmentation problems in the literature, which on average, consist of 10–20 classes [18, 101, 102].

As object *motion field* is the projection of 3D object motions, depth discontinuities correspond to motion boundaries, making optical flow an indication of scene depth. At the same time, surface normals represent the changes of depth and the alignment of object surfaces. Such information is particularly useful for semantic segmentation, as objects can be recognized not only from their appearances, but also from their shape and geometric characteristics. Thus, similar to methods that recognize objects from depth by associating each object with their motion type, it is feasible to recognize objects using geometry information signified from optical flow and surface normals.

Optical flow and semantics on surface normals prediction

Similar to the case of optical flow, semantic segmentation provides object boundary information, which, in many cases, corresponds to depth disruption. Thus it enhances object boundaries, and local coherence in predicting surface normals. Ladicky *et al*. [103] enforces smooth surface normals estimated with contextual information.

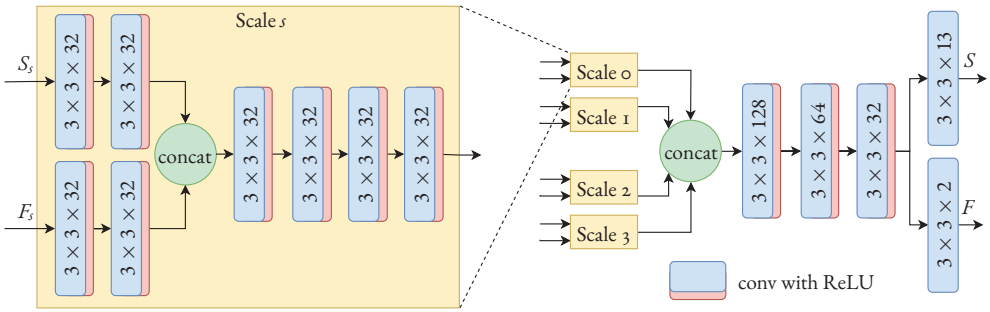
Optical flow represents motion information, yet also signifies geometry structure of a scene. As surface normals are identified by the rate of change in the location of objects, they are highly correlated. Thus, optical flow can provide useful cues to enhance surface normals, in terms of objects' inner structure as well as their boundaries.

3.3 Method

To study the relationship between the three modalities, we follow a refinement strategy based on the work by Jafari *et al*. [104], and adapt the architecture to train a joint refinement network designed for semantic segmentation, surface normals and optical flow.

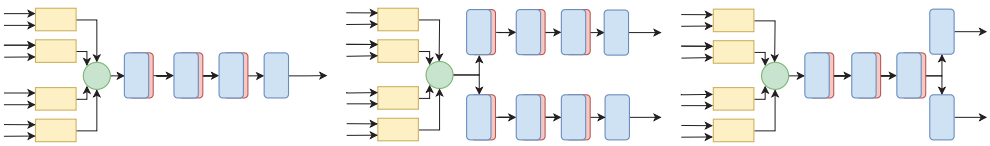
An overview of the architecture for joint refining optical flow and semantic segmentation is shown in Figure 3.3. The network input and output is adjusted according to the corresponding number of modalities. The example network in Figure 3.3(b) takes in a preliminarily predicted semantic segmentation and optical flow at different scales and couples them in a joint optimization process. The input to a branch scale s (Figure 3.3(a)) is composed of a segmentation image S_s and a flow image F_s, both sub-sampled to $\frac{1}{2^s}$ of the original size. The outputs of the scale branches are up-sampled, as the refinement network provides output at the original image size. We add a scale branch at the original size to the proposed architecture [104], and expand the network depth to increase its capacity to cope with different input modalities. When there are 3 modality inputs, the scale branch will have a third input, which is then concatenated to the other two.

We also leverage the study of cross-modality influence performed in [104], and the ability of the refinement network to learn a joint representation that benefits from both modalities. We keep the multi-scale encoder part fixed (up to the *concat* layer) and partially decouple the decoder. That allows the network to have different capacities in using the joint features learned from the encoder to refine different modalities. Specifically, we examine three different architectures that impose different coupling levels of joint features as illustrated in Figure 3.4. Namely, we study the ability of refinement when there is only one task required, hence *zero*-coupling (Figure 3.4a), when both 2 tasks are *loosely* coupled (Figure 3.4b), or *tightly* coupled (Figure 3.4c).



(a) Scale box structure (b) refinement architecture

Figure 3.3: Joint refinement network for two modalities with *tight* feature coupling (*left*), inspired by [104]. The outputs of the modal-specific networks are integrated at different scales, using scale branch architecture (*right*), and up-sampled before concatenation.



(a) (b) (c)

Figure 3.4: Coupling levels of joint features in refinement: (a) *zero* coupled, where joint features refine a single task; (b) *loosely* coupled, where joint features branch to refine each task separately; (c) *tightly* coupled, where joint features share a decoder to refine all tasks.

3.4 Experiments

3.4.1 Experimental Setup

Datasets

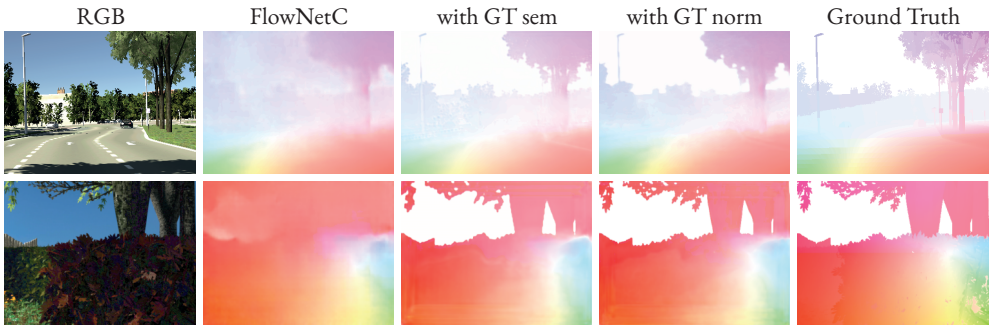
As pixel-wise annotations of optical flow and surface normals are not intuitive for manual labelling, there are no large-scale real-world dataset having annotations for both optical flow and surface normals. As a result, we rely on synthetic datasets for evaluation purpose.

EDEN. The synthetic dataset of Enclosed garDEN scenes (EDEN) features different vegetation types such as trees, bushes, flowers, and grass, as well as various terrain and landscape types. The dataset construction details are described in Chapter 6. We use the first release of the dataset, consisting of 300 images from 10 scenes in 5 different lighting conditions (clear, cloudy, overcast, sunset, and twilight), resulting in 15K images.

Virtual KITTI [18] (VKITTI) is a large-scale synthetic dataset following the setting of KITTI dataset [105] for autonomous driving problems. The VKITTI scenes are highly structured, compared to EDEN, where objects are mostly rigid and with clear boundaries,

Dataset	FlowNetC	with GT sem	with GT norm
VKITTI	2.68	**2.08**	2.09
EDEN	16.19	**12.17**	12.28

(a) Quantitative results



(b) Qualitative results

Figure 3.5: Quantitative (a) and qualitative (b) results of oracle optical flow refinement on VKITTI (*top*) and EDEN (*bottom*). Semantic information and surface normals comparably improve optical flow performance over the baselines and the crispness of objects' boundaries

thus expected to have relatively lower impact on geometry-related tasks. Each image frame comes with pixel-wise annotation of ground truth instance-object segmentation, optical flow, and depth information. To get surface normal ground truth, we convert ground truth depth images using the method described in [28]. As the depth images are produced by a simulation renderer, they are free from noise and uncertainties, producing less artifacts.

Baselines & Evaluation metrics

Each of the three modalities have their own baseline and evaluation metric.

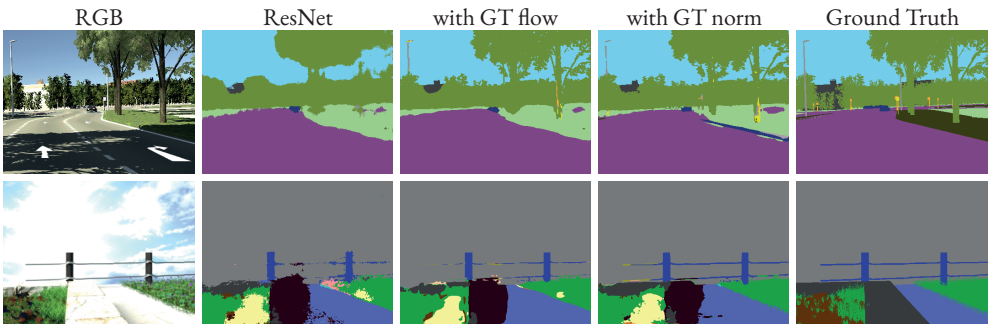
Optical Flow. We use FlowNetC [83] as our baseline because of its balance in speed and accuracy [72, 99] (and is therefore preferable over the less accurate FlowNetS or the more expensive FlowNet2 [84]). We fine-tune the network for each dataset and report the results as baseline. Performance is evaluated by average endpoint errors (EPE): lower is better.

Semantic Segmentation. We use the ResNet-101 architecture [106] as the baseline, and follow the practice of [99, 107] to add corresponding decoder layers so that the network can output full-resolution images. Performance is measured by mean intersection-over-union (mIOU), shown in percentage, the higher the better.

Surface Normals. We follow the MarrRevisited [96] architecture and report their re-

Dataset	ResNet	with GT flow	with GT norm
VKITTI	44.11	46.73	**51.12**
EDEN	37.88	45.27	**48.47**

(a) Quantitative results



(b) Qualitative results

Figure 3.6: Quantitative (a) and qualitative (b) results of oracle semantic segmentation refinement on VKITTI (*top*) and EDEN *bottom*. Adding surface normals show more improvement on semantic segmentation than adding optical flow, indicating more correlation of object shapes and types over motion. Both two modalities help capture more fine-detailed segmentation (*e.g.* tree leaves and fence wires).

sults as the baseline. Evaluation is based on the angular differences between predicted normals and ground truth [94]. The 3 error measurements *mean*, *median*, *rmse* show the difference (degrees) between predicted and ground truth normal vectors, thus lower is better. The 3 measurements 11.25°, 22.5°, 30° count the number of pixels within the indicated angular thresholds; the results are shown in percentage, and higher is better.

3.4.2 Baseline & Oracle Experiments

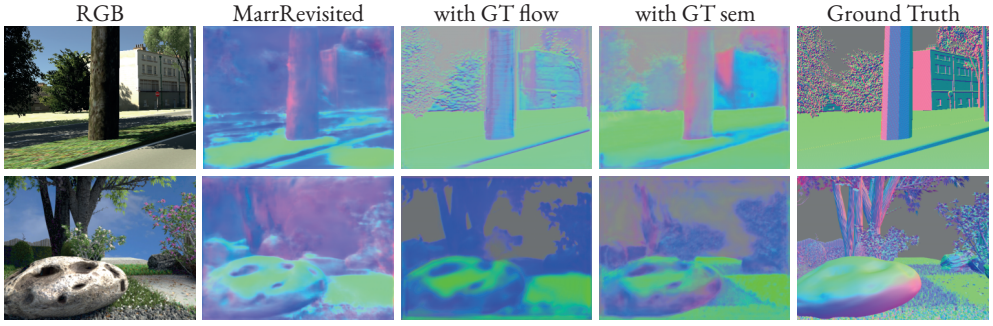
The first set of experiments is to test the hypothesis that the modalities have a positive impact on each other and to establish the baselines. For each experiment, we train the aforementioned baseline networks, and have the output results passed into the refinement architecture (described in Section 3.3) together with ground truths of other modalities.

Optical Flow

Figure 3.5a shows the baseline and refinement results for optical flow, using ground truth segmentation and surface normals. In general, both modalities help improve optical flow. The refined results in Figure 3.5b appear crispier, especially along the objects' boundaries.

		mean (↓)	median (↓)	rmse (↓)	11.25 (↑)	22.5 (↑)	30 (↑)
VKITTI	MarrRevisited	48.13	48.34	57.44	17.39	19.81	27.44
	with GT flow	11.08	3.49	16.93	**62.58**	**77.72**	87.81
	with GT sem	**10.99**	**2.97**	**16.55**	61.33	76.51	**89.22**
EDEN	MarrRevisited	40.25	46.68	50.25	29.44	30.68	34.42
	with GT flow	9.72	8.12	13.33	61.35	89.38	97.37
	with GT sem	**8.49**	**6.38**	**11.96**	**68.57**	**92.33**	**97.86**

(a) Quantitative results



(b) Qualitative results

Figure 3.7: Quantitative (a) and qualitative (b) results of oracle surface normals refinement on VKITTI (*top*) and EDEN (*bottom*). Adding oracle optical flow or semantic information significantly outperforms the baselines. Semantic information results in more improvement, indicating better correlation of object types and shapes over motion.

Semantic Segmentation

As shown in Figure 3.6a, optical flow and surface normals also improve semantic segmentation over the baseline, since the outline of the objects obtained from these modalities are more informative. The geometry information provided by flow and normals helps to retain details in semantic segmentation; e.g. the lamppost, tree branches, and fence wires are well retained in Figure 3.6b. However, as the refinement module does not have access to the original image (raw RGB), geometry information alone cannot help much in correcting semantic errors that are present in the input (yellow regions in the second row).

Surface Normals

As shown in Figure 3.7a, using optical flow and semantic segmentation helps to imrpove surface normal prediction significantly. The refinement using flow seems to be better than using segmentation for the VKITTI case, whereas it is the other way around for the EDEN

Target	Baseline	GT	Predicted			
		zero	*zero*	*loose*	*tight*	*tight+*
Semantic segmentation (↑)	44.11	46.73	**44.71**	41.2	41.1	43.9
Optical flow (↓)	2.68	2.08	**2.13**	2.43	2.41	2.42

Table 3.1: Refining segmentation and flow based on predictions on VKITTI dataset. Performance is measured in Mean IOU (↑) and Average EPE (↓) respectively. Input to the refinement module composes of segmentation and optical flow, either both predicted (*Predicted*) or with one ground truth (*GT*); the refined output is 1 single modality (*zero*) or both 2 modalities (*loose* and *tight*). *tight+* is when the input modalities are updated in an end-to-end training with RGB input.

case. This can be explained as estimating surface normals requires a network to understand the geometry information of the scene, which is easier for optical flow in EDEN as all of the objects are static and thus optical flow field depends solely on the camera ego-motion, while it is not the case for VKITTI, segmentation has more advantage as objects' shapes are more uniform (e.g. houses, cars, roads' surface) than those in EDEN (e.g. bushes, rocks, grass). The refined result using oracle flow produces sharper details, while the ones with oracle segmentation are more accurate (Figure 3.7b).

To conclude, different modalities, when being used in their most accurate form (GT), provide complementary cues to each other, thus improving the performance of other modalities: segmentation provides flow and normals with objects' identities and boundaries; optical flow provides segmentation and normals motion and geometry information; surface normals provide segmentation and flow with geometry and objects boundaries. In the following experiments, we examine the usefulness of different modalities when they are predicted and the interaction of more than one modalities.

3.4.3 Cross-Modality Influence

Model comparison

In this experiment, we consider joint learning of semantic segmentation and optical flow on the VKITTI dataset [18]. We examine the different coupling levels during refinement: *zero* coupling, *loose* coupling, and *tight* coupling (see Figure 3.4). We also expand the *tight* coupling, into an end-to-end learning pipeline (denoted by *tight+*), where the segmentation and optical flow networks are fine-tuned together with the refinement module.

The results are shown in Table 3.1. From the results, we observe that using ground-truth or predicted segmentation to refine optical flow, the performance always improves. The

Method	FlowNetC	with PR sem	with PR norm	with PR sem+norm
VKITTI	2.68	**2.13**	2.23	2.14
Nature	16.19	**12.41**	15.48	12.45

(a) Quantitative results

(b) Qualitative results

Figure 3.8: Quantitative (a) and qualitative (b) results of prediction-based optical flow refinement on VKITTI (*top*) and Nature (*bottom*). Adding predicted semantic and surface normals improve optical flow performance. Surface normals are less accurate for deformable objects in EDEN, resulting in diminishing gains compared to semantic information. Predicted semantic information and surface normals improve the object delineation in optical flow prediction.

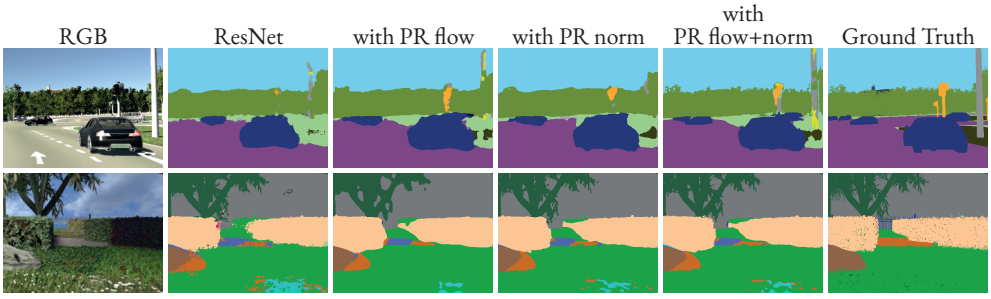
difference between ground-truth and predicted segmentation is small compared to the difference between the baseline and the refined models. However, for segmentation, refining based on flow is only beneficial when the *zero* coupling is used. Likely, this is because the predicted flow does not contain accurate semantic cues to improve segmentation. Based on this experiment we use the *zero* coupling for the remaining experiments.

Flow from Segmentation and Normals

The refinement results for optical flow using predicted modalities are shown in Figure 3.8a. Because of inaccuracies in predicted normals, the refined results do not improve as much as with predicted segmentation, or even hurt in case of EDEN dataset. In general, the two modalities help optical flow to obtain better delineation. Figure 3.8b shows an occlusion case where part of the car is occluded by a traffic sign, FlowNetC recognizes it but fails to obtain the correct shape of the occlusion, which can be recovered with surface normal information and improved using segmentation and surface normals.

Method	ResNet	with PR flow	with PR norm	with PR flow+norm
VKITTI	44.11	44.71	**45.66**	44.55
EDEN	37.88	**45.29**	45.25	45.02

(a) Quantitative results



(b) Qualitative results

Figure 3.9: Quantitative (a) and qualitative (b) results of prediction-based semantic segmentation refinement on VKITTI (*top*) and EDEN (*bottom*). Adding predicted surface normals and optical flow improves semantic segmentation performance and object delineation. Structured scenes of VKITTI result in higher gain of surface normals over optical flow compared to EDEN.

Segmentation from Flow and Normals

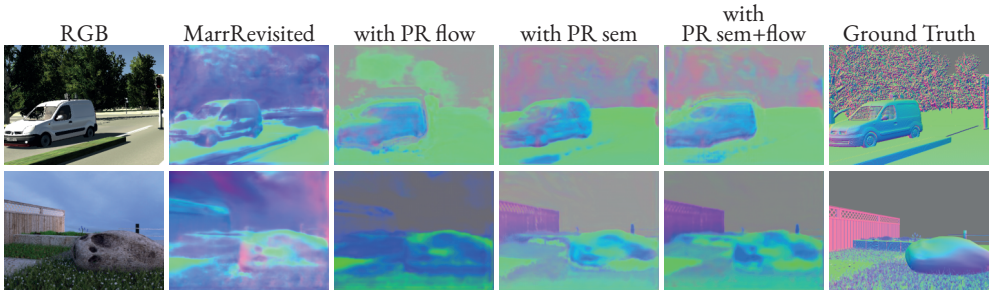
Predicted optical flow and surface normals contain different inaccuracies. Thus, when used to refine the semantic segmentation, they confuse the semantic cues, and to some extent, have negative impact on the preliminary segmentation results. This explains the decreasing results and slight improvements in Figure 3.9a. Visual inspection on Figure 3.9b shows that refinement of flow and surface normals make the boundaries smoother, reducing the effect of incorrect areas. In combination, they capture more details and help to produce better segmentation.

Normals from Flow and Segmentation

Surface normal refinement results are provided in Figure 3.10a and illustrated in Figure 3.10b. The confusion in the sky and tree regions of the baseline estimation is removed when refined with different modalities. Information of objects' categories and boundaries provided by semantic segmentation helps retaining fine details in the results (e.g. pavement in VKITTI, fences in EDEN). However, the inaccuracies of predicted flow leave some artifacts and makes the results less accurate (e.g. the sky, tree and fences regions).

	Dataset	mean (↓)	median (↓)	rmse (↓)	11.25 (↑)	22.5 (↑)	30 (↑)
VKITTI	MarrRevisited	48.13	48.34	57.44	17.39	19.81	27.44
	with PR flow	12.28	4.09	18.05	58.37	73.23	84.84
	with PR sem	**11.43**	**2.71**	**17.22**	**60.48**	74.34	**86.68**
	with PR flow+sem	11.67	3.98	17.26	59.65	**74.70**	86.48
EDEN	MarrRevisited	40.25	46.68	50.25	29.44	30.68	34.42
	with PR flow	11.33	9.77	14.33	55.61	87.08	96.71
	with PR sem	**8.95**	**6.90**	**12.40**	**66.03**	**91.52**	**97.80**
	with PR flow+sem	9.29	7.50	12.85	63.58	90.51	97.57

(a) Quantitative results



(b) Qualitative results

Figure 3.10: Quantitative (a) and qualitative (b) results of prediction-based surface normals refinement on VKITTI (*top*) and EDEN (*bottom*). Adding predicted optical flow and semantic information improve surface normals prediction. Semantic-based refinement produces best results in both datasets.

3.5 Conclusions

We have analyzed the combination of three important modalities in computer vision, namely optical flow, semantic segmentation, and surface normals, and their impact on each other. Because each modality contains different type of information, in combination, they provide complementary cues to enhance each other. We approached the problem at a modular level where the inputs are kept fixed at the preliminary estimation. Future work will include end-to-end training of modalities to exploit raw image features.

4

Automatic Generation of Dense Non-Rigid Optical Flow from Object Segmentation

There hardly exists any large-scale datasets with dense optical flow of non-rigid motion from real-world imagery as of today. The reason lies mainly in the difficulty of manual annotation of optical flow ground-truths. To circumvent the need for human annotation, we propose a framework to automatically generate optical flow from real-imagery videos. The method extracts and matches objects from video frames to compute initial constraints, and applies deformation over the objects of interest to obtain dense flow fields. Several ways to improve dataset variations are proposed. Extensive experimental results show that training on our automatically generated optical flow outperforms methods that are trained on rigid synthetic data for FlowNet-S, PWC-Net, and LiteFlowNet architectures.

