



HAL
open science

Vision for Scene Understanding

Raoul de Charette

► **To cite this version:**

Raoul de Charette. Vision for Scene Understanding. Computer Science [cs]. Sorbonne Université, 2022. tel-03969456

HAL Id: tel-03969456

<https://inria.hal.science/tel-03969456>

Submitted on 2 Feb 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Mémoire pour l'obtention de
l'Habilitation à Diriger des Recherches

Sorbonne Université

Spécialité

Informatique

Présentée par

Raoul de CHARETTE

Chercheur, Inria

Vision for Scene Understanding

Jury composé de :

M.	David FILLIAT	Professeur, ENSTA / Inria	Rapporteur
M.	Vincent LEPETIT	Professeur, ENPC ParisTech / TU Graz	Rapporteur
M.	Nicolas THOME	Professeur, CNAM	Rapporteur
M.	Matthieu CORD	Professeur, Sorbonne / Valeo.ai	Examineur
Mme.	Gabriela CSURKA	Docteur, Naver Labs	Examinatrice
M.	Fawzi NASHASHIBI	Directeur de recherche, Inria	Examineur
M.	Josef SIVIC	Professeur, Inria / CTU	Examineur

Contents

I	Research	1
1	Introduction	3
2	Vision in the 3D world	5
2.1	3D object-centric vision	6
2.1.1	Deformable objects	6
2.1.2	Tracking rigid objects.	10
2.2	3D scene understanding	14
2.2.1	Geometry completion	14
2.2.2	Semantic scene completion	19
3	Weakly supervised vision	33
3.1	Dealing with fewer labels	34
3.1.1	Generative networks	34
3.1.2	Cross-modal learning	41
3.2	Dealing with fewer data	49
3.3	Supervision from action	54
3.3.1	DRL with dense reward	56
3.3.2	DRL with sparse reward	59
4	Vision and physics	63
4.1	Physics-informed vision	64
4.1.1	Reactive scene illumination	65
4.1.2	Physics-based rendering	66
4.2	Physics-guided learning	72
4.2.1	Model-guided disentanglement	72
4.2.2	Model-guided learning	78
5	Research perspectives	85
II	Scientific career	87
6	Professional	89
7	Supervision and Teaching activities	91
7.1	Supervision	91
7.2	Teaching activities	92

8	Dissemination	93
8.1	Dissemination	93
8.1.1	Popularization	93
8.1.2	Awards	94
8.2	Grants and Research projects	94
8.3	Publications	95
8.3.1	Journal with peered reviews	95
8.3.2	Conferences with peered reviews	95
8.3.3	Scientific communications	97
	Bibliography	99

Part I
Research

Introduction

Vision is crucial for scene understanding and a prerequisite for algorithms interacting with our visual human world. Whether it is traditional cameras capturing intensity and texture, depth sensors capturing geometry, or videos capturing changes in the scene, our ability to understand and process these data will enable better interaction between computer and humans.

Since my PhD thesis (de Charette, 2012) on image processing for driving assistant systems, my research interest has broadened towards scene understanding, expanding to new fields, new sensors, and new applications. As of today, my work lies at the cross-roads of computer vision, robotics, and artificial intelligence.

At the heart of my research is the study of computer vision algorithms. Though I embraced the data-driven paradigm, most of my works differentiate by focusing on weakly-supervised learning and physics-embodied vision. The latter is key to reading my work as I believe vision algorithms can only benefit from stronger physical grounding. Robots like humans naturally evolve in the 3D physical world and physics can not only provide insight of the possible world interactions, but also carries important knowledge on materials, geometry, lighting or weather conditions. Ultimately, physics-informed learning could provide AI vision algorithms with additional *interpretability* drifting away from the black box AIs (Miller, 2019).

Apart from a few exceptions, the manuscript covers mainly the period 2017–2021 and describes the research I had with students and collaborators following three axes of study: ‘Vision in the 3D world’, ‘Weakly supervised vision’, and ‘Vision and physics’.

The first axis includes research on 3D scene understanding, from 3D objects modeling and tracking to 3D scene completion and reconstruction leveraging either supervised deep networks or what is now referred as *traditional* computer vision. For the most parts, these works are with a PhD, as well as collaborators from Inria, Mines ParisTech, and Uni. of Makedonia.

The second axis investigates vision with few labels or data, otherwise referred as weakly-supervised vision. It describes three lines of works:

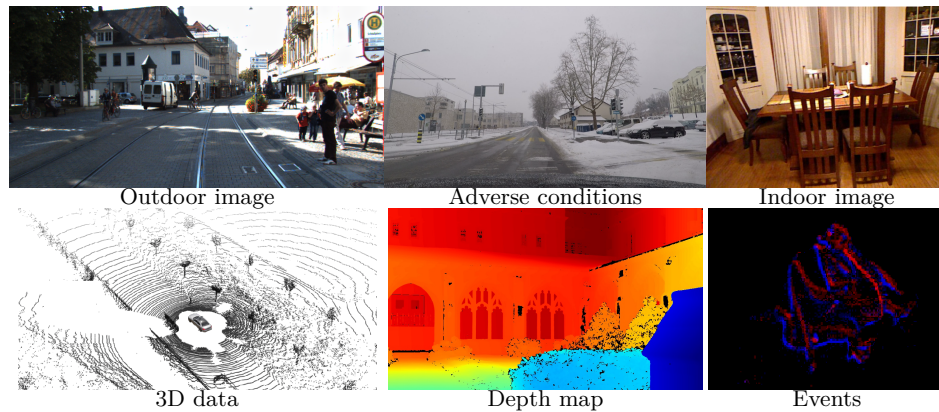


Figure 1.1: **Representative scenes and data.** Some of the type of scenes, conditions and nature of data addressed in this manuscript. Sources: (Geiger et al., 2013; Sakaridis et al., 2021; Silberman et al., 2012a; Caesar et al., 2020a; Vasiljevic et al., 2019; Dubeau et al., 2020)

domain adaptation where a trained model is transferred to a new domain having zero or few labels, few-shot where the target domain has only very few training samples, and reinforcement learning which dense or sparse supervision is obtained from a reward function. These works are with 2 PhDs, a PostDoc and other Inria collaborators.

Finally, the third axis encompasses physics-informed vision where physical models are leveraged to improve performance in adverse lighting and weather conditions. The two main paradigms studied are: physics-based rendering where synthetically augmented images are produced at virtually no cost, and physics-guided learning where generative networks are guided by simple physical models. These works were mainly conducted with a PhD, and collaborators from Inria and Uni. Laval.

Real-world applications. Sample scenes and data addressed in this manuscript are shown in Fig. 1.1. Application of my works cover mainly the field of autonomous driving, but also arts & virtual reality, and computer graphics & photo editing.

While this document focuses on the scientific achievements, a significant part of my work included engineering for real life experiments – time consuming though barely brushed in the manuscript – such as sensors calibration, autonomous driving demos, datasets recording, etc.

Vision in the 3D world

Contents

2.1 3D object-centric vision	6
2.1.1 Deformable objects	6
2.1.2 Tracking rigid objects.	10
2.2 3D scene understanding	14
2.2.1 Geometry completion	14
2.2.2 Semantic scene completion	19

We focus here on extracting a thorough 3D understanding, ranging from 3D object-centric vision to full 3D geometrical and semantical scene understanding. While humans have great ability to extract 3D geometry from images relying on strong priors (Koenderink et al., 1995), this is a notoriously ill-condition problem (Sinha and Adelson, 1993), so in addition to RGB cameras we investigate data like Depth/Events cameras, and Lidars – some of which hold specific challenges due to the sparse or asynchronous nature of their data.

In the first part (Sec. 2.1), we elaborate on the tracking and reconstruction of objects from either geometrical data only or fusion with events and color information – for the most part in the context of Arts & Virtual Reality. Among others, we focus on objects under interaction as it holds significant challenge due to occlusion.

In the second part (Sec. 2.2), we investigate 3D scene understanding, leveraging first physical priors to improve 3D grid representations of the world, surface reconstruction and finally addressing the challenging topic of predicting a dense semantic labeled representation, otherwise referred as semantic scene completion. These works mainly originate from the PhD of Luis Roldão-Jimenez co-supervised with Anne Verroust-Blondet, and to a lesser extend from PhD Maximilian Jaritz and PhD student Anh Quan Cao. This part is centered on autonomous driving applications.

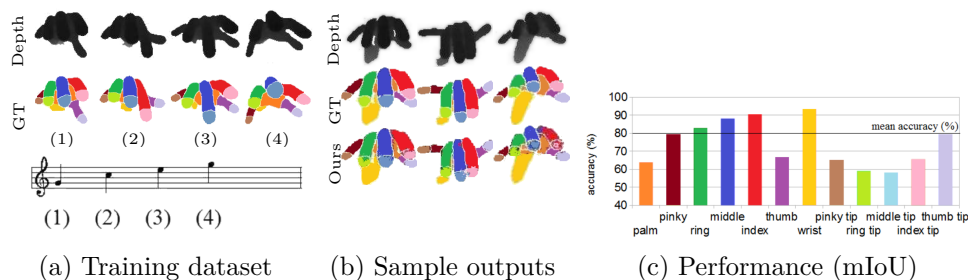


Figure 2.1: **Hand skeletal segmentation for pianist.** When trained on custom labeled arpeggio depth images (a), our RDF is able to segment real test images reasonably good (b) with an mIoU of 80% on our test set (c).

2.1 3D object-centric vision

2.1.1 Deformable objects

In 2013–2014, during my post-docs at Mines ParisTech and University of Makedonia (Greece), we addressed the tracking of deformable objects, being either body parts or pottery objects during their making process. Both works were part of a large European project, i-Treasures, which noble purpose is to capture and preserve intangible cultural heritage and rare know-how such as vocal abilities (eg. throat singing), dance/musical gestures (eg. pianist gesture), and rare handicraft (eg. pottery).

To alleviate the complexity of tracking deformable objects which appearance varies greatly, we use depth sensors (here, PMD camboard camera) providing informative geometrical cues and reducing domain gaps for humans – since skin color is not accountable.

Going further...

Dapogny, A., de Charette, R., Manitsaris, S., Moutarde, F., and Glushkova, A. (2013). Towards a hand skeletal model for depth images applied to capture music-like finger gestures. In *CMMR*

Hand skeletal segmentation. In the early [Dapogny et al. \(2013\)](#) we leveraged Random Decision Forests (RDF) to detect the skeleton of a pianist’s hand from depth images (Fig. 2.1a). RDF are improved decision tree where a complex problem is split in simple decisions (tree nodes) with leaves being the final decision. Here, we used the algorithm of [Shotton et al. \(2011\)](#); [Keskin et al. \(2011\)](#) – among the bests segmentation algorithms at the time – training our RDF to maximize the information gain of weak classifier on thresholded depth difference for pairs of pixels.

Our lowly contribution was to design two strategies for our 12 parts hand model, training either on few dozen thousands images from a scripted 3D hand model in the Autodesk Maya software, or on 500 real labeled depth image from 5 users (Fig. 2.1a). Comparatively, [Keskin et al. \(2011\)](#) used 200k synthetic images and 15k real ones. Performance reached on our test

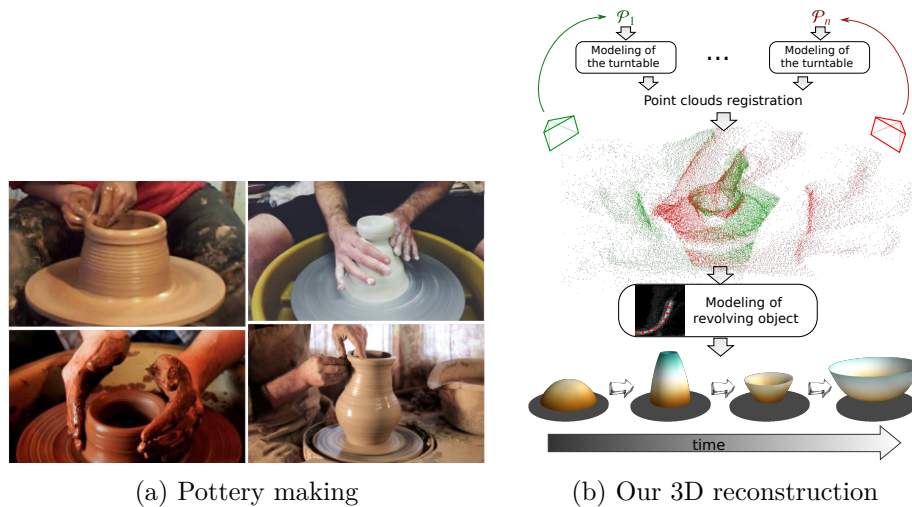


Figure 2.2: **Task and methodology.** (a) Reconstruction of unknown 3D objects in the context of wheel throwing pottery. (b) Using one or more input point clouds, our methods clusters the 3D scene and extracts the profile of revolving objects. Bottom are sample outputs of our method.

set (Fig. 2.1b) was sufficient for high-fidelity pianistic gesture detection with 0.8 mIoU error and <3 pixels error for finger tip locations. This allowed full 3D hand model retrieval with inverse kinematic as in Schröder et al. (2014) since fingers are constrained-articulated.

Reconstruction of revolving objects. In our 2014 work de Charette and Manitsaris (2019) (late published) we studied the 3D reconstruction of wheel throwing pottery object during its making process, that is as it evolves from a clay ball to its final shape. There are notable challenges. First, because hands and objects are hardly distinguishable due the wet clay covering both (see Fig. 2.2a). Second, since the pottery is suffers from heavy occlusion from the potters hands. Third, because the object shape evolves with virtually infinite Degrees of Freedom (DOF).

At the time of this work, while many works addressed rigid objects tracking (Lepetit and Fua, 2005; Smeulders et al., 2013), the standard approach to reconstruct *known* deformable object was the use of Shape from Template (SfT) (Schulman et al., 2013; Vicente and Agapito, 2013; Östlund et al., 2012; Salzmann and Fua, 2009), the common application being the tracking of planar surfaces (paper, t-shirt, etc.). Only a handful of researches explicitly reconstructed free-form unknown object, that is of arbitrary shape but known topology.

Going further...

de Charette, R. and Manitsaris, S. (2019). 3D reconstruction of deformable revolving object under heavy hand interaction. *arXiv* submitted to CVIU

Our setup several uses depth sensors radially distributed all around the turntable, thus reducing the effect of hand occlusions while being non-invasive for the artist. Our pipeline, in Fig. 2.2b, relies on two observations: the pottery object always sits on a turntable and both share a common revolving axis. Each depth sensor i acquires a 3D point cloud \mathcal{P}_i from which we estimate the cylindrical turntable model $(\vec{c}_i, \vec{n}_i, r)$ with an mSAC optimizer (Torr and Zisserman, 2000; Lebeda et al., 2012) and a Kernel Density Estimator (KDE). The turntables act as shared landmarks to solve the registration matrix \mathbf{M}_i , and build \mathcal{P}_r from the rigid registration of all point clouds: $\mathcal{P}_r = \mathcal{P}_1 \cdot \mathbf{M}_1 \cup \dots \cup \mathcal{P}_n \cdot \mathbf{M}_n$.

The originality of our work lies in the extraction of the 3D object. Considering that the turntable and pottery share the same axis of revolution, we transform all \mathcal{P}_r points into a polar coordinates (ρ, h, θ) centered around this axis. Because of noise and occlusion a radial cross-section is not sufficient to build the object profile. Instead we radially integrate points into a so-called *radial accumulator* $\Gamma(\cdot)$ and inspire from circular statistics to compensate for occluders (like the potter’s hands).

We formulate the object profile extraction problem as a swarm-particle optimization using Catmull-Rom parametrization (Catmull and Rom, 1974). The latter is a form of cubic Hermite spline, which unlike B-Spline, permit a bounded search space. The pottery profile is modeled as C^k Catmull-Rom of $k = 5$ knots $\{\kappa_1, \dots, \kappa_k\}$. Coordinates from knots j to $j + 1$ are computed with $\chi(p)$ ($p \in [0, 1]$ the progression):

$$\chi(p) = \frac{1}{2} \begin{pmatrix} 1 & p & p^2 & p^3 \end{pmatrix} \begin{pmatrix} 0 & 2 & 0 & 0 \\ -\tau & 0 & \tau & 0 \\ 2\tau & \tau - 6 & -2(\tau - 3) & -\tau \\ -\tau & 4 - \tau & \tau - 4 & \tau \end{pmatrix} \begin{pmatrix} \kappa_{j-1} \\ \kappa_j \\ \kappa_{j+1} \\ \kappa_{j+2} \end{pmatrix}, \quad (2.1)$$

with $\tau \in [0, 1]$ the curve tension. We employ a bootstrap particle filter (Candy, 2007), with N particles, which evaluates $P(C_i^5 | \Gamma) \forall i \in [1, N]$ the probability of each particle (C_i^5) to match the radial accumulator Γ (our “observation”). To approximate $P(x | \Gamma)$ the radial accumulator Γ is approximated as a Gaussian Mixture Model (GMM) so the probability of observation of any particle, is obtained from the Probability Density Function (PDF) of the GMM. A motion model $f(\cdot)$ updates the particle state after each optimization, ie $X \leftarrow f(X)$.

Experiments. To evaluate our proposal we recorded 3 pottery makings from Claude Aiello – a famous potter from Vallauris (France) – using 2 depth sensor (160x120, @25FPS), and labeled all 6030 frames. Considering 16x16 accumulator and 1000 particles for optimization, Fig. 2.3 shows our method is able to reconstruct well the 3D objects, despite challenging hands

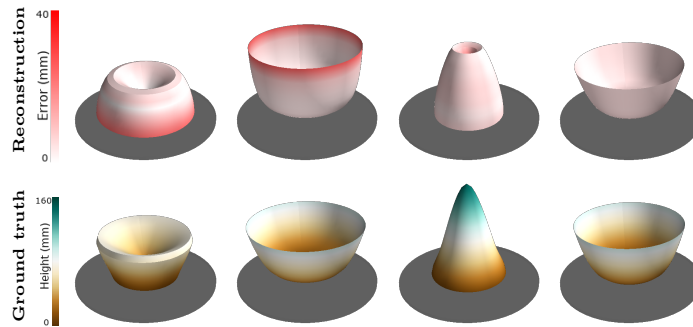


Figure 2.3: **3D pottery reconstructions.** Output of our method (top, color encodes error) and ground truths (bottom, color encodes height).

occlusions (see Fig. 2.2a for reference). Prevalence of errors at the tip of the object (see cols 2,3 Fig. 2.3) relate both to the sensor noise and the GMM smoothing edges. To quantify the error, we report the symmetrical versions of Average Error ($\bar{\delta}_{AE}$) and Hausdorff Distance ($\bar{\delta}_{HD}$) – computed from point-wise distance between the predicted and ground truth profiles.

In Tab. 2.1a we compare against the closest work in spirit (Wang et al., 2006) having B-Spline fitting, and report the 10 runs average. Our method appears twice better with a reconstruction error of $\bar{\delta}_{AE} = 8.09\text{mm}$ or $\bar{\delta}_{AE} = 7.60\text{mm}$ with particle resampling (col ‘Temp.’) to account for object motion. Noticeably, the error is smaller than the accumulator resolution (10mm) which we ascribe to the continuous GMM representation. Further ablations in our paper show that more particles or larger accumulator resolution boost performance at a significant processing cost. Ablating the depth sensors in Tab. 2.1b shows our method performs reasonably with a single sensor ($\bar{\delta}_{AE} < 11\text{mm}$), though more prone to occlusion.

Recently, capturing cultural heritage with computer vision gained momentum with special issues in IJCV and prestigious workshops in CVPR, ICCV, IROS, etc. Wu et al. (2021) elegantly address a similar problem, recovering pixel-wise material attributes (albedo, diffuse, etc.) and 3D shape from still image of occlusion-free revolutionary objects. In my opinion, modern 3D reconstruction little account for symmetry priors which are important cues. A research direction I wanted to investigate is the discovery of radial symmetries in the wild. Zhou et al. (2021) somehow address this for the simpler planar case, assuming symmetry translates in the features space. Radial integration was also shown to be simple and efficient way to boost data with low Signal to Noise Ratio (SNR). It was used in the fascinating work of Bouman et al. (2017) to turn corners into cameras.

	Temp.	$\bar{\delta}_{AE} \downarrow$	$\bar{\delta}_{HD} \downarrow$		Sensor	$\bar{\delta}_{AE} \downarrow$	$\bar{\delta}_{HD} \downarrow$
Wang et al. (2006) 5 knots	×	16.08 ±9.71	50.87 ±20.92		1	9.77 ±7.97	25.76 ±16.92
Wang et al. (2006) 8 knots	×	21.41 ±8.08	74.71 ±19.91		2	10.56 ±8.30	23.56 ±14.62
Ours	×	8.09 ±8.59	21.16 ±15.56		1 & 2	8.09 ±8.59	21.16 ±15.56
Ours	✓	7.60 ±8.64	19.84 ±18.70				

(a) Reconstruction on test set

(b) Sensors ablation

Table 2.1: **Performance on test set.** (a) Reconstruction errors in mm show we significantly outperform the (only and simple) baseline of Wang et al. (2006), both in average error ($\bar{\delta}_{AE}$) and maximum error ($\bar{\delta}_{HD}$). ‘Temp.’ is temporal filtering (0.8 particles resampling). (b) Shows that our method is robust to using a single sensor also.

2.1.2 Tracking rigid objects.

Assuming we *know* the 3D model of an object, a remaining challenge is to track its position and orientation. This is referred as 6-Degree Of Freedom (6-DOF) tracking and has applications for example in augmented reality.

Going further...

Code and dataset:
https://lvsn.github.io/rgbde_tracking
 See videos for demos.

Dubeau, E., Garon, M., Debaque, B., de Charette, R., and Lalonde, J.-F. (2020). RGB-D-E: Event camera calibration for fast 6-dof object tracking. In *ISMAR*

6-DOF tracking with RGB-D-E. In the context of our collaborative grant with Université Laval we addressed *fast* 6-DOF tracking in Dubeau et al. (2020). With relatively slow motion such tracking is reasonably solved using RGB (Manhardt et al., 2018; Li et al., 2018; Crivellaro et al., 2015) or RGB-D (Garon et al., 2018). However, high speed motion is challenging for off-the-shelf RGB and Depth cameras, as it produces motion blur and shadow artefacts (see Fig. 2.5a), respectively. Instead, in this work we proposed what we believe to be the first 6-DOF object tracker using event data. The specificity of events-based (aka neuromorphic) cameras is that all pixels operate independently and asynchronously, capturing local changes with very low latency ($20\mu s$), making them suitable for our use.

Our setup, visible in Fig. 2.4a, uses a DAVIS346 event camera rigidly mounted over the RGB-D Kinect Azure using a custom 3D-printed mount, and an infrared filter to prevent interference from the Kinect emitter.

An important challenge of this work was to calibrate data, both spatially and temporally. For the intrinsics (6 radials and 2 tangentials) we proceeded in the standard way and used PnP (Fischler and Bolles, 1981) to solve extrinsics. Temporal synchronization is a tricky endeavor one must be aware of for time-critical applications like fast tracking. To solve it we used Kinect emitted pulses and proposed a simple fix for temporal misalignment. Our calibration performed roughly twice better than the original presets.

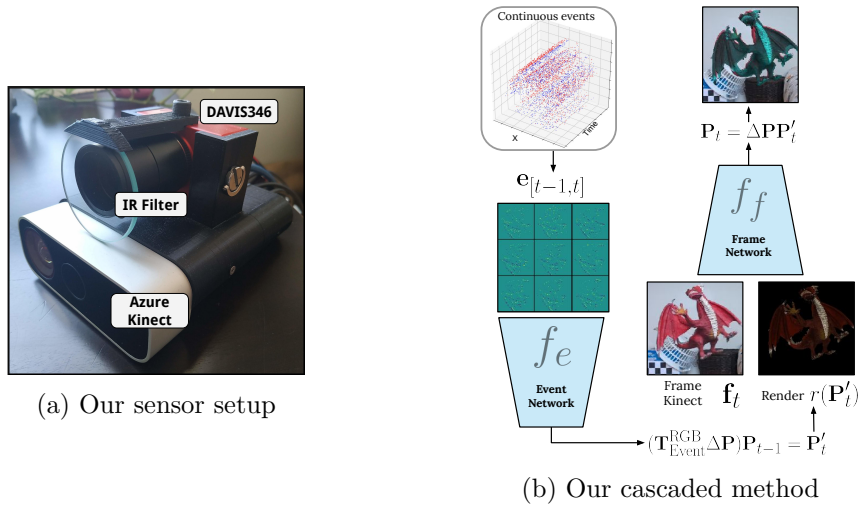


Figure 2.4: **RGB-D-E for fast 6-DOF tracking.** (a) Our setup uses a Kinect Azure (RGB-D) and a DAVIS346 event-based camera (E), spatially and temporally calibrated. (b) Overall pipeline for 6-DOF tracking using an event network and a frame network.

Using this setup, we recorded a test dataset with 2,472 RGB-D-E labeled frames (10 sequences) of a high-speed moving real-world 3D-printed dragon retrieved from Garon et al. (2018). As ground truths, we manually refined an ICP alignment (Pomerleau et al., 2013) on visible 3D model vertices. Our dataset, previewed in Fig. 2.5a, is publicly available.

Our research formulates 6-DOF object tracking as the estimation of the 6-DOF shift $\Delta \mathbf{P}$ between the last known position \mathbf{P}_{t-1} and the current RGB-D-E observation, such that $\mathbf{P}_t = \Delta \mathbf{P} \mathbf{P}_{t-1}$. Therefore, at $t = 0$, we initialize the tracking with known position but could as well use any 3D detector. To estimate $\Delta \mathbf{P}$ we rely on two networks combined in a cascaded fashion, illustrated in Fig. 2.4b. First, a novel event network $f_e(\cdot)$ is fed with events $\mathbf{e}_{[t-1,t]}$ accumulated in $[t-1, t]$ time interval and cropped around the last known position. Second, we use an existing RGB-D frame network $f_f(\cdot)$ from Garon et al. (2018), fed with the current cropped RGB-D frame \mathbf{f}_t , and the rendered image $r(\cdot)$ of the 3D object at a given position.

Interestingly, while events are much more robust to fast displacement they carry less textural information than RGB-D data so they are best combined. Hence, our cascaded approach uses the event network first 6-DOF estimate \mathbf{P}'_t , subsequently fed to the frame network for refinement:

$$\mathbf{P}'_t = (\mathbf{T}_{\text{Event}}^{\text{RGB}} f_e(\mathbf{e}_{[t-1,t]})) \mathbf{P}_{t-1}, \quad (2.2)$$

$$\mathbf{P}_t = f_f(\mathbf{f}_t, r(\mathbf{P}'_t)) \mathbf{P}'_t, \quad (2.3)$$

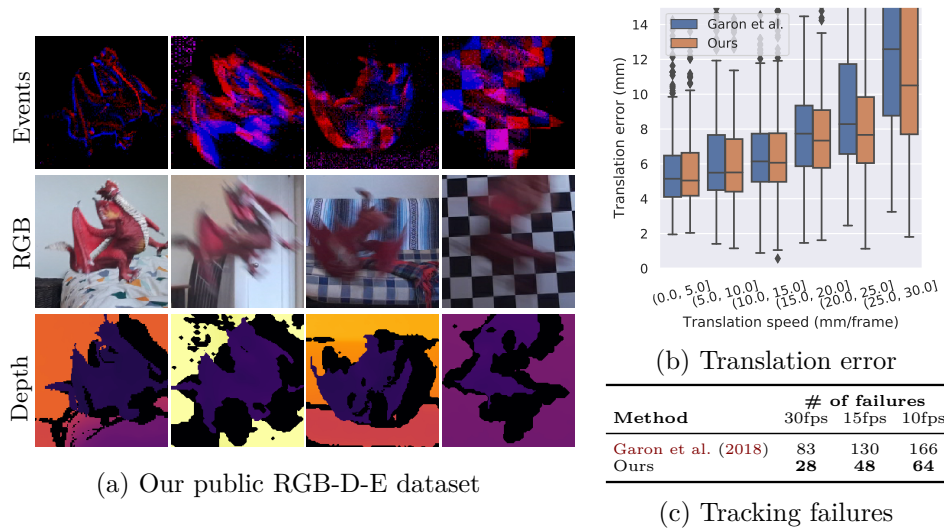


Figure 2.5: **Dataset and performance.** (a) Our public test dataset having 2,472 RGB-D-E labeled frames exhibits fast object displacement leading to motion blur and shadow effect in RGB-D sensor (bottom rows). Events (top) are encoded as blue (positive) and red (negative). (b) Translation error w.r.t. ground truth translation speed. (c) Quantifies tracking failures.

with $\mathbf{T}_{\text{Event}}^{\text{RGB}}$ the Event to RGB extrinsics. In practice $f_f(\cdot)$ is called 3 times for refinement, this helps recovering position at high motion speed.

It has to be noted that event data is fundamentally different than frame-based data as it possesses 2 *extra* dimensions T and P for time and polarity, respectively. We thus use the “Event Spike Tensor” representation from Gehrig et al. (2019) where time dimension is binned (here, 9 bins per 33 ms sample) and polarity is removed. While such sparse tensor can be processed by standard CNN as we showed in Jaritz et al. (2018b), reasoning on the nature of our data we follow Gehrig et al. (2019) and first learn a 1D filter along time dimension. This intuitively helps the network in finding dominant motion patterns. Network details are in Dubeau et al. (2020).

Experiments. We train our network solely on synthetic data. In short, we leveraged an event simulator (Rebecq et al., 2018), and put important efforts in the generation of realistic synthetic events, RGB, and depth by simulating motion of our virtual 3D dragon model in varying scene backgrounds from SUN 3D dataset (Xiao et al., 2013). This led to 180,000/18,000 training/validation samples.

We evaluate our method on our RGB-D-E dataset (Fig. 2.5a), and compare against Garon et al. (2018) – the backbone of our frame-based net-

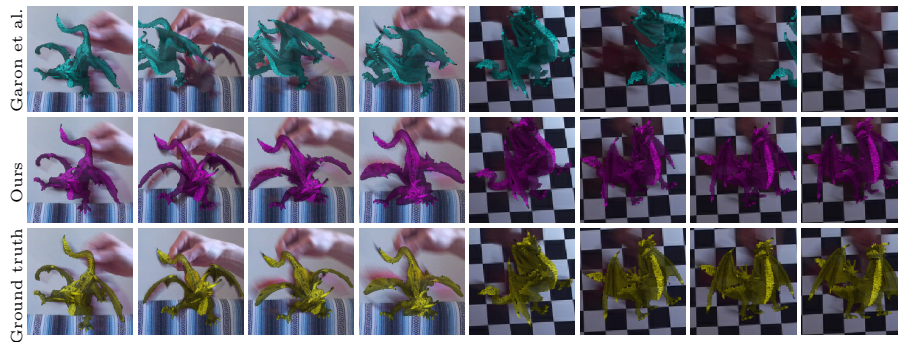


Figure 2.6: **Qualitative tracking on our RGB-D-E dataset.** The overlay is the 6-DOF position predicted by each method. Notice how ours (pink) is always closer to ground truth (yellow), compared to [Garon et al. \(2018\)](#) (green) despite fast motion and partial hand occlusion.

work. The translation error in Fig. 2.5b shows that our network outperforms the baseline, especially when displacement overpasses 20mm/frame (approx. 600mm/sec). Visually the 6-DOF estimation difference is evident in Fig. 2.6 showing sample trackings with (cols 1-4) or without (cols 5-8) hand interaction. For all, our 6-DOF prediction (middle row, overlaid in pink) is better than [Garon et al. \(2018\)](#) (top, green) w.r.t. ground truth (bottom, yellow). However, frame-based evaluation does not tell the entire story since small errors will eventually accumulate and diverge. Hence, Fig. 2.5c measures the tracking failures – that is how often the error of the predicted 6-DOF overpasses some threshold thus requiring reinitialization. In all frame-rate scenarios we outperformed the baseline with 2–3x less failures.

Tracking with other means. For autonomous driving, localizing yourself and other road users is of paramount importance. Fusing Lidar and communication data we addressed object tracking in [Flores et al. \(2018\)](#), relying on traditional geometrical and statistical priors. In [Nguyen et al. \(2018\)](#) we addressed ego tracking (localization) this time relying only on the WiFi fingerprints to locate ourselves. Both works were validated with extensive real-world driving experiments implying several road users. In [Meyer and de Charette \(2016\)](#) we showed ego-velocity can be tracked from RGB images leveraging trivial scene-physical priors to estimate scene flow, naively reaching less than 3km/h error on the KITTI dataset ([Geiger et al., 2013](#)).

Going further...

Flores, C., Merdrignac, P., de Charette, R., Navas, F., Milanés, V., and Nashashibi, F. (2018). A cooperative car-following/emergency braking system with prediction-based pedestrian avoidance capabilities. *IEEE T-ITS*

Nguyen, D.-V., de Charette, R., Nashashibi, F., Dao, T.-K., and Castelli, E. (2018). Wifi fingerprinting localization for intelligent vehicles in car park. In *IPIN*

Meyer, A. and de Charette, R. (2016). Computing ego velocity from scene flow estimation

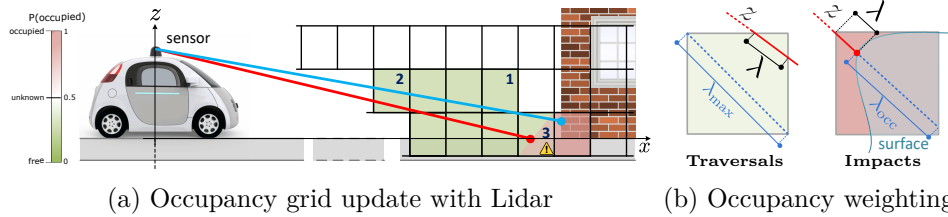


Figure 2.7: **Occupancy grids inaccuracies.** In Roldão et al. (2018) we use physical priors to solve inaccurate grid update (a) due to uneven density of observations (eg. cell 1 is only observed once), and conflicting observations (eg. cell 3). For the latter, our occupancy weighting (b) accounts for ray path information to balance individual updates.

2.2 3D scene understanding

We now investigate research on more general scene understanding, seeking an averagely better 3D scene understanding than object-specific methods.

First, we investigate the ability to reconstruct a geometrical representation of the world, either via aggregation of multiple Lidars scans (Roldão et al., 2018), reconstruction of 3D surfaces (Roldão et al., 2019), or depth completion in the image space (Jaritz et al., 2018b).

We then investigate dense semantic scene completion (SSC) from sparse 3D Lidar scans (Roldão et al., 2020), detail our recent survey on 3D SSC (Roldão et al., 2021), and conclude with our most recent 3D SSC work relying on a single RGB image (Cao and de Charette, 2021).

2.2.1 Geometry completion

Accurate and complete geometry is a crucial cue for mobile robots but sensors like Lidars only sparsely sense the scene. Inferring a dense (or denser) representation could improve the geometrical scene understanding.

We address geometrical completion for multiple Lidar scans (Roldão et al., 2018), local surface reconstruction for single scans (Roldão et al., 2019) important for rendering and physical simulation, and depth completion from sparse Lidar projection in the image space (Jaritz et al., 2018b).

Physical priors for occupancy grid. Assuming an occupancy grid, where each cell stores the probability of being physically occupied, in a noise-free world multiple observations of the same portion of the scene – ie. same cell – would lead to identical occupancy states. This however falls shorts because of sensor inaccuracies, scene changes, or partial observations.

In Roldão et al. (2018) we leverage simple physical priors to improve the accuracy of 3D occupancy grids. The starting point was the observation that the literature considers cells *occupied* whenever there is a Lidar-return

Going further...

