University of Modena and Reggio Emilia
XXXII Cycle of the International Doctorate School in
Information and Communication Technologies

Doctor of Philosophy Dissertation in
Computer Engineering and Science

Smart Cities: Bridging Driver's, Vehicle and Infrastructure Viewpoints

Andrea Palazzi

Supervisor: Prof. Rita Cucchiara
PhD Course Coordinator: Prof. Sonia Bergamaschi

Modena, 2020

Review committee composed of:

Prof. Alberto Del Bimbo
University of Firenze

Iuri Frosio, PhD
NVIDIA Research

To Francesco, Davide and Luca

(in chronological order)

The next aspect of science is its contents, the things that have been found out. This is the yield. This is the gold. This is the excitement, the pay you get for all the disciplined thinking and hard work. The work is not done for the sake of an application. It is done for the excitement of what is found out. Perhaps most of you know this. But to those of you who do not know it, it is almost impossible for me to convey in a lecture this important aspect, this exciting part, the real reason for science. And without understanding this you miss the whole point. You cannot understand science and its relation to anything else unless you understand and appreciate the great adventure of our time. You do not live in your time unless you understand that this is a tremendous adventure and a wild and exciting thing.

Richard Feynman, *The Meaning of It All*

Abstract

In this thesis we dive deep into visual understanding of the urban scene with a smart city scenario in mind. The problem is framed from multiple points of view, following a zoom out approach. First, we put our focus on the inside of the vehicle, investigating the dynamics of human attention during the driving task. We show how a deep neural network can be designed and trained to replicate the human attentional behavior while driving. We also study which parts of the scene are the most likely to capture the attention of the human driver, and in which measure these patterns can be automatically learnt from the data. A large-scale dataset of human fixations while driving in a real world scenarios - collected and made available for the research community for the first time - enables the aforementioned studies. The focus is then shifted from the vehicle to the infrastructure point of view. Vehicles are now viewed from the outside, as one of the most important agents which populate the urban scene. In this frame we introduce novel methods to infer vehicles characteristics - such as identity, model, 3D pose and occupancy - which are essential to comprehend the scene. Finally, we propose a novel framework to exploit these information to 'hallucinate' novel views of the vehicles and of the urban scene in its whole, paving the way for multiple significant applications in the domain of urban scene understanding.

Abstract (Italian)

In questa tesi si studia il problema della comprensione visuale della scena urbana in uno scenario di smart city. In questo processo il problema è inquadrato da più punti di vista, seguendo un approccio di zoom out. Per prima cosa l'attenzione è posta all'interno del veicolo stesso, indagando le dinamiche dell'attenzione umana durante la guida. Si mostra come una rete neurale profonda possa essere progettata e allenata per replicare il comportamento attentivo del guidatore umano. Si studia anche quali parti della scena abbiano maggiori probabilità di catturare l'attenzione del conducente umano e in quale misura questi pattern possano essere appresi automaticamente dai dati. Un vasto dataset di punti di fissazione dei guidatori - raccolti in scenari reali e resi disponibili alla comunità di ricerca per la prima volta - è ciò che consente gli studi sopra citati. L'attenzione viene poi spostata dal punto di vista del veicolo a quello dell'infrastruttura cittadina. I veicoli sono ora visti dall'esterno, come uno dei più importanti agenti che popolano la scena urbana. In questo scenario introduciamo nuovi metodi per inferire le caratteristiche del veicolo - identità, modello, posa 3D e occupazione spaziale - che sono essenziali per una piena comprensione della scena. Infine, proponiamo un nuovo framework che consente di sfruttare queste informazioni per generare nuovi possibili aspetti visuali dei veicoli e della scena nel suo complesso, aprendo la strada a molteplici applicazioni nel campo della comprensione della scena urbana.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The number of connected 'things' we share our life with is growing exponentially. Although it is hard to agree on a precise figure, it is safe to say that the number of connected devices already surpasses several billions [158]. Cities are among the hottest spots of this densely connected network, as the world population congregates in larger and larger urban areas.

Meanwhile, the recent technological progresses - particularly in wireless networks and AI in the broadest sense - are allowing these devices to have an increasing capability to perceive and understand the world. Given the favourable circumstances, there is reason to believe in a new generation of smart cities in which vehicles and infrastructure communicate closely to alleviate the historical issues afflicting mobility in dense urban areas, such as fatal accidents and heavy traffic. For sure, a tight interconnection between vehicles and infrastructure will be necessary for this potential to unfold.

In this work we analyze some of the challenges posed by this interaction and propose novel solutions to approach them. In particular, the thesis is organized following a zoom-out approach. We start our research work from inside the vehicle itself, diving deep into the human attentional dynamics during the driving task. We replicate the human attentional behavior through a deep neural network, trained to infer which areas of the urban scene would be more likely to attract the attention of the human

driver in a certain context. In this process, we were the first to collect and publicly release to the research community a large-scale dataset of human fixations while driving in real world scenarios.

We then progressively zoom out in the process of moving from the vehicle to the infrastructure point of view. First, we show how we can exploit object detection in a vehicle dashboard camera view to learn a semantic-aware occupancy map of the scene. We then present novel methods for identifying and re-identifying vehicles, as well as for inferring their poses in the 3D world. Finally, we present a novel image synthesis framework, which is particularly designed for the generation of novel views of vehicles and man-made rigid objects in a broader sense. Based on these results, we conclude showing how this framework could be exploited to 'hallucinate' novel views of the whole urban scene. We foresee that this imagination capability - which enable not only to predict, but also to visually depict different possible evolutions of the urban scene - might have significant applications in many domains; surveillance, vehicle re-identification and forensics to name a few.

Chapter 2

Literature Survey

In the following sections we briefly report other research approaches which are related to the topics tackled in this thesis. Although the list could be much longer, we choose to limit ourselves to the methods which are either most relevant for the community or more strictly related to the proposed algorithms.

The rest of the chapter is organized following the flow of the thesis: First we review methods for driver's gaze estimation (Sec. 2.1), then we move to approaches for urban scene understanding (Sec. 2.2). We appreciate that the latter term is very broad. Here we restrict to domains for which contributions have been proposed during the PhD: vehicle pose estimation and viewpoint changes, vehicle re-identification, and vehicle novel view generation.

2.1 Predicting the Driver's Focus of Attention

Attention in images

Coherently with psychological literature, that identifies two distinct mechanisms guiding human eye fixations [182], computational models for Focus of Attention (FoA) prediction branch into two families: *top-down* and *bottom-up* strategies. Former approaches aim at highlighting objects and cues that could be meaningful in the context of a given ongoing task. For

this reason, such methods are also known as *task-driven*. Usually, *top-down* computer vision models are built to integrate semantic contextual information in the attention prediction process [183]. This can be achieved by either merging estimated maps at different levels of scale and abstraction [62], or including a-priori cues about relevant objects for the task at hand [208, 57, 45]. Human focus in complex interactive environments (*e.g.* while playing videogames) [138, 139, 16] follows task-driven behaviors as well.
Conversely, *bottom-up* models capture salient objects or events naturally popping out in the image, independently of the observer, the undergoing task and other external factors: that is, ceteris paribus, how much each pixel of a scene attracts the observer's attention. This task is widely known in literature as *visual saliency prediction*. In this context, computational models focus on spotting visual discontinuities, either by clustering features or considering the rarity of image regions, locally [159, 114] or globally [1, 224, 32]. For a comprehensive review of visual attention prediction methods, we refer the reader to [14]. Recently, the success of deep networks involved both task-driven attention and saliency prediction, as models have become more powerful in both paradigms, achieving state-of-the-art results on public benchmarks [91, 102, 74, 37, 38].

Attention in videos

In video, attention prediction and saliency estimation are more complex compared to still images since motion heavily affects human gaze. Some models merge bottom-up saliency with motion maps, either by means of optical flow [232] or feature tracking [224]. Other methods enforce temporal dependencies between bottom-up features in successive frames. Both supervised [232, 117] and unsupervised [118, 200, 201] feature extraction can be employed, and temporal coherence can be achieved either by conditioning the current prediction on information from previous frames [154] or by capturing motion smoothness with optical flow [232, 117]. While deep video saliency models still lack, an interesting work is [8], which relies on a recurrent architecture fed with clip encodings to predict the fixation map by means of a Gaussian Mixture Model (GMM). Nevertheless, most methods limit to bottom-up features accounting for just visual discontinuities in terms of textures or contours. The model proposed in this thesis (see Sec. 3.2 and Sec.3.3), instead, is specifically tailored to the driving task

and fuses the bottom-up information with semantics and motion elements that have emerged as attention factors from the analysis of the DR(eye)VE dataset.

Attention in the driving context

Prior works addressed the task of detecting saliency and attention in the specific context of assisted driving. In such cases, however, gaze and attentive mechanisms have been mainly studied for some driving subtasks only, often acquiring gaze maps from on-screen images. Bremond *et al.* [166] presented a model that exploits visual saliency with a non-linear SVM classifier for the detection of traffic signs. The validation of this study was performed in a laboratory non-realistic setting, emulating an in-car driving session. A more realistic experiment [19] was then conducted with a larger set of targets, *e.g.* including pedestrians and bicycles.
Driver's gaze has also been studied in a pre-attention context, by means of intention prediction relying only on fixation maps [142]. The study in [191] inspects the driver's attention at T junctions, in particular towards pedestrians and motorbikes, and exploits object saliency to avoid the *looked-but-failed-to-see* effect. In absence of eye tracking systems and reliable gaze data, head orientation can be used as proxy to infer eyes off-the-road and other dangerous driving habits. In this line of work we can collocate methods such as [52, 180, 194, 12], which focus on drivers' head, detecting facial landmarks to predict head orientation. Although such mechanisms are often more robust to varying lighting conditions and occlusions, there is no certainty about the adherence of predictions to the true gaze during the driving task.

Datasets on attention and saliency

Many image saliency datasets have been released in the past few years, improving the understanding of the human visual attention and pushing computational models forward. Most of these datasets include no motion information, as saliency ground truth maps are built by aggregating fixations of several users within the same still image. Usually, a Gaussian filtering post-processing step is employed on recorded data, in order to smooth such fixations and integrate their spatial locations. Some datasets, such as the

MIT saliency benchmark [21], were labeled through an eye tracking system, while others, like the SALICON dataset [80] relied on users clicking on salient image locations. We refer the reader to [15] for a comprehensive list of available datasets.
On the contrary, datasets addressing human attention prediction in video still lack. Up to now, *Action in the Eye* [117] represents the most important contribution, since it consists in the largest video dataset accompanied by gaze and fixation annotations. That information, however, is collected in the context of action recognition, so it is heavily task-driven. A few datasets address directly the study of attention mechanisms while driving, as summarized in Tab. 3.1. However, these are mostly restricted to limited settings and are not publicly available. In some of them [166, 191] fixation and saliency maps are acquired during an in-lab simulated driving experience. In-lab experiments enable several attention drifts that are influenced by external factors (*e.g.* monitor distance and others) rather than the primary task of driving [179].
A few in-car datasets exist [19, 142], but were precisely tailored to force the driver to fulfill some tasks, such as looking at people or traffic signs. Coarse gaze information is also available in [52], while the external road scene images are not acquired. We believe that the dataset presented in [142] is, among the others, the closer to our proposal. Yet, video sequences are collected from one driver only and the dataset is not publicly available. Conversely, our `DR(eye)VE` dataset (see Sec. 3.1) - featuring 74 videos of five minutes each, for a total of more than 500K frames annotated with driver fixation points via an eye tracking device - is the first dataset addressing driver's focus of attention prediction which is made publicly available. Furthermore, it includes sequences from several different drivers and presents a high variety of landscapes (*i.e.* highway, downtown and countryside), lighting and weather conditions.

2.2 Urban Scene Understanding

Vehicle re-identification

Vehicle re-identification is the problem of matching the identities of vehicles across non-overlapping views from different cameras. Despite its huge application significance, this task is far from being solved and it still features many open challenges. Recently Liu et al. [104] proposed an hybrid

approach to vehicle re-identification, based on features fusion (FACT). The method is called hybrid as it leverages both handcrafted features (Bag of Words (BoW), SIFT [110] and color names [192]) and learned features by means of deep learning models. In particular, they rely on GoogleNet [176] network to extract high level semantic features such as the number of doors, the number of seats or the shape. For this purpose they fine-tune the network on CompCars dataset [219]. After merging texture, color and semantic features the euclidean distance is used to match the prediction against a features gallery. In a successive work, the same authors [105] introduce two new features - an embedding of the license plate and spatio-temporal properties - which are concatenated with the former. Notably, a Siamese network is trained to determine if a couple of plates actually point to the same vehicle.
With the rise of Triplet Network architectures in various challenging computer vision domains [107, 165, 163, 31] with promising results, Hoffer et al.[73] revisited the proposal from [101] to include the concept of vehicles classes. The successive work [228] further improves triplet-wise training procedure by introducing a custom classification-oriented loss to augment with the original triplet loss as well as a new triplet sampling method based on pairwise images.
In this frame, in Sec. 4.1 we propose to leverage a Triplet Network to learn a feature space where visually similar vehicles are clustered together, whereas the ones visually different are kept far apart. While using Triplet Networks for vehicle re-identification is not novel itself, in our work we mainly focus on presenting an overall pipeline that can be deployed for re-identifying vehicles across completely different views. Furthermore, we detail how a re-identification network can be trained even in absence of manually labelled data, as it was the case of the NVIDIA AI City Challenge we participated in.

Object 3D shape reconstruction

Ten years after its public release, the ImageNet [40] dataset has been the essential ingredient to many advances, as for the first time enough data were available for training very large (deep) models which in turn shook many benchmarks [167, 69, 145, 27]. More recently the large-scale database of synthetic models ShapeNet [23] dataset is having an analogous impact on the 3D community, showing that, in presence of enough data, 3D geometry

and deep learning can be integrated successfully [172, 211, 34, 171, 97, 6]. One of the areas in which this marriage is being fertile the most is the one of estimating the 3D shape of an object given an image, either in explicit (e.g. generating voxels), or implicitly (e.g. generating novel views of the same object). Indeed, pre deep learning methods [17, 65, 89, 195, 140, 170] often need multiple views at test time and rely on the assumption that descriptors can be matched across views [51, 2], handling poorly self-occlusions, lack of texture [156] and large viewpoint changes [111]. Conversely, more recent works [172, 211, 34, 171, 97, 6] are built upon powerful deep learning models trained on virtually infinite synthetic data rendered from ShapeNet [23].
From a high level perspective, we can distinguish at least two categories of methods; i) the ones that learn an implicit representation of object pose and volume and then decode it by means of another deep network [177, 43, 34, 218] and ii) the ones that infer from the image a valid 3D representation (e.g. voxel-based) that can be re-projected by means of a differentiable renderer [217, 189, 55, 205] to eventually measure its consistency w.r.t. the input image. Works leveraging the latter approach are strictly related to our proposed method in that they all found different ways to back-propagate through the renderer in order to correct the predicted object volume. [217], [55] and [205] take inspiration from the *spatial transformer network* [78] in the way the predicted volume is sampled to produce the output silhouette, even though they differ in the way the contribution of each voxel is counted for each line of sight. Rendering process proposed in [149] has to be trained via REINFORCE [206] since it is not differentiable; [189] frames the rendering phase in a probabilistic setting and define ray potential to enforce consistency.
Our method presented in Sec. 4.3 differs substantially from all these works in several aspects. First, we keep the volume fixed and back-propagate through the renderer to correct the object pose, while the aforementioned works project the predicted 3D volume from a pre-defined set of poses (e.g. 24 azimuthal angles 0°, 15°, ... 345° around y-axis) and back-propagate the alignment error to correct the volume. Furthermore, while all these works use ray-tracing algorithm for rendering, our work is the first to propose a *differentiable raster-based renderer*. Eventually, all mentioned works represent the volume using voxels, which is inefficient and redundant since most of valuable information is in the surface [168], while we use its natural parametrization by vertices and faces, i.e. the mesh.
Eventually, more recent works [209, 238] have shown that 2.5D sketches can

be a useful intermediate representation to bridge 2D and 3D worlds as well as to alleviate the gap between synthetic and real-world data. In particular, in [238] the 2.5D sketch consists of both a silhouette and a depth image rendered from a learnt low-resolution voxel grid by means of a differentiable ray-tracer. While this method is appealing for its geometrical guarantees, it is limited by a number of factors: i) it requires a custom *differentiable* ray-tracing module; ii) footprint of voxel-based representations scales with the cube of the resolution despite most of the information lying on the surface [168, 130]; iii) errors in the 3D voxel grid naturally propagate to the 2.5D sketch. We also follow this line of work to provide soft 3D priors to the synthesis process. Conversely, in the semi-parametric setting presented in Sec. 4.4 the 2.5D sketches are simply additional inputs which do not require to be differentiable and can thus be rendered from arbitrary viewpoints using standard rendering engines.

Object pose estimation

Image formation is essentially a lossy process, as during the perspective projection we loose a lot of information about 3D structure of the captured scene. For this reason, recovering back the 6-Degrees of Freedom (6-DoF) pose of an object from a single image is extremely challenging. The task of object pose estimation, traditionally framed as a Perspective-n-Points (PnP) correspondence problem between the 3D world points and their 2D projections in the image [95, 120], was recently re-framed in a deep learning context with analogous effectiveness. With respect to descriptor-based methods [35, 36, 111], recent methods relying on convolutional neural networks [108, 172, 188] can solve ambiguities and handle occluded keypoints thanks to their high representational power and composite field of view. Indeed, these have already shown impressive results in specific tasks such as the one of human pose estimation [123, 204, 184, 235]. Building upon this success, approaches as [239, 136] combine CNN-extracted keypoints and deformable shape models in a unique optimization framework to jointly estimate the object pose and shape. Differently from all these works, the pose estimation method proposed in Sec. 4.3 integrates object shape and pose estimation and model fitting into a coherent end-to-end differentiable framework. In particular, the differentiable renderer is used to correct the 6-DoF object pose estimation by back-propagating 2D information on silhouette alignment error. Furthermore, in Sec. 4.3 we abandon the

redundant voxel representation in favor of meshes, which are lightweight and better tailored to represent 3D models [168].

Learning the occupancy grid around the vehicle

Few works in literature tackle the problem of learning the vehicle's surrounding occupancy map from a single monocular image. From an application standpoint, most of these approaches aim at helping drivers during parking manoeuvres and rely on both geometry and computer vision to merge information from usually multiple cameras mounted on the vehicle. In particular, in [99] a perspective projection image is transformed into its corresponding bird's eye view, through a fitting parameters searching algorithm. In [106] authors took a dynamic programming approach, exploiting six calibrated fish eye cameras to compose a unique overall image of the scene. In [124] were described algorithms for creating, storing and viewing surround images, thanks to synchronized and aligned different cameras. [175] proposed a camera model based algorithm to reconstruct and view multi-camera images. In [187], an homography matrix is used to perform a coordinate transformation: visible markers are required in input images during the camera calibration process. Recently, [225] proposed a surround view camera solution designed for embedded systems, based on a geometric alignment, to correct lens distortions, a photometric alignment, to correct brightness and color mismatch and a composite view synthesis.

Videgames and simulators for data collection

The use of synthetic data has recently gained considerable importance in the computer vision community for several reasons. First, modern open-world games exhibit constantly increasing realism - which does not only mean that they feature photorealistic lights/textures etc, but also show plausible game dynamics and lifelike autonomous entity AI [150, 153] . Furthermore, most research fields in computer vision are now tackled by means of deep networks, which are notoriously data hungry in order to be properly trained. Particularly in the context of assisted and autonomous driving, the opportunity to exploit virtual yet realistic worlds for developing new techniques has been embraced widely: indeed, this makes possible to postpone the (very expensive) validation in real world to the moment in which a new algorithm already performs reasonably well in the simulated environment

[212, 56]. Building upon this tendency, [26] relies on TORCS simulator to learn an interpretable representation of the scene useful for the task of autonomous driving. However, while TORCS [212] is a powerful simulation tool, it's still severely limited by the fact that both its graphics and its game variety and dynamics are far from being realistic. CARLA [42] is another simulator designed to support development, training, and validation of autonomous driving systems; despite not being photorealistic, it is under active development and it is reasonable to expect that it will be even more widely used in the near future. In our work on learning the occupancy grid surrounding the vehicle (Sec. 4.3), we rely on GTAV videogame for automatically extracting annotated data. In different papers we instead relied on the ShapeNet [23] database to get individual 3D models which we then rendered using either Blender [11] or Open3D [233].

Object and vehicle novel view generation

In just few years, the widespread adoption of deep generative models [88, 64] has led to astounding results in different areas of image synthesis [144, 5, 226, 85, 199, 222]. In this scenario, conditional GANs [119] have been demonstrated to be a powerful tool to tackle image-to-image translation problems [77, 236, 237, 33]. Hallucinating novel views of the subject of a photo can be naturally framed as an image-to-image translation problem. For human subjects, this has been cast to predicting the person's appearance in different poses [113, 164, 229, 46]. Fashion and surveillance domains drew most of the attention, with much progress enabled by large real-world datasets providing multiple views of the same subject [107, 230].

For rigid objects instead, this task is usually referred to as *novel 3D view synthesis* and additional assumptions such as object symmetry are taken into account. In point of fact, symmetry is the most common assumption [234, 134, 238] to synthesise disoccluded portions of the object. Starting from a single image, Yang *et al.* [218] showed how a recurrent convolutional network can be trained via curriculum-learning to perform out-of-plane object rotation. In a similar setting Tatarchenko *et al.* [177] predicted both object appearance and depth map from different viewpoints. Successive works [234, 134] trained a network to learn a symmetry-aware appearance flow, re-casting the remaining synthesis as a task of image completion; [173] extends this approach to the case in which $N > 1$ input viewpoints are

available. However, all these works [218, 177, 234, 134, 173] assume the target view to be known at training time. As this is not usually the case in the real-world, these approaches are limited by the need to be trained solely on synthetic data and exhibit limited generalization in a real-world scenario. The recent work by Zhu *et al.* [238] exploits cycle consistency losses to overcome the need of paired data, thus training on datasets of segmented real-world cars and chairs they gathered for the purpose. Although that work shows more realistic results, it requires pixel-level segmentation for each class of interest. In contrast, we show that already available datasets for object 3D pose estimation [215, 214] can be used for this purpose, despite the extremely rough alignment between the annotated model and the image.

Non-parametric approaches to novel view synthesis

In the interactive editing setting, recent works [86, 147] have shown astounding results by keeping the human in the loop and assuming a perfect (even part-level) alignment between the 3D model and the input image. However, as pixels are warped from the input to the target view [147] it is not feasible to perform shape transfer to a completely different model. Moreover, the time required to synthesise the output is still far from real-time (few seconds). On the opposite, our semi-parametric framework presented in Sec. 4.4 enables disentangled shape and appearance transfer in real-time, with only a coarse alignment between the input image and the 3D model. In the scenario of image synthesis from semantic layout the recent work of Qi *et al.* [143] has shown that non-parametric components (i.e. a memory bank of image segments) can be integrated in a parametric image synthesis pipeline to produce impressive photo-realistic results. Despite our different setting, we similarly rely on image patches to provide hints to the Image Completion Network (ICN); however, our patches are not queried from a database but warped directly from the input view.

Chapter 3

Inside the Vehicle: Predicting the Driver's Focus of Attention

Autonomous and assisted driving are hot topics in computer vision these days. Despite the advent of autonomous cars, it's likely - at least in the near future - that human attention will still maintain a central role as a guarantee in terms of legal responsibility during the driving task. However, the driving task is extremely complex and a deep understanding of drivers' behavior is still lacking. Several researchers are now investigating the attention mechanism in order to define computational models for detecting salient and interesting objects in the scene. Nevertheless, most of these models only refer to bottom-up visual saliency and are focused on still images. Instead, during the driving experience the temporal nature and peculiarity of the task influence the attention mechanisms, leading to the conclusion that a deeper and more comprehensive study on real-life driving data is mandatory. In this chapter we describe our research effort for reducing this gap.

3.1 The DR(eye)VE Dataset

Autonomous and assisted driving have recently gained increasing momentum in the computer vision community. With the advent of deep learning, many tasks involving visual understanding –something which cannot be eluded in driving– have reached human-level performance, and sometimes overtaken it. Examples are pedestrian and vehicle detection and tracking, lanes or road signs recognition and, ultimately, semantic segmentation, where each pixel gets a label according to what it represents [30, 231]. All of these achievements are great examples of *subtasks* to autonomous and assisted driving, but we must not forget that the utmost goal is (better) driving itself. Do humans really need to detect all pedestrians or recognize all signs to drive? Do humans really need to label the whole scene?

In widely accepted psychological studies on the topic, the connection between driving, attention and gaze has been explored [179], negatively answering the above questions. It is known that humans' selective attention is a constraint required by the limited amount of resources available to our brain. Hence, it is still debatable if this approach may also bring benefits to visual computing models where the computational resources can be raised by adopting advanced performant hardware (e.g. GPUs, clusters). Nevertheless, the act of driving combines attention mechanisms influenced by the driver past experience, the temporal nature of the task and strong contextual constraints. As a result, we can drive much more safely and effectively than any automated system. One of the most relevant open questions in the field is to establish whether autonomous cars could benefit from attention-like mechanisms as well. Unluckily, this topic is under-investigated in computer vision and the lack of a realistic experimental framework does not help. In this section we describe our contribution of a new dataset available to the community, depicted in Fig. 3.1. We recorded more than six hours and 500,000 frames of driving sequences in different traffic and weather conditions. For every frame, we also acquired the driver gaze through an accurate eye tracking device. Additionally, to favor the car point of view, we projected gaze information on a HD quality video recorded from a roof-mounted camera. Given the subjective nature of both attention and driving, experimental design has played a crucial role in preparing the dataset and rule out spurious correlation between driver, weather, traffic, daytime and scenario.

Figure 3.1: An exemplar frame from the `DR(eye)VE` dataset. From left to right, from up to bottom: car-mounted view, driver's point of view, gaze map overlay and geo-referenced course.

At a computational level, human attention and eye fixation are typically modeled through the concept of *visual saliency*. Most of the literature on visual saliency focuses on filtering, selecting and synthesizing task dependent features for automatic object recognition. Nevertheless, the majority of experiments are constructed in controlled environments (*e.g.* laboratory settings) and on sequences of unrelated images [191, 21, 80]. Instead, our dataset has been collected "on the road" and it exhibits the following features:

- *It is public and open.* It provides hours of driving videos that can be used for understanding the attention phenomena;
- *It is task and context dependent.* According to the psychological studies on attention, data are collected during a real driving experience thus being as much realistic as possible;
- *It is precise and scientifically solid.* We use high end attention

recognition instruments, in conjunction with camera data and GPS information.

We believe that our proposal can be useful in several contexts aimed at understanding the driving phenomenon. It can be applied to identify and collect new features tailored for the driving experience (by analogy with what recently studied for video action recognition [117]). It can help understanding the influence of motion and semantics in salient object detection [169, 200]. It can foster the creation of new driver-centric visual ontologies, and as well serve the purpose to better understand how driver past experience affects the importance of objects in the scene.

The following is organized as follows. First, the acquisition apparatus and protocol is described in Sec. 3.1.1. In Sec. 3.1.2 we dive deep into the details about dataset design, video-gaze registration, computation of fixation maps and annotation provided. We present the attentional patterns emerging from the data analysis and exploration of the `DR(eye)VE` dataset in Sec. 3.1.3. Eventually, the section is concluded with a discussion on the possible uses of the collected data.

3.1.1 Apparatus and acquisition protocol

The driver's gaze information was captured using the commercial *SMI ETG 2w* Eye Tracking Glasses (ETG). ETG capture attention dynamics also in presence of head pose changes, which occur very often during the task of driving. While a frontal camera acquires the scene at 720p/30fps, users pupils are tracked at 60Hz. Gaze information are provided in terms of eye fixations, saccade movements, blinks and pupil dilation. In order to ensure the highest possible gaze quality, manual 3-points calibration is performed before each recorded sequence to adapt to small changes in the ETG device position.
Simultaneously, videos from the car perspective were acquired using the *GARMIN VirbX* camera mounted on the car roof (RMC, Roof-Mounted Camera). Such sensor captures frames at 1080p/25fps, and includes further information such as GPS data, accelerometer and gyroscope measurements. Figure 3.2 illustrates the aforementioned acquisition rig. During the acquisition phase, the two cameras are started simultaneously and the resulting videos are manually aligned to the frame in an offline stage to achieve the

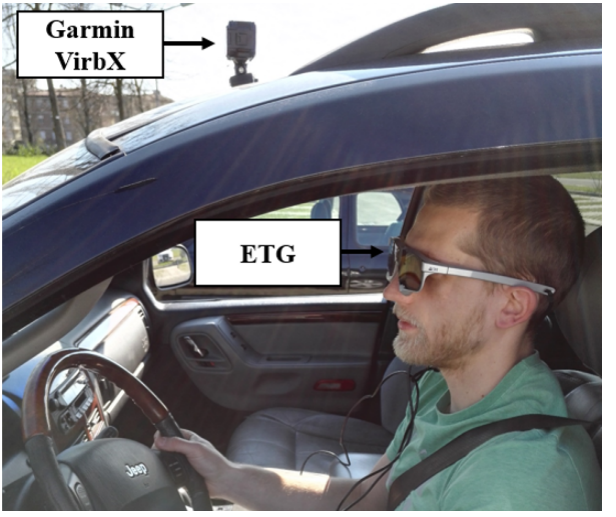


Figure 3.2: The acquisition rig for the DR(eye)VE dataset, featuring the head-mounted ETG and the car-mounted camera.

best possible synchronization. In order to re-project the gaze point on the video acquired by the car-mounted camera, local keypoints correspondences are exploited, resulting in the gaze information being present on both video sequences (see Section 3.1.2).

3.1.2 Dataset description and annotation

In this section we present the DR(eye)VE dataset (Fig. 3.16), the protocol adopted for video registration and annotation, the automatic processing

Table 3.1: Summary of the DR(eye)VE dataset characteristics. The dataset was designed to embody the most possible diversity in the combination of different features.

Videos	Frames	Drivers	Weathers	Lighting	Gaze Info	Metadata	Viewpoint
74	555,000	8	Sunny Cloudy Rainy	Day Evening Night	Raw fixations Gaze map Pupil dilation	GPS Speed Course	Driver (720p) Car (1080p)

Table 3.2: A comparison between DR(eye)VE and other datasets.

Dataset	Frames	Drivers	Scenarios	Annotation	Real-world	Public
Pugeault *et al.* [142]	158,668	–	Countryside Highway Downtown	Gaze Maps Driver's Actions	Yes	No
Simon *et al.* [166]	40	30	Downtown	Gaze Maps	No	No
Underwood *et al.* [191]	120	77	Motorway	–	No	No
Fridman *et al.* [52]	1,860,761	50	Highway	6 Gaze classes	Yes	No
DR(eye)VE [4]	555,000	8	Countryside Highway Downtown	Gaze Maps GPS, Speed, Course	Yes	Yes

of eye-tracker data and the analysis of the driver's behavior in different conditions.

Dataset design

The DR(eye)VE dataset[1] consists of 555,000 frames divided in 74 sequences, each of which is 5 minutes long. Eight different drivers of varying age from 20 to 40, including 7 men and a woman, took part to the driving experiment, that lasted more than two months. Experimental design played a crucial role in preparing the dataset to rule out spurious correlation between driver, weather, traffic, daytime and scenario. Per-sequence details are reported in Tables 3.3 and 3.4.
To cover the widest range of scenarios as possible, videos were recorded in different contexts, both in terms of landscape (downtown, countryside, highway) and traffic condition, ranging from traffic-free to highly cluttered scenarios. They were also recorded in diverse weather conditions (sunny, rainy, cloudy) and at different hours of the day (both daytime and night). Tab. 3.1 recaps the dataset features and Tab. 3.2 compares it with other related proposals. DR(eye)VE is currently the largest publicly available dataset including gaze and driving behavior in automotive settings.

Video-gaze registration

The dataset has been processed to move the acquired gaze from the egocentric (ETG) view to the car (RMC) view. In fact, the latter features a much wider field of view (FoV), and can contain fixations that are out

[1] http://imagelab.ing.unimore.it/dreyeve

of the egocentric view. For instance, this can occur whenever the driver takes a peek at something at the border of this FoV, but doesn't move his head. For every sequence, the two videos were manually aligned to cope with the difference in sensors framerate. Videos were then registered frame-by-frame through a homographic transformation that projects fixation points across views. More formally, at each timestep t the RMC frame I^t_{RMC} and the ETG frame I^t_{ETG} are registered by means of a homography matrix $H_{ETG \rightarrow RMC}$, computed by matching SIFT descriptors [110] from one view to the other (see Fig. 3.3). A further pass of Random Sample Consensus (RANSAC) [50] procedure ensures robustness to outliers. While homographic mapping is theoretically sound only across planar views - which is not the case of outdoor environments - we empirically found that projecting an object from one image to another always recovered the correct position. This makes sense if the distance between the projected object and the camera is far greater than the distance between the object and the projective plane. In Sec. 9 of the supplementary material, we derive formal bounds to explain this phenomena.

Fixation map computation.

The pipeline discussed above provides a frame-level annotation of the driver's fixations. In contrast to image saliency experiments [21], there is no clear and indisputable protocol for obtaining continuous maps from raw fixations when acquired in task-driven real-life scenarios. This is even more evident when fixations are collected in task-driven real-life scenarios. The

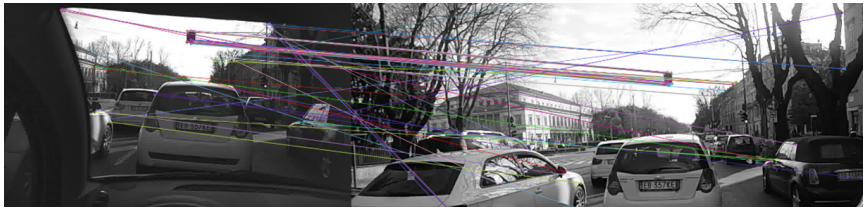


Figure 3.3: Registration between the egocentric and roof-mounted camera views by means of SIFT descriptor matching.

main motivation resides in the fact that observer's subjectivity cannot be removed by averaging different observers' fixations. Indeed two different observers cannot experience the same scene at the same time (*e.g.* two drivers cannot be at the same time in the same point of the street). The only chance to average among different observers would be the adoption of a simulation environment, but it has been proved that the cognitive load in controlled experiments is lower than in real test scenarios and it effects the true attention mechanism of the observer [155]. In our preliminary `DR(eye)VE` release [4], fixation points were aggregated and smoothed by means of a temporal sliding window. In such a way, temporal filtering discarded momentary glimpses that contain precious information about the driver's attention. Following the psychological protocol in [115] and [66], this limitation was overcome in the current release where the new fixation maps were computed without temporal smoothing. Both [115] and [66] highlight the high degree of subjectivity of scene scanpaths in short temporal windows (< 1 sec) and suggest to neglect the fixations pop-out order within such windows. This mechanism also ameliorates the *inhibition of return* phenomenon that may prevent interesting objects to be observed twice in short temporal intervals [141, 71], leading to the underestimation of their importance.

More formally, the *fixation map* F_t for a frame at time t is built by accumulating projected gaze points in a temporal sliding window of $k = 25$ frames, centered in t. For each time step $t + i$ in the window, where $i \in \{-\frac{k}{2}, -\frac{k}{2} + 1, \ldots, \frac{k}{2} - 1, \frac{k}{2}\}$, gaze points projections on F_t are estimated through the homography transformation H^t_{t+i} that projects points from the image plane at frame $t + i$, namely p_{t+i}, to the image plane in F_t. A continuous fixation map is obtained from the projected fixations by centering on each of them a multivariate Gaussian having a diagonal covariance matrix Σ (the spatial variance of each variable is set to $\sigma_s^2 = 200$ pixels) and taking the *max* value along the time axis:

$$F_t(x, y) = \max_{i \in (-\frac{k}{2}, \ldots, \frac{k}{2})} \mathcal{N}((x, y) \mid H^t_{t+i} \cdot p_{t+i}, \Sigma) \tag{3.1}$$

The Gaussian variance has been computed by averaging the ETG spatial acquisition errors on 20 observers looking at calibration patterns at different distances from 5 to 15 meters. The described process can be appreciated in Fig. 3.4. Eventually, each map F_t is normalized to sum to 1, so that it can be considered a probability distribution of fixation points.

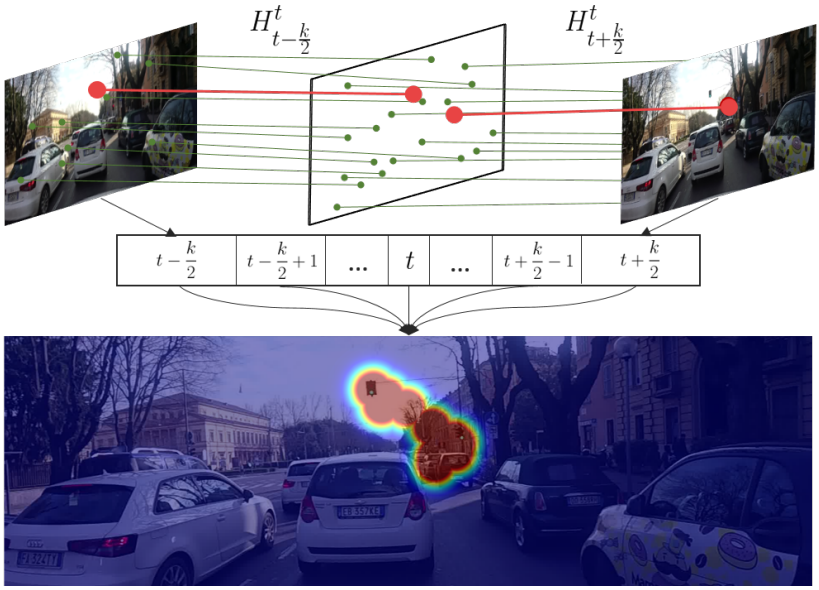
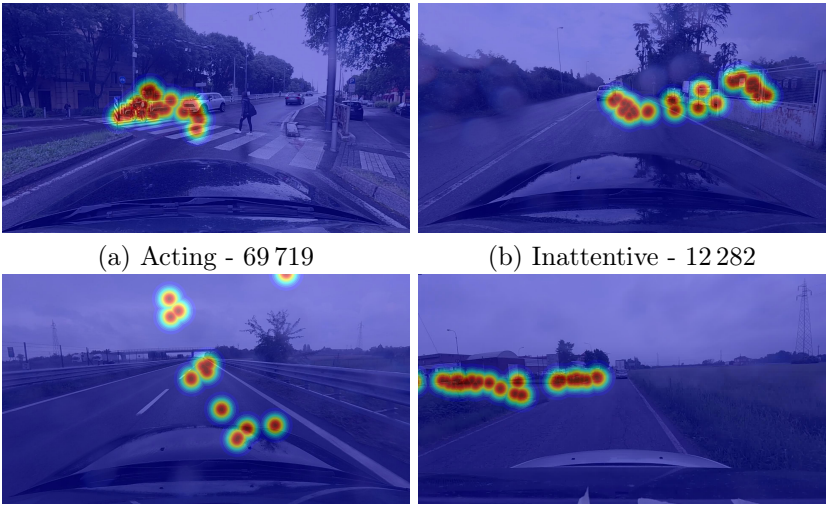


Figure 3.4: Resulting fixation map from a 1 second integration (25 frames). The adoption of the *max* aggregation of equation 3.1 allows to account in the final map two brief glances towards traffic lights.

Labeling attention drifts

Fixation maps exhibit a very strong central bias. This is common in saliency annotations [178] and even more in the context of driving. For these reasons, there is a strong unbalance between lots of easy-to-predict scenarios and unfrequent but interesting hard-to-predict events.