4.1 Introduction

Optical flow estimation has gained significant progress with the emergence of convolutional neural networks (CNNs) [83, 84, 108, 109]. With CNNs designed for optical flow, there is a growing demand for large scale datasets with corresponding dense optical flow fields. However, large-scale datasets with real world imagery and corresponding dense optical flow fields do not exist. The reason is that dense flow fields are neither measurable with a sensor nor trivial to be annotated by humans. For example, the KITTI datasets [105, 110]

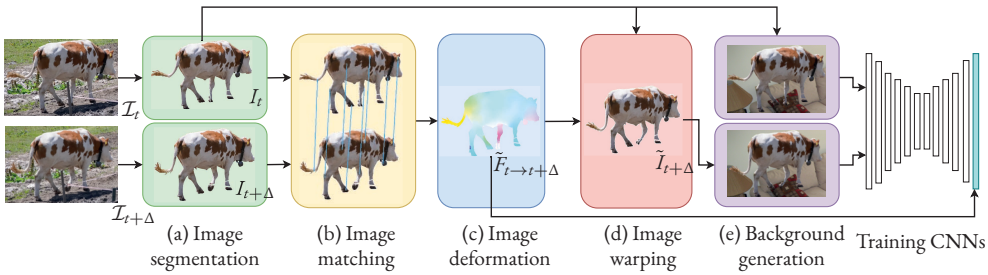


Figure 4.1: Overview of the proposed pipeline to generate a dense optical flow field from two video frames: (a) the objects of interest are extracted; (b) motion characteristics are captured by finding correspondences between the objects; (c) object deformation constrained by the correspondences results in a dense flow field; (d) the resulting flow field is used to warp the object; and (e) both the extracted first-frame object and the warped object are pasted on a random background. The resulting pair of frames is used to train a deep neural network with the dense flow field as the ground truth.

are constructed by registering point clouds from 10 consecutive frames. Then, by manual labelling, ambiguous points are removed before projecting the frames back to the image space. While being the largest optical flow dataset available today with real imagery, only 200 pairs of frames are available. This is insufficient for supervised training of CNNs for optical flow estimation.

To resolve the data demand of CNNs, synthetic data are often used. Large-scale synthetic datasets are generated from object images (*e.g.* chairs), which are deformed by affine transformations (zooming, rotation, and translation) and projected on randomly transformed backgrounds. This process is the basis of the FlyingChairs dataset [83]. Due to the large number of available frames, these datasets are useful for training optical flow CNNs.

Computer-generated imagery (CGI) datasets are also the norm. Computer-aided design (CAD) object models are used in a virtual world which are rendered to images with arbitrary lighting and environments. Examples include the MPI-Sintel [40], Virtual KITTI [18], FlyingThings3D, Monkaa, and Driving [55] datasets. CGI techniques allow datasets to vary highly in appearance (geometry, textures, lighting, *etc.*) and motion types (rigid, non-rigid, motion blur, *etc.*), thus are useful sources for assessing robustness of optical flow prediction algorithms. A well-known optical flow benchmark is provided by the MPI-Sintel [40], whose images and annotations are rendered from the CGI movie Sintel. However, the use of CGI textures in such datasets might not reflect the challenges in real-world images (*e.g.* camera noises) [111], thus limiting the generalibility of results.

Training with non-rigid motion is important for optical flow in real world imagery, since many real objects deform in a non-rigid manner. Unfortunately, non-rigid optical flow ground-truth is not available in the current datasets [105, 110].

Therefore, in this chapter, we present a new approach to automatically generate dense non-rigid optical flow fields from real-imagery videos. As illustrated in Figure 4.1, our approach collects motion statistics from real-imagery videos by computing image correspondences between segmented objects of interest. The segmented objects are warped to generate complex deformations according to physical principles to generate dense flow fields. Our method generates large amounts of optical flow data consisting of real-imagery textures and non-rigid motions to be used for training CNNs for optical flow estimation.

The contributions of this chapter are threefold. (1) We introduce the first method to automatically generate dense optical flow fields from real-videos, without manual labeling; (2) we make available a dataset with 30K frames consisting of natural textures and non-rigid optical flow, created from the DAVIS [112] video dataset; and (3) we extensively analyze optical flow methods trained using our dataset.

4.2 Related Work

4.2.1 Optical Flow Methods

As optical flow estimation is ill-posed [74], various assumptions are proposed to constrain the problem [113, 114], such as the brightness constancy, local smoothness, and Lambertian surface reflectance [74, 76]. Strategies based on coarse-to-fine warping [80, 81, 115] are employed to reduce the correspondence search space. EpicFlow [79] proposes an effective post-processing for interpolating sparse matches to dense flow [70, 116, 117, 118].

Supervised training Recently, with the success of CNNs, optical flow estimation is shifted from an energy-optimization process to a data-driven approach. Dosovitskiy *et al*. [83] propose FlowNet, a CNN which is trained end-to-end. The network is extended by Ilg *et al*. [84] to provide FlowNet2. Other methods propose ways to apply domain knowledge and classical principles such as spatial pyramid, warping, and cost volumes for fast processing. For example, LiteFlowNet [108] has 30 times fewer parameters than FlowNet2. PWC-Net [109] has 17 times fewer parameters.

Unsupervised training To avoid the need to generate optical flow ground truth, Meister *et al*. [119] replace the supervised loss by an occlusion-aware bidirectional flow estimation and trains the FlowNets in an unsupervised way. Zou *et al*. [120] use a similar approach by applying a cross-task loss. SelFlow [121] distills reliable flow estimations from non-occluded pixels, and uses these predictions as ground truth to learn optical flow. However, unsupervised methods are limited by the power of loss functions i.e. their ability to model the problem and the contribution of weights for each component loss [119].

4.2.2 Optical Flow Datasets

Most of the benchmark datasets provide optical flow for synthetically generated scenes, including FlyingChairs [83], MPI-Sintel [40], Virtual KITTI [18], FlyingThings3D, Monkaa, Driving [55], BodyFlow [122], GTAV [123], SceneNet RGBD [124], and UvA-Nature [125]. Only the KITTI datasets [105, 110] provide optical flow for real-world images, with only 200 frames of car-driving scenes, most of which are rigid motion patterns.

The first attempt to generate a large-scale dataset suitable for training deep learning models is FlyingChairs [83]. Dosovitskiy *et al*. propose to use 2D images of chairs rendered from CAD models deformed by affine transformations. The first frame of a pair is created by randomly positioning multiple chair images on an image background. Then, the second frame is generated by warping each object, using a flow field generated by the affine model, with random parameters. While the parametric model is able to generate many images, the affine transformation yields rigid optical flow fields limiting the type of motion.

SlowFlow [126] contains natural videos with non-rigid motion. The method *estimates* optical flow for image sequences captured from high-resolution and high-speed cameras ($>$ 1440p resolution and $>$ 200 fps). However, the requirement of special recording devices as well as the potential inaccuracy in the estimated optical flow limits its applicability.

Data Augmentation Data augmentation entails a plethora of strategies to create more training data. It is commonly used in many tasks, including image classification [127], image segmentation [89], and depth estimation [128].

Widely used techniques for augmenting image data is to perform geometric augmentation (such as translation, rotation, and scaling) and color augmentation (such as changing brightness, contrast, gamma, and color). Data augmentation for optical flow networks is first proposed by [83] and studied in detail by [111]. The results show that both color and geometry types of augmentation are complementary and improve the performance. Inspired by these data augmentation techniques, we propose methods to increase the diversity of the generated optical flow data by texture augmentation.

In conclusion, large scale datasets with dense optical flow of non-rigid motion from real-world imagery are not available today. This is mainly due to the difficulty of human annotation to generate optical flow ground-truth. Instead, synthetic optical flow datasets with computer-generated imagery are created. To circumvent human annotation and the use of synthetic imagery data, we propose a framework to automatically generate dense non-rigid optical flow from real-world videos.

4.3 Generating Image Pairs for Optical Flow

In this section, we describe our approach to generate a dense optical flow field from a pair of images, see Figure 4.1. The framework is described in the following sections as follows:

3.1 *Image segmentation* to extract the object of interest in both frames;

3.2 *Image matching* to obtain corresponding points between the frames;

3.3 *Image deformation* to compute the flow field, guided by the correspondences;

3.4 *Warping* of the first object with the flow field to generate a warped object;

3.5 *Random background* on which we paste the first object and the warped object as an input pair for training, with the optical flow field as ground truth.

4.3.1 Image Segmentation

Our aim is to generate flow fields from non-rigid moving objects in videos. From a pair of sequential frames $\mathcal{I}_t$ and $\mathcal{I}_{t+\Delta}$ in a video sequence, where Δ is the frame distance ($\Delta = 1$ for consecutive frames), the objects of interest I_t and $I_{t+\Delta}$ are localized, see Figure 4.1.a.

We compare different ways to localize the objects, including ground truth segments and by using a pre-trained Mask R-CNN [129]. Segments can be the entire image frame and do not need to correspond to the objects precisely (see Section 4.4.3 for more details).

To increase the amount of variations in object motion, different offsets between frames (Δ) MPI-re explored (Section 4.4.1). The localization of objects is also used to replace textures while keeping their shapes (Section 4.4.2).

4.3.2 Image Matching

The generated flow fields should adhere to the non-rigid motion of the objects in videos. To steer the computation of the flow field, the statistics of the object motion are computed by finding image matches (or correspondences) between the segmented objects I_t, and $I_{t+\Delta}$. The step is illustrated in Figure 4.1.b.

We use the Deep Matching [82] algorithm to obtain a mapping of a set of point-to-point matches. We denote with $\mathcal{M}\left(\mathbf{x}_t^k\right) = \mathbf{x}_{t+\Delta}^k$ the map of the pixel coordinates of the k-th pixel in I_t to the corresponding pixel coordinate $\mathbf{x}_{t+\Delta}^k$ in $I_{t+\Delta}$. The obtained correspondences are quasi-dense and are robust to non-rigid deformations and repetitive textures.

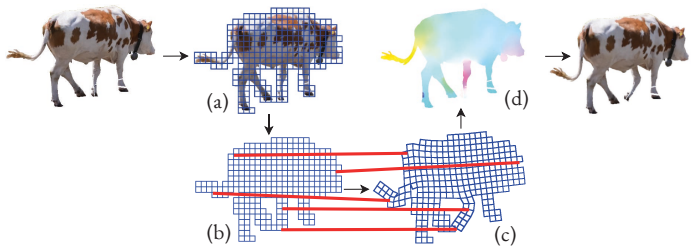


Figure 4.2: ARAP image deformation: (a) constructing a control lattice, (b) to (c) deforming the lattice steered by the image matches, (d) obtaining the flow field by interpolating the deformed lattice.

4.3.3 Image Deformation

To generate a dense flow-field, we deform the segmented object I_t to match with $I_{t+\Delta}$, using the obtained image matches $\mathcal{M}$ to guide the deformation process, see Figure 4.1.c. The *as-rigid-as-possible* (ARAP) [130, 131, 132, 133] principle is used to deform the objects. ARAP allows for large non-rigid deformations of the objects but still conforming to physical feasibility by minimizing scaling and shearing factors of the local image regions.

The deformation method is illustrated in Figure 4.2. First, we define a rectangular grid tightly bounding the object I_t, where each vertex corresponds to a pixel. Then, this grid is deformed (see Figure 4.2(b) to (c)) steered by image matches $\mathcal{M}$ and regularized by local deformations. Finally, the dense flow field $\tilde{F}_{t\to t+\Delta}$ is obtained by interpolating the vertices before and after deformation.

Mathematically, the image deformation process is formulated as an energy optimization problem over the grid structure. We minimize per image the energy of a data fitting term weighed with a regularizer:

$$E(\mathbf{d},\mathbf{R},\mathbf{x}_t,\mathcal{M}) = \sum_k w_{\text{fit}}\, E_{\text{fit}}(\mathbf{d}^k,\mathbf{x}_t^k,\mathcal{M}) + w_{\text{reg}}\, E_{\text{reg}}(\mathbf{d}^k,\mathbf{x}_t^k,\mathbf{R}^k), \tag{4.1}$$

where $\mathbf{d}$ denotes the deformed grid fitted to object $I_{t+\Delta}$. Following [132], we set $w_{\text{fit}} = 10$ and $w_{\text{reg}} = 0.1$. The data fit term is guided by the matches $\mathcal{M}$:

$$E_{\text{fit}}(\mathbf{d}^k,\mathbf{x}_t^k,\mathcal{M}) = \left|\mathbf{d}^k - \mathcal{M}(\mathbf{x}_t^k)\right|^2. \tag{4.2}$$

For pixel coordinates without an image match, $\mathcal{M}(\mathbf{x}_t^k) = \mathbf{x}_t^k$ is used instead.

As regularizer, the relative rigid rotation between neighboring pixels with rotation ma-

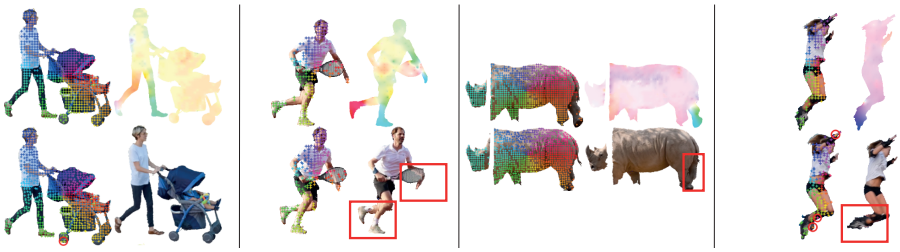


Figure 4.3: Illustration with four examples of image segments I_t and $I_{t+\Delta}$ annotated with point matches, the computed flow $\tilde{F}_{t\to t+\Delta}$ and the warped images $\tilde{I}_{t+\Delta}$. Note the significant differences between $I_{t+\Delta}$ and $\tilde{I}_{t+\Delta}$ (bottom row) due to the errors in the point matches. However $(I_t, \tilde{I}_{t+\Delta}, \tilde{F}_{t\to t+\Delta})$ form a correct triplet.

trices $\mathbf{R}^{kj} \in \mathbb{R}^{2\times 2}$ is used to enforce rigid rotation and translation [131], yielding:

$$E_{\text{reg}}(\mathbf{d}^k, \mathbf{x}_t^k, \mathbf{R}^k) = \frac{1}{4}\sum_{j=1}^{4} \left| \mathbf{R}^{kj}\left(\mathbf{x}_t^{kj} - \mathbf{x}_t^k\right) - \left(\mathbf{d}^{kj} - \mathbf{d}^k\right)\right|^2, \tag{4.3}$$

where $\mathbf{x}_t^{kj}$ denotes the j-th neighbor from pixel k and $\mathbf{d}^{kj}$ the coordinates after deformation of $\mathbf{x}_t^{kj}$. The four connected pixels to each pixel are used as neighbors.

Equation 4.1 is minimized with respect to $\mathbf{d}$ and $\mathbf{R}$, resulting in a non-linear least square problem, which is solved by the iterative Gauss-Newton method [133].

4.3.4 Image Warping

The dense optical flow field $\tilde{F}_{t\to t+\Delta}$ is obtained by interpolating $\mathbf{x}_t$ and $\mathbf{d}$. Due to possible errors introduced by the matching and deformation algorithm, it is only an approximation of the true field $F_{t\to t+\Delta}$, and hence it does not necessarily transform the object I_t to the exact shape of $I_{t+\Delta}$.

To generate correct triples, image warping $\tilde{I}_{t+\Delta} = \mathcal{W}(I_t, \tilde{F}_{t\to t+\Delta})$ is used, see Figure 4.1.d. This results in a correctly generated triple $(I_t, \tilde{I}_{t+\Delta}, \tilde{F}_{t\to t+\Delta})$*. In Figure 4.3, segmented objects I_t and $I_{t+\Delta}$ are illustrated with the obtained optical flow $\tilde{F}_{t\to t+\Delta}$ and the warped object $\tilde{I}_{t+\Delta}$, including some matching correspondence errors recovered by the warping process.

4.3.5 Background Generation

To obtain a full frame image pair, an object I_t and the warped object $\tilde{I}_{t+\Delta}$ are projected on a (static) background image (Figure 4.1.e). This background image is randomly sampled

*Image warping might introduce artifacts due to interpolation used, it is however commonly used, *e.g.* in the FlyingChairs dataset [83]

Algorithm 1 Generate optical flow from a generic video dataset $\mathcal{V}$.

Input: videos $\mathcal{V}$, frame-distance Δ, segmentation method S, texture method T
Output: optical flow dataset $\mathcal{F}$

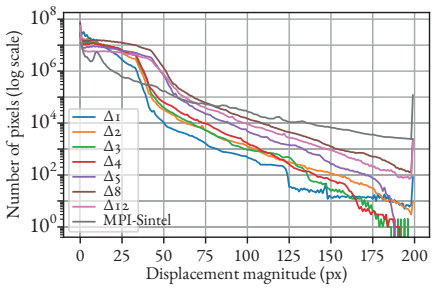
1: $\mathcal{I}_t, \mathcal{I}_{t+\Delta} \leftarrow$ sampled from $v \in \mathcal{V}$, with frame-distance Δ # Section 4.4.1
2: $I_t, I_{t+\Delta} \leftarrow$ from segmentation $\mathcal{I}_t$ and $\mathcal{I}_{t+\Delta}$ with algorithm S # Section 4.4.3
3: $\mathcal{M} \leftarrow$ image_matching$(I_t, I_{t+\Delta})$
4: **if** $\mathcal{M}$ **is** $\varnothing$: **skip frame**
5: $\tilde{F}_{t\rightarrow t+\Delta} \leftarrow$ ARAP deformation$(\mathcal{M}, I_t)$
6: $I_t \leftarrow$ replace texture of object I_t with method T # Section 4.4.2
7: $\tilde{I}_{t+\Delta} \leftarrow$ image warping $\mathcal{W}(I_t, \tilde{F}_{t\rightarrow t+\Delta})$
8: $\mathcal{I}_t, \tilde{\mathcal{I}}_{t+\Delta} \leftarrow$ pasting objects I_t and $\tilde{I}_{t+\Delta}$ on random background
9: $\mathcal{F} \leftarrow \mathcal{F} + \{(\mathcal{I}_t, \tilde{\mathcal{I}}_{t+\Delta}, \tilde{F}_{t\rightarrow t+\Delta})\}$

from a set of 8K images of general scenery obtained from Flickr images with a Public Domain license, similar to the approach used for the creation of the FlyingChairs dataset [83]. However, in contrast to the FlyingChairs dataset, static backgrounds are used instead of affine transformed backgrounds.

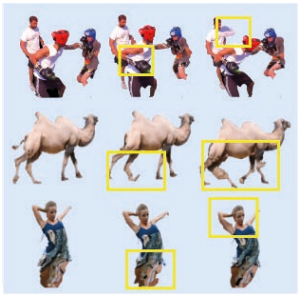
4.4 Generating the DAVIS-Mask-OpticalFlow Dataset

In this section, datasets generated from video frames using the proposed method are explored. See the pseudo-code in Algorithm 1. The algorithm takes a video dataset $\mathcal{V}$ as input, together with an integer number Δ for the frame distance, a segmentation method S, and a texture replacing method T. CNNs learn a better model when the training set consists of samples with a large variety of textures, motion patterns, and displacements [111]. Hence, the influence of different choices for Δ, S, and T to create datasets are explored.

The DAVIS [112] video dataset is used to generate optical flow datasets. DAVIS contains 6K frames of real imagery with provided segmentation masks. The generated optical flow datasets are used to train a FlowNet-S (FNS) model [83]. Evaluation is performed on a subset of 410 image pairs from the training set of MPI-Sintel [40], coined Sintel-val. Results are reported using the average *end-point-error* metric (EPE, lower is better). The obtained results are compared to a FlowNet-S model trained on the FlyingChairs [83] dataset as baseline. This dataset has 22K image pairs of chairs projected on different backgrounds with corresponding optical flow ground truths. FlowNet-S is chosen since it is fast to train, thus suitable for extensive experimentation. In Section 4.5, experiments are conducted with our final dataset using more recent architectures, and more diverse datasets.



(a) Displacement statistics



(b) Warped images

	Sintel-val
FC	5.75
Δ1	5.50
Δ1-2	5.37
Δ1-3	5.17
Δ1-4	5.10
Δ1-5	5.06
Δ1-5,8	**5.03**
Δ1-5,8,12	5.06

(c) EPE

Figure 4.4: Influence of the frame distance (Δ): Increasing the frame distance increases the motion magnitude (a), and introduces artifacts in the warped objects (b), yet increasing the dataset by adding frame distances up to $\Delta = 5$ is beneficial for performance(c). Larger frame distances have neglectable influence.

4.4.1 Displacement Variation

To increase the variation in object motion, different frame distances Δ in the video sequence are used. Larger Δ reflects motion further in time, resulting in more variation in the non-rigid motion statistics of the objects. This is reflected by the larger object displacement, see Figure 4.4a, and by the warped objects, see Figure 4.4b. Note, however, that larger frame distances also introduce artifacts, likely due to errors in the matching stage.

To study the influence of large frame distances, datasets generated with $\Delta = \{1, 2, \ldots, 12\}$ are combined into a single dataset. Despite the increase of training set, the images' appearances basically stay the same as they are extracted from the same set of videos. Thus, the performance gain can be attributed to the increased displacement.

The performance is given in Table 4.4c. From these results, it can be derived that increasing the frame distance is, in general, beneficial for EPE error on Sintel-val. We observe some diminishing gains, especially for $\Delta > 5$. This is attributed to the introduced artifacts in the warped images. More importantly, the results show that there is *no* need to strictly match the distributions of the training and testing sets to achieve the best performance [111]. For the remaining experiments Δ1-5 is used to generate optical flow, unless stated otherwise.

4.4.2 Texture Variation

Object re-texturing allows for increasing the variation in the datasets appearances, see Figure 4.5a. Moreover, re-textured objects enforce the network to disentangle semantic (class specific) information from optical flow information. This is likely to be beneficial for a generic (class agnostic) optical flow prediction model.

(a) Appearance variation

	T	Δ5 SINv	Δ5 SynR	Δ5 ReaR	Sintel-val
FC		6.28	7.29	6.92	5.10
Δ1	O	4.20	4.82	4.62	5.50
Δ1-4	O	4.02	4.75	4.42	5.10
Δ1-4	R	3.99	4.61	4.32	**4.96**
Δ1-4	C	**3.86**	**4.60**	**4.24**	4.98

(b) Performance

Figure 4.5: Re-textured objects: (a) examples of different textures and (b) performance analysis for different textures. Training with re-textured objects (both R and C) ensures better performance.

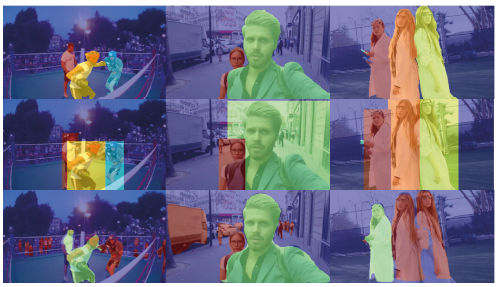
To re-texture objects, denoted by T in Algorithm 1, the following is done. After obtaining the flow field, the original texture (O) of I_t, is replaced by a new random texture (R), using the segmentation mask. The random texture is taken from the set of general natural images, used as background images. Then, the corresponding $\tilde{I}_{t+\Delta}$ is obtained by warping the re-textured I_t, using $\tilde{F}_{t\to t+\Delta}$. We explore using a re-textured dataset, denoted by R, and using a combination of original textures with random re-textures, denoted by C.