Roldão, L., de Charette, R., and Verroust-Blondet, A. (2018). A statistical update of grid representations from range sensors. *arXiv*

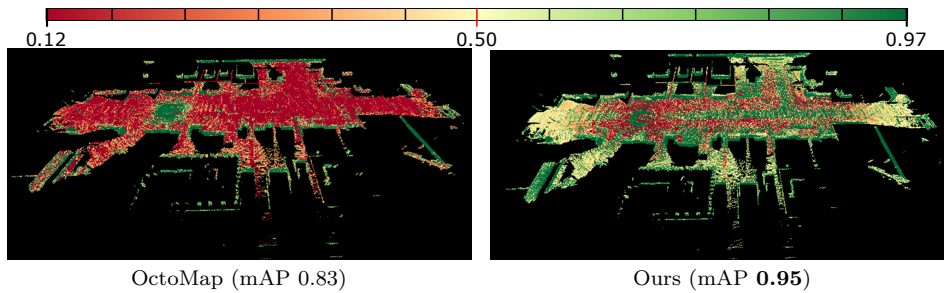


Figure 2.8: **Carla occupancy maps.** Occupancy probability for *truly occupied* voxels after 200 sequential Lidar scans. Our method outperforms OctoMap (Hornung et al., 2013) (the greener the better).

within though several rays (from same or consecutive scans) can produce different observations of the same cell. This is depicted in Fig. 2.7a where cell “3” is observed free by the blue ray and occupied by the red ray.

First, noting that rays are only partial observations, we accounted for the ray path information to weight the occupancy update depicted in 2D in Fig. 2.7b. For traversals, we account for the traversed distance λ w.r.t. maximum traversable distance λ_{\max} . For impacts, we account for the traversed distance λ w.r.t. to the occupied distance λ_{occ} . Second, noting that cells have uneven density of observations due to the heterogeneous Lidar sensing – the farther the sparser –, we weight the update of each cell \mathbf{c} as a function of its density of observations $\rho(\mathbf{c})$, analytically computed from derivation of the optical Lidar characteristics. The intuition here is that cells rarely observed should update their occupancy status faster than cells frequently observed.

Our simple proposals can improve existing inverse sensor models. In Fig. 2.8, comparing against our backbone OctoMap (Hornung et al., 2013), we show *truly occupied* voxels after 200 Lidar scans updates in Carla simulator (Dosovitskiy et al., 2017). The greener the better. The qualitatively better maps is confirmed by mean Average Precision (mAP) of the occupancy state, 12 points better with our simple proposals. In fact, on all tested hyper-parameters our method surpasses OctoMap by at least 6 points. On the real KITTI dataset (Geiger et al., 2013), we observe similar qualitative behavior though lack of ground truth prevent quantification.

Adaptive local planar reconstruction. In Roldão et al. (2019) we addressed 3D surface reconstruction from a *single* Lidar scan. The challenge of this task lies in occlusions, noise sensor, and the uneven density point cloud. In such scenario, reconstruction is a highly ill-posed problem since

Going further...

Roldão, L., de Charette, R., and Verroust-Blondet, A. (2019). 3D surface reconstruction from voxel-based lidar data. In *ITSC*

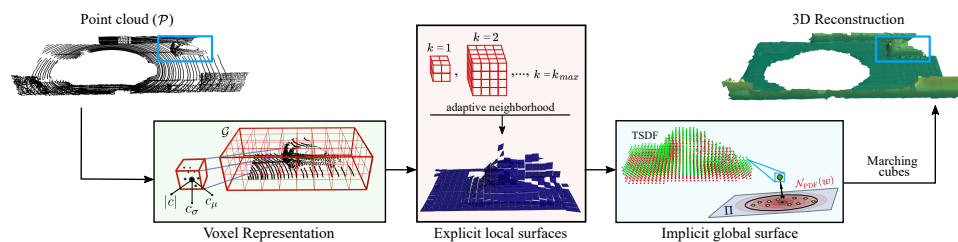


Figure 2.9: **Adaptive local planar reconstruction.** We exploit voxel-points statistics to approximate the scene as local planar surfaces using an adaptive neighborhood. The implicit global surface (TSDF) is then computed from the weighted set of probability density functions of 2D Gaussians.

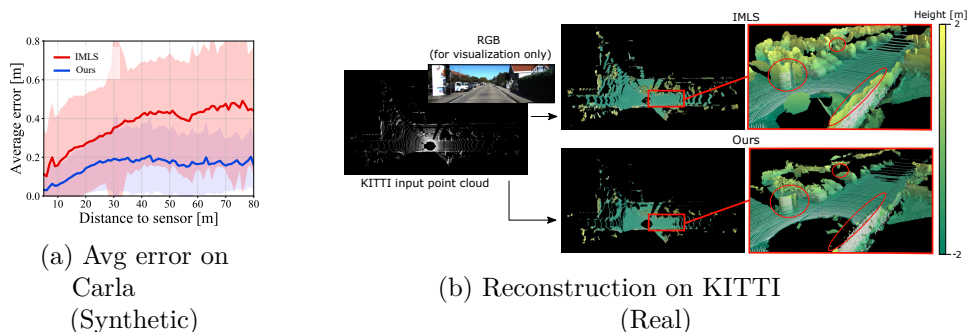


Figure 2.10: **Synthetic and real reconstruction.** (a) Error as a function of distance to sensor in Carla simulator (Dosovitskiy et al., 2017). (b) Sample results on real KITTI dataset (Geiger et al., 2013) show our method somehow preserves density while avoiding spurious extension of the reconstructed mesh as in IMLS.

there is not a unique surface for a given point cloud. At the time of this work, the literature mostly used traditional processing (Berger et al., 2017) though some deep reconstructions techniques were emerging for yet small point clouds. Our work follows the former.

The originality of our method (see Fig. 2.9) lies in the mix of explicit/implicit surface estimation from input voxel representation. The latter is built incrementally, storing voxels-points statistics (density, variance) to preserve scene details. To cope with low density point cloud, we approximate the scene as piece-wise planar though using an adaptive neighborhood strategy. For each voxel corner, we then estimate the distance to the closest surface (aka TSDF) from the set of optimal 2D Gaussians lying on the estimated planes. Marching cubes then extract the 3D surface from the TSDF.

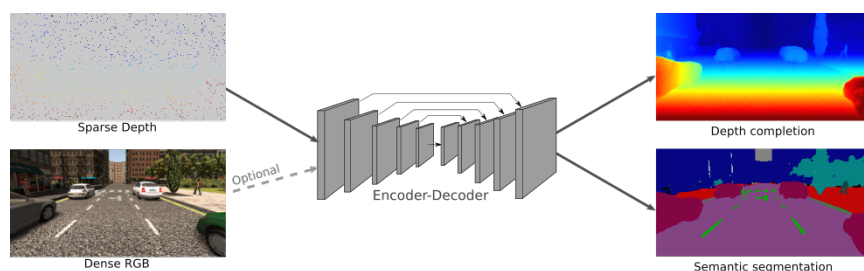


Figure 2.11: **Spaded**. Our work (Jaritz et al., 2018b) shows that a standard UNet can infer dense depth or semantics from sparse and dense inputs.

Bearing in mind we use only a single Lidar scan, the benefit of this method compared to the popular IMLS (Kolluri, 2005) or its extended version (Bouchiba et al., 2020) is that it can cope with varying points density. This is visible in Fig. 2.10a looking at the average reconstruction error as a function of distance, almost constant at far for our method. However, one must note that the comparison is not eye-to-eye since IMLS does not use any adaptive neighborhood strategy – eventually failing at far. In Fig. 2.10b, results on KITTI (Geiger et al., 2013) (64-layers Lidar) also show the higher accuracy since IMLS tends to extend surfaces over the edges.

Spaded: depth completion. Instead of completing the geometry in the 3D world, Lidar data can be projected in the image plane to get a 2.5D depth data which can be easily integrated along other modalities (eg. RGB). However, Lidar points do not align well with image pixels thus leading to sparse depth maps (see Fig. 2.11).

In Jaritz et al. (2018b) we study how sparse depth and dense RGB data can be processed with CNNs, for depth completion and semantic segmentation. At the time of this work an emerging line of researches was addressing sparse data with more or less complex mechanisms to obtain sparsity invariant CNNs (Uhrig et al., 2017; Huang et al., 2018b; Ren et al., 2018) that defined sparse convolutions only where the input domain is valid.

Instead, the simple – yet important – finding in our work is that sparse and dense data could be addressed with normal CNNs, simply by carefully adjusting the network design and training strategy. Our pipeline in Fig. 2.11 leverages a NASNet architecture (Zoph et al., 2018). To compensate for the input sparsity, we simply employ large receptive field, and train by varying the input density in $]0, 1]$ which naturally enforces invariancy to sparsity. When RGB is input along, we use a late fusion scheme.

While the popular Sparsity invariant CNNs (Uhrig et al., 2017) or

Going further...

Jaritz, M., de Charette, R., Wirbel, E., Perrotton, X., and Nashashibi, F. (2018b). Sparse and dense data with cnns: Depth completion and semantic segmentation. In *3DV*

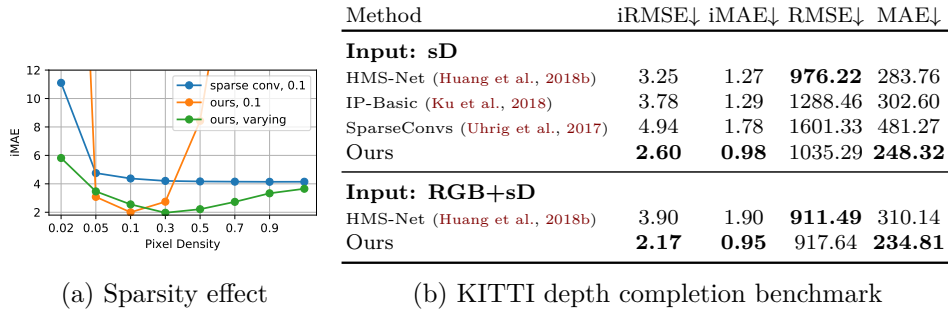


Figure 2.12: **Robustness and performance.** (a) On the depth completion task (here, Synthia), our simple UNet architecture trained with varying density outperforms the complex sparsity invariant CNNs (Uhrig et al., 2017). (b) Depth completion performance on the Kitti benchmark.

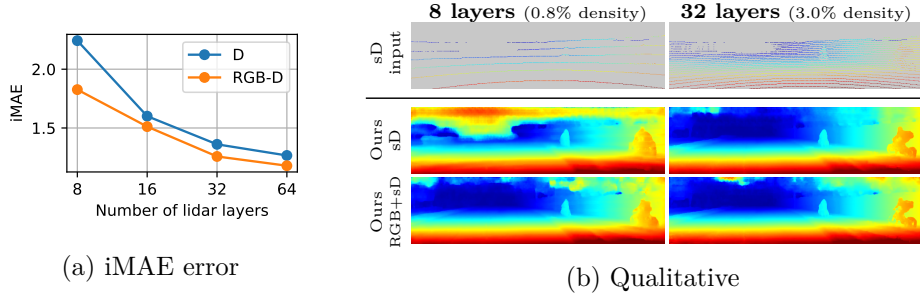


Figure 2.13: **Lidar layer ablation.** Depth completion with simulated fewer layer lidars (downsampling of 64 layers input). Even with only 8 layers (projecting on only 0.8% pixels) the depth map is remarkably completed.

SBNNet (Ren et al., 2018) use custom-designed convolutions and propagate the input validity domain with masks or coordinates, we show that this is unnecessary. On the depth completion task with different input density, Fig. 2.12a, training with 0.1 density indeed is less stable (‘Ours, 0.1’) but when trained with proper varying density scheme (‘Ours, varying’) it outperforms Uhrig et al. (2017), being also significantly simpler.

Fig. 2.12b shows we overpassed state-of-the-art on KITTI depth completion benchmark (Uhrig et al., 2017) at the time of our submission on 3 of the 4 metrics, using sparse depth (sD) alone or with RGB (RGB+sD). Of interest for autonomous driving applications, we show our method is resistant to low Lidar resolution by training and testing with fewer Lidar layers, still reaching low error as seen in Fig. 2.13. In Jaritz et al. (2018b) we also study the ability to infer semantics leading to 44 mIoU with only sD input.

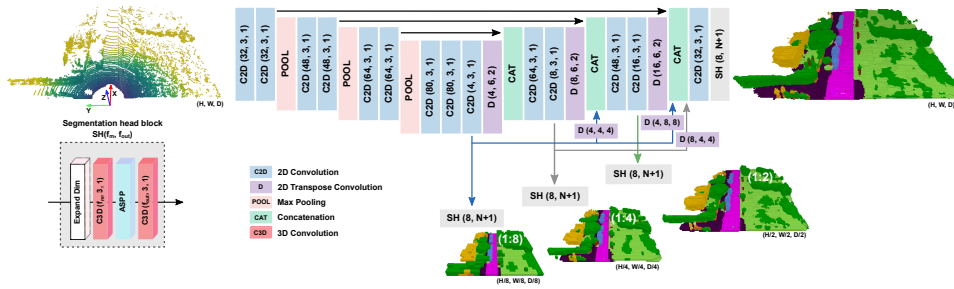


Figure 2.14: **LMSCNet: Lightweight Multiscale Semantic Completion Network.** Despite the 3D task, we use a 2D UNet turning the 3rd spatial dimension to features, and latter retrieve it with 3D segmentation head (gray blocks, see left inset). Convs show (filters, kernel, stride). For lightweight reasons, we restrict the 2D features dimension and use dilated convolutions in the segmentation heads – here, ASPP (Chen et al., 2018a).

2.2.2 Semantic scene completion

In the following two works (Roldão et al., 2020, 2021) we address the problem of *semantic scene completion* (SSC) which consists into inferring a dense semantic 3D representation of a scene from a sparse 3D scan. The task differs from completion or SLAM in that it completes large chunk of missing data and predicts semantic jointly without requiring sequence of Lidar scans. In many cases however, the SSC task can be seen as a semantic task with $N+1$ classes (N semantic classes, +1 *free* class).

Our work on the matter started in 2019. Later that year the task gained interest with the release of SemanticKITTI (Behley et al., 2019) – a dense semantically labeled version of KITTI urban dataset and an alternative to the indoor NYUv2 dataset (Silberman et al., 2012a) –. Very recently, KITTI360 (Liao et al., 2021) was released with a new SSC benchmark, foreseeing exciting novel works.

LMSCNet. In Roldão et al. (2020) we addressed SSC relying on voxelized 3D point cloud, with two major contributions: first, a lightweight architecture leveraging a mix of 2D/3D convolutions; second, a modular multiscale pipeline. Our method is coined LMSCNet for *Lightweight Multiscale 3D Semantic Completion* network.

At the time of this work, little researches addressed SSC, and most works relied on 3D CNNs to process point cloud or depth map as occupancy grid, sometimes in conjunction of 2D CNNs to extract RGB features.

Instead, our LMSCNet (Fig. 2.14) uses a lightweight UNet style architec-

Going further...

Code and data:
<https://github.com/cv-riets/LMSCNet>

Roldão, L., de Charette, R., and Verroust-Blondet, A. (2020). LMSCNet: Lightweight multiscale 3D semantic completion. In *3DV*

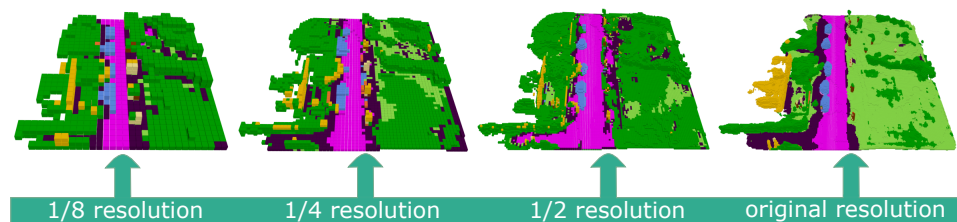


Figure 2.15: **LMSCNet multiscale samples.** To supervise coarser output, we use majority vote pooling from the original resolution ground truth.

ture to predict 3D semantic completion at multiple scales, allowing inference up to 370FPS at the coarsest resolution. Rather than greedy 3D convolutions, we infer here the dense output from the sparse input voxel grid, using a 2D UNet architecture. Though dense convolutions in the encoder imply a dilation of the input manifold (Graham et al., 2018), we found that this is beneficial for 3D semantic completion in all likelihood because of the sparse \mapsto dense nature of the SSC task.

To preserve a lightweight architecture, we use 2D convolutions along the X,Y dimensions, thus turning the height dimension (Z) into a feature dimension. This is significantly different from prior works, since 2D convolutions intentionally lose the 3rd spatial dimension, but are in the meantime significantly lighter operations than their 3D counterparts. Along with standard skip connections, we enhanced information flow in the decoder by concatenating the output of each level to all lower levels (bottom links in Fig. 2.14). This enables using high level features from coarser resolutions, and thus enhance the spatial contextual information. Both this work and the next one (Roldão et al., 2021) highlight context importance.

Because 3D SSC requires to output 4D tensor (3 spatial, 1 semantic), in LMSCNet we introduce 3D segmentation heads, depicted as gray blocks in Fig. 2.14, which are added before the SSC output at each scale. The heads (see left inset, Fig. 2.14) use a series of dense and dilated conv like Atrous Spatial Pyramid Pooling – aka ASPP (Chen et al., 2018a) –, which favor information flow from various receptive fields. The benefit of preceding ASPP with dense 3D convolutions is dual: a) to further densify the feature maps, b) to ward off features from the segmentation heads and the backbone features. In this work we experimentally measured the importance of disentangling segmentation features from backbone features. In fact, disentanglement is key to informative features flow in networks and we further studied that in some of our later works, like Jaritz et al. (2020, 2021); Pizzati et al. (2021a).

Approach	scene completion			semantic scene completion (19 classes)						mIoU↑
	Prec.↑	Rec.↑	IoU↑	road (15.30%)	building (14.1%)	car (3.92%)	terrain (9.17%)	person (0.07%)	traffic-sign (0.08%)	
SSCNet (Song et al., 2017b)	31.71	83.40	29.83	27.55	20.88	10.35	18.16	0	3.67	9.53
*SSCNet-full (Song et al., 2017b)	59.64	75.52	49.98	51.15	34.53	24.26	29.01	<u>0.25</u>	6.73	16.14
TS3D (Garbade et al., 2019)	31.58	<u>84.18</u>	29.81	28.00	23.19	10.72	18.32	0.03	3.52	9.54
TS3D+D (Behley et al., 2019)	25.85	88.25	24.99	27.53	22.05	8.04	20.22	2.33	6.99	10.19
TS3D+D+S (Behley et al., 2019)	<u>80.52</u>	57.65	50.60	62.20	34.12	<u>30.70</u>	33.09	0	<u>6.94</u>	17.70
LMSCNet (ours)	77.11	66.19	<u>55.32</u>	<u>64.04</u>	38.67	29.48	30.77	0	0.54	17.01
LMSCNet-singlescale (ours)	81.55	65.07	56.72	64.80	<u>38.08</u>	30.89	<u>32.05</u>	0	0.84	<u>17.62</u>

* Own implementation. / +D means +DarkNet53Seg / +S means +SATNet

Table 2.2: **Performance on SemanticKITTI hidden test set.** Comparing against all available baselines, our method performs 2nd in mIoU (semantic scene completion) and 1st in IoU (scene completion).

As it is multiscale, LMSCNet provides outputs at several input relative scales of $\frac{1}{2^l} \forall l \in \{0, 1, 2, 3\}$. For each scale l , we train with a cross-entropy loss defined as

$$\mathcal{L}_l = - \sum_{c=0}^N w_c \hat{y}_{i,c} \log \left(\frac{e^{y_{i,c}}}{\sum_{c'}^N e^{y_{i,c'}}} \right), \quad (2.4)$$

where y is the network output, i a voxel index, $\hat{y}_{i,c}$ a one-hot vector (i.e. $\hat{y}_{i,c} = 1$ if voxel i is labeled class c , otherwise $\hat{y}_{i,c} = 0$), and w_c is a coefficient for log class-balancing. We simply optimize the sum of all scale losses: $\mathcal{L} = \sum_{l=0}^3 \alpha_l \mathcal{L}_l$. A sample output at each scale is shown in Fig. 2.15. While it trains using all scales, an interesting property of our architecture is that the latest convolutions can be ablated for faster inference at coarse resolutions.

Experiments. In Tab. 2.2, performance at the time of publication on the popular SSC benchmark of SemanticKITTI (Behley et al., 2019) shows LMSCNet performs second in semantic scene completion (mIoU), and first in terms of scene completion (IoU) with a comfortable margin. In Fig. 2.16 sample SSC outputs show the better performance of LMSCNet against SSCNet-full (Song et al., 2017b).

Of note, in SemanticKITTI grids are 256x256x32 with 0.2m voxel size and both input *and* ground truth are sparse with average density of 6.7% and 65.8%, respectively. Furthermore, beyond LMSCNet good results, looking at its classe-wise performance in Tab. 2.2 – showing only 6 of the 19 classes –, it is noticeable that it underperforms on rare classes which we

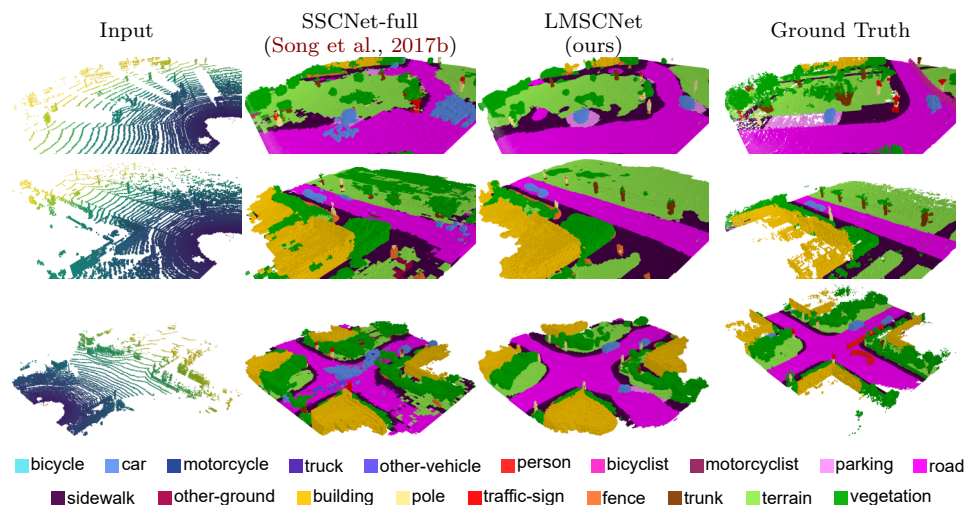


Figure 2.16: **SemanticKITTI validation set.** Compared to SSCNet-full (Song et al., 2017b) – the best open-source baseline –, LMSCNet provides smoother semantics labels and is capable of retrieving finer details. This is evident when looking at the cars or the trees.

impute on the highly class imbalance distribution of the dataset and our simple class-balancing strategy.

However, SSC performance does not illustrate the full lightweight benefit of our work. Network statistics in Tab. 2.17a show LMSCNet is significantly lighter with only 0.35 Millions parameters while others require between 2.6 and 144 times more parameters. Of greater importance for robotics applications is the inference speed. TS3D+Darknet+Satnet (Behley et al., 2019) – 1st on mIoU SemanticKITTI – runs at 1.3 FPS versus 21.3 for ours. In fact the only faster method (SSCNet) performs 1% less mIoU.

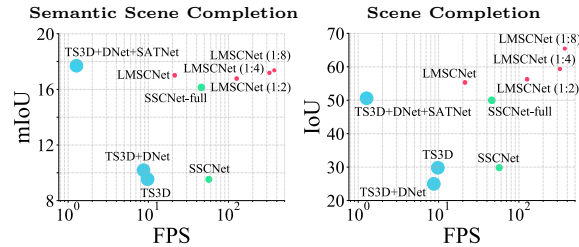
An important benefit of our method, is its multiscale capacity. Ablating the head of the network accordingly, LMSCNet runs as fast as 372 FPS at 1:8 scale. Scatter plots in Fig. 2.17b demonstrate the lightweight capacity and our great performance-speed balance at full size or coarser resolutions, being granted that best methods lie in the top-right corner of each plot.

In Roldão et al. (2020), we also demonstrated our inference is robust to lower Lidar density (eg. 0.14 mIoU for 16 Lidar layers) and generalizes to nuScenes dataset (Caesar et al., 2020b). In our survey (Roldão et al., 2021) we tested LMSCNet on NYUv2 indoor dataset with expected worse performance since 2D convs lose a spatial axis and are thus better suited for outdoor scene where data has main variance along two axes.

Method	Params (M) ↓	FLOPs (G) ↓	FPS ↑
*SSCNet	0.93	82.5	56.9
*SSCNet-full	1.09	769.6	45.9
*TS3D	43.77	2016.7	9.8
*TS3D+D	51.31	847.1	8.7
*TS3D+D+S	50.57	905.2	1.3
LMSCNet	0.35	72.6	21.3
LMSCNet (1:2)	0.32	13.7	126.4
LMSCNet (1:4)	0.28	5.7	323.5
LMSCNet (1:8)	0.24	4.4	372.2

* Own implementation to compute statistics.

(a) Network statistics



(b) Performance vs Inference speed

Figure 2.17: **Network performances.** (a) At full size, LMSCNet has less parameters and is faster than all but SSCNet which performs 1% worse on mIoU. Our multiscale versions – LMSCNet (1:x) – enable very fast inference. (b) Architectures performance versus speed. Notice that TS3D+DNet+SATNet having +0.69 mIoU w.r.t. LMSCNet, is x17 slower.

3D SSC Survey. In Roldão et al. (2021) we published the first semantic scene completion survey, which required a vast and comprehensive analysis of the SSC landscape and multiple experiments. Considering the increase of published papers on SSC (1 in 2017, 33 at the end of 2020), the benefit of our survey paper work is to help researchers to navigate the field as well as to identify new insights and directions still untouched.

While survey are always hard to synthesize in such manuscript, I brush here a few highlights of our findings.

Looking at our listing of all methods in Tab. 2.3, an evident observation is that most methods rely on inputs of geometrical nature (depth, range data, HHA images, occupancy grid, TSDF or point cloud) sometimes along textural data (RGB). While many works address 3D from RGB, none precisely studied SSC only with images, which might be explained by the complexity of the task itself. In (Roldão et al., 2021), when possible, we also computed performance of the methods on both Indoor (ScanNet (Dai et al., 2017), NYUv2 (Silberman et al., 2012a), SUNCG (Song et al., 2017a)) and Outdoor (SemanticKITTI (Behley et al., 2019)) datasets ; which exhibits that none of the outdoor top-performing methods use RGB (all rely on Lidar input *only*).

Going further...

Roldão, L., de Charette, R., and Verroust-Blondet, A. (2021). 3D semantic scene completion: a survey. *IJCV*

	Input Encoding		Architecture	Design choices				Training			Evaluation		Open source							
	RGB	Depth/Range/HHA Other (seg., normals, etc.)		2D	3D	Contextual Awareness	Position Awareness	Fusion Strategies	Lightweight Design	Refinement	End-to-end	Coarse-to-fine	Multi-scale	Adversarial	Losses	NYU ^b	SUNCG ^c	SemanticKITTI ^d	Other ^e	Framework
2017 SSCNet (Song et al., 2017b) ^a			✓	volume	DC				✓					CE	✓	✓	✓		Caffe	✓
2018 Guedes et al. (2018)	✓		✓	volume	DC				✓					CE	✓				-	
ScanComplete (Dai et al., 2018)			✓	volume			GrpConv			✓				ℓ_1 CE	✓	✓			TF	✓
VVNet (Guo and Tong, 2018)	✓	✓		view-volume	DC				✓					CE	✓	✓			TF	✓
Cherabier et al. (2018)	✓		✓	volume	PDA		MSO		✓	✓	✓			CE	✓		✓		-	
VD-CRF (Zhang et al., 2018b)			✓	volume	DC				✓					CE	✓	✓			-	
ESSCNet (Zhang et al., 2018a)			✓	volume			GrpConv Sparse			✓				CE	✓	✓			PyTorch	✓
ASSCNet (Wang et al., 2018)	✓			view-volume	Mscale. CAT						✓			CE	✓	✓			TF	
SATNet (Liu et al., 2018)	✓	✓		view-volume	ASPP		M/L		✓					CE	✓	✓			PyTorch	✓
2019 DDRNet (Li et al., 2019a)	✓	✓		view-volume	LW-ASPP	DC	M	DDR	✓					CE	✓	✓			PyTorch	✓
TS3D (Garbade et al., 2019) ^a	✓		✓	hybrid	DC		E		✓					CE	✓	✓	✓		-	
EdgeNet (Dourado et al., 2020a)	✓		✓	volume	DC	✓	M		✓					CE	✓	✓			-	
SSC-GAN (Chen et al., 2019b)			✓	volume	DC						✓			BCE CE	✓	✓			-	
TS3D+DNet (Behley et al., 2019)	✓	✓		hybrid	DC		E		✓					CE			✓		-	
TS3D+DNet+SATNet (Behley et al., 2019)	✓	✓		hybrid	DC		E		✓					CE			✓		-	
ForkNet (Wang et al., 2019d)			✓	volume	DC						✓			BCE CE	✓	✓			TF	✓
CCPNet (Zhang et al., 2019)			✓	volume	CCP DC			GrpConv	✓	✓				CE	✓	✓			-	
AM ² FNet (Chen et al., 2019a)	✓		✓	hybrid	DC	✓	M		✓	✓				BCE CE	✓				-	
2020 GRFNet (Liu et al., 2020)	✓	✓		view-volume	LW-ASPP DC		M	DDR	✓					CE	✓	✓			-	
Dourado et al. (2020b)	✓		✓	volume	DC	✓	E		✓	✓				CE	✓				-	
AMFNet (Li et al., 2020c)	✓	✓		view-volume	LW-ASPP		L	RAB	✓					CE	✓	✓			-	
PALNet (Li et al., 2020b)	✓		✓	hybrid	FAM DC	✓	M		✓					PA	✓	✓			PyTorch	✓
3DSketch (Chen et al., 2020a)	✓		✓	hybrid	DC	✓	M	DDR	✓		✓			BCE CE CCY	✓	✓			PyTorch	✓
AIC-Net (Li et al., 2020a)	✓	✓		view-volume	FAM AIC		M	Anisotropic	✓					CE	✓	✓			PyTorch	✓
Wang et al. (2020)			✓	volume				Octree-based	✓					BCE CE	✓				-	
L3DSR-Oct (Wang et al., 2019b)			✓	volume				Octree-based		✓				BCE CE	✓	✓	✓		-	
IPF-SPCNet (Zhong and Zeng, 2020)	✓		✓	hybrid					✓					CE	✓				-	
Chen et al. (2020b)			✓	volume	GA Module				✓					BCE CE	✓	✓			-	
LMSCNet (Roldao et al., 2020)			✓	view-volume	MSFA			2D	✓		✓			CE	✓		✓		PyTorch	✓
SCFusion (Wu et al., 2020)			✓	volume	DC				✓		✓			CE	✓	✓			-	
S3CNet (Cheng et al., 2020)	✓	✓	✓	hybrid			L	Sparse	✓	✓				BCE CE PA			✓		-	
JS3C-Net (Yan et al., 2021)			✓	volume				Sparse	✓					CE	✓				PyTorch	✓
Local-DIFs (Rist et al., 2020a)			✓	point-based					✓					BCE CE SCY	✓				-	
2021 SISNet (Cai et al., 2021)	✓		✓	hybrid			M	DDR	✓					CE	✓	✓	✓		PyTorch	

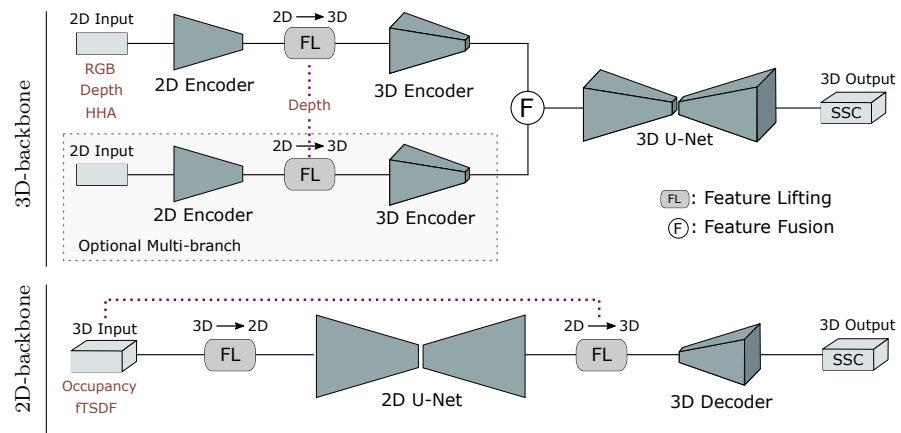
^a SSCNet was significantly extended in (Guo and Tong, 2018; Behley et al., 2019; Roldao et al., 2020). ^b Includes NYUv2 (Silberman et al., 2012b), NYUCAD (Firman et al., 2016). ^c Includes SUNCG (Song et al., 2017b), SUNCG-RGBD (Liu et al., 2018). ^d Includes SemanticKITTI (Behley et al., 2019). ^e Includes (Chang et al., 2017; Dai et al., 2017; Wu et al., 2020; Liu et al., 2018; Armeni et al., 2017) **Contextual Awareness** - DC, Dilated Convolutions. (LW)-ASPP, (Lightweight) Atrous Spatial Pyramid Pooling. CCP, Cascaded Context Pyramid. FAM, Feature Aggregation Module. AIC, Anisotropic Convolutional Module. GA, Global Aggregation. MSFA, Multi-scale Feature Aggregation. PDA, Primal-Dual Algorithm. **Fusion Strategies** - E, Early. M, Middle. L, Late. **Lightweight Design** - GrpConv, Group Convolution. DDR, Dimensional Decomposition Residual Block. RAB, Residual Attention Block. MSO, MultiScale Optimization. **Losses - Geometric**: BCE, Binary Cross Entropy. ℓ_1 , L1 norm. **Semantic**: CE, Cross Entropy. PA, Position Awareness. **Consistency**: CCY, Completion Consistency. SCY, Spatial Semantics Consistency.

Table 2.3: Methods studied in our SSC survey (Roldão et al., 2021).

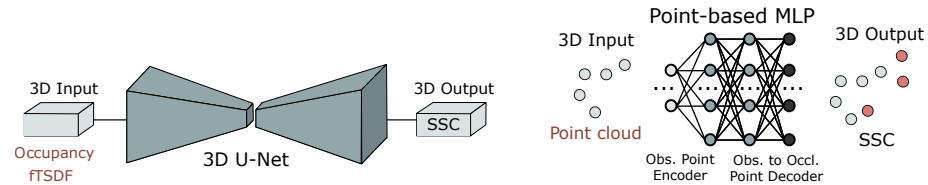
On networks shown in Fig. 2.18, a large part of the existing works leverage *volume* networks (3D CNNs) relying on input voxel grid being either occupancy or (flipped) TSDF. Apart from the recent [Zhong and Zeng \(2020\)](#); [Rist et al. \(2020b\)](#), point-based networks have been little studied for SSC, which is to be imputed first to the complexity of dealing with large scene point clouds, and second to public datasets offering only voxelized ground truth. We argue this refrained point cloud completion despite promising object-oriented works ([Yuan et al., 2018](#)). We ourselves addressed the topic of point-based SSC together with a PhD and an intern, but with no convincing results. On training (see ‘Training’ in Tab. 2.3), our survey also highlights the poor variety of training strategies for SSC, and the common ‘semantic segmentation’ formulation of the SSC task.

On the SSC task itself, our survey highlights biases in the problem formulation. For example, since ground truths originate from devices sensing only the apparent physical shell, SSC fails to predict the real volume of the objects. This is visible even in our LMSCNet outputs (Fig. 2.16) where buildings are predicted as partly empty shells. Specific to dynamic environments, aggregation of moving objects in the ground truth incorrectly penalize the SSC which *cannot* predict individual object motion. This is visible in ground truth of Fig. 2.16 where moving objects produce temporal traces¹. Finally, analyzing the datasets statistics in [Roldão et al. \(2021\)](#) also highlights the long-tail problem of the SSC. Apart from naive class or data balancing, only [Rist et al. \(2020b\)](#) addresses this problem. Future works could also inspire from [Yin et al. \(2021\)](#) which applies guided point sampling.