To enable the evaluation of computational models under such circumstances, the `DR(eye)VE` dataset has been extended with a set of further annotations. For each video, subsequences whose ground truth poorly correlates with the average ground truth of that sequence are selected. We employ Pearson's Correlation Coefficient (CC) and select subsequences with CC < 0.3. This happens when the attention of the driver focuses far from the vanishing point of the road. Examples of such subsequences are depicted in Fig. 3.5. Several human annotators inspected the selected



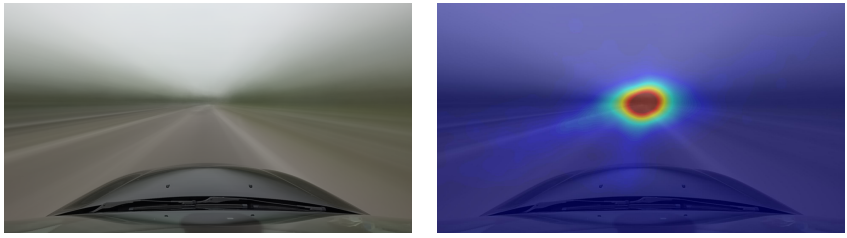
(a) Acting - 69 719 (b) Inattentive - 12 282

(c) Error - 22 893 (d) Subjective - 3 166

Figure 3.5: Examples of the categorization of frames where gaze is far from the mean. Overall, 108 060 frames (∼20% of `DR(eye)VE`) were extended with this type of information.

frames and manually split them into (a) acting, (b) inattentive, (c) errors and (d) subjective events:

- *errors* can happen either due to failures in the measuring tool (*e.g.* in extreme lighting conditions) or in the successive data processing phase (*e.g.* SIFT matching);
- *inattentive* subsequences occur when the driver focuses his gaze on objects unrelated to the driving task (*e.g.* looking at an advertisement);
- *subjective* subsequences describe situations in which the attention is closely related to the individual experience of the driver, *e.g.* a road sign on the side might be an interesting element to focus for someone that has never been on that road before but might be safely ignored by someone who drives that road every day.



(a) (b)

Figure 3.6: Mean frame (a) and fixation map (b) averaged across the whole sequence 02, highlighting the link between driver's focus and the vanishing point of the road.

- *acting* subsequences include all the remaining ones.

Acting subsequences are particularly interesting as the deviation of driver's attention from the common central pattern denotes an intention linked to task-specific actions (*e.g.* turning, changing lanes, overtaking ...). For these reasons, subsequences of this kind will have a central role in the evaluation of predictive models in Sec. 3.3.3.

3.1.3 Dataset exploration and analysis

Here we present several insights about *where* and *what* the driver is looking at while driving. We analyze what people pay attention to while driving, and which part of the scene around the vehicle is more critical for the task. In particular, we dive deep into the influence of car speed, course and the landscape over the driver's attentional behavior. The indication of which elements in the scene are likely to capture the driver's attention may benefit several applications in the context of human-vehicle interaction and driver attention analysis.

Attraction towards the vanishing point

By analyzing the dataset frames, the very first insight is the presence of a strong attraction of driver's focus towards the vanishing point of the road, that can be appreciated in Fig. 3.6. The same phenomenon was observed in previous studies [190, 13] in the context of visual search tasks. We observed

indeed that drivers often tend to disregard road signals, cars coming from the opposite direction and pedestrians on sidewalks.
This is an effect of human peripheral vision [157], that allows observers to still perceive and interpret stimuli out of - but sufficiently close to - their focus of attention (FoA). A driver can therefore achieve a larger area of attention by focusing on the road's vanishing point: due to the geometry of the road environment, many of the objects worth of attention are coming from there and have already been perceived when distant.

Influence of speed on driver's gaze

Moreover, the gaze location tends to drift from this central attractor when the context changes in terms of car speed and landscape. Indeed [151] suggests that our brain is able to compensate spatially or temporally dense information by reducing the visual field size. In particular, as the car travels at higher speed the temporal density of information (*i.e.* the amount of information that the driver needs to elaborate per unit of time) increases: this causes the useful visual field of the driver to shrink [151]. We also observe this phenomenon in our experiments, as shown in Fig. 3.7.

Car trajectory distribution

Leveraging course and speed information which come with the `DR(eye)VE` dataset, we are able to compute basic statistics about the distribution of car trajectory across different landscape scenarios. Unsurprisingly, results show that the vehicle goes straight for most of the time - up to 98% of time during highway runs. Along with the aforementioned attraction towards the vanishing point, this skewed distribution contributes in creating a strong attentional bias towards the center of the image.

Influence of semantic categories

`DR(eye)VE` data also highlight that the driver's gaze is attracted towards specific semantic categories. To reach the above conclusion, the dataset is analysed by means of the semantic segmentation model in [221] and the distribution of semantic classes within the fixation map evaluated.
More precisely, given a segmented frame and the corresponding fixation map, the probability for each semantic class to fall within the area of attention is computed as follows: First, the fixation map (which is continuous in $[0, 1]$)

is normalized such that the maximum value equals 1. Then, nine binary maps are constructed by thresholding such continuous values linearly in the interval $[0, 1]$. As the threshold moves towards 1 (the maximum value), the area of interest shrinks around the real fixation points (since the continuous map is modeled by means of several Gaussians centered in fixation points, see previous section). For every threshold, a histogram over semantic labels within the area of interest is built, by summing up occurrences collected from all `DR(eye)VE` frames. Fig. 3.9 displays the result: for each class, the probability of a pixel to fall within the region of interest is reported for each threshold value. The figure provides insight about which categories represent the real focus of attention and which ones tend to fall inside the attention region just by proximity with the formers. Object classes that exhibit a positive trend, such as road, vehicles and people, are the real focus of the gaze, since the ratio of pixels classified accordingly increases when the observed area shrinks around the fixation point. In a broader sense, the figure suggests that despite while driving our focus is dominated by road and vehicles, we often observe specific objects categories even if they contain little information useful to drive.

3.1.4 Discussion and open questions

In this section we pose a few challenges and opportunities unlocked by the availability of the proposed dataset to the computer vision community. According to a qualitative analysis, it appears that when using an image based saliency prediction method (*e.g.* [227], which achieves state of the art performance on [21]), the regions of interest heavily rely on visual discontinuities resulting in fairly different attention maps with respect to the driver actual intentions, Figure 3.8 fourth and fifth columns. While this difference has not yet been quantitatively studied, it raises a set of open questions that we believe of interest for the computer vision community. Investigating the following topics (and possibly achieving positive answers) may consequently help pushing forward the field of assisted and autonomous driving.

Can driver's gaze be predicted?

Despite a large body of psychological literature, the computer vision community has not yet seen effective computational models able to predict

human gaze while driving. In particular, the temporal nature of the driving task has never been considered. In point of fact, we qualitatively observed that during red traffic lights and jams, visual saliency models trained on images could predict driver gaze quite accurately, [91, 21, 227]. Nevertheless, as driving speed increases, the amount of attention drivers dispose of at each instant decreases, resulting into very sparse and specific attention regions. Future models will need to take into account this behavior to provide reliable accuracy. Moreover, how easier is this task going to be if we were to feed the driver intentions (*e.g.* turn right in 10s) to the model?

Can driver's intentions be anticipated from gaze data?

Here we pose the opposite challenge to gaze prediction, that is whether we can build models that given video data and related gaze (true or predicted) are able to estimate the driver next move. These estimates can include the car turning angle, instantaneous speed, breaking events and so on. On top of this, the community may build systems able to exploit intentions prediction to alert the driver in dangerous situations.

Can gaze models be employed to enhance signalization and road safety?

While driving we only observe a small part of all the road signs, cars and traffic lights. In most of the cases, this is due to drivers' confidence about the path taken or irrelevant signalization with respect to driver current intentions. At the same time, overconfidence during driving may result in mistakes whenever signals change leading to possible dangerous situations. Local administrations can take advantage from gaze models to better decide how to place road signals and traffic lights. This is not a completely new line of work [166, 19], however the availability of a public dataset can serve as a unified benchmark for the research community.

Can gaze models help autonomous cars in planning better driving strategies?

Autonomous cars leverage on many different levels of structured information, ranging from lanes detection to semantic segmentation. Nevertheless, autonomous driving is ultimately a decision task. Can gaze information be yet another level of information to input to this decision process? Can

human-like attention bring benefits to human-less vehicles? This is probably a far reaching question and we fully expect better experimental frameworks to be conceived in the future in order to answer it. Meanwhile, we make available the first dataset for the community to download and start tackling this challenge.

3.1.5 Concluding remarks

We have proposed a novel dataset that addresses the lack of public benchmarks concerning drivers' attention in real-world scenarios. It comes with pre-computed gaze maps and contextual information such as the car's speed and course. The dataset is freely available for academic research, along with the code used in the creation of gaze maps and annotation.
However, collecting the data is only the first step in our research project. In the following sections we will describe how the data from the `DR(eye)VE` dataset have been exploited to create a computational model of human attention during the driving task.

Sequence	Daytime	Weather	Landscape	Driver	Set
01	Evening	Sunny	Countryside	D8	Train Set
02	Morning	Cloudy	Highway	D2	Train Set
03	Evening	Sunny	Highway	D3	Train Set
04	Night	Sunny	Downtown	D2	Train Set
05	Morning	Cloudy	Countryside	D7	Train Set
06	Morning	Sunny	Downtown	D7	Train Set
07	Evening	Rainy	Downtown	D3	Train Set
08	Evening	Sunny	Countryside	D1	Train Set
09	Night	Sunny	Highway	D1	Train Set
10	Evening	Rainy	Downtown	D2	Train Set
11	Evening	Cloudy	Downtown	D5	Train Set
12	Evening	Rainy	Downtown	D1	Train Set
13	Night	Rainy	Downtown	D4	Train Set
14	Morning	Rainy	Highway	D6	Train Set
15	Evening	Sunny	Countryside	D5	Train Set
16	Night	Cloudy	Downtown	D7	Train Set
17	Evening	Rainy	Countryside	D4	Train Set
18	Night	Sunny	Downtown	D1	Train Set
19	Night	Sunny	Downtown	D6	Train Set
20	Evening	Sunny	Countryside	D2	Train Set
21	Night	Cloudy	Countryside	D3	Train Set
22	Morning	Rainy	Countryside	D7	Train Set
23	Morning	Sunny	Countryside	D5	Train Set
24	Night	Rainy	Countryside	D6	Train Set
25	Morning	Sunny	Highway	D4	Train Set
26	Morning	Rainy	Downtown	D5	Train Set
27	Evening	Rainy	Downtown	D6	Train Set
28	Night	Cloudy	Highway	D5	Train Set
29	Night	Cloudy	Countryside	D8	Train Set
30	Evening	Cloudy	Highway	D7	Train Set
31	Morning	Rainy	Highway	D8	Train Set
32	Morning	Rainy	Highway	D1	Train Set
33	Evening	Cloudy	Highway	D4	Train Set
34	Morning	Sunny	Countryside	D3	Train Set
35	Morning	Cloudy	Downtown	D3	Train Set
36	Evening	Cloudy	Countryside	D1	Train Set
37	Morning	Rainy	Highway	D8	Train Set

Table 3.3: `DR(eye)VE` train set: details for each sequence

Sequence	Daytime	Weather	Landscape	Driver	Set
38	Night	Sunny	Downtown	D8	Test Set
39	Night	Rainy	Downtown	D4	Test Set
40	Morning	Sunny	Downtown	D1	Test Set
41	Night	Sunny	Highway	D1	Test Set
42	Evening	Cloudy	Highway	D1	Test Set
43	Night	Cloudy	Countryside	D2	Test Set
44	Morning	Rainy	Countryside	D1	Test Set
45	Evening	Sunny	Countryside	D4	Test Set
46	Evening	Rainy	Countryside	D5	Test Set
47	Morning	Rainy	Downtown	D7	Test Set
48	Morning	Cloudy	Countryside	D8	Test Set
49	Morning	Cloudy	Highway	D3	Test Set
50	Morning	Rainy	Highway	D2	Test Set
51	Night	Sunny	Downtown	D3	Test Set
52	Evening	Sunny	Highway	D7	Test Set
53	Evening	Cloudy	Downtown	D7	Test Set
54	Night	Cloudy	Highway	D8	Test Set
55	Morning	Sunny	Countryside	D6	Test Set
56	Night	Rainy	Countryside	D6	Test Set
57	Evening	Sunny	Highway	D5	Test Set
58	Night	Cloudy	Downtown	D4	Test Set
59	Morning	Cloudy	Highway	D7	Test Set
60	Morning	Cloudy	Downtown	D5	Test Set
61	Night	Sunny	Downtown	D5	Test Set
62	Night	Cloudy	Countryside	D6	Test Set
63	Morning	Rainy	Countryside	D8	Test Set
64	Evening	Cloudy	Downtown	D8	Test Set
65	Morning	Sunny	Downtown	D2	Test Set
66	Evening	Sunny	Highway	D6	Test Set
67	Evening	Cloudy	Countryside	D3	Test Set
68	Morning	Cloudy	Countryside	D4	Test Set
69	Evening	Rainy	Highway	D2	Test Set
70	Morning	Rainy	Downtown	D3	Test Set
71	Night	Cloudy	Highway	D6	Test Set
72	Evening	Cloudy	Downtown	D2	Test Set
73	Night	Sunny	Countryside	D7	Test Set
74	Morning	Rainy	Downtown	D4	Test Set

Table 3.4: `DR(eye)VE` test set: details for each sequence

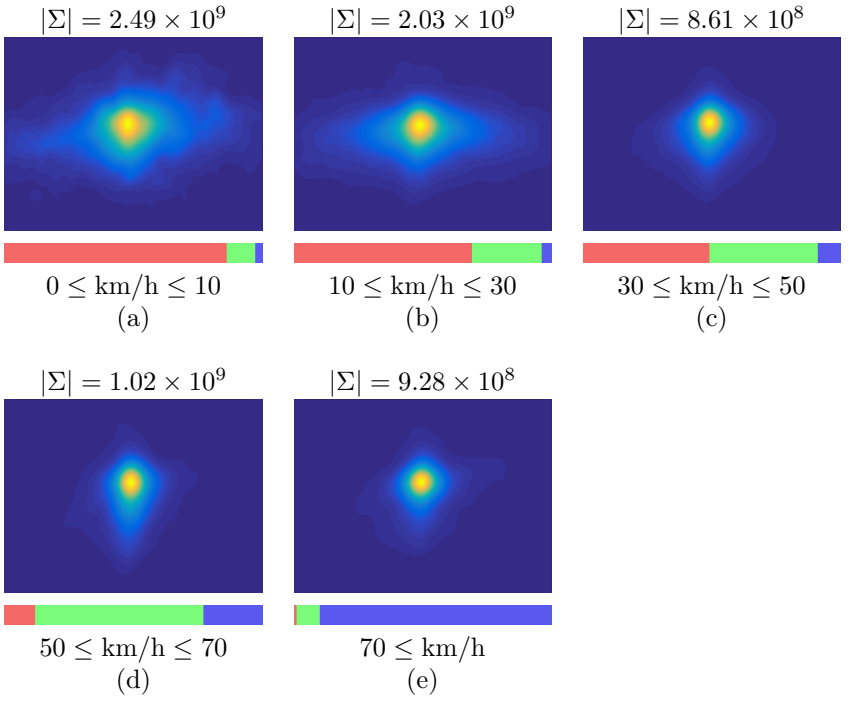


Figure 3.7: As speed gradually increases, driver's attention converges towards the vanishing point of the road. (a) When the car is approximately stationary, the driver is distracted by many objects in the scene. (b-e) As the speed increases, the driver's gaze deviates less and less from the vanishing point of the road. To measure this effect quantitatively, a two-dimensional Gaussian is fitted to approximate the mean map for each speed range, and the determinant of the covariance matrix Σ is reported as an indication of its spread (the determinant equals the product of eigenvalues, each of which measures the spread along a different data dimension). The bar plots illustrate the amount of downtown (red), countryside (green) and highway (blue) frames that concurred to generate the average gaze position for a specific speed range. Best viewed on screen.

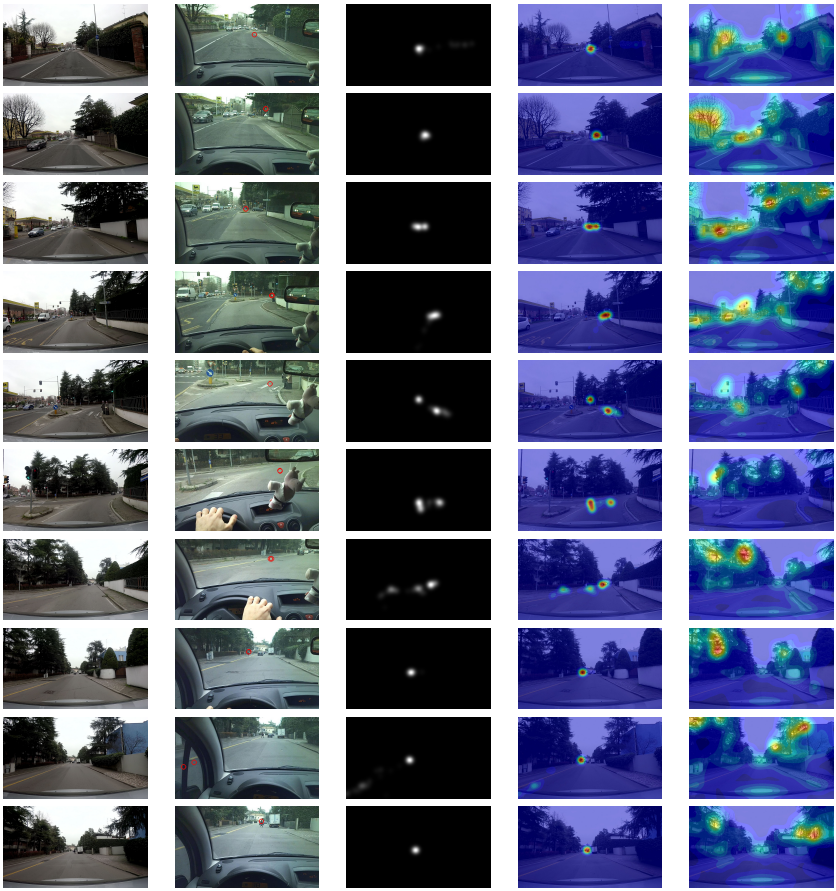


Figure 3.8: Example sequence taken from the DR(eye)VE dataset. Left to right: Garmin VirbX frames, ETG frames with fixation information, the available gaze map, overlay between the frame and the gaze map, visual saliency predicted using [227]. Up to bottom: the temporal dimension of the video with 1 frame every 30 displayed.

	Still	Going straight	Curving
Countryside	03%	92%	05%
Downtown	25%	69%	06%
Highway	01%	98%	01%

Table 3.5: Distribution of car trajectory across different landscape scenarios.

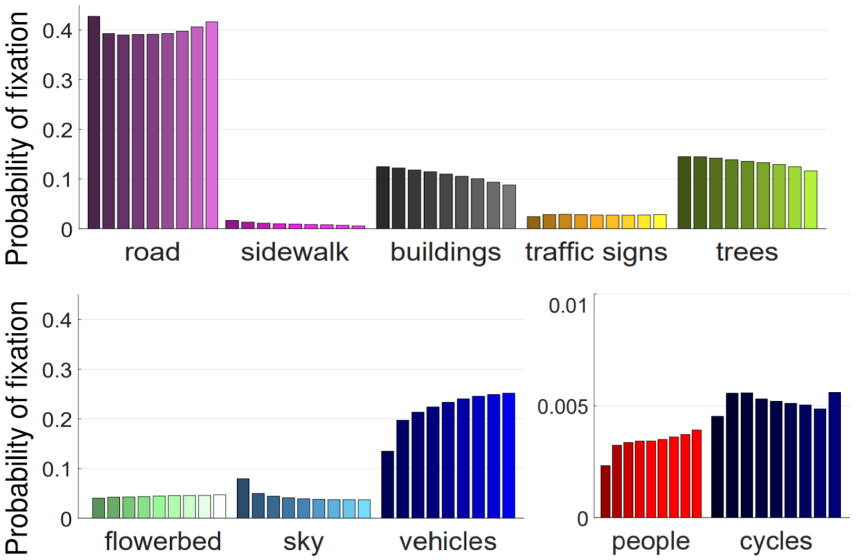


Figure 3.9: Proportion of semantic categories that fall within the driver's fixation map when thresholded at increasing values (from left to right). Categories exhibiting a positive trend (*e.g.* road and vehicles) suggest a real attention focus, while a negative trend advocates for an awareness of the object which is only circumstantial. See Sec. 3.1.3 for details.

3.2 Learning Where to Attend Like a Human Driver: a Deep Learning Model of Driver's Attention

In this section we engineer and design a computational model of human attention during the driving task. In particular, here we train a coarse-to-fine convolutional network on sequences from the DR(eye)VE dataset. Experiments against baselines and state of the art competitors show that this model already provides favourable results. The model presented here will constitute the backbone of the more powerful model designed in Sec. 3.3, in which we will explicitly integrate in our model the most influencing factors for the driver's attentional behavior (i.e. motion and scene semantics).

3.2.1 Motivation

While autonomous driving is quickly reaching maturity, it's not clear how far in time society will overlook the legal responsibility of the human driver [126]. Conversely, Advanced Driver Assistance Systems (ADAS) are human-centric and already established both in literature and in the market. The ultimate aim of these systems is to increase the safety of the driver and the road environment at large; this is usually done through collision avoidance systems, blind spot control, lane change assistance, traffic signs recognition and many others. Some of the most ambitious examples of assisted driving are related to driver monitoring systems [79, 54, 90, 121], where the attentional behavior of the driver is parsed together with the road scene to predict potentially unsafe manoeuvres and act on the car in order to avoid them (either by signaling the driver or braking). However, all these approaches are limited by their ability to capture the true attentional and intentional behavior of the driver, which is still a complex and largely unsolved task today. Conversely, we advocate for a new assisted driving paradigm which suggests to the driver, with no hard intervention, where he should focus his attention. The problem is thus shifted from a personal level (*what the driver is looking at*) to a task-driven level (*what the driver should be looking at*). Following the notion that gaze is a primary cue to human visual attention, here we tackle the challenge of building a deep network architecture to model human attention while driving (see Fig. 3.10, and evaluate the ability of the proposed approach to replicate what we

(a) roof-mounted camera (b) network prediction

Figure 3.10: Starting from raw frames and attentional maps provided with the DR(eye)VE dataset, our goal is to replicate the driver's attentional through a deep network model. In (b) we show the attentional map predicted by our model for frame (a).

observed in humans (Sec. 3.2.2-3.2.3).
We train our model on the DR(eye)VE data [4], under the hypothesis that i) individuals were thoughtfully driving and ii) distractions have no predefined pattern and can thus be considered outliers. Given that the provided gaze annotation is reliable, we aim to generalize from *what the driver was looking at* in annotated sequences to *what the driver should be looking at* in unseen scenarios.

3.2.2 A deep network model of driver's attention

Exploting DR(eye)VE data

In this section we investigate the attentional mechanisms involved in driving. To this end, we rely on the recently proposed DR(eye)VE [4] dataset, which is a collection of 74 sequences 5 minutes long, featuring different landscapes, weather scenarios, lighting conditions and 8 drivers (details in Sec. 3.1). Every sequence is composed of two videos, capturing both the driver's and the car point of view. While the driver's gaze is synchronized with the former video, ground truth attentional maps are provided on the latter through a homographic projection followed by spatial smoothing and time integration. These latter post-processing steps attenuate subjectivity of the drivers' scan-path and reduce measurement inaccuracies in the ground truth. More formally, we can define an attentional map as a 2D grid where

Figure 3.11: `COARSE` prediction architecture. The first part of the network performs the feature encoding. The input videoclip is a tensor of size $3 \times 16 \times 112 \times 112$ that undergoes a sequence of conv3D and pool3D layers that gradually squeeze it to size $512 \times 1 \times 7 \times 7$. All conv3D have kernel size (3,3,3) and ReLU activation units; all pool3D have pool size (2,2,2) except the first one that has pool size (1,2,2). In order to obtain a saliency map with the same spatial size of the input frame, the feature representation is decoded through a series of intertwined layers of conv2D and $\times 2$ upsampling on the spatial dimensions. All conv2D have kernel size (3,3) and are followed by leaky ReLU activations with $\alpha = .001$. As a result, the output of the network is a tensor of size $1 \times 112 \times 112$, *i.e.* the predicted attentional map.

each cell represents the probability that the corresponding pixel of the roof-mounted camera image is within the driver's focus of attention. The `DR(eye)VE` dataset analysis and exploration (see Sec. 3.1) provided insights on the mechanisms that govern human attention while driving. Below, we learn a deep attentional model that, given a driving sequence, is able to focus where the human driver would.

Various recent works show [185, 84] that when dealing with videos, taking explicitly into account the temporal dimension of the input in the network architecture can lead to results that easily outclass the single-frame-input baselines in various high-level video analysis tasks such as video classification and action recognition among others. To this end, we can distinguish at least two main trends that emerged in the recent literature. Those who just want to exploit short-range dependencies in the data structure often make use of 3D convolutional architectures in which data of successive time steps are stacked along an additional dimension of the input tensor. Conversely, if the task requires to capture longer-term interactions recurrent architecture (*e.g.* LSTM, GRU) are often the clear winners.
Here, we make the assumption that a short video sequence (*e.g.* half a second) contains enough information to successfully predict where the driver should look in that situation. Indeed, it can be argued that humans take even less time to react to a stimulus. For this reason, we build our

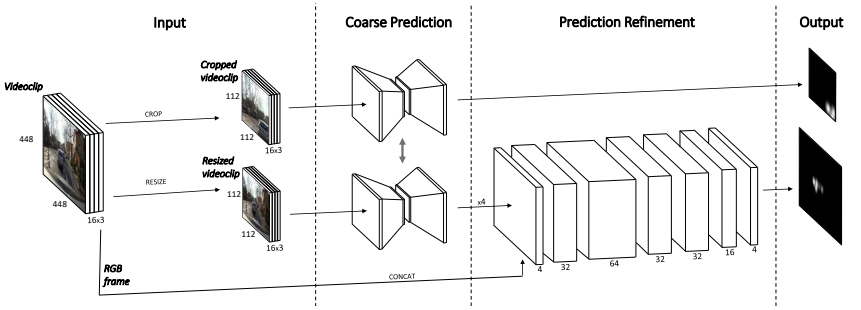


Figure 3.12: `COARSE+FINE` prediction architecture. The `COARSE` module (see Figure 3.11) is applied to both a cropped and a resized version of the input tensor, which is a videoclip of 16 consecutive frames. The cropped input is used during training to augment the data and the variety of ground truth attentional maps. The prediction of the resized input is stacked with the last RGB frame of the videoclip and fed to a series of convolutional layers (`FINE` prediction module) with the aim of refining the prediction. All convolutions have kernel size (3,3). Training is performed end-to-end and weights between `COARSE` modules are shared. At test time, only the refined predictions are used.

approach on 3D convolutions which take as input a fixed-size sequence of 16 consecutive frames from a video (called from now on *videoclip*) and outputs the gaze map for the last frame of the input clip.

Coarse gaze prediction module

The core of our deep network model is a fully convolutional network whose architecture is represented in Fig. 3.11. The first half of the network acts as an encoder and maps the input videoclip in feature space. Conversely, the second block decodes the feature representation in an attentional map which has the same width and height of the input videoclip, but singleton temporal dimension. In order to perform the encoding, we employ the C3D network by [185] with pre-trained weights and few minor modifications, such as dropping the last convolutional and the fully connected layers in order to maintain spatial information that would be otherwise discarded.

See Fig. 3.11 for further details. During training we resize the training images to 128×128 and then randomly crop them to 112×112, following the training process described in [185]. Due to the strong bias towards the vanishing point, this standard cropping policy turned out to be insufficient for creating a real variety in the location of the attentional maps; thus the network prediction resulted strongly attracted towards the center of the image.

Complete architecture

In order to solve the aforementioned problem, we moved to an extremely more aggressive crop policy. In this second attempt the training sequences were resized to 256×256 before cropping them to 112×112. This way the crop constituted less than a quarter of the original image, creating enough variety in the location of the ground truth. This posed however a new challenge: As the network was trained on small crops, it learned to predict better localized but significantly wider attentional maps, reflecting the proportion between salient and non-salient areas seen at training time. We thus enhanced the network architecture towards a more sophisticated multi-input multi-output approach, shown in Fig. 3.12. The network still takes a 16 frames videoclip as input, although now there are three different data streams. The first stream provides the model with a randomly cropped videoclip, as explained above. This videoclip passes through the encoding-decoding pipeline and produces a coarse prediction: A first loss on this output is employed to force the model to learn variety in the prediction position and avoid the trivial *hit-the-center* solution. The second stream feeds the model with the same uncropped videoclip, but resized to match the input shape. This videoclip also goes through the encoding-decoding stack, producing a new coarse uncropped saliency prediction. The prediction is then upsampled and stacked on the RGB image of the last frame of the videoclip, which is provided as third input. Eventually, the concatenated tensor is passed through a last block of convolutions, with the purpose of refining the prediction. This last step also exploits the appearance information of the original videoclip. On the output of this block - our final attentional map prediction - we compute the second loss.
In the following experimental sections we evaluate both quantitatively and qualitatively the proposed model. In Sec. 3.2.3 we rely on saliency metrics to measure the network's performance against several baselines and state-

Table 3.6: Training and test results obtained by both the baselines and the proposed networks. See text for details.

	Test sequences		CC(GT, MEAN) < 0.3	
	CC	KL	CC	KL
Baseline (gaussian)	0.33 ± 0.13	2.50 ± 0.63	0.22 ± 0.15	2.70 ± 0.74
Baseline (mean train GT)	0.48 ± 0.27	1.65 ± 1.17	0.17 ± 0.20	2.85 ± 1.31
`COARSE`	0.44 ± 0.23	1.73 ± 1.00	0.19 ± 0.18	2.74 ± 1.07
`COARSE+FINE`	0.55 ± 0.28	1.42 ± 1.07	0.30 ± 0.24	2.24 ± 1.10

of-the-art video saliency methods; while in Sec. 3.2.3 we analyze how well the model mimics the driver's focus dynamics.

3.2.3 Experimental evalutation

The performance of the proposed `COARSE` and `COARSE+FINE` models are quantitatively measured against different baselines. Following the guidelines in [22], for the evaluation phase we rely on Pearson's Correlation Coefficient (CC) and Kullback–Leibler divergence (KL) measures.

Training details. The encoding half of the `COARSE` network is initialized with pre-trained weights [185]. Training sequences are randomly mirrored to augment the data. End-to-end training minimizes the Mean Squared Error for both losses of the `COARSE+FINE` model; we employ Adam optimizer with parameters suggested in the original paper [87]. Train and test are split according to [4] and 500 central frames from each training sequence are used for validation.

Keeping it simple: Baselines from saliency

It is widely known that a centered Gaussian, stretched to the aspect ratio of the image, makes for an incredibly effective baseline for the visual saliency task. This static baseline scores better than many methods benchmarked on the `MIT300` [21] dataset. Section WHERE revealed that a similar bias affects the `DR(eye)VE` dataset (see Fig. 3.6). Thus, to validate the proposed model, we compare it against both the aforementioned baseline and a more task-driven version of it built as the average of all training set attentional maps, Fig. 3.13(c) and (d). Results are reported in Tab. 3.6: The second

Table 3.7: Comparison with video saliency state-of-the-art methods.

	CC	KL
Wang *et al.* [201]	0.08 ± 0.11	3.77 ± 0.77
Wang *et al.* [200]	0.03 ± 0.09	4.24 ± 1.13
Mathe *et al.* [117]	0.04 ± 0.08	3.92 ± 0.53

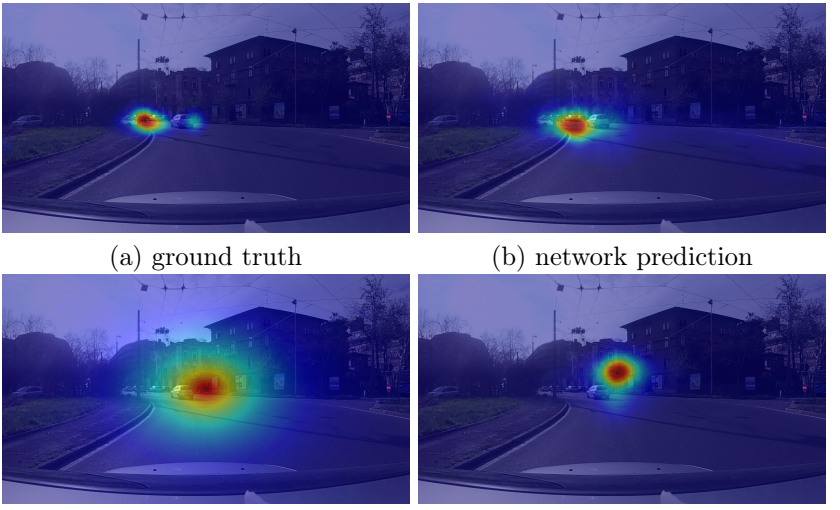
baseline performs better than both the Gaussian baseline and the `COARSE` model on the test set, but significantly worse than `COARSE+FINE` model. Indeed the latter has a relative gain of about +25% over the `COARSE` model and about +15% over the average training ground truth.

Comparison with state-of-the-art

In Tab. 3.7 we report results from two recent unsupervised video saliency methods [200, 201] and a supervised one [117] on the test set. Both unsupervised methods rely on appearance and motion discontinuities and are easily fooled by the motion of the roof-mounted camera. Unfortunately, [117] is trained on the *Action in The Eye* dataset and thus performs poorly on this domain. As far as we know, no supervised video saliency method releases the source code allowing us to re-train it, nor reports performance on `DR(eye)VE`. Results shown in Tab. 3.7 call for supervised methods aware of both the semantic of the scene and the peculiarities of the task. To our knowledge, our proposal is the first deep model for driving attention, and video saliency to a greater extent.

New annotations to escape the bias

Despite showing good results, the baselines introduced in Sec. 3.2.3 are of no interest for the driving task as they are not able to generalize when required. There is a strong unbalance between lots of trivial-to-predict scenarios of little interest and few but important hard-to-predict events. To enable the evaluation of our model under such circumstances, we select from the `DR(eye)VE` dataset those sub-sequences whose ground truth poorly correlates with the average ground truth of the whole sequence ($CC < 0.3$), under the assumption that in these situations something worth noticing is drifting the driver's focus from the vanishing point of the road. The last column of Tab. 3.6 reports the results computed on such subset. When



(a) ground truth (b) network prediction

(c) baseline: gaussian (d) baseline: average GT

Figure 3.13: Visual comparison between the ground truth, the prediction and the baselines on an example frame of the DR(eye)VE dataset.

tested on these sequences, the Gaussian baseline outperforms both the COARSE model and the average training ground truth baseline. To interpret this result, consider that when measuring a distance between distributions, a high probability but wrong prediction is severely penalized over a somehow uncertain prediction (*i.e.* a Gaussian with high variance). Nonetheless, the COARSE+FINE model scores higher than all other methods with a relative gain of about +35% over the second best.

Do we capture the attentional dynamics?

In Sec. 3.2.3 we quantitatively evaluated the proposed network. Here, we qualitatively investigate the ability of the model to learn both *where* and *what* a human driver would focus while driving. The results are then compared against the analysis previously introduced in Sec. 3.2.2. Figure 3.22 shows the average attentional maps predicted by our model, arranged by speed range. On each plot we also overlay precision errors (green) and recall errors (red), *i.e.* pixels whose value differs by more than

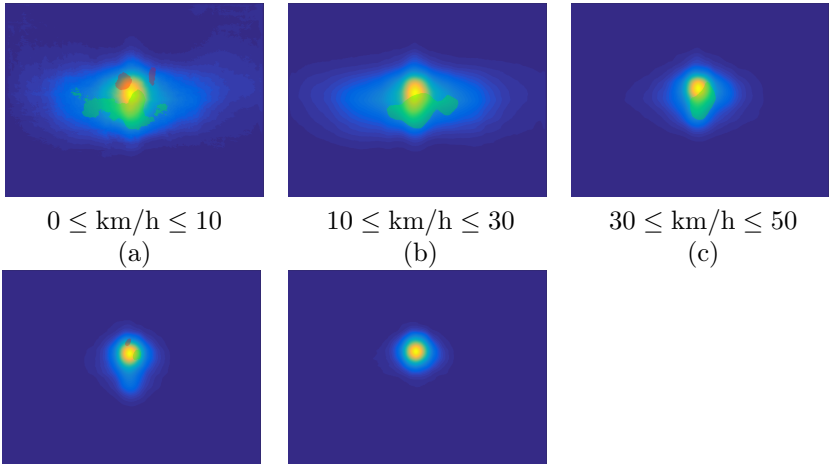


Figure 3.14: Model prediction averaged across all test sequences of `DR(eye)VE` dataset, grouped by driving speed. We highlight areas in which the mean prediction deviates by more than 10% from the mean ground truth. Precision errors are overlaid in green, while recall error in red.

10% from the analogous ground truth plots reported in Fig 3.7. We observe that i) the model generally succeeds in capturing the location of the driver gaze at different speed, ii) errors are mostly due to precision (prediction is wider than ground truth) and iii) errors decrease as the speed increases, as the lower variance of the gaze at high speed makes the modeling task easier.

3.2.4 Final remarks and future works

In this work we investigated the spatial and semantic attentional dynamics of the human driver and designed a deep network able to replicate such behavior while driving. These results eventually pave the way for a new assisted driving module, where real-time attentional maps support the driver by both decreasing fatigue and helping in keeping focus. We argue that such attentional maps can be less invasive than other Advanced Driver

Assistance Systems that directly act on the car (*e.g.* by activating breaks), whereas attentional maps leave full control to the driver.

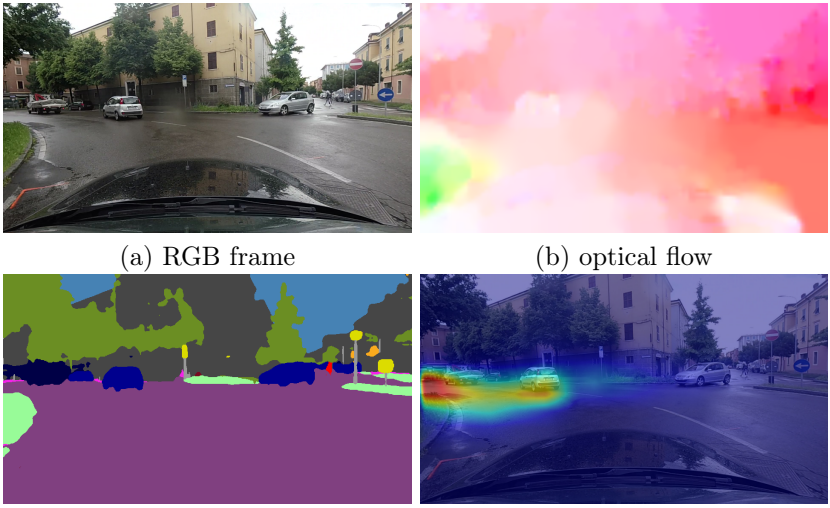
Despite the encouraging results, up to this point the proposed model architecture is fairly general-purpose and it does not take advantage of the insights got during dataset analysis in Sec. 3.1.3. Thus, in the following we customize the model architecture to exploit in explicit two factors which we know heavily influence the driver's attentional behavior: motion and scene semantics.

3.3 Integrating Motion and Semantics: a Multi-branch Architecture for Driver's Focus of Attention Prediction

In this section we build upon the data-driven study on drivers' gaze fixations under different circumstances and scenarios (see Sec. 3.1.3). The study suggests that the semantic of the scene, the speed and bottom-up features all influence the driver's gaze. Here we show how we integrated all these factors in the model architecture in a principled manner. We also advocate for the existence of common gaze patterns that are shared among different drivers; we empirically demonstrate the existence of such patterns by developing a deep learning model that can profitably learn to predict where a driver would be looking at in a specific situation.

3.3.1 Introduction

The `DR(eye)VE` data richness enables us to train an end-to-end deep network that predicts salient regions in car-centric driving videos. The network we propose is based on three branches which estimate attentional maps from a) visual information of the scene, b) motion cues (in terms of optical flow) and c) semantic segmentation (Fig. 3.15). In contrast to the majority of experiments, which are conducted in controlled laboratory settings or employ sequences of unrelated images [191, 21, 80], we train our model on data acquired on the field. Final results demonstrate the ability of the network to generalize across different day times, different weather conditions, different landscapes and different drivers.



(a) RGB frame (b) optical flow

(c) semantic segmentation (d) predicted map

Figure 3.15: An example of visual attention while driving (d), estimated from our deep model using (a) raw video, (b) optical flow and (c) semantic segmentation.