FlowNet-S is trained on the newly re-textured data (Δ1-4) and its robustness for unseen textures and displacements is studied. The models are evaluated on Δ5 and Sintel-val. The former has been re-textured with 3 texture types: *SynR*, synthetic images with repetitive patterns; *ReaR*, real images with repetitive patterns; and *SINv*, images from Sintel-val set, see examples of the re-textured images in Figure 4.5a.

The results are shown Table 4.5b. We conclude that training with re-textured data (R or C) is beneficial for good performance on both Δ5 and Sintel-val. The performance differences between the different re-textured datasets used in Δ5 show the dependency of the performance on the test images' texture. This confirms the hypothesis that the network needs to be trained with a wide variety of texture types. Hence, for the subsequent experiments, a combination of original-texture and re-textured images (C) is used.

4.4.3 Object Segmentation

In this section, different methods for selecting the object of interest are compared. So far, the ground truth segmentation masks have been used. Now, the following alternatives are considered: (1) selecting the entire frame as the object of interest; (2) using tight bounding boxes enclosing the ground-truth segments; (3) using the ground-truth segments; and (4) using segments from Mask R-CNN [129], a pre-trained off-the-shelf segmentation network. See Figure 4.6a for examples.



(a) Segment variations

Training set (Δ1-5-O)	Sintel-val
Entire frame (F)	5.69
Bounding boxes (B)	5.16
Ground truth segments (G)	5.06
Mask RCNN (M)	**4.91**

(b) Performance

Figure 4.6: Object Segmentation: (a) Examples of ground-truth segments (top), bounding boxes (middle) and Mask R-CNN predictions (bottom); and (b) Sintel-val performance. The approximately correct segments M yield best performance.

Entire-frame deformations include constraints from both backgrounds and foreground objects, which might limit the flexibility and variation in the generated deformation. The bounding boxes increase the segment sizes by including background parts while keeping the objects of interest in focus. For Mask R-CNN [129], the available pre-trained model is used, which is trained on the class labels of the MS-COCO [53] datasets. Due to uncertainties of inference, the Mask R-CNN segments may generate larger regions rather than strictly focusing on the centred objects like the ground-truth segments. This might result in creating a large variation in terms of object shapes and sizes.

The results of training FlowNet-S on the data generated using the original textures (Δ1-5-O), comparing different segmentation methods, are shown in Table 4.6b. From the results it is concluded that focusing on objects is beneficial wiht the performance of $F < B < G$. Surprisingly, the network trained with the dataset using Mask R-CNN segments outperforms the one using ground truth segments ($M > G$). This is because Mask R-CNN segments, in general, are more varied and cover more object types in a scene compared to the ground truth segments: not only the objects of interest, but also those in the background. Hence, it provides the network with a larger range of object variations, which shows to be useful for training. This indicates it is possible to use any real-world in-the-wild videos with Mask R-CNN segments for training optical flow deep networks. In the subsequent experiments, datasets using Mask R-CNN (M) are used.

4.4.4 Non-Rigid Motion Analysis

The Sintel movie is created using mostly static scenes and moving characters. In this section, the performance of the FNS models on non-rigid movements and occluded regions

	FlowNetS			LiteFlowNet			PWC-Net		
	Full	N-R	Occ	Full	N-R	Occ	Full	N-R	Occ
FC	5.09	14.56	12.32	4.32	14.70	12.86	4.01	13.52	11.27
Δ1-5-O-M	4.96	13.88	12.64	4.34	**13.59**	12.88	3.88	12.59	11.21
Δ1-5-C-M	**4.54**	**13.28**	**11.82**	**4.23**	13.83	**12.79**	**3.62**	**11.90**	**10.57**

Table 4.1: Non-Rigid Motion Analysis: Performance on subsets of Sintel-val, using the full image (F), non-rigid motion (N-R) or occluded regions (Occ), for three different architectures: FlowNet-S (FNS), LiteFlowNet (LFN), and PWC-Net (PWC). Deep networks trained on our non-rigid motion datasets outperform those trained on the FlyingChairs dataset.

is evaluated. Evaluation is performed over (a) foreground objects using the segments provided by [40]; and (b) occluded regions, defined as those regions which appear in one of the two input frames only.

Different training sets are compared (FC, Δ1-5-O-M, and Δ1-5-C-M), using FlowNet-S [83], PWC-Net [109], and LiteFlowNet [108]. The results are shown in Table 4.1. We conclude that training on our non-rigid motion datasets outperforms training on the rigid transformations from the FlyingChairs dataset. This holds especially for non-rigid and occluded regions in the images. To conclude, non-rigid motion is a necessity to train robust CNNs for optical flow prediction.

4.4.5 Training Dataset Size

In this section, the performance is evaluated as a function of the number of images in the training dataset. This aims to distinguish improvements based on the sheer number of

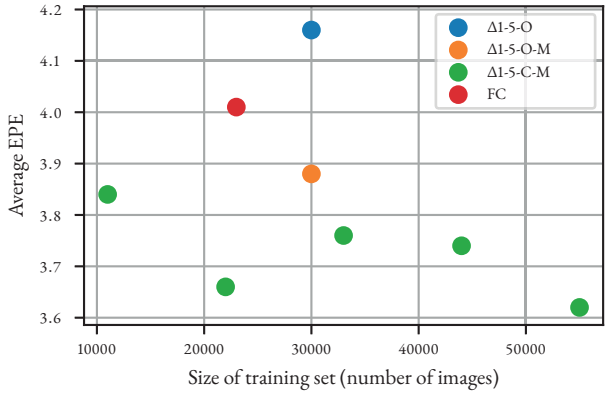


Figure 4.7: Size of Training Set: Performance of PWC-Net [109] trained on differently sized datasets on Sintel-val. For Δ1-5-C-M, we include results of sub-sampled datasets (indicated in green). Our generated datasets using Mask R-CNN consistently outperform the approaches of training on FlyingChairs, regardless of the actual training size.

annotated training examples, from the improvements based on non-rigid motion and texture variations. The 55K images from our Δ1-5-C-M dataset are sub-sampled to generate training sets with 11K, 22K, . . ., 55K examples.

Figure 4.7 shows the results on Sintel-val of PWC-Net trained on these sub-sampled datasets. It can be observed that the training set size does have an influence on the performance. Network trained on Δ1-5-C-M is gradually improving as the size of training data increases (with 22K being the outlier likely due to sub-sampling effects). The results show that regardless of size this dataset, training on Δ1-5-C-M performs best.

4.4.6 Discussion

From these extensive, yet initial, analyses of various design choices, we derive that the generated datasets with non-rigid optical flow fields are well suitable for training CNNs for optical flow prediction. In the next section, the Δ1-5-C-M dataset, generated using Mask R-CNN segments, using original and re-textured objects, is used. Which we coin the DAVIS-Mask-OpticalFlow (**DMO**) dataset.

4.5 Experiments

In this section, experiments are conducted to compare models trained on the proposed DMO dataset to various state-of-the-art baselines and benchmarks.

4.5.1 Experimental Setup

Datasets For most of the experiments, models trained on the FlyingChairs dataset [83] are used as baseline comparison. This allows to study the effect of the training set on the performance of different network architectures.

Evaluation is performed on the test-set of MPI-Sintel [40] containing large displacements of non-rigid optical flow. Additional evaluation is performed on the validation split of the HumanFlow [122], which contains 530 image pairs of non-rigid motion of human bodies, and on a subset of 50 randomly selected images from the KITTI 2012 [105] and 2015 [110] training set containing real-textures from a self driving car.

As both MPI-Sintel and HumanFlow contain CGI-rendered images, while KITTI contains mostly rigid motion, the QUVA repetition dataset [134] is used to qualitatively evaluate non-rigid motion on real imagery. The dataset consists of video sequences of repetitive activities with minor camera motion, thus mostly consisting of non-rigid object motions.

		Sintel-test		Sintel-test occ		HumanFlow	KITTI val	
		final	clean	final	clean		2012	2015
Zero flow		-	-	-	-	0.73	28.23	24.03
FNS	FC	8.16	7.17	35.88	34.02	0.63	4.63	7.71
	DMO	**7.64**	**6.61**	**34.98**	**33.17**	**0.36**	**3.53**	**5.30**
LFN	FC	7.89	6.77	38.79	37.28	0.30	2.75	7.61
	DMO	**7.73**	**6.50**	**38.68**	**36.30**	**0.26**	**2.73**	**6.27**
PWC	FC	6.97	5.61	33.58	30.61	0.30	2.22	5.36
	DMO	**6.62**	**5.52**	**31.56**	**30.00**	**0.26**	**1.72**	**3.18**

Table 4.2: Comparison of different models (FNS, PWC, LFN) trained on FC and DMO, evaluated on Sintel benchmark, Human Flow, and KITTI. The constant zero flow indicates the displacement statistics of the test sets. For all networks and all evaluations hold that training on non-rigid motion data (DMO) outperforms training on rigid/affine motion (FC).

Network Architectures Different CNN architectures are compared, namely: (i) FlowNet-S [83] which is also used in Section 4.4, (ii) PWC-Net [109] and LiteFlowNet [108] as recent supervised models, (iii) three unsupervised architectures, specifically MFOF [135], DDFlow [136], and SelFlow [121]. For each model, the standard training settings are used, including data augmentation and learning schemes as provided by the authors.

4.5.2 Comparison to State-of-the-Art

We compare different state-of-the-art algorithms for optical flow, namely LiteFlowNet (LFN) [108] and PWC-Net (PWC) [109]. For each network, we compare the performance between the models trained with DMO dataset to the same model trained on FlyingChairs. The results are evaluated on the MPI-Sintel benchmark server (Sintel-test), the HumanFlow and KITTI 2012, 2015 datasets. The results are summarized in Table 4.2.

The networks trained with our dataset outperform those trained with FlyingChairs on all tests. The results on Sintel occluded regions show that the proposed dataset improves models' robustness even in the challenging occlusion conditions. The results on Human Flow show that non-rigid optical flow estimation (e.g. human body movement) benefits from DMO. In particular, although FNS is well-known for poor performance on small-displacement data [122], the performance on HumanFlow trained with DMO is close to that of the other powerful algorithms, whereas FNS trained on FlyingChairs is close to zero-flow. This suggests that the weakness of the methods can be improved using the proposed

	Training dataset		Sintel-test		Sintel-test occ	
			final	clean	final	clean
MFOF [135]	RoamingImages	T	8.81	7.23	39.70	36.78
DDFlow [136]	FC	T	7.40	6.18	39.94	38.05
SelFlow [121]	Sintel movie	U	6.57	6.56	34.72	38.30
SelFlow-ft [121]	Sintel movie → KITTI → Sintel	Ft	**4.26**	**3.74**	**22.37**	**22.50**
PWC	FC	T	6.97	5.61	33.58	30.61
PWC	FC → Sintel	Ft	6.22	5.38	30.46	29.69
PWC	DMO	U/T	6.62	5.52	31.56	30.00
PWC	DMO → Sintel	Ft	**5.86**	**5.26**	**29.09**	**29.75**

Table 4.3: Comparison to unsupervised (U), transferred (T) and fine-tuned methods (Ft). PWC trained on DMO outperforms all the transferred methods and compares favourably to fine-tuned methods after pre-training on FC. The gap between unsupervised and transferred methods and fine-tuned methods indicate the necessity of annotated optical flow.

dataset. The results on KITTI val set indicate how the networks perform on real-world texture images. Although KITTIs contain mostly rigid motion (cars moving on streets) with sparse resolution, the network trained with our dataset still outperforms those trained with FlyingChairs.

We conclude that optical flow CNNs benefit from training on natural textures and non-rigid movements, as generated by our dense optical flow method.

4.5.3 Comparison to unsupervised and finetuned methods

In this section, we measure the ability of our dataset DMO to transfer to different domains. To this end, we compare PWC trained on our dataset with unsupervised methods such as MFOF [135], DDFlow [136], and SelFlow [121]. We train the network on a dataset generated in an unsupervised way, without using any ground truth optical flow from the test domain, i.e. the MPI-Sintel dataset. We also show the results of finetuning on the Sintel train set with results reported by finetuned state-of-the-art methods. The results are shown in Table 4.3.

For the transferred methods, PWC trained on our dataset outperforms the others, showing the transferrability of our method. For the finetuning methods, SelFlow performs the best. Note its improvement of the supervised results (Ft) over the unsupervised (U) and how it is mainly trained on Sintel-related data. This shows the necessity of having annotated data suited for the target domain.

(a) LiteFlowNet *(b) PWC-Net*

Figure 4.8: Qualitative results on QUVA dataset [134] for LFN (a) and PWC (b), trained on FC (midlle) and DMO (bottom). The networks trained using our pipeline capture the non-rigid motion of objects in the scenes with higher detail and delineation. (Best viewed in color.)

4.5.4 Performance on real-world images

As there are currently no optical flow benchmarks with real-texture and non-rigid motion, the QUVA repetition dataset [134] is employed to qualitatively demonstrate the effectiveness of our generated dataset. Figure 6.6 shows the optical flow prediction by LiteFlowNet (top) and PWC-Net (bottom) trained with FlyingChairs and our DMO. The models trained with our non-rigid flow set capture better the objects' delineation and details, especially for non-rigid movements of human body parts indicated by the color changes.

4.6 Conclusions

In this chapter, we introduced a pipeline to generate densely annotated optical flow datasets from videos to train supervised deep networks for optical flow.

Extensive experimental results show that it is possible to create a large amount of data with optical flow ground truths from real-world videos using off-the-shelf segmentation algorithms (e.g. Mask R-CNN). Increasing of training set size, in general, improves CNNs performance regardless of architectures. The generated data from the proposed framework shows superiority over the commonly used FlyingChairs for pre-training networks.

5 Novel View Synthesis from Single Images via Point Cloud Transformation

In this chapter the argument is made that for true novel view synthesis of an object, where the object can be synthesized from any viewpoint, an explicit 3D shape representation is desired. Our method captures the object's geometry by reconstructing a partial point cloud from a predicted depth map, which can be freely rotated into the target viewpoint and projected on the image space. This coarse image is completed by a generative adversarial network to obtain the dense target view. The image completion and depth prediction networks can be trained end-to-end without depth supervision. The benefit of using point clouds as an explicit 3D shape for novel view synthesis is experimentally validated on the 3D ShapeNet benchmark.

5.1 Introduction

Novel view synthesis infer the appearances of an object from unobserved points of view. The synthesis of unseen views of objects could be important for image-based 3D object manipulation [137], robot traversability [138], or 3D object reconstruction [139]. Generating coherent views of objects' unseen parts requires non-trivial understanding of the object's inherent properties such as (3D) geometry, texture, shading, and illumination.

Different algorithms make use of provided source images in different ways. Model-based approaches use similar-look open stock 3D models [137], or through user interac-

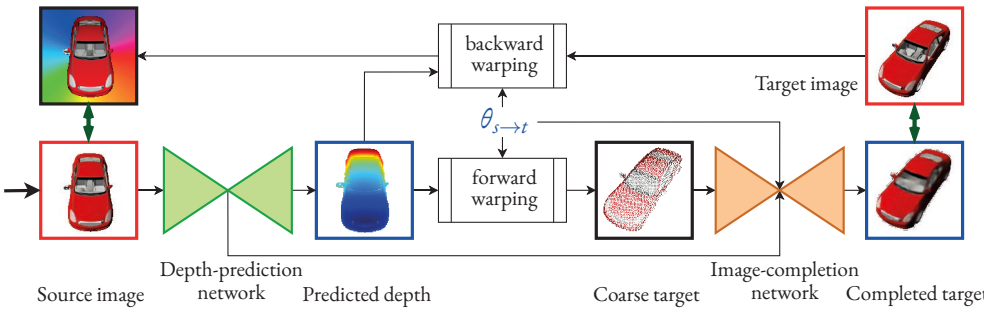


Figure 5.1: Overview of the proposed model for training and inference. From a single input image, the pixel-wise depth map is predicted. The depth map is subsequently used to compute a coarse novel view (forward warping), and trained by making use of backward warping (from the target view back to the source view). The model is trained end-to-end.

tive construction [140, 47, 141]. Image-based methods [139, 142, 143, 144, 145] assume an underlying parametric model of object appearances conditioned on viewpoints and try to learn it using statistical frameworks. Despite their differences, both approaches use 3D information in predicting object new views. The former imposes stronger assumptions on the full 3D structure and shifts the paradigm to obtain the full models, while the latter captures the 3D information in latent space to cope with (self) occlusion.

The principle is that the generation of a new view of an object is composed of (1) relocating pixels in source images that will be visible in the target view to the corresponding positions, (2) removing the pixels that will be occluded, and (3) adding disoccluded pixels that are not in the source but will be revealed in the target view [143]. With the advance of convolution neural networks (CNNs) and generative adversarial networks (GANs), [142, 143, 144] show that (1) and (2) can be done by learning an appearance flow field that "flows" pixels from a source image to the corresponding positions in the target view, and (3) can be done by a completion network with an adversarial loss.

In this chapter, we leverage the explicit use of geometric information and show that objects' geometry provides the natural basis for the problem of novel view synthesis. We argue that (1) and (2) can be done in a straightforward manner by having access to the geometry of the objects: the appearance flows [142, 143, 144] which associate pixels of the source view to their positions in the target view are, indeed, the projection of the 3D displacement of objects' points before and after transformation; occluded regions can be identified based on the surface normals' orientation and the view directions. The same arguments can also be extended naturally to multiple input images.

In contrast to geometry-based methods, the proposed approach does not require 3D supervision. The method predicts a depth map in a self-supervised manner by formulating

the depth estimation problem in the context of novel view synthesis. The predicted depth is used to partly construct the target views and to assist the completion network.

The main contributions of this chapter are: (1) a novel methodology for novel view synthesis using explicit transformations of estimated point clouds; (2) an integrated model combining self-supervised monocular depth estimation and novel view synthesis, which can be trained end-to-end; (3) natural extensions to multi-view inputs and full point cloud reconstruction from a single image; and (4) experimental benchmarking to validate the proposed method, which outperforms the current state-of-the art for novel view synthesis.

5.2 Related Work

5.2.1 Geometry-based view synthesis

View synthesis via 3D models Full models (textured meshes or colored point clouds) of objects or scenes are constructed from multiple images taken from various viewpoints [146, 147, 148] or are given and aligned interactively by users [137, 141]. The use of 3D models allows for extreme pose estimation, re-texturing and flexible (re-)lighting by applying rendering techniques [149, 148]. However, obtaining complete 3D models of objects or scenes is a challenging task in itself. Therefore, these approaches require additional user input to identify objects boundaries [140, 47], select and align 3D models with image views [137, 141], or use simple textured-mapped 3-planar billboard models [150]. In contrast, the proposed method makes use of objects partial point clouds constructed from a given source view and does not require a predefined (explicit) 3D model.

View synthesis via depth Methods using 3D models assume a coherent structure between the desired objects and the obtained 3D models [47, 137]. To relax the need of obtaining full 3D models, depth images are used as an intermediate representation to capture hidden surfaces from one or multiple viewpoints. [151] proposes to use layered depth images, [152] creates 3D plane sweep volumes by projecting images onto target viewpoints at different depths, [153] uses multi-plane images at fix-distances to the camera, and [154] estimates depth probability volumes to leverage depth uncertainty in occluded regions.

In this chapter, we estimate depth directly from monocular views to construct the objects' partial point clouds. Self-supervised monocular depth estimation is an active research topic [155, 156, 157, 158, 159]. We show that self-supervised depth prediction and novel view synthesis can be trained in an end-to-end system.

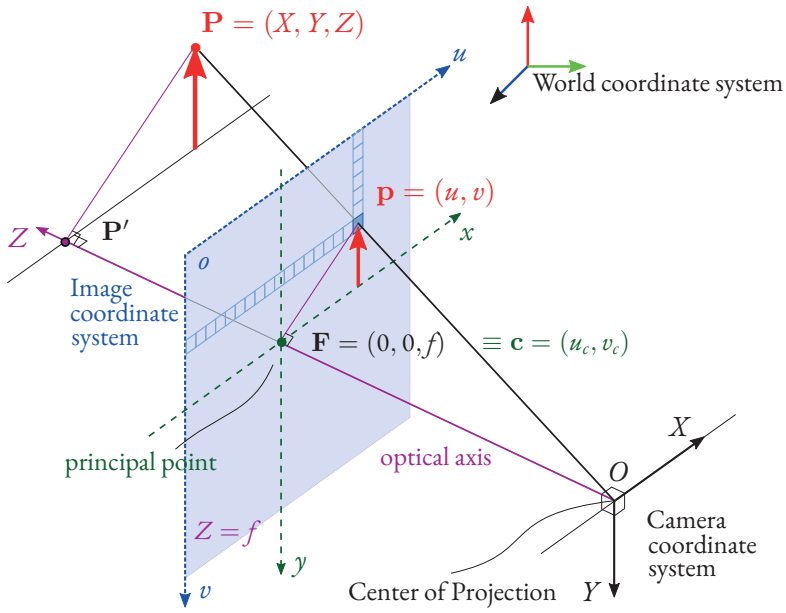


Figure 5.2: Image formation with pinhole camera model: the point **P** in the 3D space forms the image **p** on the image plane.

5.2.2 Image-based view synthesis

Requiring explicit geometrical structures of objects or scenes as a precursor severely limits the applicability of a method. With the advance of neural networks (CNNs), generative adversarial networks [160] (GANs) achieve impressive results in image generation, allowing view synthesis without explicit geometrical structures of objects or scenes.

View synthesis via embedded geometry Zhou *et al.* [142] proposes learning a flow field that maps pixels in input images to their corresponding locations in target views to capture latent geometrical information. [145] learns a volumetric representation in a transformable bottleneck layer, which can generate corresponding views for arbitrary transformations. The former explicitly utilizes input (source) image pixels in constructing new views, either fully [142], or partly with the rest being filled by a completion network [143, 144]. The latter explicitly applies transformations on the volumetric representation in latent space and generates new views by means of pixel generation networks.

The proposed method takes the best of both worlds. By directly using object geometry the source pixels are mapped to their target positions based on given transformation parameters, hence making the best use of the given information synthesizing new views. Our approach is fundamentally different from [143]: we estimate the object point cloud

using self-supervised depth predictions and obtain coarse target views from purely geometrical transformations, while [143] learns mappings from input images and ground truth occluded regions to generate coarse target views using one-hot encoded vectors.