¹In Fig. 2.16, *best seen on a screen*, only pedestrians are moving (notice red voxels traces in bottom row, ground truth). All cars are parked – thus stationary in ground truth.

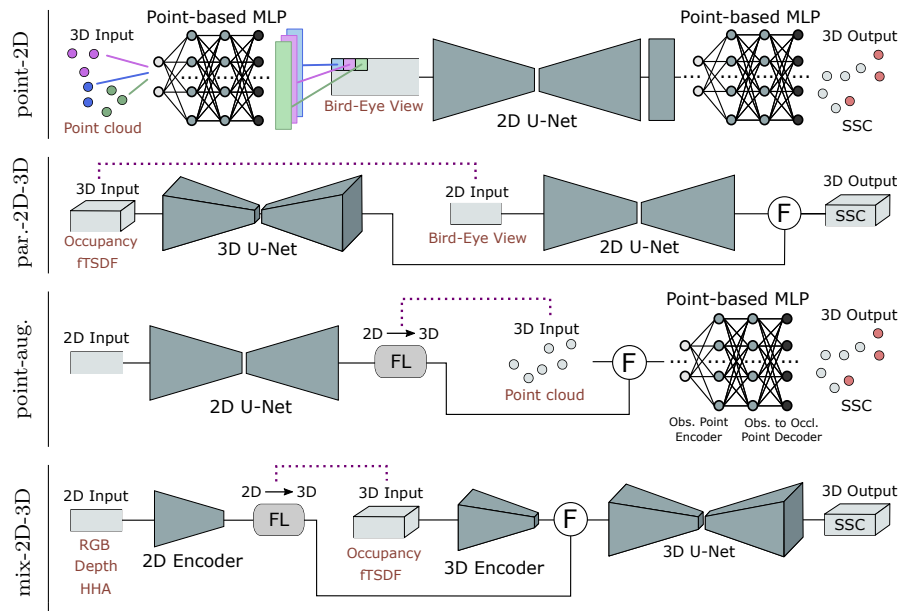


(a) View-Volume Nets., 3D-backbone: (Guo and Tong, 2018; Liu et al., 2018; Li et al., 2019a; Liu et al., 2020; Li et al., 2020c,a), 2D-backbone.: (Roldao et al., 2020)



(b) Volume Nets. (Song et al., 2017b; Guedes et al., 2018; Zhang et al., 2018b,a; Dourado et al., 2020a; Chen et al., 2019b; Wang et al., 2019d; Zhang et al., 2019; Wang et al., 2020; Chen et al., 2020b; Yan et al., 2021; Dourado et al., 2020b; Wu et al., 2020; Cherabier et al., 2018; Dai et al., 2018; Wang et al., 2019b)

(c) Point-based Nets. (Zhong and Zeng, 2020)



(d) Hybrid Nets., point-2D: (Rist et al., 2020a,b), parallel-2D-3D: (Cheng et al., 2020), point-augmented: (Zhong and Zeng, 2020), mix-2D-3D: (Garbade et al., 2019; Behley et al., 2019; Li et al., 2020b; Chen et al., 2020a; Cai et al., 2021)

Figure 2.18: **Architectures for SSC.** Notice the predominance of researches on volume networks, and the little works conducted on point-based networks. \textcircled{F} stands for any type of fusion.

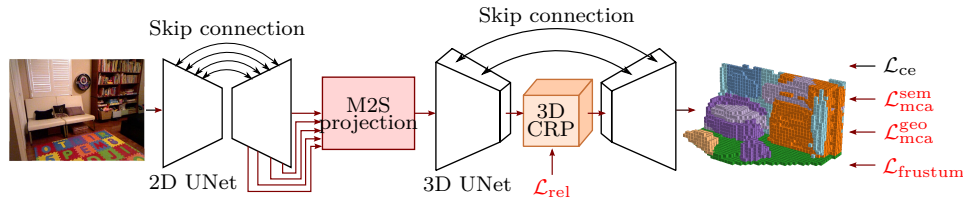


Figure 2.19: **MonoScene**. Our pipeline relies on successive 2D and 3D UNets, bridged with a multi-scale projection module (M2S). In the 3D UNet, we introduce a 3D Context Relation Prior module (3D CRP) to enforce spatio-semantic consistency, and further train our pipeline with a class-/scene-wise loss (\mathcal{L}_{mca}) and a frustum loss ($\mathcal{L}_{frustum}$).

MonoScene. Departing from our survey showing that all SSC rely on 3D input, in Cao and de Charette (2021) (*submitted*) we address 3D SSC from a single RGB image. This task is significantly harder than standard SSC since even sparse 3D input provides a strong geometrical cue – being a subset of the 3D semantic ground truth. Instead, we reconstruct the complete scene geometry and semantics with only a 2D RGB image as input.

Our overall pipeline is shown in Fig. 2.19, and is composed of a 2D UNet (here, a pretrained EfficientNetB7) and a shallow 3D UNet. To boost 2D contextual knowledge we unproject 2D UNet multiscale features (M2S) to 3D, along each pixel line-of-sight, and introduce a new 3D Contextual Relation Prior layer (3D CRP) that enforces 3D spatio-semantic consistency. Finally, new losses are added to ease the SSC task. We briefly describe each component here.

Multi-scale 2D-3D projection (M2S). To avoid the ill-posed problem of finding 2D-3D correspondences, we propose a mechanism where 2D features project to *all possible* 3D correspondences. This intuitively lets the 3D network discover guidance from the ensemble of 2D multiscale features, working toward a single unique consistent 3D resolution. Our process is illustrated in Fig. 2.20a. In practice, considering known intrinsics, we project the coordinates of the centroids of all voxels and sample the corresponding features (if any) in the 2D decoder feature map $F_{2D}^{1:1}$, obtaining a 3D feature map $F_{3D}^{1:1}$. Because 2D to 3D holds inherent ambiguities, it is important to enhance contextual information flow in the network. We do so by repeating the 2D sampling at all scales, summing all obtained sparse 3D features maps into a single F_{3D} feature map.

We found a similar-in-spirit idea in Popov et al. (2020), called ‘ray-tracing skip connection’, though in contrast our proposal favors better 2D-3D flow as it enables 2D-3D disentangled resolutions – leading to better results.

Going further...

Code:
<https://github.com/cv-rits/MonoScene>

Cao, A.-Q. and de Charette, R. (2021). MonoScene: Monocular 3d semantic scene completion. *arXiv (submitted)*

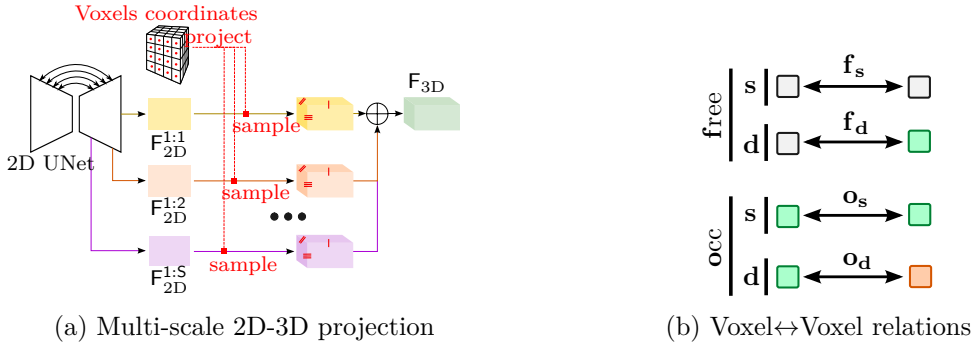


Figure 2.20: **2D-3D features and Relation prior.** (a) We boost contextual 2D-3D flow by projecting multiscale 2D features along their individual 3D line-of-sight. (b) Our 3D CRP module learns spatio-semantic relations prior. Here, 2D illustration of 4-ways bilateral Voxel↔Voxel relations are shown. (□ free voxel, □ occupied voxels - colors denote semantic class)

3D Context Relation Prior (3D CRP). Our 3D CRP layer, inserted at the 3D UNet bottleneck, enforces 3D spatial-semantic consistency by learning to predict the relationship of voxels in the scene. Doing so provides additional guidance to the network to learn inter-/intra- spatio semantic relations.

In our work, we consider 4 bilateral voxel↔voxel relations, illustrated in Fig. 2.20b, grouped into **free** and **occupied** corresponding respectively to, ‘at least one voxel is free’ and ‘both voxels are occupied’. For each group we encode whether the voxels semantic classes are **similar** or **different**, leading to the 4 relations: $\mathcal{M} = \{\mathbf{f}_s, \mathbf{f}_d, \mathbf{o}_s, \mathbf{o}_d\}$. In practice instead, our 3D CRP learns supervoxel↔voxel relations which are more compact and provide better guidance, but not detailed here for brevity.

In [Cao and de Charette \(2021\)](#) we show our layer can work with arbitrary number of relation priors, supervised or unsupervised.

SSC losses. We introduce two losses to train our SSC task.

First, to explicitly let the network be aware of the SSC performance, we build upon the binary affinity loss ([Yu et al., 2020](#)) and introduce a multi-class version directly optimizing the scene- *and* class- wise metrics. Our loss optimizes the class-wise Precision (P_c), Recall (R_c) and specificity (S_c) where P_c and R_c optimize performance of similar-class voxels, and S_c optimizes the dissimilar-class voxels. We define the general \mathcal{L}_{mca} to maximize the above class-wise metrics. Considering ground truth p and prediction \hat{p} it writes:

$$\mathcal{L}_{\text{mca}}(\hat{p}, p) = -\frac{1}{C} \sum_{c=1}^C (P_c(\hat{p}, p) + R_c(\hat{p}, p) + S_c(\hat{p}, p)). \quad (2.5)$$

In practice, we optimize both semantics ($\mathcal{L}_{\text{mca}}^{\text{sem}}$) and geometry ($\mathcal{L}_{\text{mca}}^{\text{geo}}$) with our multi-class loss, each one with its corresponding ground truth.

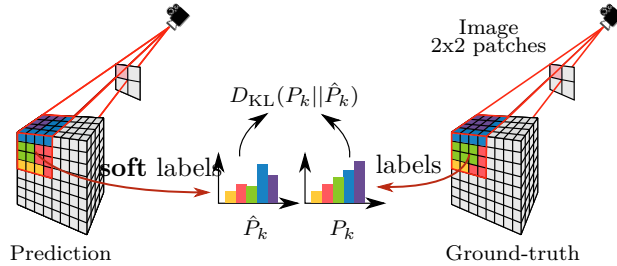


Figure 2.21: **Frustum loss.** We align the distribution statistics of *soft labels* in local frustums (here, 2×2) w.r.t. its corresponding local ground truth. This provides cues to the network for occlusions disambiguation.

Our second loss inspires from the fact that disambiguation of occlusions is impossible and that we observed that occluded voxels tend to be predicted as part of the object that obscures them. To mitigate this effect, we propose a novel Frustum loss function that explicitly optimizes the class distribution per frustum, as illustrated in Fig. 2.21. Rather than optimizing the camera frustum distribution directly, we divide the input image into $\ell \times \ell$ local patches of equal size and apply our loss on each local frustum (the union of the individual pixels frustum in the patch). Intuitively, aligning the frustums distributions provide additional cues to the network on the scene visible and *occluded* structure, giving a sense on what is likely to be occluded (eg. cars are likely to occlude road). In practice, considering a frustum k we apply a soft-alignment on the predicted semantic probabilities (\hat{P}_k) of all the voxels in k w.r.t. the corresponding distribution (P_k):

$$\mathcal{L}_{\text{frustum}} = \sum_{k=1}^{\ell^2} D_{\text{KL}}(P_k || \hat{P}_k), \quad (2.6)$$

$$= \sum_{k=1}^{\ell^2} \sum_{c \in C^k} P_k(c) \log \frac{P_k(c)}{\hat{P}_k(c)}. \quad (2.7)$$

Because frustums image small scene portions, considering all classes (C), some may be locally missing making KL *undefined*. Hence, notice C^k refers to classes *defined* in the local frustum k .

Experiments. We evaluate our method on both indoor NYUv2 (Silberman et al., 2012a) and outdoor SemanticKitti (Behley et al., 2019). As baselines, we chose the best opensource ones available – selecting two indoors designed methods, 3DSketch (Chen et al., 2020a) and AICNet (Li et al., 2020a), and two outdoors designed, LMSCNet (Roldão et al., 2020) and JS3CNet (Yan et al., 2021). Since baselines require 3D inputs (which would provide them with an unfair geometric advantage), we predict the required 3D input

NYUv2 (test set)													
Method	Input	scene completion			semantic scene completion (11 classes)							mIoU	
		Prec.	Rec.	IoU	ceiling (1.37%)	floor (17.58%)	window (1.99%)	chair (3.01%)	sofa (4.70%)	table (4.31%)	furniture (30.04%)		
2D	LMSCNet	\hat{x}^{occ}	56.78	45.74	33.93	4.49	88.41	0.25	3.94	15.44	6.57	14.51	15.88
	AICNet	$x^{\text{RGB}}, \hat{x}^{\text{depth}}$	35.63	81.90	30.03	7.58	82.97	0.05	6.93	22.92	11.11	15.90	18.15
	3DSketch	$x^{\text{RGB}}, \hat{x}^{\text{TSDf}}$	47.03	68.42	38.64	8.53	90.45	5.67	10.64	29.21	13.88	23.83	22.91
	Ours	x^{RGB}	<u>55.17</u>	64.93	42.51	8.89	93.50	12.57	13.72	36.11	15.13	27.96	26.94
Semantic KITTI (hidden test set)													
Method	Input	scene completion			semantic scene completion (19 classes)							mIoU	
		Prec.	Rec.	IoU	road (15.30%)	building (14.1%)	car (3.92%)	bicycle (0.08%)	bicyclist (0.07%)	motorcyclist (0.05%)	pole (0.29%)		
2D	LMSCNet	\hat{x}^{occ}	50.15	45.61	31.38	46.70	10.30	14.30	0.00	0.00	0.00	0.00	7.07
	3DSketch	$x^{\text{RGB}}, \hat{x}^{\text{TSDf}}$	29.77	73.22	26.85	37.70	12.10	17.10	0.00	0.00	0.00	0.00	6.23
	AICNet	$x^{\text{RGB}}, \hat{x}^{\text{depth}}$	25.12	83.51	23.93	15.30	0.00	0.00	0.00	9.60	1.90	0.10	7.09
	JS3C-Net	\hat{x}^{pts}	47.00	55.15	34.00	<u>47.30</u>	<u>12.70</u>	20.10	0.00	0.20	0.20	1.90	<u>8.97</u>
	Ours	x^{RGB}	<u>47.03</u>	55.53	34.16	54.70	14.40	<u>18.80</u>	0.50	<u>1.40</u>	<u>0.40</u>	3.30	11.08

Table 2.4: **Comparative performance.** We report metrics on the indoor NYUv2 (top) and outdoor Semantic KITTI (bottom), showing that despite the various datasets we significantly outperform other SSC 2D-adapted baselines, in both mIoU and IoU.

directly from the RGB image, relying on the best found methods (details in our publication). Notation for predicted inputs is with a hat, eg. \hat{x}^{depth} .

Tab. 2.4 reports the performance on NYUv2 and SemanticKITTI. In both settings we outperform the mIoU of all 2D adapted baselines by a significant margin of +4.03 on NYUv2 and +2.11 on SemanticKITTI. The improved or on par IoU (+3.87 and +0.16) demonstrates our network captures the scene geometry despite a single RGB input – avoiding the naive mIoU increase by lowering the IoU. On individual classes, we perform either best or second. Specifically, our method excels at large structural classes on both datasets, while on SemanticKITTI, we get outperformed mostly on small moving objects classes (car, motorcycle, person, bicyclist, etc.). We ascribe this to the moving objects aggregation in SemanticKITTI ground truth, also highlighted in Roldão et al. (2020); Rist et al. (2020a), which requires to predict the individual objects motion. We argue the later is harder when relying on RGB input only. Some

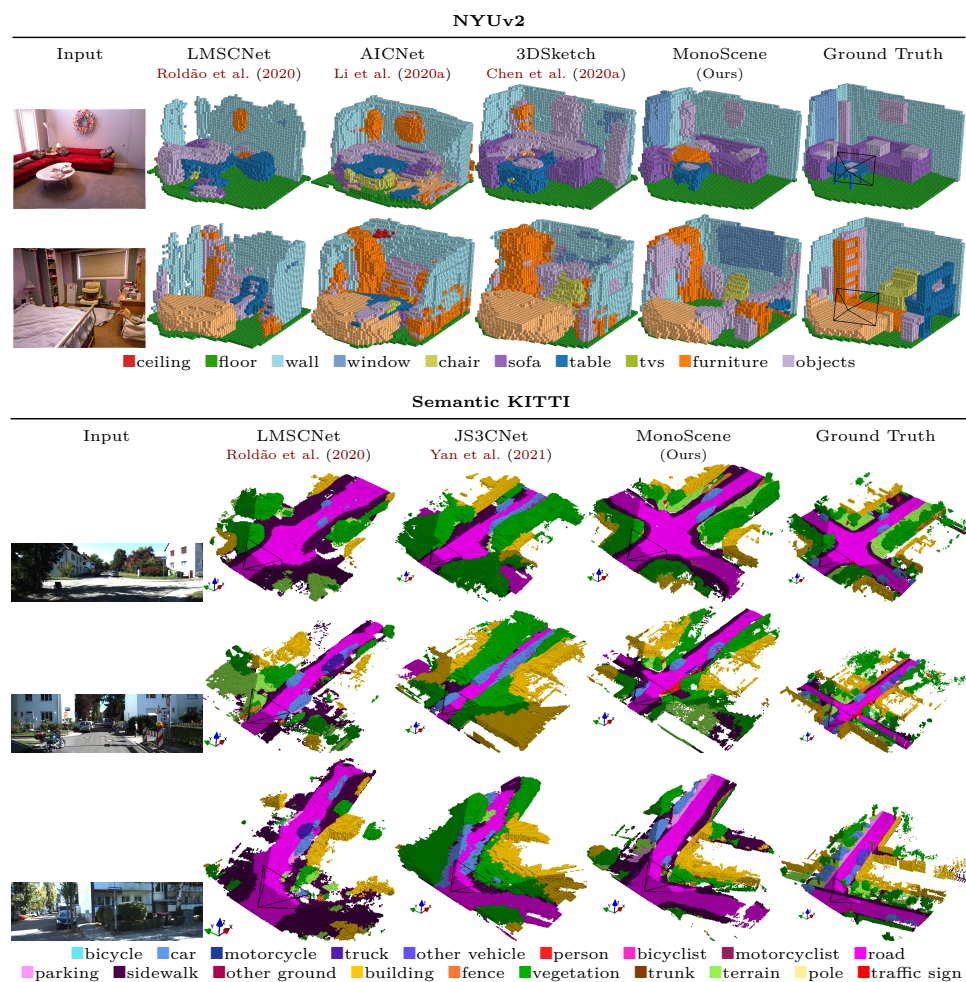


Figure 2.22: **Qualitative results.** Comparison of our MonoScene outputs on NYUv2 (top) or SemanticKITTI (bottom) show we better match ground truth (rightmost). In general our method better approximates the scene structure (notice the road layout in bottom results), better reconstructing also thin objects (NYUv2: chair, table, Sem.KITTI: cars). The camera position is highlighted in the ground truth. Notice our method still reconstructs plausible scene even outside of the camera FOV (darker voxels, bottom).

Method	NYUv2		SemanticKITTI		2D scales	NYUv2	
	IoU \uparrow	mIoU \uparrow	IoU \uparrow	mIoU \uparrow		IoU \uparrow	mIoU \uparrow
Ours	<u>42.51</u> ± 0.15	26.94 ± 0.10	37.12 ± 0.15	11.50 ± 0.14			
w/o $\mathcal{L}_{\text{frustum}}$	41.90 ± 0.26	<u>26.37</u> ± 0.16	36.74 ± 0.33	11.11 ± 0.24			
w/o M2S	41.57 ± 0.11	25.61 ± 0.43	36.35 ± 0.15	10.61 ± 0.05	1, 2, 4, 8	42.51 ± 0.15	26.94 ± 0.10
w/o $\mathcal{L}_{\text{mca}}^{\text{sem}}$	42.82 ± 0.22	25.33 ± 0.26	<u>36.78</u> ± 0.34	9.89 ± 0.11	1, 2, 4	<u>42.08</u> ± 0.69	<u>26.28</u> ± 0.24
w/o $\mathcal{L}_{\text{mca}}^{\text{geo}}$	40.96 ± 0.28	26.34 ± 0.23	34.92 ± 0.34	<u>11.35</u> ± 0.22	1, 2	41.56 ± 0.18	25.66 ± 0.21
w/o 3D CRP	41.39 ± 0.08	26.27 ± 0.15	36.20 ± 0.19	10.96 ± 0.21	1	41.57 ± 0.11	25.61 ± 0.43

(a) Component ablations

(b) Multiscale 2D-3D proj.

Table 2.5: **Ablations (3 runs average)**. (a) Ablating our components show all contribute to semantics (up to +1.61 mIoU) or completion (up to +2, 2 IoU) in NYUv2 and Sem.KITTI (val sets). (b) Using more scales for 2D-3D projection perform best and tend to lower the mIoU variance.

qualitative results are in Fig. 2.22 and advocate that our method is capable of predicting fine structures (note the furnitures in NYUv2 and cars in Sem.KITTI). In Semantic KITTI, it is interesting to note that while the scene FOV is larger than the camera FOV (shown in ground truth column), our method still outputs plausible 3D scene structure outside of the camera FOV (darker voxels in Sem.KITTI). We attribute this to our Frustum loss, providing global class distribution insights to the network.

Tab. 2.5a reports the 3 runs average when removing either of our components, showing that in average all contribute to the increase of performance in both datasets. While our publication thoroughly evaluates all detailed contributions, we describe in Tab. 2.5b only the effect of the multiscale 2D-3D projection, by ablating the scales at which features are projected to the 3D features map. Note again here, that multiscale 2D features project to a single scale 3D (cf. Fig. 2.20a). Results validate our intuition, since more 2D scales projections seem to help the 3D network disambiguation.

Weakly supervised vision

Contents

3.1	Dealing with fewer labels	34
3.1.1	Generative networks	34
3.1.2	Cross-modal learning	41
3.2	Dealing with fewer data	49
3.3	Supervision from action	54
3.3.1	DRL with dense reward	56
3.3.2	DRL with sparse reward	59

Irrefutably, supervised learning is a dead end for computer vision because it relies on costly human-biased labeling, and assumes all conditions are in the training sets. Both of these requirements are unbearable in the long term, and we must relax the need of supervision. For example, mobile robots like autonomous driving must operate safely in all conditions – some of which are rare events, hard to capture or label. While truly unsupervised learning is a unicorn, we focus on our vision algorithms requiring weak or ‘no’ supervision. The section is organized in twofold: leveraging vision with fewer labels, or using action rewards as supervision.

In the first part (Sec. 3.1), we address research that deals with labels scarcity following two different paradigms. Either using generative adversarial network (GAN) to hallucinate new training data, or relying on the unsupervised discovery of statistics during training.

In the second part (Sec. 3.2), we address vision with fewer data, aka few shots, where GANs are used to perform few-shot image-to-image translation.

In the third part (Sec. 3.3) which is drastically different, we investigate supervision from action where virtual agents evolve in interactive simulators and receive rewards for their past action – leveraging reinforcement learning.

Of note, some of the works in Sec. 4 also address weak-/un- supervised vision but in a physics-guided fashion and are thus not listed here.

3.1 Dealing with fewer labels

The research in this section originates mainly from PhD Maximilian Jaritz who defended in 2020, to a lesser extent from PhD student Fabio Pizzati (defense planned in 2022), and also includes active collaborators from Inria, Univ. of Parma, Valeo, Valeo AI and Vislab Ambarella.

In these works we consider a labeled source domain X and a semi or un-labeled domain Y for which we aim at solving some vision task. Our researches roughly leverage two distinct line of works. In Pizzati et al. (2020b); Dell’Eva et al. (2021) (Sec. 3.1.1), we leverage generative networks to learn the mapping $F : X \mapsto Y$ and benefit from the label-consistency property¹ to train ‘supervisedly’ on the translations. In Jaritz et al. (2020, 2021) (Sec. 3.1.2), we consider training supervisedly on samples from X while unsupervisedly discovering statistical similarities in Y .

Similarly to work in Sec. 4 these researches were applied to vision algorithms in adverse lighting or weather conditions – a central research interest of my work – for autonomous driving.

3.1.1 Generative networks

The following two works (Pizzati et al., 2020b; Dell’Eva et al., 2021) contribute somehow equally to the field of image-to-image translation (i2i) and unsupervised domain adaptation (UDA). Both of these works leverage high level knowledge about the domains at stake.

Going further...

Pizzati, F., Charette, R. d., Zaccaria, M., and Cerri, P. (2020b). Domain bridge for unpaired image-to-image translation and unsupervised domain adaptation. In *WACV*

Domain bridge for transfer learning. In Pizzati et al. (2020b), we address i2i translation with GANs when source and target domains are far. Our research originated from the observation that i2i networks collapse when networks fail to map representations in source and target. We found that this happens more often when source-target cumulate gaps such as weather, sensor setup, viewpoint changes, etc. A practical example is learning the Cityscapes \rightarrow BDD_{rain} mapping which collapse with popular i2i like CycleGAN (Zhu et al., 2017) or MUNIT (Huang et al., 2018a).

Our work made two contributions somehow independent: an automatic domains bridging strategy that boost i2i realism, and an Unsupervised Domain Adaptation (UDA) strategy easing transfer learning.

¹Because generative networks aims at preserving content, the source domain labels are preserved when translated.

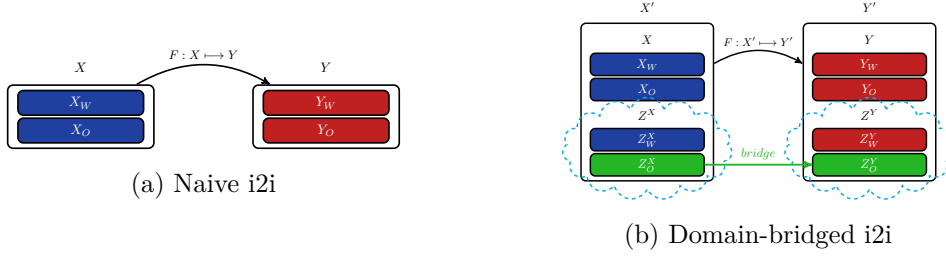


Figure 3.1: **Domain bridging.** When domains gap is large, naive image-to-image (i2i) translation (a) may fail. Instead, we ease i2i (b) by leveraging automatically web-crawled images (Z^X, Z^Y) to bridge domain gap.

Instead of learning the mapping $F : X \mapsto Y$ like naive i2i (Fig. 3.1a), we resonated from domains decomposition and add images from bridge domains Z^X and Z^Y that share characteristics with source and target, respectively. Considering the case of weather changes in X and Y , source and target may not only encompass weather changes (X_W and Y_W , respectively) but also other changes (X_O and Y_O). In this work we argued that more a stable i2i is obtained when domain gap is minimized. Adding Z^X and Z^Y in source and target, respectively, fulfills that role and subsequently acts as a domain bridge (Fig. 3.1b). The criteria to choose $Z^X = \{Z^X_W, Z^X_O\}$ and $Z^Y = \{Z^Y_W, Z^Y_O\}$ are:

$$\max |Z^X_W \cap X_W|, \quad (3.1)$$

$$\max |Z^Y_W \cap Y_W|, \quad (3.2)$$

$$\max |Z^X_O \cap Z^Y_O|, \quad (3.3)$$

where $|\cdot|$ is the set cardinality. This intuitively translates into Z^X and Z^Y having the same weather conditions than X and Y , respectively, and Z^X and Z^Y sharing similar *others* conditions (sensor setup, locations, etc.). This reduces the divergence (eg. Kullback-Leibler) of our new domain sets $X' = \{X, Z^X\}$ and $Y' = \{Y, Z^Y\}$ w.r.t. the original domains, $\text{KL}(P_{X'}, P_{Y'}) < \text{KL}(P_X, P_Y)$, to ease the i2i task.

In practice, our bridge domains Z^X and Z^Y are extracted from automatically web-crawled images. We leverage a YouTube channel having hours long dashcam sequences and split movies into *clear* (Z^X) and *rainy* (Z^Y) images. Because they are recorded with the same setup and often in the same location, they meet criteria in Eq. 3.3, and the Z^X/Z^Y gap is fairly reduced to weather changes only (Eqs. 3.1,3.2). Translations in Fig. 3.3a show that when bridged with Z , MUNIT is able to properly learn clear to rainy translations with realistic reflections, opposite to standard MUNIT that collapsed presumably because of the large domain gap.

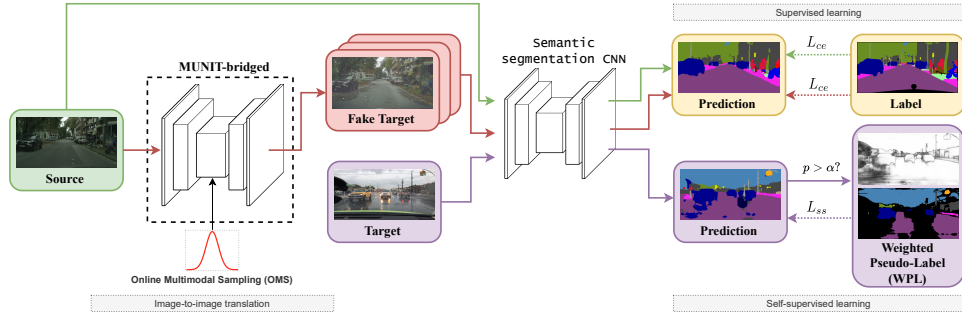


Figure 3.2: **UDA pipeline.** We leverage our domain-bridged i2i, Online Multimodal Sampling (OMS) and Weight Pseudo Label (WPL) strategies. We train alternately on source, translations (fake target), and target.

To ease transfer learning from clear to rainy, we leverage our MUNIT-bridged translations along with a new UDA strategy. The UDA pipeline is in Fig. 3.2. First, we leverage the multi-modal capacity of MUNIT and apply Online Multimodal Sampling (OMS) such that the task network is fed with random style as input, increasing variability, and thus robustness. Second, we leverage Pseudo Label (PL) (Lee et al., 2013) – a self-supervised technique aligning source and target distributions. The principle is to self-train a network whenever its pixel-wise prediction confidence is above some threshold α , thus reinforcing the network own beliefs. Instead than discrete thresholding, we introduced a new Weighted Pseudo-Label (WPL) strategy which regresses α within the network optimization process and applies a linear pixel-wise weighting. For brevity, we refer to Pizzati et al. (2020b) for details. Overall, the optimization of α leads to a pseudo-label expansion during training, illustrated in Fig. 3.3b. It shows that early in the training α is conservative, including as pseudo-labels only pixels with very high confidence, while after along training α is reduced to increase supervision.

Performance. We evaluate our proposal on the task of translating images, as well as on the unsupervised domain adaptation task, leveraging Cityscapes (Cordts et al., 2016) and the rainy set of Berkeley Deep Drive dataset (Xu et al., 2017). On Cityscapes \rightarrow BDD $_{\text{rain}}$, leveraging our MUNIT-bridged translations along with the OMS and WPL strategy we compare against the best two UDA baselines at the time: BDL (Li et al., 2019b) and AdaptSegNet (Tsai et al., 2018). Performance in Fig. 3.3c show we significantly outperform the baseline trained on source only (+8.37 mIoU), and slightly over BDL (+0.44) but with a much simpler UDA strategy.

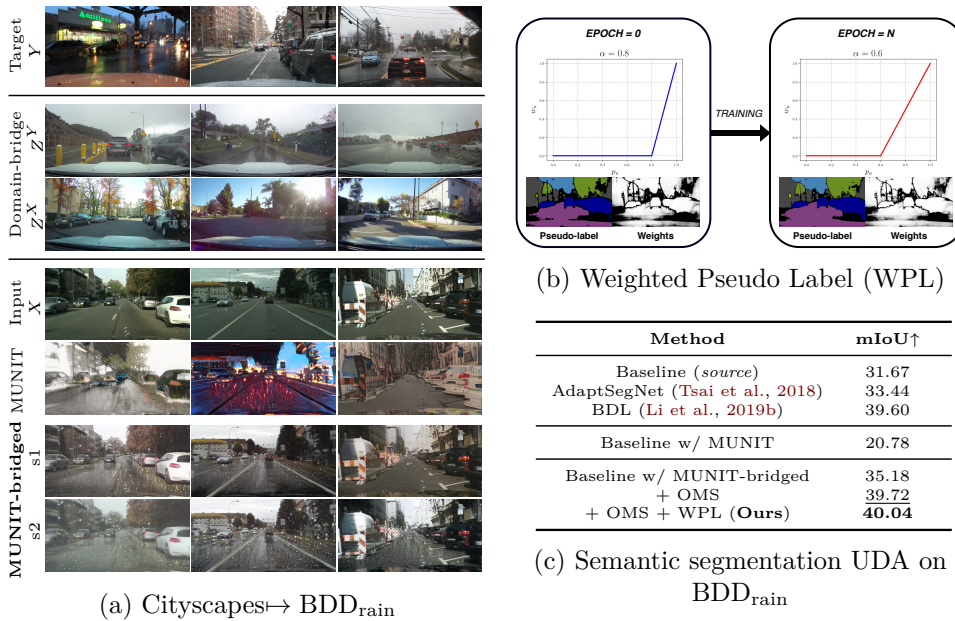


Figure 3.3: **Performance of domain-bridged UDA.** (a) Outputs of MUNIT (Huang et al., 2018a) and our MUNIT bridged. Samples from target domain Y and domain-bridge Z are shown above. Notice, Z^X/Z^Y have similar viewpoints and scenery, with only weather changes as gap. While MUNIT collapses drastically, when bridged (bottom 2 rows) realistic rainy traits are learned (reflections, drops, etc.). (b) Our WPL strategy acts as a pseudo-label expansion during training. (c) UDA on BDD_{rain} (Xu et al., 2017), considering source only (baseline), two UDA baselines, training with MUNIT translations (baseline w/ MUNIT) or training with our MUNIT-bridged images and our UDA strategies.

Leveraging local domains. In the previous work we introduced two new domains Z^X and Z^Y somehow lying between source and target to bridge the translation task. Instead, in Dell’Eva et al. (2021) we reason about high-level knowledge of source and target and *hallucinate a new domain* which significantly boosts target tasks *without ever seeing target* at training.

Since i2i seeks to map global changes while preserving content, they excel at learning complex changes – winter↔summer, painting style, etc. – but struggle to learn subtle local changes even when consistent across images. Our intuition is that we can leverage high-level knowledge about domain-specific spatial characteristics, which we refer to as *local domain*, translating the latter in the source domain X to hallucinate X' . A simple example of pairs of local domains is in Fig. 3.4 with markings/road, snow/no-snow, etc. For example, continuously translating markings to

Going further...

Dell’Eva, A., Pizzati, F., Bertozzi, M., and de Charette, R. (2021). Leveraging local domains for image-to-image translation. In *VISAPP*

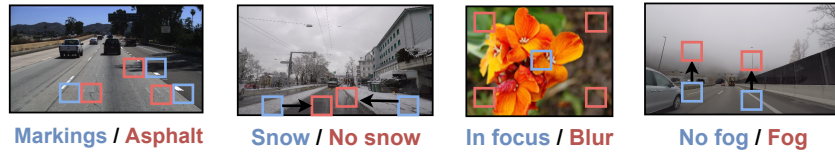


Figure 3.4: **Local domains.** We introduce *local domains* as being domain-specific local characteristics and propose to benefit from local domain translation to boost transfer learning, training on source-translated only without seeing target.

asphalt on the US highway TuSimple dataset is an efficient and simple way to accommodate to IDD (Varma et al., 2019) – an Indian unstructured environment dataset with deteriorated road and lane markings.