Eventually, we believe our work can be complementary to the current semantic segmentation and object detection literature [221, 196, 125, 30, 122] by providing a diverse set of information. According to [179], the act of driving combines complex attention mechanisms guided by the driver's past experience, short reactive times and strong contextual constraints. Thus, very little information is needed to drive if guided by a strong focus of attention (FoA) on a limited set of targets: our model aims at predicting them. The way humans favor some entities in the scene, along with key factors guiding eye fixations in presence of a given task (e.g. visual search) has been extensively studied for decades [186, 207]. The main difficulty that rises when approaching the subject is the variety of perspectives under which it can be cast. Indeed, visual attention has been approached by psychologists, neurobiologists and computer scientists, making the field highly interdisciplinary [53]. We are particularly interested in the computational

Figure 3.16: Examples taken from a random sequence of DR(eye)VE . From left to right: frames from the eye tracking glasses with gaze data, from the roof-mounted camera, temporal aggregated fixation maps (as defined in Sec. 3.1.2) and overlays between frames and fixation maps.

perspective, in which predicting human attention is often formalized as an estimation task delivering the probability of each point in a given scene to attract the observer's gaze.

3.3.2 Multi-branch architecture for focus of attention prediction

The DR(eye)VE dataset is sufficiently large to allow the construction of a deep architecture to model common attentional patterns. Here, we describe our neural network model to predict human FoA while driving.

Architecture design. In the context of high level video analysis (*e.g.* action recognition and video classification), it has been shown that a method leveraging single frames can be outperformed if a sequence of frames is used as input instead [185, 84]. Temporal dependencies are usually modeled either by 3D convolutional layers [185], tailored to capture short range correlations, or by recurrent architectures (*e.g.* LSTM, GRU), that can

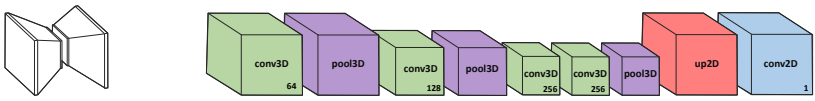


Figure 3.17: The `COARSE` module is made of an encoder based on C3D network [185] followed by a bilinear upsampling (bringing representations back to the resolution of the input image) and a final 2D convolution. During feature extraction, the temporal axis is lost due to 3D pooling. All convolutional layers are preceded by zero paddings in order keep borders, and all kernels have size 3 along all dimensions. Pooling layers have size and stride of (1, 2, 2, 4) and (2, 2, 2, 1) along temporal and spatial dimensions respectively. All activations are ReLUs.

model longer term dependencies [7, 133]. Our model follows the former approach, relying on the assumption that a small time window (*e.g.* half a second) holds sufficient contextual information for predicting where the driver would focus in that moment. Indeed, human drivers can take even less time to react to an unexpected stimulus. Our architecture takes a sequence of 16 consecutive frames ($\approx$ 0.65s) as input (called *clips* from now on) and predicts the fixation map for the last frame of such clip.
Many of the architectural choices made to design the network come from insights from the dataset analysis presented in Sec.3.1.3. In particular, we rely on the following results:

- the drivers' FoA exhibits consistent patterns, suggesting that it can be reproduced by a computational model;
- the drivers' gaze is affected by a strong prior on objects semantics, *e.g.* drivers tend to focus on items lying on the road;
- motion cues, like vehicle speed, are also key factors that influence gaze.

Accordingly, the model output merges three branches with identical architecture, unshared parameters and different input domains: the RGB image, the semantic segmentation and the optical flow field. We call this architecture `multi-branch` model. Following a bottom-up approach, in Sec. 3.3.2 the building blocks of each branch are motivated and described.

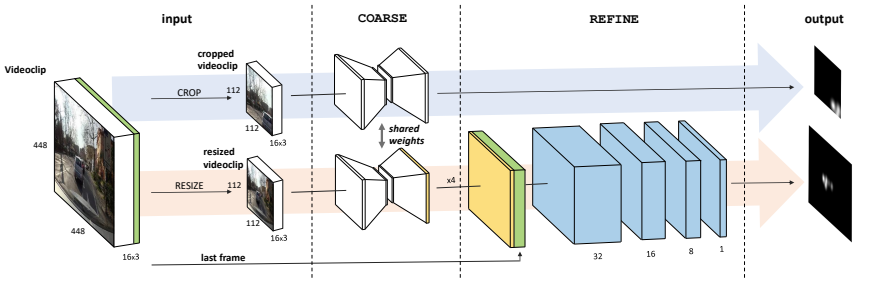


Figure 3.18: A single FoA branch of our prediction architecture. The `COARSE` module (see Fig. 3.17) is applied to both a cropped and a resized version of the input tensor, which is a videoclip of 16 consecutive frames. The cropped input is used during training to augment the data and the variety of ground truth fixation maps. The prediction of the resized input is stacked with the last frame of the videoclip and fed to a stack of convolutional layers (refinement module) with the aim of refining the prediction. Training is performed end-to-end and weights between `COARSE` modules are shared. At test time, only the refined predictions are used. Note that the complete model is composed of three of these branches (see Fig. 3.19), each of which predicting visual attention for different inputs (namely image, optical flow and semantic segmentation). All activations in the refinement module are LeakyReLU with $\alpha = 10^{-3}$, except for the last single channel convolution that features ReLUs. Crop and resize streams are highlighted by light blue and orange arrows respectively.

Later, in Sec. 3.3.2 it will be shown how the branches merge into the final model.

Single FoA branch

Each branch of the `multi-branch` model is a two-input two-output architecture composed of two intertwined streams. The aim of this peculiar setup is to prevent the network from learning a central bias, that would otherwise stall the learning in early training stages [2]. To this end, one of the streams

[2]For further details the reader can refer to Sec. 10 and Sec. 11 of the supplementary material.

is given as input (output) a severely cropped portion of the original image (ground truth), ensuring a more uniform distribution of the true gaze, and runs through the `COARSE` module, described below. Similarly, the other stream uses the `COARSE` module to obtain a rough prediction over the full resized image and then refines it through a stack of additional convolutions called `REFINE` model. At test time, only the output of the `REFINE` stream is considered. Both streams rely on the `COARSE` module, the convolutional backbone (with shared weights) which provides the rough estimate of the attentional map corresponding to a given clip. This component is detailed in Fig. 3.17.
The `COARSE` module is based on the C3D architecture [185] that encodes video dynamics by applying a 3D convolutional kernel on the 4D input tensor. As opposed to 2D convolutions that stride along the width and height dimension of the input tensor, a 3D convolution also strides along time. Formally, the j-th feature map in the i-th layer at position (x, y) at time t is computed as:

$$v_{i,j}^{x,y,t} = b_{i,j} + \sum_{m} \sum_{p=0}^{P_{i-1}} \sum_{q=0}^{Q_{i-1}} \sum_{r=0}^{R_{i-1}} w_{i,j,m}^{p,q,r} v_{i-1,m}^{x+p,y+q,t+r} \tag{3.2}$$

where m indexes different input feature maps, $w_{i,j,m}^{p,q,r}$ is the value at the position (p, q) at time r of the kernel connected to the m-th feature map, and P_i, Q_i and R_i are the dimensions of the kernel along width, height and temporal axis respectively; $b_{i,j}$ is the bias from layer i to layer j.
From C3D, only the most general-purpose features are retained by removing the last convolutional layer and the fully connected layers which are strongly linked to the original action recognition task. The size of the last pooling layer is also modified in order to cover the remaining temporal dimension entirely. This collapses the tensor from 4D to 3D, making the output independent of time. Eventually, a bilinear upsampling brings the tensor back to the input spatial resolution and a 2D convolution merges all features into one channel. See Fig. 3.17 for additional details on the `COARSE` module.

Training the two streams together The architecture of a single FoA branch is depicted in Fig. 3.18. During training, the first stream feeds the `COARSE` network with random crops, forcing the model to learn the current focus of attention given visual cues rather than prior spatial location. The

C3D training process described in [185], employs a 128×128 image resize, and then a 112×112 random crop. However, the small difference in the two resolutions limits the variance of gaze position in ground truth fixation maps and is not sufficient to avoid the attraction towards the center of the image. For this reason, training images are resized to 256×256 before being cropped to 112×112. This crop policy generates samples that cover less than a quarter of the original image thus ensuring a sufficient variety in prediction targets. This comes at the cost of a coarser prediction: as crops get smaller, the ratio of pixels in the ground truth covered by gaze increases, leading the model to learn larger maps.
In contrast, the second stream feeds the same `COARSE` model with the same images, this time *resized* to 112×112 – and not cropped. The coarse prediction obtained from the `COARSE` model is then concatenated with the final frame of the input clip, *i.e.* the frame corresponding to the final prediction. Eventually, the concatenated tensor goes through the `REFINE` module to obtain a higher resolution prediction of the FoA.
The overall two-stream training procedure for a single branch is summarized in Algorithm 1.

Training objective Prediction cost can be minimized in terms of Kullback-Leibler divergence:

$$D_{KL}(Y \| \hat{Y}) = \sum_i Y(i) \ \log \left(\epsilon + \frac{Y(i)}{\epsilon + \hat{Y}(i)} \right) \tag{3.3}$$

where Y is the ground truth distribution, $\hat{Y}$ is the prediction, the summation index i spans across image pixels and ϵ is a small constant that ensures numerical stability[3]. Since each single FoA branch computes an error on both the cropped image stream and the resized image stream, the branch loss can be defined as:

$$\begin{aligned} \mathcal{L}_b(\mathcal{X}_b, \mathcal{Y}) = \sum_m \Big(& D_{KL}(\phi(Y^m) \| \mathcal{C}(\phi(X_b^m))) + \\ & D_{KL}(Y^m \| \mathcal{R}(\mathcal{C}(\psi(X_b^m)), X_b^m))) \Big) \end{aligned} \tag{3.4}$$

[3]Please note that D_{KL} inputs are always normalized to be a valid probability distribution despite this may be omitted in notation to improve equations readability.

Algorithm 1 TRAINING. The model is trained in two steps: first each branch is trained separately through iterations detailed in **procedure** SINGLE_BRANCH_TRAINING_ITERATION, then the three branches are fine-tuned altogether as shown by **procedure** MULTI_BRANCH_FINE-TUNING_ITERATION. For clarity, we omit from notation: i) the subscript b denoting the current domain in all X, x and $\hat{y}$ variables in the single branch iteration and ii) the normalization of the sum of the outputs from each branch in line 13.

1: **procedure A:** SINGLE_BRANCH_TRAINING_ITERATION
input: domain data $X = \{x_1, x_2, \ldots, x_{16}\}$, true attentional map y of last frame x_{16} of videoclip X
output: branch loss $\mathcal{L}_b$ computed on input sample (X, y)
2: $X_{\text{res}} \leftarrow \texttt{resize}(X,\ (112,\ 112))$
3: $X_{\text{crop}},\ y_{\text{crop}} \leftarrow \texttt{get\_crop}((X,\ y),\ (112,\ 112))$
▷ Get coarse prediction on uncentered crop
4: $\hat{y}_{\text{crop}} \leftarrow \texttt{COARSE}(X_{\text{crop}})$
▷ Get refined prediction over whole image
5: $\hat{y} \leftarrow \texttt{REFINE}(\texttt{stack}(x_{16},\ \texttt{upsample}(\texttt{COARSE}(X_{\text{res}}))))$
▷ Compute branch loss as in Eq. 3.4
6: $\mathcal{L}_b(X, Y) \leftarrow D_{KL}(y_{\text{crop}} \| \hat{y}_{\text{crop}}) + D_{KL}(y \| \hat{y})$

7: **procedure B:** MULTI_BRANCH_FINE-TUNING_ITERATION
input: data $X = \{x_1, x_2, \ldots, x_{16}\}$ for all domains, true attentional map y of last frame x_{16} of videoclip X
output: overall loss $\mathcal{L}$ computed on input sample (X, y)
8: $X_{\text{res}} \leftarrow \texttt{resize}(X, (112, 112))$
9: $X_{\text{crop}},\ y_{\text{crop}} \leftarrow \texttt{get\_crop}((X,\ y), (112, 112))$
10: **for** branch $b \in \{\text{RGB}, \text{flow}, \text{seg}\}$ **do**
▷ As in line 4 of the above procedure
11: $\hat{y}_{b_{\text{crop}}} \leftarrow \texttt{COARSE}(X_{b_{\text{crop}}})$
▷ As in line 5 of the above procedure
12: $\hat{y}_b \leftarrow \texttt{REFINE}(\texttt{stack}(x_{b_{16}},\ \texttt{upsample}(\texttt{COARSE}(X_{b_{\text{res}}}))))$
▷ Compute overall loss as in Eq. 3.5
13: $\mathcal{L}(X, Y) \leftarrow D_{KL}(y_{\text{crop}} \| \sum_b \hat{y}_{b_{\text{crop}}}) + D_{KL}(y \| \sum_b \hat{y}_b)$

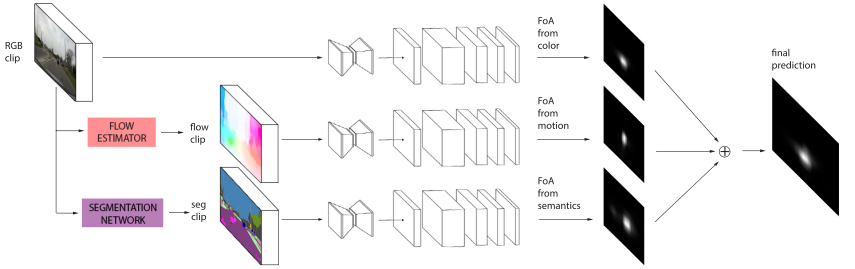


Figure 3.19: The `multi-branch` model is composed of three different branches, each of which has its own set of parameters, and their predictions are summed to obtain the final map. Note that in this figure cropped streams are dropped to ease representation, but are employed during training (as discussed in Sec. 3.3.2 and depicted in Fig. 3.18.

where $\mathcal{C}$ and $\mathcal{R}$ denote `COARSE` and `REFINE` modules, $(X_b^m, Y^m) \in \mathcal{X}_b \times \mathcal{Y}$ is the m-th training example in the b-th domain (namely RGB, optical flow, semantic segmentation), and ϕ and ψ indicate the crop and the resize functions respectively.

Inference step While the presence of the $\mathcal{C}(\phi(X_b^m))$ stream is beneficial in training to reduce the spatial bias, at test time only the $\mathcal{R}(\mathcal{C}(\psi(X_b^m)), X_b^m))$ stream producing higher quality prediction is used. The outputs of such stream from each branch b are then summed together, as explained in the following section.

Multi-Branch model

As described at the beginning of this section and depicted in Fig. 3.19, the `multi-branch` model is composed of three identical branches. The architecture of each branch has already been described in Sec. 3.3.2 above. Each branch exploits complementary information from a different domain and contributes to the final prediction accordingly. In detail, the first branch works in the RGB domain and processes raw visual data about the scene X_{RGB}. The second branch focuses on motion through the optical flow representation X_{flow} described in [61]. Eventually, the last branch takes as input semantic segmentation probability maps X_{seg}. For this last

branch, the number of input channels depends on the specific algorithm used to extract the results, 19 in our setup (Yu and Koltun [221]). The three independent predicted FoA maps are summed and normalized to result in a probability distribution.
To allow for larger batch size, we choose to bootstrap each branch independently by training it according to Eq. 3.4. Then, the complete `multi-branch` model which merges the three branches is fine-tuned with the following loss:

$$\mathcal{L}(\mathcal{X},\mathcal{Y}) = \sum_m \Bigg(D_{KL}(\phi(Y^m) \| \sum_b \mathcal{C}(\phi(X_b^m))) + D_{KL}(Y^m \| \sum_b \mathcal{R}(\mathcal{C}(\psi(X_b^m)), X_b^m))) \Bigg). \tag{3.5}$$

The algorithm describing the complete inference over the `multi-branch` model in detailed in Alg. 2.

3.3.3 Experiments

In this section we evaluate the performance of the proposed `multi-branch` model. First, we start by comparing our model against some baselines and other methods in literature. Following the guidelines in [22], for the evaluation phase we rely on Pearson's Correlation Coefficient (CC) and Kullback–Leibler Divergence (D_{KL}) measures. Moreover, we evaluate the Information Gain (IG) [92] measure to assess the quality of a predicted map P with respect to a ground truth map Y in presence of a strong bias,

Algorithm 2 INFERENCE. At test time, the data extracted from the resized videoclip is input to the three branches and their output is summed and normalized to obtain the final FoA prediction.

input: data $X = \{x_1, x_2, \ldots, x_{16}\}$ for all domains
output: predicted FoA map $\hat{y}$
1: $X_{\text{res}} \leftarrow \texttt{resize}(X, (112, 112))$
2: **for** branch $b \in \{\text{RGB}, \text{flow}, \text{seg}\}$ **do**
3: $\quad \hat{y}_b \leftarrow \texttt{REFINE}(\texttt{stack}(x_{b_{16}}, \texttt{upsample}(\texttt{COARSE}(X_{b_{\text{res}}}))))$
4: $\hat{y} \leftarrow \sum_b \hat{y}_b / \sum_i \sum_b \hat{y}_b(i)$

as:

$$IG(P, Y, B) = \frac{1}{N} \sum_i Y_i[(\log_2(\epsilon + P_i) - \log_2(\epsilon + B_i)] \tag{3.6}$$

where i is an index spanning all the N pixels in the image, B the bias computed as the average training fixation map and ϵ ensures numerical stability.
Furthermore, we conduct an ablation study to investigate how different branches affect the final prediction and how their mutual influence changes in different scenarios. We then study whether our model captures the attention dynamics observed in Sec. 3.1.3. Eventually, we assess our model from a human perception perspective.

Implementation details. The three different pathways of the `multi-branch` model (namely FoA from color, from motion and from semantics) have been pre-trained independently using the same cropping policy of Sec. 3.3.2 and minimizing the objective function in Eq. 3.4. Each branch has been respectively fed with:

- 16 frames clips in raw RGB color space;
- 16 frames clips with optical flow maps, encoded as color images through the flow field encoding [61];
- 16 frames clips holding semantic segmentation from [221] encoded as 19 scalar activation maps, one per segmentation class.

During individual branch pre-training clips were randomly mirrored for data augmentation. We employ Adam optimizer with parameters as suggested in the original paper [87], with the exception of the learning rate that we set to 10^{-4}. Eventually, batch size was fixed to 32 and each branch was trained until convergence. The `DR(eye)VE` dataset is split into train, validation and test set as follows: sequences 1-38 are used for training, sequences 39-74 for testing. The 500 frames in the middle of each training sequence constitute the validation set.
Moreover, the complete `multi-branch` architecture was fine-tuned using the same cropping and data augmentation strategies minimizing cost function in Eq. 3.5. In this phase batch size was set to 4 due to GPU memory constraints and learning rate value was lowered to 10^{-5}. Inference time of each branch of our architecture is $\approx$ 30 milliseconds per videoclip on an NVIDIA Titan X.

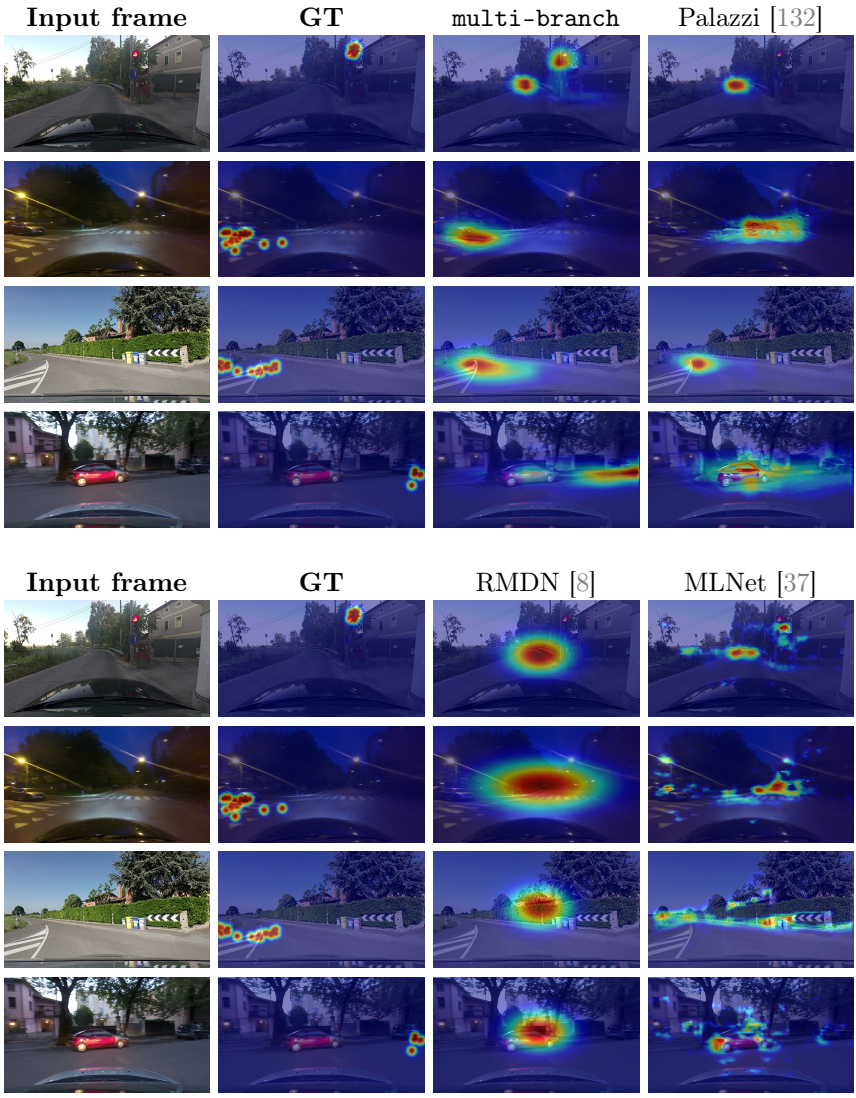


Figure 3.20: Visual assessment of the predicted fixation maps. From left to right: input clip, ground truth map, our prediction, prediction of the previous version of the model [132], prediction of RMDN [8] and prediction of MLNet [37].

Model evaluation

In Tab. 3.8 we report results of our proposal against other state-of-the-art models [200, 117, 37, 132, 8, 201] evaluated both on the complete test set and on *acting* subsequences only. All the competitors, with the exception of [132] are bottom-up approaches and mainly rely on appearance and motion discontinuities. To test the effectiveness of deep architectures for saliency prediction we compare against the Multi-Level Network (MLNet) [37], which scored favourably in the `MIT300` saliency benchmark [21], and the Recurrent Mixture Density Network (RMDN) [8], which represents the only deep model addressing video saliency. While MLNet works on images discarding the temporal information, RMDN encodes short sequences in a similar way to our `COARSE` module, and then relies on a LSTM architecture to model long term dependencies and estimates the fixation map in terms of a GMM. For a fair comparison, both models were re-trained on the `DR(eye)VE` dataset.
Results highlight the superiority of our `multi-branch` architecture on all test sequences. The gap in performance with respect to bottom-up unsupervised approaches [200, 201] is higher, and is motivated by the peculiarity of the attention behavior within the driving context, which calls for a task-oriented training procedure. Moreover, MLNet's low performance testifies for the need of accounting for the temporal correlation between consecutive frames that distinguishes the tasks of attention prediction in images and videos. Indeed, RMDN processes video inputs and outperforms MLNet on both D_{KL} and IG metrics, performing comparably on CC. Nonetheless, its performance is still limited: indeed, qualitative results reported in Fig. 3.20 suggest that long term dependencies captured by its recurrent module lead the network towards the regression of the mean, discarding contextual and frame-specific variations that would be preferrable to keep. To support this intuition, we measure the average D_{KL} between RMDN predictions and the mean training fixation map (Baseline Mean), resulting in a value of 0.11. Being lower than the divergence measured with respect to groundtruth maps, this value highlights the closer correlation to a central baseline rather than to groundtruth. Eventually, we also observe improvements with respect to our previous proposal [132], that relies on a more complex backbone model (also including a deconvolutional module) and processes RGB clips only. The gap in performance resides in the greater awareness of our `multi-branch` architecture of the aspects that

characterize the driving task as emerged from the analysis in Sec. 3.1.3. The positive performances of our model are also confirmed when evaluated on the *acting* partition of the dataset. We recall that *acting* indicates sub-sequences exhibiting a significant task-driven shift of attention from the center of the image (Fig. 3.5). Being able to predict the FoA also on *acting* sub-sequences means that the model captures the strong centered attention bias but is capable of generalizing when required by the context. This is further shown by the comparison against a centered Gaussian baseline (BG) and against the average of all training set fixation maps (BM). The former baseline has proven effective on many image saliency detection tasks [21] while the latter represents a more task-driven version. The superior performance of the `multi-branch` model w.r.t. baselines highlights that despite the attention is often strongly biased towards the vanishing point of the road, the network is able to deal with sudden task-driven changes in gaze direction.

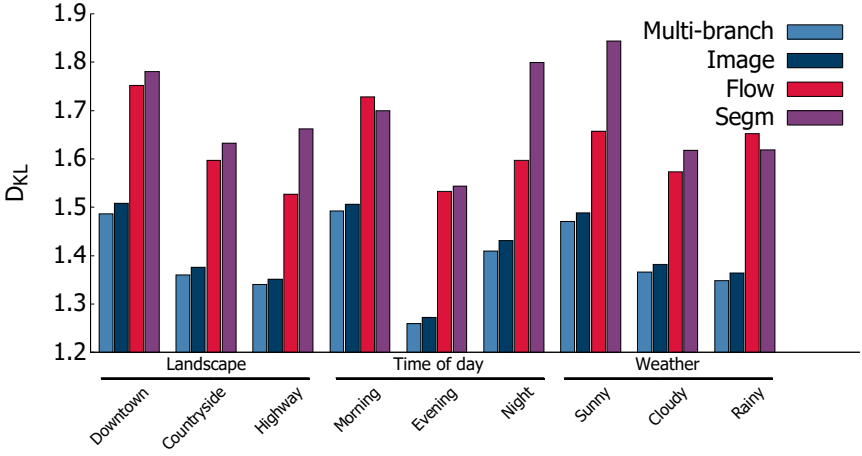


Figure 3.21: D_{KL} of the different branches in several conditions (from left to right: downtown, countryside, highway, morning, evening, night, sunny, cloudy, rainy). Underlining highlights difference of aggregation in terms of landscape, time of day and weather. Please note that lower D_{KL} indicates better predictions.

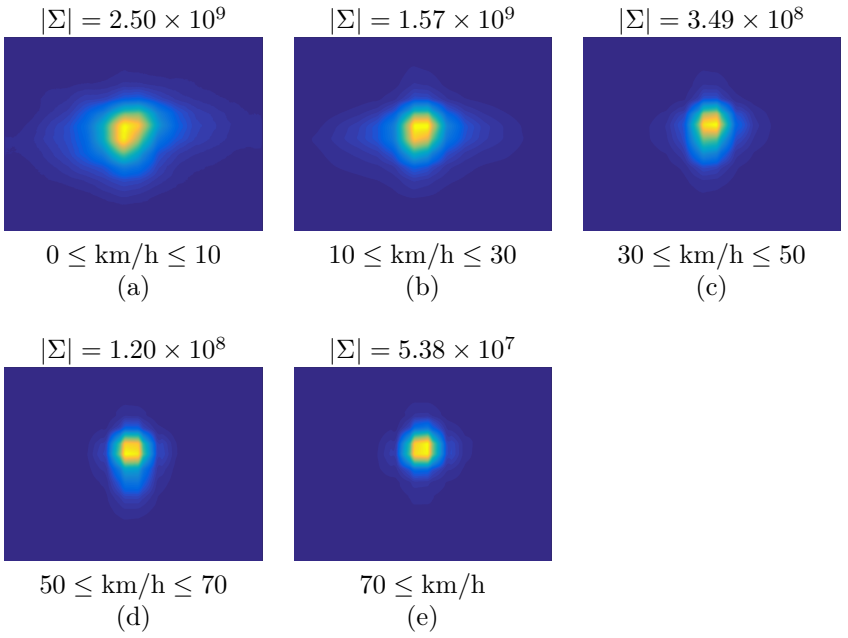


Figure 3.22: Model prediction averaged across all test sequences and grouped by driving speed. As the speed increases, the area of the predicted map shrinks, recalling the trend observed in ground truth maps. As in Fig. 3.7, for each map a two dimensional Gaussian is fitted and the determinant of its covariance matrix Σ is reported as a measure of the spread.

Model analysis

In this section we investigate the behavior of our proposed model under different landscapes, time of day and weather (Sec. 3.3.3); we study the contribution of each branch to the FoA prediction task (Sec. 3.3.3); and we compare the learned attention dynamics against the one observed in the human data (Sec. 3.3.3).

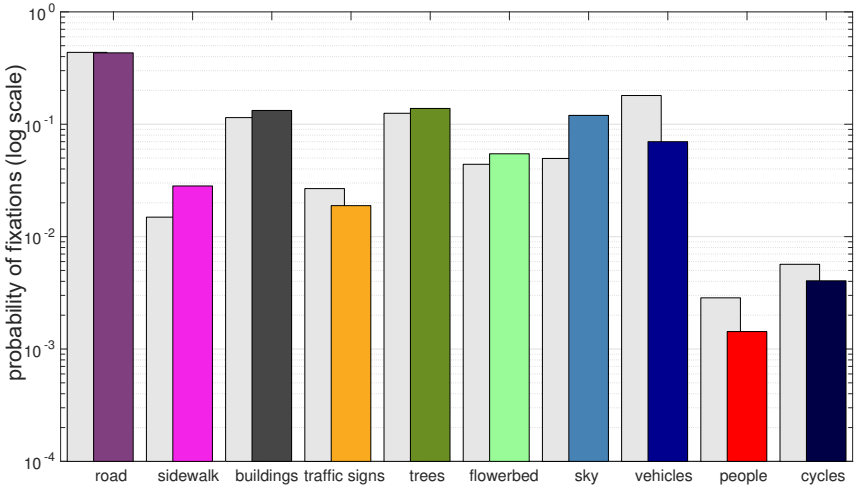


Figure 3.23: Comparison between ground truth (gray bars) and predicted fixation maps (colored bars) when used to mask semantic segmentation of the scene. The probability of fixation (in log-scale) for both ground truth and model prediction is reported for each semantic class. Despite absolute errors exist, the two bar series agree on the relative importance of different categories.

Dependency on driving environment

The `DR(eye)VE` data has been recorded under varying landscapes, time of day and weather conditions. We tested our model in all such different driving conditions. As would be expected, Fig. 3.21 shows that the human attention is easier to predict in highways rather than downtown, where the focus can shift towards more distractors. The model seems more reliable in evening scenarios, rather than morning or night, where we observed better lightning conditions and lack of shadows, over-exposure and so on. Lastly, in rainy conditions we notice that human gaze is easier to model, possibly due to the higher level of awareness demanded to the driver and his consequent inability to focus away from vanishing point. To support the latter intuition, we measured the performance of BM baseline (*i.e.* the average training fixation map), grouped for weather condition. As expected,

the D_{KL} value in rainy weather (1.53) is significantly lower than the ones for cloudy (1.61) and sunny weather (1.75), highlighting that when rainy the driver is more focused on the road.

Ablation study

In order to validate the design of the `multi-branch` model (see Sec. 3.3.2), here we study the individual contributions of the different branches by disabling one or more of them.
Results in Tab. 3.9 show that the RGB branch plays a major role in FoA prediction. The motion stream is also beneficial and provides a slight improvement, that becomes clearer in the *acting* subsequences. Indeed, optical flow intrinsically captures a variety of peculiar scenarios that are non-trivial to classify when only color information is provided, *e.g.* when the car is still at a traffic light or is turning. The semantic stream, on the other hand, provides very little improvement. In particular, from Tab. 3.9 and by specifically comparing I+F and I+F+S, a slight increase in the *IG* measure can be appreciated. Nevertheless, such improvement has to be considered negligible when compared to color and motion, suggesting that in presence of efficiency concerns or real-time constraints the semantic stream can be discarded with little losses in performance. However, we expect the benefit from this branch to increase as more accurate segmentation models will be released.

Do we capture the attention dynamics?

The previous sections validate quantitatively the proposed model. Now, we assess its capability to attend like a human driver by comparing its predictions against the analysis performed in Sec. 3.1.3.
First, we report the average predicted fixation map in several speed ranges in Fig. 3.22. The conclusions we draw are twofold: i) generally, the model succeeds in modeling the behavior of the driver at different speeds, and ii) as the speed increases fixation maps exhibit lower variance, easing the modeling task, and prediction errors decrease.
We also study how often our model focuses on different semantic categories, in a fashion that recalls the analysis of Sec. 3.1.3, but employing our predictions rather than ground truth maps as focus of attention. More precisely, we normalize each map so that the maximum value equals 1, and

apply the same thresholding strategy described in Sec. 3.1.3. Likewise, for each threshold value a histogram over class labels is built, by accounting all pixels falling within the binary map for all test frames. This results in nine histograms over semantic labels, that we merge together by averaging probabilities belonging to different threshold. Fig. 3.23 shows the comparison. Color bars represent how often the predicted map focuses on a certain category, while gray bars depict ground truth behavior and are obtained by averaging histograms in Fig. 3.9 across different thresholds. Please note that, to highlight differences for low populated categories, values are reported on a logarithmic scale. The plot shows a certain degree of absolute error is present for all categories. However, in a broader sense, our model replicates the relative weight of different semantic classes while driving, as testified by the importance of roads and vehicles, that still dominate, against other categories such as people and cycles that are mostly neglected. This correlation is confirmed by Kendall rank coefficient, which scored 0.51 when computed on the two bar series.

Do subtasks help in FoA prediction?

The driving task is inherently composed of many subtasks, such as turning or merging in traffic, looking for parking and so on. While such fine-grained subtasks are hard to discover (and probably to emerge during learning) due to scarcity, here we show how the proposed model has been able to leverage on more common subtask to get to the final prediction. These subtasks are: turning left/right, going straight, being still. We gathered automatic annotation through GPS information released with the dataset. We then train a linear SVM classifier to distinguish the above 4 different actions starting from the activations of the last layer of `multi-path` model, unrolled in a feature vector. The SVM classifier scores a 90% of accuracy on the test set (5000 uniformely sampled videoclips), supporting the fact that network activations are highly discriminative for distinguishing the different driving subtasks. Please refer to Fig. 3.24 for further details.

3.3.4 Conclusions

This section presented a study of human attention dynamics underpinning the driving experience. Our main contribution is a multi-branch deep network capable of capturing such factors and replicating the driver's focus

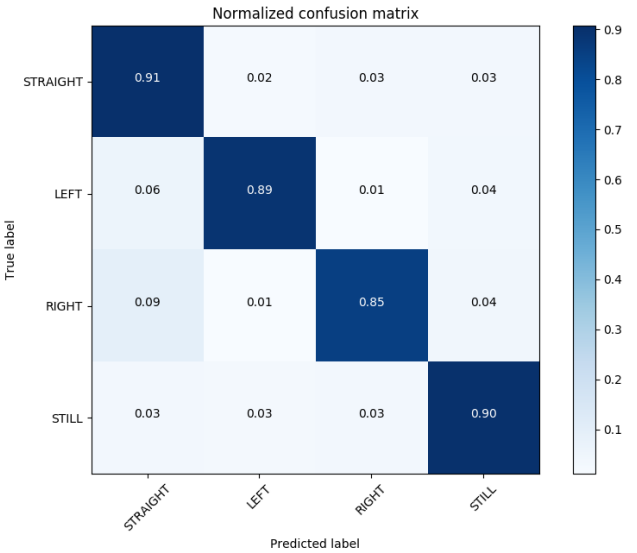


Figure 3.24: Confusion matrix for SVM classifier trained to distinguish driving actions from network activations. The accuracy is generally high, which corroborates the assumption that the model benefits from learning an internal representation of the different driving sub-tasks.

of attention from raw video sequences. The design of our model has been guided by a prior analysis highlighting i) the existence of common gaze patterns across drivers and different scenarios; and ii) a consistent relation between changes in speed, lightning conditions, weather and landscape, and changes in the driver's focus of attention. Experiments with the proposed architecture and related training strategies yielded state-of-the-art results. To our knowledge, our model is the first able to predict human attention in real-world driving sequences. As the model only input are car-centric videos, it might be integrated with already adopted ADAS technologies.

Table 3.8: Experiments illustrating the superior performance of the `multi-branch` model over several baselines and competitors. We report both the average across the complete test sequences and only the *acting* frames.

	Test sequences			Acting subsequences		
	CC ↑	D_{KL} ↓	IG ↑	CC ↑	D_{KL} ↓	IG ↑
Baseline Gaussian	0.40	2.16	-0.49	0.26	2.41	0.03
Baseline Mean	0.51	1.60	0.00	0.22	2.35	0.00
Mathe *et al.*[117]	0.04	3.30	-2.08	-	-	-
Wang *et al.*[200]	0.04	3.40	-2.21	-	-	-
Wang *et al.*[201]	0.11	3.06	-1.72	-	-	-
MLNet[37]	0.44	2.00	-0.88	0.32	2.35	-0.36
RMDN[8]	0.41	1.77	-0.06	0.31	2.13	0.31
Palazzi *et al.*[132]	0.55	1.48	-0.21	0.37	2.00	0.20
`multi-branch`	**0.56**	**1.40**	**0.04**	**0.41**	**1.80**	**0.51**

Table 3.9: The ablation study performed on our `multi-branch` model. I, F and S represent image, optical flow and semantic segmentation branches respectively.

	Test sequences			Acting subsequences		
	CC ↑	D_{KL} ↓	IG ↑	CC ↑	D_{KL} ↓	IG ↑
I	0.554	1.415	-0.008	0.403	1.826	0.458
F	0.516	1.616	-0.137	0.368	2.010	0.349
S	0.479	1.699	-0.119	0.344	2.082	0.288
I+F	0.558	1.399	0.033	**0.410**	1.799	0.510
I+S	0.554	1.413	-0.001	0.404	1.823	0.466
F+S	0.528	1.571	-0.055	0.380	1.956	0.427
I+F+S	**0.559**	**1.398**	**0.038**	**0.410**	**1.797**	**0.515**

3.4 Perceptual Assessment of Predicted Fixation Maps

3.4.1 A perceptual experiment

To further validate the predictions of our model from the human perception perspective, 50 people with at least 3 years of driving experience were asked to participate in a visual assessment[4]. First, a pool of 400 videoclips (each 40 seconds long) is sampled from the `DR(eye)VE` dataset. Sampling is weighted such that resulting videoclips are evenly distributed among different scenarios, weathers, drivers and daylight conditions. Also, half of these videoclips contain sub-sequences that were previously annotated as *acting*.

To approximate as realistically as possible the visual field of attention of the driver, sampled videoclips are pre-processed following the procedure in [197]. As in [197] we leverage the *Space Variant Imaging Toolbox* [137] to implement this phase, setting the parameter that halves the spatial resolution every 2.3° to mirror human vision [197, 94] (please see Sec. 3.4.2 and Sec. 3.4.3 for additional details). The resulting videoclip preserves details near to the fixation points in each frame, whereas the rest of the scene gets more and more blurred getting farther from fixations until only low-frequency contextual information survive. Coherently with [197] we refer to this process as *foveation* (in analogy with human foveal vision). Thus, pre-processed videoclips will be called *foveated videoclips* from now on. To appreciate the effect of this step the reader is referred to Fig. 3.25.

Foveated videoclips were created by randomly selecting one of the following three fixation maps: the ground truth fixation map (G videoclips), the fixation map predicted by our model (P videoclips) or the average fixation map in the `DR(eye)VE` training set (C videoclips). The latter central baseline allows to take into account the potential preference for a "stable" attentional map (*i.e.* lack of switching of focus).

Each participant was asked to watch five randomly sampled foveated videoclips. After each videoclip, he answered the following question:

[4] These were students (11 females, 39 males) of age between 21 and 26 ($\mu = 23.4, \sigma = 1.6$) recruited at University of Modena and Reggio Emilia on a voluntary basis through an online form.

Figure 3.25: The figure depicts a videoclip frame that underwent the foveation process. The attentional map (above) is employed to blur the frame in a way that approximates the foveal vision of the driver[137]. In the foveated frame (below), it can be appreciated how the ratio of high-level information smoothly degrades getting farther from fixation points.

- Would you say the observed attention behavior comes from a human driver? (yes/no)

Each of the 50 participant evaluates five foveated videoclips, for a total of 250 examples.
The confusion matrix of provided answers is reported in Fig. 3.26. Participants were not particularly good at discriminating between human's gaze and model generated maps, scoring about the 55% of accuracy which is comparable to random guessing; this suggests our model is capable of producing plausible attentional patterns that resemble a proper driving behavior to a human observer.