View synthesis directly from image Since the introduction of image-to-image translation [161], there is a paradigm shift towards pure image-based approaches [139]. [162] synthesizes bird view images from a single frontal view image, while [163] generates cross-views of aerial and street-view images. The networks can be trained to predict all the views in an orbit from a single-view object [164, 159], or generate a view in an iterative manner [165]. Additional features can be embedded such as view-independent intrinsic properties of objects [166]. In this chapter, we employ GANs to generate complete views, which is conditioned on the geometrical features and the relative poses between source and target views. Our approach can be interpreted as a reverse and end-to-end process of [159]: we estimate objects' arbitrary new views via point clouds constructed from self-supervised depth maps, while [159] predict objects' fixed orbit views for 3D reconstruction.

5.3 Proposed Method

5.3.1 Point-Cloud based Transformations

The core of the proposed novel view synthesis method is to use point clouds for geometrically aware transformations. Using the pinhole camera model as shown in Figure 5.2 and known camera intrinsics $\mathbf{K}$, the point cloud can be reconstructed when the pixel-wise depth map (D) is available. The camera intrinsics can be obtained by camera calibration, yet for the synthetic data used in our experiments, $\mathbf{K}$ is given. A pixel on the source image plane $\mathbf{p}_s = [u \quad v \quad 1]^\top$ (using homogeneous coordinates), corresponds to a point $\mathbf{P}_s = [X \quad Y \quad Z]^\top$ in the source camera space:

$$D_s\,\mathbf{p}_s^\top = \mathbf{K}\,\mathbf{P}_s^\top \qquad\qquad \mathbf{P}_s^\top = \mathbf{K}^{-1}\,D_s\,\mathbf{p}_s^\top \tag{5.1}$$

Rigid transformations can be obtained by matrix multiplications. The relative transformation to the *target* viewpoint from the *source* camera, is given by:

$$\theta_{s\to t} = \left[\begin{array}{c|c} \mathbf{R} & \mathbf{t} \\ \hline 0 & 1 \end{array}\right] \tag{5.2}$$

where $\mathbf{R}$ denotes the desired rotation matrix and $\mathbf{t}$ the translation vector. Points in the target camera view are given by $\mathbf{P}_t = \theta_{s\to t}\mathbf{P}_s$. This can also be regarded as an image-based

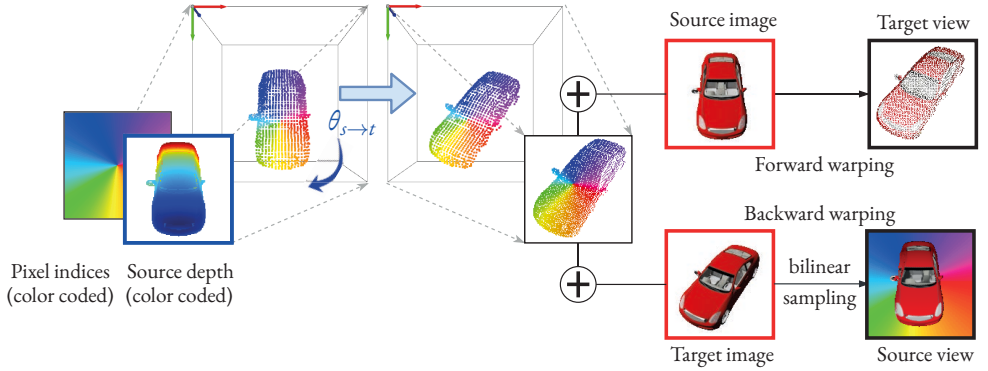


Figure 5.3: Illustration of the forward and backward warping operation of point clouds. The forward warping is used to generate a coarse target view, while the backward warping is used to reconstruct the source view from a target view for self-supervised depth estimation.

flow field $\varphi : \mathbb{R}^2 \to \mathbb{R}^2$ parameterized by $\theta_{s\to t}$ (*c.f.* [142, 143, 144]). The flow field $\varphi(\mathbf{p}_s; \theta_{s\to t})$ returns the homogeneous coordinates of the pixels in the target image for each pixel in the source image:

$$\varphi(\mathbf{p}_s; \theta_{s\to t}) = \mathbf{K}\, \theta_{s\to t}\, \mathbf{K}^{-1}\, D_s \mathbf{p}_s^{\top} \tag{5.3}$$

By observing that $\varphi(\mathbf{p}_s; \theta_{s\to t}) = D_t \mathbf{p}_t$, the Cartesian pixel coordinates in the target view can be extracted. The advantage of the flow field interpretation is that it provides a direct mapping between the image planes of the source view and the target view.

Forward warping The flow field is used to generate the target view from the source:

$$\tilde{I}_t(\varphi(\mathbf{p}_s; \theta_{s\to t})) = I_s(\mathbf{p}_s). \tag{5.4}$$

The resulted image is sparse and contains missing details due to discrete coordinates and occlusion (see Figure 5.3 *top-right*). The image is completed by a network (Section 5.3.2).

Backward warping The flow field is used to generate the source view from the target:

$$\tilde{I}_s(\mathbf{p}_s) = I_t(\varphi(\mathbf{p}_s; \theta_{s\to t})). \tag{5.5}$$

The process assigns a value to every pixel (u, v) in $\tilde{I}_s$ resulting in a dense image, as illustrated in Figure 5.3 (*bottom-right*). The generated source view may contain artifacts due to (dis)occlusion in the target view. To sample $\varphi(\mathbf{p}_s; \theta_{s\to t})$ from I_t, a differentiable bi-linear sampling layer [167] is used. The generated source view is used for self-supervised monocular depth prediction (Section 5.3.3).

5.3.2 Novel View Synthesis

The point-cloud-based forward warping relocates the visible pixels of the object in the source view to their corresponding positions in the target view. For novel view synthesis, however, two more steps are required: (1) obtaining the target coarse view by discarding occluded pixels, and (2) filling in the pixels that are not seen in the source view.

Coarse view construction The goal is to remove the visible pixels in the source image which will be occluded in the target view. To this end, pixels that have surface normals (after transformation) pointing away from the viewing direction are removed, similarly to [143]. Surface normals are obtained from normalized depth gradients.

An illustration of the coarse view construction is shown in Figure 5.4 for different target views. The first row depicts the target views, the second row indicates the visible parts from the input image (third column). The third and fourth row show the coarse view with and without occlusion removal (or backface culling). Finally, the fifth row shows an enhanced version of the coarse view, where the object is assumed to be left-right symmetric [143]. The proposed method directly identifies and removes occlusion pixels from the input view using *estimated* depth, which contrasts to [143], where ground truth visibility mask are required for each target view to train a visibility prediction network.

View completion The obtained coarse view is already in the target viewpoint, but it remains sparse. To synthesize the final dense image, an image completion network is used.

The completion network uses the hour-glass architecture [168]. Following [143], we concatenate the depth bottleneck features and embedded transformation to the completion network bottleneck. By conditioning the completion network on the input features and the desired transformation $\theta_{s\to t}$, the network can fix artifacts and errors due to estimated depth and cope better with extreme pose transformations, *i.e.* when coarse view image is near empty (*e.g.* columns 9-11 in Figure 5.4).

The image completion network is trained in a GAN-manner by using a generator $\mathcal{G}$, a discriminator $\mathcal{D}$, an input image I_s and a target image I_t. The combination of losses that are used is given by:

$$\mathcal{L}_{\mathcal{D}} = (\mathcal{D}(I_s) - 1)^2 + \mathcal{D}(\mathcal{G}(I_s)))^2, \qquad \text{Discriminator loss} \qquad (5.6)$$

$$\mathcal{L}_{\mathcal{G}} = [1 - \text{SSIM}(I_t, \mathcal{G}(I_s))] + \|I_t - \mathcal{G}(I_s)\|_1, \qquad \text{Generator loss} \qquad (5.7)$$

$$\mathcal{L}_{Perc} = \left\|\mathcal{F}^{\mathcal{D}}_{I_t} - \mathcal{F}^{\mathcal{D}}_{\mathcal{G}(I_s)}\right\|_2 + \left\|\mathcal{F}^{\text{VGG}}_{I_t} - \mathcal{F}^{\text{VGG}}_{\mathcal{G}(I_s)}\right\|_2, \qquad \text{Perceptual loss} \qquad (5.8)$$

where the perceptual loss uses $\mathcal{F}^{\mathcal{D}}$ and $\mathcal{F}^{\text{VGG}}$ to denote features extracted from image

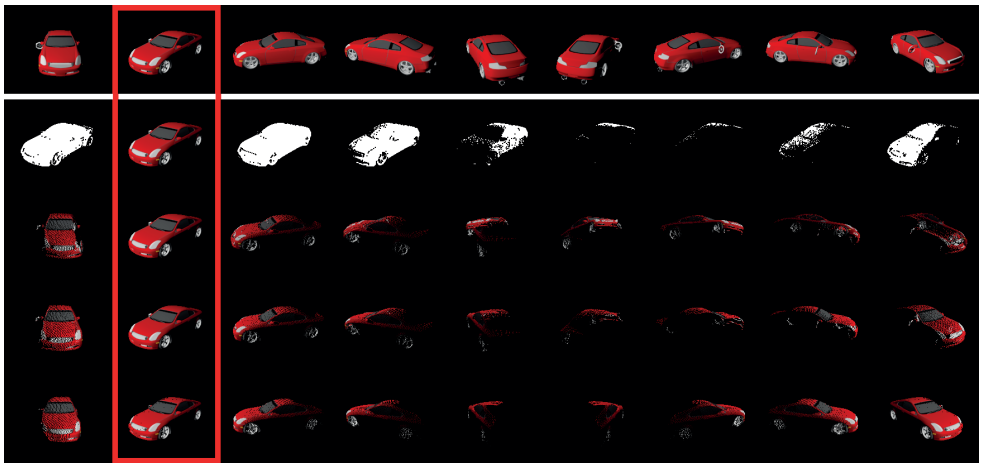


Figure 5.4: Image coarse views for different target viewpoints. The input image is depicted in the third column (red box). From top to bottom: (1) target views, (2) source region visible in each target viewpoint, coarse view (3) naive (4) with occlusion removal, and (5) with occlusion removal and symmetry.

I_t and $\mathcal{G}(I_s)$ from the discriminator network and pre-trained VGG network respectively, *c.f.* [169]. SSIM is the structural similarity index measure, see Section 5.4.

The total loss is given by:

$$\mathcal{L}_c = w_1\mathcal{L}_{\mathcal{D}} + w_2\mathcal{L}_{\mathcal{G}} + w_3\mathcal{L}_{Perc}, \tag{5.9}$$

where w denotes the weighting of the losses ($w_1 = 1$, $w_2 = 100$, and $w_3 = 100$, *c.f.* [143]).

5.3.3 Self-supervised Monocular Depth estimation

The discussion so far has assumed that pixel-wise depth maps are available. In this section, the method used to estimate depth from a single *RGB* image is detailed. In order to make the minimum assumption about the training data, self-supervised methods are considered, which do not require ground-truth depth [155, 156, 157, 158, 159].

For the depth prediction an encoder-decoder network with bottleneck architecture is used, similar to [158]. The network is optimised using a set of (reconstruction) losses between the source image I_s and its synthesized version $\tilde{I}_s$, using the backward warping, Equation (5.5), from a second (target) image I_t and the predicted depth map. The underlying rationale is that a more realistic depth map will have a lower reconstruction loss.

The losses are as follows:

$$\mathcal{L}_p\left(I_s, \tilde{I}_s\right) = \frac{\alpha}{2}\left[1 - \text{SSIM}\left(I_s, \tilde{I}_s\right)\right] + (1-\alpha)\left\|I_s - \tilde{I}_s\right\|_1, \quad \text{Photometric loss} \quad (5.10)$$

$$\mathcal{L}_s(d) = \left|\partial_x d\right| e^{-\left|\partial_x I_s\right|} + \left|\partial_y d\right| e^{-\left|\partial_y I_s\right|}, \quad \text{Smoothness loss} \quad (5.11)$$

$$\mathcal{L}_d\left(I_s, \tilde{I}_s, d\right) = \mu\mathcal{L}_p\left(I_s, \tilde{I}_s\right) + w_d\mathcal{L}_s, \quad \text{Total loss} \quad (5.12)$$

where $\alpha = 0.85$, $w_d = 10^{-3}$, $d = \frac{\overline{D}}{D}$ is the mean-normalized inverse depth, and μ is an indicator function which equals 1 iff $\mathcal{L}_p(I_s, \tilde{I}_s) < \mathcal{L}_p(I_s, I_t)$, see [158] for more details. The smoothness loss [157] encourages nearby pixels to have similar depths, while the artifacts due to (dis)occlusion are excluded by the per-pixel minimum-projection mechanism.

5.4 Experiments

In this section, the proposed method is analysed on the 3D ShapeNet benchmark including an ablation study and comparrison to state-of-the-art.

Dataset We use the object-centered car and chair images rendered from the 3D ShapeNet models [39] using the same render engine* and set up as in [142, 143, 144, 145]. Specifically, there are 7497 car and 698 chair models with high-quality textures, split by 80%/20% for training and test. The images are rendered at 18 azimuth angles (in $[0, 340]$, 20°-separation) and 3 elevation angles ($0°, 10°, 20°$). Input and output images are of size 256×256.

Metrics We evaluate the generated images using the standard L_1 pixel-wise error (normalized to the ranged $[0, 1]$, lower is better) and the structural similarity index measure (SSIM) [170] (value range of $[-1, 1]$, higher is better). L_1 indicates the proximity of pixel values between a completed image and the target, while SSIM measures the perceived quality and structural similarity between the images.

Baseline We compare the results of our method with the following state-of-the-art methods: AFN [142], TVSN [143], M2NV [144], and TBN [145].

5.4.1 Initial Experiments

Comparison to image-based completion In this section, we compare the intermediate views generated by the forward warping using estimated point clouds and those by image-based flow field prediction by DOAFN [143] and M2NV [144]. For this experiment, the coarse view after occlusion removal and left-right symmetric enhancements are used. The

*The specific render engine and setup is to guarantee fair comparison with reported methods as none of the authors-provided weights perform at the similar level on images rendered with different rendering setups.

(a) Coarse vs Completed

	Coarse view		Completed view	
	L1 (↓)	SSIM (↑)	L1 (↓)	SSIM (↑)
DOAFN [143]	.220	.876	.121	.910
M2NV [144]	.226	.879	.154	.906
Ours	**.203**	**.882**	**.118**	**.924**

(b) Ablation study

LS	PL	Sym	IC	SL	L1 (↓)	SSIM (↑)
	✓	✓			.118	.924
✓	✓	✓			.101	.939
✓		✓			.103	.939
✓	✓				.107	.933
✓	✓	✓	✓		**.097**	.939
✓	✓	✓	✓	✓	**.097**	**.942**

Table 5.1: Quality of coarse and completed view (*a*) and ablation study (*b*). The proposed transformation within estimated point clouds generates better coarse images over image-based predicted flow fields. Subsequently, the completed view quality is improved. The ablation study shows the best results are obtained by using LSGAN (LS), perceptual loss (PL), symmetry (Sym), bottleneck inter-connection (IC), and SSIM loss (SL).

image completion network is the basis variant, using DCGAN, without bottleneck inter-connections. The results are shown in Table 5.1a. The transformation of estimated point clouds provides coarse views which are closer to the target view, and these help to obtain a higher quality of completed views.

Ablation Study We analyze the effects of the different component of the proposed pipeline. The results are shown in Table 5.1b. The use of the LSGAN loss shows a relative large improvement over the traditional DCGAN. The drop of performance by removing symmetry assumption shows the importance of prior knowledge on target objects, which is intuitive. The inter-connection from the depth network and the embedded transformation to the completion network allow the model to not rely solely on intermediate views. This is important for overcoming errors and artifacts which occur in the coarse images (due to inevitable uncertainties in depth prediction) and generate in general higher quality images. The SSIM loss, first employed by [145], shows improvement in SSIM metric, which is intuitive as training objectives are closer to evaluation metrics.

5.4.2 Comparison to State-of-the-Art

In this section, the proposed method is compared with state-of-the-art methods. The quantitative results are shown in Table 5.2. The proposed method performs consistently performs (slightly) better on both evaluation metrics for both types of objects. The qualitative results are shown in Figure 5.5 where challenging cases are shown in the last 2 rows. Notice the better ability in retaining objects' textures (such as color patterns and texts on cars) of methods that explicitly use input pixel values in generating new views to that of TBN.

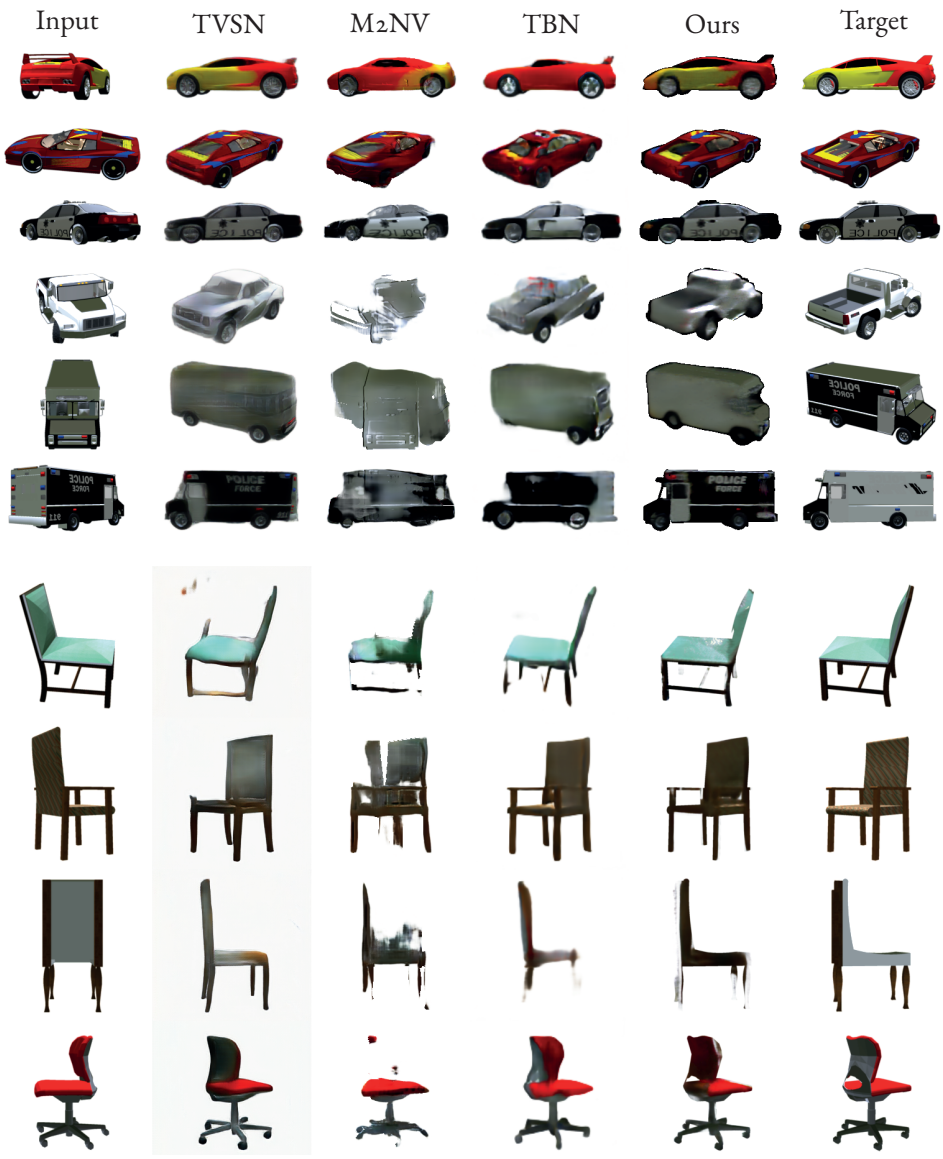


Figure 5.5: Qualitative comparisons of synthesized cars (*top*) and chairs (), given a single input image (*first column*) and a given target view (*last column*). The last two rows show a more challenging examples. The proposed method captures better the geometry of the object and the fine (texture) details.

Methods	cars		chairs	
	L1 (↓)	SSIM (↑)	L1 (↓)	SSIM (↑)
Same elevation				
AFN [142]	.148	.877	.229	.871
TVSN [143]	.119	.913	.202	.889
M2VN [144]	.098	.923	.181	.895
TBN [145]	.091	.927	.178	.895
Ours	**.096**	**.945**	**.175**	**.914**
Cross-elevation				
TBN	.199	.910	.215	.902
Ours	**.122**	**.934**	**.207**	**.905**

Table 5.2: Quantitative comparison with state-of-the-art methods on novel view synthesis: our method consistently performs (slightly) better than the other methods for both categories where target views have the same or different elevation angles with input views.

The results of cars are constantly higher than that of chairs due to the intricate structures of chairs. However, by having access to object geometry, geometrical assumptions such as symmetry and occlusion can be applied directly to intermediate views (instead of having to learn from annotated data *c.f.* [143]), improving the results for near-to symmetry targets.

Table 5.2 also shows the evaluation when target viewpoints are from different elevation angles. Methods such AFN, TVSN, and M2NV encode transformation as one-hot vectors and thus, are limited to operate within a pre-defined set of transformations (18 azimuth angles, same elevation). This is not the case for our method and TBN which apply direct transformation. We use the same azimuth angles as in the standard test set while randomly sample new elevation angles for input images in $(0°, 10°, 20°)$. The results are shown with networks trained with the regular fixed-elevation settings. The new transformations produces different statistics from what the networks have been trained, resulting in a performance drop for both methods. Nevertheless, the proposed method can still maintain high quality image synthesis.

5.4.3 Multi-View Synthesis and Point Cloud Reconstruction

Multi-view inputs The proposed method can be naturally extended to use multi-view inputs as follows: for each image depth is predicted independently and combined into a single point cloud. The resulting coarse target image will be denser when more images are used, and is passed through the image completion network.

No. views	Coarse		Final	
	L1 (↓)	SSIM (↑)	L1 (↓)	SSIM (↑)
1	.203	.882	.090	.945
2	.188	.888	.089	.945
4	.152	.906	.085	.946
8	**.111**	**.907**	**.084**	**.947**

Table 5.3: Performance by extending single-view-trained networks for multi-view inputs.