In our work, we consider source X and target Y to be the composition of sub domains, some of which are local domains of interest denoted with Greek letter subscript $\alpha, \beta, \dots, \omega$ and the remaining is denoted with o (where "o" stands for others). We assume the case where X and Y share at least a local domain, say α , for example $X = \{X_o, X_\alpha, X_\beta\}$ and $Y = \{Y_o, Y_\alpha\}$. Instead of learning $X \mapsto Y$, our idea is to learn local domain mappings, such as $X_\beta \mapsto X_\alpha$. If such mapping is applied systematically on all samples from X , we get a new domain X' without X_β , so:

$$X' = \{X_o, X_\alpha\}, \quad (3.4)$$

where domain X' is never seen and thus hallucinated. Considering that X' and Y have lower local domains discrepancy they are subsequently closer, i.e. $\text{KL}(Y, X') < \text{KL}(Y, X)$.

In practice, we leverage a GAN trained in a patch-based manner. For each image x , we geometrically guide the patches extraction from a local domain mask $M(x)$ – often simply derivable from the image labels. We also demonstrate that a continuous z -parametrized $X_\beta \mapsto X_\alpha$ geometrical interpolation can be learned with a Variational Auto Encoder (VAE) and a patch-wise composite blending. The overall architecture is in Fig. 3.5, with details in Dell’Eva et al. (2021).

Experiments. In Dell’Eva et al. (2021), we evaluate our method on 3 different tasks: lane degradation, snow addition and deblurring, leveraging the local domains shown in Fig. 3.4, and evaluating our translations both against i2i baselines and on proxy tasks. We report only the first two tasks here. We train with *only 15 images* and 30 patches per image. At inference, the GAN is simply processing the whole image.

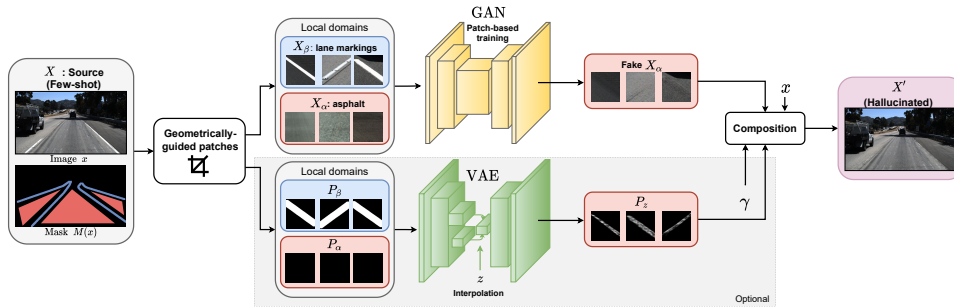


Figure 3.5: **i2i with local domains.** Leveraging local domains priors we learn to translate the latter, from simple geometrical guidance and interpolation when ad-hoc. The resulting domain X' is hallucinated and intended to reduce the gap with unseen target Y .

Fig. 3.6 shows sample local domains translations for *continuous* lane \mapsto asphalt trained on [TuSimple \(2017\)](#) and *discrete* non-snowy-road \mapsto snowy-sidewalk trained on [ACDC_{snow} \(Sakaridis et al., 2021\)](#). On *lane degradation*, zooming in the images show that degradation is effectively increasing along the z parametrized space. In [Dell’Eva et al. \(2021\)](#) GAN metrics show our translations outperform recent continuous GANs by a large margin (-20 FID and -0.10 LPIPS). On *snow addition*, the task consists into adding snow on the road taking as model the snowy sidewalks. Since our interpolation is geometric, it would not make any sense here and is simply deactivated.

To validate the benefit of our local domain i2i, we use the above trained networks to translate TuSimple and Cityscapes into ‘TuSimple with degraded lane markings’ and ‘Cityscapes with snow’, respectively. In Fig. 3.7 we show performance of state-of-the-art lane detectors and semantic segmentation networks on test sets of IDD and ACDC_{snow}² either trained on original TuSimple or Cityscapes, respectively, or on our translated versions. In all tested metrics but one, we outperform the original results – often by a significant margin, demonstrating the benefit of local domain translation to boost transfer learning.

Despite a naive key idea – after all, we only translate patches – this work has major interests: a) we use virtually free high-level domains knowledge, b) only source is used to train, c) our pipeline can train few-shots.

The following works depart from generative networks and study transfer learning without any target-domain reconstruction, but leveraging self-supervised (aka *unsupervised*) features alignment.

²We hand labeled IDD lane markings 110 images. ACDC_{snow} comes with labels.

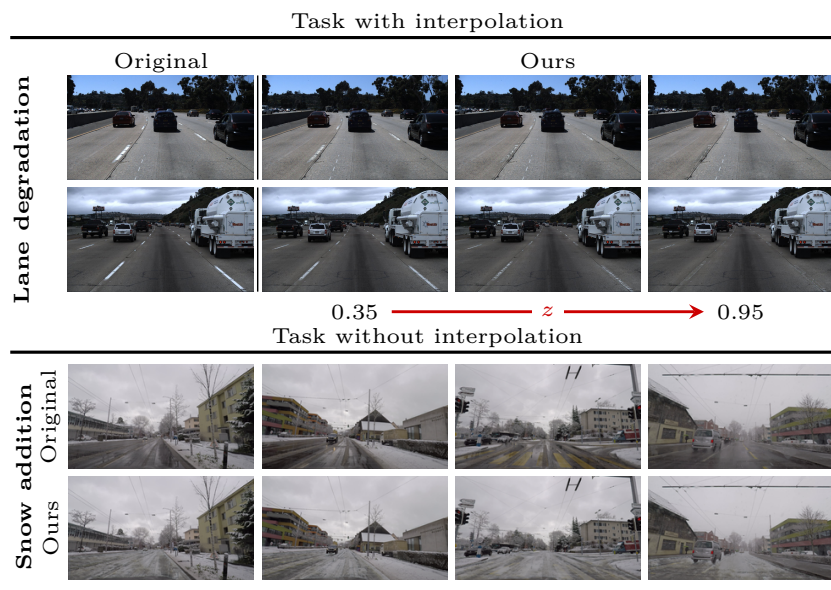


Figure 3.6: **Qualitative local domain translations.** Leveraging local domains shown in Fig. 3.4 and training only on 15 images, we perform continuous ‘lane degradation’ translation on TuSimple (2017), and discrete ‘snow addition on road’ on images from ACDC_{snow} (Sakaridis et al., 2021). Notice the local lane degradation (top, cols 3-4) and the snow addition on road preserving car traces (bottom, row ‘Ours’).

Detector	Trans-lation	TuSimple			IDD		
		Acc.↑	FP↓	FN↓	Acc.↑	FP↓	FN↓
SCNN <small>(Pan et al., 2017)</small>	none	0.95	0.05	0.07	0.62	0.54	0.74
	Ours	0.95	0.06	0.07	0.73	0.45	0.58
RESA <small>(Zheng et al., 2021)</small>	none	0.95	0.06	0.07	0.64	0.72	0.80
	Ours	0.95	0.06	0.07	0.67	0.69	0.76

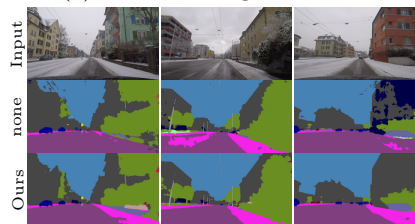
(a) Lane detector



(b) SCNN lane detection on IDD

Model	Trans-lation	ACDC _{snow}		
		road↑	sidewalk↑	mIoU↑
DeepLabv3+ <small>(Chen et al., 2018b)</small>	none	75.0	39.5	45.3
	Ours	80.6	49.5	47.6
PSANet <small>(Zhao et al., 2018)</small>	none	74.3	30.7	43.0
	Ours	74.0	36.3	43.9
OCRNet <small>(Yuan et al., 2020)</small>	none	82.3	45.6	54.5
	Ours	82.8	54.7	55.5

(c) Semantic segmentation



(d) DeepLabv3+ on ACDC_{snow}

Figure 3.7: **Proxy tasks performance.** (a-b) State-of-the-art lane detectors when trained on original TuSimple (*none*) or on our lane degraded translation (*ours*). (c-d) State-of-the-art semantic segmentation networks when trained on original Cityscapes (*none*) or on our road/sidewalk snowy Cityscapes (*Ours*).

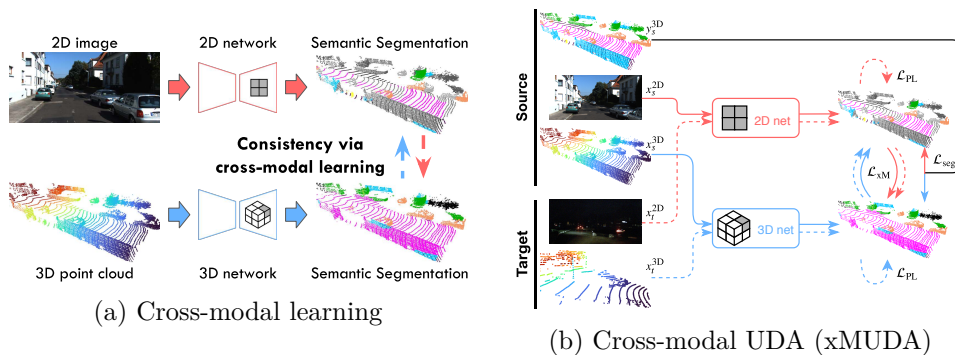


Figure 3.8: **Cross-modal learning.** (a) Leveraging cross-modal learning (here, 2D/3D) our research enforces consistency in the uni-modal predictions. On Unsupervised Domain Adaptation (UDA) (b), xMUDA learns from supervision on the source domain (plain lines) and self-supervision on the target domain (dashed) thanks to cross-modal learning between 2D/3D.

3.1.2 Cross-modal learning

A single scene can be captured by different means and popular datasets are now multimodal although existing works are rarely truly multi-modal. Different sensors do not only capture data of different nature, they capture unique knowledge of the scene, differently impacted by the domain gaps. Here, we aim to answer a simple question: *can we leverage cross-modal data to ease transfer learning?*

xMUDA / xMoSSDA. In Jaritz et al. (2020, 2021), along with Valeo/Valeo.ai collaborators, we investigate how 2D and 3D modalities can learn 3D semantic segmentation from each others and proposed a new *self-supervised* cross-modal learning strategy (see Fig. 3.8a). Here, because 2D (camera) and 3D (Lidar) data are of different nature, in addition of shared knowledge, each modality holds exclusive knowledge – not accessible to the other modality. This makes the domain gaps differ across modalities. For example since it emits its own light pulses, Lidar is more robust to lighting changes (e.g., day/night) than a camera. On the other hand, Lidar data density varies (eg. due to angular deviation and absorbing materials) while cameras always output dense images.

In our work we investigate how cross-modal discrepancies can help preserving the best performance of each sensor – thus avoiding that the limitations of one modality negatively affect the other modality’s performance. We proposed a *cross-modal* objective, implemented as a mutual mimicking game between modalities, that drives toward consistency across predictions

Going further...

Code and adaptation scenarios: <https://github.com/valeoai/xmuda>

Jaritz, M., Vu, T.-H., Charette, R. d., Wirbel, E., and Pérez, P. (2020). xMUDA: Cross-modal unsupervised domain adaptation for 3D semantic segmentation. In *CVPR*

Jaritz, M., Vu, T.-H., de Charette, R., Wirbel, É., and Pérez, P. (2021). Cross-modal learning for domain adaptation in 3D semantic segmentation. *arXiv* submitted to PAMI

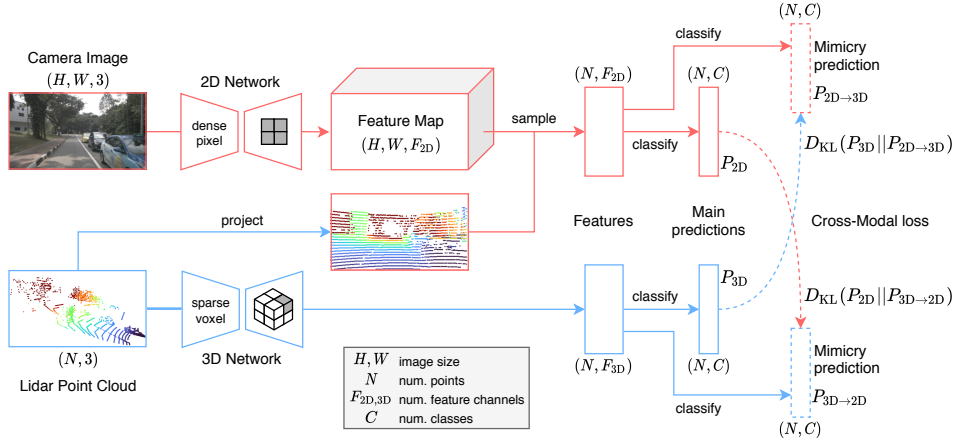


Figure 3.9: **Architecture for cross-modal 3D segmentation.** Independent network streams: **2D** and **3D** take respectively, an image and a point cloud as input, and output tensor of first dimension N , the number of 3D points to predict. The four segmentation outputs consist of the main predictions P_{2D}, P_{3D} and the mimicry predictions $P_{2D \rightarrow 3D}, P_{3D \rightarrow 2D}$. Knowledge transfer across modalities is enforced with KL divergences $D_{KL}(P_{3D} || P_{2D \rightarrow 3D})$, where the objective of the 2D mimicry prediction is to estimate the main 3D prediction, and, vice versa, $D_{KL}(P_{2D} || P_{3D \rightarrow 2D})$.

from different modalities. Specifically, we thoroughly investigated first the cross-modal domain adaptation in the unsupervised setting (Jaritz et al., 2020) (UDA), later extended to semi-supervised (Jaritz et al., 2021) (SSDA), which we coined xMUDA and xMoSSDA, respectively.

Let’s consider an architecture where 2D and 3D are two independent streams, enabling unimodal inference, respectively taking as inputs a 2D image and a 3D point cloud. The goal of cross-modal learning is to let each modality learn from each other. To enable this, a mimicking game can be established between the 2D and 3D output probabilities, *i.e.*, each modality should predict the other modality’s output. The overall objective drives the two modalities toward an agreement, thus enforcing consistency between outputs.

Now, in a naive approach, the cross-modal optimization objective aligns the outputs of both modalities segmentation heads. We found that this leads to an important pitfall: the mimicking objective competes directly with the main segmentation objective. In other words, transfer from the weak modality can degrade the performance of the strong one.

Instead, our contribution focuses on disentangling the mimicry objective

from the segmentation one. Therefore, we propose a dual-head architecture shown in Fig. 3.9. In this setup, the 2D and 3D streams both have two segmentation heads: one *main* head for the best possible prediction, and one *mimicry* head to estimate the other modality’s output.

Considering the 3D segmentation task, the outputs of the four segmentation heads (see Fig. 3.9) are of size (N, C) , with C the number of classes, such that we obtain a vector of class probabilities for each 3D point. The two main heads produce the best possible segmentation predictions, P_{2D} and P_{3D} respectively for each branch. The two mimicry heads estimate the other modality’s output: 2D estimates 3D ($P_{2D \rightarrow 3D}$) and 3D estimates 2D ($P_{3D \rightarrow 2D}$). The benefit of separating heads is that mimicry pushes toward a soft features alignment of F_{2D} and F_{3D} without competing with the main optimization objective. Additionally, mimicking heads are only used for the cross-modal loss and removed for inference.

Cross-modal loss. We inspired from teacher-student distillation (Hinton et al., 2014) to enforce cross-modal alignment. Specifically for our semantic segmentation problem, rather than aligning only the output class, we align the whole distribution with KL divergence. This ensure more information is exchanged, leading to softer labels. The loss is defined as:

$$\mathcal{L}_{xM}(\mathbf{x}) = D_{\text{KL}}(\mathbf{P}_x^{(n,c)} \parallel \mathbf{Q}_x^{(n,c)}) \quad (3.5)$$

$$= -\frac{1}{N} \sum_{n=1}^N \sum_{c=1}^C \mathbf{P}_x^{(n,c)} \log \frac{\mathbf{P}_x^{(n,c)}}{\mathbf{Q}_x^{(n,c)}}, \quad (3.6)$$

with $(\mathbf{P}, \mathbf{Q}) \in \{(\mathbf{P}_{2D}, \mathbf{P}_{3D \rightarrow 2D}), (\mathbf{P}_{3D}, \mathbf{P}_{2D \rightarrow 3D})\}$ where \mathbf{P} is the target distribution from the main prediction which is to be estimated by the mimicking prediction \mathbf{Q} .

Of interest, our cross-modal loss \mathcal{L}_{xM} (Eq. 3.5) is self-supervised and as such can be used in supervised or unsupervised setting. An important property is the complementary of our loss with the pseudo-labels (PL) (Lee et al., 2013) popular for unsupervised adaptation, as they pursue the different objective of strengthening the belief of each modality.

I’m quickly brushing here how we use cross-modal learning for unsupervised and semi-supervised domain adaptation, respectively UDA and SSDA.

Cross-modal UDA (‘xMUDA’), shown in Fig. 3.8b, leverages source \mathbf{x}_s having labels \mathbf{y}_s^{3D} and target \mathbf{x}_t without labels. We train with the combination of a supervised loss (\mathcal{L}_{seg}) on source and our cross-modal loss (\mathcal{L}_{xM})

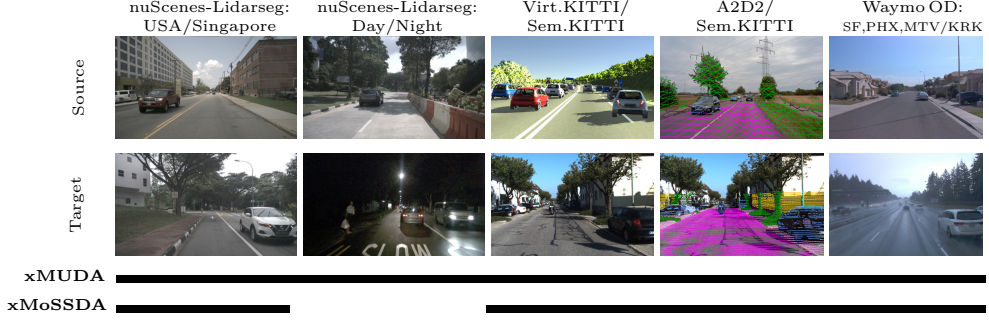


Figure 3.10: **Our five new domain adaptation scenarios.** We leverage metadata of five recent datasets to create adaptation scenarios that reflect various challenges covering country-to-country, day-to-night, synthetic-to-real, change of sensor setup, and clear-to-rainy.

on both source and target ; thus optimizing

$$\min_{\theta} \left[\frac{1}{|\mathcal{S}|} \sum_{\mathbf{x}_s \in \mathcal{S}} \left(\mathcal{L}_{\text{seg}}(\mathbf{x}_s, \mathbf{y}_s^{3D}) + \lambda_s \mathcal{L}_{\text{xM}}(\mathbf{x}_s) \right) + \frac{1}{|\mathcal{T}|} \sum_{\mathbf{x}_t \in \mathcal{T}} \lambda_t \mathcal{L}_{\text{xM}}(\mathbf{x}_t) \right], \quad (3.7)$$

where λ_s, λ_t are hyperparameters to weight \mathcal{L}_{xM} on source and target domain respectively and θ the respective 2D/3D networks weights. When using pseudo labels, computed offline, we denote our proposal xMUDA_{PL} .

Cross-modal SSSA (‘xMoSSDA’) is also addressed in our research as it is of high interest for practical applications where often a small subset of the target dataset is labeled. It considers 3 sets: the source one \mathbf{x}_s with labels \mathbf{y}_s^{3D} , the small target one \mathcal{T}_ℓ with labels $\mathbf{y}_{t\ell}^{3D}$, and the large unlabeled \mathcal{T}_u . Thus we optimize

$$\min_{\theta} \left[\frac{1}{|\mathcal{S}|} \sum_{\mathbf{x}_s \in \mathcal{S}} \left(\mathcal{L}_{\text{seg}}(\mathbf{x}_s, \mathbf{y}_s^{3D}) + \lambda_s \mathcal{L}_{\text{xM}}(\mathbf{x}_s) \right) + \frac{1}{|\mathcal{T}_\ell|} \sum_{\mathbf{x}_{t\ell} \in \mathcal{T}_\ell} \left(\mathcal{L}_{\text{seg}}(\mathbf{x}_{t\ell}, \mathbf{y}_{t\ell}^{3D}) + \lambda_{t\ell} \mathcal{L}_{\text{xM}}(\mathbf{x}_{t\ell}) \right) + \frac{1}{|\mathcal{T}_u|} \sum_{\mathbf{x}_{tu} \in \mathcal{T}_u} \lambda_{tu} \mathcal{L}_{\text{xM}}(\mathbf{x}_{tu}) \right], \quad (3.8)$$

with $\lambda_s, \lambda_{t\ell}$ and λ_{tu} the cross-modal loss weights ; using $\lambda_s = \lambda_{t\ell}$ for simplicity. Again, $\text{xMoSSDA}_{\text{PL}}$ refers to using additional pseudo-labels.

Experiments. We consider 5 newly proposed scenarios to validate our novel ‘cross-modal domain adaptation’ task, leveraging a combination of multi-modal datasets, namely: nuScenes-Lidarseg (Caesar et al., 2020b), Virtual KITTI (Gaidon et al., 2016), SemanticKITTI (Behley et al., 2019), A2D2 (Sun et al., 2020) and Waymo Open Dataset (Sun et al., 2020).

Method	mSc-Lidarseg: USA/Singap.			mSc-Lidarseg: Day/Night			Virt.KITTI/ Sem.KITTI			A2D2/ Sem.KITTI		
	2D	3D	2D+3D	2D	3D	2D+3D	2D	3D	2D+3D	2D	3D	2D+3D
Baseline (src only)	58.4	62.8	68.2	47.8	68.8	63.3	26.8	42.0	42.2	34.2	35.9	40.4
Deep logCORAL (Morerio et al., 2018)	<u>64.4</u>	63.2	69.4	47.7	68.7	63.7	41.4*	36.8	47.0	35.1*	41.0	42.2
MinEnt (Vu et al., 2019)	57.6	61.5	66.0	47.1	68.8	63.6	39.2	43.3	47.1	37.8	39.6	42.6
PL (Li et al., 2019b)	62.0	<u>64.8</u>	<u>70.4</u>	47.0	69.6	63.0	21.5	44.3	35.6	34.7	41.7	<u>45.2</u>
FDA (Yang and Soatto, 2020)	60.8	-	-	48.4	-	-	32.8*	-	-	37.6*	-	-
xMUDA	<u>64.4</u>	63.2	69.4	<u>55.5</u>	<u>69.2</u>	67.4	<u>42.1</u>	<u>46.7</u>	<u>48.2</u>	<u>38.3</u>	<u>46.0</u>	44.0
xMUDAPL	67.0	65.4	71.2	57.6	69.6	<u>64.4</u>	45.8	51.4	52.0	41.2	49.8	47.5
Oracle (trg only)	75.4	76.0	79.6	61.5	69.8	69.2	66.3	78.4	80.1	59.3	71.9	73.6
Domain gap (Oracle – Baseline)	17.0	13.3	11.5	13.6	1.1	5.9	39.5	36.4	37.9	25.1	36.0	33.2

Table 3.1: **xMUDA on 3D semantic segmentation**. Comparing against the best existing DA techniques, we demonstrate the benefit of cross-modal learning for UDA. We report mIoU (with **best** and 2nd best) on target test set for unimodal 2D *or* 3D prediction and ensembling 2D+3D (means of 2D and 3D probabilities). ‘Baseline’ is trained on the source set \mathcal{S} only, and ‘Oracle’ is the upper bound, trained supervisedly on the target set \mathcal{T} using labels that are otherwise not used for baselines and xMUDA/xMUDAPL.

Fig. 3.10 exposes a sample source/target for each of our 5 tasks, obtained by leveraging image metadata. Most importantly, each task encompasses a distinct adaptation challenge, from left to right: country to country (note USA drives right, Singapore drives left), day to night, synthetic to real, different sensor setup (note the change of lidar pattern), clear to rainy. On implementation, we used a modified UNet with ResNet34 (He et al., 2016) as 2D network, and SparseConvNet (Graham et al., 2018) as 3D network.

The benefit of our cross-modal learning for both UDA or SSDA is evident in Tabs. 3.1 and 3.2, where the semantics mIoU on the target test set is compared to the four best adaptation baselines at the time. Again, since inference is unimodal we report individual 2D or 3D, and the ensembling performance (‘2D+3D’) obtained by taking the mean of the predicted 2D and 3D probabilities after softmax. In all cases but one, xMUDA (Tab. 3.1) alone outperforms all compared baselines, demonstrating the beneficial exchange of information between 2D and 3D. The 2D/3D ‘Oracles’ (trained supervisedly on *target*) indicate that overall LiDAR (3D) is always the strongest modality, which resonates with the choice of 3D segmentation task. However, xMUDA consistently improves *both* modalities (2D and 3D) *i.e.*, even the strong modality can learn from the weaker one. Interestingly for semi-supervised adaptation, xMoSSDA *alone* rarely outperforms pseudo-labels (‘PL’) which we attribute to the fact that pseudo-labels quality is significantly increased by the use of \mathcal{T}_ℓ . When pseudo-labels are combined with cross-modal learning (i.e. xMUDAPL, xMoSSDAPL), it outperforms on all 12 unimodal 2D or 3D metrics, and on 5 out of 6 ensembling 2D+3D experiments. A few qualitative results are also in Fig. 3.11.

Method	Train set	nuSc-Lidarseg: USA/Singap.			A2D2/ Sem.KITTI			Waymo OD: SF,PHX,MTV/KRK		
		2D	3D	2D+3D	2D	3D	2D+3D	2D	3D	2D+3D
Baseline (src only)	\mathcal{S}	58.8	63.2	68.5	37.9	32.8	43.3	61.4	50.8	64.4
Baseline (lab. trg only)	\mathcal{T}_ℓ	70.5	74.1	74.2	51.3	57.7	59.2	56.5	57.1	60.3
Baseline (src and lab. trg)	$\mathcal{S} + \mathcal{T}_\ell$	72.3	73.1	78.1	54.8	62.4	66.2	64.5	56.3	69.3
Domain gap (\mathcal{S} vs. $\mathcal{S} + \mathcal{T}_\ell$)		13.5	9.9	9.6	16.9	29.5	22.9	3.2	5.5	4.9
xMUDA	$\mathcal{S} + \mathcal{T}_u$	63.1	64.2	67.8	38.6	44.5	44.4	61.8	54.0	66.7
xMUDA _{PL}	$\mathcal{S} + \mathcal{T}_u$	66.2	65.1	70.1	41.4	49.5	48.6	<u>68.3</u>	55.2	<u>71.9</u>
Deep logCORAL (Moreiro et al., 2018)	$\mathcal{S} + \mathcal{T}_\ell + \mathcal{T}_u$	71.7	73.1	78.2	55.1*	62.2	64.7*	61.4	56.5	66.1
MinEnt (Vu et al., 2019)	$\mathcal{S} + \mathcal{T}_\ell + \mathcal{T}_u$	72.6	73.3	76.6	56.3	62.5	65.0	64.3	56.6	69.1
PL (Li et al., 2019b)	$\mathcal{S} + \mathcal{T}_\ell + \mathcal{T}_u$	73.6	<u>74.4</u>	79.3	<u>57.2</u>	<u>66.9</u>	<u>68.5</u>	67.4	56.7	70.2
xMoSSDA	$\mathcal{S} + \mathcal{T}_\ell + \mathcal{T}_u$	<u>74.3</u>	74.1	78.5	56.5	63.4	65.9	65.2	<u>57.4</u>	69.4
xMoSSDA _{PL}	$\mathcal{S} + \mathcal{T}_\ell + \mathcal{T}_u$	75.5	74.8	<u>78.8</u>	59.1	68.2	70.7	70.1	58.5	73.1
Unsupervised advantage (relative)		3.1 (+4.3%)	1.7 (+2.3%)	0.7 (+0.9%)	4.3 (+7.8%)	5.8 (+9.3%)	4.5 (+6.8%)	5.6 (+8.7%)	2.2 (+3.9%)	3.8 (+5.5%)

* The 2D network is trained with batch size 6 instead of 8 to fit into GPU memory.

Table 3.2: **xMoSSDA on 3D semantic segmentation**. See caption of Tab. 3.1 for notation details. Considering semi-supervised adaptation (SSDA), we have a source dataset \mathcal{S} like in UDA, while, unlike UDA, the target dataset has a small labeled part \mathcal{T}_ℓ and a large unlabeled part \mathcal{T}_u . The three uni-modal SSDA baselines as well as our ‘xMoSSDA’ and ‘xMoSSDA_{PL}’ are trained supervisedly on $\mathcal{S} + \mathcal{T}_\ell$ and unsupervisedly on \mathcal{T}_u . We report the ‘Unsupervised advantage’, i.e. the difference between xMoSSDA_{PL} and ‘Baseline (src and lab. trg)’ and relative improvement.

Our new adaptation scenarios also allow insights on the difficulty of domain adaptation. For example, studying the difference between the Oracle (ie. trained supervisedly on source *and* target) and the Baseline (ie. trained on source only), which we refer to as ‘Domain gap’ in Tab. 3.1, shows that intra-dataset domain gaps (nuScenes-Lidarseg: USA/Singapore, Day/Night), in [1.1, 17.0], are much smaller than the inter-dataset domain gaps (A2D2/Sem.KITTI, Virt.KITTI/Sem.KITTI), in [25.1, 39.5]. This suggests that a change in sensor setup (A2D2/Sem.KITTI) is actually a very hard domain adaptation problem as for synthetic-to-real (Virt.KITTI/Sem.KITTI).

In above results, despite a cross-modal training, we considered unimodal inference or naive fusion via ensembling. However, in this research project we also wanted to address the question: can cross-modal learning help *fusion* setup where both modalities make a joint prediction? Because vanilla fusion (Fig. 3.12a left) implies a single head for the two modalities, we cannot replicate the cross-modal architecture as is. Instead, for ‘xMUDA Fusion’ an additional segmentation head is added to both 2D and 3D network streams *prior* to the fusion layer with the purpose of mimicking the central fusion head (Fig. 3.12a, right). Again, the performance in Fig. 3.12b shows we outperform baselines using vanilla fusion. This interestingly suggests

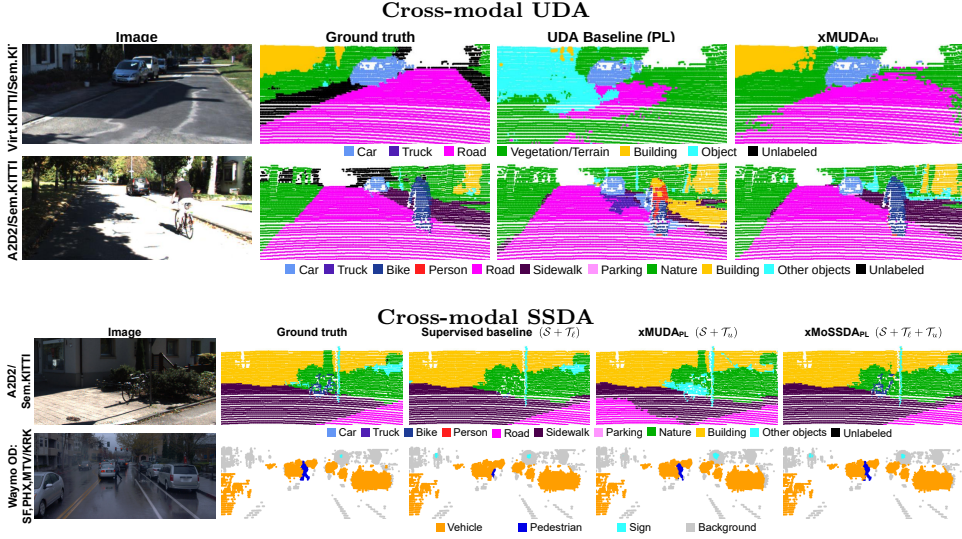
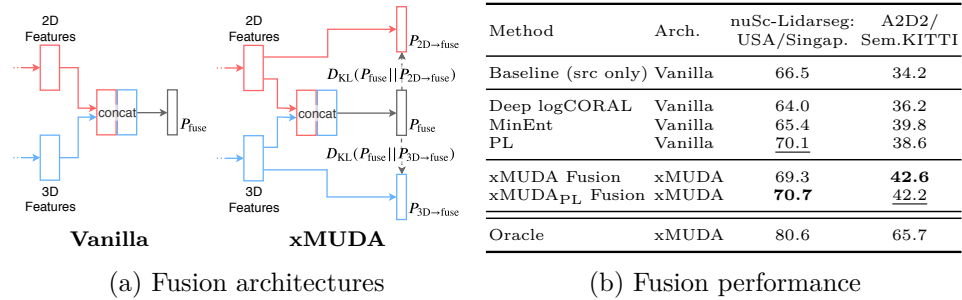


Figure 3.11: **Qualitative results for xMUDA and xMoSSDA.** Selective outputs on 4 scenarios of our cross-modal UDA and SSDA demonstrate the benefit of cross-modal learning which consistently improves segmentation.



(a) Fusion architectures

(b) Fusion performance

Figure 3.12: **Cross-modal fusion.** (a) Comparison of Vanilla fusion and xMUDA Fusion. Different from the former, we add a mimicry segmentation head to each modality, which goal is to predict the fusion output. (b) Reporting mIoU on 2 adaptation scenarios, we show that xMUDA Fusion, outperforms Vanilla Fusion.

that even 2D/3D fusion scheme can benefit from more direct cross-modal exchange.

In [Jaritz et al. \(2021\)](#) we ablate all our contributions, demonstrate the stability of our dual-head architecture, and provide further evaluation and analyses. In all setups, even when only considering a fully supervised training (ie. no adaptation), 2D-3D cross-modal learning was shown to boost performance.

On a wider note, this opens doors to future works on cross-modal training. Especially, while 2D/3D are of relatively close nature (both contain visual, geometry, etc.) it would be interesting to investigate cross-modal learning for dramatically different modalities such as Audio/Image, Text/Image, etc. Of note, [Peng et al. \(2021\)](#) nicely extended our work with a new inter-domain cross-modal learning and deformable inter-modal matching.

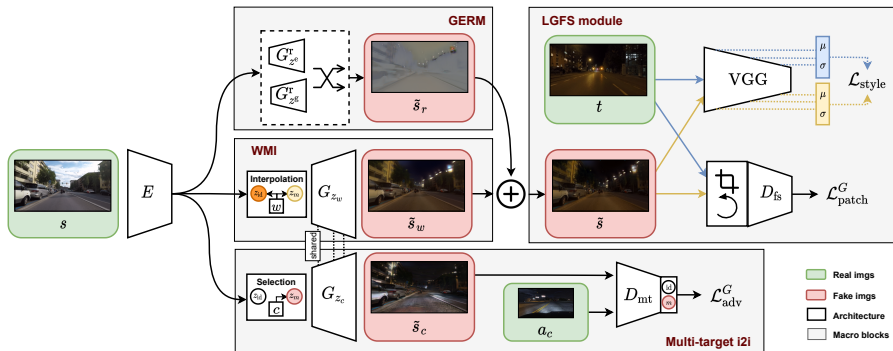


Figure 3.13: **Few-shot translation architecture.** Our approach learns a multi-target i2i manifold \mathbb{A} from a set of anchors, from which it departs in two-fold. First, using our weighted manifold interpolation (WMI) along \mathbb{A} , second by estimating residuals to resemble the target few-shots \mathcal{T} (‘GERM’). We train by exploiting statistics alignment with a VGG network and a patch-based adversarial learning in our Local-Global Few-Shot module (‘LGFS’).

