Unsupervised depth prediction from monocular sequences: Improving performances through instance segmentation

Ambroise Moreau
Isia Lab
University of Mons
Mons, Belgium
ambroise.moreau@umons.ac.be

Matei Mancas
Isia Lab
University of Mons
Mons, Belgium
matei.mancas@umons.ac.be

Thierry Dutoit
Isia Lab
University of Mons
Mons, Belgium
thierry.dutoit@umons.ac.be

***Abstract*—Depth is a valuable piece of information for robots and autonomous vehicles. Indeed, it enables them to move in space and avoid obstacles. Nevertheless, depth alone is not enough to let them interact with their surroundings. They also need to locate the different objects that are present in their environment. In this paper, we propose a deep learning model that solves unsupervised monocular depth estimation and supervised instance segmentation at the same time with a common architecture. The first task is solved through novel view synthesis while the second is solved by minimising an embedding loss function. Our approach is motivated by the idea that knowing where objects are in the scene could improve the depth estimation of unsupervised monocular depth models. We tested our architecture on two datasets, Kitti and Cityscapes and reached state-of-the-art depth estimation results while solving a second task.**

***Keywords*-Computer vision; Monocular depth estimation; Instance segmentation; Multi-task learning**

I. Introduction

Recovering depth from RGB images has been an active field of research for decades. Indeed, depth perception is one of our most valuable assets when it comes to performing daily tasks. It enables us to reach for objects in our surroundings and to move in space while avoiding obstacles. Therefore, depth prediction from images has numerous applications in robotics and autonomous driving. Nevertheless, depth alone is not enough to enable machines to understand their environment. Indeed, robots will never be able to grasp one particular object if they cannot identify it among others and autonomous vehicles will not be able to avoid cars or pedestrians if they do not know that the item that is located two meters away from them is either one of those. In other words, to fully understand their environment, computers need to solve a second task that aims to identify and locate objects in an image. This task is known as segmentation. There are two main types of segmentation: semantic segmentation and instance segmentation [1]. The first one, which is the easiest, assigns pixels to different classes (e.g. car, tree, person, etc.) while the second groups together pixels that belong to one particular object of a class [2]. Recently, depth estimation and segmentation have profited from the advances in deep learning to achieve new state-of-the-art results.

In [3], Moreau et al. proposed a taxonomy of depth estimation algorithms based on the training strategy and the number of images needed to infer depth. The unsupervised monocular setting is the most challenging since depth prediction from a single view is an ill-posed problem. Nevertheless, it is also the one that imposes the fewest constraints on the datasets that can be used to train models. Indeed, video sequences recorded with a hand-held camera are usually enough to train such models. They rely on view synthesis to recover depth. More precisely, from a pair of images and the camera motion from the first to the second, these models learn to synthesise one of the views. During training, the models minimise the photometric error between the synthesised view and the real one and depth is only a by-product [4]–[6].

Different strategies have also been proposed to solve the instance segmentation task. Two categories of algorithms can be identified: proposal-based methods and proposal-free methods. As explained in [1], the first category is based on a detect-and-segment approach. It means that objects are first detected using a bounding-box detection method and then, that a binary mask is generated for each object. Proposal-based methods are accurate, but they are also slow and the resolution of the binary mask is low. On the contrary, the proposal-free methods avoid the use of the bounding-box detection component and work with embedding loss functions. Since these methods are built on dense-prediction networks, the predicted instance masks have a higher resolution. The most common embedding is the distance to the centroid of the instance mask [1], [2].

Instead of using two independent models to solve the tasks, a practical solution consists in sharing some components to compute shared features. This strategy in which parts are shared between different tasks is known as multi-task learning. The advantages of the approach are twofold. First, it reduces the number of parameters that have to be learned compared to the case where each task has its own model. Therefore, it also reduces training and processing time. Second, it has been shown that multi-task learning could improve the results. Nevertheless, combining tasks in a multi-task setting is not straightforward. Indeed, choosing uniform weights for each task's loss is suboptimal [2]. Moreover, there are various ways to create a shared architecture.