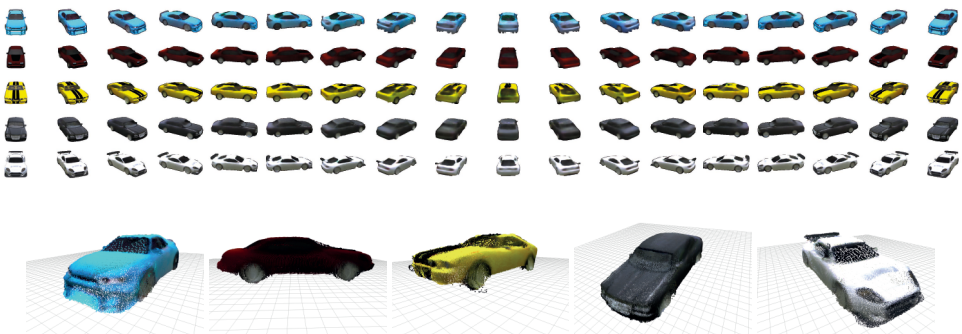
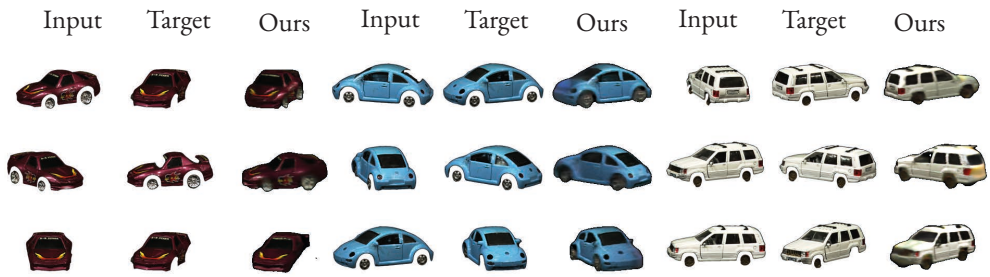


Figure 5.6: *Top* 360°-views generation from single input images (first column), *bottom* point cloud reconstruction using estimated depths of each generated views. Depth estimation trained on real images can perform well on synthesized ones.

In this experiment, the model trained for single-view prediction is used and evaluated using multiple (1 to 8) input images. The results in Table 5.3 show that the quality of the coarse view increases, as expected, when more input images are used and hence the point clouds are denser. Surprisingly, however, the image completion network only marginally improves, indicating that the coarse view contains enough information for the image completion network to synthesis a high quality target image.

Point cloud reconstruction In this final experiment, the aim is to reconstruct a full dense point cloud from a single image, using the models trained for novel view synthesis. In order to do so, 360°-views are generated from a single view of an object, see Figure 5.6 (top). Each of these views are fed to the depth estimation network and the obtain estimated depth is used to generate a partial point cloud. These point clouds are stitched together, using corresponding transformations, resulting in a high quality dense point cloud, as shown in Figure 5.6 (bottom).



5

Figure 5.7: Qualitative results on real-imagery ALOI [171] dataset. The inputs and targets are shown with provided object masks, while the synthesized images are with predicted masks. The completion network does not need to be finetuned, yet provide competent results.

5.4.4 Results on real-world imagery

We apply the trained car model to the car images of the real-imagery ALOI dataset [171], consisting of 100 objects, captured at 72 viewing angles. We use 4 cars for fine-tuning only the depth network, which requires no ground truths, while the image-completion network is left untouched. The quantivative resuls on the remaining 3 cars are shown in Figure 5.7.

5.5 Conclusion

In this chapter partial point clouds are estimated from a single image, by a self-supervised depth prediction network and used to obtain a coarse image in the target view. The final image is produced by an image completion network which uses the coarse image as input. Experimentally the proposed method outperforms any of the current SOTA methods on the ShapeNet Benchmark on novel view synthesis. Qualitative results show high quality and dense point clouds, obtained from a single image, by synthesizing and combining 360° views. Based on these results, we conclude that point clouds are a suitable, geometry aware representation for true novel view synthesis.

6 Multimodal Synthetic Dataset of Enclosed Gardens

Large-scale multimodal datasets for outdoor scenes are mostly designed for urban driving problems. The scenes are highly structured and semantically different from scenarios seen in nature-centered scenes such as gardens or parks. To promote machine learning methods for nature-oriented applications, such as agriculture and gardening, we propose the multimodal synthetic dataset for Enclosed garDEN scenes (EDEN). The dataset features more than 300K images captured from more than 100 garden models. Each image is annotated with various low/high-level vision modalities, including semantic segmentation, depth, surface normals, intrinsic colors, and optical flow. Experimental results on the state-of-the-art methods for semantic segmentation and monocular depth prediction, two important tasks in computer vision, show positive impact of pre-training deep networks on our dataset for unstructured natural scenes. The dataset and related materials will be available at https://lhoangan.github.io/eden.

6.1 Introduction

Synthetic data have been used to study a wide range of computer vision problems since the early days [113, 172, 173]. Compared to real-world imagery (RWI), computer-generated imagery (CGI) data provides allows for less expensive and more accurate annotation. Since the emergence of deep learning, synthetic datasets using CGI have become essential due

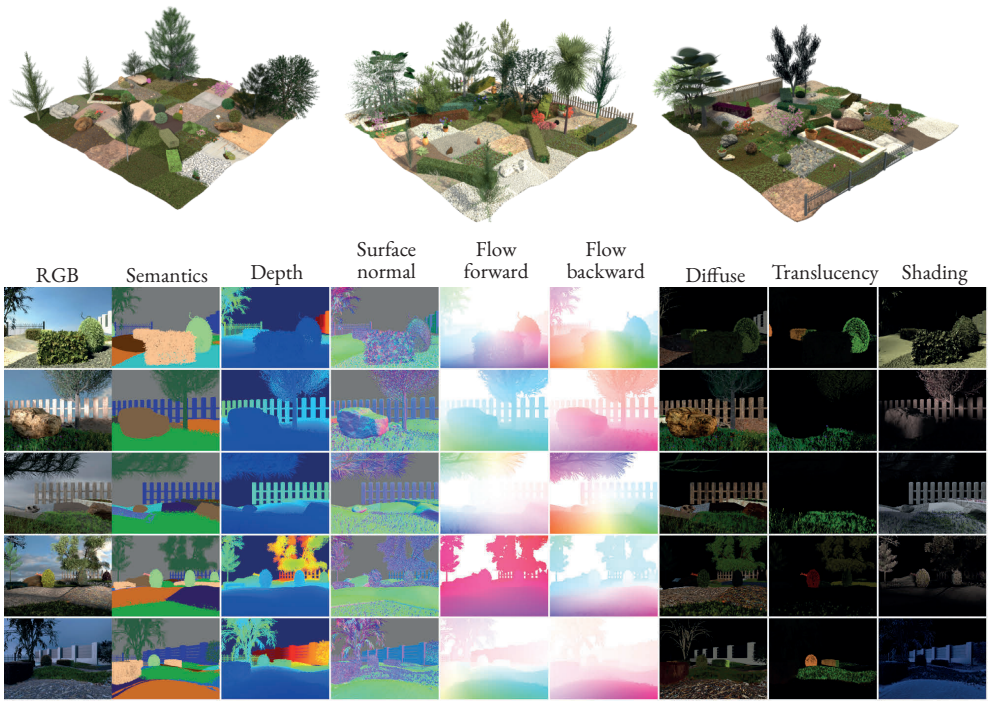


Figure 6.1: An overview of multiple data types provided in the dataset. The dataset includes data for both low- and high-level tasks such as (stereo) RGB, camera odometry, instant and semantic segmentation, depth, surface normal, forward and backward optical flow, intrinsic images (albedo, shading for diffuse materials, translucency)

to the data-hungry nature of deep learning methods and the difficulty of annotating real-world images. Most of the large-scale RWI datasets (with more than 20K annotated data points) are focusing on higher-level computer vision tasks such as (2D/3D) detection, recognition, and segmentation [53, 54, 174, 175, 176, 177]. In contrast, datasets for low-level image processing such as optical flow, visual odometry (KITTI [105, 110]) and intrinsic image decomposition (MIT [56], IIW [59], SAW [178]) are limited in the number of samples (around 5K annotated images).

CGI-based synthetic datasets [18, 19, 20, 40, 55, 123] provide more and diverse annotated modalities. High quality game data can be extracted from the game pixel shaders to train low-level vision tasks such as optical flow, visual odometry, and intrinsic image decomposition. The continuous progress of computer graphics and video-game industry results in improved photo-realism in render engines. The use of physics-based renderers facilitates the simulation of different lighting conditions (*e.g.* morning, sunset, nighttime).

Most of the existing datasets focus on car driving scenarios and are mostly composed of

simulations of urban/suburban scenes [19, 18, 123, 20]. City scenes are structured containing objects that are geometrically distinctive with clear boundaries. However, natural or agriculture scenes are often unstructured. The gaps between them are large and required distinctive attentions. For example, there are only trails and no drive ways nor lane marks for travelling; bushes and plants are deformable and often entangled; obstacles such as small boulders may cause more trouble than tall grass.

To facilitate the development of computer vision and (deep) machine learning for farming and gardening applications, which involve mainly unstructured scenes, in this chapter, we propose the synthetic dataset of Enclosed garDEN scenes (EDEN), the first large-scale multimodal dataset with >300K images, containing a wide range of botanical objects (*e.g.* trees, shrubs, flowers), natural elements (*e.g.* terrains, rocks), and garden objects (hedges, topiaries). The dataset is created within the TrimBot2020 project* for gardening robots, and have pre-released versions used in the 3DRMS challenge [179] and in several work [125, 179, 25].

In contrast to man-made (structured) objects in urban scenarios (such as buildings, cars, poles, *etc.*), the modelling of natural (unstructured) objects is more challenging. Natural objects appear with their own patterns and shapes. Rendering techniques using rotating billboards of real photos may provide realistic appearances, but lack close-up geometrical features. Although synthetic datasets and video-games may offer natural objects and scenes, they often come with generic labels (e.g. tree, grass, and simple vegetation), since their focus is on the gaming dynamics.

Therefore, objects in our dataset are developed using high-fidelity parametric models and CADs created by artists to obtain natural looking scenes. The object categories are selected for the purpose of gardening and agricultural scenarios to include a large variety of plant species and terrain types. The dataset contains different lighting conditions to simulate the intricate aspects of outdoor environments. The different data modalities are useful for both low- and high-level computer vision tasks.

In addition to the new dataset itself, we provide analyses and benchmarks of the dataset on state-of-the-art methods of two important tasks in computer vision, namely semantic segmentation and depth prediction.

* http://trimbot2020.webhosting.rug.nl/

6.2 Related Work

6.2.1 Real-imagery datasets

To accommodate the emergence of deep learning and its data-demanding nature, many efforts have been spent on creating large-scale generic datasets, starting with the well-known ImageNet [54], COCO [53], and Places [180]. These are real-world imagery (RWI) datasets with more than 300K annotated images at object and scene-level. Also in the domain of semantic segmentation, there are a number of datasets available such as ADE20K [174] (20,210 images, 150 categories) and Pascal-Context [181] (10,103 images, 540 categories).

Annotation is expensive. Lower-level task annotation is even more expensive. In contrast to the availability of large datasets for higher-level computer vision tasks, there are only a few RWI datasets for low-level tasks such as optical flow, visual odometry, and intrinsic image decomposition due unintuitive data annotation. Middlebury [182] and KITTI [105, 110] are the only datasets providing optical flow for real-world images, yet too small to train a deep network effectively. For intrinsic image decomposition, the MIT [56] dataset provides albedo and shading ground truths for only 20 objects in controlled lighting conditions, while IIW [59] and SAW [178] provide for up to 7K in-the-wild and indoor images. Indoor-scene datasets [176, 177, 183, 184] provide a larger number of images (up to 2.5M) and with more modalities (such as depth) than generic datasets. However, their goal is to provide data for 3D (higher-level) indoor computer vision tasks.

Outdoor scenes are subject to changing imaging conditions, such as lighting, viewpoint, occlusion, and object appearance, resulting in annotation difficulties. A number of methods are proposed focusing on scene understanding for autonomous driving [101, 105, 110, 175, 185, 186]. However, these datasets are limited in number of images and/or the number modalities. Mapillary [175, 187] is the most diverse dataset with varying illumination conditions, weather, and seasonal changes. Their focus is on semantic segmentation and place recognition. Large-scale multimodal datasets are restricted to synthetic data.

6.2.2 Synthetic datasets

Computer vision research uses synthetic datasets since the early days to study low-level tasks, *e.g.* optical flow [113, 172, 173]. Synthetic data provide cheaper and more accurate annotations. It can facilitate noise-free and controlled environments for otherwise costly problems [188, 189] or for intrinsic understanding [190] and proof of concept [145, 164].

Obviously, the quality of synthetic data and annotation depends on the realism of mod-

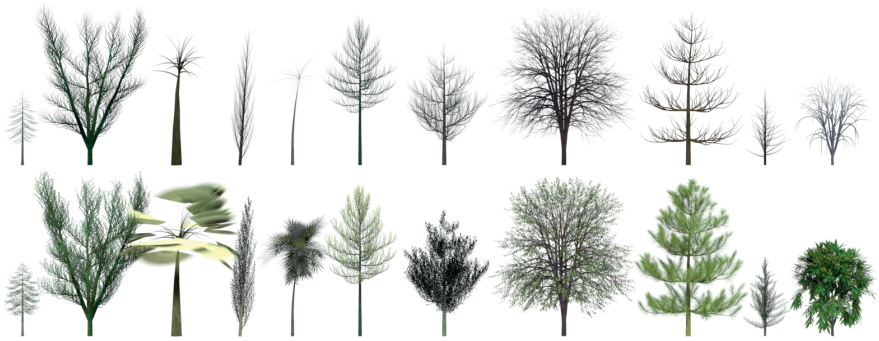


Figure 6.2: Sample tree models (top: tree stems, bottom: with leaves) for various tree species

elling and rendering algorithms. The development of computer graphic techniques has led to physics-based render engines and the improvement of photo-realistic computer-generated imagery (CGI). SYNTHIA [19] and Virtual KITTI [18] simulate various daylight conditions (morning, sunset), weather (rain, snow), and seasonal variations (spring, summer, fall, winter) for autonomous (urban) driving datasets. Datasets obtained from video-games [123, 20, 191] and movies [40, 55] show adequate photo-realism. These datasets provide not only dense annotations for low and high-level tasks, but also multi-view images, different illumination/weather/seasonal settings. They have proven useful for training robust deep models under different environmental conditions [123, 20].

However, datasets for outdoor scenes focus mostly on either generic or urban driving scenarios. They mainly consist of scenes containing man-made (rigid) objects, such as lane-marked streets, buildings, vehicles, *etc*. Only a few datasets contain (non-rigid) nature environments (e.g. forests or gardens [179, 192]).

CGI-based datasets rely on the details of object models, and computer-aided designed (CAD) model repositories, such as ShapeNet [39], play an important role in urban driving datasets [19, 18]. However, the models usually include rigid objects with low fidelity. Others focus on capturing the uniqueness of living creatures, such as humans [193, 194], and trees [195, 196, 197] to generate highly detailed models with realistic variations. Synthetic garden datasets have been used in [125, 179, 25], albeit these datasets are relatively small and have just one or two modalities and are not all publicly available. In this chapter, we use different parametric models, *e.g*. [195], to generate different botanical objects in an garden. We create multiple gardens with different illumination conditions, and extract multi-modal data (including RGB, semantic segmentation, depth, surface normals *etc*.) from each frame, yielding over 300K garden frames, which we will make publicly available.

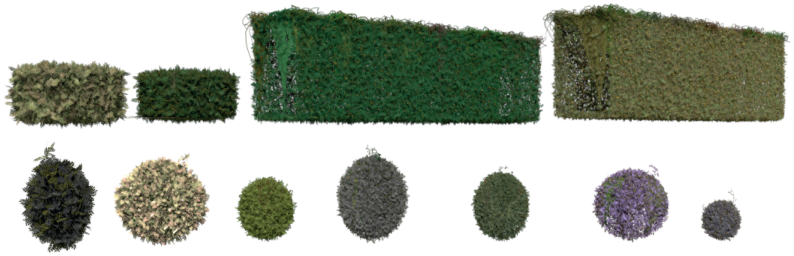


Figure 6.3: Sample models for hedges (top) and topiaries (bottom). The bushes can be generated with various sizes, leaf colors, and internal stem structures.

6.3 Dataset Generation

We create synthetic gardens using the free and open-source software of Blender[†], and render using the physics-based Cycles render engine. Each garden consists of a ground with different terrains and random objects (generated with random parameters or randomly chosen from a pre-designed models). The modelling details of each component object and the rendering settings are presented in the following sections.

6.3.1 Modelling

To expand the diversity of objects and scenes, we propose to combine parametric and pre-built models in the generation process.

Trees We use the tree parametric model described in [195], implemented by the Blender Sapling Add-on [‡]. A tree is constructed recursively from common predefined tree shapes (conical, (hemi-)spherical, (tapered) cylindrical, *etc*.) with the first level being the trunk. The parameters define the branch features such as length, number of splits, curvatures, pointing angles, *etc*., each with a variation range for random sampling. Leaves are also defined in a similar manner as stems, besides a fractional value determining their orientation to simulate phototropism. The model can generate different tree species such as quaking aspens, maples, weeping pillows, and palm trees. We use the parameter presets provided in the sampling add-on and Arbaro[‡] (Figure 6.2). Totally there are 19 common tree species.

Bushes Hedges and topiaries are generated by growing an ivy adhering to a rectangular or spherical skeleton object using the Ivy Generator[‡], implemented by the Blender IvyGen add-on[‡] (Figure 6.3). An ivy is recursively generated from a single root point by forming curved objects under different forces including a random influence to allow over-

[†] `blender.com`, GPL GNU General Public License version 2.0
[‡] See Section 6.5 for the reference link

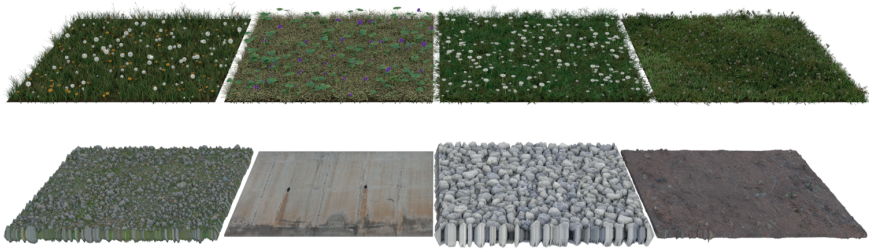


Figure 6.4: Example tiles of different terrain types: grass with weed (*top*), gravel, pavement, pebble stones, dirt (*bottom*). The grass and weed species are chosen and combined randomly.

growing, an adhesion force to keep it attached to the trellis, a gravity pulling down, and an up-vector simulating phototropism. The add-on is known for creating realistic-looking ivy objects (Figure 6.3). We use more than 20 leaf types with different color augmentation for both topiaries and hedges.

Landscapes and terrain The landscape is created from a subdivided plane using a displacement modifier with the Blender cloud gradient noise, a representation of Perlin noise [198]. The modifier displaces each sub-vertex on the plane according to the texture intensity, creating the undulating ground effect. The base ground is fixed at 10x10 square meters, on which are paved the terrain patches of 1x1 square meter. Each patch is assigned to one of the terrain types, including grass, pebble stones, gravels, dirt and pavement.

The grass is constructed using Blender particle modifier which replicates a small number of elemental objects, known as particles, over a surface. We use the grass particles provided by the Grass Essentials[‡], and the Grass package[‡], containing expert-designed realistic-looking grass particles. There are more than 30 species of grass (*e.g.* St. Augustine grass, bahiagrass, and centipedegrass) and weed (*e.g.* dandelions, speedwell, and prickly lettuce), each has up to 49 model variations. The appearance of the grass patch is controlled via numerical parameters, such as freshness, brownness, wetness, trimmed levels, lawn stripe shape, *etc*. Illustrations for different grass and weed species are shown in Figure 6.4 (top).

The other terrains are designed using textures from the Poliigon collection[‡] of high quality photo-scanned textures. Illustrations are shown in Figure 6.4 (bottom). Each texture contains a reflectance, surface normal, glossy, and reflection map with expert-designed shaders for photo-realism. The combined landscapes and terrains can be seen in Figure 6.1.

Environment Lighting in our dataset is created by 2 sources, a sun lamp and a sky texture. A sun lamp is a direct parallel light source, simulating an infinitely far light source. The source parameters include direction, intensity, size (shadow sharpness), and color. A

Figure 6.5: Illustration for scene appearance changed according to different illumination conditions.

sky texture provides the environmental background of rendered images and a source of ambient lights. We use the Pro-Lighting: Skies package[‡] composing of 95 realistic equirectangular HDR sky images of various illuminations. The images are manually chosen and divided into 5 scenarios, namely clear (sky), cloudy, overcast, sunset, and twilight. We also use 76 HDR scenery images[‡] to create more various and complex backgrounds, some with night lighting, coined scenery. An example of lighting effects is shown in Figure 6.5.

Pre-built models To enhance the model variations in the dataset, we also include models prebuilt from different artists, including rocks[‡], flowers[‡], garden assets such as fences, flower pots[‡], *etc*.

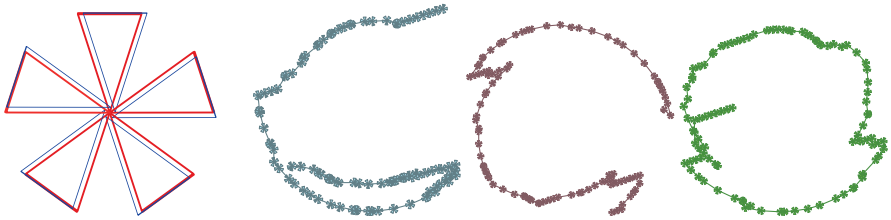
6.3.2 Rendering

Camera setup We follow the real-world camera setup in the 3DRMS challenge to create a ring of 5 pairs of virtual stereo cameras, angular separation of 72° (Figure 6.6a). Each stereo pair has a baseline of 0.03 meters, and each camera has a virtual focal length of 32mm on a 32mm wide simulated sensor. The rendered resolution is set to VGA-standard of 480x640 pixels. The camera intrinsic matrix is as follows:

$$\mathbf{K} = \begin{bmatrix} 640 & 0 & 320 \\ 0 & 640 & 240 \\ 0 & 0 & 1 \end{bmatrix}. \tag{6.1}$$

We generate a random trajectory for the camera ring for each illumination variation of each garden model. The speed is set to about 0.5m/s, frame rate of 10*fps*, simulating a trimming robot in a garden. To improve the variability, the camera ring is set to randomly turn after a random number of steps and avoid running through the objects. The turning angles are also randomized to include both gradual and abrupt angles. The trajectory lengths are set to be at least 100 steps. The examples are shown in Figure 6.6b.

Render engine Blender Cycles is a probabilistic ray-tracing render engine that de-



(a) *(b)*

Figure 6.6: The camera system includes 5 pairs of stereo cameras at 72° angular separation (a) and random trajectories used in rendering process (b)

rives the color at each pixel by tracing the paths of light from the camera back to the light sources. The appearances of the objects are determined by the objects' material properties defined by the bidirectional scattering distribution function (BSDF) shaders, such as diffuse BSDF, glossy BSDF, translucent BSDF, *etc*.