3.2 Dealing with fewer data

The only work in this section (Pizzati et al., 2021c) originates from PhD student Fabio Pizzati and also includes active collaborator from Uni. Laval.

Close to our prior works Pizzati et al. (2020b); Dell’Eva et al. (2021), we address here the ability to learn image-to-image (i2i) translation $\mathcal{S} \mapsto \mathcal{T}$ though considering here \mathcal{T} having very few training samples (e.g. $|\mathcal{T}| \leq 25$).

ManiFest (few-shot i2i). In Pizzati et al. (2021c) (*submitted*) we propose a few-shot i2i framework which is shown to be resistant to highly unstructured transformations as adverse weather generation or night rendering. Our work departs from the observation that features consistency is crucial for i2i (Ma et al., 2019), and impractical with only few-shot samples. Therefore, rather than directly addressing the i2i few-shot, we deform a stable learned manifold towards our few-shot target \mathcal{T} , benefiting also from style transfer and patch-based training to enable few-shot learning. Importantly, our framework can approximate the *general* style of the entire few-shot set or reproduce any image in an *exemplar* manner.

Fig. 3.13 presents an overview of our approach. First, we learn a style manifold in a standard multi-target GAN fashion from a set of so-called *anchor* domains \mathbb{A} having large amounts of training data. Our Weighted Manifold Interpolation (WMI) then finds the optimal point along the style manifold to inject the few-shot domain appearance, which is learned with the Local-Global Few-Shot loss (LGFS). We allow to further depart

Going further...

Code:
<https://github.com/cv-rits/ManiFest>

Pizzati, F., Lalonde, J.-F., and de Charette, R. (2021c). ManiFest: Manifold deformation for few-shot image translation. *arXiv (submitted)*

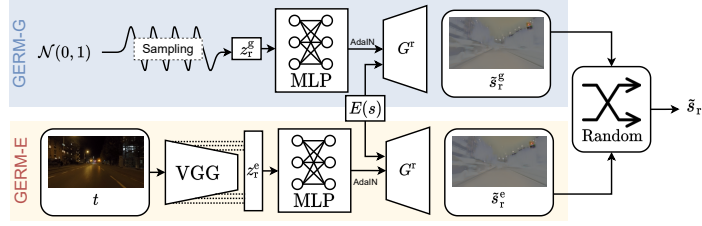


Figure 3.14: **GERM-based residuals.** We perform either *exemplar-* or *general-*based transformations on the few-shot set by learning residuals conditioned on extracted statistics or noise, respectively.

from the interpolated manifold with a General-Exemplar Residual Module (GERM) which learns a residual refining the overall image appearance. We briefly describe our contributions below.

Hereafter, *real* images are $s \in \mathcal{S}, t \in \mathcal{T}$, and *fake* ones $\tilde{s} \in \mathcal{T}$ where \tilde{s} is the output produced by a model.

Multi-target i2i. Rather than learning $\mathcal{S} \mapsto \mathcal{T}$ directly, we learn the $\mathcal{S} \mapsto \mathbb{A}$ mapping, assuming the availability of a set $\mathbb{A} = \{\mathcal{A}_{\text{id}}, \mathcal{A}_m\}$ of *anchor* domains for which we have lots of data available (equivalent to the “base” categories in few-shot image classification). By construction, one anchor is always the identity domain ($\mathcal{A}_{\text{id}} = \mathcal{S}$), while the other (\mathcal{A}_m) contain images easier to collect with respect to \mathcal{T} .

Weighted manifold interpolation (WMI). Our intuition is that encoding \mathcal{T} by linearly interpolating style representations in the manifold spanned by \mathbb{A} should enforce feature consistency in \mathcal{T} . For instance, assuming $\mathcal{S} = \text{day}$, $\mathcal{T} = \text{night}$, $\mathcal{A}_m = \text{synthetic night}$, the network will be provided with the information that all sky pixels should be darkened together.

Departing from the idea that distance between \mathbb{A} and \mathcal{T} manifolds varies greatly, we seek the optimal interpolation point along \mathbb{A} manifold where this distance is minimized. In practice, we learn the weights w_i which encode an image \tilde{s}_w by interpolating the anchor domains style representations:

$$z_w = \sum_{i \in \{\text{id}, m\}} w_i z_i, \quad \tilde{s}_w = G_{z_w}(E(s)). \quad (3.9)$$

General-Exemplar Residual Module (GERM). Because \mathcal{T} is likely not to be on \mathbb{A} , it is important to allow departing from the latter. This is done by learning a residual image \tilde{s}_r , which helps encode characteristics from \mathcal{T} that are absent from \mathbb{A} . In practice, we process the input image features $E(s)$ with a generator G^r such that

$$\tilde{s}_r = G^r(E(s)), \quad \text{and} \quad \tilde{s} = \tilde{s}_w + \tilde{s}_r. \quad (3.10)$$

We provide two ways of generating the residual image, i.e., $\tilde{s}_r \in \{\tilde{s}_r^e, \tilde{s}_r^g\}$. In both cases, we draw inspiration from AdaIN style injection (Huang et al., 2018a) and randomly condition the injected parameters either on learned feature statistics (*exemplar*), or on random gaussian noise (*general*).

To reproduce the style of a specific image $t \in \mathcal{T}$ as in Ma et al. (2019), we provide an *exemplar* residual (“GERM-E” in Fig. 3.14) by conditioning on t . In this case,

$$z_r^e = (\mu_k(t), \sigma_k(t)) \parallel_k, \quad \tilde{s}_r^e = G_{z_r^e}^r(E(s)), \quad (3.11)$$

where $\mu_k(x) = \mu(\phi_k(x))$ and $\sigma_k(x) = \sigma(\phi_k(x))$ are the mean and variance of the k -th layer outputs ϕ_k of a pretrained VGG network (Huang and Belongie, 2017), respectively, and \parallel is the concatenation operator.

We identify as *general* residual some \tilde{s}_r^g which moves the generated image towards \mathcal{T} by mimicking an average style learned from all images in \mathcal{T} . This is illustrated as “GERM-G” in Fig. 3.14, and, mathematically:

$$z_r^g \sim \mathcal{N}(0, 1), \quad \tilde{s}_r^g = G_{z_r^g}^r(E(s)). \quad (3.12)$$

Local-global few-shot loss (LGFS). Finally, the quality of the resulting image \tilde{s} is compared against the few-shot training set \mathcal{T} with a combination of two loss functions. First, we take inspiration from the state-of-the-art of image style transfer where one image is enough for transferring the *global* appearance of the style scene (Huang and Belongie, 2017). Our intuition is that feature statistics alignment, vastly used in style transfer, could be less prone to overfitting with respect to adversarial training. So, we align features between \tilde{s} and a target image $t \in \mathcal{T}$ using a style loss as in Huang and Belongie (2017) on N layers of a pretrained VGG network. This writes

$$\mathcal{L}_{\text{style}} = \sum_{k=0}^N \|\mu_k(\tilde{s}) - \mu_k(t)\|_2 + \|\sigma_k(\tilde{s}) - \sigma_k(t)\|_2, \quad (3.13)$$

where (μ_k, σ_k) are the same as for GERM. While this is effective in modifying the general image appearance, aligning statistics alone is insufficient to produce realistic outputs. Thus, to provide *local* guidance, i.e. on more fine-grained characteristics, we employ an additional discriminator, trained to distinguish between rotated patches sampled from \tilde{s} and t . We refer to Pizzati et al. (2021c) for details.

Experiments. Our i2i is trained end-to-end with MUNIT backbone, alternatively optimizing GERM-G or GERM-E. In our work we thoroughly evaluate our framework on 3 translation tasks, leveraging 4 popular datasets – the real ACDC (Sakaridis et al., 2021), Dark Zurich (Sakaridis et al., 2020), Cityscapes (Cordts et al., 2016), and the synthetic VIPER (Richter et al., 2017). We report here only a small subset of our evaluation.

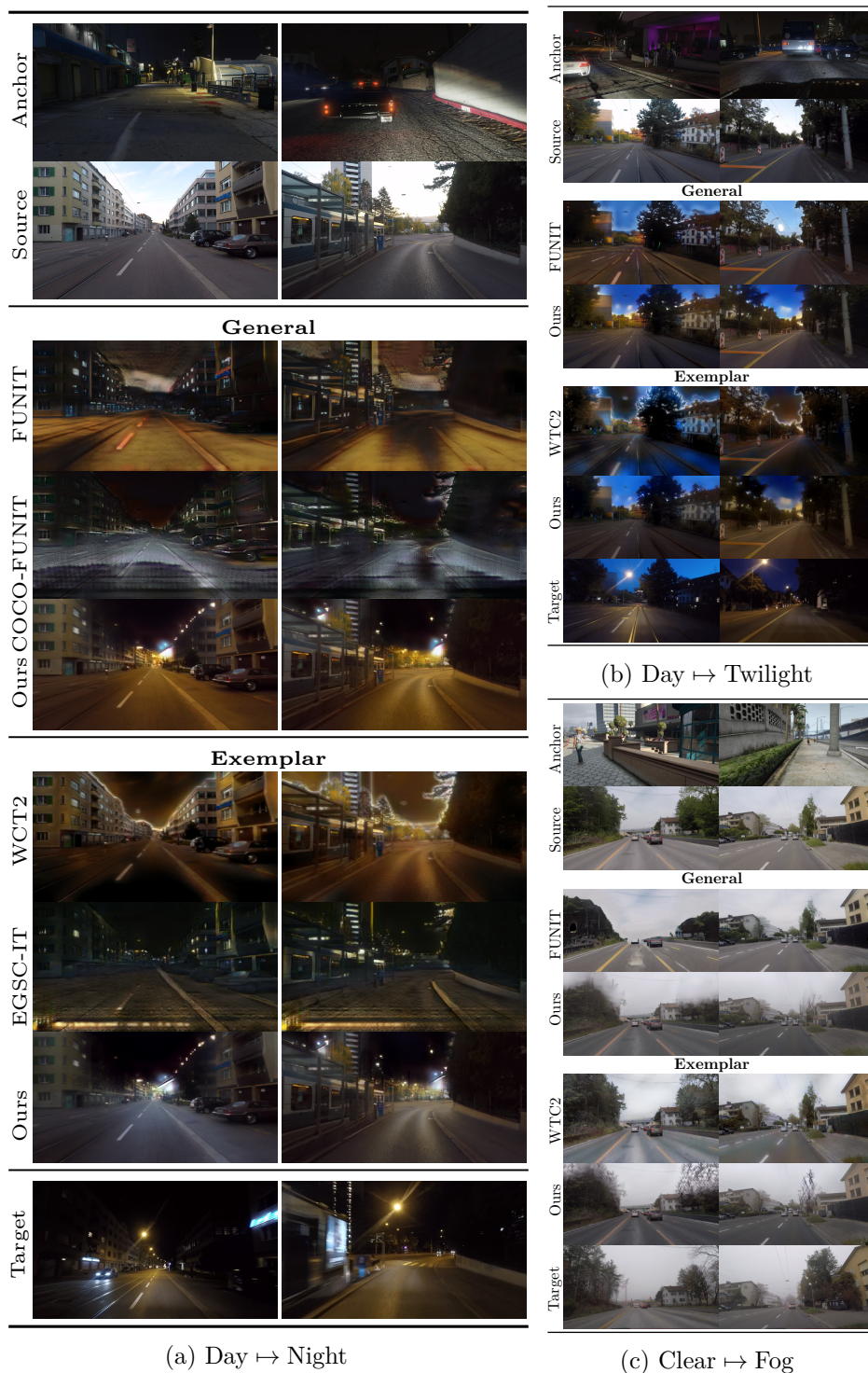


Figure 3.15: **Qualitative evaluation.** We evaluate on (a) Day \mapsto Night, (b) Day \mapsto Twilight, and (c) Clear \mapsto Fog tasks (with $N = 25$ few-shot images). In all, our approach learns a *general* realistic representation of the few-shot target, and correctly reproduces the style of paired *exemplar* target images. In comparison, existing baselines either suffer from entanglement with the anchor domain (e.g. FUNIT, COCO-FUNIT, EGSC-IT) or from unrealistic artifacts (e.g. WCT2).

	Method	Anchor	Target	FID \downarrow	LPIPS \downarrow
G	MUNIT	0	400	79.20	0.529
	MUNIT	3090	0	132.72	0.613
	MUNIT	0	25	<u>91.61</u>	<u>0.553</u>
	FUNIT	3090	25	156.97	0.573
	COCO-FUNIT	3090	25	201.67	0.644
	Ours	3090	25	81.01	0.535
E	MUNIT	0	400	87.71	0.522
	MUNIT	3090	0	142.04	0.559
	MUNIT	0	25	128.73	<u>0.562</u>
	EGSC-IT	3090	25	106.68	0.574
	WCT2	-	-	<u>105.58</u>	0.580
	Ours	3090	25	80.57	0.525

(a) GAN metrics

Model	mIoU \uparrow	Acc. \uparrow
Baseline (src only)	12.93	45.15
MUNIT-single	17.21	54.67
MUNIT	21.22	56.65
Ours - G	<u>21.62</u>	<u>58.06</u>
Ours - E	24.31	60.50

(b) Semantic seg. ($N = 25$)

Images (N)	Ours logFID	MUNIT logFID	Ours LPIPS	MUNIT LPIPS
25	4.50	4.50	0.55	0.55
15	4.50	4.50	0.55	0.55
5	4.50	4.50	0.55	0.55

(c) Robustness to N (Ours-G)

Figure 3.16: **Day \mapsto Night evaluation.** Our *general* (G) and *exemplar* (E) translations outperform all baselines (a). Translating Cityscapes to night with only $N = 25$ shows an improved performance on the proxy semantic segmentation task (b). Finally, (c) shows robustness to varying N .

We train our framework on 3 main tasks having Source/Target: Day \mapsto Night with $\text{ACDC}_{\text{day}}/\text{ACDC}_{\text{night}}$, Clear \mapsto Fog with $\text{ACDC}_{\text{day}}/\text{ACDC}_{\text{fog}}$, and Day \mapsto Twilight with $\text{DZ}_{\text{day}}/\text{DZ}_{\text{twilight}}$. As anchor domains we choose the synthetic $\text{VIPER}_{\text{night}}$ for Day \mapsto Night and Day \mapsto Twilight, and $\text{VIPER}_{\text{day}}$ for Clear \mapsto Fog.

Qualitative results are in Fig. 3.15 for $N = 25$ images in \mathcal{T} . We compare against FUNIT (Liu et al., 2019a) and COCO-FUNIT (Saito et al., 2020) for *general* translations (GERM-G), and against WCT² (Yoo et al., 2019) and EGSC-IT (Ma et al., 2019) for *exemplar* translations (GERM-E) (baseline details in our publication). In Day \mapsto Night (Fig. 3.15a), even if the appearance of images in \mathcal{T} is partially transferred on translated images (e.g. road color, darker sky), FUNIT and COCO-FUNIT still strongly focus on typical textures of the anchor domains in \mathbb{A} (note, for example, how the street texture is similar to the anchor) which negatively impacts the overall image realism. The same can be observed with EGSC-IT, where the hood of the ego-vehicle present in anchor images (first column) is retained and significantly impact visual results. While WCT² exhibits sharp results, it does not correctly map the image context, and it is limited to appearance alignment which leads to artifacts (e.g. yellow sky with white halos).

The quantitative evaluation in Fig. 3.16a is aligned with the qualitative results since our approach outperforms all few-shot baselines. On *exemplar*

	\mathcal{S}	\mathcal{T}	\mathcal{A}_m	FID $_{\downarrow}$	LPIPS $_{\downarrow}$
Intra-dataset	ACDC-Day	ACDC-Night	Day	85.73	0.553
			Night	81.01	0.535
			Rain	<u>81.38</u>	0.549
			Snow	86.74	0.554
			Sunset	83.83	0.571
			All	83.71	<u>0.547</u>
	DZ-Day	DZ-Twilight	Day	64.19	0.505
			Night	<u>63.15</u>	0.510
			Rain	65.33	<u>0.501</u>
			Snow	64.09	0.513
			Sunset	63.78	0.504
			All	60.98	0.469
Cross-dataset	ACDC-Day	DZ-Twilight	Day	89.61	*
			Night	90.48	*
			Rain	<u>89.47</u>	*
			Snow	91.49	*
			Sunset	91.77	*
			All	85.15	*

* Paired images are unavailable for LPIPS evaluation.

Table 3.3: **Impact of anchor domains \mathbb{A} .** The relatively stable performance across all tested anchors demonstrates robustness of our method. Using all available anchors (“All”) also generally improve performance.

translations, it performs even better than our backbone trained on the entire set of 400 training images. This shows that GERM-E, as opposed to AdaIN style injection, enables better disentanglement of the exemplar style by leveraging a realistic starting point interpolated in the manifold. Aside from realism, we demonstrate the usability of these translations for a proxy segmentation task in Fig. 3.16b, by retraining HRNet (Wang et al., 2019a) with our Day \mapsto Night versions of the Cityscapes dataset, and those of our backbone, in single (MUNIT-single) or multi- (MUNIT) modal settings. In Fig. 3.16c we vary the number of samples (N) for our *general* translations (plain lines) showing we consistently outperform the backbone (dashed).

Finally, we wish to report here another interesting finding. One may think that the anchor domains in $\mathbb{A} = \{\mathcal{A}_{id}, \mathcal{A}_m\}$ should resemble target \mathcal{T} . To size the influence of anchors, in Tab. 3.3 we ablate anchors \mathcal{A}_m by selecting different conditions from the VIPER dataset, namely {Day, Night, Rain, Snow, Sunset}. Our results instead show that the proxy multi-target i2i task implicitly encodes consistency in the transformation, and is therefore robust to the choice of anchors. Considering a multi-anchor setup (“All”, Tab. 3.3) generally improves results, ranking either first or second in all cases for at least one metric, which we ascribe to the more informative \mathbb{A} manifold to depart from.

3.3 Supervision from action

We now divert a little from other researches in this manuscript. If we consider the context of robotics, the purpose of vision algorithms is to

extract a humanly interpretable scene understanding – may it be on scene semantics, objects locations, geometry, or else – in order to safely plan the actions of a robot. Still, even partial supervision of these proxy tasks is costly. Hence, we ask ourselves a simple question: *Can we learn to control a robot without any supervised scene understanding ?*

In this section, we investigate the ability to learn directly a robot policy $\pi(\cdot)$ from a given environment state s . This is referred as end-to-end learning since the goal is to map the sensory space to the action space without needs of intermediary supervised representations. Unlike vision tasks, supervising robot actions is light and cheap.

We focus here on end-to-end driving. Despite early interest for the latter (Pomerleau, 1989), at the time of our first work (2017) few had addressed end-to-end driving (Bojarski et al., 2016; Xu et al., 2016; Rausch et al., 2017; Mnih et al., 2016) most of which used imitation learning (aka behavioral cloning). In the latter, $\pi(\cdot)$ mimics an expert driver – ultimately leading to distributions mismatch since the expert rarely encounters driving failures (eg. off road driving, lane shift, wrong way driving, etc.).

Instead, we rely on reinforcement learning (RL) where $\pi(\cdot)$ is learned from rewards which are either dense (after each iteration) or sparse (after *some* iterations). Because RL requires trial-and-error, which would be dramatically dangerous for cars, end-to-end driving almost always train in simulators. There are roughly three RL strategies: value-based, policy-based, or model-based. *Value-based* RL chooses the optimal action $a^*(s)$ from the approximated action-value function \mathcal{Q} – *implicitly* learning an optimal policy. Instead, *policy-based* RL explicitly learns the policy $\pi(\cdot)$ mapping $s \mapsto a$. A standard hybrid strategy is *actor-critic* which learns both policy (actor) and value (critic) ; further stabilizing the training. Finally, *Model-based* RL considers a priori known deterministic model of the environment and is commonly employed for games (chess, pong, go, etc.), but generally disregarded for complex simulation like car driving. We refer to the many surveys about RL (Kaelbling et al., 1996), Deep RL (DRL) (Arulkumaran et al., 2017) or DRL for driving (Kiran et al., 2021).

In our work, we investigated actor-critic DRL in simulated environment. In Perot et al. (2017); Jaritz et al. (2018a), with Valeo collaborators, we were among the firsts to address end-to-end driving in a realistic simulator ; considering dense reward (ie. frame-wise). In the recent Agarwal et al. (2021), with Inria collaborators, we investigated the challenging task of learning to drive with only sparse rewards (ie. episode-wise). These works have strong technical and exploratory components though modest scientific contribution.

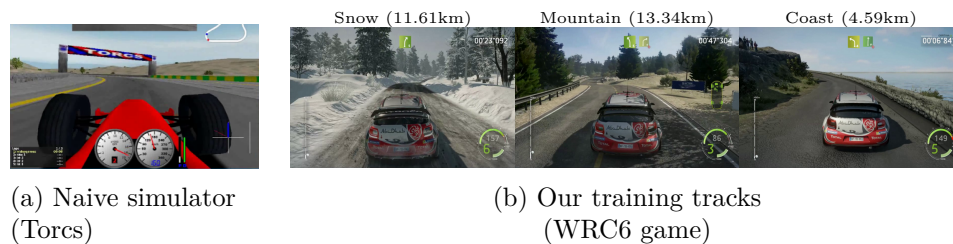


Figure 3.17: **Simulators.** Instead of the often used TORCS (a), we leveraged WRC6 a graphically and physically realistic car racing game (b).

3.3.1 DRL with dense reward

End-to-end race driving. In Perot et al. (2017); Jaritz et al. (2018a) we addressed end-to-end driving with DRL in a car racing game, therefore seeking to drive as fast as possible on challenging tracks while disregarding safety or style consideration. Putting things in context, few works had addressed DRL for driving prior to ours, mainly learning only partial control in non-realistic arcade simulators. Instead, we learn the full car control (steering, gas, brake, hand brake) from the interaction with WRC6 (Kylo-tonn, 2016) – a realistic rally game with stochastic behavior (animations, lights) – and bring practical contributions that boost performance.

Compared to the commonly employed TORCS (Wymann et al., 2000) (Fig. 3.17a), our WRC6 simulator (Fig. 3.17b) exhibits variable environments (snow, mountain, coast) and extreme physics changes (road adherence, tire friction, etc.). As such the agent must understand the scene layout and physics to infer the correct control. As DRL framework, we leveraged the Asynchronous RL strategy (A3C) from Mnih et al. (2016) as it is well suited for experience decorrelation. For fair comparison with drivers’ knowledge, our agent receives as inputs the current front image view³ and driving speed.

A3C framework. In our DRL setup (Fig. 3.18a), at discrete time steps t the RL agent receives the game state (s_t) on which basis it selects an action a_t as a function of policy π with probability $\pi(a_t|s_t)$ and sends it to the environment where a_t is executed and the next state s_{t+1} is reached with associated reward r_t . In general, the RL agent seeks to maximize the discounted reward $R_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k}$ with $\gamma \in [0, 1[$, however in A3C the discounted reward R_t is in fact estimated with a value function $V^{\pi_\theta}(s) = \mathbb{E}[R_t|s_t = s]$. The remaining rewards can be estimated after some steps as the sum of the above value function and the actual rewards: $\hat{R}_t = \sum_{i=0}^{k-1} \gamma^i r_{t+i} + \gamma^k \hat{V}^{\pi_\theta}(s_{t+k})$ where k varies between 0 and $t_{max} = 5$

³In-game displays visible in Fig. 3.17b are obviously deactivated.

Going further...

Videos: <https://team.inria.fr/rits/computer-vision/drl/>

Perot, E., Jaritz, M., Toromanoff, M., and de Charette, R. (2017). End-to-end driving in a realistic racing game with deep reinforcement learning. In *CVPR Workshops*

Jaritz, M., de Charette, R., Toromanoff, M., Perot, E., and Nashashibi, F. (2018a). End-to-end race driving with deep reinforcement learning. In *ICRA*

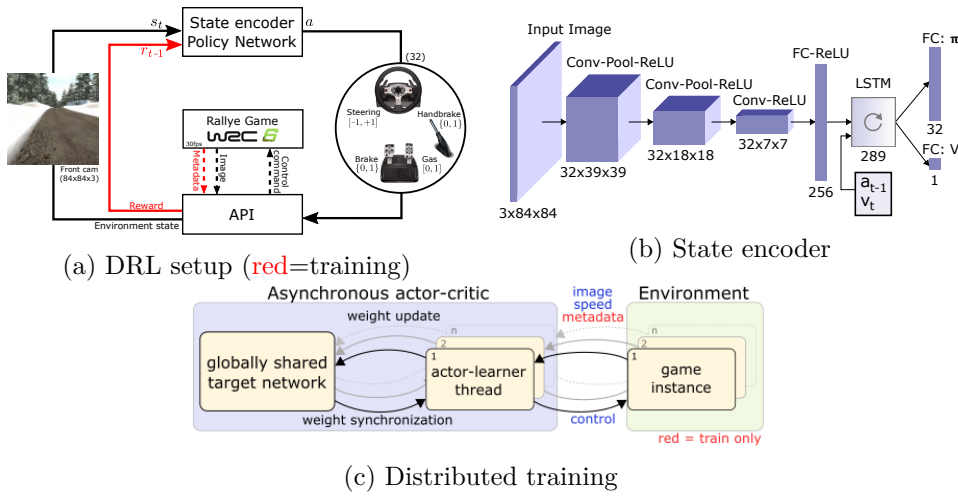


Figure 3.18: **End-to-end driving with DRL.** (a) A dedicated API enables our RL agents to interact with the WRC6 rally racing game (cf. Fig. 3.17b), receiving the game metadata (image and speed) – inputted in our state encoder (b) to predict the next action. In total, 15 agents are trained simultaneously in a distributed asynchronous setting (c).

the sampling update. The quantity $\hat{R}_t - \hat{V}^{\pi_\theta}(s_t)$ can be interpreted as the advantage, i.e. whether the actions $a_t, a_{t+1}, \dots, a_{t+t_{max}}$ were actually better or worse than expected; which allows correction when non optimal strategies are encountered. In practice, the networks for policy $\pi(a_t|s_t; \theta)$ (aka Actor) and value function $\hat{V}(s_t; \theta')$ (aka Critic) share all layers but the last fully connected. The output probabilities of the discrete control commands (i.e. steering, gas, brake, hand brake) are determined by the policy π_θ (θ the network weight) which we optimize with the REINFORCE method (Williams, 1992) which computes an unbiased estimate of $\nabla_{\theta} \mathbb{E}[R_t]$.

We proposed several practical contributions, summarized here:

- Compared to the A3C (Mnih et al., 2016), our state encoder (Fig. 3.18b) is deeper and uses speed v_t along previous action a_{t-1} in an LSTM network.
- We study different reward shaping and demonstrate the benefit of our reward which accounts for lateral drifts.
- We use an imbalance proportion of actions with more gas command and dissociate break from hand-brake, which experimentally performs better and allow the agent to learn drifting.
- To maximize variance of the environment, and benefit from the decorrelation property of A3C, we simultaneously train on 3 tracks

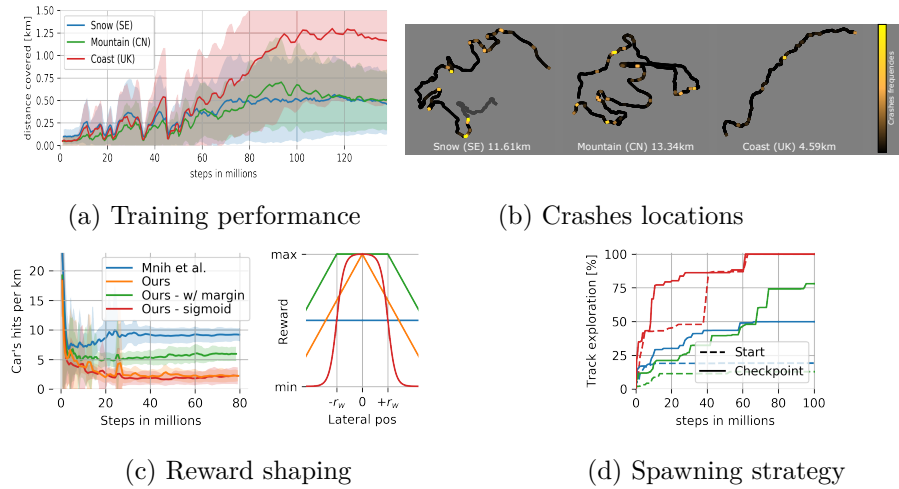


Figure 3.19: **Performance on training tracks.** (a) Our agents successfully learned to drive despite the challenging tracks appearance and physics, taking sharp curves and hairpins (b). Our reward shaping (c) and spawning strategy (c) lead to less car hits and better track exploration, respectively.

(Fig. 3.17b) with various visuals and physics, and spawn the agent randomly along the tracks.

Experiments. Our DRL is trained simultaneously on 3 three tracks - a total of 29.6km - using multiple distributed RL clients (Fig. 3.18c) to benefit from the asynchronous A3C capabilities. Training the agents took about a week on 3 computers with 9 agents total – each one interacting with a separate instance of the WRC6 game through a custom designed API, see Fig. 3.18c. This required an intense and cumbersome engineering work.

On training tracks, Fig. 3.19a, the agents successfully learn to drive despite the challenging graphics and physics at an average speed of 72.88km/h and cover an average distance of 0.72km per run. A run is interrupted if the bot either stops progressing or goes in the wrong direction (off road, wrong ways). We refer to the latter as “crashes” and their locations are visible in Fig. 3.19b, demonstrating that *snow* and *mountain* are harder tracks but also that bots can go through slopes, sharp curves and even some hairpin bends. While our contributions in Jaritz et al. (2018a) are mostly technical, our reward shaping and spawning strategies were shown to improve significantly the driving either by reducing the hits per kilometer (2.3 hits/km vs 9.2 for Mnih et al. (2016), Fig. 3.19c), or by favoring exploration (Fig. 3.19d).

In Jaritz et al. (2018a) we also ask ourself: *Can our training generalize*

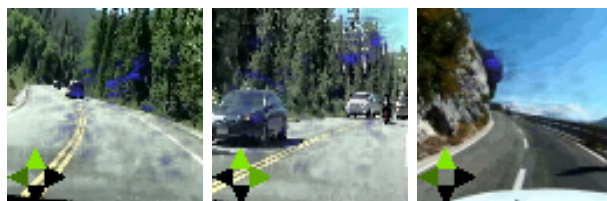


Figure 3.20: **Sample predictions on real web footage.** Despite scenarios being unseen during training (other road users, multi-lanes, etc.), our RL agent seems to take ad-hoc control decisions (shown as bottom left inset).

to unseen tracks and real images ? Referring to our video⁴ and paper, we show our agent is able to drive on two unseen tracks having different road layout proving that it incorporated general driving concepts rather than just learned the tracks by heart. To also get a feel of transferrability to the *real* domain, we inferred control on real image sequences (web footage, cropped and resized) having situations never encountered during training (other road users, multi-lanes). Despite our open-loop setup (i.e. control commands are never applied), the early qualitative results in video and Fig. 3.20 demonstrate surprisingly good performance. Retrospectively, this is likely to be because the vision task is rather simple (road detection) and the complexity lies in fact in the control stability – not assessed in open loop tests. Interestingly, these preliminary results showed that one day RL could be used as initialization strategy for decision making networks.

Leveraging an optimized version of this RL work, Valeo conducted a real experiment at the Consumer Electronics Show in 2017 where an RL agent was competing against real human players. Similarly, this work was featured in a few press releases.

Since this work, real-world RL driving has been demonstrated (Kendall et al., 2019) but transferring RL knowledge is a challenging task “due to a series of assumptions that are rarely satisfied in practice” as mentioned in Dulac-Arnold et al. (2019). The latter highlights the need in RL for smaller synthetic/real gap, continuous action space, incorporating hard safety constraints, and accounting for real control inaccuracies. An other important problem is the interpretability of RL decision. Schmidt et al. (2021) show for example the benefit of combining RL and decision trees, easy to interpret.

3.3.2 DRL with sparse reward

A common pitfall of dense rewards in RL is that they prevent agents from freely exploring the policy space. This is because the explored policies

⁴<https://www.youtube.com/watch?v=AF0sryuSHdY>

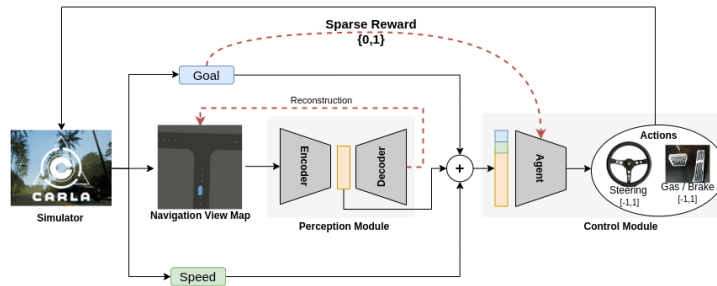


Figure 3.21: **Goal-constrained sparse RL.** We address RL using a VAE encoded top-view map along with speed and goal data, training with PPO (Schulman et al., 2017) and sparse binary reward upon goal completion.

are being penalized if they diverge from the reward shaping function. An example of interest for us is ‘taking a turn from the inside’ which – although optimal for driving – can hardly be discovered by an agent that is penalized by negative rewards as it drifts away from the road center.

To relax guidance, sparse RL assigns only simple reward when the task completed or failed. Because of its weaker guidance, sparse RL is known to be very challenging but encourages self discovery of policies.

Going further...

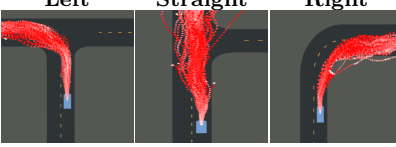
Agarwal, P., de Beaucorps, P., and de Charette, R. (2021). Sparse curriculum reinforcement learning for end-to-end driving. *arXiv*

Goal-constrained end-to-end driving. In Agarwal et al. (2021) we investigate sparse RL end-to-end driving with goal-constrained binary rewards. While sparse RL has been vastly explored for robotics tasks (Vecerik et al., 2017; Strudel et al., 2020; Zuo et al., 2020), it was not yet applied to driving due to the high-dimensionality and long-term horizon of the driving task. In fact, while our work was initially meant to leverage driving rules (speed limits, give-ways, traffic lights) with virtual driving-license points, our early experiments unveiled the complexity of the task. Instead, we addressed driving in simple city-like environments with limited success.

Rather than front-view images, here we use as input a top-view map of the vehicle’s surroundings, shown in Fig. 3.21, compressed with a Variational AutoEncoder (VAE). The benefit is not only to reduce the input dimensionality and ease sparse learning but also to lower the virtual/real domain gap. The agent receives only a binary reward ; 1 if the goal is reached before the episode ends, 0 if not.

In reality, because of the reward sparsity it is nearly impossible to solve a long-horizon task like driving. Common practices to address this imply using curiosity or curriculum learning. We follow the latter and

	Goal distance				
	20m	50m	100m	200m	300m
Ours	0.91	0.90	0.90	0.69	0.51
w/o curr. revert	0.51	0.08	0.02	0	0
w/o g. constraints	0.87	0.44	0.11	0	0
w/o ep. duration	0.36	0.04	0.00	0	0



(b) Driving characteristics
(100 runs)

(a) Success rate

Figure 3.22: **Test tracks performances.** (a) Success rate of our method and variants without our curriculum revert, goal constraints or variable episode duration. Notice the agents are trained only up to 100m, but can still generalize to longer distances. (b) Driving styles on *unseen* road layouts exhibit the agent learned to generalize and discovered some natural driving behavior (eg. turning from the inside, keeping its right).

break driving into smaller sub-tasks addressed in a curriculum fashion – where the goal is to reach a virtual finish line which distance from start grows with complexity. We introduce two mechanisms. First, rather than monotonically increasing the complexity, we propose a revert strategy – inspired by Dasagi et al. (2019) – where complexity is only increased upon success, and decreased upon failure. This plays the important role to prevent the policy to be stuck in an invalid local minimum. Second, we add dynamic goal constraints a maximum distance ρ from the goal center and a maximum angle deviation α from the road. This intuitively encourages better policy quality by preventing suboptimal goal achievements.