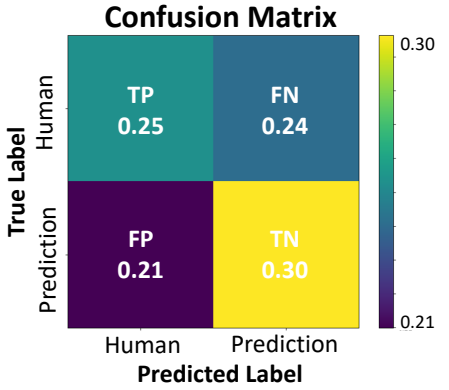
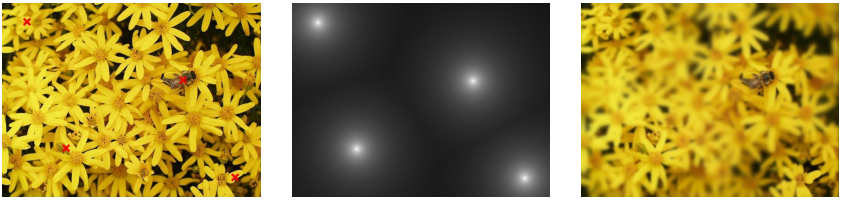


Figure 3.26: The confusion matrix reports the results of participants' guesses on the source of fixation maps. Overall accuracy is about 55% which is fairly close to random chance.

3.4.2 Space Variant Imaging System (SVIS)

The aforementioned perceptual experiment would have been impossible without a well-grounded algorithm for performing image foveation. We decided to rely on Space Variant Imaging System (SVIS), a MATLAB toolbox that allows to foveate images in real-time[137], which has been used in a large number of scientific works to approximate human foveal vision since its introduction in 2002. In this frame, the term *foveated imaging* refers to the creation and display of static or video imagery where the resolution varies across the image. In analogy to human foveal vision, the highest resolution region is called the foveation region. In a video, the location of the foveation region can obviously change dynamically. It is also possible to have more than one foveation region in each image.

The foveation process is implemented in the SVIS toolbox as follows: first the the input image is repeatedly low-passed filtered and down-sampled to half of the current resolution by a *Foveation Encoder*. In this way a low-pass pyramid of images is obtained. Then a foveation pyramid is created selecting regions from different resolutions proportionally to the distance from the foveation point. Concretely, the foveation region will be at the highest resolution; first ring around the foveation region will be taken



(a) (b) (c)

Figure 3.27: Foveation process using SVIS software is depicted here. Starting from one or more fixation points in a given frame (a), a smooth resolution map is built (b). Image locations with higher values in the resolution map will undergo less blur in the output image (c).

from half-resolution image; and so on. Eventually, a *Foveation Decoder* up-sample, interpolate and blend each layer in the foveation pyramid to create the output foveated image.

The software is open-source and publicly available here: `http://svi.cps.utexas.edu/software.shtml`. The interested reader is referred to the SVIS website for further details.

3.4.3 A deeper look into videoclip foveation

From fixation maps back to fixations. The SVIS toolbox allows to foveate images starting from a list of (x, y) coordinates which represent the foveation points in the given image (please see Fig. 3.27 for details). However, we do not have this information as in our work we deal with continuous attentional maps rather than discrete points of fixations. To be able to use the same software API we need to regress from the attentional map (either true or predicted) a list of approximated yet plausible fixation locations. To this aim we simply extract the 25 points with highest value in the attentional map. This is justified by the fact that in the phase of dataset creation the ground truth *fixation map* F_t for a frame at time t is built by accumulating projected gaze points in a temporal sliding window of $k = 25$ frames, centered in t (see Sec.3 of the paper). The output of this phase is thus a fixation map we can use as input for the SVIS toolbox.

Taking the blurred-deblurred ratio into account. To the visual assessment purposes, keeping track the amount of blur that a videoclip has undergone is also relevant. Indeed, a certain video may give rise to higher perceived safety only because a more delicate blur allows the subject to see a clearer picture of the driving scene. In order to consider this phenomenon we do the following.
Given an input image $I \in \mathbb{R}^{h,w,c}$ the output of the Foveation Encoder is a resolution map $R_{map} \in \mathbb{R}^{h,w,1}$, taking value in range $[0, 255]$, as depicted in Fig. 3.27 (b). Each value indicates the resolution that a certain pixel will have in the foveated image after decoding, where 0 and 255 indicates minimum and maximum resolution respectively.
For each video $\mathbf{v}$, we measure video average resolution after foveation as follows:

$$\mathbf{v}_{res} = \frac{1}{N} \sum_{f=1}^{N} \sum_{i} R_{map}(i, f)$$

where N is the number of frames in the video (1000 in our setting) and $R_{map}(i, f)$ denotes the i^{th} pixel of the resolution map corresponding to the f^{th} frame of the input video. The higher the value of v_{res} the more information is preserved in the foveation process. Due to the sparser location of fixations in ground truth attentional maps, these result in much less blurred videoclips. Indeed videos foveated with model predicted attentional maps have in average only the 38% of the resolution w.r.t. videos foveated starting from ground truth attentional maps. Despite this bias, model predicted foveated videos still gave rise to higher perceived safety to assessment participants.

3.4.4 Perceived safety assessment

The assessment of predicted fixation maps described in Sec 5.3 has also been carried out for validating the model in terms of perceived safety. Indeed, participants were also asked to answer the following question:

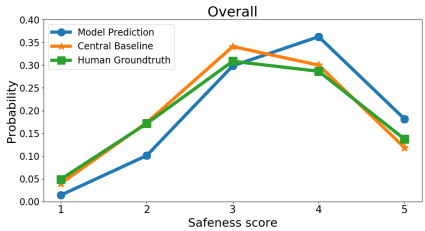
- If you were sitting in the same car of the driver whose attention behavior you just observed, how safe would you feel? (rate from 1 to 5)

The aim of the question is to measure the comfort level of the observer during a driving experience when suggested to focus at specific locations in

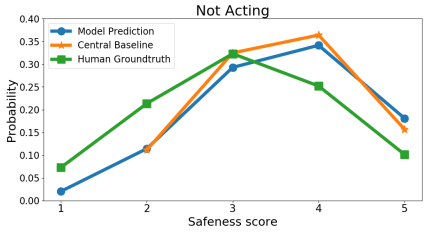
the scene. The underlying assumption is that the observer is more likely to feel safe if he agrees that the suggested focus is lighting up the right portion of the scene, that is what he thinks it is worth looking in the current driving scene. Conversely, if the observer wishes to focus at some specific location but he cannot retrieve details there, he is going to feel uncomfortable.

The answers provided by subjects, summarized in Fig. 3.28, indicate that perceived safety for videoclips foveated using the attentional maps predicted by the model is generally higher than for the ones foveated using either human or central baseline maps. Nonetheless the central bias baseline proves to be extremely competitive, in particular in non-acting videoclips in which it scores similarly to the model prediction. It is worth noticing that in this latter case both kind of automatic predictions outperform human ground truth by a significant margin (Fig. 3.28b). Conversely, when we consider only the foveated videoclips containing acting subsequences, the human ground truth is perceived as much safer than the baseline, despite still scores worse than our model prediction (Fig. 3.28c). These results hold despite due to the localization of the fixations the average resolution of the predicted maps is only the 38% of the resolution of ground truth maps (*i.e.* videos foveated using prediction map feature much less information). We did not measure significant difference in perceived safety across the different drivers in the dataset ($\sigma^2 = 0.09$).

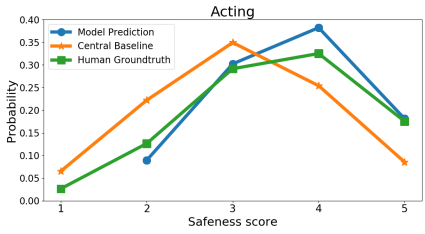
We report in Fig 3.29 the composition of each score in terms of answers to the other visual assessment question ("Would you say the observed attention behavior comes from a human driver? (yes/no)"). This analysis aims to measure participants' bias towards human driving ability. Indeed, increasing trend of false positives towards higher scores suggests that participants were tricked into believing that "safer" clips came from humans. The reader is referred to Fig. 3.29 for further details.



(a)



(b)



(c)

Figure 3.28: Distributions of safeness scores for different map sources, namely Model Prediction, Center Baseline and Human Groundtruth. Considering the score distribution over all foveated videoclips (a) the three distributions may look similar, even though the model prediction still scores slightly better. However, when considering only the foveated videos containing acting subsequences (b) the model prediction significantly outperforms both center baseline and human ground truth. Conversely, when the videoclips did not contain acting subsequences (*i.e.* the car was mostly going straight) the fixation map from human driver is the one perceived as less safe, while both model prediction and center baseline perform similarly.

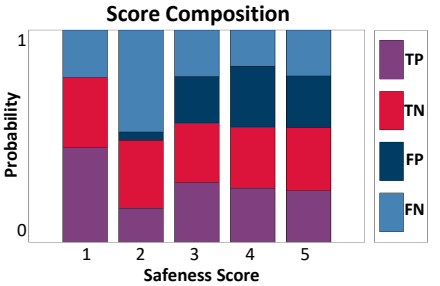


Figure 3.29: The stacked bar graph represents the ratio of TP, TN, FP and FN composing each score. The increasing score of FP – participants falsely thought the attentional map came from a human driver – highlights that participants were tricked into believing that "safer" clips came from humans.

Chapter 4

Outside the Vehicle: Infrastructure-level Understanding of the Urban Scene

The previous chapter tackled the problem of urban scene understanding from an inside-the-vehicle point of view. There we presented an extensive study on driver's attention, as well as introducing novel deep learning models to predict which visual regions are more likely to be salient for the task of driving. Conversely, here we zoom out from the vehicle to the infrastructure point of view. Although the focus will still be on vehicles, these will be viewed from the outside - as the most important agents which populate the urban scene. In this perspective, the ability to infer fine grained vehicles information - such as identity, model, 3D pose and occupancy - becomes essential to comprehend the scene; Sec 4.1, Sec. 4.2 and Sec 4.3 present novel methods in this direction. Eventually, we propose a novel framework to exploit these information to 'hallucinate' novel views of the vehicles and of the urban scene in its whole (Sec. 4.4). We foresee that the capability to imagine different visual appearance of the scene in the next future might have significant applications in many domains;

surveillance, vehicle re-identification and forensics to name a few.

4.1 Unsupervised Vehicle Re-Identification using Triplet Networks

Vehicle re-identification is a problem with a huge application significance, and it already plays a major role in modern smart surveillance systems. From an high-level perspective, the problem is basically the one of matching vehicles identities across non-overlapping views from different cameras. More formally, it can be cast as a ranking problem: given a probe image of a vehicle, the model needs to rank candidate images based on their similarities w.r.t the probe image.

Here we present a metric learning model that employs a supervision based on local constraints. In particular, we leverage pairwise and triplet constraints for training a network capable of assigning a high degree of similarity to samples sharing the same identity, while keeping different identities far apart in feature space. Eventually, we show how vehicle tracking can be exploited to automatically generate a weakly labelled dataset that can be used to train the deep network for the task of vehicle re-identification.

4.1.1 A pipeline for vehicle re-identification

Overall, the system is composed of three main modules (Figure 4.2):

1. A detector identifies all vehicles appearing in the region of interest. Each detection is either assigned to an existing tracklet or a new tracker is initialized from it (Sec. 4.1.1).

2. Exploiting the aforementioned tracklets, a triplet network is trained to keep vehicles belonging to the same tracklet close in a learned feature space. (Sec. 4.1.1)

3. A matching strategy is employed to re-identify vehicles between different videos. (Sec. 4.1.1)

In the following we detail each of these components separately.

Figure 4.1: Examples of real-world settings in which the task of re-identification is particularly challenging. Large illumination changes (a), (b), completely different scales (c), cluttered scenes (d). Images taken from the NVIDIA AI city challenge videos.

Detection and Tracking

The goal of a detector is to detect all objects belonging to a particular class in a scene, regardless of their intra-class variation. In the case of vehicles appearing in real-world videos as the ones in the NVIDIA AI City Challenge, detection is made challenging by many factors of variation (e.g. different scales, poses and lighting conditions).

In order to alleviate these issues, in each of the challenge video a region of interest (ROI) is manually selected in order to preserve as much information as possible while reducing computational effort. Vehicles outside the RoI are likely to be too far to provide reliable information. This cropping policy allows to greatly reduce the number of false positives, trading off this gain with a small loss on the detector recall. An example of the considered ROI for one of the challenge videos is depicted in Figure 4.4. Privileging

Figure 4.2: Our re-identification pipeline that is composed of three modules. The first phase is vehicle detection and tracking. Detections are either assigned to an existing tracklet or used to initialize a new tracker. Tracklets are exploited to automatically annotate the videos and train a triplet network for vehicle re-identification. The output vector of the triplet network is used as feature vector to represent each detected vehicle. Eventually, these feature vectors are compared between different probes and the gallery to generate a ranking. We refer the reader to Section 4.1.1 for further details.

precision over recall is particularly important since the output of the tracker is later used to automatically label the dataset.

After qualitatively evaluating the performance of various state-of-the-art detectors [146, 60, 148], we employ as detector the Single Shot MultiBox Detector (SSD) [103] architecture since it gave us the best results on the challenge videos. The SSD network is built upon a VGG-16 backbone and is trained using the COCO dataset and then fine-tuned using only the vehicle class. For details on the SSD architecture we refer the reader to the original SSD paper [103].

Detections are filtered in order to remove the ones in which the vehicle is only partially present in the bounding box, e.g. at the edge of the frame. We then use the detection to initialize the same correlation tracker as [39]. Whenever a new vehicle is detected, the tracker is initialized and then updated with new detections until the vehicle leaves the region of interest. Furthermore each different vehicle track has a different *ID* in case it appears among different videos.

Representation learning

As mentioned above, NVIDIA AI city challenge videos do not come along with any annotation. Thus, we apply the method shown in [202] to the vehicle re-identification task to create an annotation in an unsupervised manner - exploiting visual tracking to produce a (weakly) labelled training set for our task. As result, for each video we identify as *positive examples*

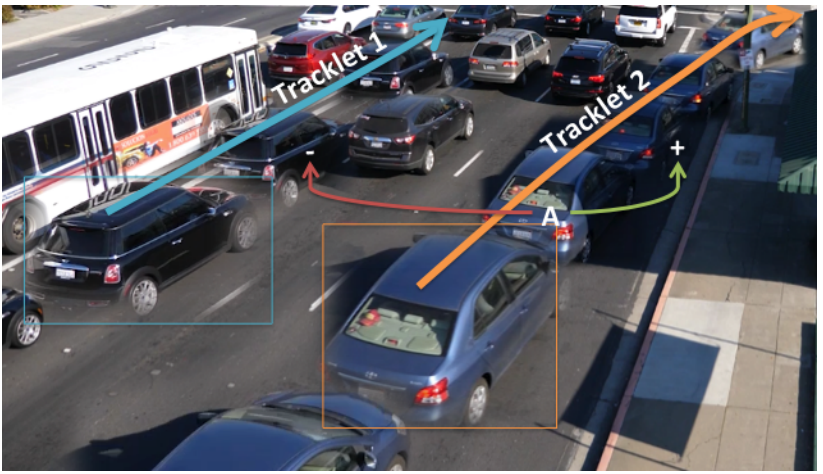


Figure 4.3: Here the automatic labeling of the NVIDIA AI city challenge videos is schematized. Each detected car is tracked until it exits the region of interest. Different detections belonging to the same tracklet constitute positive examples for the triplet network (green arrow). Conversely, patches that belong to different tracklets are labelled as negative examples (red arrow). See Section 4.1.1 for details.

different detections belonging to the same tracklet and *negative examples* couples of detections belonging to different tracklets. More formally, for each detection of a particular vehicle x_i we define the set of positive and negative pairs as follows:

$$X_i^+ = \{x_j | t(x_j) = t(x_i)\} \tag{4.1}$$

$$X_i^- = \{x_j | t(x_j) \neq t(x_i)\} \tag{4.2}$$

where $t(x_i)$ indicates the tracklet to which detection x_i belongs to, *i.e.* the tracklet *ID*. We can now form a set of triplets $\mathcal{T}$ as follows:

$$\mathcal{T} = \{(x_i, x_i^+, x_i^-) | x^+ \in X_i^+, x^- \in X_i^-\} \tag{4.3}$$

where x_i are detections from the NVIDIA AI city challenge videos and similar and different vehicles are sampled from X_i^+ and X_i^- sets respectively.

Figure 4.4: Example of considered Region of Interest (ROI) for location 4 of NVIDIA AI city challenge. It can be appreciated how the farthest vehicles are ignored, thus trading off the detector recall for an improved precision. Ignored detections usually correspond to the farthest vehicles, which would be anyway very hard to track correctly. Since in the successive phase tracklets are used as ground truth annotations for the challenge videos we choose to privilege the precision w.r.t. the recall of the tracker.

The underlying assumption is that the tracker is always correct: despite this is not the case, we empirically verify that the generated labelled dataset is reasonable enough to be useful in practice.

In general, a common approach in re-identification is to map any given example (possibly of variable size) to a vector of fixed low dimensionality. This dense representation can be later used for the matching stage. Specifically, the input bounding box $b_i \in \mathbb{R}^{w \times h}$ is transformed into a vector $v_i \in \mathbb{R}^d$, where w, h indicate the width and height of the detected bounding box and d is the dimensionality of the representation space. Commonly $d << w \times h$, which greatly speeds up the successive matching phase.

In order to tell different detections of the same vehicle apart, we need to represent the vehicle's visual appearance in a feature space in which

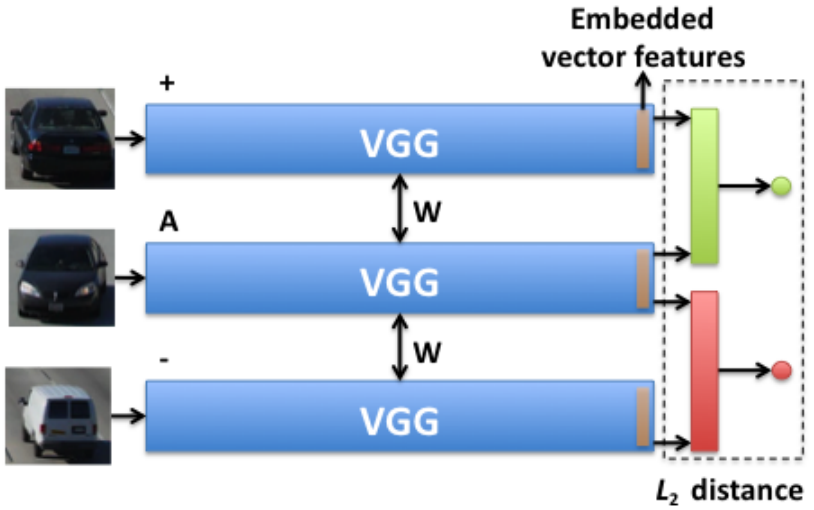


Figure 4.5: Triplet network architecture. The network is composed of three branches with shared weights, initialized from VGG-16 [167] parameters pre-trained on ImageNet [40] dataset. Details in Sec. 4.1.1.

similar vehicles lie closer than different ones. To this end we leverage on the triplet network architecture [73] to represent each detected vehicle with the output vector of the network. This architecture is based on three VGG-16 networks sharing the same weights and is depicted in Figure 4.5. The very last layer of the network is a fully connected layer of dimension d: this is used as feature vector. The triplet network can be trained for the task of vehicle re-identification using the set automatically labelled triplet $\mathcal{T}$. The intuition is that the distance between negative pairs is required to be larger than distance from positive pairs (plus a margin). Formally we want to minimize the following hinge loss:

$$d_i = ||f(x_i) - f(x_i^+)||_2^2 - ||f(x_i) - f(x_i^-)||_2^2$$
$$L_T = \sum_{(x_i, x_i^+, x_i^-) \in \mathcal{T}} \max(0, d_i + \gamma) \tag{4.4}$$

where $\gamma \geq 0$ is a positive margin and $f(x_i)$ is the network output for detection x_i.

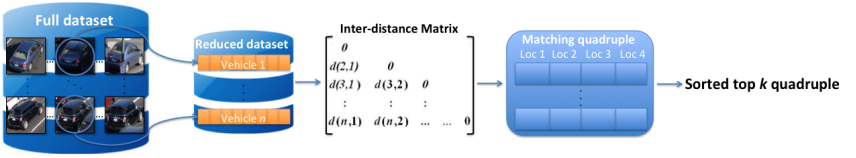


Figure 4.6: During the matching phase a single dense representation need to be extracted from each tracklet. Indeed, frames in each tracklet exhibit a high redundancy (*i.e.* the visual appearance of the vehicle hardly changes from one frame to the next). Thus we represent each tracklet with the feature vector of the vehicle in the middle of the tracklet.

Matching strategy

To be able to match identities of vehicles which belong to different tracklets, a single dense representation need to be extracted from each tracklet. Also, since a tracklet can last for several hundreds of frames, information is extremely redundant (*i.e.* the visual appearance of the vehicle hardly changes from one frame to the next). Thus, during the matching phase we choose to encode each tracklet with the feature vector of the vehicle in the median frame of the tracklet (see Figure 4.6). Furthermore, tracklets are grouped by the location of the video $(1 \dots 4)$, under the assumption that a car does not appear more than once in each location. To compute the matches, we iterate over all different vehicle *ID*s, each one represented by the feature vector of the median frame of the tracklet. Euclidean distance is used to compare the two feature vectors::

$$d_{ij} = ||f(x_i) - f(x_j)||_2 \tag{4.5}$$

This distance can be used to compute the best match with vehicles from different video locations, where lower distance indicates a better match. Although there is always a best match candidate for each vehicle, matches are validated only in case the distance is lower than a definite threshold $\theta \geq 0$. Furthermore, following the indication of the NVIDIA AI City Challenge, we keep only one ID correspondence for each of the four locations. We then consider the quadruple composed by

$$\{ID_{min(loc_1)}, ID_{min(loc_2)}, ID_{min(loc_3)}, ID_{min(loc_4)}\}$$

as the proposed vehicle re-identified. Moreover, once a vehicle ID is assigned to one quadruple, we remove the correspondent ID to avoid it to be chosen again in future comparisons. Eventually, once all the IDs are processed, we compute the average distance among the members of each quadruple. This distance is then normalized to lie in range $[0, 1]$ as used as measure of re-identification confidence to sort the matches. In this way we can keep only the top k similar groups.

4.1.2 Implementation details

The methodology is applied over all 15 videos of the NVIDIA AI city challenge, a total of 15 hour approx. of recording. Videos are captured at 30 frames per second (fps) with a Canon EOS 550D camera at four different locations (I280 and Winchester, I280 and Wolfe, San Tomas and Saratoga, Stevens Creek and Winchester) and feature a resolution of 1920×1080 pixels.
To reduce the computational burden, each vehicle's detection is resized to 80×80 pixels in RGB color space. Overall, our automatically-annotated dataset is composed by 2,198,829 vehicles belonging to 67,825 different tracklets.
The triplet network is trained using a batch size equal to 64 for a total of 10 epoch. We minimize the mean squared error loss using a SGD optimizer with a learning rate of 0.01. We empirically choose the size of the feature vector equal to 100 since it gave the best results.
Detections whose centroid is closer than 100 pixels from the edge of the frame are ignored. Eventually, the threshold θ used during the matching phase is set to 3,500.

4.2 Mapping Vehicles into Bird's Eye View

Awareness of the surrounding road scene is becoming an essential component for any ground vehicle. This information is exploited by Advanced Driver Assistance Systems (ADAS), while there is still a human driver behind the wheel. In the longer run, this will be an essential ability required to all autonomous agents to safely navigate in their environment. In fact, vision-based algorithms and models have massively been adopted in cur-

rent generation ADAS solutions. Moreover, recent research achievements on scene semantic segmentation [59, 100], road obstacle detection [9, 96] and driver's gaze, pose and attention prediction [41, 193] are playing a major role in the newborn autonomous mobility sector. In this frame, here we present a way to learn a semantic-aware transformation which maps detections from a single dashboard camera frame onto a broader bird's eye occupancy map of the scene. We demonstrate the effectiveness of our model against several baselines and observe that is able to generalize on real-world data despite having been trained solely on synthetic ones.

Dataset, code and pre-trained model are publicly available and can be found at `http://imagelab.ing.unimore.it/scene-awareness`.

4.2.1 An interpretable proxy of the road state

As suggested in [26], three major paradigms can be individuated for vision-based autonomous driving systems: *mediated perception* approaches, based on the total understanding of the scene around the car, *behavior reflex* methods, in which driving action is regressed directly from the sensory input, and *direct perception* techniques, that fuse elements of previous approaches and learn a mapping between the input image and a set of interpretable indicators which summarize the driving situation.
Following this last line of work, here we develop a model for mapping vehicles across different views. In particular, our aim is to warp vehicles detected from a dashboard camera view into a bird's eye occupancy map of the surroundings, which is an easily interpretable proxy of the road state. Being almost impossible to collect a dataset with this kind of information in real-world, we exclusively rely on synthetic data for learning this projection. We aim to create a system close to surround vision monitoring ones, also called around view cameras that can be useful tools for assisting drivers during maneuvers by, for example, performing trajectory analysis of vehicles out from own visual field.
In this framework, our contribution is twofold:

- We make available a huge synthetic dataset (> 1 million of examples) which consists of couple of frames corresponding to the same driving scene captured by two different views. Besides the vehicle location, auxiliary information such as the distance and yaw of each vehicle at each frame are also present.

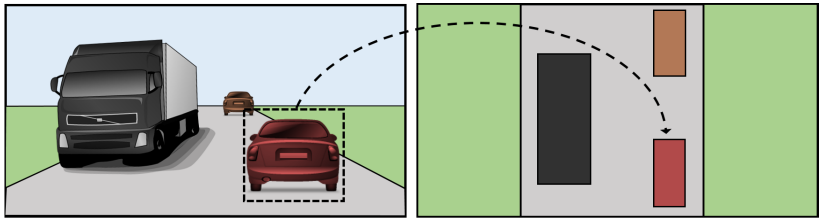


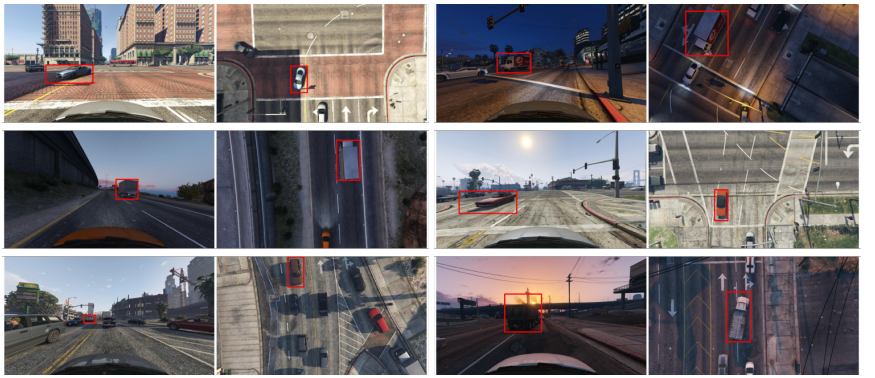
Figure 4.7: Simple outline of our task. Vehicle detections in the frontal view (left) are mapped onto a bird's-eye view (right), accounting for the positions and size.

- We propose a deep learning architecture for generating bird's eye occupancy maps of the surround in the context of autonomous and assisted driving. Our approach does not require a stereo camera, nor more sophisticated sensors like radar and lidar. Conversely, we learn how to project detections from the dashboard camera view onto a broader bird's eye view of the scene (see Fig.4.7). To this aim we combine learned geometric transformation and visual cues that preserve objects size and orientation in the warping procedure.

Many elements mark as original our approach. In principle, we want our surround view to include not only nearby elements, like commercial geometry-based systems, but also most of the elements detected into the acquired dashboard camera frame. Additionally, no specific initialization or alignment procedures are necessary: in particular, no camera calibration and no visible alignment points are required. Eventually, we aim to preserve the correct dimensions of detected objects, which shape is mapped onto the surround view consistently with their semantic class.

4.2.2 Surround Vehicle Awareness (SVA) Dataset

In order to collect data, we exploit *Script Hook V* library [10], which allows to use Grand Theft Auto V (GTAV) video game native functions [3]. We develop a framework in which the game camera automatically toggle between frontal and bird-eye view at each game time step: in this way we are able to gather information about the spatial occupancy of the vehicles in



(a)

(b)

Figure 4.8: (a) Randomly sampled couples from our SVA dataset, which highlight the huge variety in terms of landscape, traffic condition, vehicle models etc. Each detection is treated as a separate training example (see Sec. 4.2.2 for details). (b) Random examples rejected during the post-processing phase. These are mostly due to the game engine failing to provide the right bounding box coordinates around an entity. Best viewed zoomed on screen.

the scene from both views (*i.e.* bounding boxes, distances, yaw rotations). We associate vehicles information across the two views by querying the game engine for entity IDs. More formally, for each frame t, we compute the set of entities which appear in both views as

$$E(t) = E_{frontal}(t) \cap E_{birdeye}(t) \tag{4.6}$$

where $E_{frontal}(t)$ and $E_{birdeye}(t)$ are the sets of entities that appear at time t in frontal and bird's eye view, respectively. Entities $e(t) \in E(t)$ constitute the candidate set for frame t $C(t)$; other entities are discarded. Unfortunately, we found that raw data coming from the game engine are not always accurate (Fig. 4.8). To deal with this problem, we implement a post-processing pipeline in order to discard noisy data from the candidate set $C(t)$. We define a discriminator function

$$f(e(t)) : C \mapsto \{0, 1\} \tag{4.7}$$

which is positive when information on dumped data $e(t)$ are reliable and zero otherwise. Thus we can define the final filtered dataset as

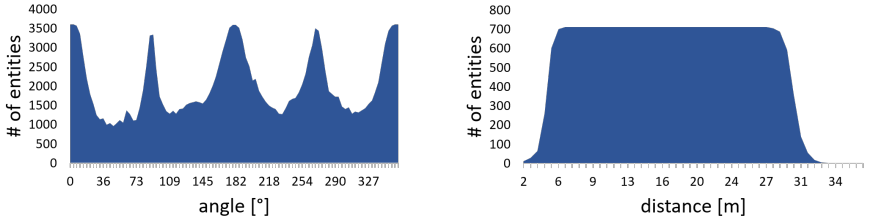
$$\bigcup_{t=0}^{T} D(t) \qquad \text{where} \quad D(t) = \{c_i(i) \mid f(c_i(t)) > 0\} \tag{4.8}$$

being T the total number of frames recorded. From an implementation standpoint, we employ a rule-based ontology which leverage on entity information (*e.g.* vehicle model, distance etc.) to decide if the bounding box of that entity can be considered reasonable. This implementation has two main values: first it's lightweight and very fast in filtering massive amounts of data. Furthermore, rule parameters can be tuned to eventually generate different dataset distribution (*e.g.* removing all trucks, keeping only cars closer than 10 meters, etc.).
Each entry of the dataset is a tuple containing:

- $frame_f$, $frame_b$: 1920×1080 frames from the frontal and bird's eye camera view, respectively;
- ID_e, $model_e$: identifiers of the entity (e) in the scene and of the vehicle's type;
- $frontal\_coords_e$, $birdeye\_coords_e$: the coordinates of the bounding box that encloses the entity;

	Total
Number of runs	300
Number of bounding boxes	1125187
Unique entity IDs	56454
Unique entity models	198

Table 4.1: Overview of the statistics on the collected SVA dataset. See text for details.



(a) (b)

Figure 4.9: Unnormalized distribution of vehicle orientation (a) and distances (b) present in the collected dataset. Distribution of angles conversely presents two prominent modes around 0°/360° and 180° respectively, due to the fact that the major part of vehicles encountered travel in parallel to the player's car, on the same (0/360°) or the opposite (180°) direction. Conversely, distance is almost uniformly distributed between 5 and 30 meters.

- $distance_e$, yaw_e : distance and rotation of the entity w.r.t. the player.

Fig. 4.9 shows the distributions of entity rotation and distance across the collected data.

4.2.3 Semantic-aware Dense Projection Network

At a first glance, the problem we address could be mistaken with a bare geometric warping between different views. Indeed, this is not the case since targets are not completely visible from the dashboard camera view and their dimensions in the bird's eye map depend on both the object visual appearance and semantic category (*e.g.* a truck is longer than a car).

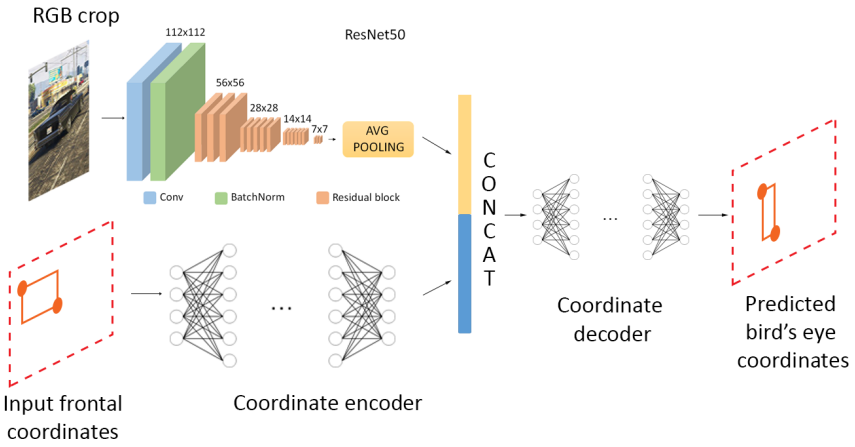


Figure 4.10: A graphical representation of the proposed `SDPN` (see Sec. 4.2.3). All layers contain *ReLU* units, except for the top decoder layer which employs *tanh* activation. The number of fully connected units is $(256, 256, 256)$ and $(1024, 1024, 512, 256, 128, 4)$ for the coordinate encoder and decoder respectively.

Additionally, it cannot be cast as a correspondence problem, since no bird's eye view information are available at test time. Conversely, we tackle the problem from a deep learning perspective: dashboard camera information are employed to learn a spatial occupancy map of the scene seen from above.

Our proposed architecture composes of two main branches, as depicted in Fig. 4.10. The first branch takes as input image crops of vehicles detected in the dashboard camera view. We extract deep representations by means of *ResNet50* deep network [69], taking advantage of pre-training for image recognition on ImageNet [40]. To this end we discard the top fully-connected dense layer which is tailored for the original classification task. This part of the model is able to extract semantic features from input images, even though it is unaware of the location of the bounding box in the scene.
Conversely, the second branch consists of a deep *Multi Layer Perceptron*

	IoU ↑	CD ↓	hE ↓	wE ↓	arE ↓
homo	0.13	191.8	0.28	0.34	0.38
grid	0.18	154.3	0.74	0.70	1.30
MLP	0.32	96.5	0.25	0.25	0.29
SDPN	**0.37**	**78.0**	**0.21**	**0.24**	**0.29**

Table 4.2: Table summarizing results of proposed SDPN model against the baselines.

(MLP), composed by 4 fully-connected layers, which is fed with bounding boxes coordinates (4 for each detection), learning to encode the input into a 256 dimensional feature space. Due to its input domain, this segment of the model is not aware of objects' semantic, and can only learn a spatial transformation between the two planes.
Both appearance features and encodings of bounding box coordinates are then merged through concatenation and undergo a further fully-connected decoder which predicts vehicles' locations in the bird's eye view. Since our model combines information about object's location with semantic hints on the content of the bounding box, we refer to it as *Semantic-aware Dense Projection Network* (SDPN in short).

Training Details: ImageNet [40] mean pixel value is subtracted from input crops, which are then resized to 224×224 before being fed to the network. During training, we freeze *ResNet50* parameters. Ground truth coordinates in the bird's eye view are normalized in range $[-1, 1]$. Dropout is applied after each fully-connected layer with drop probability 0.25. The whole model is trained end-to-end using *Mean Squared Error* as objective function and exploiting *Adam* [87] optimizer with the following parameters: $lr = 0.001, \beta_1 = 0.9, \beta_2 = 0.999$.

4.2.4 Experimental results

We now assess our proposal comparing its performance against some baselines. Due to the peculiar nature of the task, the choice of competitor models is not trivial.
To validate the choice of a learning perspective against a geometrical one, we introduce a first baseline model that employs a projective transforma-

tion to estimate a mapping between corresponding points in the two views. Such correspondences are collected from bottom corners of both source and target boxes in the training set, then used to estimate an homography matrix in a least-squares fashion (*e.g.* minimizing reprojection error). Since correspondences mostly belong to the street, which is a planar region, the choice of the projective transformation seems reasonable. The height of the target box, however, cannot be recovered from the projection, thus it is cast as the average height among training examples. We refer to this model as *homography model*.
Additionally, we design second baseline by quantizing spatial locations in both views in a regular grid, and learn point mappings in a probabilistic fashion. For each cell G_i^f in the frontal view grid, a probability distribution is estimated over bird's eye grid cells G_j^b, encoding the probability of a pixel belonging to G_i^f to fall in the cell G_j^b. During training, top-left and bottom-right bounding box corners in both views are used to update such densities. At prediction stage, given a test point p_k which lies in cell G_i^f we predict destination point by sampling from the corresponding cell distribution. We fix grid resolution to 108x192, meaning a 10x quantization along both axes, and refer to this baseline as *grid model*.
It could be questioned if the appearance of the bounding box content in the frontal view is needed at all in estimating the target coordinates, given sufficient training data and an enough powerful model. In order to determine the importance of the visual input in the process of estimating the bird's eye occupancy map, we also train an additional model with approximately the same number of trainable parameters of our proposed model `SDPN`, but fully connected from input to output coordinates. We refer to this last baseline as *MLP*.
For comparison, we rely on three metrics:

- *Intersection over Union* (IoU): measure of the quality of the predicted bounding box BB_p with respect to the target BB_t:

$$IoU(BB_p, BB_t) = \frac{A(BB_p \cap BB_t)}{A(BB_p \cup BB_t)}$$

  where $A(R)$ refers to the area of the rectangle R;

- *Centroid Distance* (CD): distance in pixels between box centers, as

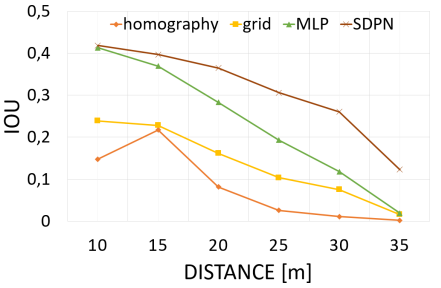


Figure 4.11: Degradation of Intersection over Union (IoU) performance as the distance to the detected vehicle increases.

an indicator of localization quality[1];

- *Height, Width Error* (hE,wE): average error on bounding box height and width respectively, expressed in percentage w.r.t. the ground truth BB_t size;
- *Aspect ratio mean Error* (arE): absolute difference in aspect ratio between BB_p and BB_t:

$$arE = \left| \frac{BB_p.w}{BB_p.h} - \frac{BB_t.w}{BB_t.h} \right| \tag{4.9}$$

The evaluation of baselines and proposed model is reported in Fig. 4.11 (a). Results suggest that both *homography* and *grid* are too naive to capture the complexity of the task and fail in properly warping vehicles into the bird's eye view. In particular, *grid* baseline performs poorly as it only models a point-wise transformation between bounding box corners, disregarding information about the overall input bounding box size. On the contrary, MLP processes the bounding box in its whole and provides a reasonable estimation. However, it still misses the chance to properly recover the length of the bounding box in the bird's eye view, being unaware of entity's visual appearance. Instead, `SDPN` is able to capture the object's semantic, which is a primary cue for correctly inferring vehicle's location and shape

[1]Please recall that images are 1920x1080 pixel size.