Scene aspects such as geometry, motion and the object material properties are rendered into individual images before being combined into a final image. The formation of a final image $I(\mathbf{x})$ at position $\mathbf{x}$ is as follows[§]:

$$f_g(\mathbf{x}) = g_{\text{color}}(\mathbf{x})(g_{\text{direct}}(\mathbf{x}) + g_{\text{indirect}}(\mathbf{x})), \tag{6.2}$$

$$I(\mathbf{x}) = f_D(\mathbf{x}) + f_G(\mathbf{x}) + f_T(\mathbf{x}) + B(\mathbf{x}) + E(\mathbf{x}), \tag{6.3}$$

where D, G, T, B, E are respectively the diffuse, glossy, transmission, background, and emission passes. D_{color} is the object colors returned by the diffuse BSDF, also known as albedo; D_{direct} is the lighting coming directly from light sources, the background, or ambient occlusion returned by the diffuse BSDF, while D_{indirect} after more than one reflection or transmission off a surface. Similar are G and T with glossy and transmission BSDFs. Emission and background are pixels from directly visible objects and environmental textures. The intermediate image contains at each pixel the corresponding data or zeros otherwise. All the computations are carried out in the linear RGB space. Blender converts the composite image to sRGB space using the following gamma-correction formula and clipped to $[0, 1]$ before saving to disk:

[§]*c.f.* Blender 2.83 Manual, last access July 2020

Split	Training (127)	Test (20)	
		full	20K
clear	74,913	10,035	3,333
cloudy	73,785	10,030	3,378
overcast	73,260	10,015	3,349
sunset	73,715	10,040	3,250
twilight	73,992	10,045	3,369
total	369,663	50,165	20,000

Table 6.1: Number of images per scene and split; the number of models are in parentheses

$$\gamma(u) = \begin{cases} 12.92u & u \leq 0.0031308 \\ 1.055u^{1/2.4} - 0.055 & \text{otherwise} \end{cases} \tag{6.4}$$

In our dataset, besides the *RGB* stereo pairs and cameras' poses, we provide the images from intermediate stages, namely albedo, shading, translucency, *etc*. for the left cameras. As the rendering are physics-based, the intermediate images represent different intrinsic modalities. Examples are shown in Figure 6.1.

6.4 Experiments

In this section, the goal is to quantitatively analyze the newly created dataset to assess its realism and usability. The evaluation is performed via two proxy tasks: semantic segmentation and monocular depth estimation. We split the dataset into training (127 models, 369,663 monocular images) and test set (20 models, 60,195 images). To speed up the evaluation process, we uniformly sample 20K images from the full test set. The statistics are shown in Table 6.1. The sample list will also be released together with the dataset.

6.4.1 Semantic segmentation

For semantic segmentation, we use the state-of-the-art DeepLabv3+ architecture with Xception-65 backbone [199]. Three aspects of the dataset are analyzed, namely (1) training size, (2) lighting conditions, and (3) compatibility with real-world datasets. The label set is from the 3DRMS challenge [179, 200]: void, grass, ground, pavement, hedge, topiary, flower, obstacle, tree, background. Background contains the sky and objects outside of the gar-

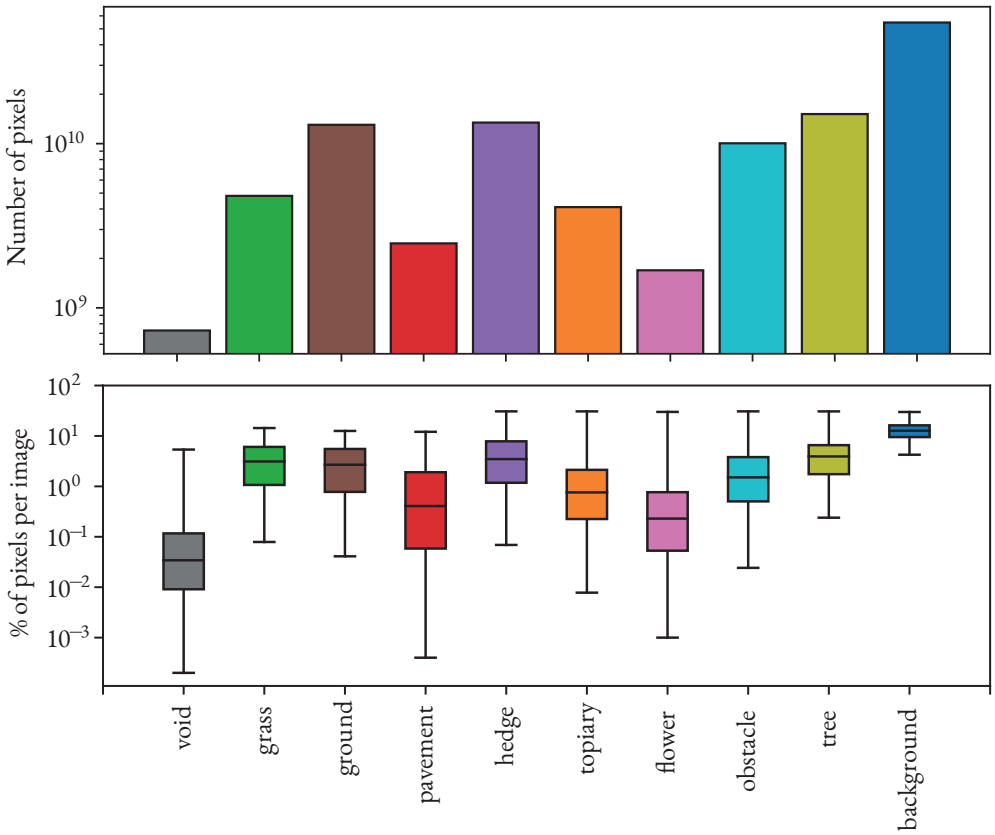


Figure 6.7: Number of pixels per class in the dataset (*top*) and distributions in the images (*bottom*). The boxplot shows the 1st, 2nd (median) and 3rd quartile of the number of pixels in each frame, with the whisker value of 1.5. *background* includes sky and object outside of the garden, while *void* indicates unknown pixels, which should be ignored.

den, while void indicates unknown objects to be ignored. The label statistics are shown in Figure 6.7. We also follow the network's training setup and report mean intersection-over-union (mIOU). The results are shown in percentage and higher is better.

Training and testing size We first show the benefit of an increasing training set and the performance on the full and reduced test set. The results are shown in Table 6.2. The performance increases when the training size increases, showing the advantage of having large amount of training samples. The evaluation on the reduced test set is similar to the full set. Thus, unless mentioned otherwise, the test20K split will be used for evaluation in later experiments.

Lighting conditions Our dataset contains the same garden models in various lighting conditions, allowing in-depth analysis of illumination dependency of different meth-

Sampling	Test	
	full	20K
25%	75.71	75.89
50%	79.42	79.52
100%	81.96	82.09

Table 6.2: Performance with respect to different training size and at 2 test splits. The network performance increase when being trained on higher number of images. The performance on the reduced test set is on par with the full set.

Training	Test					
	clear	cloudy	overcast	sunset	twilight	20K
clear	**76.10**	*76.91*	76.43	72.23	75.91	72.03
cloudy	75.09	***77.59***	77.16	72.37	76.40	**72.30**
overcast	65.75	75.52	***78.41***	70.76	74.63	70.22
sunset	73.21	75.76	77.17	**74.44**	*77.28*	71.84
twilight	66.19	72.86	76.21	70.55	***78.16***	68.83

Table 6.3: Cross-lighting analysis. Each row corresponds to a model trained on the specific lighting condition (highest values are in italics), while each column corresponds to the results evaluated on the specific subset (highest values are in boldface). Lighting-specific training gives better results on the specific lighting, while the results in the cross-lighting vary depending on the conditions of the training and test images.

ods for different tasks. In this section we perform cross-lighting analysis on semantic segmentation. We conduct lighting-specific training of the networks, and evaluate the results on each lighting subset of the full test set, as well as the reduced test set. The results are shown in Table 6.3. All experiments are trained with the same epoch numbers.

For almost all of the categories, training on the specific lighting produces the best results on that same categories. This is not surprising, as networks always perform the best on the most similar domains. In general, training with cloudy images gives the highest performance, while twilight are the lowest. This could be due to relatively bright images and less intricate cast shadows in cloudy scenes, in contrast to the mostly dark and color cast twilight images.

Compared to training with all the full training set in Table 6.2, the results from training with lighting-specific images are generally lower and near to the 25% subset. This agrees to the training size conclusion as the lighting-specific training sets account only for around 20% of the data. Testing on the same lighting gives a boost in performance, similarly to training with double data size.

Pre-training	Test	
	3DRMS	Freiburg
Generic	24.35	41.33
Cityscapes	31.11	50.08
EDEN	**34.55**	**52.45**

Table 6.4: Adaptability of features pre-trained on different datasets to unstructured natural real-world scenes. The network pre-trained on EDEN outperforms all other alternative approaches on both 3DRMS and Freiburg test sets.

Real-world datasets Semantic segmentation requires to recognize objects from the appearance models learned during training. Therefore, it indicates the closeness of training data to the testing domain. By analyzing the features learned from EDEN on real images of unstructured natural scenes, the results indicate the realism level of our dataset. To that end, the real-imagery datasets 3DRMS [200, 179] (garden scenes, 221 annotated real images for training, 268 for validation, 10 classes), Freiburg forest [192] (forested scenes, 228 annotated real images for training, 15 for validation, 6 classes) are used for evaluation.

The baselines include (1) the network pre-trained on combination of generic datasets, COCO [53], ImageNet [54], and augmented PASCAL-VOC 2012 [102], and (2) the network pre-trained on ImageNet and urban driving scene dataset Cityscapes [101]. The pre-trained weights are all provided by the networks' authors [199]. We keep the encoder part frozen and finetune only the decoder using the train split of each target set for 50K iterations. The results are shown in Table 6.4.

The networks using the features learned from EDEN out-perform all alternative approaches. Both 3DRMS and Freiburg features highly unstructured scenes with mostly deformable and similar objects found in the nature, drastically different from the generic images and structured urban scenes. The results show the realism of our datasets to natural scenes and its benefit on training deep networks. The results on Freiburg test are higher than on 3DRMS due to the relatively simpler and general class labels (*e.g.* trails, grass, vegetation, and sky) compared to the garden-specific label sets of 3DRMS (*e.g.* hedges, topiaries, roses, tree, *etc.*).

6.4.2 Monocular depth prediction

Monocular depth prediction is an ill-posed problem. Often the ambiguity is mitigated by learning from a large-scale depth-annotated dataset [201, 202] or imposing photometric

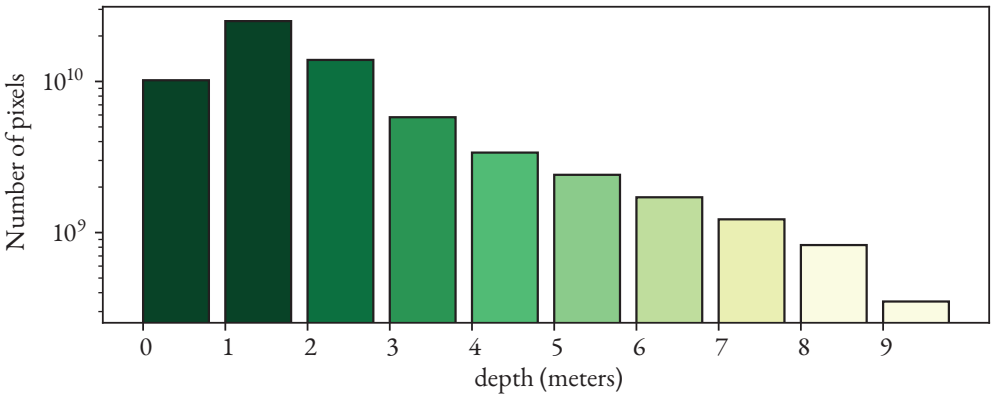


Figure 6.8: Number of pixels per depth range in the dataset. Each range is a left-inclusive half-open interval.

constraints on image sequences using relative camera poses [157, 158] As camera pose prediction can be formulated using depth constraint, the depth-pose prediction problems can be combined in a self-supervised learning pipeline.

Synthetic datasets are favored for being noise-free, which can act as controlled environments for algorithm analysis. In this section, we use EDEN to test different monocular depth prediction networks. Specifically, we examine the effectiveness of using supervised signals in learning depth prediction for unstructured natural scenes. The statistics of the depth in the dataset are shown in Figure 6.8.

We show the results of training state-of-the-art architectures using different ground truth information, namely depth and camera pose. To that end, the 2 methods, VNL [202] and MD2 [158] are used. VNL is trained with supervised depth, while MD2 can be trained with ground truth camera pose or in self-supervised manner. Both are trained using the schedules and settings provided by the respective authors. The results are shown in Table 6.5 with 3 error metrics (rel, log10, rms, lower is better) after the original work and included the reported results on the KITTI dataset for comparison.

Generally, supervised method always produce better results than their self-supervised counterpart as shown by the smaller errors. The difference are less for the KITTI dataset compared to EDEN. As KITTI contains mostly rigid objects and surfaces, it is simpler to obtain predicted camera poses with high accuracy. On the other hand, camera pose prediction for self-supervised learning on EDEN are unreliable because of deformable objects and their similarities. The errors are, therefore, also higher for supervised methods on EDEN than on KITTI, showing the more challenging dataset. KITTI has higher RMS numbers due to the larger depth ranges, approximately 80m vs. 15m of EDEN.

Method	Supervised	Dataset	rel	log10	rms
MD2	None	KITTI	0.115	0.193	4.863
VNL	Depth	KITTI	0.072	0.117	3.258
MD2	None	EDEN	0.438	0.556	1.403
MD2	Pose	EDEN	0.182	0.220	0.961
VNL	Depth	EDEN	0.181	0.083	1.061

Table 6.5: Performance of different SOTA methods for monocular depth prediction when trained on KITTI and EDEN. The gap is larger between unsupervised and supervised methods on EDEN, showing the necessity of having supervised signals for learning unstructured scenes. The errors on EDEN are generally higher than on KITTI, showing the more challenging scenes of the (unstructured) dataset.

6.5 Conclusions

The chapter presents EDEN, a large-scale multimodal dataset for unstructured garden scenes, and provides baseline results and analysis on two important computer vision tasks, namely the problems of semantic segmentation and monocular depth prediction. The experiments show favorable results of using the dataset over generic and urban-scene datasets for nature-oriented tasks. The dataset comes with several computer vision modalities and is expected to stimulate applying machine and deep learning to agricultural domains.

Add-ons packages

Following list the packages and their corresponding links, which are used in the construction of the datasets. All links are last accessed in June 2020.

Blender Sapling add-on, Royalty-Free License
https://docs.blender.org/manual/en/latest/addons/add_curve/sapling.html
Arbaro, GNU General Public License version 2.0
https://sourceforge.net/projects/arbaro/
Ivy generator, Thomas Luft
http://graphics.uni-konstanz.de/~luft/ivy_generator
Blender IvyGen Add-on
https://docs.blender.org/manual/en/dev/addons/add_curve/ivy_gen.html
Grass essential package, Royalty-Free License
https://blendermarket.com/products/the-grass-essentials

Grass package, Royalty-Free License
https://www.3d-wolf.com/products/grass.html
Pro-Lighting Skies package, Royalty-Free License
https://blendermarket.com/products/pro-lighting-skies
Poliigon, Royalty-Free license
https://www.poliigon.com/
HDRI Haven, CC0 license
hdrihaven.com
Essential rock package, Royalty-Free license
https://blendermarket.com/products/the-rock-essentials
Flower package 1, Royalty-Free license
https://blendermarket.com/products/flowers-pack-1
Flower package 2, Royalty-Free license,
https://blendermarket.com/products/flowers-pack-2
Garden asset package, Royalty-Free license
https://blendermarket.com/products/garden-asset-pack

The eye is the lamp of the body.
So, if your eye is healthy, your whole body will be full of light;
but if your eye is unhealthy, your whole body will be full of darkness.

Matthew 6:22

7

Summary and Conclusions

Summary

The thesis investigates various computer vision modalities, which, taken from the broad definitions, include both sensory data as well as subsequent interpretation such as *RGB*, depth, intrinsic images, semantic maps, surface normals, optical flow, and point clouds. Specifically, the thesis focuses on the research question *how can various computer vision modalities be exploited and combined?* The question is tackled from multiple perspectives, starting with decomposing a primary modality, followed by the study of modality complement and combination. The subsequent chapters explore multimodality from a generative perspective, how a modality benefits generation of the others, and concludes with the construction of a multimodal synthetic dataset.

Chapter 2 studies the decomposition of photometric information into objects' texture colors (albedo) and geometry and illumination effects (shadings). We propose two deep learning architectures combining domain knowledge from the well-established Retinex method and constrained by physics-based reflection models. The proposed models are evaluated on synthetic, real world and in-the-wild images. Quantitative results show that the new model outperforms existing methods, while visual inspection shows that the image formation loss function augments color reproduction and the use of gradient information produces sharper edges.

Chapter 3 investigates the complementary information among the important modalities of computer vision, namely optical flow, surface normals and semantic segmentation.

We analyze the combination of the three modalities and their impact on one another. Surface normals signify the objects' shapes, optical flow represents motion information which depends on object types and shapes, and semantic segmentation provides the categories and extension of objects in a scene. We approach the problem at a modular level where each modality input is refined from preliminary estimation using only the other modalities and not RGB images. Experimental results show that semantic information helps object boundaries, optical flow improves scene structures, and surface normals facilitate object recognition with geometric cues.

Chapter 4 focuses on capturing and generating non-rigid optical flow. The motion patterns are confined to complex movements of objects in real world videos. To that end, object segmentation is applied to extract the objects of interest whose motion characteristics are recorded by correspondences among the frames along the video sequence. The as-rigid-as-possible deforming principle is applied to the object segments to generate the flow field. By image warping, the generated flow fields can be used as ground truths for the first image and its warped version. Extensive experimental results show that the generated data are helpful for training deep networks for optical-flow prediction.

Chapter 5 exploits objects' geometry in predicting their appearances from unobserved points of views. The relationship between geometric and photometric information is implemented by the forward and backward warping principles to jointly train a monocular depth prediction network and image completion network. The proposed method shows quantitative superiority over the current state-of-the-art as well as impressive qualitative results in 360° views image generation, and point cloud reconstruction from single images.

Chapter 6 presents a multimodal dataset and analyzes its performance on 2 important computer vision tasks, semantic segmentation and monocular depth prediction. Semantic segmentation results show that features learned from the dataset are more efficient in pretraining deep networks for unstructured natural scenes, compared to other well-known large scale datasets, thus proving the dataset reality level. Depth is predicted in both supervised and self-supervised monocular settings Camera odometry ground truths are also employed within the context of depth prediction, which shows comparable results to using ground truth depths. The experiments show favorable results of using the dataset over generic and urban-scene datasets for nature-oriented tasks. The dataset comes with several computer vision modalities and is expected to stimulate applying machine and deep learning to agricultural domains.

General Conclusions

The thesis has offered a number of studies for computer vision modalities and their use cases, inspired by the human's abilities of utilizing various perceptual modalities. Conveying different perspectives of the world, the visual modalities are extracted within humans' cognitive process for better environmental experience.

However, the thesis has only scratched the surface of understanding and exploiting visual multi-modalities. Although visual perception once was to assist living entities' survivability, it has since evolved to be the effective channel, a lamp (*c.f.* the epigraph), for humans to deepen their understanding of the world. As humans, driven by the pure desire to know [203], constantly seek intellectual knowledge and rational meaning in the surroundings, our perception and cognition have grown complex and intertwined [204], leading to more sophisticated comprehension capabilities.

Therefore, a logical next step would be involving multiple modalities in improving learning capacity and expansion to the time domain. Optical flow relating information along the temporal axis is studied in Chapter 3 and 4 yet only to the extent of a limited number of local points in time. The desire could be a unified and continuous representation of multiple modalities, which improve each component modality and/or give raise to subsequently new modalities. The use of multimodality could be the solution for the well-known catastrophic forgetting problem of artificial intelligence [205, 206]. As multisensory integration is an important biological factor in lifelong learning [207], multimodal data should play a role in tackling the challenges of continuous learning in artificial intelligence.

Humans' experiences are enriched via various encounters and interactions over the course of life. In the similar manner, multimodality should also be studied in different applications. Recognition of faces, emotions, actions, activities, scenes, *etc*. are all different scenarios where multimodality can be of use and explored. Understanding the combination of different modalities within these contexts would eventually lead to more insights for the unified representation.

Furthermore, as the topics presented in this thesis mostly deal with generic computer vision problems, systematic and careful studies would be desired to adapt the concepts to specialized domains, such as earth observation and remote sensing, or medical computer vision. Each field uses its own typical data types and disciplines, such as radargrams, sonograms, thermograms, x-ray images, MRI, *etc*. obtained from non-optical sensors like radar, LiDAR, SoNaR, infrared vision devices, *etc*. These technologies, despite proving to be useful in their applicative contexts, provide information beyond the visible spectrum, hence

not straightforwardly compatible with methods designed for conventional computer vision modalities. Multimodal research with divergent sensors is helpful for understanding machine learning capability and extend its applicability.

Computer vision, once started as a loosely separate problem from artificial intelligence and going after the mechanism of the visual system [208], has achieved advanced and impressive results which might surpass human-level performance [209] since the breakthrough of deep neural networks. As multimodality is being learned and artificial intelligence is approaching human cognitive level, humans can be liberated from repetitive and dangerous jobs *to focus on what truly makes us human: loving and being loved... For despite all of AI's astounding capabilities, the one thing that only humans can provide turns out to also be exactly what is most needed in our lives: love.* [210]

8

Samenvatting

Begrip van Beelden uit Meerdere Modaliteiten van Scènes in de Openlucht

Deze thesis onderzoekt verschillende modaliteiten binnen de beeldbewerking, die breed gezien, zowel sensor data als de resulterende interpretatie van deze sensor data omvat. De data die meegenomen is in deze thesis zijn RGB-beelden, diepte-beelden, intrinsieke beelden, semantische beelden, oppervlak normaalvectoren, optisch stroomveld, en puntenwolken. Deze thesis gaat in op de volgende onderzoeksvraag: *Hoe kunnen verschillende beeldmodaliteiten kunnen worden geëxploiteerd en gecombineerd?* De onderzoeksvraag wordt belicht vanuit meerdere perspectieven, beginnend bij het ontleden van de hoofd-modaliteit, gevolgd door het combineren met andere modaliteiten. De volgende hoofdstukken verkennen multi-modaliteit in zijn algemeenheid: Hoe heeft modaliteit profijt bij het generen van andere modaliteiten? De thesis wordt afgesloten met het ontwerpen en construeren van een multimodale synthetische dataset.