We train agents in Carla simulator (Dosovitskiy et al., 2017) with the Proximal Policy Optimization (PPO) algorithm (Schulman et al., 2017) which optimizes a surrogate objective L_θ (θ the network parameters) that learns policy π while avoiding large deviation from last policy $\pi_{\theta_{\text{old}}}$. It writes

$$L_\theta = E_t \left[\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)} A_t^{\pi_{\theta_{\text{old}}}} \right], \quad (3.14)$$

s_t and a_t being the environment state and action, respectively. In practice, we train 3 individual policies (drive straight, turn left and turn right) on small road portions, with complexity from 1m to 100m and episodes duration scaled with the goal distance to encourage speed. Fig.3.22a shows success rates on unseen test tracks. It is worth noting the task complexity and the generalization capability of our agent since it was trained only up to 100m, but can still drive 300m distances.

While our research results are still at an early stage – and far from reward shaping RL –, we notice the emergence of driving characteristics. For example in Fig. 3.22b where the agent discovered natural behaviors,

such as keeping its right or taking a turn from the inside – without any such guidance. In fact, sparse RL appears a good alternative to reward shaping to circumvent its pitfalls for driving (Knox et al., 2021), but we denote an absence of research in that direction. Similar choices than ours are observed in Zhang et al. (2021) which considers task-oriented RL for driving but leveraging both front-view and map-view images with reward shaping.

Vision and physics

Contents

4.1	Physics-informed vision	64
4.1.1	Reactive scene illumination	65
4.1.2	Physics-based rendering	66
4.2	Physics-guided learning	72
4.2.1	Model-guided disentanglement	72
4.2.2	Model-guided learning	78

In the previous works we have addressed computer vision in a physics-agnostic manner, considering images as matrices from which we seek to discover statistical relationship between pixels to solve a vision task. However, visual data captures photons which interaction with the world is for the most part well understood from centuries of physics studies. While there are evident basic understanding of physics in any vision algorithm, we question here *How more physics grounding could help vision ?*

An evident interest for algorithms to be more physics grounding is to overcome the shortcoming of pure data-driven method. Indeed, no dataset will ever be *complete* because the continuous nature of the physical world make it virtually impossible. In other words, no dataset – even having billions of data – will ever encompass all natural conditions and algorithms are forever doomed to train on a minimal subsampling of the world. While deep networks have great generalization properties, besides interpolation they cannot (without additional guidance) accommodate efficiently to



Figure 4.1: **Adverse weather and lighting conditions.** We demonstrate that physics models can improve vision, especially in adverse lighting and weather conditions. Sources: (Sakaridis et al., 2021; Xu et al., 2017).

unseen conditions.

Let’s consider a naive training on *day* and *night* images. A walk on the learned manifold will lead to intermediate conditions to be an interpolation of the two, whereas naive physics tells us that as the sun goes down, shadows move and sky turns redish. The same holds for other conditions like dynamic weathers as their visual effects cannot be trivially interpolated or extrapolated¹. This is the case for rain or snow as their visual effects drastically change with the weather intensity due to changing particle size distribution (Garg and Nayar, 2007). Physics-grounded algorithms could not only provide a better understanding of the scene appearances, but could also help compensating for data scarcity by providing a model for inter-/extra-polation of the seen conditions.

At the heart of this section is the application of vision in adverse lighting and weather conditions. Since the seminal work of the CAVE laboratory (Narasimhan and Nayar, 2002; Garg and Nayar, 2007), adverse weathers have been fascinating for me because they break the premise that the atmosphere behaves as a transparent medium. As opposed to clear weather, the particles in degraded weather alter our perception of the scene.

We detail the two lines of work addressed in this section.

In Sec. 4.1 we investigate how highly realistic physical models of adverse weather can boost vision in the *real world*, or used to augment clear weather images to boost performance of deep networks.

Instead in Sec. 4.2 we gradually relax the need of realistic physical models. Considering generative networks, we show that models can be used to learn visual disentanglement, or to guide continuous image translation.

4.1 Physics-informed vision

Here, we leverage high-precision physical-models to model the dynamics and photometrics of particles in the atmosphere to improve the visibility in adverse particulate weathers such as fog, rain and snow.

We first brush a former work conducted at Carnegie Mellon University (de Charette et al., 2012) on an adaptive lighting device fueled by in-depth study of particles physics, which we reused in our recent works to aug-

¹Narasimhan and Nayar (2002) group particulate weathers as *static* or *dynamic*, whether the particles in suspension are in motion (ie. whether gravity is stronger than the air pressure, pulling particles down). Static weathers (eg. mist, fog) have a monotical effect on vision as a function of distance, whereas dynamic weathers (rain, snow, hail) produce spatio-temporal visual artefacts which is harder to model or anticipate.

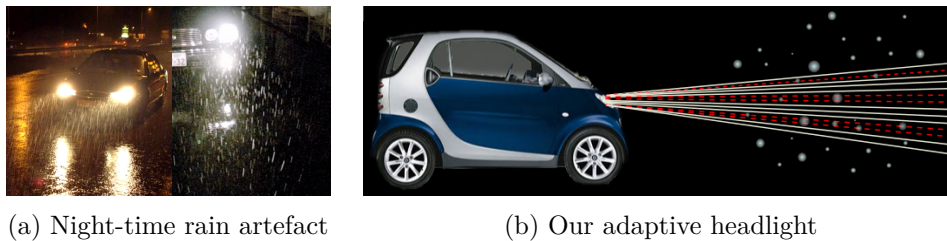


Figure 4.2: **Smart adaptive headlights.** (a) At night, rain or snow produce a bright flickering (distracting) pattern reducing driver visibility. (b) Our reactive lighting device deactivates only those light rays intersecting raindrops, snowflakes, or hailstones to diminish the effect of falling precipitation.

ment clear weather images with synthetic fog (Kahraman and de Charette, 2017) or rain (Halder et al., 2019; Tremblay et al., 2020).

4.1.1 Reactive scene illumination

When driving at night in rain, raindrops behave as small lenses reflecting the surrounding lights and producing bright flickering rain streaks that disrupt the scene visibility (Fig. 4.2a). In de Charette et al. (2012) we take advantage of the fact that rain is only visible when illuminated by light sources, and propose an approach that directly removes the appearance of rain *in* the scene by *selectively* illuminating the scene, as in Fig. 4.2b.

Our custom device (Fig. 4.3a) considers an imaging system collocated with an illumination device having controllable light rays. To illuminate between particles, we locate raindrops in consecutive frames, predict their dynamics leveraging physical models, and then reactively deactivate rays of light that would intersect particles. Because the latter are moving at high speed² and since visibility is crucial we identified two challenges: the need of a responsive system, and the need to preserve high light throughput. In practice, we limit our operating range to drops within 3m since our lab experiments showed drops are not visible beyond that.

Relying on our high precision particle simulator – which models all physics, imaging, lighting, and processing – we showed that a realistic 120Hz system having 13ms response time would successfully preserve 95% throughput while avoiding 87% of the raindrops when stationary and 43% when driving at 90 km/h (see light maps in Fig. 4.3b). The finding was validated building a prototype made of a DLP projector and a fast camera of 120×244 resolution. Using a controllable rain test bed, our lab

²A falling 3mm raindrop rushes towards the earth at $\approx 9\text{m/sec}$.

Going further...

Our particle simulator is freely distributed here: <https://github.com/cv-rits/weather-particle-simulator>

de Charette, R., Tamburo, R., Barnum, P. C., Rowe, A., Kanade, T., and Narasimhan, S. G. (2012). Fast reactive control for illumination through rain and snow. In *ICCP*

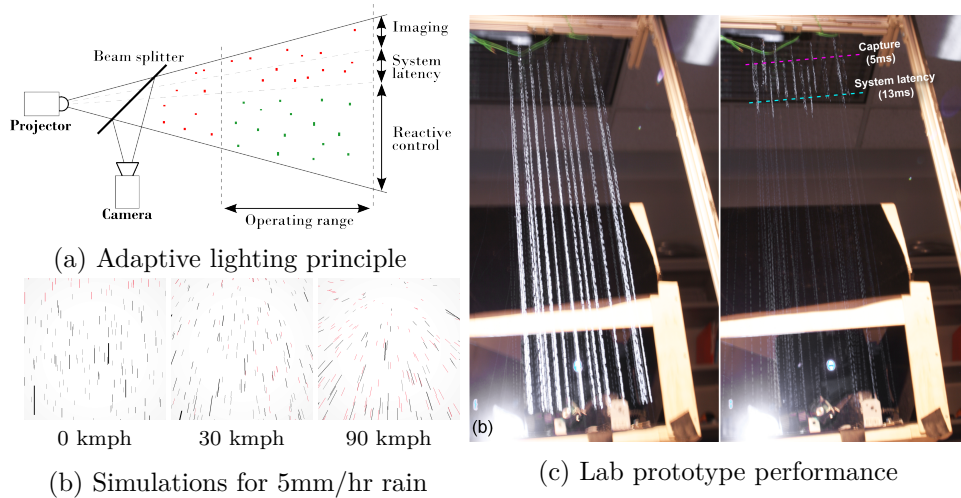


Figure 4.3: **Concept, Simulations and Experiments.** (a) We assume a co-located camera-projector system images to first briefly illuminate falling particles which are detected, and their future locations predicted for selective illumination. (b) Simulations for 5mm/hr rain shows that limited light rays are turned off (black pixels) and few particles remain shined (red). (c) Using an indoor rain test bed, our real prototype shows raindrop visibility is significantly diminished while preserving overall illumination.

experiments show that drops are significantly less visible with our adaptive lighting device (Fig. 4.3c) while preserving scene illumination. The same prototype was later tested in real-rain or snow conditions³ showing the same benefit.

A compact lighting device, fitting in a SUV headlight storage, was later developed by the Carnegie team. In [Tamburo et al. \(2014\)](#) they also extended the concept of selective illumination to avoid blinding upcoming drivers or to highlight danger.

4.1.2 Physics-based rendering

Instead of reducing bad weather appearance, we now use physics models to oppositely render unseen weather conditions on clear weather images. Adding arbitrarily controlled weather (eg. 15mm/hr rain) would not only allow benchmarking performance of vision algorithms but also improving their robustness.

Indeed, while all computer vision practitioners know that bad weather affect our algorithms, few knows how much it affects them. This is be-

³See videos at: <https://www.cs.cmu.edu/smartheadlight/index2.html>

cause little datasets encompass adverse weathers and *none* include weather-calibrated data.

Fog. As first steps, in our research report [Kahraman and de Charette \(2017\)](#) we studied fog rendering. Fog is by far the easiest weather because its particles are small (between $1\mu\text{m}$ – $10\mu\text{m}$), not individually detected by cameras, so that rendering only requires simulation of light attenuation and atmospheric airlight⁴. Hence, fog can be rendered on a clear image I_0 with:

$$I = I_0 e^{-\beta d} + L_\infty (1 - e^{-\beta d}), \quad (4.1)$$

where d is the scene depth, L_∞ is the horizon radiance and β is the atmospheric extinction coefficient (ie. how thick is the fog). In our work, we also account for the spatial heterogeneous characteristics of fog by replacing β with $N(\beta)$ where $N(\cdot)$ is a 3D Perlin noise with smooth gradient ([Gustavson, 2005](#)). Our fog rendering capacity is demonstrated in the next work.

Rain. Fueled by our collaborative grant, we also worked with Université Laval (Canada) on Physics-Based Rendering (PBR) of rain in [Halder et al. \(2019\)](#), studying various vision tasks, and extending to GAN and GAN+PBR in [Tremblay et al. \(2020\)](#). Rain is significantly harder to render due to the large falling particles and their complex light refractive behavior.

Relying on well-understood physical models we are able to control the *amount* of synthetic rain in order to generate arbitrary weather, ranging from very light rain (1 mm/hour rainfall) to very heavy storms (200+ mm/hour). This key feature allowed us to produce the first (still the only) calibrated rain-augmented datasets, ie. where the rainfall rate is known *and* physically calibrated. We used our weather augmented versions of KITTI ([Geiger et al., 2013](#)), Cityscapes ([Cordts et al., 2016](#)) and nuScenes ([Caesar et al., 2020a](#)) to evaluate the robustness of 14 popular algorithms – for object detection, semantic segmentation and depth estimation – in adverse weather. In Fig. 4.4 left, sample algorithms outputs on clear and augmented rain versions show the extent of degradation caused by adverse weather. Spoiling our findings, our benchmark (Fig. 4.4, right) indicates that stormy rain affects *all* algorithms with a performance drop of 15% mAP for object detection, 60% AP for semantic segmentation, and a 6-fold error increase in depth estimation.

Physics-Based Rendering (PBR). We simulate a controllable rain in an arbitrary image with the pipeline summarized in Fig. 4.5a, inspiring from the vast literature of raindrops physics. Unlike fog, rain particles are bigger and

⁴In fog, the atmosphere behaves as a source of light due to the scattering of environmental illumination (e.i. sunlight, skylight, etc.) by particles in the atmosphere.

Going further...

Kahraman, S. and de Charette, R. (2017). Influence of fog on computer vision algorithms

Going further...

Code and Weather augmented KITTI, Cityscapes, nuScenes: <https://github.com/cv-riits/rain-rendering>

Halder, S. S., Lalonde, J.-F., and Charette, R. d. (2019). Physics-based rendering for improving robustness to rain. In *ICCV*

Tremblay, M., Halder, S. S., de Charette, R., and Lalonde, J.-F. (2020). Rain rendering for evaluating and improving robustness to bad weather. *IJCV*

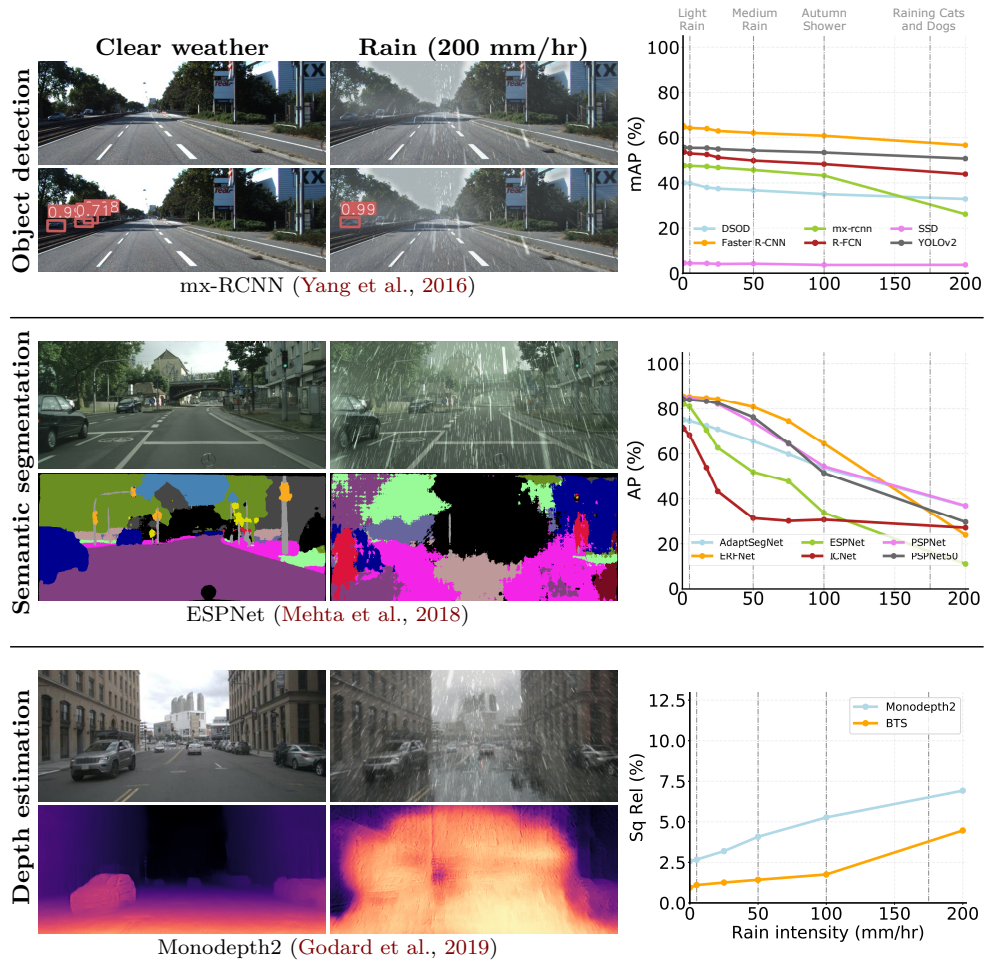


Figure 4.4: **Vision tasks in clear and rain-augmented images.** Our rain rendering framework can augment clear-weather images in a controlled manner. It was used to produce rain-augmented versions of KITTI (Geiger et al., 2013) (rows 1-2), Cityscapes (Cordts et al., 2016) (rows 3-4) and nuScenes (Caesar et al., 2020b) (rows 5-6). Leftmost column shows clear-weather performance for each vision task and its performance on our rain-augmented version in the middle column. Rightmost column shows the corresponding benchmark for each vision task on a total of 14 recent networks. Overall, all algorithms are quite significantly affected by rainy conditions, and when *raining cats and dogs* (200mm/hr) it leads to a performance drop of 15% mAP for object detection, 60% AP for semantic segmentation, and a 6-fold error increase in depth estimation

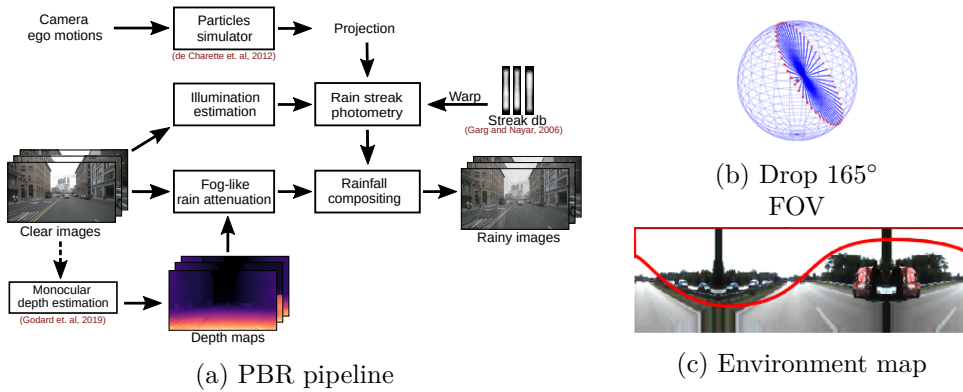


Figure 4.5: **Physics-based rendering of rain.** (a) Our PBR pipeline leverages the vast literature about rain, our prior particle simulator (de Charette et al., 2012), and the CAVE appearance streak dataset (Garg and Nayar, 2006) to render both volumetric rain and individual falling raindrops. To render the photometry of raindrops given their large FOV (b), we approximate a scene environment map (c) and compute the drop radiance from the projection of its FOV in the latter (in red).

thus harder to synthesize. Our pipeline follows the definition of Garg and Nayar (2007) and models successively fog-like rain and individual drops.

For the former, given a clear image I_0 , the effect of drops that project on less than 1 pixel is rendered with a volumetric attenuation:

$$I_{\text{att}}(\mathbf{x}) = I_0 L_{\text{ext}}(\mathbf{x}) + A_{\text{in}}(\mathbf{x}), \quad (4.2)$$

where

$$\begin{aligned} L_{\text{ext}}(\mathbf{x}) &= e^{-0.312R^{0.67}d(\mathbf{x})}, \\ A_{\text{in}}(\mathbf{x}) &= \beta_{\text{HG}}(\theta)\bar{E}_{\text{sun}}(1 - L_{\text{ext}}(\mathbf{x})), \end{aligned} \quad (4.3)$$

with R the rainfall rate (in mm/hr), $d(\mathbf{x})$ the pixel depth, β_{HG} the standard Heynyey-Greenstein coefficient, and \bar{E}_{sun} the average sun irradiance.

Large close drops are instead rendered individually, relying on our particle simulator (de Charette et al., 2012) providing us with positions and dynamics of all raindrops for a given fallrate. Synthesizing the exact photometry of a drop is complex for two reasons: a) as it falls a drop oscillates making its visual effect uneven during shutter opening, b) the field of view of a drop is larger than common cameras (165° vs approx. 70–100°) hence each drop images a large portion of the scene (Figs. 4.5b,c). To overcome these, we used the CAVE streak appearance dataset (Garg and Nayar, 2006) warping the queried streaks to match our particle simulator outputs.

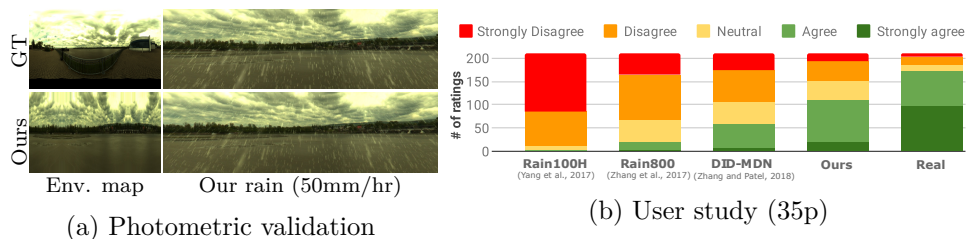


Figure 4.6: **Evaluation of our rain realism.** (a) We compare our rain rendering on the Outdoor HDR ULaval dataset (Hold-Geoffroy et al., 2019) using either ground truth (GT) map or our approximated version (Ours). Results show only subtle differences. (b) User study comparing 30 randomly selected images of our renderings, competitors and real rainy images, where users judged ‘if rain looks realistic’ on a 5-point Likert scale.

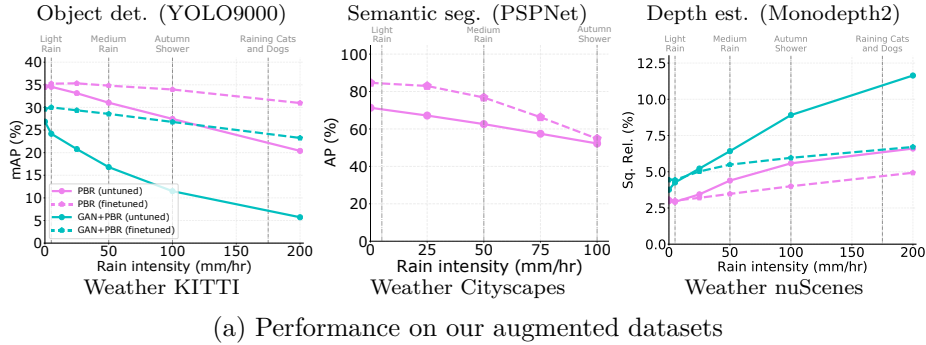
Rather than computing the exact drop photometry – virtually impossible as it requires perfect knowledge of the scene materials and geometry – we estimate a coarse environment map E from a series of optical approximation and compute each drop radiance from the projection F of its field-of-view on the former (Fig. 4.5c, red line). Finally, since a drop refracts 94% of its field of view radiance and reflects 6% of the entire scene radiance (Garg and Nayar, 2007), the final streak appearance S' is:

$$S' = \mathcal{H}(S)(0.94\bar{F} + 0.06\bar{E}), \quad (4.4)$$

where $\mathcal{H}(S)$ is the warped S streak appearance and $\bar{\cdot}$ is the mean operator. The final rain appearance is obtained from the rainfall composite of all individual streaks on the fog-like rain image I_{att} .

Because no rain-calibrated dataset exists, we evaluated our rain realism in two-fold. First, in Fig. 4.6a we rendered rain on the Outdoor ULaval HDR dataset (Hold-Geoffroy et al., 2019) using either the provided ground truth illumination or our approximation which shows that the rain realism is comparable. Second, we conducted a user-study with 35 participants asked to judge rain realism with a 5-point Likert scale, and compared our performance against competitors and real rain images. Histograms in Fig. 4.6b show our realism is judged significantly better.

Image-to-image translation (GAN, GAN+PBR). A limitation of Physics-Based Rendering is that it ignores major rainy characteristics such as wetness, reflections, clouds and thus may fail at conveying the overall look of a rainy scene. Hence, in Tremblay et al. (2020) we compare our PBR against image-to-image clear \mapsto rain translations (hereafter, GAN) and a hybrid mix of the two (GAN+PBR). We train a CycleGAN (Zhu et al.,



	Object detection YOLO9000		Semantic seg. PSPNet		Depth estimation Monodepth2	
	mAP (%) \uparrow		AP (%) \uparrow		Sq. err. (%) \downarrow	
	Clear	Rain	Clear	Rain	Clear	Rain
Untuned	32.53	16.30	40.8	18.7	2.96	3.53
Finetuned (PBR)	33.51	<u>19.68</u>	<u>39.0</u>	25.6	3.15	3.54
Finetuned (GAN)	32.26	18.07	*	*	2.89	3.40
Finetuned (GAN+PBR)	30.59	19.73	*	*	3.01	3.29
De-rained (Liu et al., 2019b)	32.60	18.30	*	*	2.25	3.09

* Lack of semantic labels for GAN training.

(b) Performance on real nuScenes (Caesar et al., 2020b)

Figure 4.7: **Finetuned performance.** (a) On our rain-augmented datasets YOLO9000 (Redmon and Farhadi, 2017), PSPNet (Zhao et al., 2017) and Monodepth2 (Godard et al., 2019) perform better for all fallrate when finetuned. (b) On real rainy images from nuScenes a boost is also observed when finetuned with any of our rain augmentations.

2017) on unpaired clear/rainy images of nuScenes (Caesar et al., 2020b).

Assessing and improving robustness to rain. In Halder et al. (2019); Tremblay et al. (2020) we thoroughly evaluate 14 state-of-the-art algorithms – 6 object detection, 6 semantic segmentation, and 2 depth estimation. Plots in Fig. 4.4 (right) show performance on our augmented weather KITTI, Cityscapes, and nuScenes, respectively from top to bottom. Among all 3 tasks, semantic suffers significantly more from rain artefacts.

While the effect of rain augmentation is evident, the story is missing a piece since the goal is rather to improve robustness to real rain. To assess the benefit of our augmentations, we retrained 3 algorithms – YOLO9000 for object detection (Redmon and Farhadi, 2017), PSPNet for segmentation (Zhao et al., 2017) and Monodepth2 for depth estimation (Godard et al., 2019) – finetuning them on our PBR, GAN or GAN+PBR augmentations in a curriculum fashion (i.e. successively on 0, 25, ..., 100 mm/hr fallrate). In Fig. 4.7a, finetuned networks (dashed lines) show an important boost on our rain-augmented datasets over untuned versions (plain lines) on all three tasks.

A comparable but smaller boost is observed in Fig. 4.7b on real clear/rainy images of the nuScenes dataset. It is interesting to note that finetuning on PBR rain is sufficient to boost performance in real rain *without needs of real rainy images at training*, though GAN+PBR combination provides the best results but requires real training rain images not always available. We also demonstrate that deraining first images performs always worse than our finetuning. Still, even with our improvement we notice the significant clear/rain performance gap showing that further efforts are required to be truly robust to such challenging conditions.

4.2 Physics-guided learning

Some visual traits (eg. wetness, puddles, etc.) are just too complex to be physically rendered on images but are *easily* learned by generative networks. Inversely, the latter are well known for being visually pleasant but not physically realistic since they only optimize a perceptual difference.

In the PhD of Fabio Pizzati we address the use of physics to guide the training of generative networks, by either enforcing *disentanglement* between the learned visual traits and the ones physically modeled (Pizzati et al., 2020a, 2021b), or by *guiding* the learned manifold discovery with simple physical models (Pizzati et al., 2021a).

4.2.1 Model-guided disentanglement

Naive combination of GAN and physics-based rendering was done in our work (Tremblay et al., 2020) but has important limitations since the GAN may partially entangle some physically rendered traits. This is visible in Fig. 4.8 (top) where a standard image-to-image translation (i2i) trained on a clear \leftrightarrow rain task evidently entangles blurred raindrops in the outputs (red circles) because they are present in the target dataset (ie. raindrops on the lenses or on the windshield). For what is more, we show in the following that entanglement also hinders the real underlying translation task.

Going further...

Pizzati, F., Cerri, P., and de Charette, R. (2020a). Model-based occlusion disentanglement for image-to-image translation. In *ECCV*

Pizzati, F., Cerri, P., and de Charette, R. (2021b). Guided disentanglement in generative networks. *arXiv* submitted to IJCV

Guided disentanglement in generative networks. In recent works we investigated how to disentangle the learned representation of a GAN from a model acting as disentanglement-guidance, may it be a physics-based rendering model (Pizzati et al., 2020a, 2021b) or an other GAN output (Pizzati et al., 2021b). We focus here on the physics model guidance only. The benefit to learn a disentangled representation is to allow simple-to-render characteristics (eg. raindrops) to be physically modeled and to learn

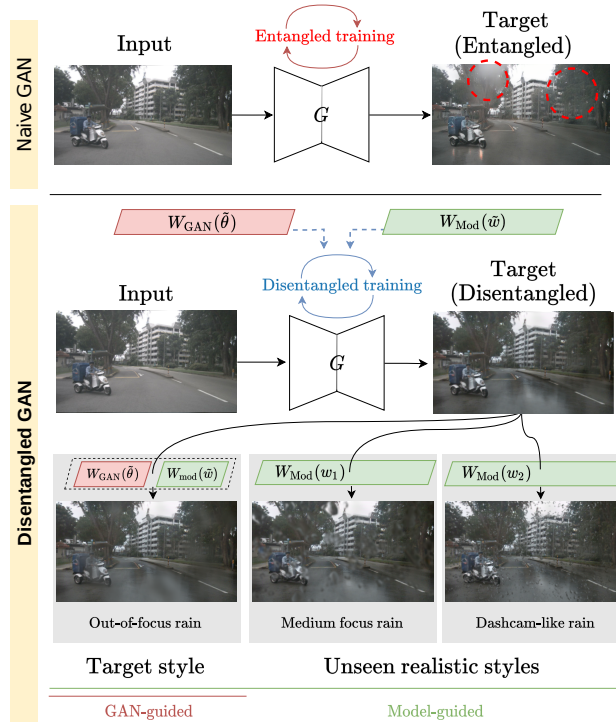


Figure 4.8: **Guided disentanglement.** *Figure is best seen on a screen.* We learn characteristics disentanglement in target, from $W(\cdot)$ guidance which might be neural or physical models (differentiable or not). Different from naive GANs generating *entangled target* images, we learn a *disentangled* version of the scene from guidance of model $W(\cdot)$ with target estimated physical (\tilde{w}) or neural ($\tilde{\theta}$) parameters.

others (eg. reflections) with a generative network. This allows us to inject arbitrary realistic styles *unseen during training* (Fig. 4.8, bottom).

Considering a standard $X \mapsto Y$ image-to-image translation, the task of the generator is to approximate the probability distributions P_X and P_Y associated with the problem domains, such as

$$\begin{aligned} \forall x \in X, x &\sim P_X(x), \\ \forall y \in Y, y &\sim P_Y(y). \end{aligned} \quad (4.5)$$

Let us assume a subdomain decomposition of target such that $Y = \{Y_W, Y_T\}$ separates the physically modeled traits (Y_W) from the other ones (Y_T). If we hypothesize both traits are independent, we can formalize P_Y as a joint probability distribution with independent marginals, such as

$$P_Y(y) = P_{Y_W, Y_T}(y_W, y_T) = P_{Y_W}(y_W)P_{Y_T}(y_T), \quad (4.6)$$

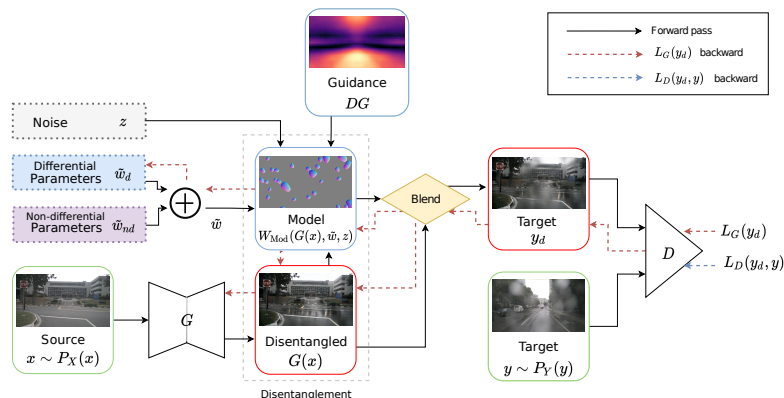


Figure 4.9: **Model-guided disentanglement.** Our disentanglement leverages a physical model $W_{\text{Mod}}(\cdot)$ – parameterized by differentiable and non-differentiable parameters – added to the generated image $G(x)$. This pushes the GAN to learn the non-modeled characteristics in a disentangled manner. Green stands for real data, red for fake ones.

so that by imposing a strict guidance on P_{Y_W} , the GAN is let to learn P_{Y_T} in a disentangled manner. In practice, the guidance is enforced by injecting features belonging to Y_W before forwarding the images to the discriminator, which provides feedback on the general realism of the image and pushes the generator to estimate the remaining P_{Y_T} .

Fig. 4.9 depicts the overall pipeline. Notice the model $W_{\text{mod}}(\cdot)$ rendering the physical traits, and the learned disentangled representation $G(x)$. Though illustrated here with raindrops, in Pizzati et al. (2020a) we experiment on 4 tasks (raindrops, dirt, and other composite tasks) and extend in Pizzati et al. (2021b) to 1 new task (fog) having interesting entanglement challenges – since fog is physically entangled with the scene.

To ensure a proper disentanglement, it is important that the guidance realistically estimates P_{Y_W} ; but models are by nature parameters dependent. For example, a raindrop appears drastically different whether in-focus or out-of-focus (see samples in Fig. 4.8, bottom). In our work, we introduce an adversarial mechanism to estimate the optimal target-style parameters \tilde{w} . For differentiable parameters (Fig. 4.10a), we rely on a premise that given a frozen pretrained discriminator (D^{ent} trained to differentiate X and Y), and a source image on which is applied our physical model W_{mod} , a global minimum can only be found by adjusting the differentiable model parameters w_d to mimic target, thus leading to \tilde{w}_d . Non differentiable parameters w_{nd} are optimized on target leveraging genetic optimization (Fig. 4.10b), thus leading to \tilde{w}_{nd} .

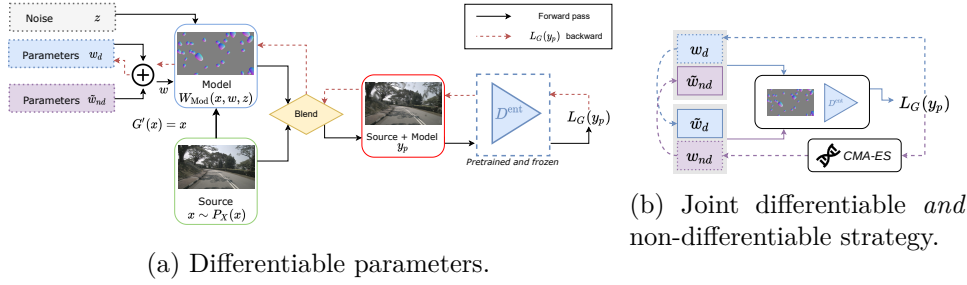


Figure 4.10: **Model-guided parameters estimation.** To properly guide disentanglement, the optimal model parameters are regressed on target. (a) For those differentiable, we exploit a pretrained *frozen* discriminator D^{ent} which makes the gradient flows only in the parameters direction. (b) For non-differentiable parameters, we optimize until convergence alternatively differentiable parameters as mentioned and non-differentiable parameters using a black-box genetic optimization (Hansen et al., 2003).

It is interesting to note that a naive solution exists to our optimization problem since W_{mod} can be injected where local $X \mapsto Y$ translations are too complex to learn⁵. To prevent this, we introduce in our work a learnable disentanglement guidance-map (DG in Fig. 4.9) which acts as a regularizer to the disentanglement process.