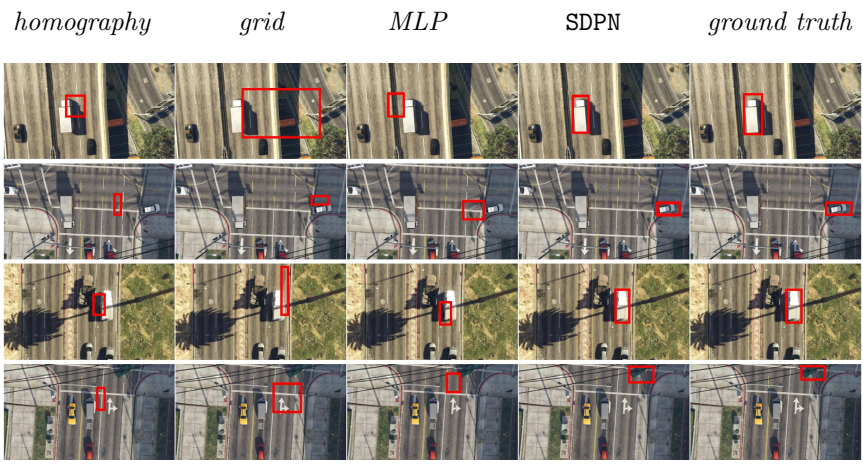


Figure 4.12: Visual comparison between different models. Baselines often predict reasonable locations for the bounding boxes. SDPN is also able to learn the orientation and type of the vehicle (*e.g.* a truck is bigger than a car etc.).

in the target view.
A second experiment investigates how vehicle's distance affects the warping accuracy. Fig. 4.11 (b) highlights that all the models' performance degrades as the distance of target vehicles increases. Indeed, closer examples exhibit lower variance (*e.g.* are mostly related to the car ahead and the ones approaching from the opposite direction) and thus are easier to model. However, it can be noticed that moving forward along distance axis the gap between the SDPN and MLP gets wider. This suggests that the additional visual input adds robustness in these challenging situations. We refer the reader to Fig. 4.12 for a qualitative comparison.

A real-world case study

In order to judge the capability of our model to generalize on real-world data, we test it using authentic driving videos taken from a roof-mounted camera [4]. We rely on state-of-the-art detector [103] to get the bounding boxes of vehicles in the frontal view. As the ground truth is not available for

Figure 4.13: Visual results on real-world examples. Predictions look reasonable even if the whole training was conducted on synthetic data.

these sequences, performance is difficult to quantify precisely. Nonetheless, we show qualitative results in Fig. 4.13: it can be appreciated how the network is able to correctly localize other vehicles' positions, despite having been trained exclusively on synthetic data.
SDPN can perform inference at approximately $100Hz$ on a NVIDIA TitanX GPU, which demonstrates the suitability of our model for being integrated in an actual assisted or autonomous driving pipeline.

Concluding remarks

In this section we presented two main contributions. A new high-quality synthetic dataset, featuring a huge amount of dashboard camera and bird's eye frames, in which the spatial occupancy of a variety of vehicles (i.e. bounding boxes, distance, yaw) is annotated. Furthermore, we presented a deep learning based model to tackle the problem of mapping detections onto a different view of the scene. We argue that these maps could be useful in an assisted driving context, in order to facilitate driver's decisions by making available in one place a concise representation of the road state. Furthermore, in an autonomous driving scenario, inferred vehicle positions could be integrated with other sensory data such as radar or lidar by means of *e.g.* a Kalman filter to reduce overall uncertainty.

4.3 End-to-end 6-DoF Object Pose Estimation through Differentiable Rasterization

In the previous chapter we saw how a convolutional neural network can be trained to infer the bird's eye occupancy map of the scene given a single frame from the vehicle dashboard camera. Here we take a complementary approach, proposing a novel model for estimating the pose of the vehicles in the scene given a minimal amount of monocular visual information.
Before diving deep into this topic, it might be worth spending a few words on the challenge of this task. Image formation is essentially a lossy process, as during the perspective projection we lose a lot of information about the 3D structure of the captured scene. For this reason, inferring the six degrees of freedom (6-DoF) pose (3D rotations + 3D translations) of an object given a single RGB image is extremely challenging. Estimating the pose requires the distillation of a lot of information from a single frame; the object 3D structure, as well as the 3D roto-translation that leads to visually plausible outputs must be inferred jointly.

Code related to this project is open-source at:
https://github.com/ndrplz/tensorflow-mesh-renderer.

4.3.1 Differentiable rendering for 6-DoF pose estimation

From a high level view, a renderer can be thought as a black-box receiving two inputs and producing one output. It takes as input (i) a given representation of the 3D object (e.g. voxels, mesh etc.) and (ii) the 6-DoF pose of the object w.r.t. the camera and produces the 2D image of the object or, as in our setting, solely its silhouette.
Typically a rendering algorithm includes many non-differentiable operations (e.g. rounding, hard assignments etc.), preventing it to be used in a deep learning architecture as it would break the back-propagation chain. Nonetheless, in the context of 3D volume estimation recent works [78, 217, 109, 55, 189] have been proposed which exploit approximated differentiable renderers to back-propagate the loss to the first renderer input, namely the 3D representation of the object, but leaving fixed the set of possible camera poses.

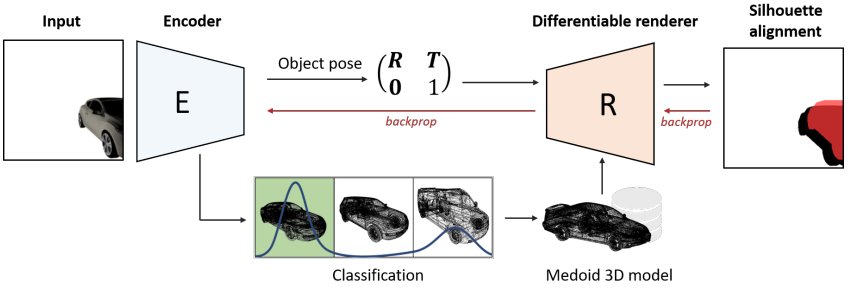


Figure 4.14: The overall proposed framework. A deep convolutional encoder is fed with the object mask and predicts both the object's class and 6-DoF pose. By means of a differentiable renderer the predicted cluster medoid can be projected back according to the predicted pose, adding a further online alignment supervision w.r.t. the input mask.

In this work, we propose to leverage a differentiable renderer to re-project the 3D object model on the image according to the pose predicted by a convolutional encoder (see Fig. 4.14). The alignment error between the observed and the re-projected object silhouette can then be measured and - being the renderer differentiable - back-propagated through it to the encoder, correcting the estimated pose. We demonstrate that this differentiable block can be stacked on a 6-DoF pose estimator to significantly refine the estimated pose using only the 2D alignment information between the input object mask and the rendered silhouette. Notably, this can also happen iteratively at inference time, in an online learning fashion.
The convolutional encoder produces a coarse classification of the object to profitably re-project a representative model of the predicted class (i.e. a medoid) instead of the exact 3D model of the object - as we reckon that the exact 3D model is hardly available in a real setting. Experimental results show that the proposed pipeline is able to correct the estimated pose effectively even when using surrogate models.

4.3.2 Model architecture

We design our model as composed of two main components: i) the convolutional encoder, responsible for classifying the object and performing a first

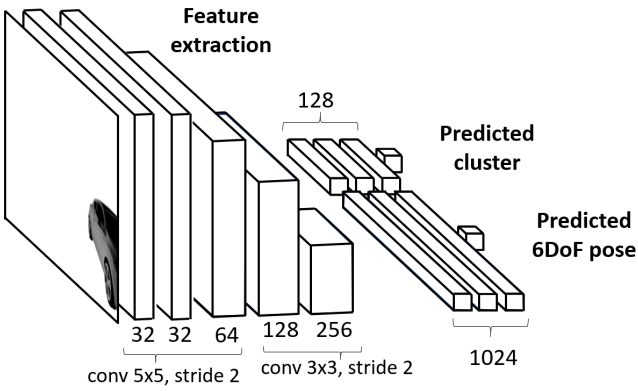


Figure 4.15: Architecture of the encoder network. Visual features are extracted from the input image by means of 2D convolutions (first three layers have 5x5 kernel, last two have 3x3 kernel. All convolutional layers have stride 2 and are followed by leaky ReLu non-linearities). The flattened feature vector is fed to two fully connected branch, which estimate the object class and pose respectively.

estimation of its pose; and ii) the differentiable renderer introduced above. In this section we provide details on both modules.

Convolutional encoder for pose and class estimation

The deep convolutional encoder network is schematized in Fig. 4.15. The first part of the network is dedicated to feature extraction and it is shared by the classification and the pose estimation branch. The network has been designed inspired by [177] which showed favorable results in a related task. Features extracted are then used by two fully-connected independent branches to infer the object class and the camera pose respectively. All layers but the last are followed by leaky ReLu activation with $\alpha = 0.2$. Differently from most of the literature [217, 55, 205] we do not quantize the pose space into a discrete set of pre-defined poses to ease the task. Conversely, given a rotation matrix $\mathbf{R}_{3\times3}$ and a translation vector $\mathbf{t}_{3\times1}$ we regress the object pose

$$\mathbf{P}_{3\times4} = \begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix} \tag{4.10}$$

by optimizing the mean square error between the predicted and the true pose:

$$\mathcal{L}(\mathcal{X}, \mathcal{Y}_p, \theta) = \frac{1}{N} \sum_i ||y_i - f_p(x_i, \theta)||^2 \quad x_i \in \mathcal{X}, y_i \in \mathcal{Y}_p \tag{4.11}$$

where $\mathcal{X}$ is the set of RGB images, $\mathcal{Y}_p$ is the set of true $\mathbf{P}_{3\times4}$ pose matrices and $f_p(x_i, \theta)$ is the pose predicted by the encoder for example x_i according to its weights θ. From a technical standpoint, for each X, Y, Z axis the encoder regresses the cosine of the Euler rotation angle and the respective translation. The output roto-translation matrix is then composed following Euler ZYX convention: in this way predicted matrices are guaranteed to be always geometrically consistent. For the classification branch we instead optimize the following categorical cross-entropy function:

$$\mathcal{L}(\mathcal{X}, \mathcal{Y}_c, \theta) = -\frac{1}{N} \sum_i y_i \log f_c(x_i, \theta) \tag{4.12}$$

where $x_i \in \mathcal{X}$ is an input RGB image, $f_c(x_i, \theta)$ is the encoder predicted distribution over possible clusters for example x_i and y_i in the true one-hot distribution for example x_i.

Differentiable Renderer

To measure the reliability of the predicted 6-DoF pose and to be able to correct it at test time, we design a fully differentiable renderer for re-projecting the silhouette of the 3D model on the image according to the predicted object pose. This allows to refine the estimated pose by back-propagating the alignment error between the 2D silhouettes. To the best of our knowledge, it is the first time that a fully-differentiable raster-based renderer is used to this purpose. Differently from concurrent works such as [217], our rendering process starts from the raw mesh triangles and not from a 3D voxel representation. While the latter is easier to predict by a neural network since it has a static shape, its footprint scales with the cube of the resolution and forces to use ray-tracing techniques to render the final image, known to be slow and harder to parallelize. Despite rastering does not allow for photo-realistic shaded images, as it does not imply light sources rays tracing, it is still well suited for all tasks which require the object shape silhouette from different point of views as in our case.
Our renderer is composed of two main parts:

- A *rastering algorithm*, which applies the predicted camera to the 3D triangles meshes to obtain 2D projected floating point coordinates of the corners;
- An *in/out test* to determine which projected points lie inside the triangles, *i.e.* which triangles must be filled.

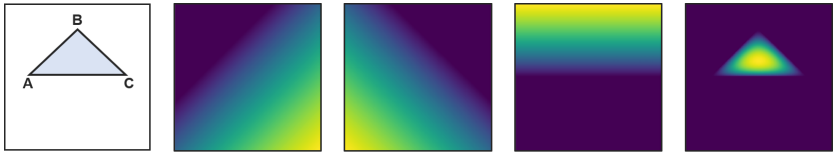


Figure 4.16: Visual explanation of the proposed approximated rastering process. First each triangle composing the mesh is projected in the 2D image (a) using Eq. 4.13. The determinant product inside the max of Eq. 4.14 selects the points which lie on the left side of each edge of the triangle (b), (c), (d). The product of these three terms gives an approximated yet differentiable rendering of the triangle's silhouette (e).

While the first step is fully differentiable, a naive implementation of the latter exploits boolean masks to select the pixels to be filled, which eventually breaks the backpropagation through the network. Inspired by [78], we employed a spatial transformation to assign a value to each pixel based on a relation between its coordinates and those of the triangles corners. While a boolean mask represents hard membership, this approach assigns each pixels a continuous value, thus applying a soft (differentiable) membership. From a more technical standpoint, given all triangles T which compose the mesh of current model, we project the 3D triangle vertices V_{3D} as follows:

$$\begin{bmatrix} V_{2D} \\ 1 \end{bmatrix} = \begin{bmatrix} x/z \\ y/z \\ 1 \end{bmatrix} = \mathbf{K}_{3\times3}\mathbf{P}_{3\times4}^{-1} \begin{bmatrix} V_{3D} \\ 1 \end{bmatrix} \tag{4.13}$$

where $\mathbf{K}_{3x3}$ is the camera calibration matrix and $\mathbf{P}_{3\times4}^{-1}$.
Then, defined as $E^{(i)} = [(v_1, v_0), (v_2, v_1), (v_0, v_2)]$ the three edges of the *i-th* projected triangle, the renderer's output for pixel in location (u, v) can be

computed as:

$$g_{u,v} = \sum_{i}^{T} F_{norm} \left(\prod_{(v_j, v_k)}^{E^{(i)}} \max \left(\begin{vmatrix} v_j - v_k \\ v_j - (u,v) \end{vmatrix} \begin{vmatrix} v_1 - v_0 \\ v_2 - v_1 \end{vmatrix}, 0 \right) \right), \quad (u,v) \in H \times W$$

$$where \quad F_{norm}(x) = \tanh \frac{x - min(x)}{max(x) - min(x)} \tag{4.14}$$

and H, W indicate the image height and width in pixels. We refer the reader to Fig. 4.16 for a better intuition of Equation 4.14. It is worth noticing that the *i-th* triangle contributes to the output only if all the three determinant products are positive, meaning that (u, v) point lies on the left side of all three triangle edges *i.e.* it is inside the triangle.

4.3.3 Experimental results

Dataset

We train our model on ShapeNetCore(v2) [23] dataset, which comprises more than 50K unique 3D models from 55 distinct man-made objects. We focus in particular on the car synset since it is one of the most populated category with 7497 different 3D CAD vehicle models. Each model is stored in .obj format along with its materials and textures: dimensions, number of vertices and details vary greatly from one model another.

Data collection To collect the data, we first load a random model on the origin $\mathbf{t} = (0, 0, 0)$ of our reference system. We then create a camera in location $\mathbf{t} = (x, y, z)$. While on xy plane the location is randomly sampled in a $q_x \times q_y$ grid, we keep fixed $z = k$ under the assumption that the camera is mounted somewhere at height k on a moving agent (e.g. an unmanned vehicle). We then force the camera to point an empty object e that is randomly sampled at $z = 0$ and x, y sampled as above in a $e_x \times e_y$ grid: in this way we make the object to appear translated in the camera image. Eventually, the camera image is dumped along with the camera pose to constitute an example x_i. We refer the reader to Fig. 4.17 to get a better insight into the procedure. *Data collection details:* In our experiments we set $q_x = q_y = 10$ and $k = 1.5$, which is the average height of a European vehicle. For the empty object we set $e_x = e_y = 3$. Models are standardized

s.t. the major dimension has length 6. For each cluster, the models are split with ratio 0.6-0.2-0.2 into train, validation and test set respectively. Medoids are expected to be known at test-time and do not belong to any of the splits. Models are rendered using Blender CYCLES engine [11] to maximize photo-realism.

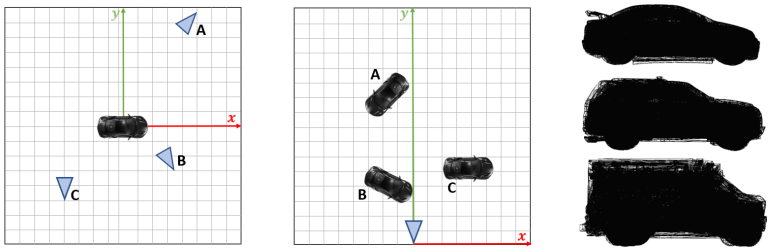


Figure 4.17: On the left is depicted how all camera poses predicted by the encoder independently for each object (a) can be roto-translated to a common origin to reconstruct the overall scene (b), also in Fig. 4.20. On the right, the average silhouette of vehicles belonging to *sedan*, *SUV* and *cargo* is shown (c). For each cluster all 3D meshes are overlaid before taking the snapshot from the side view; the high overlap highlights the low intra-cluster variance.

Selecting the representative 3D model Since the true 3D object model is hardly available at test time, we want to verify if a surrogate 3D model can be instead successfully employed for the rendering process. Analogously to Du *et al.* [44] we distinguish three main vehicle clusters, namely i) *Sedan* passenger cars, ii) *Sport-utility vehicles* (SUV, which are also passenger cars but have off-road features like raised ground clearance) and iii) *Cargo* vehicles such as trucks and ambulances. Following Tatarchenko *et al.* [177] we selected the representative model for each cluster, by extracting and comparing the HOG descriptors from two standard rendered views of each CAD model (i.e. frontal and side). The low intra-cluster variance can be appreciated in Fig. 4.17(c). Eventually we compute the $L2$ distance between descriptors and for each cluster we retain the cluster medoid, *i.e.* the model with the least average distance from all the others.

Table 4.3: Table summarizing model performance. It is worth noticing that none of the metrics in the table is explicitly optimized during refinement. Results of concurrent works on the vehicle class are shown for reference, despite the task of [188, 172] is only viewpoint estimation (not 6-DoF pose) and all are trained on different dataset.

Model	**Accuracy** ↑	**mIoU** ↑	**MVE** ↓	**Acc**$_{\frac{\pi}{6}}$ ↑
encoder	0.89	0.59	5.7	0.86
encoder+refinement	**0.89**	**0.72**	**4.5**	**0.90**
Pavlakos *et al.* [136]	-	-	6.9	-
Tulsiani and Malik [188]	-	-	9.1	0.89
Su *et al.* [172]	-	-	6.0	0.88

Model Evaluation

Metrics The encoder ability to estimate the 3D pose of the object is measured by means of geodesic distance between predicted and true rotation matrix [188, 75] as:

$$\Delta(\mathbf{R}_{true}, \mathbf{R}_{pred}) = \frac{||\log(\mathbf{R}_{true}^T \mathbf{R}_{pred})||_F}{\sqrt{2}} \tag{4.15}$$

where $||\mathbf{A}||_F = \sqrt{\sum_{i,j} |a_{ij}|^2}$ indicates the Frobenius norm. In particular, we report the median value of the aforementioned distance over all predictions in test set as Median Viewpoint Error (MVE). We also report the percentage of examples in which the pose rotation error is smaller than $\pi/6$ as $Acc_{\frac{\pi}{6}}$. To measure the re-projection alignment error we instead rely on mean intersection over union (mIoU) metric defined over the N test examples as $\frac{1}{N}\sum_i \frac{S_i \cap \tilde{S}_i}{S_i \cup \tilde{S}_i} \quad i = 1, \ldots, N$:

where S_i is the ground truth silhouette and $\tilde{S}_i = g(f_p(x_i), f_c(x_i), \mathbf{K})$ is the renderer output given the predicted object pose, cluster and camera intrinsics $\mathbf{K}$.

Model performance To prove the effectiveness of the proposed method we first train the 6-DoF pose estimation network alone to jointly estimate the object class and its 6-DoF pose. In this way, we get a baseline to measure the successive contribute of the prediction refinement through

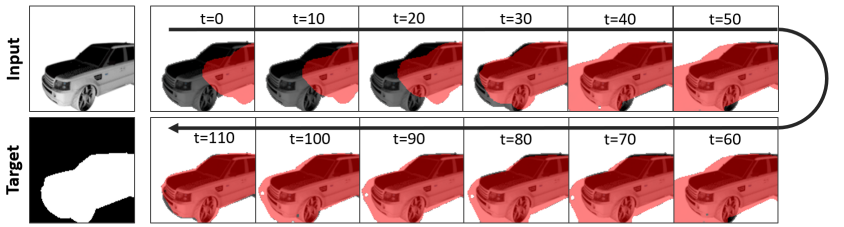


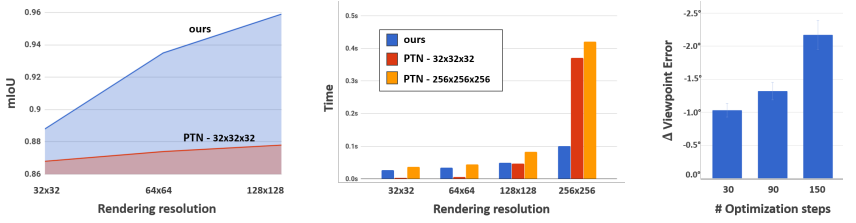
Figure 4.18: Online refinement of the estimated pose; We overlay in red the predicted silhouette for each optimization step. Despite the initial estimate (t=0) was noticeably wrong, the 6-DoF object pose is gradually corrected using only 2D silhouette alignment information.

our differentiable rendering module. State-of-the-art results on test set reported in Table 4.3(first row) indicate this to be already a strong baseline. The prediction refinement module is then plugged-in, and the evaluation is repeated. For each example, the medoid of the predicted class is rendered according to the predicted pose, back-propagating the alignment error between the true and the rendered silhouette for 30 optimization steps. Results of this analysis are reported in Table 4.3(second row) and indicate a huge performance gain (20%) obtainable by maximizing the 2D alignment between object masks. The significant improvement in all the metrics, despite none of these is optimized explicitly, suggests that the proposed differentiable rendering module is a viable solution for refining the predicted 6-DoF even at test time, requiring minimal information (*i.e.* only the object mask). The process of prediction refinement can be appreciated in Fig. 4.18.

Renderer ablation study We measure, at first, the impact of rendering resolution on the optimization process by refining the object 6-DoF estimated pose using different rendering resolutions. Results reported in Table 4.4 show that working at higher resolution is definitely helpful while very-low resolution are hardly beneficial, if not detrimental, for the optimization process. This supports the need to abandon the voxel-based representation, whose computational footprint increases with the cube of resolution. We then compare our renderer with the publicly available implementation of Perspective Transformer Network (PTN) by Yan *et al.* [217]. Results are shown in Fig. 4.19(a). Since PTN relies on a fixed 32x32x32 voxel

Table 4.4: Gains obtained in pose estimation using different rendering resolutions. Increasing the resolution used for rendering the silhouette is much beneficial to the optimization process. Conversely, for very low resolution this phase is hardly helpful. ΔV and ΔT indicate viewpoint and translation error respectively.

Renderer Resolution	**Δ IoU ↑**	**ΔV Error ↓**	**ΔT Error ↓**
16x16	+0.00	+0.15	+0.02
32x32	+0.03	-0.26	+0.00
64x64	+0.05	-0.57	+0.00
128x128	+0.11	-1.03	-0.01
256x256	+0.13	-1.29	-0.03



(a) (b) (c)

Figure 4.19: (a) Intersection over union between rendered silhouette and the ground truth one for both our renderer and Perspective Transformer Networks (PTN) [217], at different rendering resolutions. (b) Rendering time for different image (and PTN voxel) resolutions. (c) Average viewpoint error improvement for different number of optimization steps. See text for details.

representation, rendering at higher resolution hardly changes the output's fidelity w.r.t. the true silhouette. Conversely, our mesh-based renderer is able to effectively take advantage of the higher resolution. Comparing our rendering time with PTN [217] in Fig. 4.19(b), we see that PTN scores favorably only for very-low voxel and image resolutions, while as resolution increases the PTN rendering time increases exponentially due to the voxel-based representation. Eventually, in Fig. 4.19(c) we show that our average viewpoint error continues to decrease along with the number

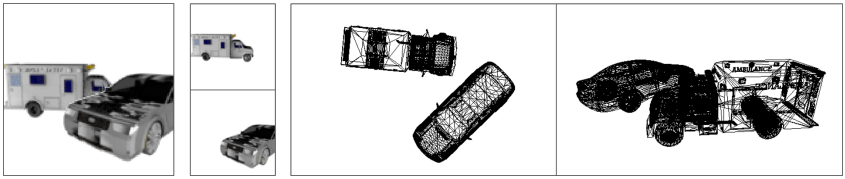


Figure 4.20: Visual results for scenes with multiple objects. Since all predicted poses lie in the same reference system (see Fig. 4.17), different views of the scene can be produced by means of any rendering engine. It is worth noticing that each object has been substituted by the representative model for its predicted class.

of refinement optimization steps.

Training details Encoder is trained until convergence with batch size=64 and ADAM optimizer with learning rate 10^{-5} (other hyper-parameters as suggested in the original paper [87]). Batch size is decreased to 20 and learning rate to 10^{-6} during renderer fine-tuning. We find useful dropout ($p = 0.5$) after all dense layers and $L2$ weight decay over feature extraction for regularization purposes.

4.3.4 Details on the differentiable renderer

Number of triangles used for rendering

Our approximated rendered abandons the voxel-based representation and makes instead use of the 3D triangle mesh of the object. However, even if a 3D mesh is often composed of thousands of triangles, we found that there was no need to render all the triangles to produce a good render of the object silhouette. Figure 4.21 shows how the rendering output is affected by the number of triangles used. In this work all experiments are performed using only the 1000 widest-area triangle of each mesh (another sub-sampling strategy could have been to increase the density of triangles towards the edges, leaving a sparser mesh on the inside of the object). These heuristic allowed to save precious running time while maintaining a good output quality.

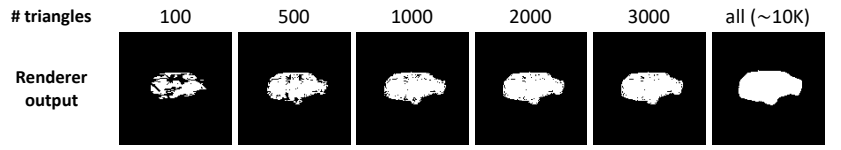


Figure 4.21: Influence of triangle mesh down-sampling on the renderer output.

Output post-processing

Renderer output formation is described in Sec. 3.2 of the paper; in particular, Eq. 5 which shows how the output is computed for each pixel (u, v) in the image. Since the value of each pixel is proportional to the number of triangles which lie over it once projected into the image, nearby pixels can have significantly different values (see *raw output* in Fig. 4.22). From an implementation standpoint, we found useful to post-process the renderer output to make it homogeneous, moving all positive pixels towards 1. Since we could not perform hard threshold to keep the output back-propagable, we found as the easiest solution was to train apart a nano-CNN to perform only this post-processing. Being $\mathrm{c}N_{k\times k}$ a convolutional layers with N filters and squared kernel of side k followed by ReLu activation, the architecture of this nano-CNN follows: $\mathrm{c10}_{5\times5}$, $\mathrm{c10}_{5\times5}$, $\mathrm{c10}_{3\times3}$, $\mathrm{c1}_{3\times3}$. It is worth noticing that this nano-CNN is trained apart once and then it is used only in inference mode to post-process the renderer output: thus the renderer has still zero learnable parameters during the pose optimization phase.

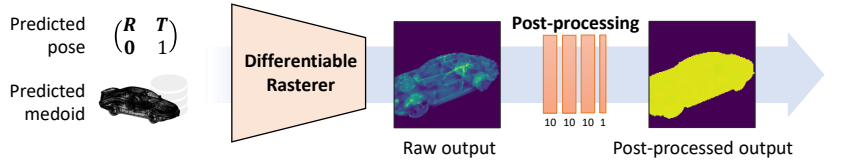


Figure 4.22: Visualization of renderer output post-processing. See text for details.

4.3.5 Concluding remarks

In this work we introduce a 6-DoF pose estimation framework which allows an *online* refinement of the predicted pose from minimal 2D information (*i.e.* the object mask). A fully differentiable raster-based renderer is developed for re-projecting the object silhouette on the image according to the predicted 6-DoF pose: this allows to correct the predicted pose by simply back-propagating the alignment error between the observed and the rendered silhouette. Experimental results indicate i) the overall effectiveness of the online optimization phase, ii) that proxy representative models can be profitably used in place of the true ones in case these are not available and iii) the benefit of working in higher resolution, well-handled by our raster-based renderer but hardly managed by concurrent ray-tracing, voxel-based algorithms. While these results are encouraging, additional efforts must be spent for applying the proposed model in a real-world setting, where the vehicles must also segmented from a possibly cluttered background.

4.4 Warp and Learn: Generating Novel Views of the Urban Scene

A smart infrastructure has to solve a number of challenging problems towards a deep understanding of the urban scene. In the last chapters we already introduced and proposed novel models to tackle some of those, such as vehicle re-identification (Sec. 4.1) and registration of vehicle and infrastructure viewpoints (Sec. 4.2, Sec. 4.3).

Here we make a step forward and we address the more ambitious goal of generating the visual appearance of the whole urban scene in the future. Indeed, being able to generate novel views from arbitrary virtual cameras in an urban scene promises to have huge impacts is many domains: surveillance, vehicle re-identification and forensics to mention a few. Since the topic is extremely broad we make a few assumptions not to be overwhelmed by the ill-definedness of the problem. In particular, we consider vehicles the only agents in the scene, ignoring other classes of moving objects such as pedestrian or bicycles. Finding a solution to the problem under this constraint would still be extremely interesting, due to vehicles ubiquity in urban scene understanding applications [220, 181, 49, 160, 116, 104].

Figure 4.23: We propose a semi-parametric framework to generate realistic novel views of a vehicle and / or to transfer its appearance to different models.

Code, data and multimedia related to this project are available at:
https://github.com/ndrplz/semiparametric

4.4.1 Inferring the visual aspect of vehicles from novel viewpoints

How would you see an object from another point of view? Given a single view of an object in the world, predicting how it would look like from arbitrarily different viewpoints is definitely non-trivial for both humans and machines. Still, people with a good *visual-spatial intelligence* [58] can easily imagine objects' rotation, zoom and translation shifts, especially if objects have a well known shape and feature some degree of symmetry. Indeed, humans have been shown to perform mental transformations for decision-taking about their surrounding environment [161, 20, 223].
Machines are still far from this level of intelligence. Still, powerful *parametric* (i.e. entirely learning-based) deep learning models [88, 64] made it possible to frame the generation of novel viewpoints as a *conditioned image synthesis* problem. However, this is an holistic approach that under-exploits the fact that man-made objects 3D models are roughly distributed according to few prototypes (e.g. sedan, VAN, pick-up, truck etc. for vehicles). Up

to now, the generation is constrained to be visually plausible with almost no geometric support from prototypes' shapes [177, 134]. Furthermore, even though generated images may look realistic *per se*, fine-grained visual appearance characterizing the particular object instance (e.g. texture) is often lost due to its high frequency which hardly survives being encoded through a deep network [177, 238, 46]. Eventually, most methods are supervised on the target image, which is seldom available for real data. Also, vast amount of data are required for the network to generalize to arbitrary transformations (i.e. a sufficient number of images for every possible viewpoint). This constrains many methods to be trained solely on synthetic data and to be restricted to a discrete set of viewpoints: in fact, most recent works only handle a small set of transformations (e.g. rotating around the object at constant radius) [218, 177, 234, 134].

At the same time, an independent line of research has shown that a *non-parametric* approach can be a viable path for photorealism, as also pointed out by Qi *et al.* [143]. For instance, new images can be generated by collaging [67, 93, 29, 82, 76] or by leveraging multiple photographs to synthesize novel views via image-based rendering [25, 24, 127, 70, 128]. Still, these methods require a large amount of data at test time: entire image banks for collaging, multiple photographs and depth data for image-based rendering.

Here we propose a new approach - inherently *semi-parametric* - being based on both learning and geometry, self-supervised and efficient to be used in real-time. By taking the best from both worlds, we exploit geometric constraints to roughly sketch the target shape of the object and its textures while still relying on deep view synthesis to refine the generated view. The rationale behind this work is that many man-made objects adhere to a-priori geometric rules: vehicles in particular, exhibit a symmetric, piece-wise planar structure. Therefore, those properties may be exploited to approximately represent them by a small set of piece-wise planar patches, which can be warped almost exactly from source to destination viewpoint via a symmetry-aware homography transformation. Although these warped patches provide a rich hint about the visual content of the target viewpoint, they are far from being useful on their own. Thus, a fully-convolutional network is seeded with these patches along with a 2.5D CAD-rendered sketch to be used as guidance; it is then trained in a self-supervised manner to discriminate which part of the image must be completed or in-painted for the result to look realistic (see Fig. 4.14).

We tested our solution in particular in vehicle generation due to their ubiquity in urban scene understanding applications [220, 181, 49, 160, 116, 104]. Moreover, to highlight that a decomposition in planar patches holds for different types of rigid objects, we evaluate our semi-parametric framework on both convex objects (*vehicles*) and concave ones (*chairs*). We leave as future work the analysis of a broader set of object categories.

In summary, our main contributions in this topic follow:

- We propose an original formulation of the problem of object novel viewpoint synthesis in a semi-parametric setting. Loose geometrical assumptions about the object shape provide rich hints about its appearance (*non-parametric*); this information guides a fully-convolutional network (*parametric*) in the generation process.

- We design our model to be trainable on existing datasets for 3D object detection in a *self-supervised* manner, with no need for paired source/target viewpoint images. Furthermore, we leverage 2D keypoints for real-world images where foreground segmentation is not provided.

- We demonstrate how our method excels in preserving visual details (e.g. texture) and in performing realistic shape transfer to completely different 3D models, while still being resilient to a much wider range of 3D transformations than competitors.

Our method can be employed to generate realistic novel views of an object from an arbitrary zoom, viewpoint and distance, as depicted in Figures 4.23, 4.26, 4.28, 4.29, 4.31, 4.33, 4.34. Also, our approach allows a disentangled editing of object shape and appearance (i.e. shape can be changed while preserving appearance or the other way around). This enables applications in interactive 3D manipulation and design, as well as data augmentation (Fig. 4.35).

A thorough experimental analysis is conducted comparing our proposal with state-of-the-art methods, considering both the quantitative and the perceptual point of view.

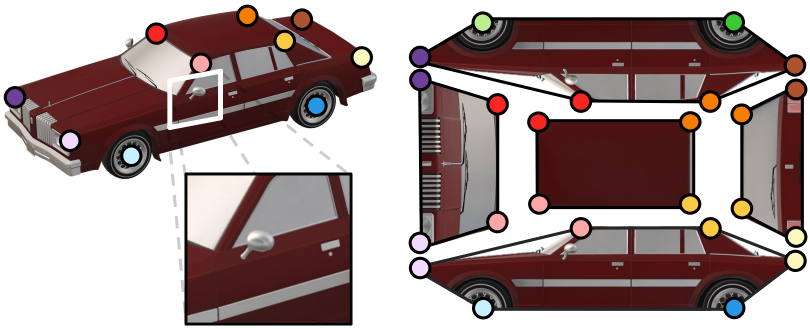


Figure 4.24: We model a rigid object with a small of piece-wise planar patches, whose vertices are defined by 2D keypoints. We also include a small central crop as appearance prior to carry low-frequency information.

4.4.2 Semi-parametric model architecture

Our model generates novel views of objects in a semi-parametric setting - relying on both geometry and learning. To this end, 2D keypoints and additional 3D information are extracted from a single view of the object. Keypoints are used as a proxy to describe 2D geometrical abstractions of the 3D shape (i.e. planar patches), which are transformed to the novel viewpoint. Eventually, a convolutional neural network seamlessly fuses this prior information to generate a realistic image from the novel view.
More into details, our model takes as input an image depicting a single object $\mathbf{x}_s \in \mathbb{R}^{H \times W \times 3}$ viewed from the source viewpoint $\mathbf{V}_s \in \mathbb{R}^{4 \times 4}$, its 2D keypoints $\boldsymbol{\mathcal{K}}_s$ and its associated 3D CAD model $C \in \mathbb{R}^{faces \times 3 \times 3}$ having 3D keypoints $\boldsymbol{\mathcal{K}}_{3D}$.
Training (Fig. 4.25, top) is performed in a *self-supervised* fashion maximizing the consistency between the input image and the generated one when projected onto the source viewpoint, with no need for coupled images from the two viewpoints as supervision. Given $\mathbf{x}_s$ and $\boldsymbol{\mathcal{K}}_s$, planar patches are extracted (Sec. 4.4.2). The patches are then projected to the target viewpoint $\mathbf{V}_d$ through an intermediate view (Sec. 4.4.2) according to a visibility model (Sec. 4.4.2). The 3D model C is also rendered from the target viewpoint $\mathbf{V}_d$ to get a 2.5D sketch of the object (Sec. 4.4.2). Eventually, the image completion network (ICN) starts from these visual seeds to generate a realistic final image (Sec. 4.4.2, 4.4.2).

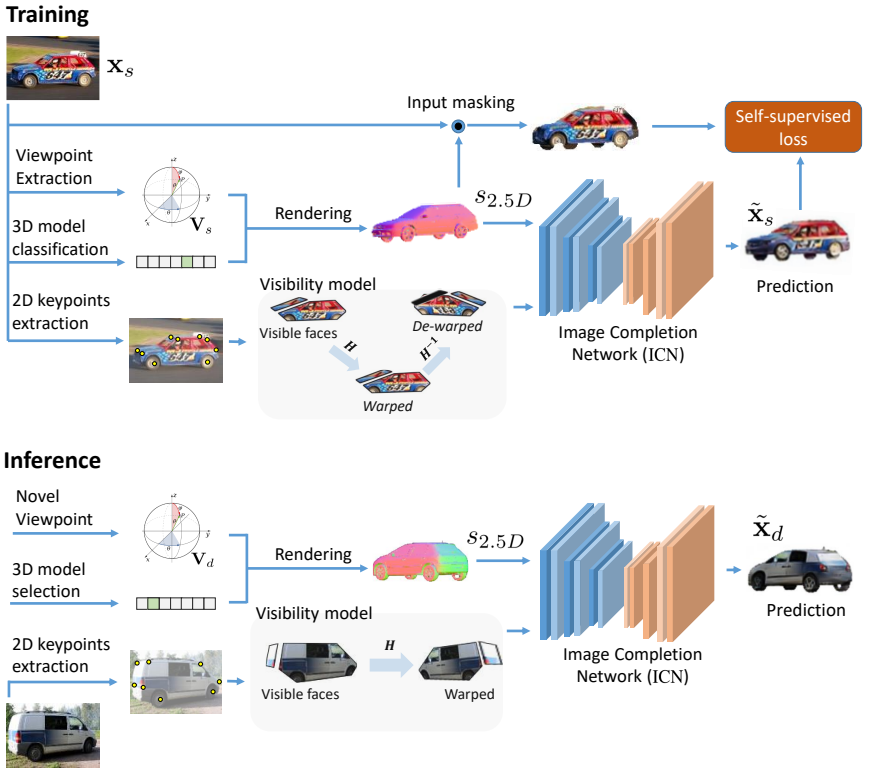


Figure 4.25: Model architecture overview. Approximately planar patches are extracted from the 2D keypoints locations. The Image Completion Network (ICN) uses the synthetic 2.5D sketches as templates to reconstruct object's appearance from the patches in a self-supervised fashion. During training, input patches are warped forth and back to a randomly sampled viewpoint to enforce resilience against homography issues that are likely to be encountered at test time. During inference, novel views of the input object are synthesised by providing the ICN a novel viewpoint and a (possibly different) rendered 3D model to be used as shape guideline.

Inference (Fig. 4.25, bottom) follows a similar flow. However, in this case only 2D keypoints $\boldsymbol{\mathcal{K}}_s$ and source viewpoint $\mathbf{V}_s$ are needed; the 3D model can be either inferred from the input or arbitrarily selected to perform shape transfer.

Without loss of generality, in this work we rely on ground truth data whenever possible, as our focus lies on the overall viewpoint generation pipeline. Off-the-shelf detectors [68, 136, 188] can be used to provide these information in an in-the-wild scenario.

Keypoint-based decomposition into planar patches

We leverage 2D keypoints to approximate the visible shape of the object with a simple polyhedron with a small set of faces, as depicted in Fig. 4.24. Since keypoints mark characteristic locations in the object shape (e.g. corners), a face defined from at least three of those could carry a perceptual / semantic meaning (e.g. the roof of a car). Exploiting 2D keypoints to find object faces is appealing for a number of reasons. First, this makes straightforward computing the homography matrix between planes in different viewpoints (see Sec. 4.4.2). Furthermore, a number of datasets provide object landmark annotations in real-world scenarios (e.g. [98, 215, 214, 203, 210, 198]) and solid keypoints detection methods exist [188, 136, 68].
Specifically, for each source image $\mathbf{x}_s \in \mathbb{R}^{H \times W \times 3}$ an array of 2D keypoints $\boldsymbol{\mathcal{K}}_s \in \mathbb{R}^{|\boldsymbol{\mathcal{K}}_s| \times 2}$ is available, being $|\boldsymbol{\mathcal{K}}_s|$ the category-specific number of keypoints (we set $|\boldsymbol{\mathcal{K}}_s| = 12$ for vehicles). From these a set of planar patches are extracted according to the geometry of the object:

$$\boldsymbol{\mathcal{P}}_s = \{\mathbf{p}_s^{(0)}, \mathbf{p}_s^{(1)}, \dots, \mathbf{p}_s^{(|\mathcal{P}|)}\}, \quad \mathbf{p}_s^{(i)} = \Gamma(\boldsymbol{k}_s^{(i)}) \tag{4.16}$$

where each patch $\mathbf{p}_s^{(i)}$ is defined as the convex hull Γ of the subset of keypoints $\boldsymbol{k}_s^{(i)} \subseteq \boldsymbol{\mathcal{K}}_s$. Prior knowledge about the object class can be leveraged to choose the planar patches $\boldsymbol{\mathcal{P}}_s$: for instance, we choose roof, left, right, front and back sides for vehicles.