In Hoofdstuk 2 wordt de decompositie van fotometrische informatie in de kleur van een object en geometrische- en belichtingseffecten bestudeerd. We stellen twee Deep Learning architecturen voor die domein-relevante kennis van de bekende Retinex methode met beperkingen uit natuurkunde-gebaseerde reflectiemodellen combineren. De voorgestelde modellen worden geëvalueerd op synthetische, natuurlijke, en "in-het-wild" beelden. Kwantitatieve resultaten tonen aan dat de nieuwe voorgestelde modellen beter presteren dan bestaande methoden. Kwalitatieve inspectie laat zien dat de energie functie voor het construeren van de beelden, kleur voorspellingen verbetert. Daarnaast is er sprake van scherpere randen door het gebruik van afgeleiden.

In Hoofdstuk 3 wordt de complementerende werking van informatie in meerdere modaliteiten binnen de beeldbewerking onderzocht: optisch stroomveld, oppervlak normaalvectoren en seman-

tische segmentatie. We analyseren de combinatie van deze drie modaliteiten en hun impact op elkaar. Oppervlak normalen zijn gedefinieerd als de vormen van de objecten, optische stroomvelden zijn gedefinieerd als bewegings informatie en hangen af van de vorm en het type object, en semantische segmentatie levert verschillende categorieën of extensies van objecten aan in een scène. We benaderen het probleem op een modulair niveau, waar elke begin-modaliteit apart wordt verbeterd uit een eerste schattig door alleen gebruik te maken van de andere modaliteiten en niet de RGB-beelden. Experimentele resultaten laten zien dat semantische informatie hulp biedt in het schatten van object grenzen, optische stroming verbetert het schatten van scène structuren, en dat oppervlak normaalvectoren faciliteren objectherkenning with geometrische informatie.

Hoofdstuk 4 legt de focus op het vangen en genereren van niet-rigide optische stroming. The bewegingspatronen zijn beperkt tot complexe bewegingen van objecten in natuurlijke videos. Om daarmee te helpen, is object segmentatie toegepast om objecten te scheiden wiens karakteristieke bewegingen zijn vastgelegd door correspondentie tussen meerdere beelden uit een video. Het zo-rigide-als-mogelijk vervormingsprincipe is toegepast op segmenten van objecten om het stromingsveld te genereren. Door het vervormen van een beeld, kunnen de gegenereerde optische stroomvelden gebruikt worden als grondwaarheden voor het eerste plaatje en de vervormde versie. Uitgebreide experimenten laten zien dat de gegenereerde data helpen in het trainen van diepe neurale netwerken voor het voorspellen van optische stroming.

In Hoofdstuk 5 wordt de geometrie van een object gebruikt in het voorspellen van het aanzicht van een object uit niet geobserveerde perspectieven. De relatie tussen geometrische en fotometrische informatie wordt geïmplementeerd met behulp van voorwaartse en terugwaartse vervormingsprincipes om tegelijkertijd twee netwerken te trainen: Een netwerk dat diepte voorspelt uit monoculaire beelden en een netwerk dat beelden compleet maakt. Kwantitatief gezien haalt de voorgestelde methode superieure resultaten ten opzichte van de state-of-the-art. Ook haalt de methode indrukwekkende resultaten in het genereren van 360 graden beelden en het voorspellen van puntenwolken uit enkele beelden.

In het laatste Hoofdstuk presenteren we een multimodale dataset en analyseren behaalde resultaten op de dataset voor twee belangrijke taken in beeldbewerking: semantische segmentatie en diepte schatting. Resultaten in semantische segmentatie doen vermoeden dat geleerde kenmerken uit de dataset efficiënter zijn in het voor-trainen van diepe neurale netwerken voor ongestructureerde natuurlijke scènes in vergelijking met andere grote bekende datasets. Dit bewijst dat de nieuwe dataset een sterke mate van realiteit encodeert. Diepte wordt geschat zowel in een gesuperviseerde en niet zelf-gesuperviseerde manier in monoculaire context. Camera odometrie grondwaarheden worden ook ingezet in de context van diepte schattingen. Hier worden resultaten behaald gelijk aan resultaten met diepte grondwaarheden. Experimenten laten zien dat er gunstige resultaten worden behaald ten opzichte van generieke in stedelijke datasets specifiek voor natuur-georiënteerde taken. De dataset bevat enkele modaliteiten die belangrijk zijn voor beeldbewerking en de verwachting is dat het onderzoek in deep learning zal stimuleren voor het agriculturele domein.

Bibliography

[1] S. E. Palmer, *Vision science : photons to phenomenology*. Cambridge, Mass.: MIT Press, 1999.

[2] D. L. Williams, "Light and the evolution of vision," *Eye*, vol. 30, no. 2, 2015.

[3] E. Liscum, S. K. Askinosie, D. L. Leuchtman, J. Morrow, K. T. Willenburg, and D. R. Coats, "Phototropism: Growing towards an Understanding of Plant Movement," *The Plant Cell*, vol. 26, no. 1, 2014.

[4] M. A. B. Schwalbe and J. F. Webb, "Sensory basis for detection of benthic prey in two Lake Malawi cichlids," *Zoology*, vol. 117, no. 2, 2014.

[5] B. Müller, M. Glösmann, L. Peichl, G. C. Knop, C. Hagemann, and J. Ammermüller, "Bat Eyes Have Ultraviolet-Sensitive Cone Photoreceptors," *PLoS ONE*, vol. 4, no. 7, 2009.

[6] I. R. Schwab, "The evolution of eyes: major steps. The Keeler lecture 2017: centenary of Keeler Ltd," *Eye*, vol. 32, no. 2, 2017.

[7] M. F. Land and R. D. Fernald, "The Evolution of Eyes," *Annual Review of Neuroscience*, vol. 15, no. 1, 1992.

[8] T. Poggio, "Marr's computational approach to vision," *Trends in Neurosciences*, vol. 4, 1981.

[9] B. A. Wandell, *Foundations of Vision*. Oxford University Press, Incorporated, 1995.

[10] F. O. Bartell, E. L. Dereniak, and W. L. Wolfe, "The Theory and Measurement of Bidirectional Reflectance Distribution Function (BRDF) and Bidirectional Transmittance Distribution Function (BTDF)," in *Radiation Scattering in Optical Systems*, 1981.

[11] S. Shafer, "Using color to separate reflection components," *Color research and applications*, 1985.

[12] T. Gevers, A. Gijsenij, J. van de Weijer, and J.-M. Geusebroek, *Color in Computer Vision*. John Wiley & Sons, Inc., 2012.

[13] M. Nawrot, M. Ratzlaff, Z. Leonard, and K. Stroyan, "Modeling depth from motion parallax with the motion/pursuit ratio," *Frontiers in Psychology*, vol. 5, 2014.

[14] B. Julesz, *Foundations of Cyclopean Perception*. Chicago: The University of Chicago Press, 1971.

[15] J. Malik, P. Arbeláez, J. Carreira, K. Fragkiadaki, R. Girshick, G. Gkioxari, S. Gupta, B. Hariharan, A. Kar, and S. Tulsiani, "The three R's of computer vision: Recognition, reconstruction and reorganization," *Pattern Recognition Letters*, vol. 72, 2016.

[16] H. Bilen and A. Vedaldi, "Integrated perception with recurrent multi-task neural networks," in *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, vol. 29, 2016.

[17] I. Kokkinos, "UberNet: Training a Universal Convolutional Neural Network for Low-, Mid-, and High-Level Vision Using Diverse Datasets and Limited Memory," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[18] A. Gaidon, Q. Wang, Y. Cabon, and E. Vig, "Virtual Worlds as Proxy for Multi-Object Tracking Analysis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[19] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez, "The SYNTHIA dataset: A large collection of synthetic images for semantic segmentation of urban scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[20] S. R. Richter, Z. Hayder, and V. Koltun, "Playing for Benchmarks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2017.

[21] B. Ummenhofer, H. Zhou, J. Uhrig, N. Mayer, E. Ilg, A. Dosovitskiy, and T. Brox, "DeMoN: Depth and Motion Network for Learning Monocular Stereo," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[22] J. L. Schonberger, M. Pollefeys, A. Geiger, and T. Sattler, "Semantic Visual Localization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[23] K.-N. Lianos, J. L. Schönberger, M. Pollefeys, and T. Sattler, "VSO: Visual Semantic Odometry," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[24] H. Zhou, B. Ummenhofer, and T. Brox, "DeepTAM: Deep Tracking and Mapping," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[25] A. S. Baslamisli, T. T. Groenestege, P. Das, H.-A. Lê, S. Karaoglu, and T. Gevers, "Joint Learning of Intrinsic Images and Semantic Segmentation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[26] D. Skocaj, A. Leonardis, and G.-J. M. Kruijff, *Cross-Modal Learning*. Boston, MA: Springer US, 2012.

[27] H. G. Barrow and J. M. Tenenbaum, "Recovering intrinsic scene characteristics from images," *Computer Vision Systems*, 1978.

[28] J. T. Barron and J. Malik, "Intrinsic scene properties from a single RGB-D image," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[29] M. F. Tappen, W. T. Freeman, and E. H. Adelson, "Recovering Intrinsic Images from a Single Image," in *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, 2003.

[30] Y. Weiss, "Deriving intrinsic images from image sequences," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2001.

[31] E. H. Land and J. J. McCann, "Lightness and retinex theory," *Journal of Optical Society of America*, 1971.

[32] G. Ye, E. Garces, Y. Liu, Q. Dai, and D. Gutierrez, "Intrinsic video and applications," *ACM Transactions on Graphics (TOG)*, 2014.

[33] A. Meka, M. Zollhöfer, C. Richardt, and C. Theobalt, "Live intrinsic video," *ACM Transactions on Graphics (TOG)*, 2016.

[34] S. Beigpour and J. s. s. Weijer, "Object recoloring based on intrinsic image decomposition," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2011.

[35] S. Duchêne, C. Riant, G. Chaurasia, J. L. Moreno, P. Y. Laffont, S. Popov, A. Bousseau, and G. Drettakis, "Multi-view intrinsic images of outdoors scenes with an application to relighting," *ACM Transactions on Graphics (TOG)*, 2015.

[36] A. Bousseau, S. Paris, and F. Durand, "User-assisted intrinsic images," *ACM Transactions on Graphics (TOG)*, 2009.

[37] X. Yan, J. Shen, Y. He, and X. Mao, "Retexturing by intrinsic video," in *Proceedings of the International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, 2010.

[38] T. Narihira, M. Maire, and S. s. Yu, "Direct Intrinsics: Learning Albedo-Shading Decomposition by Convolutional Regression," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2015.

[39] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu, "ShapeNet: An Information-Rich 3D Model Repository," 2015.

[40] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black, "A naturalistic open source movie for optical flow evaluation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2012.

[41] P. V. Gehler, C. Rother, M. Kiefel, L. Zhang, and B. Schölkopf, "Recovering intrinsic images with a global sparsity prior on reflectance," in *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, 2011.

[42] L. Shen and C. Yeo, "Intrinsic images decomposition using a local and global sparse representation of reflectance," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[43] B. V. Funt, M. S. Drew, and M. Brockington, "Recovering shading from color images," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 1992.

[44] L. Shen, P. Tan, and S. Lin, "Intrinsic image decomposition with non-local texture cues," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[45] Q. Zhao, P. Tan, Q. Dai, L. Shen, E. Wu, and S. Lin, "A closed-form solution to retinex with nonlocal texture constraints," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2012.

[46] L. Shen, X. Yang, X. Li, and Y. Jia, "Intrinsic Image Decomposition Using Optimization and User Scribbles," *IEEE Transactions on Cybernetics*, 2013.

[47] T. Chen, Z. Zhu, A. Shamir, S.-M. Hu, and D. Cohen-Or, "3-Sweep: Extracting Editable Objects from a Single Photo," *ACM Transactions on Graphics (TOG)*, vol. 32, no. 6, 2013.

[48] K. J. Lee, Q. Zhao, X. Tong, M. Gong, S. Izadi, S. U. Lee, P. Tan, and S. Lin, "Estimation of intrinsic image sequences from image+depth video," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2012.

[49] P.-Y. Laffont and J.-C. Bazin, "Intrinsic decomposition of image sequences from local temporal variations," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2015.

[50] Y. Matsushita, S. Lin, S. B. Kang, and H. Y. Shum, "Estimating Intrinsic Images from Image Sequences with Biased Illumination," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2004.

[51] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in *International Conference on Learning Representations*, 2015.

[52] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[53] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. L. Zitnick, "Microsoft COCO: Common Objects in Context," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014.

[54] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[55] N. Mayer, E. Ilg, P. Häusser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox, "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[56] R. Grosse, M. K. Johnson, E. H. Adelson, and W. T. Freeman, "Ground truth dataset and baseline evaluations for intrinsic image algorithms," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2009.

[57] S. Kim, K. Park, K. Sohn, and S. Lin, "Unified Depth Prediction and Intrinsic Image Decomposition from a Single Image via Joint Convolutional Neural Fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[58] J. Shi, Y. Dong, H. Su, and S. s. Yu, "Learning Non-Lambertian Object Intrinsics across ShapeNet Categories," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[59] S. Bell, K. Bala, and N. Snavely, "Intrinsic images in the wild," *ACM Transactions on Graphics (TOG)*, 2014.

[60] T. Narihira, M. Maire, and S. s. Yu, "Learning Lightness from Human Judgement on Relative Reflectance," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[61] T. Zhou, P. Krähenbühl, and A. s. Efros, "Learning Data-driven Reflectance Priors for Intrinsic Image Decomposition," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2015.

[62] D. Zoran, P. Isola, D. Krishnan, and W. T. Freeman, "Learning Ordinal Relationships for Mid-Level Vision," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2015.

[63] M. Bell and W. T. Freeman, "Learning local evidence for shading and reflectance," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2001.

[64] M. F. Tappen, E. H. Adelson, and W. T. Freeman, "Estimating intrinsic component images using non-linear regression," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[65] Q. Fan, J. Yang, G. Hua, B. Chen, and D. Wipf, "A Generic Deep Architecture for Single Image Reflection Removal and Image Smoothing," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2017.

[66] X. Mao, C. Shen, and Y. Yang, "Image restoration using very deep fully convolutional encoder-decoder networks with symmetric skip connections," in *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, 2016.

[67] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2015.

[68] K. He, X. Zhang, S. Ren, and J. Sun, "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2015.

[69] L. Sevilla-Lara, D. Sun, V. Jampani, and M. J. Black, "Optical Flow With Semantic Segmentation and Localized Layers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[70] M. Bai, W. Luo, K. Kundu, and R. Urtasun, "Exploiting semantic information and deep matching for optical flow," in *Advances in Artificial Intelligence*, vol. 9910 LNCS, Springer, 2016.

[71] B. Lee, E. Erdenee, S. Jin, M. Y. Nam, Y. G. Jung, and P. K. Rhee, "Multi-class multi-object tracking using changing point detection," in *Proceedings of the European Conference on Computer Vision Workshop (ECCVw)*, vol. 9914 LNCS, 2016.

[72] X. Zhu, Y. Wang, J. Dai, L. Yuan, and Y. Wei, "Flow-Guided Feature Aggregation for Video Object Detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2017.

[73] X. Zhu, Y. Xiong, J. Dai, L. Yuan, and Y. Wei, "Deep Feature Flow for Video Recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[74] S. S. Beauchemin and J. L. Barron, "The Computation of Optical Flow," *ACM Computing Surveys*, vol. 27, no. 3, 1995.

[75] P. Baraldi, E. D. Micheli, and S. Uras, "Motion and Depth from Optical Flow," in *Procedings of the Alvey Vision Conference 1989*, 1989.

[76] A. Wedel and D. Cremers, "Optical Flow Estimation," in *Stereo Scene Flow for 3D Motion Analysis*, London: Springer London, 2011.

[77] D. Fortun, P. Bouthemy, C. Kervrann, D. Fortun, P. Bouthemy, and C. Kervrann, "Optical flow modeling and computation : a survey," *Computer Vision and Image Understanding (CVIU)*, vol. 134, 2015.

[78] M. J. Black and A. Rangarajan, "On the unification of line processes, outlier rejection, and robust statistics with applications in early vision," *International Journal of Computer Vision (IJCV)*, vol. 19, no. 1, 1996.

[79] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid, "EpicFlow: Edge-preserving interpolation of correspondences for optical flow," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[80] L. Xu, J. Jia, and Y. Matsushita, "Motion Detail Preserving Optical Flow Estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 34, no. 9, 2012.

[81] T. Brox and J. Malik, "Large Displacement Optical Flow: Descriptor Matching in Variational Motion Estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 33, no. 3, 2011.

[82] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid, "DeepFlow: Large Displacement Optical Flow with Deep Matching," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2013.

[83] A. Dosovitskiy, P. Fischer, E. Ilg, P. Häusser, C. Hazirbacs, V. Golkov, P. v.d. Smagt, D. Cremers, and T. Brox, "FlowNet: Learning optical flow with convolutional networks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2015.

[84] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[85] J. Shotton, J. Winn, C. Rother, and A. Criminisi, "TextonBoost for Image Understanding: Multi-Class Object Recognition and Segmentation by Jointly Modeling Texture, Layout, and Context," *International Journal of Computer Vision (IJCV)*, 2009.

[86] G. Csurka and F. Perronnin, "An Efficient Approach to Semantic Segmentation," *International Journal of Computer Vision (IJCV)*, 2011.

[87] D. Comaniciu and P. Meer, "Mean Shift: A Robust Approach Toward Feature Space Analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 24, no. 5, 2002.

[88] G. Mori, X. Ren, A.-A. Efros, and J. Malik, "Recovering human body configurations: combining segmentation and recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004.

[89] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 07-12-June, 2015.

[90] F. Yu and V. Koltun, "Multi-Scale Context Aggregation by Dilated Convolutions," in *International Conference on Learning Representations (ICLR)*, 2016.

[91] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 40, no. 4, pp. 834–848, 2018.

[92] D. Hoiem, A. A. Efros, and M. Hebert, "Recovering Surface Layout from an Image," *International Journal of Computer Vision (IJCV)*, vol. 75, no. 1, 2007.

[93] A. Gupta, A. A. Efros, and M. Hebert, "Blocks World Revisited: Image Understanding Using Qualitative Geometry and Mechanics," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2010.

[94] D. F. Fouhey, A. Gupta, and M. Hebert, "Data-Driven 3D Primitives for Single Image Understanding," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2013.

[95] X. Wang, D. F. Fouhey, and A. Gupta, "Designing Deep Networks for Surface Normal Estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[96] A. Bansal, B. Russell, and A. Gupta, "Marr Revisited: 2D-3D Alignment via Surface Normal Prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

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

[98] Y. He, W.-C. Chiu, M. Keuper, and M. Fritz, "STD2P: RGBD Semantic Segmentation Using Spatio-Temporal Data-Driven Pooling," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[99] J. Cheng, Y.-h. Tsai, S. Wang, M.-H. Yang, and S. W. M.-h. Yang, "SegFlow : Joint Learning for Video Object Segmentation and Optical Flow," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2017.

[100] S. Jain, B. Xiong, and K. Grauman, "FusionSeg: Learning to combine motion and appearance for fully automatic segmention of generic objects in videos," *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2017.

[101] M. Cordts, M. Omran, S. Ramos, T. Scharwächter, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The Cityscapes Dataset," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshop (CVPRw)*, vol. 3, 2016.

[102] M. Everingham, L. Van∼Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes (VOC) Challenge," *International Journal of Computer Vision (IJCV)*, vol. 88, no. 2, 2010.

[103] L. Ladický, B. Zeisl, and M. Pollefeys, "Discriminatively Trained Dense Surface Normal Estimation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014.

[104] O. H. Jafari, O. Groth, A. Kirillov, M. Y. Yang, and C. Rother, "Analyzing modular CNN architectures for joint depth prediction and semantic segmentation," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

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

[106] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[107] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid Scene Parsing Network," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[108] T.-W. Hui, X. Tang, and C. C. Loy, "LiteFlowNet: A Lightweight Convolutional Neural Network for Optical Flow Estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[109] D. Sun, X. Yang, M.-Y. Liu, J. Kautz, and J. K. Nvidia, "PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[110] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 07-12-June, 2015.

[111] N. Mayer, E. Ilg, P. Fischer, C. Hazirbas, D. Cremers, A. Dosovitskiy, and T. Brox, "What Makes Good Synthetic Training Data for Learning Disparity and Optical Flow Estimation?," *International Journal of Computer Vision (IJCV)*, vol. 126, no. 9, 2018.

[112] J. Pont-Tuset, F. Perazzi, S. Caelles, P. Arbeláez, A. Sorkine-Hornung, and L. Van Gool, "The 2017 DAVIS Challenge on Video Object Segmentation," 2018.

[113] B. K. Horn and B. G. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, no. 1-3, 1981.

[114] B. D. Lucas and T. Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision," in *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2*, 1981.

[115] A. Bruhn, J. Weickert, and C. Schnörr, "Lucas Kanade meets Horn Schunck: Combining local and global optic flow methods," *International Journal of Computer Vision (IJCV)*, vol. 61, no. 3, 2005.

[116] Y. Hu, R. Song, and Y. Li, "Efficient coarse-to-fine patchmatch for large displacement optical flow," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[117] C. Bailer, K. Varanasi, and D. Stricker, "CNN-based patch matching for optical flow with thresholded hinge embedding loss," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, 2017.

[118] J. Xu, R. Ranftl, and V. Koltun, "Accurate optical flow via direct cost volume processing," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[119] S. Meister, J. Hur, and S. Roth, "UnFlow: Unsupervised Learning of Optical Flow with a Bidirectional Census Loss," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.

[120] Y. Zou, Z. Luo, and J.-B. Huang, "DF-Net: Unsupervised Joint Learning of Depth and Flow using Cross-Task Consistency," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[121] P. Liu, M. R. Lyu, I. King, and J. Xu, "SelFlow: Self-Supervised Learning of Optical Flow," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[122] A. Ranjan, J. Romero, and M. J. Black, "Learning Human Optical Flow," in *Proceedings of the British Machine Vision Conference (BMVC)*, 2018.

[123] P. Krahenbuhl, "Free Supervision from Video Games," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[124] J. Mccormac, A. Handa, S. Leutenegger, A. J. Davison, and A. J.Davison, "SceneNet RGB-D: Can 5M Synthetic Images Beat Generic ImageNet Pre-training on Indoor Segmentation?," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2017.

[125] H.-A. Lê, A. S. Baslamisli, T. Mensink, and T. Gevers, "Three for one and one for three: Flow, Segmentation, and Surface Normals," in *Proceedings of the British Machine Vision Conference (BMVC)*, 2018.

[126] J. Janai, F. Güney, J. Wulff, M. J. Black, and A. Geiger, "Slow flow: Exploiting high-speed cameras for accurate and diverse optical flow reference data," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[127] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, 2012.

[128] D. Eigen, C. Puhrsch, and R. Fergus, "Depth Map Prediction from a Single Image Using a Multi-scale Deep Network," in *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, 2014.

[129] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2017.

[130] M. Alexa, D. Cohen-Or, and D. Levin, "As-rigid-as-possible Shape Interpolation," in *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, 2000.