Experiments. Evaluating the quality of our disentanglement on existing datasets is non trivial since none of the latter provide disentangled versions (eg. rainy scenes *without* raindrops, foggy scene *without* fog, etc.). We thus introduced 5 experiments denoted as source \mapsto target_{ent} where subscript denotes the *entangled* traits in target, with the aim to learn the *disentangled* source \mapsto target (notice the absence of ent). Selected results for some tasks are in Fig. 4.11.

For fair comparison against baselines, in Figs. 4.11b,c we provide not only our *disentangled* versions ($\mathcal{T}_{W_{\text{Mod}}}^{w_d}$ for model-guided, $\mathcal{T}_{W_{\text{GAN}}}$ for GAN-guided) but also our disentangled output *with re-injection* of the guided traits to resemble target-style (ie. $\mathcal{T}_{W_{\text{Mod}}}(\tilde{w})$, $\mathcal{T}_{W_{\text{GAN}}}(\tilde{w})$). Our disentangled versions exhibit that neither raindrops (Fig. 4.11b) nor dirt (Fig. 4.11c) are entangled though other target characteristics are fully preserved – wetness and color, respectively. To quantitatively evaluate our translations (in the absence of disentangled ground truths), we com-

⁵While counter intuitive, in Pizzati et al. (2020a, 2021b) we elaborate on the fact that complex translations are those having *little visual changes* from source to target. For example, in Fig. 4.9 notice how the entangled drops are always located in the trees. This is because they look alike when dry or wet so entangling raindrops is a trivial simple optimization minimum to fool the discriminator.

Experiment datasets	Method	Guidance	IS \uparrow	LPIPS \uparrow	CIS \uparrow
clear \mapsto rain_{drop} nuScenes	CycleGAN	-	1.15	0.473	-
	AttentionGAN	-	1.41	0.464	-
	U-GAT-IT	-	1.04	0.489	-
	DRIT	-	1.19	0.492	1.12
	MUNIT	-	1.21	0.495	1.03
	Model-guided $\mathcal{T}_{W_{\text{Mod}}^{w_d}(\tilde{w}_d)}$	raindrop (1 9)	1.53	0.515	1.15
gray \mapsto color_{dirt} WoodScape	MUNIT	-	1.06	0.656	1.08
	Model-guided $\mathcal{T}_{W_{\text{Mod}}^{w_d}(\tilde{w}_d)}$	dirt (2 0)	1.25	0.590	1.15
	GAN-guided $\mathcal{T}_{W_{\text{GAN}}(\tilde{\theta})}$	GAN	1.58	0.663	1.47
synth \mapsto WCS_{fog} Synthia Weather Cityscapes	MUNIT	-	1.22	0.429	1.13
	Model-guided $\mathcal{T}_{W_{\text{Mod}}^{w_d}(\tilde{w}_d)}$	fog (1* 0)	1.33	0.420	1.17

(a) GAN metrics.

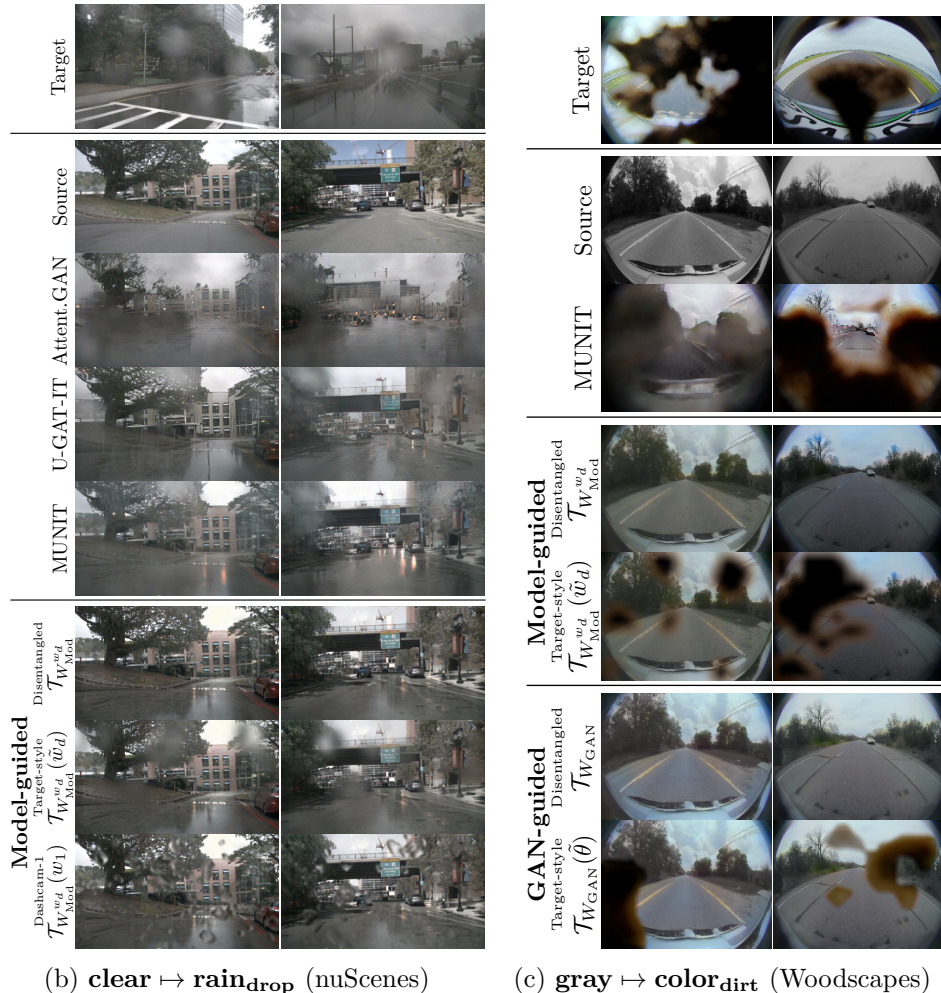


Figure 4.11: **Guided disentanglement performance.** In (a), we quantify GAN metrics for some of our tasks comparing our disentangled translations *with re-injection* of the modeled traits and target images. Bottom, we show sample outputs (b,c) and selected baselines. Our guided network is able to disentangle the generation of peculiar rain/color characteristics from the raindrop/dirt on the windshield (‘disentangled’ rows $\mathcal{T}_{W_{\text{Mod}}}$ / $\mathcal{T}_{W_{\text{GAN}}}$). In last rows of model-/GAN-guided outputs we re-inject droplets with optimal target style parameters \tilde{w} (‘Target-style’ rows $\mathcal{T}_{W_{\text{Mod}}^{w_d}(\tilde{w}_d)}$) or new *unseen* style (‘Dashcam-1’ row, left).

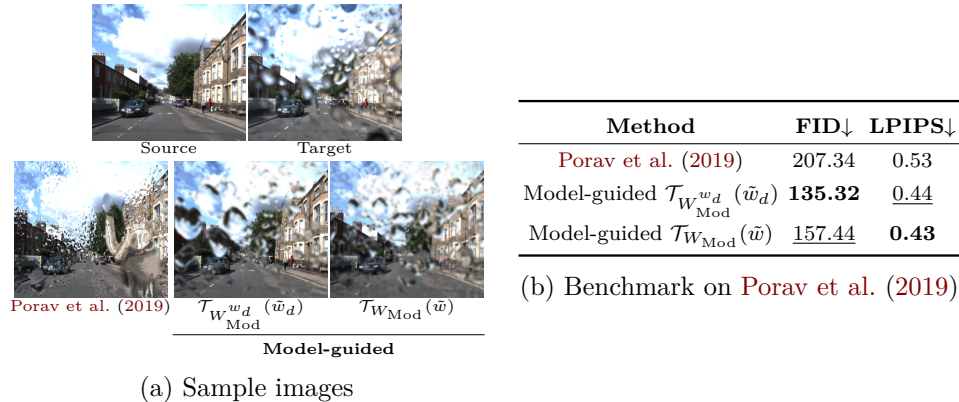


Figure 4.12: **Validity of our parameters estimation.** Our increased realism (a) in raindrop rendering on the RobotCar dataset (Porav et al., 2019) is also assessed quantitatively (b) with FID and LPIPS distances. Notice our translations better resembles target images (drops size/focus/etc.).

pare our target-style reinjected images against target ones in Fig. 4.11a, reporting the Inception Score (measuring quality and diversity), LPIPS (diversity), and Conditional Inception Score (multi-modal diversity). We outperform almost all metrics, which translate qualitatively in Figs. 4.11b,c.

In Pizzati et al. (2021b) we also ablate model-guidance, showing that even naive model can guide disentanglement. However, accurate parameters estimation is crucial for a proper disentanglement. Notably, for raindrops we leverage the RobotCar dataset (Porav et al., 2019) having pairs of clear/water-sprayed images and compare our automatic estimation of the raindrop parameters in Fig. 4.12 to their highly customized version. Our images better resemble target target qualitatively and quantitatively – whether considering the full model \mathcal{T}_{Mod} or only the differentiable one $\mathcal{T}_{Mod}^{w_d}$.

A crucial benefit of disentangling representations is to accommodate to unseen characteristics. This is especially important for traits like raindrops exhibiting high appearance variability with different camera setups. In Fig. 4.11b (bottom row) we exhibit our unique ability to re-inject *unseen* style of characteristics (here, dashcam-like raindrops never seen during training). Our work Pizzati et al. (2021b) shows these translations can also be used to boost vision tasks like semantic segmentation – allowing us to train semantics on a labeled dataset having unfocused raindrops, while performing well on dashcam-like images.

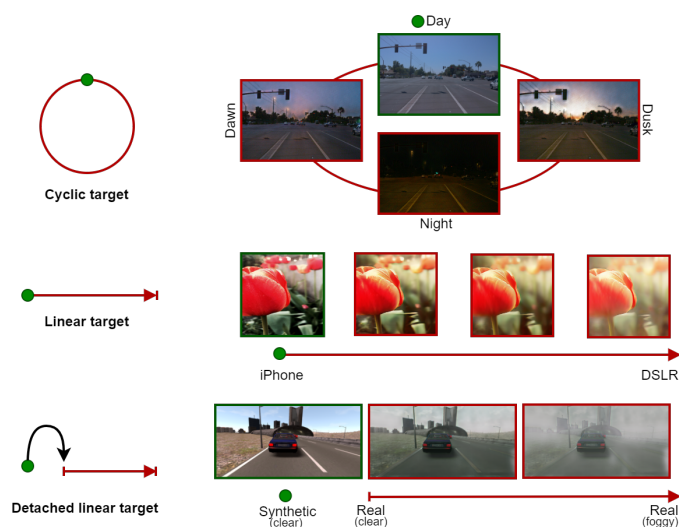


Figure 4.13: **CoMoGAN: continuous model-guided image-to-image translation.** Sample translation tasks for our architecture-agnostic framework learns the *continuous* mapping from source domain (green point) to a target domain (red lines). A key property, is the unsupervised reorganization of the data along a functional manifold (top: cyclic, middle/bottom: linear). From top to bottom: day to timelapse, in-focus to shallow depth of field, or synthetic clear images to realistic foggy images.

4.2.2 Model-guided learning

While we gradually relaxed the use of physics models, it is notable in the former works that physical rendering still condition the translations realism. Investigating the interaction of machine learning, physics and vision, we question now how physics can *guide* the manifold discovery so as to learn the *complete* i2i mapping.

Going further...

Code: <https://github.com/cv-rits/CoMoGAN>

Pizzati, F., Cerri, P., and de Charette, R. (2021a). Co-MoGAN: continuous model-guided image-to-image translation. In *CVPR*

CoMoGAN. In Pizzati et al. (2021a) we proposed a novel continuous model-guided image-to-image translation, coined CoMoGAN, where simple physics-inspired models are leveraged to guide the learning. Different from the previous works, we fully relax the model dependency by introducing a continuous disentanglement of domain features – making naive model (eg. tone-mapping, blurring, etc.) sufficient to guide the learning of complex mapping. Experiments show that it significantly and consistently outperforms the literature.

Fig. 4.13 shows some of our continuous translations for various manifold shapes. An interesting property we found is that CoMoGAN discovers the target data manifold ordering, unsupervised.

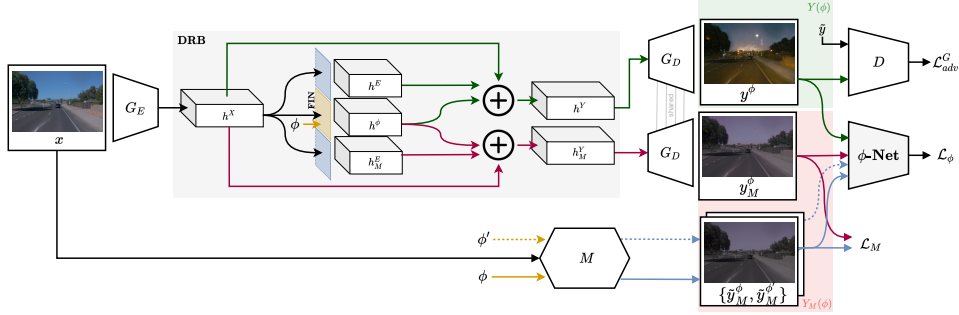


Figure 4.14: **CoMoGAN framework.** Our framework learns $X \mapsto Y(\phi)$ in an end-to-end, architecture-agnostic manner. The Disentanglement Residual Block (DRB) – between encoder/decoder (G_E/G_D) – uses a new Functional Instance Normalization (FIN, yellow layer) to learn manifold reshaping and continuous translation, guided with simple physics-inspired model M . On top of the standard ones, our losses optimize model reconstruction (\mathcal{L}_M) and manifold consistency (\mathcal{L}_ϕ) by enforcing ϕ distances between GAN output and model outputs $\{\phi, \phi'\}$ with a pair-wise estimator (ϕ -Net).

More in depth, CoMoGAN learns a continuous domain translation controlled by ϕ , that is $X \mapsto Y(\phi)$, while reshaping the data manifold guided by simple physics-inspired models. Importantly, we consider *unknown* ϕ labels in Y . Our architecture-agnostic framework in Fig. 4.14 relies on model-guidance $M(x, \phi)$ – with x the source image and $M(\cdot)$ the model.

A key feature is that we learn two domains, the target domain $Y(\phi)$ and the model one $Y_M(\phi)$. To avoid strict guidance (leading the GAN to only mimic the model) we allow $Y(\phi)$ and $Y_M(\phi)$ to have shared *modeled* features but also discover private *non-modeled* features. This is enabled with our Disentanglement Residual Block (DRB, shown in Fig. 4.14) whose goal is to extract disentangled representations for a given ϕ .

To inject guidance in the target domain $Y(\phi)$, we introduce constraints on the discovered manifold. First, we encode ϕ continuity with a novel Functional Instance Normalization layer (FIN, yellow in Fig. 4.14), taking advantage of our model guidance – continuous by nature –. It builds on prior Instance Normalization (IN) which carries style-related information (Ulyanov et al., 2017; Huang and Belongie, 2017) although, instead of a unique affine transformation of the input feature statistics (μ, β) , our FIN learns a distribution of transformations f_γ and f_β :

$$\text{FIN}(x, \phi) = \frac{x - \mu}{\sigma} f_\gamma(\phi) + f_\beta(\phi), \quad (4.7)$$

allowing the network to shape the ϕ -manifold based on how the transformation evolves. Depending on the nature of the transformation, we can parametrize f_γ and f_β accordingly: *cyclically* for daytime translations, or

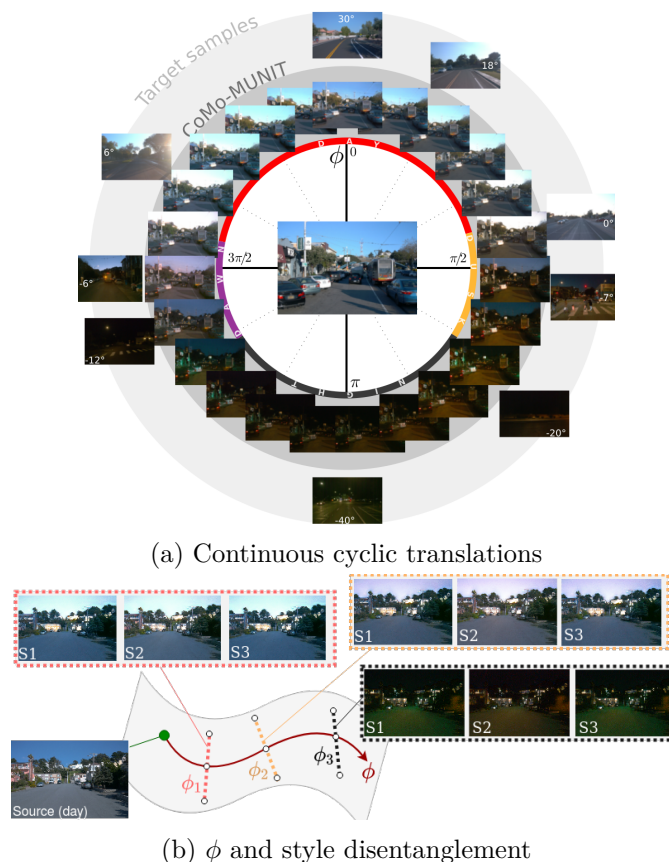
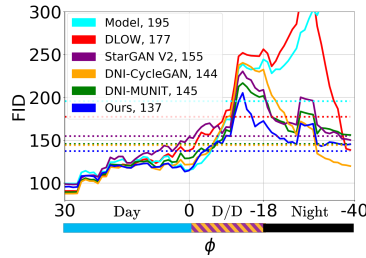


Figure 4.15: **Manifold organization.** (a) Our translations (dark circle) of a source image (center) are properly ordered and have non-modeled visual features (eg. frontal sun also in target samples, outer circle). With dawn/dusk singularities and night time stable appearance this assesses the manifold quality. (b) Disentanglement of ϕ (red) and style (dotted) is demonstrated since styles vary slightly per ϕ (notice hue and brightness).

linearly for translations like adverse weathers evolving monotonically.

The second constraint is to impose ϕ distances in the two discovered manifolds. Because we do *not* use target ϕ values, we consider pairs of *random* ϕ values and use our ϕ -Net (Fig. 4.14, right) to optimize pair-wise ϕ differences between the model $M(\cdot)$ and the *learned* target $Y(\cdot)$ or model $Y_M(\cdot)$ domains. The benefit of this distance constraint is to ensure images follow some similarity criteria despite differences between the model output and the learned translation. Importantly, this leads to an organization of the latent space guided by the physical model.

Experiments. We adapt our architecture-agnostic CoMoGAN to the popular MUNIT (Huang et al., 2018a) and CycleGAN (Zhu et al., 2017),



(a) Rolling FID

Method	Error	
	Mean↓	Std↓
Model	21.12	10.15
DLOW (Gong et al., 2019)	17.39	9.02
StarGANV2 (Choi et al., 2020)	15.91	10.00
DNI-CycleGAN (Wang et al., 2019c)	13.84	7.91
DNI-MUNIT (Wang et al., 2019c)	13.80	8.30
CoMo-MUNIT	9.84	7.20
Real data	3.61	4.52

(b) ϕ regression

Figure 4.16: **Translations realism.** (a) Rolling FID shows our method is more effective, especially at Dawn/Dusk (‘D/D’) *despite less supervision* (cf. Text). (b) Comparing the error between input ϕ translation values and the regressed ϕ with an InceptionV3 network (trained on real data), also advocate we outperform others.

referred as CoMo-MUNIT and CoMo-CycleGAN, respectively, and consider 3 continuous image-to-image translation tasks where source data lie on a fixed ϕ_0 point and target ϕ is *unknown*. The challenge is to learn simultaneously the orderly ϕ -manifold and the continuous image translation.

Leveraging the recent Waymo Open dataset (Sun et al., 2020), split into ‘Day’ (*source*) and ‘Dawn/Dusk/Night’ (*target*), we map ϕ to the sun elevation and demonstrate the ability to learn the cyclic Day \mapsto Timelapse translations. A tone-mapping serves as model-guidance $M(\cdot)$, simply darkening the image at night and shifting hue at dawn/dusk⁶. The cyclic translations learned with CoMo-MUNIT are in shown Fig. 4.15a, having source in the center. Apart from the visually pleasant translations, CoMo-MUNIT translations (inner circle) are correctly ordered and show we learned non-modeled features like frontal sun, sunset/sunrise, material reflectance at night, and importantly the stable nighttime appearance. None of these features are modeled in $M(\cdot)$ though present in target images (outer circle), which advocates from the effective disentanglement of $Y(\phi)$ and $Y_M(\phi)$ with our DRB. A fair concern here would be that ϕ could be entangled as style (since MUNIT is multimodal) but Fig. 4.15b shows the latter evolve correctly on different axes – as expected since ϕ is regulated by model-guided features.

Other qualitative translations are shown in Fig. 4.17 for CoMo-MUNIT and CoMo-CycleGAN for the linear iPhone \mapsto DSLR and detached linear Synthetic_{clear} \mapsto Real_{clear, foggy} tasks. The model-guidance is a simple gaussian blurring for the former, and our physical fog model (Kahraman

⁶Model guidance is detailed in Pizzati et al. (2021a), main and supp.

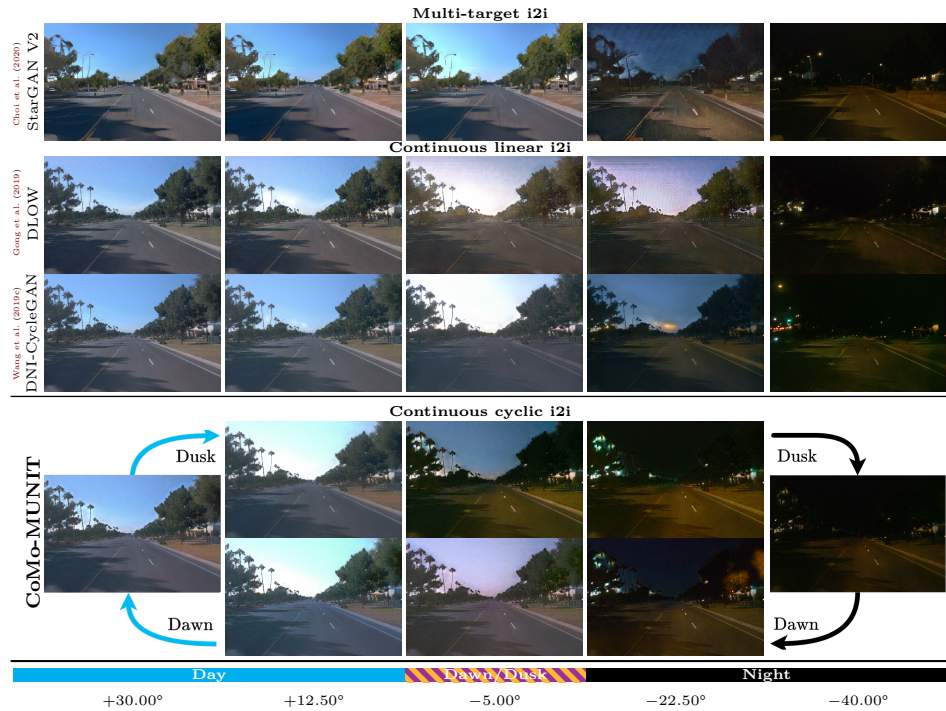
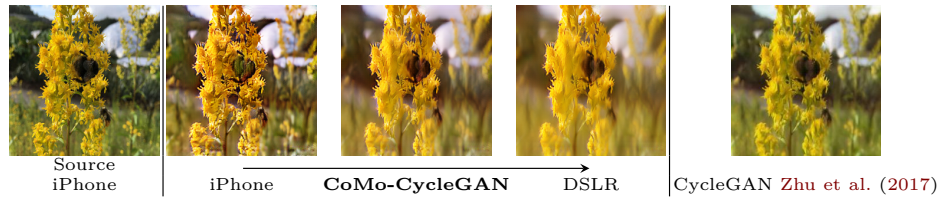
(a) Day \mapsto Timelapse (guidance: tone-mapping)(b) Synthetic_{clear} \mapsto Real_{clear, foggy} (guidance: fog model)(c) iPhone \mapsto DSLR (guidance: blur)

Figure 4.17: **CoMoGAN translations.** Translations on three $X \mapsto Y(\phi)$ continuous tasks show CoMoGAN discovers unmodeled features for all. (a) Unlike baselines, our framework enables ϕ to be cyclically encoded for timelapses allowing to distinguish Dawn and Dusk, and learns the – otherwise undiscovered – stable night appearance. (b) We experiment a detached linear target, where source (Synthia (Ros et al., 2016)) and target (clear/foggy Cityscapes (Cordts et al., 2016; Halder et al., 2019)) are not connected (ie. $X \subset Y$ or $X \not\subset Y$). (c) CoMo-CycleGAN translations on the iPhone \mapsto DSLR task, using iphone2dslr dataset (Zhu et al., 2017). Despite naive blur guidance, it learns continuous DSLR depth of field, though Zhu et al. (2017) only learns target translations.

and de Charette, 2017) for the latter. Again, there are evident non-modeled features discovered such as shallow depth of field, real textures, etc. Importantly, notice the selected baselines in Fig. 4.17a all use more supervision than us since they use Dawn/Dusk annotations.

While the perceptual benefit is evident, we also question the realism of our translations and their benefit for proxy tasks. Hence, in Fig. 4.16a we plot the rolling Frechet Inception Distance (ie. FID per ϕ interval) of timelapse translations versus baselines and real data (mean as dashed lines). This shows that CoMoGAN is particularly capable of capturing the unique Dawn/Dusk appearance – again, despite less supervision. The quality of our ϕ mapping is also evaluated with a proxy ϕ -estimation task, where an InceptionV3 network (Szegedy et al., 2016) is trained to regress sun elevation from real images and ϕ ground truths. Reporting the average ϕ estimation error of translated images and real images, in Fig. 4.16b, show our translations are significantly closer to real data. An important aspect is to note the difference between mean error of the tone-mapping ‘Model’ translations (21.12) which serve as guidance, and our ‘CoMo-MUNIT’ (9.84). This once more show CoMoGAN is *not* simply mimicking a model.

In Pizzati et al. (2021a) we also show that our translations can boost semantic segmentation (+3.2 mIoU) for example to adapt a Cityscapes model to Foggy Driving, using our $\text{Synthetic}_{\text{clear}} \mapsto \text{Real}_{\text{clear, foggy}}$ translations. We also demonstrate the ability to address other task like $\text{Cat} \mapsto \text{Dog}$ or translations trained with domain-confusion.

Research perspectives

In the following years, I plan to pursue my research on vision for scene understanding in the direct continuation of the three axes of studies developed before. Transversal to these line of works, my research will be conducted keeping in mind two important intertwined aspects for AI:

Relaxing supervision. Despite the ever increasing amount of training data, relying on full supervision appears as an intuitive non-sense even though ‘truly unsupervised vision’ is arguably possible. In fact, machine learning in its current form always relies on some sort of supervision may it be from a model, synthetic labels, labels in another domain, or else. The cost of supervision is however different. In that spirit, my near future works will focus on weakly-/self- and model-supervision. Among others, I intend to develop work on cross-modal learning (2D/3D, audio/video, etc.), and to rely more on physics knowledge to relax the need of labeled data. On a longer note I wish to investigate the self-supervised discovery of physics laws in data inspired by recent works (Chari et al., 2019), which I believe could help both weak supervision and interpretable AI.

Increasing interpretability. Machine learning is yet often seen as a black box and while transparent AI is a long term goal (Arrieta et al., 2020), interpretability appears as a reasonable next step for human to understand (to some extent) the output of AI machines. This appears to me an important investigation direction for safe and ethical AI. On scene understanding from vision, this means enforcing physical consistency in the learning process, and physics realism in the algorithms outputs. We are currently pursuing our work on physics guided learning for vision in adverse conditions.

On a general note, I wish to diversify my field of applications to more general vision – expanding to new topics and types of sceneries, and to strengthen open and reproducible research. As a wish list, I hope to open stronger bounds with other groups – as I find collaborative research exciting and inspiring – while developing collaboration with researchers from other fields (eg. physics, ethnology, archaeology, anthropology, etc.).

Part II

Scientific career

Professional

Since my PhD started, I worked in 4 research laboratories, of which two were abroad, and experienced working in a private company:

- Since 2020. Permanent researcher at Inria, RITS team (France). I lead a small group on vision for scene understanding.
- 2015-2020. Post-Doc at Inria, RITS team (France). Following a period where I took over applied projects on autonomous driving, I then led the computer vision group in the team.
- 2015. Developer/CTO of the startup Design Your Cube (France). Working on web 3D interactive tools.
- 2014. Post-Doc at the University of Makedonia (Greece). Short position on 3D reconstruction of revolving pottery object from depth maps.
- 2013-2014. Post-Doc in Robotics Centre of Mines ParisTech (France). I worked on skeleton and object estimation from depth map data.
- 2012. PhD from Mines ParisTech, Robotics Centre (France) in “Informatique temps-réel, robotique et automatique”. PhD topic: “Vision algorithms for Rain and Traffic Lights in Driver Assistance Systems”.
- 2011. PhD visit at Carnegie Mellon University, ILIM Lab (USA) on fast-reactive illumination through rain and snow.

Associate positions. Since 2019 I am also Associate Professor of Université Laval, Canada.

Supervision and Teaching activities

Contents

7.1 Supervision	91
7.2 Teaching activities	92

7.1 Supervision

In the 9 years since my PhD defense in 2012, I fully co-supervised 5 PhDs (3 are ongoing), 1 Post-Doc, 1 engineer, 18 interns (inc. 3 PhDs interns).

The 2 PhD theses already defended are:

- 2017-2021. **3D Scene Reconstruction and Completion for Autonomous Driving**, Luis Roldão Jimenez.
CIFRE with Akka. Co-supervised with Anne Verroust-Blondet.
Publications: (Roldão et al., 2018, 2019, 2020, 2021)
- 2017-2020. **2D-3D scene understanding for autonomous driving**, Maximilian Jaritz.
CIFRE with Valeo/Valeo.ai. Co-supervised with Fawzi Nashashibi.
Publications: (Perot et al., 2017; Jaritz et al., 2018a,b, 2020, 2021)

The 3 ongoing PhD theses are:

- 2019-2022. **Style transfer and domain adaptation for semantic segmentation**, Fabio Pizzati.
CIFRE with Vislab Ambarella.
Publications: (Pizzati et al., 2020b,a, 2021a,b,c)
- 2020-2023. **3D Semantic Scene Reconstruction and Completion from 2D Image**, Anh Quan Cao.
Publications: (Cao and de Charette, 2021)
- 2021-2024. **Physic-guided learning for vision in adverse weather conditions**, Ivan Lopes.

7.2 Teaching activities

Aside from research, I have been involved in the following teaching activities:

- 2018-2021. Computer Vision for Scene Understanding, *Master Artificial Intelligence and Movement (AI-Move)*, Mines ParisTech, Paris, France. Audience: Master, English.
- 2017. Computer Vision for Autonomous Driving. *Master of Engineering, Universidad Simon Bolivar (USB)*, Caracas, Venezuela. Audience: Master, English.
- 2018-2019. Artificial Intelligence. *Puck & Ribambelle*, Montpellier, France. Audience: primary school, French.
- 2017. Signal Processing with Python. *Paris Science et Lettre (PSL)*, Paris, France. Audience: PhDs, English.
- 2014. Introduction to Computer Vision. *University of Makedonia (UOM)*, Thessaloniki, Greece. Audience: Pro. Master, English.

Dissemination

Contents

8.1	Dissemination	93
8.1.1	Popularization	93
8.1.2	Awards	94
8.2	Grants and Research projects	94
8.3	Publications	95
8.3.1	Journal with peered reviews	95
8.3.2	Conferences with peered reviews	95
8.3.3	Scientific communications	97

8.1 Dissemination

The research I supervised was published in peer-reviewed conferences and journals listed in Sec. 8.3.

In the last years I also promoted open research and – when legal framework allowed it – recent works are openly distributed.

8.1.1 Popularization

I have an important activity on the popularization of science which accounts for talks, press releases, and involvements in associations. I’m brushing the most important actions here.

On press targeting a general audience I was interviewed in roughly 25 press releases (magazine, web, radio, TVs), plus over 60 on our research on *illumination through rain* (CMU, 2011) which gained large media coverage¹.

On young audience targets, I was mostly involved with: Arbre des connaissances (2017-2018) for popularizing science in secondary schools and encouraging girls in science, Puck et Ribambelle primary school (2018-2019) to introduce artificial intelligence to young kids.

¹<https://cs.cmu.edu/~ILIM/projects/IL/smartHeadlight/index2.html#press>

8.1.2 Awards

- Outstanding reviewer CVPR 2021
- ICCP 2012, Best honorable paper award. (de Charette et al., 2012) de Charette, R., Tamburo, R., Barnum, P. C., Rowe, A., Kanade, T., and Narasimhan, S. G. (2012). Fast reactive control for illumination through rain and snow. In *ICCP*

8.2 Grants and Research projects

Some of the researches presented were funded by the following grants I obtained:

- 2021-2024. **ANR JCJC SIGHT** investigates invariant algorithms for complex weather conditions (rain, snow, hail). The project leverages un-/self-supervised algorithms with physic-guidance to model physically realistic weather, and learn weather-invariant representations.
- 2018-2022. **Samuel de Champlain** is a collaborative grant with J-F. Lalonde (Uni. Laval, Canada) on computer vision in non homogeneous lighting conditions and fine modeling of dynamic scenes. Up to now, it led to 4 co-supervisions, 2 seminars, 5 visits, 4 co-publications.

I was also involved, scientifically or administratively, in the following projects:




- 2020-2022. **PIA project SAMBA** seeks to improve the safety of autonomous driving. We study here how to boost 3D scene understanding leveraging sparse 3D or dense 2D sensing.
- 2016-2021. **FUI PACV2x** is an applied project on augmented perception via communication for cooperative driving. It focuses on complex interaction scenarios: highway merging, overtaking, intersections, etc.
- 2013-2017. **H2020 i-Treasures** studies the intangible cultural heritage and how to learn the rare know-how of living. It included the reconstruction of fast-evolving 3D objects like pottery object.
- 2014-2017. **ECOS Nord** is a collaborative grant with a broad spectrum. It eased the exchanges and collaborations between Inria Paris and Universidad Simon Bolivar, Venezuela.
- 2011-2016. **H2020 Furbot** goal is to build an autonomous freight urban robotic vehicle for the ‘last mile problem’ in controlled conditions. It covers object detection, vehicle planning and control.
- 2011-2016. **ANR CAMPUS** seeks to improve the perception of the environment with 3D vision sensor, for autonomous driving.

- 2011-2016. **H2020 AutoNet2030** is centered around cooperative systems in support of networked automated driving by 2030.
- 2009-2012. **ICADAC** targets Improved Camera based Detection under Adverse Conditions.
- 2008-2011. **Intersafe2** aims at crossing intersection safely with autonomous driving.



8.3 Publications








Legend:  = opensource,  = dataset shared.