Warping and dewarping

Warping patches Source patches $\boldsymbol{\mathcal{P}}_s$ are warped to the destination viewpoint to get a set of warped patches $\boldsymbol{\mathcal{P}}_d$ that are employed to bootstrap the

novel viewpoint synthesis. To this end, we define the destination viewpoint $\mathbf{V}_d \in \mathbb{R}^{4\times4}$ to be an arbitrary rigid transformation of the camera:

$$\mathbf{V}_d = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \tag{4.17}$$

Locations of 2D keypoints $\boldsymbol{\mathcal{K}}_d \in \mathbb{R}^{|\boldsymbol{\mathcal{K}}_s|\times2}$ in the novel viewpoint can be now computed by using the classical pinhole camera model as:

$$k_d^{(i)} = \begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix} \mathbf{V}_d^{-1} k_{3D}^{(i)} \tag{4.18}$$

where $k_{3D}^{(i)}$ is the i^{th} 3D keypoint in the CAD model and c_x, c_y are the principal point coordinates. As the focal f is unknown, we set it to an high value to minimize perspective effects; we choose $f = 3000$ as in Pascal3D+ [215]. A set of homography transformations $\mathbf{H}$ relating planar surfaces in the two views can be estimated from correspondences between $\boldsymbol{\mathcal{K}}_s$ and $\boldsymbol{\mathcal{K}}_d$. In this way patches in the destination viewpoint (*warped patches* from now on) can be computed as:

$$\boldsymbol{\mathcal{P}}_d = \{\mathbf{p}_d^{(0)}, \mathbf{p}_d^{(1)}, \ldots, \mathbf{p}_d^{(|\mathbf{P}|)}\}, \quad \mathbf{p}_d^{(i)} = \mathbf{H}^{(i)}(\mathbf{p}_s^{(i)}) \tag{4.19}$$

De-warping patches Since real-world datasets do not provide paired views, it is not possible to supervise the destination image $\tilde{\mathbf{x}}_d$; hence, we propose to train the ICN network in a *self-supervised* manner. A straightforward approach would be to reconstruct $\mathbf{x}_s$ from $\boldsymbol{\mathcal{P}}_s$. Nonetheless, this would create a distribution shift between the data fed to the network during training and inference stages. In fact, while $\boldsymbol{\mathcal{P}}_s$ is perfectly aligned with $\mathbf{x}_s$, $\boldsymbol{\mathcal{P}}_d$ might be affected by homography failures and interpolation errors among other issues. For example, when the destination area is smaller than the source one many source pixel land onto the same destination pixel, and the inverse warping cannot recover all the information in the original patch. To alleviate this shift, we train the network to reconstruct the image $\mathbf{x}_s$ from a third set of patches (called *dewarped patches* in what follows):

$$\tilde{\boldsymbol{\mathcal{P}}}_{\boldsymbol{s}} = \{\tilde{\mathbf{p}}_s^{(0)}, \tilde{\mathbf{p}}_s^{(1)}, \ldots, \tilde{\mathbf{p}}_s^{(|\mathbf{P}|)}\}, \quad \tilde{\mathbf{p}}_s^{(i)} = (\mathbf{H}^{(i)})^{-1}(\mathbf{p}_d^{(i)}) \tag{4.20}$$

During training, patches are warped towards a random viewpoint sampled from the training set distribution before being warped back to the source

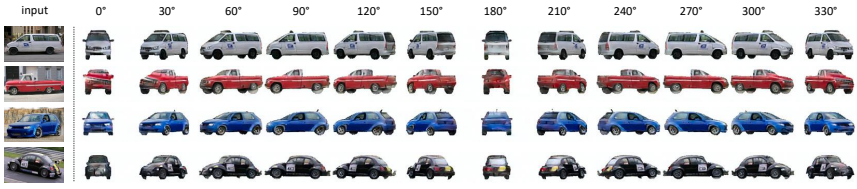


Figure 4.26: Results of 360° rotation. Our output is consistent for the whole rotation circle. Best viewed zoomed on screen.

one. In this way the network learns to cope with possible transformation errors and cannot simply short-circuit input patches to the output. The importance of this *dewarping trick* for a well-behaved network training is highlighted in Sec. 4.4.3.

Visibility model

Whenever a 3D object is projected into a 2D image, self-occlusions almost inevitably arise. Consequently, not all planar patches into the set $\boldsymbol{\mathcal{P}}_s$ are effectively visible. Were they to be warped regardless of their visibility, following parts of our architecture would require to discern which of them to keep or discard. Furthermore, when warping between $\boldsymbol{\mathcal{P}}_s$ and $\boldsymbol{\mathcal{P}}_d$ the visibility of some of those planes may vary. To take these dynamics into account, we first render the object 3D model from the camera viewpoint to obtain the 3D planes corresponding to the detected 2D patches. The z-buffer computed through ray-casting is then exploited to filter the patches which are not visible from the source viewpoint. These are 'dropped', in the sense that they are zeroed before feeding them to the ICN. As during training the intermediate viewpoint $\mathbf{V}_d$ is sampled randomly, the warping-dewarping phase results in a random dropout at patch-level, where the chance of drop is inversely proportional to the frequency of visibility of the patch. This forces the network to hallucinate missing patches during training, thus improving generalization when source and destination viewpoint differ.

Leveraging 2.5D sketches

While image patches carry rich information about the appearance of the object, they bear few cues about the object shape. In other words, visual

aspect and shape are disentangled by design. This is a desirable property enabling multiple applications which require to change one of the two while keeping the other fixed. In this section we propose a method to constrain the synthesised object shape. Let

$$\mathcal{C} = \{C^{(0)}, C^{(1)}, \dots, C^{(|\mathcal{C}|)}\}, \quad C^{(i)} \in \mathbb{R}^{faces \times 3 \times 3} \tag{4.21}$$

be the set of 3D CAD models which approximate the intra-class variation for the current object class, each $C^{(i)}$ being a 3D mesh composed of f faces. The number of CADs $|\mathcal{C}|$ needed to cover the intra-class variation reasonably depends on the object category, but it is often relatively low (e.g. $|\mathcal{C}| = 10$ for the vehicle class in the Pascal3D+ dataset [215]). Each training example i is thus composed by an image $\mathbf{x}^{(i)}$ and its associated viewpoint $\mathbf{V}^{(i)}$ and CAD index $\alpha \in \{0, 1, \dots, |\mathcal{C}|\}$ which can be possibly selected through a classifier. Therefore, a virtual camera can be used to render the CAD $C^{(\alpha)}$ from viewpoint $\mathbf{V}^{(i)}$. In particular, following [209], we render the 2.5D sketch of CAD surface normals:

$$s_{2.5D}(C^{(\alpha)}, \mathbf{V}^{(i)}) \in \mathbb{R}^{H \times W \times 3} \tag{4.22}$$

which provides rich information about the object's 3D shape. During training, this 2.5D sketch is fed to the ICN together with de-warped patches $\tilde{\boldsymbol{\mathcal{P}}}_{\boldsymbol{s}}$ to reconstruct $\mathbf{x}_s$.

Appearance prior

Our method relies on warped patches to transfer the object appearance from a source to a destination viewpoint. Still it might happen that viewpoints $\mathbf{V}_s$ and $\mathbf{V}_d$ are so far apart (e.g. front to back) that an object shares no visible faces across the two even with symmetry constraints. To alleviate this issue, we crop from the center of the input image $\mathbf{x}_s$ a small patch c_s with side 10% of the image size and give it as an additional input to the image completion network as a prior knowledge about the rough object appearance in absence of other hints (depicted in Fig. 4.24). The network can extract from this crop coarse information about the object visual aspect (e.g. the average color) as a prior to cope with large changes between viewpoints.

Image Completion Network

The Image Completion Network (ICN) $g(\cdot \mid \theta)$ is a fully convolutional network parametrized by θ trained to reconstruct a realistic image $\mathbf{x}_s$ from dewarped patches $\tilde{\boldsymbol{\mathcal{P}}}_{\boldsymbol{s}}$, 2.5D sketch $s_{2.5D}$ and appearance prior c_s:

$$\tilde{\mathbf{x}}_s = g(\tilde{\boldsymbol{\mathcal{P}}}_{\boldsymbol{s}}, s_{2.5D}(C^{(\alpha)}, \mathbf{V}_s), c_s \mid \theta) \tag{4.23}$$

Architecture Our ICN features an encoder / decoder structure as in [236, 238]. The encoder is composed of 3 convolutional blocks to reduce the spatial resolution and 3 additional residual blocks applied to the lowest resolution code. Except for the first one, every convolution has kernel size 4x4 and it's preceded by reflection padding, while ReLU activation and Instance normalization are applied afterwards. The decoder follows the same structure in reverse order. For the discriminator network we rely on the two-scale PatchGAN classifier [77, 236, 238].

Objective A number of recent works [81, 28, 143] indicate that loss functions based on high-level features extracted from pretrained networks can lead to much more realistic results compared to naive *per-pixel* losses between the output and ground-truth image. Given a set of layers $\{\Phi_l\}$ from a network Φ and a training pair consisting of a real and a generated images $(\mathbf{x}_s, \tilde{\mathbf{x}}_s)$, we define the perceptual loss function as

$$\mathcal{L}^{VGG}_{\mathbf{x}_s, \tilde{\mathbf{x}}_s}(\theta) = \sum_l \lambda_l \|\Phi_l(\mathbf{x}_s) - \Phi_l(\tilde{\mathbf{x}}_s)\|_1. \tag{4.24}$$

We employ each second convolutional layer of each block in VGG-19 [167] as feature extractor Φ_l; $\{\lambda_l\}$ is set as in [28].

As mentioned above, images generated from novel viewpoints $\tilde{\mathbf{x}}_d$ cannot be directly supervised if the dataset does not provide paired views. Nevertheless, we can still enforce the realism of ICN output in an adversarial fashion. Given a generic image $\tilde{\mathbf{x}}$ synthesised by ICN either in the source ($\tilde{\mathbf{x}}_s$) or the destination ($\tilde{\mathbf{x}}_d$) viewpoint, we set up a min-max game as follows:

$$\mathcal{L}^{adv}_{\mathbf{x}_s, \tilde{\mathbf{x}}} = \mathbb{E}_{\mathbf{x}_s}[\log D(\mathbf{x}_s)] + \mathbb{E}_{\tilde{\mathbf{x}}}[\log(1 - D(\tilde{\mathbf{x}}))] \tag{4.25}$$

where D is the discriminator network aiming to distinguish between real and synthesised images. Our total loss is defined as:

$$\mathcal{L} = \mathcal{L}^{VGG}_{\mathbf{x}_s, \tilde{\mathbf{x}}_s} + \gamma \mathcal{L}^{adv}_{\mathbf{x}_s, \tilde{\mathbf{x}}} \tag{4.26}$$

where γ modulates the contribution of the adversarial term.

Figure 4.27: Comparison with ablated versions of the proposed method on Pascal3D+ test set. Better viewed zoomed on screen. Please refer to Sec. 4.4.3 for details.

Figure 4.28: Visual results comparison with competitors on Pascal3D+ test set. Better viewed zoomed on screen. Please refer to Sec. 4.4.3 for details.

4.4.3 Experimental results

The experiments we propose concern with the evaluation of the quality of the generated object images across different viewpoints. First, both visual comparison (Sec. 4.4.3) and quantitative experiments (Sec. 4.4.3) against state-of-the-art competitors are reported. Then, we keep the human in the loop by relying on human judgement to measure the output quality via A/B preference tests (Sec. 4.4.3). Eventually, we extend the evaluation to other classes (Sec. 4.4.4) and we investigate the contribution of complementary synthetic data for modelling extreme viewpoint changes (Sec. 4.4.5).

Datasets Although large-scale 3D shape repositories providing object geometries such as Princeton Shape Benchmark [162] and Shapenet [23] exist, they do not come with real-world images aligned. As we want to work with real-world data, we rely on Pascal3D+ [215], an in-the-wild 3D object detection dataset which augments the 12 rigid categories of the PASCAL Visual Object Classes (VOC) [47] with 3D annotation -roughly [174]- aligned. In particular, we use the *car* and *chair* subsets, which consist of around 5000 and 1500 images respectively.

Competitors We evaluate our method against six state-of-the-art works in the task of novel viewpoint synthesis. The first one is *VON* [238], an ad-

Figure 4.29: Predictions of our model from different viewpoints. The geometry-aware design of our semi-parametric method allows the model to be resilient to large viewpoint variations, including rotation, elevation and camera distance.

	0°	30°	60°	90°	120°	150°	180°	210°	240°	270°	300°	330°	Avg
ours$_{\text{real+synth}}$	174.2	57.0	47.9	53.8	52.9	59.9	168.8	61.2	52.1	54.2	46.8	60.2	74.1
ours$_{\text{real}}$	178.9	54.6	51.0	53.2	56.8	58.1	202.5	62.2	54.1	49.8	46.3	56.1	77.0
MV2NV [173]	144.5	170.9	136.5	192.3	144.7	163.1	152.2	179.1	137.5	186.0	136.0	161.3	158.7
MV3D [177]	257.5	263.9	259.7	284.2	273.0	271.9	267.7	263.7	261.5	277.9	266.5	265.3	267.7
TVSN [134]	70.8	71.3	74.1	78.9	79.0	82.2	89.4	81.1	79.0	78.9	78.0	72.9	78.0
VUnet [46]	202.4	90.7	79.9	88.3	78.8	96.3	203.3	94.0	77.5	85.2	82.0	92.6	105.9
VON [238]	134.1	107.2	150.0	126.5	124.4	114.4	151.7	113.8	127.2	128.9	132.0	107.3	126.4
VON$_{FT}$ [238]	165.2	100.6	137.8	125.9	137.6	108.1	190.5	155.7	134.8	123.1	117.1	102.1	133.2

Table 4.5: Fréchet Inception Distances [72] results for *car*. Each row lists the average distance between real and generated images for each method on the left. Results are reported from 12 evenly spaced azimuthal angles while rotating around the object at fixed elevation and radius. Details in Sec. 4.4.3.

versarial learning framework in which object shape, viewpoint and texture are treated as three conditionally independent factors that contribute to the synthesis of the novel viewpoint. Since *VON* was originally trained on a custom car dataset collected by the authors [238], for a fair comparison we implement a second baseline VON_{FT} by fine-tuning their network on Pascal3D+. For both competitors we provide ground truth voxelized shapes from [238] matching the images' CADs, relying on the texture encoder network from [238] for extracting the textures from Pascal3D+ images. Third, we compare to *VUnet* [46], a state-of-the-art framework for conditional image generation based on variational autoencoder [88], which shows a good generalization capability across a variety of poses and viewpoints. In the authors' implementation [46] a U-net [152] architecture is fed with keypoint-based skeletons to perform pose-guided human generation. We re-train their model on Pascal3D+ to perform pose-guided object generation. For this process, we feed their shape-encoding network with our 2.5D sketches rendered from the object CAD - which is a more informative reference signal than the one used in the original implementation (i.e. skeleton or edges).
We also compare with three recent *pairwise-trained* models: MV3D [177], TVSN [134] and MV2NV [173]. Pairwise methods share the need for both source and target pairs during training; thus we cannot re-train or finetune them on Pascal3D+ and we rely on pre-trained models released by the authors. To maximize evaluation fairness, in what follows we only sample novel viewpoints rotating around the z-axis at fixed distance and elevation, which is the only setting handled by competitors. Still, our method can handle general roto-translations as well as variation in camera intrinsic. Fig. 4.29 and Fig. 4.31 show visual results for large viewpoint changes in both elevation and azimuth; even more extreme transformations are depicted in Fig. 4.34.

Implementation details The 2D bounding box of each example of Pascal3D+ [215] is padded to a squared aspect ratio and resized to 128x128 pixels. We work in LAB space relying on the training procedure from [213]. Following [188, 136] truncated and occluded objects are discarded, resulting in 4081 training and 1042 testing examples respectively. For vehicles, the $\text{ours}_{\text{real+synth}}$ model is trained for 20 epochs with batch size 8. During training images undergo small random rotations, translations and shearing for data augmentation purposes. We use Adam [87] optimizer with

constant learning rate $2e^{-4}$. Loss balancing term γ is set to 8. The code is developed in PyTorch [135]: we depend on Open3D library [233] for 3D data manipulation and rendering. Random search has been employed for hyper-parameter tuning. Without any optimization, inference for a 128x128 image takes $\sim 3ms$ on a NVIDIA GTX 1080, making it suitable for real-time applications.

Visual results

Our model produces high-quality results for a variety of camera viewpoints, preserving fine-grained object appearance, as it can be appreciated in Figures 4.23, 4.26, 4.27, 4.28, 4.29, 4.33 and 4.34.

Competitors The key differences between the proposed method and competitors can be appreciated in Fig. 4.28. From left to right. MV2NV [173] seems to suffer the most the reality gap as well as the lack of multiple views, leading to the worse quality results in our setting. MV3D [177] predicts at least a reasonable overall shape, although images are blurry to the point that in some cases one can barely recognize the vehicle. Results from TVSN [134] show the highest variability: while looking generally fine, they are disastrous for less common poses such as (b) and (e). The output from Visual Object Networks [238] (*VON*) is generally realistic, but hardly reflects the visual appearance of the input. Furthermore, both *VON* and VON_{FT} generator networks do not generalize to poses which are less common in the training set such as the frontal pose in (d). VUnet [46] suffers from blurred results typical from variational autoencoders [63]; also, due to skip connections, input appearance may leak to the output when the two viewpoints are very different (b). More generally, the drawbacks of a solely learning-based viewpoint synthesis are evident in (a, c): complex textures cannot be recovered once compressed in a feature vector.

Ablation study Ablated versions of our model are shown in Fig. 4.27. The effect of removing the appearance prior is showcased in *No-prior* column. Without prior information, the ICN fails to infer the object appearance when no planar patch is provided, as shown in (b, e). Removing the adversarial term (*No-adv* column) results in slightly blurred outputs. We also investigate the aid of the *dewarping trick* presented in Sec. 4.4.2. In *No-dewarp* column, the ICN was trained to reconstruct the image from $\boldsymbol{\mathcal{P}}_s$ instead

of $\tilde{\boldsymbol{\mathcal{P}}}_{\boldsymbol{s}}$ (see Sec. 4.4.2). As expected, despite the very low reconstruction error at training time (due to the similarity between $\mathbf{x}_s$ and $\boldsymbol{\mathcal{P}}_s$), the model fails to generalize to the synthesis of novel viewpoints where the textures $\boldsymbol{\mathcal{P}}_d$ are the result of an homography transformation. This highlights the importance of the *dewarping trick* for a well-behaved training. Eventually, *Sil-only* shows the ablated versions in which the input sketch is constituted only by 2D silhouette. Although results do not differ dramatically, it can be appreciated how the network benefits from additional information to resolve ambiguous situations such as self-occlusions (e) and details such as side windows, lights, wheels.

Shape transfer Visual results for shape transfer are showed in Fig. 4.23. In this setting, the network is requested to complete the warped faces $\boldsymbol{\mathcal{P}}_d$ using the 2.5D sketch rendered from a totally different CAD. It can be appreciated how novel viewpoints are still realistic, since the network exploits the 2.5D sketch to complete the warped appearance in a CAD-agnostic manner.

Metrics and quantitative results

Fréchet Inception Distance To quantitatively measure the similarity between generated and real images we rely on Fréchet Inception Distance (FID), which was shown to consistently correlate with human judgment [72, 112][2]. We employ activations from the last convolutional layer of an InceptionV3 model pretrained on ImageNet [40] as features. Assuming a multidimensional Gaussian distribution for these features, we compute the FID as follow:

$$\text{FID} = \|\boldsymbol{m} - \boldsymbol{m}_w\|_2^2 + \text{Tr}\big(\boldsymbol{C} + \boldsymbol{C}_w - 2\big(\boldsymbol{C}\boldsymbol{C}_w\big)^{1/2}\big) \tag{4.27}$$

Where $\boldsymbol{m}$, $\boldsymbol{C}$ are the mean and covariance of the features extracted from Pascal3D+ data, while $\boldsymbol{m}_w$ and $\boldsymbol{C}_w$ are the corresponding statistics extracted from the generated images.
To enable the comparison with other works, we sample novel viewpoints while rotating around the object at fixed distance and elevation: results are

[2]There are also critic voices who point out that these measures can be easily fooled by optimizing for the evaluated measure, resulting in images which score very high despite being visibly (for a human) unrealistic [28]. How to quantitatively evaluate the realism of an image without a human in the loop is still open to debate.

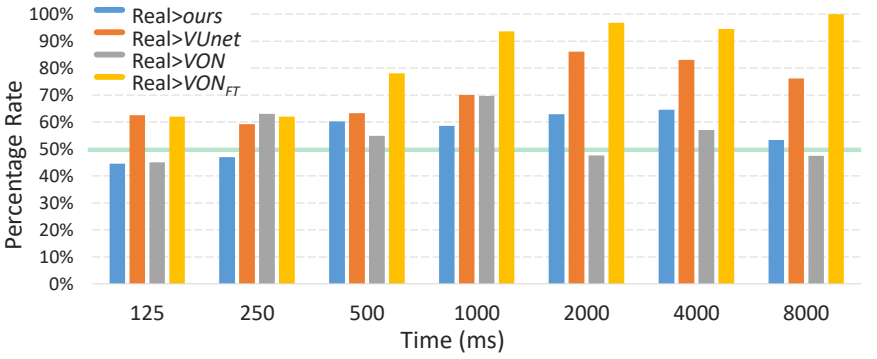


Figure 4.30: Results of time-limited A/B preference test against real images. Both *VON* and our method are resilient to human judgement over time. Green line denotes random chance. Please refer to Sec.4.4.3 for details.

	Car (plain)	Car (textured)	Avg
ours > *VUnet* [46]	76.0%	85.0%	78.0%
ours > *VON* [238]	88.0%	98.0%	91.0%
ours > VON_{FT} [238]	96.0%	99.0%	97.0%

Table 4.6: Blind randomized A/B test results. Each row lists the percentage of workers who preferred the novel viewpoint generated with our method with respect to each baseline (chance is at 50%).

reported in Table 4.5, binned in 12 equidistant azimuthal angles. Fréchet Inception Distance rewards the realism we can get with our semi-parametric approach; our method outperforms all competitors. In particular, our method preserves both low (i.e. shape) and high-frequency (i.e. texture) image statistics which are both contribute to the overall FID score.

Perceptual experiments

To assess the quality of our results also from a perceptual point of view, randomized A/B preference tests were performed by 43 human workers, following the experimental protocol of previous works [28, 143, 238]. Due

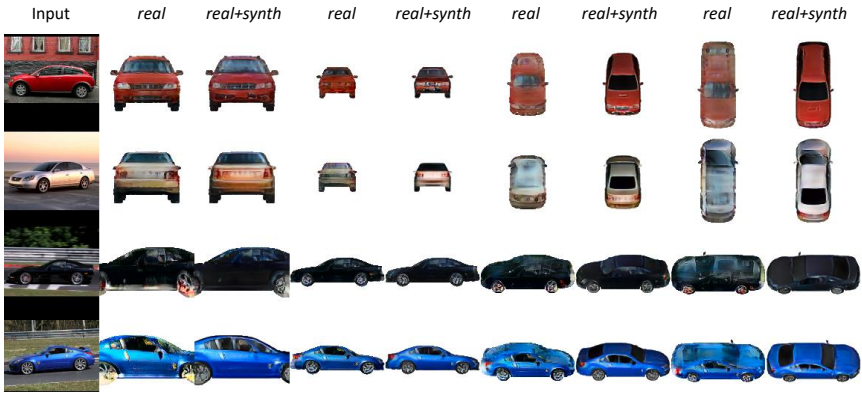


Figure 4.31: Visual comparison showing the effect of adding synthetic data to Pascal3D+ training set. The ICN network trained on the mixture of the two domains performs significantly better under extreme viewpoint transformations. Please see supplementary material for more examples.

to time constraint, for this phase we select only the three most recent competitors, namely *VON* [238], VON_{FT} [238] and *VUnet* [46]. As we want to evaluate both the realism and the appearance coherence of our method, we perform two different tests.

View transfer coherence In the first setting, the subject is presented with three images: while the first one comes from Pascal3D+ test set, A and B depict a novel viewpoint of the object generated with two different methods. The human worker is then asked whether rotating the input object would better lead to A or B. Results reported in Tab. 4.6 indicate that our method is largely preferred to competitors, likely because of the built-in realism that comes from warping the original image. As a further analysis we split by manual annotation Pascal3D+ images into *plain* and *textured* sets, the latter set containing vehicles which feature characteristic textures. Table 4.6 highlights that workers expressed almost unanimous preference for our method on the *textured* set. The fact that human attention was caught by these appearance details highlights the importance of preserving fine-grained details in the synthesized output.

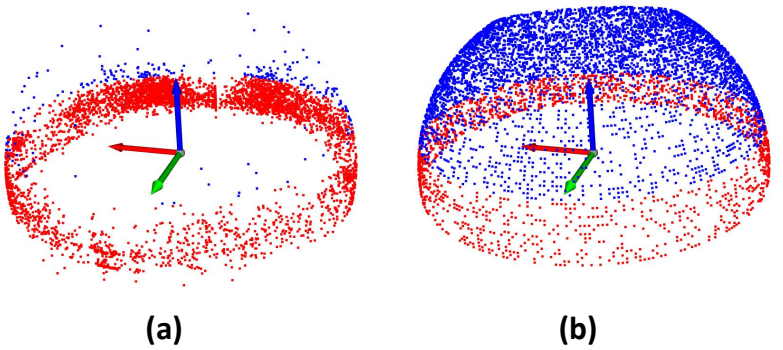


Figure 4.32: Viewpoints' distributions for real (a) and synthetic data (b) of the car class in Pascal3D+. Radii have been normalised to unit length for clearness. In red viewpoints with elevation lesser or equal than $\frac{\pi}{8}$ rad.

Output realism The second experiment consists of a two-alternative forced choice aimed at evaluating the relative realism of each method. Here the subject is presented with only two images for a determined amount of time. The worker is then asked which of the two appeared more realistic and the experiment is repeated by varying the amount of time. Results depicted in Fig. 4.30 follow two trends. On the one hand, workers clearly discern *VUnet* and VON_{FT} images from real ones as more time is available. *VUnet* is hurt by excessive blur and visual artifacts; VON_{FT} suffers from a severe loss of realism w.r.t. the original *VON* method, which may be related to the great variety of viewpoints in the Pascal3D+ dataset compared with the one used in Zhu *et al.* [238]. On the other hand, both *VON* and our method produce realistic images workers struggle to distinguish from the real ones even in 8000ms.

4.4.4 Extending the evaluation on different classes

Similarly to [173, 134, 238], we test our method also on the *chair* subset from Pascal3D+ dataset, consisting of 1195 images annotated with 10 different CAD models to assess the generalisation capability of our method. As for vehicles, also for chairs we define a set of planes from 3D annotated keypoints to approximate the surface of the objects, namely: left, right,

Figure 4.33: Visual results comparison with competitors for *chair* class on Pascal3D+ test set. Better viewed zoomed on screen.

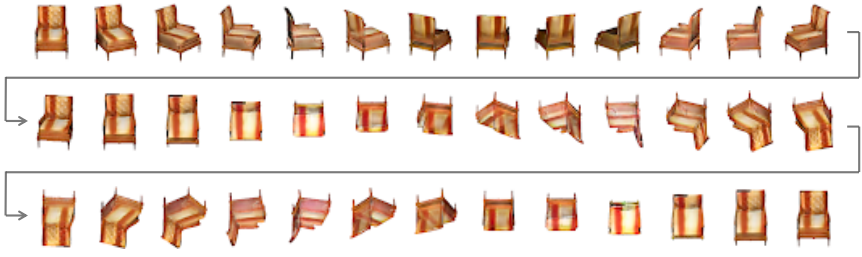


Figure 4.34: Geometric guidance built in our semi-parametric model allows to perform extreme object transformations which are currently unfeasible for any fully-parametric method. In this case we are able to rotate the armchair upside-down, although this configuration never appears in the training set. Best viewed zoomed on screen.

	0°	30°	60°	90°	120°	150°	180°	210°	240°	270°	300°	330°	Avg
ours$_{\text{real+synth}}$	83.9	57.9	66.5	120.1	82.5	76.1	102.3	77.2	83.4	117.0	65.2	60.9	82.8
ours$_{\text{real}}$	92.6	62.4	68.0	122.5	84.8	84.7	117.3	82.3	85.0	127.5	70.1	64.0	88.4
TVSN [134]	84.7	86.6	90.8	95.1	94.0	97.2	93.9	94.6	95.0	93.0	87.8	82.7	91.6
VON [238]	125.5	107.7	100.1	121.1	122.6	136.8	203.9	141.4	123.8	114.2	102.1	96.1	124.6
VUnet [46]	118.9	97.1	123.1	171.0	160.5	137.9	151.2	141.9	155.8	154.9	120.1	95.8	135.7
MV2NV [173]	180.3	168.0	178.3	187.8	184.1	177.8	184.7	184.3	200.2	191.0	184.7	166.4	182.3

Table 4.7:]
Fréchet Inception Distances [72] results for *chair* class. Each row lists the average distance between real and generated images for each method on the left. Results are reported from 12 evenly spaced azimuthal angles while rotating around the object at fixed elevation and radius.

seat and back. Even though chairs often feature holes and slits leaking some of the background, we treat them as filled planes. This is due to having access to the foreground segmentation mask through the rendered CAD model, which can be used to mask out background areas. However, the very coarse alignment of the chairs CAD models in Pascal3D+ leads to an additional difficulty when training the ICN.
Table 4.7 reports results against competitors in terms of Fréchet Inception Distance score: MV3D [177] is omitted since it only releases pre-trained model for cars. Our method outperforms all other competitors also for this class of objects; visual examples are reported in Fig. 4.33. While MV2NV [173] and TVSN [134] often struggle to generate the object from the correct viewpoint, VON [238] fails to transfer visual details in the final output. Although the generated image looks realistic, it doesn't resemble the input one. Contrarily, our method successfully generate realistic views of the input object, even under severe viewpoint transformations 4.34. Still, when background leaks in the input mask due to the CAD misalignment (first and fourth row) the final output quality decreases.

4.4.5 On the use of synthetic data

Even though the semi-parametric nature of our proposed method copes with a variety of viewpoint, it still rely on data to learn how to stitch together the warped patches. Therefore, for dramatic changes of viewpoint that are completely uncovered in the dataset, performances may drop significantly.

As shown in Fig. 4.32 (a), Pascal3D+ viewpoints' distribution for the car set is profoundly skewed towards low elevation values and is polarized with regard to the azimuth to frontal and lateral views, reflecting how the images were acquired. As it is of great interest to produce realistic images from more varied viewpoints (e.g. bird's eye view), we include synthetic data in the training set for balancing the viewpoints' distribution. To this end, we sample 59 models from the car synset in the ShapeNet [23] dataset and we annotate them with 3D keypoints to define the planes of interest. We then render 6950 images sampling viewpoints uniformly in a 3D semi-spheres around the origin. As shown in Fig. 4.32 (b) this viewpoints' distribution is much more uniformly distributed in terms of azimuth and elevation. We name $\text{ours}_{\text{real+synth}}$ the ICN network trained on a mixture of synthetic and real data. We also experimented pre-training our model on synthetic data and fine-tuning on the real ones, although in our experience that policy led to worse results.
Visual comparison between $\text{ours}_{\text{real}}$ and $\text{ours}_{\text{real+synth}}$ is shown in Fig. 4.31. It can be seen how training only on Pascal3D+ entails artefacts for out-of-distribution viewpoints (e.g. bird's eye view). Conversely, combining the two domains the network learns from synthetic data a prior about the overall structure and color.
We perform an analogous augmentation of the chair class. In this case we included 1858 freshly rendered synthetic images from 73 annotated models. As the rendered images are perfectly aligned with the foreground mask, this also contributes to reduce the planes' misalignment introduced by images from Pascal3D+.

4.4.6 Concluding remarks

In this chapter we introduced a novel formulation of the problem of vehicle novel viewpoint synthesis in a semi-parametric setting. Notably, our model is designed to be trainable on existing datasets for 3D object detection in a self-supervised manner, without the need for paired source/target viewpoint images - although it can be complemented with synthetic data. Non-parametric visual hints act as prior information to guide a deep parametric model for generating realistic images, disentangling by design appearance and shape. This enables truly continuous manipulation of the viewpoint and shape transfer to different 3D models. As completing the image is much easier than generating it from scratch, we can train our ICN on just

Figure 4.35: Artificial data created stitching generated vehicles onto Pascal3D+ [215] backgrounds.

few thousands images from the Pascal3D+ dataset and still be able to generalize to unseen viewpoints.
Although vehicles were the main focus of our work, we show that our framework is generic enough to handle rigid objects of completely different geometric structure such as chairs. Perceptual experiments results as well as image-quality metrics reward our method for its realism and the visual consistency of the synthesised object across arbitrary points of view.

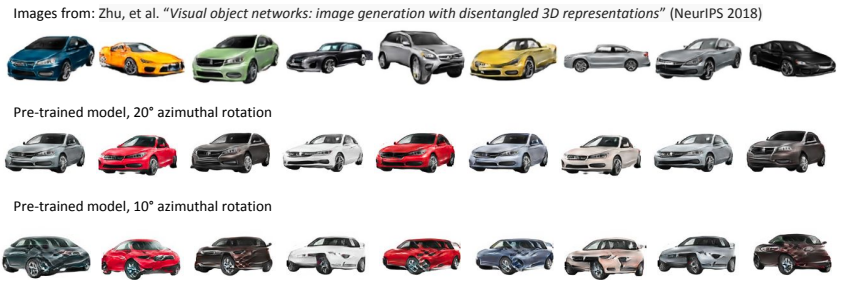


Figure 4.36: Qualitative results from VON [238]. Since the model has no geometric constraints, a very small viewpoint variation from a situation where it performs impressively (middle) can lead to a dramatic failure (bottom). Please see Sec. 4.4.7 for details.

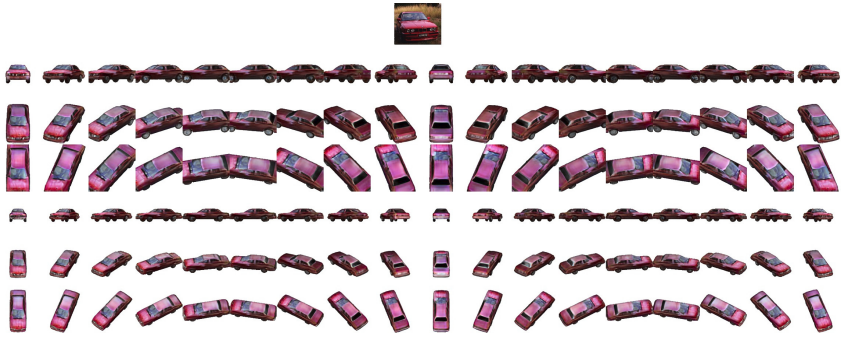


Figure 4.37: The the explicit texture warping and the geometric guidance given by the 2.5 sketches allows our method to produce consistent predictions from arbitrary viewpoint even tough no explicit consistency loss between different views is optimized. First row shows input. Best viewed in color.

4.4.7 Additional discussion

Regarding output consistency

In this work we argue that a semi-parametric method for object novel view synthesis has very appealing properties coming from the fact that the output is not generated from scratch; on the opposite, much of the information is warped from the input viewpoint in a geometrically-principled manner. We find that one of the major advantages of this semi-parametric approach is that the geometric guidance helps avoiding catastrophic failures in the generation process. In particular, our model does not have to explicitly learn the hard concept of distance between two viewpoints in 3D space. Since the 2.5D sketch which guide the generation process are rendered from a 3D model in a purely geometric fashion, continuous change in viewpoint will lead by construction to continuous variation in the 2.5D sketch.
Conversely, in a completely learning-based pipeline the model is let to learn the concept of 3D viewpoint proximity from a massive amount of data. Still, it is a very hard concept: the model may or may not learn it properly. In Fig. 4.36 we make use of the VON [238] pre-trained model to showcase one situation of this kind. Even though the model is state-of-the-art and its performance is impressive, very little viewpoint variation can result in a dramatic failure in the image generation for no apparent reason.

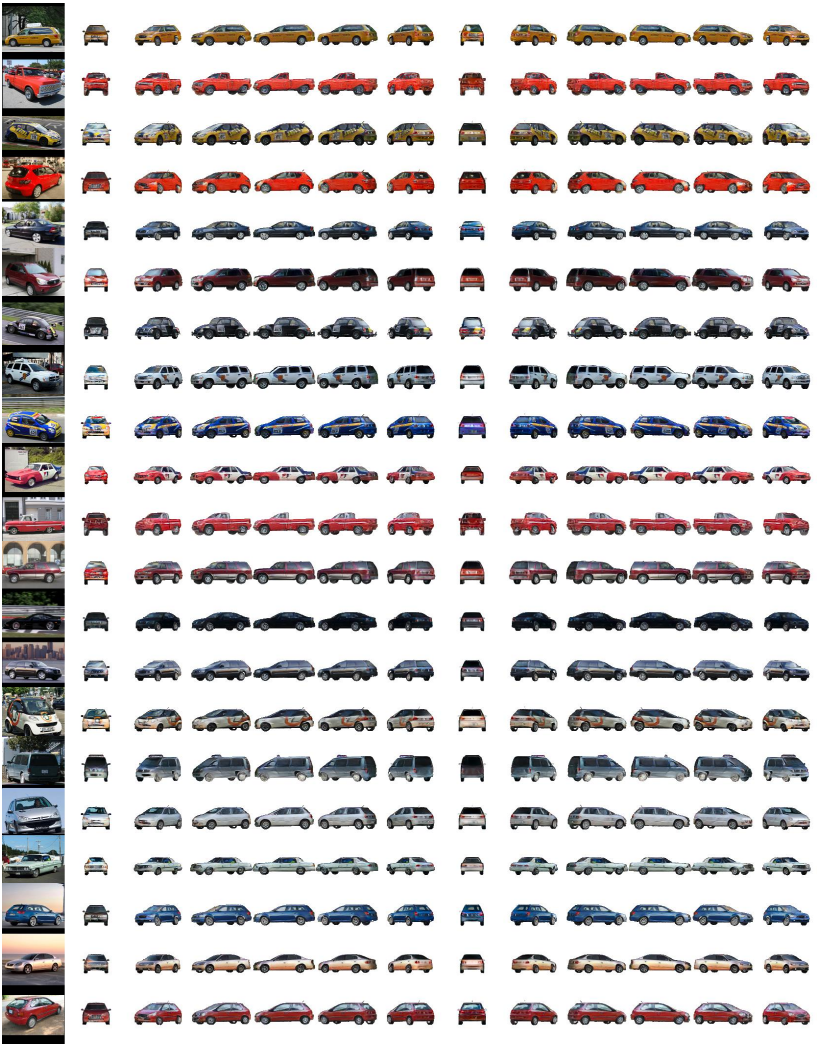


Figure 4.38: Visual examples of 360 degrees car rotation. Best viewed zoomed on screen.

Figure 4.39: Additional visual results from our model, showing how training the ICN on a mix of synthetic and real data dramatically improves the performance for out-of-distribution viewpoints.