[131] Y. Wang, K. Xu, Y. Xiong, and Z.-Q. Cheng, "2D Shape Deformation Based on Rigid Square Matching," *Comput. Animat. Virtual Worlds*, vol. 19, no. 3-4, 2008.

[132] M. Dvorožňák, "Interactive As-Rigid-As-Possible Image Deformation and Registration," in *The 18th Central European Seminal on Computer Graphics*, 2014.

[133] Z. DeVito, M. Mara, M. Zollöfer, G. Bernstein, C. Theobalt, P. Hanrahan, M. Fisher, and M. Nießner, "Opt: A Domain Specific Language for Non-linear Least Squares Optimization in Graphics and Imaging," *ACM Transactions on Graphics (TOG)*, 2017.

[134] T. F. H. Runia, C. G. M. Snoek, and A. W. M. Smeulders, "Real-World Repetition Estimation by Div, Grad and Curl," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[135] J. Janai, F. Güney, A. Ranjan, M. J. Black, and A. Geiger, "Unsupervised Learning of Multi-Frame Optical Flow with Occlusions," in *Proceedings of the European Conference on Computer Vision (ECCV)*, vol. Lecture No, 2018.

[136] P. Liu, I. King, M. R. Lyu, and J. Xu, "DDFlow: Learning Optical Flow with Unlabeled Data Distillation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019.

[137] N. Kholgade, T. Simon, A. Efros, and Y. Sheikh, "3D Object Manipulation in a Single Photograph Using Stock 3D Models," *ACM Transactions on Graphics (TOG)*, vol. 33, no. 4, 2014.

[138] N. Hirose, A. Sadeghian, F. Xia, R. Martín-Martín, and S. Savarese, "VUNet: Dynamic Scene View Synthesis for Traversability Estimation Using an RGB Camera," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, 2019.

[139] M. Tatarchenko, A. Dosovitskiy, and T. Brox, "Multi-view 3D Models from Single Images with a Convolutional Network," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.

[140] Y. Zheng, X. Chen, M.-M. Cheng, K. Zhou, S.-M. Hu, and N. J. Mitra, "Interactive Images: Cuboid Proxies for Smart Image Manipulation," *ACM Transactions on Graphics (TOG)*, vol. 31, no. 4, 2012.

[141] K. Rematas, C. H. Nguyen, T. Ritschel, M. Fritz, and T. Tuytelaars, "Novel Views of Objects from a Single Image," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 39, no. 8, 2017.

[142] T. Zhou, S. Tulsiani, W. Sun, J. Malik, and A. A. Efros, "View Synthesis by Appearance Flow," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.

[143] E. Park, J. Yang, E. Yumer, D. Ceylan, and A. C. Berg, "Transformation-grounded image generation network for novel 3D view synthesis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[144] S.-H. Sun, M. Huh, Y.-H. Liao, N. Zhang, and J. J. Lim, "Multi-view to novel view: Synthesizing novel views with self-learned confidence," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[145] K. Olszewski, S. Tulyakov, O. Woodford, H. Li, and L. Luo, "Transformable Bottleneck Networks," *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[146] P. E. Debevec, C. J. Taylor, and J. Malik, "Modeling and Rendering Architecture from Photographs: A Hybrid Geometry- and Image-Based Approach," in *Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques*, 1996.

[147] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski, "A Comparison and Evaluation of Multi-View Stereo Reconstruction Algorithms," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[148] M. Meshry, D. B. Goldman, S. Khamis, H. Hoppe, R. Pandey, N. Snavely, and R. Martin-Brualla, "Neural Rerendering in the Wild," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[149] T. Nguyen-Phuoc, C. Li, S. Balaban, and Y.-L. Yang, "RenderNet: A deep convolutional network for differentiable rendering from 3D shapes," in *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, 2018.

[150] D. Hoiem, A. A. Efros, and M. Hebert, "Automatic Photo Pop-Up," in *ACM Transactions on Graphics (TOG)*, 2005.

[151] C. L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, "High-Quality Video View Interpolation Using a Layered Representation," in *ACM Transactions on Graphics (TOG)*, 2004.

[152] J. Flynn, I. Neulander, J. Philbin, and N. Snavely, "Deep Stereo: Learning to Predict New Views from the World's Imagery," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[153] T. Zhou, R. Tucker, J. Flynn, G. Fyffe, and N. Snavely, "Stereo Magnification: Learning View Synthesis using Multiplane Images," in *ACM Transactions on Graphics (TOG)*, 2018.

[154] I. Choi, O. Gallo, A. Troccoli, M. H. Kim, and J. Kautz, "Extreme View Synthesis," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[155] R. Garg, B. G. V. Kumar, G. Carneiro, and I. Reid, "Unsupervised CNN for single view depth estimation: Geometry to the rescue," in *Proceedings of the European Conference on Computer Vision (ECCV)*, Springer, 2016.

[156] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised Learning of Depth and Ego-Motion from Video," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[157] C. Godard, O. Mac Aodha, and G. J. Brostow, "Unsupervised Monocular Depth Estimation with Left-Right Consistency," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[158] C. Godard, O. Mac Aodha, M. Firman, and G. J. Brostow, "Digging into Self-Supervised Monocular Depth Prediction," *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[159] A. Johnston and G. Carneiro, "Single View 3D Point Cloud Reconstruction using Novel View Synthesis and Self-Supervised Depth Estimation," in *Proceedings of the International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, 2019.

[160] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Nets," in *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, 2014.

[161] P. Isola, J.-Y. Y. Zhu, T. Zhou, A. A. Efros, and B. A. Research, "Image-to-Image Translation with Conditional Adversarial Networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[162] X. Zhu, Z. Yin, J. Shi, H. Li, and D. Lin, "Generative Adversarial Frontal View to Bird View Synthesis," in *Proceedings of the International Conference on 3D Vision (3DV)*, IEEE, 2018.

[163] K. Regmi and A. Borji, "Cross-View Image Synthesis Using Conditional GANs," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[164] B. Kicanaoglu, R. Tao, and A. W. M. Smeulders, "Estimating small differences in car-pose from orbits," in *Proceedings of the British Machine Vision Conference (BMVC)*, 2018.

[165] Y. Galama and T. Mensink, "IterGANs: Iterative GANs to learn and control 3D object transformation," *Computer Vision and Image Understanding (CVIU)*, vol. 189, 2019.

[166] X. Xu, Y.-C. Chen, and J. Jia, "View Independent Generative Adversarial Network for Novel View Synthesis," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[167] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial Transformer Networks," in *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*.

[168] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.

[169] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.

[170] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *tip*, vol. 13, no. 4, 2004.

[171] J.-M. Geusebroek, G. J. Burghouts, and A. W. Smeulders, "The Amsterdam Library of Object Images," *International Journal of Computer Vision (IJCV)*, vol. 61, no. 1, 2005.

[172] J. K. Aggarwal and N. Nandhakumar, "On the computation of motion from sequences of images-A review," *Proceedings of the IEEE*, vol. 76, no. 8, 1988.

[173] J. L. Barron, D. J. Fleet, S. S. Beauchemin, and T. A. Burkitt, "Performance of optical flow techniques," *International Journal of Computer Vision (IJCV)*, vol. 12, no. 1, 1994.

[174] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene Parsing through ADE20K Dataset," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[175] G. Neuhold, T. Ollmann, S. Rota Bulò, and P. Kontschieder, "The Mapillary Vistas Dataset for Semantic Understanding of Street Scenes," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2017.

[176] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[177] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niessner, M. Savva, S. Song, A. Zeng, and Y. Zhang, "Matterport3D: Learning from RGB-D Data in Indoor Environments," *International Conference on 3D Vision (3DV)*, 2017.

[178] B. Kovacs, S. Bell, N. Snavely, and K. Bala, "Shading Annotations in the Wild," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[179] R. Tylecek, T. Sattler, H.-A. Lê, T. Brox, M. Pollefeys, R. B. Fisher, and T. Gevers, "The Second Workshop on 3D Reconstruction Meets Semantics: Challenge Results Discussion," in *Proceedings of the European Conference on Computer Vision Workshop (ECCVw)*, 2019.

[180] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, "Learning Deep Features for Scene Recognition using Places Database," in *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, 2014.

[181] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, A. Yuille, R. Urtasun, and A. Yuille, "The Role of Context for Object Detection and Semantic Segmentation in the Wild," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[182] S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. J. Black, and R. Szeliski, "A Database and Evaluation Methodology for Optical Flow," *International Journal of Computer Vision (IJCV)*, vol. 92, no. 1, 2011.

[183] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor Segmentation and Support Inference from RGBD Images," 2012.

[184] I. Armeni, A. Sax, A. Zamir, and S. Savarese, "Joint 2D-3D-Semantic Data for Indoor Scene Understanding," *ArXiv e-prints*, 2017.

[185] B. Leibe, N. Cornelis, K. Cornelis, and L. Van Gool, "Dynamic 3D Scene Analysis from a Moving Vehicle," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[186] G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla, "Segmentation and Recognition Using Structure from Motion Point Clouds," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2008.

[187] F. Warburg, S. Hauberg, M. Lopez-Antequera, P. Gargallo, Y. Kuang, and J. Civera, "Mapillary Street-Level Sequences: A Dataset for Lifelong Place Recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[188] G. R. Taylor, A. J. Chosak, and P. C. Brewer, "OVVV: Using Virtual Worlds to Design and Evaluate Surveillance Systems," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[189] M. Mueller, N. Smith, and B. Ghanem, "A Benchmark and Simulator for UAV Tracking," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.

[190] B. Kaneva, A. Torralba, and W. T. Freeman, "Evaluation of image features using a photorealistic virtual world," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2011.

[191] S. R. Richter, V. Vineet, S. Roth, and V. Koltun, "Playing for Data: Ground Truth from Computer Games," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.

[192] A. Valada, G. Oliveira, T. Brox, and W. Burgard, "Deep Multispectral Semantic Scene Understanding of Forested Environments using Multimodal Fusion," in *International Symposium on Experimental Robotics (ISER)*, 2016.

[193] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, "SMPL: A Skinned Multi-Person Linear Model," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 6, 2015.

[194] J. Han, S. Karaoglu, H.-A. Lê, and T. Gevers, "Improving Face Detection Performance with 3D-Rendered Synthetic Data," in *Proceedings of IEEE Conference on Pattern Recognition (ICPR)*, 2020.

[195] J. Weber and J. Penn, "Creation and rendering of realistic trees," *ACM Transactions on Graphics (TOG)*, 1995.

[196] C. Hewitt, *Procedural Generation of Tree Models for Use in Computer Graphics*. PhD thesis, Cambridge Trinity College, 2017.

[197] R. Barth, J. IJsselmuiden, J. Hemming, and E. J. van Henten, "Data synthesis methods for semantic segmentation in agriculture: A Capsicum annuum dataset," *Computers and Electronics in Agriculture*, vol. 144, 2018.

[198] K. Perlin, "An Image Synthesizer," *ACM Transactions on Graphics*, vol. 19, no. 3, 1985.

[199] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[200] T. Sattler, R. Tylecek, T. Brox, M. Pollefeys, and R. B. Fisher, "3D Reconstruction meets Semantics – Reconstruction Challenge," in *ccvwl*, ICCV Workshops, 2017.

[201] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao, "Deep Ordinal Regression Network for Monocular Depth Estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[202] W. Yin, Y. Liu, C. Shen, and Y. Yan, "Enforcing geometric constraints of virtual normal for depth prediction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[203] B. Lonergan, *Insight: a Study of Human Understanding*. London: Longmans, 1958.

[204] H. Montgomery, "Perception: The interface between cognition and the external world," *Scandinavian Journal of Psychology*, vol. 32, no. 1, 1991.

[205] X. Li, Y. Zhou, T. Wu, R. Socher, and C. Xiong, "Learn to grow: A continual structure learning framework for overcoming catastrophic forgetting," in *Proceedings of the 36th International Conference on Machine Learning*, vol. 97 of *Proceedings of Machine Learning Research*, pp. 3925–3934, PMLR, 09–15 Jun 2019.

[206] W. Masarczyk and I. Tautkute, "Reducing catastrophic forgetting with learning on synthetic data," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020.

[207] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, "Continual lifelong learning with neural networks: A review," *Neural Networks*, vol. 113, pp. 54–71, 2019.

[208] R. Szeliski, *Computer vision algorithms and applications*. New York: Springer, 2011.

[209] J. Hestness, N. Ardalani, and G. Diamos, "Beyond Human-Level Accuracy: Computational Challenges in Deep Learning," in *Proceedings of the 24th Symposium on Principles and Practice of Parallel Programming*, 2019.

[210] K.-F. Lee, *AI Superpowers: China, Silicon Valley, and the New World Order*. Boston, New York: Houghton Mifflin Harcourt, 2018.

Faithful friends are a sturdy shelter; whoever finds one finds a treasure.
Faithful friends are beyond price, no amount can balance their worth.

Sirach 6:14–15

Acknowledgments

This journey of pursuing the doctorate is indeed the toughest that I have endured so far in my life. It would not have been possible without the tremendous supports, intellectually, physically, mentally, and spiritually, downpoured by many people. This part of the thesis is not intended to just acknowledge that, but dedicated as a commemoration hall for all the cherished people and the precious memories that they have bestowed during the journey.

First and foremost gratitude is due to my promoter, Prof. Theo Gevers, who has picked me for this journey and let me join him in his amazing project. Theo, I know that I was not the best PhD student you deserved and many of your ideas could have been materialized by someone else more qualified. But you see it through to the end with tremendous patience to my clumsiness and shortcomings. Thank you for all the pushes. They were not fun but they showed that you cared and meant for my bests. Thank you for all the jokes despite our different tastes, for they show that you could be close. Thank you for all your supports. It was truly an honor to be your student.

My deepest gratitude extends to my co-promoter, mentor, and role model researcher, Dr. Thomas Mensink. I have to admit that I was little put off and overwhelmed by your critical questions when I first came, yet I have grown to appreciate and go after them observing your curiosity and interest in discovering and understanding. Working with you teaches me that true research seeks knowledge and the truth and that is the real goal we are after. Thank you for the empathy and the inspiration that helped straighten out and rekindle my burned-out motivation. Thanks for the tremendous patience and sorry (for the thousandth time) for all the disappointments. I was so fortunate to have you as mentor.

I am grateful to the members of my PhD defense committee: Prof. Robert Fisher (it was a pleasure to collaborate in the TrimBot2020 project), Prof. Sébastien Lefèvre (it is now a thrill to be on board with you), Prof. Cees Snoek, Dr. Arnoud Visser, and Dr. Sezer Karaoğlu, for accepting to be in my committee in such short notice and the flexibility that allows for the timely defense. It is a pleasure and an honor to have you all on my committee.

Deep gratefulness goes to Dr. Leo Dorst, for being the stretching hand during the hard time and the encouragement that got me back on the horse. I am indebted to your sensitivity and thoughtfulness. Thanks also for the brief but inspired lecture on mathematical morphology and all the assistance. I wish I had more time to learn some math with you.

Special thanks are to my two terrific paranymphs, labmates, and friends: Anıl Sırrı Başlamışlı and Ngô Lê Minh. Thanks, Anıl, for being by my side in this journey (and the office), being the "AnbelievabLe" sounding board for my everyday half-cocked ideas, and the true appreciator of all my cringe-making jokes. I learned from your research as much as your calmness and open-mindedness. And Minh, you are the geekiest that I known, I admire your skillfulness and the willing to go at length with all the technical problems. Thanks for perking up those dark days and the pep talks during coffee and dinner breaks. It was pleasant to converse with you on these topics and discover how knowledgeable you are.

Hanan M. R. A. ElNaghy, you are my best friend at this place and one of the most benevolent hearts that I know. Thank you for presenting the soft part of this journey and lending your ears with great empathy to all my emotions. Thank you for our friendship. Our lunchtimes would be the pearl at the heart of this commemoration hall.

Wei Zeng Gē and Jian Han Gē, you guys are super awesome. Thank you for the good laughs and the culture talks. I owe you big time for all the rescheduling attempts of the reading and presentation groups (I know it was not easy, but they made this journey breathable), and Jian Gē, for the video-game enticement (you know what I am talking about).!

Big thanks to all you guys who have been part of my PhD life: Dr. Dennis Koelma, for being there (ready to educating me) with whatever technical problem that I managed to get myself into, and making the Linux system the 2nd-most thing that I've learned in these 4 years; Rick Groenendijk, for the willing, time, and effort in translating the thesis summary; Partha Das, for the good talks (most of which were laughs), and letting me pick your brain (and Lua skills), enjoy my precious spot; Dr. Shaodi You, for the great book; Wei Wang and Yahui Zhang for willing to help with the machine connection when I was away; William Thong, for sharing authentic French experiences; Berkay Kıcanaoğlu, for "teaching" me yemek (sorry, Anıl, it's him); Yunlu Chen, for the sharing the research interest and nice discussion; Mert Kılıçkaya, for all the nice talks and empathy; and Gjorgji Strezoski, for coming by every morning for a handshake. I hope that we can have a chance to collaborate.

I would like to thank the colleagues and friends in the TrimBot2020 consortium, especially Dr. Radim Tylecek, Dr. Nikolaus Mayer, Dr. Nicola Strisciuglio, Hanz Cuevas Velasquez, Nanbo Li, and all the others, for the opportunity of collaboration and the privilege of being the first-hand witness of the brilliant minds at work.

Many thanks to Dr. Rafael Bidarra, for proofreading the thesis, for the spiritual advises, and for the various inspiring talks on science and faith. Thanks also to Aristide Black, my English "teacher" and Vietnamese "student", for proofreading and for keeping me company in many weekends with language and culture discussion.

My heartfelt gratitude goes to the Parish of Blessed Trinity Amsterdam, my second family here, who have wholeheartedly welcomed and accompanied me, physically and spiritually, particularly the late Reverend Father Thomas Murray, the Reverend Father Peter Klos, Pearl and Claude Hoogland, Caroline Thangavelu, Ina Paveela Joseph, and Joseph Siengo (for sharing the joy of catechizing the kids), and all other parishioner friends.

I would also like to include all my friends, dear, here and far away, who, in one way or another, have encouraged me in the trials and made this journey more than just about papers. Special thanks to my friends in Hội HS lang thang Hà Lan, especially Vũ Hải Đăng, Vũ Minh Phương, Quách Thành Lâm, Trần Công Nguyên, Nguyễn Ngọc Mai, Vũ Thị Vân Anh, and Linh Chi for the the parties and hangouts; in the VN-AMS data science group, especially anh Bùi Quốc Chính , anh Phạm Việt Thắng, chị Vũ Thùy Dương, anh Chu Mạnh Dũng, anh Đinh Việt Cường, anh Minh Lê for the engaging meetings and inspired talks. Thank you, anh Lương Vĩ Minh, for considering me a brother and for all consolation during the hard time, Đoàn Thảo Vy, for the spiritual discussions and countless inspiration. Shout-outs to my long-time friends, Mạc Cự Khôi Nguyên, Bùi Quốc Minh, Hoàng Xuân Quang Nhật (thanks for the visits and photo-taking), Phạm Trường An, Nguyễn Quốc Khang, Nguyễn Hoàng Khôi, Nguyễn Hoàng Nhật Minh, Nguyễn Ngọc Nam Phương, Lê Anh Thiên Tú (for all the intergalactic overhyped talks), and all the unsung heroes who always have me in their best thoughts and prayers.

I owe deep gratitude to my former teachers and professors, especially Prof. Dương Nguyên Vũ and Assoc/Prof. Trần Minh Triết, who have wholeheartedly paved the steps of the ways leading me to this journey. Thank you for all the lessons, encouragement, recommendation, and inspiration. I was fortunate to have your guidance.

The dearest thoughts are to my cousins, especially Trần Ngọc Dung Hạnh, and all my relatives who, despite the long geographical distance, always have me in their hearts and accompany me in prayers. Thank you for always being there to love and cheer me up.

Une pensée particulière revient à les Philippon: Jean, Cathy, Ewen, Maël, Franciske, Yến Trang, aussi bien qu'Agnès Lê-Thị et Yann Méar. Merci d'être ma famille élargie, de me recevoir à chaque vacances avec des cœurs chaleureux et pour les superbes repas.

Con cám ơn ba mẹ vì tất cả tình yêu và sự hi sinh, đã luôn chăm sóc con từ những giây phút đầu tiên, đã dạy dỗ như một người thầy và chơi cùng con như một người bạn.

Con cám ơn mẹ vì đã luôn đặt việc học của con lên hàng đầu, đã cùng học và kiên trì với con trong những bài học đầu tiên trên căn gác ở nhà, là hình mẫu để con làm tốt hơn là "giỏi hơn đứa kế bên", đã luôn động viên để con cố gắng, nhưng lại lo lắng và chăm sóc cho những lúc con mỏi mệt.

Con cám ơn ba vì đã đồng hành cùng việc học của con từ hồi còn nhỏ xíu, luôn khơi gợi trí tò mò và không ngừng ủng hộ con mở rộng học hỏi, những quyển sách tham khảo, mà ba phải bỏ giờ nghiên cứu và hi sinh để mua, cám ơn ba vì những giây phút cả nhà cùng ngồi lại và tính toán chi tiêu, vì "việc học của con là quan trọng."

Con cám ơn mẹ vì đã luôn yêu quý sách, những quyển sách mẹ mua và những quyển sách mẹ giữ lại, dù nhà mình không đủ lớn, nhưng mẹ luôn giữ lại, mọi thứ, cả ký ức và kỷ niệm, để một ngày nào đó con có cơ hội dùng hay học.

Con cám ơn ba vì đã dạy con phải cố gắng hết mình nhưng không tạo áp lực, và giúp con biết yêu thích sự hiểu biết: nhờ ba mà việc học và được hiểu biết, từ lúc nào, đã luôn là niềm vui hơn là trách nhiệm.

Con cám ơn mẹ vì đã đặt niềm vui và sự bình an của con lên trên, ở bên cạnh con trong những lúc buồn, vui, thành công, thất bại, không quan trọng hóa các phần thưởng, và không nghiêm trọng hóa các sai lầm, miễn là con vui và bình an.

Con cám ơn ba vì đã không tìm vinh dự trong các phần thưởng, hay đặt cho con những chuẩn mực vật chất, nhưng vui trong sự phát triển, và dạy con sống làm Người, như Người, điều mà có lẽ con vẫn còn phải học, phải sống suốt cả đời con.

The highest and all tributes are to my Heavenly Father, who bestows on me the gift of life, health, intellect, and countless others, who makes this journey realized from the beginning and sustains me to the last, who does not invasive intervene along the way but gently organizes the entire nexus of causes and events that lead to this day. *O Lord, it is You who have accomplished all that we have done.* (Isaiah 26:12)

'Tis grace hath brought me safe thus far, and grace will lead me home.
Amazing Grace