8.3.1 Journal with peered reviews



- (Pizzati et al., 2021b) Pizzati, F., Cerri, P., and de Charette, R. (2021b). Guided disentanglement in generative networks. *arXiv* (submitted to IJCV)
- (Roldão et al., 2021) Roldão, L., de Charette, R., and Verroust-Blondet, A. (2021). 3D semantic scene completion: a survey. *IJCV*
- (Jaritz et al., 2021) Jaritz, M., Vu, T.-H., de Charette, R., Wirbel, É., and Pérez, P. (2021). Cross-modal learning for domain adaptation in 3D semantic segmentation. *arXiv* (submitted to PAMI) 
- (Tremblay et al., 2020) Tremblay, M., Halder, S. S., de Charette, R., and Lalonde, J.-F. (2020). Rain rendering for evaluating and improving robustness to bad weather. *IJCV*  
- (Flores et al., 2018) Flores, C., Merdrignac, P., de Charette, R., Navas, F., Milanés, V., and Nashashibi, F. (2018). A cooperative car-following/emergency braking system with prediction-based pedestrian avoidance capabilities. *IEEE T-ITS*

8.3.2 Conferences with peered reviews

- (Pizzati et al., 2021c) Pizzati, F., Lalonde, J.-F., and de Charette, R. (2021c). Manifest: Manifold deformation for few-shot image translation. *arXiv (submitted)* 
- (Cao and de Charette, 2021) Cao, A.-Q. and de Charette, R. (2021). MonoScene: Monocular 3d semantic scene completion. *arXiv (submitted)* 
- (Dell’Eva et al., 2021) Dell’Eva, A., Pizzati, F., Bertozzi, M., and de Charette, R. (2021). Leveraging local domains for image-to-image translation. In *VISAPP*

- (Pizzati et al., 2021a) Pizzati, F., Cerri, P., and de Charette, R. (2021a). CoMoGAN: continuous model-guided image-to-image translation. In *CVPR oral* 
- (Roldão et al., 2020) Roldão, L., de Charette, R., and Verroust-Blondet, A. (2020). LMSCNet: Lightweight multiscale 3D semantic completion. In *3DV oral* 
- (Dubeau et al., 2020) Dubeau, E., Garon, M., Debaque, B., de Charette, R., and Lalonde, J.-F. (2020). RGB-D-E: Event camera calibration for fast 6-dof object tracking. In *ISMAR*  
- (Pizzati et al., 2020a) Pizzati, F., Cerri, P., and de Charette, R. (2020a). Model-based occlusion disentanglement for image-to-image translation. In *ECCV*
- (Jaritz et al., 2020) Jaritz, M., Vu, T.-H., Charette, R. d., Wirbel, E., and Pérez, P. (2020). xMUDA: Cross-modal unsupervised domain adaptation for 3D semantic segmentation. In *CVPR* 
- (Pizzati et al., 2020b) Pizzati, F., Charette, R. d., Zaccaria, M., and Cerri, P. (2020b). Domain bridge for unpaired image-to-image translation and unsupervised domain adaptation. In *WACV*
- (Roldão et al., 2019) Roldão, L., de Charette, R., and Verroust-Blondet, A. (2019). 3D surface reconstruction from voxel-based lidar data. In *ITSC*
- (Halder et al., 2019) Halder, S. S., Lalonde, J.-F., and Charette, R. d. (2019). Physics-based rendering for improving robustness to rain. In *ICCV*  
- (Nguyen et al., 2018) Nguyen, D.-V., de Charette, R., Nashashibi, F., Dao, T.-K., and Castelli, E. (2018). Wifi fingerprinting localization for intelligent vehicles in car park. In *IPIN*
- (Jaritz et al., 2018b) Jaritz, M., de Charette, R., Wirbel, E., Perrotton, X., and Nashashibi, F. (2018b). Sparse and dense data with cnns: Depth completion and semantic segmentation. In *3DV*
- (Jaritz et al., 2018a) Jaritz, M., de Charette, R., Toromanoff, M., Perot, E., and Nashashibi, F. (2018a). End-to-end race driving with deep reinforcement learning. In *ICRA*
- (Perot et al., 2017) Perot, E., Jaritz, M., Toromanoff, M., and de Charette, R. (2017). End-to-end driving in a realistic racing game with deep reinforcement learning. In *CVPR Workshops*
- (Dapogny et al., 2013) Dapogny, A., de Charette, R., Manitsaris, S., Moutarde, F., and Glushkova, A. (2013). Towards a hand skeletal model for depth images applied to capture music-like finger gestures.

In *CMMR*

- (de Charette et al., 2012) de Charette, R., Tamburo, R., Barnum, P. C., Rowe, A., Kanade, T., and Narasimhan, S. G. (2012). Fast reactive control for illumination through rain and snow. In *ICCP* **best honorable paper** 
- (Nashashibi et al., 2010) Nashashibi, F., de Charrette, R., and Lia, A. (2010). Detection of unfocused raindrops on a windscreen using low level image processing. In *ICARCV*
- (de Charette and Nashashibi, 2009b) de Charette, R. and Nashashibi, F. (2009b). Traffic light recognition using image processing compared to learning processes. In *IROS*
- (de Charette and Nashashibi, 2009a) de Charette, R. and Nashashibi, F. (2009a). Real time visual traffic lights recognition based on spot light detection and adaptive traffic lights templates. In *IV* 

8.3.3 Scientific communications

- (de Charette and Manitsaris, 2019) de Charette, R. and Manitsaris, S. (2019). 3D reconstruction of deformable revolving object under heavy hand interaction. *arXiv* (journal submission)
- (Agarwal et al., 2021) Agarwal, P., de Beaucorps, P., and de Charette, R. (2021). Sparse curriculum reinforcement learning for end-to-end driving. *arXiv*
- (Roldão et al., 2018) Roldão, L., de Charette, R., and Verroust-Blondet, A. (2018). A statistical update of grid representations from range sensors. *arXiv*
- (Meyer and de Charette, 2016) Meyer, A. and de Charette, R. (2016). Computing ego velocity from scene flow estimation
- (Kahraman and de Charette, 2017) Kahraman, S. and de Charette, R. (2017). Influence of fog on computer vision algorithms

Bibliography

- Agarwal, P., de Beaucorps, P., and de Charette, R. (2021). Sparse curriculum reinforcement learning for end-to-end driving. *arXiv*.
- Armeni, I., Sax, S., Zamir, A., and Savarese, S. (2017). Joint 2D-3D-semantic data for indoor scene understanding. *arXiv*.
- Arrieta, A. B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., García, S., Gil-López, S., Molina, D., Benjamins, R., et al. (2020). Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information Fusion*.
- Arulkumaran, K., Deisenroth, M. P., Brundage, M., and Bharath, A. A. (2017). A brief survey of deep reinforcement learning. *arXiv*.
- Behley, J., Garbade, M., Milioto, A., Quenzel, J., Behnke, S., Stachniss, C., and Gall, J. (2019). Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *ICCV*.
- Berger, M., Tagliasacchi, A., Seversky, L. M., Alliez, P., Guennebaud, G., Levine, J. A., Sharf, A., and Silva, C. T. (2017). A survey of surface reconstruction from point clouds. In *Computer Graphics Forum*.
- Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L. D., Monfort, M., Muller, U., Zhang, J., et al. (2016). End to end learning for self-driving cars. *arXiv*.
- Bouchiba, H., Santoso, S., Deschaud, J.-E., Rocha-Da-Silva, L., Goulette, F., and Coupez, T. (2020). Computational fluid dynamics on 3D point set surfaces. *Journal of Computational Physics*.
- Bouman, K. L., Ye, V., Yedidia, A. B., Durand, F., Wornell, G. W., Torralba, A., and Freeman, W. T. (2017). Turning corners into cameras: Principles and methods. In *ICCV*.
- Caesar, H., Bankiti, V., Lang, A. H., Vora, S., Liong, V. E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., and Beijbom, O. (2020a). nuScenes: A multimodal dataset for autonomous driving. *CVPR*.
- Caesar, H., Bankiti, V., Lang, A. H., Vora, S., Liong, V. E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., and Beijbom, O. (2020b). nuScenes: A multimodal dataset for autonomous driving. *CVPR*.
- Cai, Y., Chen, X., Zhang, C., Lin, K.-Y., Wang, X., and Li, H. (2021). Semantic scene completion via integrating instances and scene in-the-loop. In *CVPR*.
- Candy, J. (2007). Bootstrap particle filtering. *Signal Processing Magazine*.
- Cao, A.-Q. and de Charette, R. (2021). MonoScene: Monocular 3d semantic scene completion. *arXiv (submitted)*.
- Catmull, E. and Rom, R. (1974). A class of local interpolating splines. *Computer aided geometric design*.
- Chang, A. X., Dai, A., Funkhouser, T. A., Halber, M., Nießner, M., Savva, M., Song, S., Zeng, A., and Zhang, Y. (2017). Matterport3D: Learning from RGB-D data in indoor environments. In *3DV*.

- Chari, P., Talegaonkar, C., Ba, Y., and Kadambi, A. (2019). Visual physics: Discovering physical laws from videos. *arXiv*.
- Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. (2018a). DeepLab: Semantic image segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE TPAMI*.
- Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., and Adam, H. (2018b). Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*.
- Chen, R., Huang, Z., and Yu, Y. (2019a). Am2fnet: Attention-based multiscale & multi-modality fused network. *ROBIO*.
- Chen, X., Lin, K.-Y., Qian, C., Zeng, G., and Li, H. (2020a). 3D sketch-aware semantic scene completion via semi-supervised structure prior. In *CVPR*.
- Chen, X., Xing, Y., and Zeng, G. (2020b). Real-time semantic scene completion via feature aggregation and conditioned prediction. In *ICIP*.
- Chen, Y., Garbade, M., and Gall, J. (2019b). 3D semantic scene completion from a single depth image using adversarial training. In *ICIP*.
- Cheng, R., Agia, C., Ren, Y., Li, X., and Bingbing, L. (2020). S3CNet: A sparse semantic scene completion network for LiDAR point clouds. In *CoRL*.
- Cherabier, I., Schönberger, J. L., Oswald, M., Pollefeys, M., and Geiger, A. (2018). Learning priors for semantic 3D reconstruction. In *ECCV*.
- Choi, Y., Uh, Y., Yoo, J., and Ha, J.-W. (2020). Stargan v2: Diverse image synthesis for multiple domains. In *CVPR*.
- Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., and Schiele, B. (2016). The cityscapes dataset for semantic urban scene understanding. In *CVPR*.
- Crivellaro, A., Rad, M., Verdie, Y., Moo Yi, K., Fua, P., and Lepetit, V. (2015). A novel representation of parts for accurate 3d object detection and tracking in monocular images. In *ICCV*.
- Dai, A., Chang, A. X., Savva, M., Halber, M., Funkhouser, T. A., and Nießner, M. (2017). ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In *CVPR*.
- Dai, A., Ritchie, D., Bokeloh, M., Reed, S., Sturm, J., and Nießner, M. (2018). ScanComplete: Large-scale scene completion and semantic segmentation for 3D scans. In *CVPR*.
- Dapogny, A., de Charette, R., Manitsaris, S., Moutarde, F., and Glushkova, A. (2013). Towards a hand skeletal model for depth images applied to capture music-like finger gestures. In *CMMR*.
- Dasagi, V., Bruce, J., Peynot, T., and Leitner, J. (2019). Ctrl-z: Recovering from instability in reinforcement learning. *arXiv*.
- de Charette, R. (2012). *Vision Algorithms for Rain and Traffic Lights in Driver Assistance Systems*. PhD thesis, Ecole Nationale Supérieure des Mines de Paris.
- de Charette, R. and Manitsaris, S. (2019). 3D reconstruction of deformable revolving object under heavy hand interaction. *arXiv*.

- de Charette, R. and Nashashibi, F. (2009a). Real time visual traffic lights recognition based on spot light detection and adaptive traffic lights templates. In *IV*.
- de Charette, R. and Nashashibi, F. (2009b). Traffic light recognition using image processing compared to learning processes. In *IROS*.
- de Charette, R., Tamburo, R., Barnum, P. C., Rowe, A., Kanade, T., and Narasimhan, S. G. (2012). Fast reactive control for illumination through rain and snow. In *ICCP*.
- Dell’Eva, A., Pizzati, F., Bertozzi, M., and de Charette, R. (2021). Leveraging local domains for image-to-image translation. In *VISAPP*.
- Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., and Koltun, V. (2017). CARLA: An open urban driving simulator. In *CoRL*.
- Dourado, A., de Campos, T. E., Kim, H. S., and Hilton, A. (2020a). EdgeNet: Semantic scene completion from RGB-D images. *ICPR*.
- Dourado, A., Kim, H., de Campos, T. E., and Hilton, A. (2020b). Semantic scene completion from a single 360-Degree image and depth map. In *VISIGRAPP*.
- Dubeau, E., Garon, M., Debaque, B., de Charette, R., and Lalonde, J.-F. (2020). RGB-D-E: Event camera calibration for fast 6-dof object tracking. In *ISMAR*.
- Dulac-Arnold, G., Mankowitz, D., and Hester, T. (2019). Challenges of real-world reinforcement learning. *arXiv*.
- Firman, M., Aodha, O. M., Julier, S. J., and Brostow, G. J. (2016). Structured prediction of unobserved voxels from a single depth image. In *CVPR*.
- Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*.
- Flores, C., Merdrignac, P., de Charette, R., Navas, F., Milanés, V., and Nashashibi, F. (2018). A cooperative car-following/emergency braking system with prediction-based pedestrian avoidance capabilities. *IEEE T-ITS*.
- Gaidon, A., Wang, Q., Cabon, Y., and Vig, E. (2016). Virtual worlds as proxy for multi-object tracking analysis. In *CVPR*.
- Garbade, M., Sawatzky, J., Richard, A., and Gall, J. (2019). Two stream 3D semantic scene completion. In *CVPR Workshops*.
- Garg, K. and Nayar, S. K. (2006). Photorealistic rendering of rain streaks. *ACM TOG*.
- Garg, K. and Nayar, S. K. (2007). Vision and rain. *IJCV*.
- Garon, M., Laurendeau, D., and Lalonde, J.-F. (2018). A framework for evaluating 6-DOF object trackers. In *ECCV*.
- Gehrig, D., Loquercio, A., Derpanis, K., and Scaramuzza, D. (2019). End-to-end learning of representations for asynchronous event-based data. *ICCV*.
- Geiger, A., Lenz, P., Stiller, C., and Urtasun, R. (2013). Vision meets robotics: The kitti dataset. *IJRR*.

- Godard, C., Mac Aodha, O., Firman, M., and Brostow, G. J. (2019). Digging into self-supervised monocular depth estimation. In *ICCV*.
- Gong, R., Li, W., Chen, Y., and Gool, L. V. (2019). Dlow: Domain flow for adaptation and generalization. In *CVPR*.
- Graham, B., Engelcke, M., and van der Maaten, L. (2018). 3D semantic segmentation with submanifold sparse convolutional networks. *CVPR*.
- Guedes, A. B. S., de Campos, T. E., and Hilton, A. (2018). Semantic scene completion combining colour and depth: preliminary experiments. *arXiv*.
- Guo, Y.-X. and Tong, X. (2018). View-volume network for semantic scene completion from a single depth image. In *IJCAI*.
- Gustavson, S. (2005). Simplex noise demystified.
- Halder, S. S., Lalonde, J.-F., and Charette, R. d. (2019). Physics-based rendering for improving robustness to rain. In *ICCV*.
- Hansen, N., Müller, S. D., and Koumoutsakos, P. (2003). Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (cma-es). *Evolutionary computation*.
- He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *CVPR*.
- Hinton, G., Vinyals, O., and Dean, J. (2014). Distilling the knowledge in a neural network. In *NIPS Workshops*.
- Hold-Geoffroy, Y., Athawale, A., and Lalonde, J.-F. (2019). Deep sky modeling for single image outdoor lighting estimation. In *CVPR*.
- Hornung, A., Wurm, K. M., Bennewitz, M., Stachniss, C., and Burgard, W. (2013). OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*.
- Huang, X. and Belongie, S. (2017). Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*.
- Huang, X., Liu, M.-Y., Belongie, S., and Kautz, J. (2018a). Multimodal unsupervised image-to-image translation. In *ECCV*.
- Huang, Z., Fan, J., Cheng, S., Yi, S., Wang, X., and Li, H. (2018b). Hms-net: Hierarchical multi-scale sparsity-invariant network for sparse depth completion. *arXiv*.
- Jaritz, M., de Charette, R., Toromanoff, M., Perot, E., and Nashashibi, F. (2018a). End-to-end race driving with deep reinforcement learning. In *ICRA*.
- Jaritz, M., de Charette, R., Wirbel, E., Perrotton, X., and Nashashibi, F. (2018b). Sparse and dense data with cnns: Depth completion and semantic segmentation. In *3DV*.
- Jaritz, M., Vu, T.-H., Charette, R. d., Wirbel, E., and Pérez, P. (2020). xMUDA: Cross-modal unsupervised domain adaptation for 3D semantic segmentation. In *CVPR*.
- Jaritz, M., Vu, T.-H., de Charette, R., Wirbel, É., and Pérez, P. (2021). Cross-modal learning for domain adaptation in 3D semantic segmentation. *arXiv*.

- Kaelbling, L. P., Littman, M. L., and Moore, A. W. (1996). Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*.
- Kahraman, S. and de Charette, R. (2017). Influence of fog on computer vision algorithms.
- Kendall, A., Hawke, J., Janz, D., Mazur, P., Reda, D., Allen, J.-M., Lam, V.-D., Bewley, A., and Shah, A. (2019). Learning to drive in a day. In *ICRA*.
- Keskin, C., Kırac, F., Kara, Y. E., and Akarun, L. (2011). Real time hand pose estimation using depth sensors. In *ICCV Workshops*.
- Kiran, B. R., Sobh, I., Talpaert, V., Mannion, P., Al Sallab, A. A., Yogamani, S., and Pérez, P. (2021). Deep reinforcement learning for autonomous driving: A survey. *IEEE T-ITS*.
- Knox, W. B., Allievi, A., Banzhaf, H., Schmitt, F., and Stone, P. (2021). Reward (mis) design for autonomous driving. *arXiv*.
- Koenderink, J. J., van Doorn, A. J., and Kappers, A. M. (1995). Depth relief. *Perception*.
- Kolluri, R. (2005). Provably good moving least squares. In *SIGGRAPH*.
- Ku, J., Harakeh, A., and Waslander, S. L. (2018). In defense of classical image processing: Fast depth completion on the cpu. In *Conference on Computer and Robot Vision*.
- Kylo-tonn (2016). World rally championship 6 (wrc 6).
- Lebeda, K., Matas, J., and Chum, O. (2012). Fixing the locally optimized ransac—full experimental evaluation. In *BMVC*.
- Lee, D.-H. et al. (2013). Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*.
- Lepetit, V. and Fua, P. (2005). *Monocular model-based 3D tracking of rigid objects*. Now Publishers Inc.
- Li, J., Han, K., Wang, P., Liu, Y., and Yuan, X. (2020a). Anisotropic convolutional networks for 3D semantic scene completion. In *CVPR*.
- Li, J., Liu, Y., Gong, D., Shi, Q., Yuan, X., Zhao, C., and Reid, I. D. (2019a). RGBD based dimensional decomposition residual network for 3D semantic scene completion. In *CVPR*.
- Li, J., Liu, Y. W., Yuan, X., Zhao, C., Siegart, R., Reid, I., and Cadena, C. (2020b). Depth based semantic scene completion with position importance aware loss. *Robotics and Automation Letters (RA-L)*.
- Li, S., Zou, C., Li, Y., Zhao, X., and Gao, Y. (2020c). Attention-based multi-modal fusion network for semantic scene completion. In *AAAI*.
- Li, Y., Wang, G., Ji, X., Xiang, Y., and Fox, D. (2018). Deepim: Deep iterative matching for 6d pose estimation. In *ECCV*.
- Li, Y., Yuan, L., and Vasconcelos, N. (2019b). Bidirectional learning for domain adaptation of semantic segmentation. In *CVPR*.

- Liao, Y., Xie, J., and Geiger, A. (2021). KITTI-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. *arXiv*.
- Liu, M.-Y., Huang, X., Mallya, A., Karras, T., Aila, T., Lehtinen, J., and Kautz, J. (2019a). Few-shot unsupervised image-to-image translation. In *ICCV*.
- Liu, S., Hu, Y., Zeng, Y., Tang, Q., Jin, B., Han, Y., and Li, X. (2018). See and think: Disentangling semantic scene completion. In *NeurIPS*.
- Liu, X., Suganuma, M., Sun, Z., and Okatani, T. (2019b). Dual residual networks leveraging the potential of paired operations for image restoration. In *CVPR*.
- Liu, Y. W., Li, J., Yan, Q., Yuan, X., Zhao, C.-X., Reid, I., and Cadena, C. (2020). 3D gated recurrent fusion for semantic scene completion. *arXiv*.
- Ma, L., Jia, X., Georgoulis, S., Tuytelaars, T., and Van Gool, L. (2019). Exemplar guided unsupervised image-to-image translation with semantic consistency. In *ICLR*.
- Manhardt, F., Kehl, W., Navab, N., and Tombari, F. (2018). Deep model-based 6d pose refinement in rgb. In *ECCV*.
- Mehta, S., Rastegari, M., Caspi, A., Shapiro, L., and Hajishirzi, H. (2018). Espnet: Efficient spatial pyramid of dilated convolutions for semantic segmentation. In *ECCV*.
- Meyer, A. and de Charette, R. (2016). Computing ego velocity from scene flow estimation.
- Miller, T. (2019). Explanation in artificial intelligence: Insights from the social sciences. *Artificial intelligence*.
- Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T. P., Harley, T., Silver, D., and Kavukcuoglu, K. (2016). Asynchronous methods for deep reinforcement learning. In *ICML*.
- Morerio, P., Cavazza, J., and Murino, V. (2018). Minimal-entropy correlation alignment for unsupervised deep domain adaptation. In *ICLR*.
- Narasimhan, S. G. and Nayar, S. K. (2002). Vision and the atmosphere. *IJCV*.
- Nashashibi, F., de Charrette, R., and Lia, A. (2010). Detection of unfocused raindrops on a windscreen using low level image processing. In *ICARCV*.
- Nguyen, D.-V., de Charette, R., Nashashibi, F., Dao, T.-K., and Castelli, E. (2018). Wifi fingerprinting localization for intelligent vehicles in car park. In *IPIN*.
- Östlund, J., Varol, A., Ngo, D. T., and Fua, P. (2012). Laplacian meshes for monocular 3d shape recovery. In *ECCV*.
- Pan, X., Shi, J., Luo, P., Wang, X., and Tang, X. (2017). Spatial as deep: Spatial cnn for traffic scene understanding. In *AAAI*.
- Peng, D., Lei, Y., Li, W., Zhang, P., and Guo, Y. (2021). Sparse-to-dense feature matching: Intra and inter domain cross-modal learning in domain adaptation for 3d semantic segmentation. In *ICCV*.
- Perot, E., Jaritz, M., Toromanoff, M., and de Charette, R. (2017). End-to-end driving in a realistic racing game with deep reinforcement learning. In *CVPR Workshops*.

- Pizzati, F., Cerri, P., and de Charette, R. (2020a). Model-based occlusion disentanglement for image-to-image translation. In *ECCV*.
- Pizzati, F., Cerri, P., and de Charette, R. (2021a). CoMoGAN: continuous model-guided image-to-image translation. In *CVPR*.
- Pizzati, F., Cerri, P., and de Charette, R. (2021b). Guided disentanglement in generative networks. *arXiv*.
- Pizzati, F., Charette, R. d., Zaccaria, M., and Cerri, P. (2020b). Domain bridge for unpaired image-to-image translation and unsupervised domain adaptation. In *WACV*.
- Pizzati, F., Lalonde, J.-F., and de Charette, R. (2021c). Manifest: Manifold deformation for few-shot image translation. *arXiv (submitted)*.
- Pomerleau, D. A. (1989). Alvin: An autonomous land vehicle in a neural network. In *NeurIPS*.
- Pomerleau, F., Colas, F., Siegwart, R., and Magnenat, S. (2013). Comparing ICP variants on real-world data sets. *Autonomous Robots*.
- Popov, S., Bauszat, P., and Ferrari, V. (2020). Corenet: Coherent 3d scene reconstruction from a single rgb image. In *ECCV*.
- Porav, H., Bruls, T., and Newman, P. (2019). I can see clearly now: Image restoration via de-raining. In *ICRA*.
- Rausch, V., Hansen, A., Solowjow, E., Liu, C., Kreuzer, E., and Hedrick, J. K. (2017). Learning a deep neural net policy for end-to-end control of autonomous vehicles. In *American Control Conference (ACC)*.
- Rebecq, H., Gehrig, D., and Scaramuzza, D. (2018). ESIM: an open event camera simulator. In *CoRL*.
- Redmon, J. and Farhadi, A. (2017). Yolo9000: better, faster, stronger. In *CVPR*.
- Ren, M., Pokrovsky, A., Yang, B., and Urtasun, R. (2018). Sbnnet: Sparse blocks network for fast inference. In *CVPR*.
- Richter, S. R., Hayder, Z., and Koltun, V. (2017). Playing for benchmarks. In *ICCV*.
- Rist, C. B., Emmerichs, D., Enzweiler, M., and Gavrilu, D. M. (2020a). Semantic scene completion using local deep implicit functions on LiDAR data. *arXiv*.
- Rist, C. B., Schmidt, D., Enzweiler, M., and Gavrilu, D. M. (2020b). SCSSnet: Learning spatially-conditioned scene segmentation on LiDAR point clouds. In *IV*.
- Roldão, L., de Charette, R., and Verroust-Blondet, A. (2018). A statistical update of grid representations from range sensors. *arXiv*.
- Roldão, L., de Charette, R., and Verroust-Blondet, A. (2019). 3D surface reconstruction from voxel-based lidar data. In *ITSC*.
- Roldão, L., de Charette, R., and Verroust-Blondet, A. (2020). LMSCNet: Lightweight multiscale 3D semantic completion. In *3DV*.
- Roldao, L., de Charette, R., and Verroust-Blondet, A. (2020). LMSCNet: Lightweight multiscale 3D semantic completion. In *3DV*.

- Roldão, L., de Charette, R., and Verroust-Blondet, A. (2021). 3D semantic scene completion: a survey. *IJCV*.
- Ros, G., Sellart, L., Materzynska, J., Vazquez, D., and Lopez, A. M. (2016). The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *CVPR*.
- Saito, K., Saenko, K., and Liu, M.-Y. (2020). COCO-FUNIT: Few-shot unsupervised image translation with a content conditioned style encoder. In *ECCV*.
- Sakaridis, C., Dai, D., and Gool, L. V. (2021). Acdc: The adverse conditions dataset with correspondences for semantic driving scene understanding. In *ICCV*.
- Sakaridis, C., Dai, D., and Van Gool, L. (2020). Map-guided curriculum domain adaptation and uncertainty-aware evaluation for semantic nighttime image segmentation. *IEEE TPAMI*.
- Salzmann, M. and Fua, P. (2009). Reconstructing sharply folding surfaces: A convex formulation. In *CVPR*.
- Schmidt, L. M., Kontes, G., Plinge, A., and Mutschler, C. (2021). Can you trust your autonomous car? interpretable and verifiably safe reinforcement learning. In *IV*.
- Schröder, M., Maycock, J., Ritter, H., and Botsch, M. (2014). Real-time hand tracking using synergistic inverse kinematics. In *ICRA*.
- Schulman, J., Lee, A., Ho, J., and Abbeel, P. (2013). Tracking deformable objects with point clouds. In *ICRA*.
- Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv*.
- Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., and Blake, A. (2011). Real-time human pose recognition in parts from single depth images. In *CVPR*.
- Silberman, N., Hoiem, D., Kohli, P., and Fergus, R. (2012a). Indoor segmentation and support inference from rgb-d images. In *ECCV*.
- Silberman, N., Hoiem, D., Kohli, P., and Fergus, R. (2012b). Indoor segmentation and support inference from RGBD images. In *ECCV*.
- Sinha, P. and Adelson, E. (1993). Verifying the 'consistency' of shading patterns and 3-d structures. In *Workshop on Qualitative Vision*.
- Smeulders, A. W., Chu, D. M., Cucchiara, R., Calderara, S., Dehghan, A., and Shah, M. (2013). Visual tracking: An experimental survey. *IEEE TPAMI*.
- Song, S., Yu, F., Zeng, A., Chang, A. X., Savva, M., and Funkhouser, T. (2017a). Semantic scene completion from a single depth image. *CVPR*.
- Song, S., Yu, F., Zeng, A., Chang, A. X., Savva, M., and Funkhouser, T. A. (2017b). Semantic scene completion from a single depth image. In *CVPR*.
- Strudel, R., Pashevich, A., Kalevatykh, I., Laptev, I., Sivic, J., and Schmid, C. (2020). Learning to combine primitive skills: A step towards versatile robotic manipulation. In *ICRA*.
- Sun, P., Kretschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., et al. (2020). Scalability in perception for autonomous driving: Waymo open dataset. In *CVPR*.

- Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *CVPR*.
- Tamburo, R., Nurvitadhi, E., Chugh, A., Chen, M., Rowe, A., Kanade, T., and Narasimhan, S. G. (2014). Programmable automotive headlights. In *ECCV*.
- Torr, P. and Zisserman, A. (2000). MLESAC: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*.
- Tremblay, M., Halder, S. S., de Charette, R., and Lalonde, J.-F. (2020). Rain rendering for evaluating and improving robustness to bad weather. *IJCV*.
- Tsai, Y.-H., Hung, W.-C., Schuster, S., Sohn, K., Yang, M.-H., and Chandraker, M. (2018). Learning to adapt structured output space for semantic segmentation. In *CVPR*.
- TuSimple, A. (2017). Tusimple benchmark. [urlhttps://github.com/TuSimple/tusimple-benchmark](https://github.com/TuSimple/tusimple-benchmark).
- Uhrig, J., Schneider, N., Schneider, L., Franke, U., Brox, T., and Geiger, A. (2017). Sparsity invariant cnns. In *3DV*.
- Ulyanov, D., Vedaldi, A., and Lempitsky, V. (2017). Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis. In *CVPR*.
- Varma, G., Subramanian, A., Namboodiri, A., Chandraker, M., and Jawahar, C. V. (2019). Idd: A dataset for exploring problems of autonomous navigation in unconstrained environments. In *WACV*.
- Vasiljevic, I., Kolkin, N., Zhang, S., Luo, R., Wang, H., Dai, F. Z., Daniele, A. F., Mostajabi, M., Basart, S., Walter, M. R., and Shakhnarovich, G. (2019). DIODE: A Dense Indoor and Outdoor DEpth Dataset. *CoRR*.
- Vecerik, M., Hester, T., Scholz, J., Wang, F., Pietquin, O., Piot, B., Heess, N., Rothörl, T., Lampe, T., and Riedmiller, M. (2017). Leveraging demonstrations for deep reinforcement learning on robotics problems with sparse rewards. *arXiv*.
- Vicente, S. and Agapito, L. (2013). Balloon shapes: Reconstructing and deforming objects with volume from images. In *3DV*.
- Vu, T.-H., Jain, H., Bucher, M., Cord, M., and Pérez, P. (2019). Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In *CVPR*.
- Wang, J., Sun, K., Cheng, T., Jiang, B., Deng, C., Zhao, Y., Liu, D., Mu, Y., Tan, M., Wang, X., Liu, W., and Xiao, B. (2019a). Deep high-resolution representation learning for visual recognition. *IEEE TPAMI*.
- Wang, P.-S., Liu, Y., and Tong, X. (2020). Deep Octree-based CNNs with output-guided skip connections for 3D shape and scene completion. In *CVPR Workshops*.
- Wang, W., Pottmann, H., and Liu, Y. (2006). Fitting B-spline curves to point clouds by curvature-based squared distance minimization. *ACM TOG*.
- Wang, X., Oswald, M., Cherabier, I., and Pollefeys, M. (2019b). Learning 3D semantic reconstruction on octrees. In *German Conference on Pattern Recognition*.
- Wang, X., Yu, K., Dong, C., Tang, X., and Loy, C. C. (2019c). Deep network interpolation for continuous imagery effect transition. In *CVPR*.

- Wang, Y., Tan, D. J., Navab, N., and Tombari, F. (2018). Adversarial semantic scene completion from a single depth image. In *3DV*.
- Wang, Y., Tan, D. J., Navab, N., and Tombari, F. (2019d). ForkNet: Multi-branch volumetric semantic completion from a single depth image. In *ICCV*.
- Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*.
- Wu, S., Makadia, A., Wu, J., Snavely, N., Tucker, R., and Kanazawa, A. (2021). De-rendering the world’s revolutionary artefacts. In *CVPR*.
- Wu, S.-C., Tateno, K., Navab, N., and Tombari, F. (2020). SCFusion: Real-time incremental scene reconstruction with semantic completion. In *3DV*.
- Wymann, B., Espié, E., Guionneau, C., Dimitrakakis, C., Coulom, R., and Sumner, A. (2000). Torcs, the open racing car simulator.
- Xiao, J., Owens, A., and Torralba, A. (2013). Sun3D: A database of big spaces reconstructed using SfM and object labels. In *ICCV*.
- Xu, H., Gao, Y., Yu, F., and Darrell, T. (2016). End-to-end learning of driving models from large-scale video datasets. *arXiv*.
- Xu, H., Gao, Y., Yu, F., and Darrell, T. (2017). End-to-end learning of driving models from large-scale video datasets. In *CVPR*.
- Yan, X., Gao, J., Li, J., Zhang, R., Li, Z., Huang, R., and Cui, S. (2021). Sparse single sweep LiDAR point cloud segmentation via learning contextual shape priors from scene completion. In *AAAI*.
- Yang, F., Choi, W., and Lin, Y. (2016). Exploit all the layers: Fast and accurate cnn object detector with scale dependent pooling and cascaded rejection classifiers. In *CVPR*.
- Yang, Y. and Soatto, S. (2020). FDA: Fourier domain adaptation for semantic segmentation. In *CVPR*.
- Yin, W., Zhang, J., Wang, O., Niklaus, S., Mai, L., Chen, S., and Shen, C. (2021). Learning to recover 3d scene shape from a single image. In *CVPR*.
- Yoo, J., Uh, Y., Chun, S., Kang, B., and Ha, J.-W. (2019). Photorealistic style transfer via wavelet transforms. In *ICCV*.
- Yu, C., Wang, J., Gao, C., Yu, G., Shen, C., and Sang, N. (2020). Context prior for scene segmentation. In *CVPR*.
- Yuan, W., Khot, T., Held, D., Mertz, C., and Hebert, M. (2018). PCN: Point completion network. In *3DV*.
- Yuan, Y., Chen, X., and Wang, J. (2020). Object-contextual representations for semantic segmentation. In *ECCV*.
- Zhang, J., Zhao, H., Yao, A., Chen, Y., Zhang, L., and Liao, H. (2018a). Efficient semantic scene completion network with spatial group convolution. In *ECCV*.
- Zhang, L., Wang, L., Zhang, X., Shen, P., Bennamoun, M., Zhu, G., Shah, S. A. A., and Song, J. (2018b). Semantic scene completion with dense crf from a single depth image. *Neurocomputing*.

- Zhang, P., Liu, W., Lei, Y., Lu, H., and Yang, X. (2019). Cascaded context pyramid for full-resolution 3D semantic scene completion. In *ICCV*.
- Zhang, X., Wu, M., Ma, H., Hu, T., and Yuan, J. (2021). Multi-task long-range urban driving based on hierarchical planning and reinforcement learning. In *ITSC*.
- Zhao, H., Shi, J., Qi, X., Wang, X., and Jia, J. (2017). Pyramid scene parsing network. In *CVPR*.
- Zhao, H., Zhang, Y., Liu, S., Shi, J., Loy, C. C., Lin, D., and Jia, J. (2018). Psanet: Point-wise spatial attention network for scene parsing. In *ECCV*.
- Zheng, T., Fang, H., Zhang, Y., Tang, W., Yang, Z., Liu, H., and Cai, D. (2021). Resa: Recurrent feature-shift aggregator for lane detection. In *AAAI*.
- Zhong, M. and Zeng, G. (2020). Semantic point completion network for 3D semantic scene completion. In *ECAI*.
- Zhou, Y., Liu, S., and Ma, Y. (2021). Nerd: Neural 3d reflection symmetry detector. In *CVPR*.
- Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In *CVPR*.
- Zoph, B., Vasudevan, V., Shlens, J., and Le, Q. V. (2018). Learning transferable architectures for scalable image recognition. In *CVPR*.
- Zuo, G., Zhao, Q., Lu, J., and Li, J. (2020). Efficient hindsight reinforcement learning using demonstrations for robotic tasks with sparse rewards. *International Journal of Advanced Robotic Systems*.