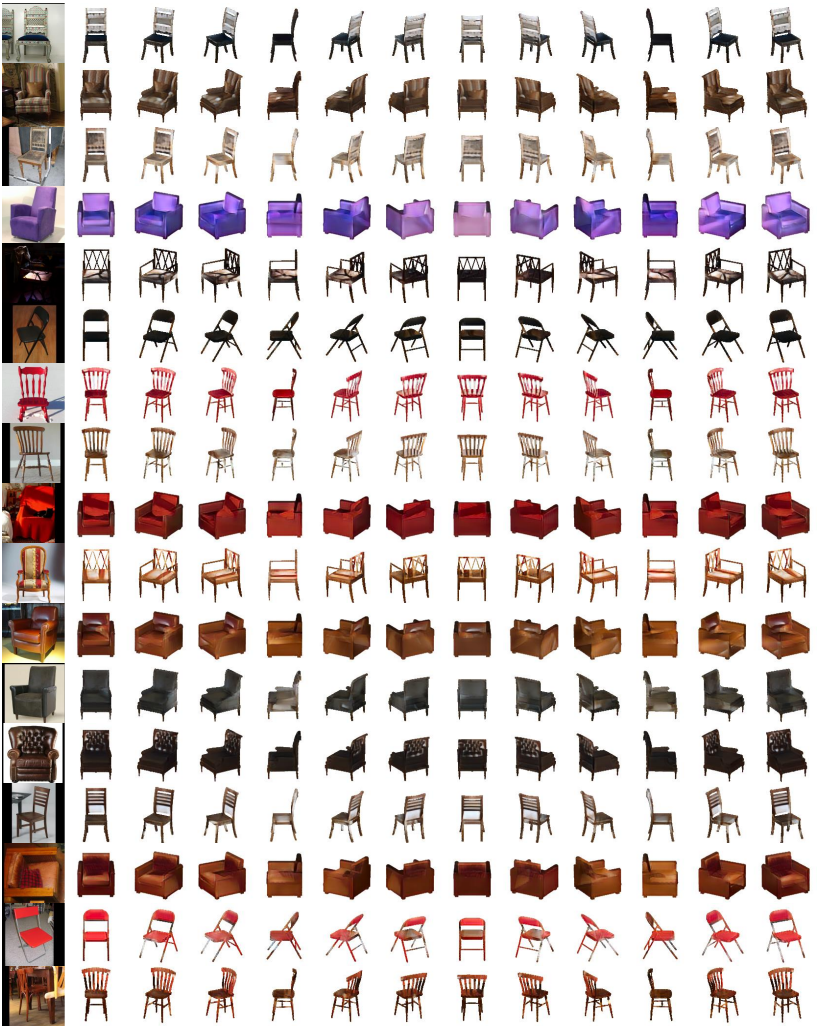


Figure 4.40: Visual examples of 360 degrees chair rotation. Best viewed zoomed on screen.

Figure 4.41: Additional visual results for shape transfer on vehicle class.

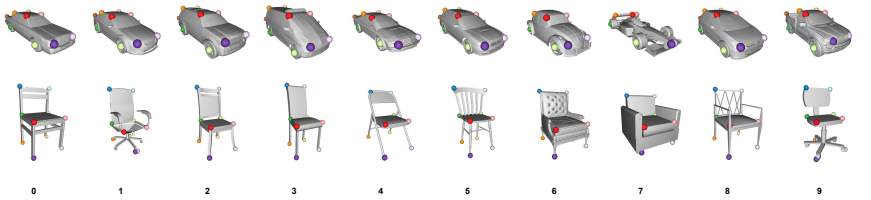


Figure 4.42: Semantic keypoints annotated for each 3D model in the Pascal3D+ [215] dataset: bottom row indicates the CAD index in the dataset. These are the same keypoints we choose to annotate in our synthetic dataset. Please refer to Sec. 4.4.7 for further discussion.

On the opposite, the explicit texture warping and the geometric guidance given by the 2.5 sketch allows our method to produce consistent predictions from arbitrary viewpoints even tough no explicit consistency loss between different views is optimized. Visual examples from our methods are shown in Figures 4.37,4.39,4.38,4.40, where we drastically jointly change both azimuth and elevation of the target viewpoint. Additional visual results for shape transfer on the vehicle class are depicted in Fig. 4.41.

Decomposing the object using the 'right' keypoints and planes

In this work semantic keypoints are leveraged as a proxy to decompose the object in the image into a small set of planar faces, with the keypoints constituting the vertices of each face.
There is no universal agreement about the number of significant objects' landmarks, not even for most common classes such as vehicles. Indeed, different works have often used a different number of keypoints to characterize the same objects [98, 215, 214, 203, 210, 198].
Here we follow the convention of the Pascal3D+ [215] dataset, which defines 12 keypoints for vehicles (front and back wheels, upper windshield, upper rear window, front light, back trunk - left and right) and 10 for chairs (back upper, seat upper, seat lower, leg upper, leg lower - left and right).
The location of these keypoints on the Pascal3D+ models can be visually inspected in Fig. 4.42; some examples of annotated models are depicted in Fig. 4.43. These keypoints are then used to define a small set of planes to approximate the object, each one delimited by at least 3 distinct keypoints.

Figure 4.43: Random 3D models from our synthetic dataset, rendered together with their annotated 3D keypoints.

The size of this set is an hyper-parameter for our model. In fact, this value must be set accordingly to balance two factors. On the one side, as the number of planes increases the same must hold true for the number of available keypoints. A very large number of keypoints would lead to a good approximation of the whole 3D surface of the object - however, in practice the number of annotated keypoints in a dataset is seldom higher than a few dozens. On the other hand, when the number of keypoints is extremely low the set of planes hardly approximate the surface of the object. Here we empirically found that six planes are enough for the car setting, while four suffices for the chair one. We believe this to be due to the presence of textures and other high-frequency details on these planes. Although the side of a vehicle is not a single flat surface (thus introducing a strong approximation while relying on planar homography for the warping) the presence of those details tricks the human's eye in believing the surface has

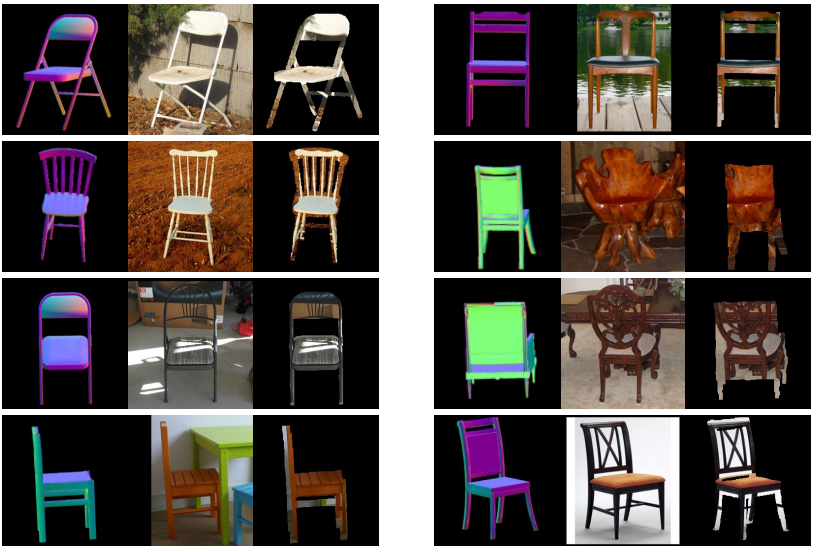


Figure 4.44: Examples of misalignment between the source image and the annotated cad model. Mainly sources of error consist of wrong annotated model viewpoint and the model not resembling the depicted object.

in fact three dimensions. Then, the ICN is trained to fix inconsistencies at the borders between warped planes and to merge all the patches into a seamless figure.

CAD alignment details

Perfectly aligning a 3D model with a 2D image is hard, even when is a human the one attempting the process. Indeed, many datasets featuring 3D annotation come with annotation issues - and Pascal3D+ [215] dataset makes no exception. This is due to two main reasons. Firstly, the CAD models have been placed by human workers and then refined using an automatic algorithm. Several geometric simplifications and assumptions are introduces (e.g. about camera intrinsic), and re-projection errors and human mistakes may arise. According to Pascal3D+ [215]:

> The annotator (...) rotates the 3D CAD model (...). The

Figure 4.45: Most common failure cases of our model on the vehicle class. Please see Sec. 4.4.7 for details.

> alignment provides us with rough azimuth and elevation angles, which are used as initialization (...). We assume a simplified camera model, where the world coordinate is defined on the 3D CAD model and the camera is facing the origin of the world coordinate system (...).

Secondly, those CAD models don't match the object portrayed, as they hail from a very reduced subset of only 10 elements. Again, from Pascal3D+ [215]:

> The annotator first selects the 3D CAD model that best resembles the object instance (...).

Figure 4.44 shows some example of the effects of this misalignment for the chair subset, where holes and slits present an additional source of error.

Failure cases

We refer the reader to Fig. 4.45 for visual examples of failure cases for the vehicle class. In (a) the ICN is not able to recover in case of input patches which are grossly wrong, either due to a failure in keypoint estimation or to further objects occluding the car such as the people in the first image. Furthermore, we report cases in which the ICN fails to guarantee a realistic output which is consistent with the input image. This happens particularly from rarer viewpoints such as bird's eye (b) and back (c) views.

Figure 4.46: The background leakage is the most common failure case for the *chair* class. This can be solved by providing a 3D model which more closely resembles the one of the input. Please refer to Sec. 4.4.7 for details.

Even more than for vehicles, the concave geometry of the *chair* object makes easier for unwanted portions of the background to leak in the generation process. In theory, our method can perfectly deal with these cases, as it masks the generated image with the silhouette of the rendered model. However, in some cases the 3D chair model rendered is so different from the one in the input image that spurious background region make their way to the final output. In Fig. 4.46 we provide visual examples of this case. Arguably, this problem would be much alleviated if more 3D models were used for inference (at the current state we only choose among the 10 Pascal3D+ models).

Visualisation Tool

We release at https://github.com/ndrplz/semiparametric the visualization tool we used to inspect the results and create most of the images in this paper. It is written in Python and depends on the Open3D [233] and OpenCV [18] libraries. The interface simulates a camera moving around an object centered in the origin. The movement are described in terms of spherical coordinates and radius, and each of the three components can be

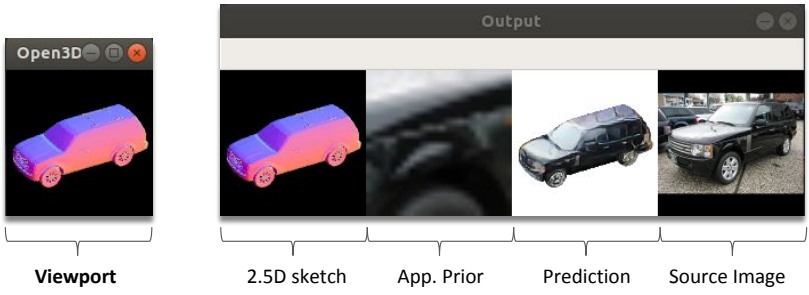


Figure 4.47: Visualisation Tool interface example.

manipulated individually. The user can change:

- the elevation value, thus moving from lateral to bird-eye camera;
- the azimuth value, thus moving around the object centered in the origin describing a 360 degrees circular trajectory;
- the radius value, thus simulating a zoom.

The Open3D library provides a 3D environment where the CAD model is loaded, and callbacks enable rendering the scene as a 2D image. Normals visualisation is also supported. Along with the manipulation of the previous variables, the user can also independently change the CAD model and the appearance image. When started, the tool present the CAD aligned with the input image accordingly to the annotated viewpoint in the data. By changing the spherical coordinates and the radius new viewpoints of the object are synthesised, while by selecting a different CAD model a shape transfer is performed. An example of the interface is shown in 4.47.

A/B Perceptual Experiments

We developed a tool to perform A/B test preferences based on Django web server. The system is centralised and can be accessed from multiple terminals simultaneously. We wrote asking for volunteers on the University of Modena and Reggio Emilia mailing list, and the tests were performed from the volunteer's computer. We didn't interact directly with participants to avoid introducing any bias inadvertently. To ensure the correctness of the

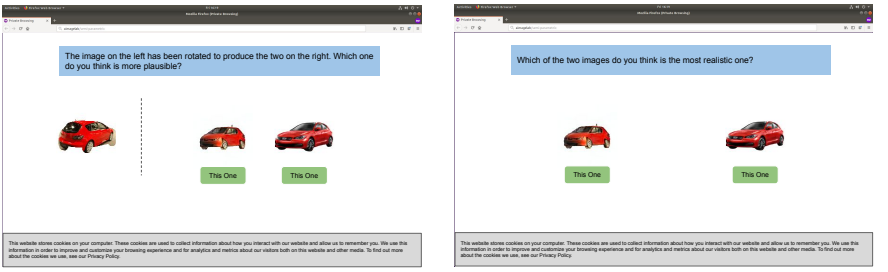


Figure 4.48: Screenshots for the two A/B preference tests settings.

data we discarded the first results from each session, which were considered as a briefing. For all experiments, images were shown at the same resolution of 128x128 pixels. As all methods produce a white background, ground truth aligned 3D CAD is used to mask Pascal3D+ real images. Both sampling order and left-right order of A and B were randomized. Raw results have been stored into a MySQL DB to ensure persistence, while aggregated statistics were presented on an administrator interface. Figure 4.48 depicts screenshots from the two perceptual experiments described in Sec.4.

4.4.8 From the vehicles to the whole scene

In the previous sections we introduced a novel semi-parametric approach to generate novel views of a vehicle from a single monocular image. Here we sketch how the aforementioned method can be applied to hallucinate the visual appearance of the whole scene in the next future; the task overview is depicted in Fig. 4.49. In this section we outline the main ideas and preliminary results (see also Fig. 4.50 and Fig. 4.51) as the current research is still in an early stage.

Problem overview

The capacity to imagine how the visual aspect of a scene will evolve in the future is definitely non-trivial. This should not discourage an attempt to solve it, as the rise of deep learning during the last years demonstrated that many tasks considered extremely hard or even unsolvable can instead

be tackled - often with superhuman performance - given enough data, computational power and a large enough model. In this setting, intuition might suggest as appropriate to borrow models from the image/video generation literature. For example, an image translation framework [77] might be adopted with small tweaks to translate the current scene into the future one; alternatively, a recurrent convolutional model [216] could be trained to recursively generate the next frame until the desired distance in the future. However, we argue that this problem is still too complex to be currently solved in an end-to-end fashion, no matter how powerful the model is. Indeed, all the aforementioned approach share a number of drawbacks, the three main ones being: i) the 3D structure of the world is not explicitly taken into account, leading to implausible outputs; ii) trying to predict jointly agents motion and appearance complicates the task further; iii) error propagation leads the output quality to degrade just a few frames in the future.

Method outline

In contrast to the aforementioned end-to-end approach, we advocate for the use of a semi-parametric framework in which prior information about the objects and the world are properly taken into account. We follow a divide-and-conquer strategy to split the overall problem of scene generation in a number of sub-problems which are well-studied in the research community: object detection, vehicle classification, keypoint localization, 3D pose estimation, trajectory prediction, image inpainting. Although these sub-problems are themselves challenging, at least algorithms and models providing robust solutions to these tasks already exist. Given the image depicting the actual scene, we can sketch the overall pipeline as follows:

1. An object detector produces the bounding boxes for all the vehicles in the scene.

2. For each bounding box, we classify the vehicle and localize its keypoints as described in Sec. 4.4.2.

3. An iterative Perspective-n-Points (PnP) algorithm is used to find the 3D pose of each vehicle, minimizing the re-projection error of the localized keypoints w.r.t. the ones of the 3D model of the predicted class.

After these three steps, a massive amount of additional information is available both about each vehicle and about the geometry of the scene. Each vehicle can now be arbitrarily moved in a geometrically plausible manner - taking the geometry of the scene into account. The next location of each vehicle can be obtained either i) predicting its future trajectory from the past locations or ii) manually manipulating the object location if we consider an interactive editing setting. Once the future location is decided, the previous one is inpainted and our proposed framework for vehicle novel view synthesis (Sec. 4.4) can be used to generate the novel appearance of the vehicle.

Visual results and ongoing work

Preliminary visual results for future scene generation are available in Fig. 4.50 and Fig. 4.51, where we show how vehicles can be stitched and moved in the image in a plausible manner. This is a currently ongoing work and we are working in these very days to produce more detailed quantitative as well as visual results; we hope to be able to formalize in our scene generation framework in a peer-reviewed publication in the next future.

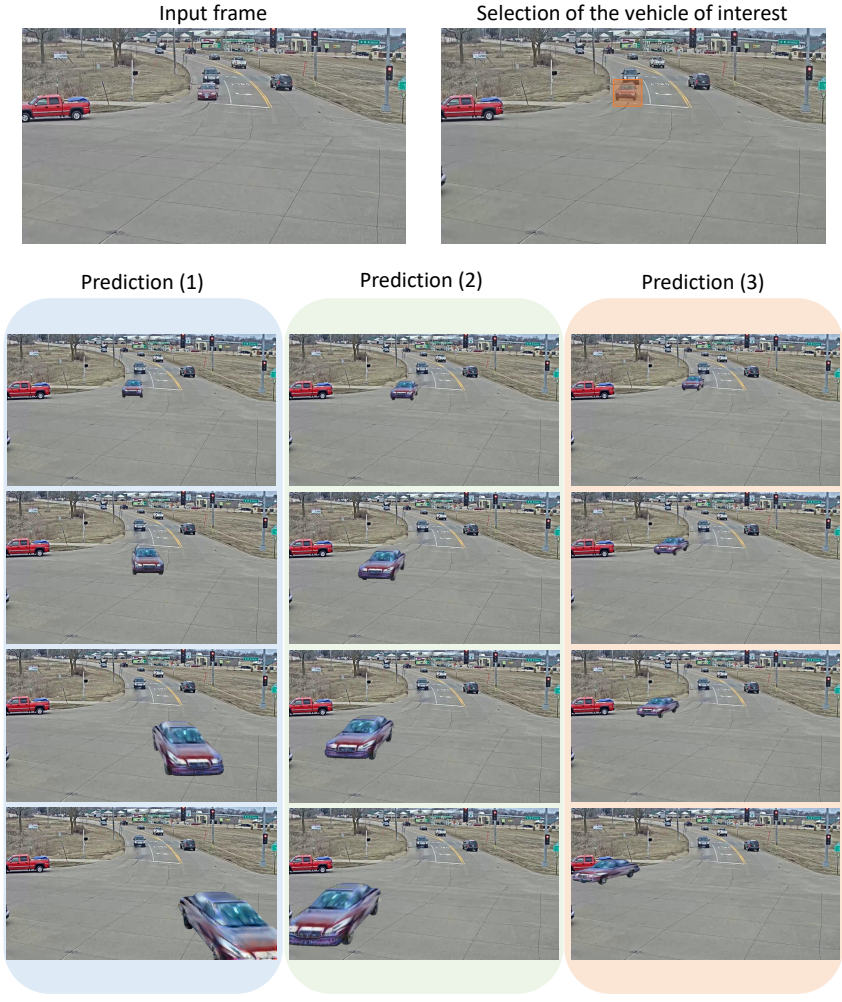


Figure 4.49: Overview of our task. We want to predict the visual appearance of the scene in the next future, considering the most likely trajectories of each vehicle.

Figure 4.50: Preliminary visual results from the proposed urban scene synthesis framework. In each scene, one vehicle is not real. Can you spot which one? Answer in Fig. 4.51.

.

Figure 4.51: Preliminary visual results from the proposed urban scene synthesis framework. In each scene, the highlighted vehicle is generated.

Chapter 5

Conclusions

The aim of this thesis - and of most of the research work carried out during the PhD - was to study the problem of visual understanding of the urban scene from multiple viewpoints. Our inquiry started from the point of view of the vehicle - investigating which parts of the scene are more likely to draw the attention of the human driver - and progressively zoomed out to the infrastructure viewpoint - first understanding vehicles' locations and poses in the world, then inferring the visual aspect of the vehicles from novel viewpoints, finally hallucinating the possible evolution of the visual appearance of the whole urban scene.

In the following we summarize the major contributions presented in this thesis, drawing a few concluding remarks on the results achieved so far.

Summary of Contributions

The `DR(eye)VE` dataset

In Sec. 3.1 we introduced a novel, publicly available dataset for driver's gaze estimation, acquired during real-world driving. Our dataset, composed by more than 500,000 frames, contains drivers' gaze fixations and their temporal integration providing task-specific saliency maps. Geo-referenced locations, driving speed and course enrich the set of released data. The `DR(eye)VE` dataset was the first publicly available dataset of this kind at the time of the publication. Three years later its first publication, it

has been downloaded several hundreds of times and fostered discussion on better understanding, exploiting and reproducing the driver's attention process in the autonomous and assisted cars of future generations.
Thanks to the DR(eye)VE dataset, we were able to perform large-scale analysis of driver's attentional behavior on real-world data. The study presented in Sec. 3.1.3 resulted in many insights about *where* and *what* the driver is looking at while driving. We analyzed what people pay attention to while driving, and which parts of the scene around the vehicle are more critical for the task. We also investigated the dynamics of the driver's gaze and use it as a proxy to understand related attentional mechanisms. In this process, we dived deep into the influence of car speed, course and the landscape over the driver's attentional behavior, as the information about which elements in the scene are likely to capture the driver's attention may benefit several applications in the context of human-vehicle interaction and driver attention analysis.

The first deep learning based model of driver's attention

In Sec. 3.2 and Sec. 3.3 we engineered, designed and trained a computational model of human attention during the driving task. First we trained a coarse-to-fine convolutional network on short sequences extracted from the DR(eye)VE dataset. Moving a step forward, we built upon the insights gained during the DR(eye)VE dataset analysis to explicitly integrate in our model the factors which most influenced the driver's attentional behavior, i.e. motion and scene semantics. To this end we proposed a new model based on a multi-branch deep architecture integrating all these three sources of information: raw RGB video, motion and scene semantics. Experimental results highlighted that several attention patterns are shared across drivers and can be reproduced to some extent. The comprehensive experimental evaluation in Sec. 3.3.3 indicated that our multi-branch model of driver's attention obtains state-of-the-art performance on a number of metrics commonly used to benchmark methods for image and video saliency. These encouraging results allowed us to envision an hypothetical new assisted driving paradigm which suggests to the driver, with no intervention, where he/she should focus his/her attention. In this perspective, besides the quantitative evaluation, we carefully designed and setup a perceptual experiment to know how a human would find the predicted attentional maps. In Sec. 3.4 we introduced a protocol to evaluate the predicted attention maps,

where we processed the predicted videoclips to approximate as realistically as possible the visual field of attention of the driver and validated our model also from a perceptual point of view.

Novel methods for vehicle (re)identification, pose and occupancy estimation

We delved into the problem of vehicle re-identification in occasion of the NVIDIA AI City Challenge 2018; our pipeline, based on a triplet network trained on automatically annotated data, was described in Sec. 4.1. Our work in vehicle occupancy estimation was introduced in Sec. 4.2, where we introduced a *learnt* semantic-aware transformation to map detections from a single dashboard camera frame onto a bird's eye occupancy map of the scene. This inferred map thus constitutes an interpretable and concise representation of the road state. We demonstrated the effectiveness of our model for occupancy estimation against several baselines and observed its ability to generalize on real-world data despite having been trained solely on synthetic ones. Indeed, the model training was enabled by the collection of a high-resolution synthetic dataset (*SVA dataset*, also publicly released) featuring a huge amount of coupled dashboard camera and bird's eye frames. These frames are annotated 'for free' with the information extracted from the GTAV game engine: precise 3D poses and spatial occupancy of the vehicles, bounding boxes, reciprocal distances. Notably, this dataset set the foundations for another dataset on pedestrian tracking - the *JTA dataset* [48] - which we collected shortly later using the same software framework. Eventually, in Sec. 4.3 we described a novel method for vehicle pose estimation, where a differentiable renderer component is used to refine the estimated pose by back-propagating the silhouette alignment error.

A novel formulation to object novel viewpoint synthesis

In Sec. 4.4 we introduced a novel formulation of the problem of object novel viewpoint synthesis in a *semi-parametric* setting. Differently from most existing methods, we designed our model to be trainable on existing datasets for 3D object detection in a self-supervised manner, with no need for paired source/target viewpoint images - although it can be complemented with synthetic data. In our proposed framework, non-parametric visual hints

act as prior information to guide a deep parametric model for generating realistic images, disentangling by design appearance and shape; this enables continuous manipulation of the viewpoint and shape transfer to different 3D models. As completing the image is much easier than generating it from scratch, an Image Completion Network (ICN) can be trained on just few thousands images from the Pascal3D+ dataset and still be able to generalize to unseen viewpoints. Although vehicles were the leading thread of our work, we demonstrated that our framework is generic enough to handle rigid objects of completely different geometric structure such as chairs. We eventually showed how both perceptual experiments results and image-quality metrics rewarded our method for its realism and the visual consistency of the synthesised object across arbitrary points of view. Finally, we outline how this method can fit into a more general framework for generating the visual appearance of the urban scene in the next future, providing preliminary yet encouraging results.

Discussion and Future Works

This thesis collects a variety of results on different facets of the task of urban scene understanding. Although many of the outcomes are exciting and might possibly trail-blaze future applications in a smart city scenario, it is worth recalling that there is often a long way from research to real-world. Here we list the most evident limitations for each line of research.

Driver attention prediction

First, the study is not conclusive about the role of the segmentation branch within the full architecture. Indeed, such a branch was introduced after a detailed analysis of the DR(eye)VE dataset highlighted a strong correlation between some semantic classes (e.g. street and cars) and the driver's eye fixations. However, the presented ablation study does not show a significant boost in performance when the segmentation branch is employed, with respect to a baseline composed of the RGB and optical flow branch. This outcome may be due to the fact that the model we employed to compute segmentation maps (Dilation-10) was pre-trained on a different dataset (i.e. Cityscapes), and showed poor generalization to the DR(eye)VE dataset. Indeed, especially in very challenging sequences (e.g. rainy, night), we observed poor predictions. Since our work was published, the research

community yielded better and better solutions for semantic segmentation, as well as new large scale datasets of urban scenes (e.g. Mapillary). It is worth questioning whether the outcomes of the ablation study would change by updating the semantic segmentation network and its pre-training.
From the application standpoint, the work could sound more complete by devising a prototypical ADAS (Advanced Driver-Assistance System) integrating the presented model for attention prediction. Such prototype could involve an head pose of an eye tracking system monitoring the driver inside the cabin, and infer where he/her is actually looking in 2 the outside scene. The prediction of the multi-branch network, providing a sort of 'expectation' describing what most drivers would pay attention to in that same situation, could be integrated in (at least) two ways:

- acting as a prior for where the driver is looking, e.g. by refining the predicted fixation point of the driver;
- detecting situations in which the driver's gaze frequently diverges from the expected, indicating potential drowsiness or distraction.

In this context, preliminary results from perceptual experiments in Sec. 3.4 should be explored further, and a few assumptions currently lying 'under the hood' discussed, the main open question being: can we trust the ability of a human observer to provide an objective evaluation of the safety level from the foveated videoclip?

Scene understanding

Starting from re-identification, our model was engineer for participating to the NVIDIA AI City Challenge 2018, and evaluation was thus conducted only on the data provided for the competition. It would be interesting to measure the method performance on more standard academic benchmarks such as [83, 203]. As far as our proposed model for learning to transform detections from dashboard view to bird's eye view, more exhaustive experiments would be necessary to prove the capacity of the model to generalize on real-world data. Indeed, the model is trained only on synthetic data and generalization on real-world data is measured only qualitatively, as the ground truth is difficult to provide (this was the reason why we resorted to synthetic data for training in the first place). Eventually, the main bottleneck and open issue of our work in pose estimation is probably related

to the performance of the differentiable renderer. Indeed, our implemented rendered can only render the grayscale silhouette of the objects, which clearly poses a lot of constraints on the architecture and performance of the overall pipeline. Recently, major deep learning frameworks released branches providing differentiable layers for working with 3D data (e.g. TensorFlow Graphics, PyTorch3D). It would be definitely very interesting to test our proposed pipeline for object pose refinement using one of the much more powerful differentiable renderers provided by these frameworks.

Object/scene novel viewpoint synthesis

While we find that our semi-parametric framework for novel view synthesis introduces several improvements over the state of the art, there are also notable issues which are still open. First, in all experiments we set our output resolution to 128x128 pixels. Although this choice enabled us to run a number of experiments in a reasonable time, it is definitely necessary to provide results to higher resolutions. Also, most of the evaluation was conducted on only two classes of rigid objects, namely vehicles and chairs; a more satisfactory evaluation would necessarily need to consider additional object classes. In the mid to long-term, it would be interesting to come up with a solution for applying this intuition of semi-parametric synthesis to natural (not man-made) objects. How to apply this method to the generation of the whole urban scene is still subject to debate. While we already designed and trained a model and we provided a few encouraging preliminary results in the last chapter of the thesis, additional research and extensive experiments are surely needed before this method can be called solid.

Notable Achievements

The research work on driver's focus of attention led to several publications in conferences and it eventually made its way in the top-tier journal of Transaction on Pattern Analysis and Machine Intelligence (TPAMI). Our paper about learning a mapping between dashboard and bird's eye views [131] has been awarded the special mention prize in the International Conference on Image Analysis and Processing (ICIAP). Furthermore, several datasets have been collected and publicly released to the research community during these years: the `DR(eye)VE` dataset [4, 129], the SVA dataset [131], the

JTA dataset [48]; all three datasets have met with a significant interest in the community and have been downloaded several hundreds of times from all over the world in these years.

In the following pages, we also report the complete list of our publications. The reader may observe that some of them were not discussed in the previous chapters, as they did not fall under the leitmotiv of the thesis.

Appendix A

List of publications

In this section we briefly report the research papers published during my PhD period, as well as some pre-prints which are currently under review. Some of them did not make it in the final thesis, either because they have been improved or replaced by a successive work, either because their topic did not overlap with the main flow of this thesis.

Content and experimental results published in some of this papers has been included, even *verbatim*, in the previous chapters.

- Alletto, S., Palazzi, A., Solera, F., Calderara, S. and Cucchiara, R., 2016. Dr (eye) ve: a dataset for attention-based tasks with applications to autonomous and assisted driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops* (pp. 54-60).

- Palazzi, A., Calderara, S., Bicocchi, N., Vezzali, L., di Bernardo, G.A., Zambonelli, F. and Cucchiara, R., 2016, September. Spotting prejudice with nonverbal behaviours. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing* (pp. 853-862). ACM.

- Di Bernardo, G.A., Vezzali, L., Palazzi, A., Calderara, S., Bicocchi, N., Zambonelli, F., Cucchiara, R. and Cadamuro, A., 2017. A new era in the study of intergroup nonverbal behaviour: Studying intergroup

dyadic interactions "online". In *18th General Meeting of the European Association of Social Psychology.*

- Palazzi, A., Solera, F., Calderara, S., Alletto, S. and Cucchiara, R., 2017, June. Learning where to attend like a human driver. In *2017 IEEE Intelligent Vehicles Symposium (IV)* (pp. 920-925). IEEE.

- Palazzi, A., Borghi, G., Abati, D., Calderara, S. and Cucchiara, R., 2017, September. Learning to map vehicles into bird's eye view. In *International Conference on Image Analysis and Processing (pp. 233-243).* Springer, Cham.

- Cornia, M., Abati, D., Baraldi, L., Palazzi, A., Calderara, S. and Cucchiara, R., 2017, November. Attentive models in vision: computing saliency maps in the deep learning era. In *Conference of the Italian Association for Artificial Intelligence* (pp. 387-399). Springer, Cham.

- Cornia, M., Abati, D., Baraldi, L., Palazzi, A., Calderara, S. and Cucchiara, R., 2018. Attentive models in vision: Computing saliency maps in the deep learning era. *Intelligenza Artificiale*, 12(2), pp.161-175.

- Fabbri, M., Lanzi, F., Calderara, S., Palazzi, A., Vezzani, R. and Cucchiara, R., 2018. Learning to detect and track visible and occluded body joints in a virtual world. In *Proceedings of the European Conference on Computer Vision (ECCV)* (pp. 430-446).

- Antonio Marin-Reyes, P., Palazzi, A., Bergamini, L., Calderara, S., Lorenzo-Navarro, J. and Cucchiara, R., 2018. Unsupervised vehicle re-identification using triplet networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops* (pp. 166-171).

- Di Bernardo, G.A., Vezzali, L., Giovannini, D., Palazzi, A., Calderara, S., Bicocchi, N., Zambonelli, F., Cucchiara, R., Cadamuro, A. and Cocco, V.M., 2018. Comportamento non verbale intergruppi "oggettivo": una replica dello studio di Dovidio, kawakami e Gaertner (2002). In *XV Congresso Nazionale della Sezione di Psicologia Sociale dell'Associazione Italiana di Psicologia.*

- Palazzi, A., Bergamini, L., Calderara, S. and Cucchiara, R., 2018. End-to-end 6-DoF Object Pose Estimation through Differentiable Rasterization. In *Proceedings of the European Conference on Computer Vision (ECCV)* (pp. 0-0).

- Palazzi, A., Abati, D., Solera, F. and Cucchiara, R., 2018. Predicting the Driver's Focus of Attention: the DR (eye) VE Project. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(7), pp.1720-1733.

- Bergamini, L., Trachtman, A.R., Palazzi, A., Del Negro, E., Dondona, A.C., Marruchella, G. and Calderara, S., 2019, September. Segmentation Guided Scoring of Pathological Lesions in Swine Through CNNs. In International Conference on Image Analysis and Processing (pp. 352-360). Springer, Cham.

- Bicocchi, N., Calderara, S., Porrello, A., Palazzi, A., Di Bernardo, G. A., Vezzali, L., Cucchiara, R., Zambonelli, F., 2019. Revealing implicit forms of prejudice with automated analysis of nonverbal behaviours. (under review)

- Palazzi, A., Bergamini, L., Calderara, S. and Cucchiara, R., 2019. Warp and Learn: Novel Views Generation for Vehicles and Other Objects, *arXiv preprint* arXiv:1907.10634. (under review)

Appendix B

Activities carried out during the PhD

Besides the research activities described in this thesis and those listed in Appendix A, I also took part in other teaching and service activities which are briefly reported below.

Participation in projects

- National project "Città educante" (ctn01 00034 393801) of the National Technological Cluster on Smart Communities cofunded by the Italian Ministry of Education, University and Research - MIUR.
- MIUR PRIN project *"PREVUE: PRediction of activities and Events by Vision in an Urban Environment"*, grant ID E94I19000650001.
- IARPA Deep Intermodal Video Analytics (DIVA) (IARPA-BAA-16-13) on development of robust automated activity detection in a multi-camera streaming video environment.
- Modena Automotive Smart Area (MASA) - open air test bed for the experimentation and certification of autonomous driving and connected driving technologies.

Foreign collaborations

- Joint participation to NVIDIA AI City Challenge 2018 with Dr. Pedro A. Marìn Reyes, University of Las Palmas de Gran Canaria (Spain) - track of vehicle re-identification.
- Research internship in the Amazon computer vision team. Berlin (Germany), May - August 2018.

Teaching activities

- Lecturer for the *Deep Learning* postgraduate course in Master of Visual Computing (2017)
- Laboratory lecturer for the *Computer Vision* graduate course, Prof. Cucchiara, at University of Modena and Reggio Emilia
- Laboratory lecturer for the *Machine Learning and Pattern Recognition* graduate course, Prof. Calderara, at University of Modena and Reggio Emilia
- Laboratory lecturer for the *Neural Network Computing, AI and Machine Learning for Automotive* graduate course, Prof. Cucchiara, at University of Modena and Reggio Emilia

Grants, within the AImagelab group

- Italian Supercomputing Resource Allocation (ISCRA) Grant from CINECA, for accessing the Galileo HPC Platform (from 2015 to 2017).
- Facebook AI Partnership, with the donation of a GPU-based server.

Thesis co-advisor

- Moving object detection via deep autoencoder - Angelo Porrello (MSc).

- Automatic extraction of pedestrian joints from videogames - Marco Gianelli (BSc).
- Convolutional Neural Networks for Vehicle Model Classification - Paolo Bertellini (BSc).
- Future urban scene generation: a deep learning approach for a 3D temporal vehicle reconstruction starting from monocular images - Alessandro Simoni (MSc).

Other academic services

- Reviewer for Robotics and Automation Letters.
- Site admin and events organizer for Master of Visual Computing and Multimedia Technologies (MUMET), 2017.

Conferences, courses, seminars attended

Conferences

- ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp 2016) - Heidelberg, Germany.
- International Conference on Computer Vision - ICCV, Venice, 2017
- International Conference on Image Analysis and Processing (ICIAP) 2019, 9-13 September - Trento Italy.

Courses and seminars

- Summer School Regularization Methods for Machine Learning (RegML) 2016, held by Lorenzo Rosasco at IIT (Genoa).
- Udacity Self-Driving Car Nanodegree Program.
- Academic English Workshop I - Dr. Silvia Cavalieri, University of Modena and Reggio Emilia.
- Academic English Workshop II - Dr. Silvia Cavalieri, University of Modena and Reggio Emilia.

- Faces, deep learning and the pursuit of training data - Prof. Tal Hassner, Open University of Israel - May 17th, 2016.
- Multi-camera tracking: following people in large camera networks - Dr. Ergys Ristani - October 18th, 2016.
- Syncronization problems in computer vision - Prof. Andrea Fusiello - October 21st 2016.
- Internet privacy: towards more transparency - Balachander Krishnamurthy - November 21st 2016.
- The eye of the machine. - Prof. Simone Arcagni, University of Palermo - September 9th, 2017.
- Security and quantistic technology: potential uses and risks - Dr. Enrico Prati, CNR - November 21st, 2017.
- Deep learning technologies: from hardware components to vertical frameworks - Dr. Piero Altoè, NVIDIA - November 29th, 2017.
- Visual appearance acquisition of real objects - Dr. Massimiliano Corsini, CNR - February 8th, 2018.
- Computational Aspects of Deep Reinforcement Learning - Dr. Iuri Frosio, NVIDIA Research - July 15th, 2019.

Bibliography

[1] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk. Frequency-tuned salient region detection. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1597–1604, 2009. 4

[2] Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M Seitz, and Richard Szeliski. Building rome in a day. *Communications of the ACM*, 54(10):105–112, 2011. 8

[3] Ruano Aitor. DeepGTAV. *Software available at https://github.com/aitorzip/DeepGTAV*, 2017. 81

[4] Stefano Alletto, Andrea Palazzi, Francesco Solera, Simone Calderara, and Rita Cucchiara. Dr(eye)ve: a dataset for attention-based tasks with applications to autonomous and assisted driving. In *IEEE Internation Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 54–60, 2016. 18, 20, 34, 38, 89, 150

[5] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International Conference on Machine Learning*, pages 214–223, 2017. 11

[6] Mathieu Aubry, Daniel Maturana, Alexei A Efros, Bryan C Russell, and Josef Sivic. Seeing 3d chairs: exemplar part-based 2d-3d alignment using a large dataset of cad models. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3762–3769, 2014. 8

[7] Lorenzo Baraldi, Costantino Grana, and Rita Cucchiara. Hierarchical boundary-aware neural encoder for video captioning. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017. 45

[8] Loris Bazzani, Hugo Larochelle, and Lorenzo Torresani. Recurrent mixture density network for spatiotemporal visual attention. In *International Conference on Learning Representations (ICLR)*, 2017. 4, 53, 54, 62

[9] Nicola Bernini, Massimo Bertozzi, Luca Castangia, Marco Patander, and Mario Sabbatelli. Real-time obstacle detection using stereo vision for autonomous ground vehicles: A survey. In *Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on*, pages 873–878. IEEE, 2014. 80

[10] Alexander Blade. Script Hook V. *Software available at http://www.dev-c.com/gtav/scripthookv/*, 2017. 81

[11] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Blender Institute, Amsterdam, 2017. 11, 97

[12] Guido Borghi, Marco Venturelli, Roberto Vezzani, and Rita Cucchiara. Poseidon: Face-from-depth for driver pose estimation. In *CVPR*, 2017. 5

[13] Ali Borji, Mengyang Feng, and Huchuan Lu. Vanishing point attracts gaze in free-viewing and visual search tasks. *Journal of vision*, 16(14):18–18, 2016. 23

[14] Ali Borji and Laurent Itti. State-of-the-art in visual attention modeling. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):185–207, 2013. 4

[15] Ali Borji and Laurent Itti. Cat2000: A large scale fixation dataset for boosting saliency research. *CVPR 2015 workshop on "Future of Datasets"*, 2015. arXiv preprint arXiv:1505.03581. 6

[16] Ali Borji, Dicky N Sihite, and Laurent Itti. What/where to look next? modeling top-down visual attention in complex interactive environments. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 44(5):523–538, 2014. 4

[17] Edmond Boyer and Jean-Sébastien Franco. A hybrid approach for computing visual hulls of complex objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 695–701. IEEE Computer Society Press, 2003. 8

[18] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000. 136

[19] R. Brémond, J.-M. Auberlet, V. Cavallo, L. Désiré, V. Faure, S. Lemonnier, R. Lobjois, and J.-P. Tarel. Where we look when we drive: A multidisciplinary approach. In *Proceedings of Transport Research Arena (TRA'14)*, 2014. http://perso.lcpc.fr/tarel.jean-philippe/publis/tra14b.html. 5, 6, 26

[20] Claus Bundesen and Axel Larsen. Visual transformation of size. *Journal of Experimental Psychology: Human Perception and Performance*, 1(3):214, 1975. 104

[21] Zoya Bylinskii, Tilke Judd, Ali Borji, Laurent Itti, Frédo Durand, Aude Oliva, and Antonio Torralba. Mit saliency benchmark. http://saliency.mit.edu/. 6, 15, 19, 25, 26, 38, 42, 54, 55

[22] Zoya Bylinskii, Tilke Judd, Aude Oliva, Antonio Torralba, and Frédo Durand. What do different evaluation metrics tell us about saliency models? *arXiv:1604.03605*, 2016. 38, 51

[23] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 7, 8, 11, 96, 115, 125

[24] Gaurav Chaurasia, Sylvain Duchene, Olga Sorkine-Hornung, and George Drettakis. Depth synthesis and local warps for plausible image-based navigation. *ACM Transactions on Graphics (TOG)*, 32(3):30, 2013. 105

[25] Gaurav Chaurasia, Olga Sorkine, and George Drettakis. Silhouette-aware warping for image-based rendering. In *Computer Graphics Forum*, volume 30, pages 1223–1232. Wiley Online Library, 2011. 105

[26] Chenyi Chen, Ari Seff, Alain Kornhauser, and Jianxiong Xiao. Deepdriving: Learning affordance for direct perception in autonomous driving. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2722–2730, 2015. 11, 80

[27] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2018. 7

[28] Qifeng Chen and Vladlen Koltun. Photographic image synthesis with cascaded refinement networks. In *IEEE International Conference on Computer Vision (ICCV)*, volume 1, page 3, 2017. 113, 119, 120

[29] Tao Chen, Ming-Ming Cheng, Ping Tan, Ariel Shamir, and Shi-Min Hu. Sketch2photo: Internet image montage. In *ACM transactions on graphics (TOG)*, volume 28, page 124. ACM, 2009. 105

[30] Xiaozhi Chen, Kaustav Kundu, Yukun Zhu, Andrew G Berneshawi, Huimin Ma, Sanja Fidler, and Raquel Urtasun. 3d object proposals for accurate object class detection. In *Advances in Neural Information Processing Systems*, pages 424–432, 2015. 14, 43

[31] De Cheng, Yihong Gong, Sanping Zhou, Jinjun Wang, and Nanning Zheng. Person re-identification by multi-channel parts-based cnn with improved triplet loss function. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1335–1344, 2016. 7

[32] M. M. Cheng, N. J. Mitra, X. Huang, P. H. S. Torr, and S. M. Hu. Global contrast based salient region detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(3):569–582, 2015. 4

[33] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8789–8797, 2018. 11

[34] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *European Conference on Computer Vision*, pages 628–644. Springer, 2016. 8

[35] Alvaro Collet, Dmitry Berenson, Siddhartha S Srinivasa, and Dave Ferguson. Object recognition and full pose registration from a single image for robotic manipulation. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 48–55. IEEE, 2009. 9

[36] Alvaro Collet, Manuel Martinez, and Siddhartha S Srinivasa. The moped framework: Object recognition and pose estimation for manipulation. *The International Journal of Robotics Research*, 30(10):1284–1306, 2011. 9

[37] Marcella Cornia, Lorenzo Baraldi, Giuseppe Serra, and Rita Cucchiara. A Deep Multi-Level Network for Saliency Prediction. In *International Conference on Pattern Recognition (ICPR)*, 2016. 4, 53, 54, 62

[38] Marcella Cornia, Lorenzo Baraldi, Giuseppe Serra, and Rita Cucchiara. Predicting human eye fixations via an lstm-based saliency attentive model. *arXiv preprint arXiv:1611.09571*, 2016. 4

[39] Martin Danelljan, Gustav Häger, Fahad Khan, and Michael Felsberg. Accurate scale estimation for robust visual tracking. In *British Machine Vision Conference*. BMVA Press, 2014. 74

[40] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009. 7, 77, 85, 86, 119

[41] Yanchao Dong, Zhencheng Hu, Keiichi Uchimura, and Nobuki Murayama. Driver inattention monitoring system for intelligent vehicles: A review. *IEEE transactions on intelligent transportation systems*, 12(2):596–614, 2011. 80

[42] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In

Proceedings of the 1st Annual Conference on Robot Learning, pages 1–16, 2017. 11

[43] Alexey Dosovitskiy, Jost Tobias Springenberg, Maxim Tatarchenko, and Thomas Brox. Learning to generate chairs, tables and cars with convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(4):692–705, 2017. 8

[44] Xinxin Du, Marcelo H Ang, Sertac Karaman, and Daniela Rus. A general pipeline for 3d detection of vehicles. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3194–3200. IEEE, 2018. 97

[45] Lior Elazary and Laurent Itti. A bayesian model for efficient visual search and recognition. *Vision research*, 50(14):1338–1352, 2010. 4

[46] Patrick Esser, Ekaterina Sutter, and Björn Ommer. A variational u-net for conditional appearance and shape generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8857–8866, 2018. 11, 105, 116, 117, 118, 120, 121, 124

[47] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010. 115

[48] Matteo Fabbri, Fabio Lanzi, Simone Calderara, Andrea Palazzi, Roberto Vezzani, and Rita Cucchiara. Learning to detect and track visible and occluded body joints in a virtual world. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 430–446, 2018. 147, 151

[49] Di Feng, Lars Rosenbaum, and Klaus Dietmayer. Towards safe autonomous driving: Capture uncertainty in the deep neural network for lidar 3d vehicle detection. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 3266–3273. IEEE, 2018. 103, 106

[50] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and

automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 19

[51] Andrew Fitzgibbon and Andrew Zisserman. Automatic 3d model acquisition and generation of new images from video sequences. In *Signal Processing Conference (EUSIPCO 1998), 9th European*, pages 1–8. IEEE, 1998. 8

[52] Lex Fridman, Philipp Langhans, Joonbum Lee, and Bryan Reimer. Driver gaze estimation without using eye movement. *arXiv:1507.04760*, 2015. 5, 6, 18

[53] Simone Frintrop, Erich Rome, and Henrik I Christensen. Computational visual attention systems and their cognitive foundations: A survey. *ACM Transactions on Applied Perception (TAP)*, 7(1):6, 2010. 43

[54] Björn Fröhlich, Markus Enzweiler, and Uwe Franke. Will this car change the lane? turn signal recognition in the frequency domain. In *IEEE Intelligent Vehicles Symposium*, pages 37–42, 2014. 33

[55] Matheus Gadelha, Subhransu Maji, and Rui Wang. 3d shape induction from 2d views of multiple objects. *3D Vision*, 2017. 8, 91, 93

[56] Adrien Gaidon, Qiao Wang, Yohann Cabon, and Eleonora Vig. Virtual worlds as proxy for multi-object tracking analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4340–4349, 2016. 11

[57] Dashan Gao, Vijay Mahadevan, and Nuno Vasconcelos. On the plausibility of the discriminant centersurround hypothesis for visual saliency. *Journal of Vision*, pages 1–18, 2008. 4

[58] Howard Gardner. *Frames of mind: The theory of multiple intelligences*. Hachette UK, 2011. 104

[59] Golnaz Ghiasi and Charless C Fowlkes. Laplacian pyramid reconstruction and refinement for semantic segmentation. In *European Conference on Computer Vision*, pages 519–534. Springer, 2016. 80

[60] Ross Girshick. Fast r-cnn. *arXiv preprint arXiv:1504.08083*, 2015. 74

[61] Theodosios Gkamas and Christophoros Nikou. Guiding optical flow estimation using superpixels. In *Digital Signal Processing (DSP), 2011 17th International Conference on*, pages 1–6. IEEE, 2011. 50, 52

[62] Stas Goferman, Lihi Zelnik-Manor, and Ayellet Tal. Context-aware saliency detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(10):1915–1926, 2012. 4

[63] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016. 118

[64] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 11, 104

[65] Steven J Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F Cohen. The lumigraph. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 43–54. ACM, 1996. 8

[66] Rudolf Groner, Franziska Walder, and Marina Groner. Looking at faces: Local and global aspects of scanpaths. *Advances in Psychology*, 22:523–533, 1984. 20

[67] James Hays and Alexei A Efros. Scene completion using millions of photographs. *ACM Transactions on Graphics (TOG)*, 26(3):4, 2007. 105

[68] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision (ICCV)*, pages 2961–2969, 2017. 109

[69] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 7, 85

[70] Peter Hedman, Tobias Ritschel, George Drettakis, and Gabriel Brostow. Scalable inside-out image-based rendering. *ACM Transactions on Graphics (TOG)*, 35(6):231, 2016. 105

[71] John M Henderson. Human gaze control during real-world scene perception. *Trends in cognitive sciences*, 7(11):498–504, 2003. 20

[72] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, pages 6626–6637, 2017. xiii, xiv, 116, 119, 124

[73] Elad Hoffer and Nir Ailon. Deep metric learning using triplet network. In *International Workshop on Similarity-Based Pattern Recognition*, pages 84–92. Springer, 2015. 7, 77

[74] X. Huang, C. Shen, X. Boix, and Q. Zhao. Salicon: Reducing the semantic gap in saliency prediction by adapting deep neural networks. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 262–270, 2015. 4

[75] Du Q Huynh. Metrics for 3d rotations: Comparison and analysis. *Journal of Mathematical Imaging and Vision*, 35(2):155–164, 2009. 98

[76] Phillip Isola and Ce Liu. Scene collaging: Analysis and synthesis of natural images with semantic layers. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3048–3055, 2013. 105

[77] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5967–5976. IEEE, 2017. 11, 113, 139

[78] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Advances in neural information processing systems*, pages 2017–2025, 2015. 8, 91, 95

[79] Ashesh Jain, Hema S Koppula, Bharad Raghavan, Shane Soh, and Ashutosh Saxena. Car that knows before you do: Anticipating maneuvers via learning temporal driving models. In *Proceedings of the IEEE international conference on computer vision (ICCV)*, pages 3182–3190, 2015. 33

[80] Ming Jiang, Shengsheng Huang, Juanyong Duan, and Qi Zhao. Salicon: Saliency in context. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 6, 15, 42

[81] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016. 113

[82] Micah K Johnson, Kevin Dale, Shai Avidan, Hanspeter Pfister, William T Freeman, and Wojciech Matusik. Cg2real: Improving the realism of computer generated images using a large collection of photographs. *IEEE Transactions on Visualization and Computer Graphics*, 17(9):1273–1285, 2011. 105

[83] Aytaç Kanacı, Xiatian Zhu, and Shaogang Gong. Vehicle re-identification in context. In *German Conference on Pattern Recognition*, pages 377–390. Springer, 2018. 149

[84] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1725–1732, 2014. 35, 44

[85] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *International Conference on Learning Representations (ICLR)*, 2018. 11

[86] Natasha Kholgade, Tomas Simon, Alexei Efros, and Yaser Sheikh. 3d object manipulation in a single photograph using stock 3d models. *ACM Transactions on Graphics (TOG)*, 33(4):127, 2014. 12

[87] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015. 38, 52, 86, 101, 117

[88] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations (ICLR)*, 2014. 11, 104, 117

[89] Kalin Kolev, Maria Klodt, Thomas Brox, and Daniel Cremers. Continuous global optimization in multiview 3d reconstruction. *International Journal of Computer Vision*, 84(1):80–96, 2009. 8

[90] Puneet Kumar, Mathias Perrollaz, Stéphanie Lefevre, and Christian Laugier. Learning-based approach for online lane change intention prediction. In *IEEE Intelligent Vehicles Symposium*, pages 797–802, 2013. 33

[91] Matthias Kümmerer, Lucas Theis, and Matthias Bethge. Deep gaze I: boosting saliency prediction with feature maps trained on imagenet. *CoRR*, abs/1411.1045, 2014. 4, 26

[92] Matthias Kümmerer, Thomas SA Wallis, and Matthias Bethge. Information-theoretic model comparison unifies saliency metrics. *Proceedings of the National Academy of Sciences*, 112(52):16054–16059, 2015. 51

[93] Jean-François Lalonde, Derek Hoiem, Alexei A Efros, Carsten Rother, John Winn, and Antonio Criminisi. Photo clip art. In *ACM transactions on graphics (TOG)*, volume 26, page 3. ACM, 2007. 105

[94] Adam M Larson and Lester C Loschky. The contributions of central versus peripheral vision to scene gist recognition. *Journal of Vision*, 9(10):6–6, 2009. 63

[95] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Epnp: An accurate o (n) solution to the pnp problem. *International journal of computer vision*, 81(2):155, 2009. 9

[96] Dan Levi, Noa Garnett, Ethan Fetaya, and Israel Herzlyia. Stixelnet: A deep convolutional network for obstacle detection and road segmentation. In *BMVC*, pages 109–1, 2015. 80

[97] Joseph J Lim, Aditya Khosla, and Antonio Torralba. Fpm: Fine pose parts-based model with 3d cad models. In *European Conference on Computer Vision*, pages 478–493. Springer, 2014. 8

[98] Joseph J. Lim, Hamed Pirsiavash, and Antonio Torralba. Parsing IKEA Objects: Fine Pose Estimation. *ICCV*, 2013. 109, 132

[99] Chien-Chuan Lin and Ming-Shi Wang. A vision based top-view transformation model for a vehicle parking assistant. *Sensors*, 12(4):4431–4446, 2012. 10

[100] Guosheng Lin, Chunhua Shen, Anton van den Hengel, and Ian Reid. Efficient piecewise training of deep structured models for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3194–3203, 2016. 80

[101] Hongye Liu, Yonghong Tian, Yaowei Yang, Lu Pang, and Tiejun Huang. Deep relative distance learning: Tell the difference between similar vehicles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2167–2175, 2016. 7

[102] Nian Liu, J. Han, D. Zhang, Shifeng Wen, and T. Liu. Predicting eye fixations using convolutional neural networks. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pages 362–370, 2015. 4

[103] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European Conference on Computer Vision*, pages 21–37. Springer, 2016. 74, 89

[104] Xinchen Liu, Wu Liu, Huadong Ma, and Huiyuan Fu. Large-scale vehicle re-identification in urban surveillance videos. In *2016 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2016. 6, 103, 106

[105] Xinchen Liu, Wu Liu, Tao Mei, and Huadong Ma. Provid: Progressive and multimodal vehicle reidentification for large-scale urban surveillance. *IEEE Transactions on Multimedia*, 20(3):645–658, 2018. 7

[106] Yu-Chih Liu, Kai-Ying Lin, and Yong-Sheng Chen. Bird's-eye view vision system for vehicle surrounding monitoring. In *International Workshop on Robot Vision*, pages 207–218. Springer, 2008. 10

[107] Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. Deepfashion: Powering robust clothes recognition and retrieval with

rich annotations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1096–1104, 2016. 7, 11

[108] Jonathan L Long, Ning Zhang, and Trevor Darrell. Do convnets learn correspondence? In *Advances in Neural Information Processing Systems*, pages 1601–1609, 2014. 9

[109] Matthew M Loper and Michael J Black. Opendr: An approximate differentiable renderer. In *European Conference on Computer Vision*, pages 154–169. Springer, 2014. 91

[110] David G Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999. 7, 19

[111] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004. 8, 9

[112] Mario Lucic, Karol Kurach, Marcin Michalski, Sylvain Gelly, and Olivier Bousquet. Are gans created equal? a large-scale study. In *Advances in neural information processing systems*, pages 698–707, 2018. 119

[113] Liqian Ma, Xu Jia, Qianru Sun, Bernt Schiele, Tinne Tuytelaars, and Luc Van Gool. Pose guided person image generation. In *Advances in Neural Information Processing Systems*, pages 406–416, 2017. 11

[114] Yu-Fei Ma and Hong-Jiang Zhang. Contrast-based image attention analysis by using fuzzy growing. In *Proceedings of the eleventh ACM international conference on Multimedia*, pages 374–381. ACM, 2003. 4

[115] SK Mannan, KH Ruddock, and DS Wooding. Fixation sequences made during visual examination of briefly presented 2d images. *Spatial vision*, 11(2):157–178, 1997. 20

[116] Pedro Antonio Marín-Reyes, Luca Bergamini, Javier Lorenzo-Navarro, Andrea Palazzi, Simone Calderara, and Rita Cucchiara. Unsupervised vehicle re-identification using triplet networks. In *2018 IEEE/CVF*

Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pages 166–1665. IEEE, 2018. 103, 106

[117] S. Mathe and C. Sminchisescu. Actions in the eye: Dynamic gaze datasets and learnt saliency models for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(7):1408–1424, 2015. 4, 6, 16, 39, 54, 62

[118] Thomas Mauthner, Horst Possegger, Georg Waltner, and Horst Bischof. Encoding based saliency detection for videos and images. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 4

[119] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014. 11

[120] Francesc Moreno-Noguer, Vincent Lepetit, and Pascal Fua. Accurate non-iterative o (n) solution to the pnp problem. In *Computer vision, 2007. ICCV 2007. IEEE 11th international conference on*, pages 1–8. IEEE, 2007. 9

[121] Brendan Morris, Anup Doshi, and Mohan Trivedi. Lane change intent prediction for driver assistance: On-road design and evaluation. In *IEEE Intelligent Vehicles Symposium*, pages 895–901, 2011. 33

[122] Arsalan Mousavian, Dragomir Anguelov, John Flynn, and Jana Kosecka. 3d bounding box estimation using deep learning and geometry. In *CVPR*, 2017. 43

[123] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European Conference on Computer Vision*, pages 483–499. Springer, 2016. 9

[124] Frank Nielsen. Surround video: a multihead camera approach. *The visual computer*, 21(1):92–103, 2005. 10

[125] David Nilsson and Cristian Sminchisescu. Semantic video segmentation by gated recurrent flow propagation. *arXiv preprint arXiv:1612.08871*, 2016. 43

[126] U.S. Department of Transportation. Federal automated vehicles policy. https://www.transportation.gov/sites/dot.gov/files/docs/AV%20policy%20guidance%20PDF.pdf, 2016. 33

[127] Rodrigo Ortiz-Cayon, Abdelaziz Djelouah, and George Drettakis. A bayesian approach for selective image-based rendering using superpixels. In *International Conference on 3D Vision-3DV*, 2015. 105

[128] Rodrigo Ortiz-Cayon, Abdelaziz Djelouah, Francisco Massa, Mathieu Aubry, and George Drettakis. Automatic 3d car model alignment for mixed image-based rendering. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 286–295. IEEE, 2016. 105

[129] Andrea Palazzi, Davide Abati, Francesco Solera, Rita Cucchiara, et al. Predicting the driver's focus of attention: the dr (eye) ve project. *IEEE transactions on pattern analysis and machine intelligence*, 41(7):1720–1733, 2018. 150

[130] Andrea Palazzi, Luca Bergamini, Simone Calderara, and Rita Cucchiara. End-to-end 6-dof object pose estimation through differentiable rasterization. In *Second Workshop on 3D Reconstruction Meets Semantics (3DRMS)*, 2018. 9

[131] Andrea Palazzi, Guido Borghi, Davide Abati, Simone Calderara, and Rita Cucchiara. Learning to map vehicles into bird's eye view. In *International Conference on Image Analysis and Processing*, pages 233–243. Springer, 2017. 150

[132] Andrea Palazzi, Francesco Solera, Simone Calderara, Stefano Alletto, and Rita Cucchiara. Where should you attend while driving? In *IEEE Intelligent Vehicles Symposium Proceedings*, 2017. 53, 54, 62

[133] Pingbo Pan, Zhongwen Xu, Yi Yang, Fei Wu, and Yueting Zhuang. Hierarchical recurrent neural encoder for video representation with application to captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1029–1038, 2016. 45

[134] Eunbyung Park, Jimei Yang, Ersin Yumer, Duygu Ceylan, and Alexander C Berg. Transformation-grounded image generation network for novel 3d view synthesis. In *2017 IEEE Conference on Computer*

Vision and Pattern Recognition (CVPR), pages 702–711. IEEE, 2017. 11, 12, 105, 116, 117, 118, 122, 124

[135] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. 118

[136] Georgios Pavlakos, Xiaowei Zhou, Aaron Chan, Konstantinos G Derpanis, and Kostas Daniilidis. 6-dof object pose from semantic keypoints. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2011–2018. IEEE, 2017. 9, 98, 109, 117

[137] Jeffrey S Perry and Wilson S Geisler. Gaze-contingent real-time simulation of arbitrary visual fields. In *Human vision and electronic imaging*, volume 57, 2002. 63, 64, 65

[138] Robert J Peters and Laurent Itti. Beyond bottom-up: Incorporating task-dependent influences into a computational model of spatial attention. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007. 4

[139] Robert J Peters and Laurent Itti. Applying computational tools to predict gaze direction in interactive visual environments. *ACM Transactions on Applied Perception (TAP)*, 5(2):9, 2008. 4

[140] Marc Pollefeys, Reinhard Koch, Maarten Vergauwen, and Luc Van Gool. Metric 3d surface reconstruction from uncalibrated image sequences. In *European Workshop on 3D Structure from Multiple Images of Large-Scale Environments*, pages 139–154. Springer, 1998. 8

[141] Michael I Posner, Robert D Rafal, Lisa S Choate, and Jonathan Vaughan. Inhibition of return: Neural basis and function. *Cognitive neuropsychology*, 2(3):211–228, 1985. 20

[142] Nicolas Pugeault and Richard Bowden. How much of driving is preattentive? *IEEE Transactions on Vehicular Technology*, 64(12):5424–5438, 2015. 5, 6, 18

[143] Xiaojuan Qi, Qifeng Chen, Jiaya Jia, and Vladlen Koltun. Semi-parametric image synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8808–8816, 2018. 12, 105, 113, 120

[144] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *International Conference on Learning Representations (ICLR)*, 2016. 11

[145] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 7

[146] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017. 74

[147] Konstantinos Rematas, Chuong H Nguyen, Tobias Ritschel, Mario Fritz, and Tinne Tuytelaars. Novel views of objects from a single image. *IEEE transactions on pattern analysis and machine intelligence*, 39(8):1576–1590, 2017. 12

[148] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 74

[149] Danilo Jimenez Rezende, SM Ali Eslami, Shakir Mohamed, Peter Battaglia, Max Jaderberg, and Nicolas Heess. Unsupervised learning of 3d structure from images. In *Advances In Neural Information Processing Systems*, pages 4996–5004, 2016. 8

[150] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *European Conference on Computer Vision*, pages 102–118. Springer, 2016. 10

[151] Joceline Rogé, Thierry Pébayle, Elina Lambilliotte, Florence Spitzenstetter, Daniele Giselbrecht, and Alain Muzet. Influence of age, speed

and duration of monotonous driving task in traffic on the driver's useful visual field. *Vision research*, 44(23):2737–2744, 2004. 24

[152] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 117

[153] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3234–3243, 2016. 10

[154] Dmitry Rudoy, Dan B. Goldman, Eli Shechtman, and Lihi Zelnik-Manor. Learning video saliency from human gaze using candidate selection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1147–1154, June 2013. 4

[155] R Rukšėnas, Jonathan Back, Paul Curzon, and Ann Blandford. Formal modelling of salience and cognitive load. *Electronic Notes in Theoretical Computer Science*, 208:57–75, 2008. 20

[156] Philip Saponaro, Scott Sorensen, Stephen Rhein, Andrew R Mahoney, and Chandra Kambhamettu. Reconstruction of textureless regions using structure from motion and image-based interpolation. In *Image Processing (ICIP), 2014 IEEE International Conference on*, pages 1847–1851. IEEE, 2014. 8

[157] Jill Sardegna, Susan Shelly, and Scott Steidl. *The encyclopedia of blindness and vision impairment*. Infobase Publishing, 2002. 24

[158] Gartner Says. 8.4 billion connected "things" will be in use in 2017, up 31 percent from 2016, gartner, february 7, 2017. 1

[159] B. Schölkopf, J. Platt, and T. Hofmann. *Graph-Based Visual Saliency*, pages 545–552. MIT Press, 2007. 4

[160] Yantao Shen, Tong Xiao, Hongsheng Li, Shuai Yi, and Xiaogang Wang. Learning deep neural networks for vehicle re-id with visual-spatio-temporal path proposals. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1900–1909, 2017. 103, 106

[161] Roger N Shepard and Jacqueline Metzler. Mental rotation of three-dimensional objects. *Science*, 171(3972):701–703, 1971. 104

[162] Philip Shilane, Patrick Min, Michael Kazhdan, and Thomas Funkhouser. The princeton shape benchmark. In *Proceedings Shape Modeling Applications, 2004.*, pages 167–178. IEEE, 2004. 115

[163] Wataru Shimoda and Keiji Yanai. Learning food image similarity for food image retrieval. In *IEEE Third International Conference on Multimedia Big Data (BigMM)*, pages 165–168. IEEE, 2017. 7

[164] Aliaksandr Siarohin, Enver Sangineto, Stéphane Lathuilière, and Nicu Sebe. Deformable gans for pose-based human image generation. In *CVPR 2018-Computer Vision and Pattern Recognition*, 2018. 11

[165] Edgar Simo-Serra and Hiroshi Ishikawa. Fashion style in 128 floats: joint ranking and classification using weak data for feature extraction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 298–307. IEEE, 2016. 7

[166] L. Simon, J. P. Tarel, and R. Bremond. Alerting the drivers about road signs with poor visual saliency. In *Intelligent Vehicles Symposium, 2009 IEEE*, pages 48–53, 2009. 5, 6, 18, 26

[167] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 7, 77, 113

[168] Ayan Sinha, Asim Unmesh, Qixing Huang, and Karthik Ramani. Surfnet: Generating 3d shape surfaces using deep residual networks. In *IEEE CVPR*, volume 1, 2017. 8, 9, 10

[169] Nasim Souly and Mubarak Shah. Visual saliency detection using group lasso regularization in videos of natural scenes. *International Journal of Computer Vision*, 117(1):93–110, 2015. 16

[170] Jonathan Starck and Adrian Hilton. Model-based human shape reconstruction from multiple views. *Computer Vision and Image Understanding*, 111(2):179–194, 2008. 8

[171] Michael Stark, Michael Goesele, and Bernt Schiele. Back to the future: Learning shape models from 3d cad data. In *Bmvc*, volume 2, page 5. Citeseer, 2010. 8

[172] Hao Su, Charles R Qi, Yangyan Li, and Leonidas J Guibas. Render for cnn: Viewpoint estimation in images using cnns trained with rendered 3d model views. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2686–2694, 2015. 8, 9, 98

[173] Shao-Hua Sun, Minyoung Huh, Yuan-Hong Liao, Ning Zhang, and Joseph J Lim. Multi-view to novel view: Synthesizing novel views with self-learned confidence. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 155–171, 2018. 11, 12, 116, 117, 118, 122, 124

[174] Xingyuan Sun, Jiajun Wu, Xiuming Zhang, Zhoutong Zhang, Chengkai Zhang, Tianfan Xue, Joshua B Tenenbaum, and William T Freeman. Pix3d: Dataset and methods for single-image 3d shape modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2974–2983, 2018. 115

[175] Kapje Sung, Joongryoul Lee, Junsik An, and Eugene Chang. Development of image synthesis algorithm with multi-camera. In *Vehicular Technology Conference (VTC Spring), 2012 IEEE 75th*, pages 1–5. IEEE, 2012. 10

[176] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 7

[177] Maxim Tatarchenko, Alexey Dosovitskiy, and Thomas Brox. Multi-view 3d models from single images with a convolutional network. In *European Conference on Computer Vision*, pages 322–337. Springer, 2016. 8, 11, 12, 93, 97, 105, 116, 117, 118, 124

[178] Benjamin W Tatler. The central fixation bias in scene viewing: Selecting an optimal viewing position independently of motor biases and image feature distributions. *Journal of vision*, 7(14):4–4, 2007. 21

[179] Benjamin W. Tatler, Mary M. Hayhoe, Michael F. Land, and Dana H. Ballard. Eye guidance in natural vision: Reinterpreting salience. *Journal of Vision*, 11(5), 2011. 6, 14, 43

[180] A. Tawari and M. M. Trivedi. Robust and continuous estimation of driver gaze zone by dynamic analysis of multiple face videos. In *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*, pages 344–349, 2014. 5

[181] Marvin Teichmann, Michael Weber, Marius Zoellner, Roberto Cipolla, and Raquel Urtasun. Multinet: Real-time joint semantic reasoning for autonomous driving. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1013–1020. IEEE, 2018. 103, 106

[182] Jan Theeuwes. Top–down and bottom–up control of visual selection. *Acta psychologica*, 135(2):77–99, 2010. 3

[183] Antonio Torralba, Aude Oliva, Monica S Castelhano, and John M Henderson. Contextual guidance of eye movements and attention in real-world scenes: the role of global features in object search. *Psychological review*, 113(4):766, 2006. 4

[184] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1653–1660, 2014. 9

[185] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. *arXiv:1412.0767*, 2014. 35, 36, 37, 38, 44, 45, 47, 48

[186] Anne M Treisman and Garry Gelade. A feature-integration theory of attention. *Cognitive psychology*, 12(1):97–136, 1980. 43

[187] Din Chang Tseng, Tat Wa Chao, and Jiun Wei Chang. Image-based parking guiding using ackermann steering geometry. In *Applied Mechanics and Materials*, volume 437, pages 823–826. Trans Tech Publ, 2013. 10

[188] Shubham Tulsiani and Jitendra Malik. Viewpoints and keypoints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1510–1519, 2015. 9, 98, 109, 117

[189] Shubham Tulsiani, Tinghui Zhou, Alexei A Efros, and Jitendra Malik. Multi-view supervision for single-view reconstruction via differentiable ray consistency. In *CVPR*, volume 1, page 3, 2017. 8, 91

[190] Yoshiyuki Ueda, Yusuke Kamakura, and Jun Saiki. Eye movements converge on vanishing points during visual search. *Japanese Psychological Research*, 59(2):109–121, 2017. 23

[191] Geoffrey Underwood, Katherine Humphrey, and Editha van Loon. Decisions about objects in real-world scenes are influenced by visual saliency before and during their inspection. *Vision Research*, 51(18):2031 – 2038, 2011. 5, 6, 15, 18, 42

[192] Joost Van De Weijer, Cordelia Schmid, Jakob Verbeek, and Diane Larlus. Learning color names for real-world applications. *IEEE Transactions on Image Processing*, 18(7):1512–1523, 2009. 7

[193] Marco Venturelli, Guido Borghi, Roberto Vezzani, and Rita Cucchiara. Deep head pose estimation from depth data for in-car automotive applications. In *Proceedings of the 2nd International Workshop on Understanding Human Activities through 3D Sensors*, 2016. 80

[194] Francisco Vicente, Zehua Huang, Xuehan Xiong, Fernando De la Torre, Wende Zhang, and Dan Levi. Driver gaze tracking and eyes off the road detection system. *IEEE Transactions on Intelligent Transportation Systems*, 16(4):2014–2027, 2015. 5

[195] George Vogiatzis, Carlos Hernández Esteban, Philip HS Torr, and Roberto Cipolla. Multiview stereo via volumetric graph-cuts and occlusion robust photo-consistency. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(12):2241–2246, 2007. 8

[196] Panqu Wang, Pengfei Chen, Ye Yuan, Ding Liu, Zehua Huang, Xiaodi Hou, and Garrison Cottrell. Understanding convolution for semantic segmentation. *arXiv preprint arXiv:1702.08502*, 2017. 43

[197] Panqu Wang and Garrison W Cottrell. Central and peripheral vision for scene recognition: A neurocomputational modeling exploration-wang & cottrell. *Journal of Vision*, 17(4):9–9, 2017. 63

[198] Peng Wang, Xinyu Huang, Xinjing Cheng, Dingfu Zhou, Qichuan Geng, and Ruigang Yang. The apolloscape open dataset for autonomous driving and its application. *IEEE transactions on pattern analysis and machine intelligence*, 2019. 109, 132

[199] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8798–8807, 2018. 11

[200] Wenguan Wang, Jianbing Shen, and Fatih Porikli. Saliency-aware geodesic video object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3395–3402, June 2015. 4, 16, 39, 54, 62

[201] Wenguan Wang, Jianbing Shen, and Ling Shao. Consistent video saliency using local gradient flow optimization and global refinement. *Trans. on Image Processing*, 24(11):4185–4196, 2015. 4, 39, 54, 62

[202] Xiaolong Wang and Abhinav Gupta. Unsupervised learning of visual representations using videos. *arXiv preprint arXiv:1505.00687*, 2015. 74

[203] Zhongdao Wang, Luming Tang, Xihui Liu, Zhuliang Yao, Shuai Yi, Jing Shao, Junjie Yan, Shengjin Wang, Hongsheng Li, and Xiaogang Wang. Orientation invariant feature embedding and spatial temporal regularization for vehicle re-identification. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 109, 132, 149

[204] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4724–4732, 2016. 9

[205] Olivia Wiles and Andrew Zisserman. Silnet: Single-and multi-view reconstruction by learning from silhouettes. *British Machine Vision Conference*, 2017. 8, 93

[206] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. In *Reinforcement Learning*, pages 5–32. Springer, 1992. 8

[207] Jeremy M Wolfe. Visual search. *Attention*, 1:13–73, 1998. 43

[208] Jeremy M. Wolfe, Kyle R. Cave, and Susan L. Franzel. Guided search: an alternative to the feature integration model for visual search. *Journal of Experimental Psychology: Human Perception & Performance*, 1989. 4

[209] Jiajun Wu, Yifan Wang, Tianfan Xue, Xingyuan Sun, Bill Freeman, and Josh Tenenbaum. Marrnet: 3d shape reconstruction via 2.5 d sketches. In *Advances in neural information processing systems*, pages 540–550, 2017. 8, 112

[210] Jiajun Wu, Tianfan Xue, Joseph J Lim, Yuandong Tian, Joshua B Tenenbaum, Antonio Torralba, and William T Freeman. 3d interpreter networks for viewer-centered wireframe modeling. *International Journal of Computer Vision*, 126(9):1009–1026, 2018. 109, 132

[211] Jiajun Wu, Chengkai Zhang, Tianfan Xue, Bill Freeman, and Josh Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *Advances in Neural Information Processing Systems*, pages 82–90, 2016. 8

[212] Bernhard Wymann, Eric Espié, Christophe Guionneau, Christos Dimitrakakis, Rémi Coulom, and Andrew Sumner. Torcs, the open racing car simulator. *Software available at http://torcs. sourceforge. net*, 2000. 11

[213] Wenqi Xian, Patsorn Sangkloy, Varun Agrawal, Amit Raj, Jingwan Lu, Chen Fang, Fisher Yu, and James Hays. Texturegan: Controlling deep image synthesis with texture patches. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8456–8465, 2018. 117

[214] Yu Xiang, Wonhui Kim, Wei Chen, Jingwei Ji, Christopher Choy, Hao Su, Roozbeh Mottaghi, Leonidas Guibas, and Silvio Savarese. Objectnet3d: A large scale database for 3d object recognition. In

European Conference on Computer Vision, pages 160–176. Springer, 2016. 12, 109, 132

[215] Yu Xiang, Roozbeh Mottaghi, and Silvio Savarese. Beyond pascal: A benchmark for 3d object detection in the wild. In *Applications of Computer Vision (WACV), 2014 IEEE Winter Conference on*, pages 75–82. IEEE, 2014. x, 12, 109, 110, 112, 115, 117, 126, 132, 134, 135

[216] SHI Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Advances in neural information processing systems*, pages 802–810, 2015. 139

[217] Xinchen Yan, Jimei Yang, Ersin Yumer, Yijie Guo, and Honglak Lee. Perspective transformer nets: Learning single-view 3d object reconstruction without 3d supervision. In *Advances in Neural Information Processing Systems*, pages 1696–1704, 2016. 8, 91, 93, 94, 99, 100

[218] Jimei Yang, Scott E Reed, Ming-Hsuan Yang, and Honglak Lee. Weakly-supervised disentangling with recurrent transformations for 3d view synthesis. In *Advances in Neural Information Processing Systems*, pages 1099–1107, 2015. 8, 11, 12, 105

[219] Linjie Yang, Ping Luo, Chen Change Loy, and Xiaoou Tang. A large-scale car dataset for fine-grained categorization and verification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3973–3981, 2015. 7

[220] Maoke Yang, Kun Yu, Chi Zhang, Zhiwei Li, and Kuiyuan Yang. Denseaspp for semantic segmentation in street scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3684–3692, 2018. 103, 106

[221] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015. 24, 43, 51, 52

[222] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 11

[223] Jeffrey M Zacks, Jon Mires, Barbara Tversky, and Eliot Hazeltine. Mental spatial transformations of objects and perspective. *Spatial Cognition and Computation*, 2(4):315–332, 2000. 104

[224] Yun Zhai and Mubarak Shah. Visual attention detection in video sequences using spatiotemporal cues. In *Proceedings of the 14th ACM international conference on Multimedia*, pages 815–824. ACM, 2006. 4

[225] Buyue Zhang, Vikram Appia, Ibrahim Pekkucuksen, Yucheng Liu, Aziz Umit Batur, Pavan Shastry, Stanley Liu, Shiju Sivasankaran, and Kedar Chitnis. A surround view camera solution for embedded systems. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 662–667, 2014. 10

[226] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5907–5915, 2017. 11

[227] J. Zhang and S. Sclaroff. Exploiting surroundedness for saliency detection: A boolean map approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP(99), 2015. 25, 26, 31

[228] Yiheng Zhang, Dong Liu, and Zheng-Jun Zha. Improving triplet-wise training of convolutional neural network for vehicle re-identification. In *IEEE International Conference on Multimedia and Expo (ICME)*, pages 1386–1391. IEEE, 2017. 7

[229] Bo Zhao, Xiao Wu, Zhi-Qi Cheng, Hao Liu, Zequn Jie, and Jiashi Feng. Multi-view image generation from a single-view. In *2018 ACM Multimedia Conference on Multimedia Conference*, pages 383–391. ACM, 2018. 11

[230] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification: A benchmark. In *Computer Vision, IEEE International Conference on*, 2015. 11

[231] Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip

Torr. Conditional random fields as recurrent neural networks. In *International Conference on Computer Vision (ICCV)*, 2015. 14

[232] Sheng-hua Zhong, Yan Liu, Feifei Ren, Jinghuan Zhang, and Tongwei Ren. Video saliency detection via dynamic consistent spatio-temporal attention modelling. In *AAAI*, 2013. 4

[233] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3D: A modern library for 3D data processing. *arXiv:1801.09847*, 2018. 11, 118, 136

[234] Tinghui Zhou, Shubham Tulsiani, Weilun Sun, Jitendra Malik, and Alexei A Efros. View synthesis by appearance flow. In *European conference on computer vision*, pages 286–301. Springer, 2016. 11, 12, 105

[235] Xiaowei Zhou, Menglong Zhu, Spyridon Leonardos, Konstantinos G Derpanis, and Kostas Daniilidis. Sparseness meets deepness: 3d human pose estimation from monocular video. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4966–4975, 2016. 9

[236] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *IEEE International Conference on Computer Vision (ICCV)*, 2017. 11, 113

[237] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A Efros, Oliver Wang, and Eli Shechtman. Toward multimodal image-to-image translation. In *Advances in Neural Information Processing Systems*, pages 465–476, 2017. 11

[238] Jun-Yan Zhu, Zhoutong Zhang, Chengkai Zhang, Jiajun Wu, Antonio Torralba, Josh Tenenbaum, and Bill Freeman. Visual object networks: image generation with disentangled 3d representations. In *Advances in Neural Information Processing Systems*, pages 118–129, 2018. x, 8, 9, 11, 12, 105, 113, 115, 116, 117, 118, 120, 121, 122, 124, 126, 127

[239] Menglong Zhu, Xiaowei Zhou, and Kostas Daniilidis. Single image pop-up from discriminatively learned parts. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 927–935, 2015. 9