In this work, we propose a multi-task model that solves unsupervised depth prediction and instance segmentation at the same time. To the best of our knowledge, ours is the first attempt to combine the unsupervised approach of depth prediction with supervised instance segmentation. Our idea is that, having some knowledge about where objects are in the scene could help the model for the depth prediction task. Indeed, such knowledge is helpful for humans. For instance, we know that the depth of a glass door is close to that of the walls it is attached to, not to the depth of what we see through the glass.

The remainder of this paper is organised as follows. In Section II, we present a brief state of the art in depth prediction, instance segmentation and multi-task learning. Section III presents our multi-task model. It describes our architecture and the way we address both tasks. In Section IV, we detail our experiments on two well-known datasets: Kitti and Cityscapes. Finally, we conclude on our work and give some prospects for improvement.

II. Related Work

In the last few years, many solutions have been proposed to use deep learning to predict depth from RGB images. Eigen et al. were the first to leverage convolutional neural networks' abilities to solve the task in a supervised manner [7]. Garg et al. later solved the unsupervised setting through novel view synthesis [4]. More precisely, their model learns to synthesise the right image of a stereo pair by sampling from the left image. If constrained adequately, the model learns to predict disparity (i.e.inverse depth) as a by-product. Godard et al. improved this method by adding a left-right consistency check to the algorithm [5]. Indeed, since depth prediction from monocular images is an ill-posed problem, nothing prevents the model from synthesising a realistic view while predicting a wrong disparity map. The left-right consistency check encourages the disparity computed in one way, e.g. from left to right, to be consistent with the disparity computed in the other way. This reduces significantly the number of possible 3D structures that created the two views of the stereo pair. To enable training on monocular datasets, Zhou et al. proposed to solve camera motion estimation in addition to depth prediction [6]. Indeed, such datasets are easier to record. Their model is divided in two parts. The first one takes care of computing the camera motion, also known as "egomotion", between successive frames while the second computes the disparity map. Note that the solutions of Garg et al. and Godard et al. are only particular cases in which the egomotion is known beforehand thanks to stereo calibration and does not need to be computed. Others have built on this idea of computing depth and egomotion at the same time and have proposed variants of the algorithm. Mahjourian et al. argued that using a 2D photometric loss to compute 3D points was not the most robust approach and explicitly considered the inferred 3D geometry of the whole scene through an Iterative Closest Point term in the loss [8]. Wang et al. asserted that the pose network was suboptimal and replaced it by a differentiable and deterministic objective for pose prediction that is commonly used in Direct Visual Odometry. The advantage is that it does not require any learning compared to the pose network. In a subsequent work, Godard et al. improved the monocular approach with three simple architectural and loss innovations [9]. First, they address the problem of occluded pixels that occurs in the monocular setting. Then, they automatically ignore pixels where no relative camera motion is observed. Finally, their appearance matching loss performs all image sampling at the input resolution.

The first attempts at instance segmentation with convolutional neural networks were inspired by the advances in the object detection task. To solve the latter, Girschick et al. proposed R-CNN, an object detection system that consists of three modules [10]. The first one uses selective search [11] to generate 2000 category-independent region proposals. From these, a large convolutional neural network extracts fixed-length feature vectors. Finally, the third module uses these feature vectors as the inputs of a set of class-specific linear Support Vector Machines. R-CNN has an important drawback: each of the 2000 region proposals has to go through the convolutional neural network, which represents a heavy load. To address this shortcoming, Girschick et al. slightly modified the pipeline and added a Region of Interest (RoI) pooling layer [12]. With this architecture called Fast R-CNN, the entire image goes through the CNN only once and the feature vectors of each region are computed by the RoI pooling layer on the feature map. In Faster R-CNN, Ren et al. removed the selective search module and replaced it by a region proposal network, which enabled faster processing at test time. Mask R-CNN was built on Faster R-CNN. He et al. added a branch that takes care of computing a binary mask for the objects detected by the model [13]. As stated in [1], Mask R-CNN is still the most used method for instance segmentation, but proposal-free approaches are gaining traction. Compared to the previous, these methods run faster and generate high resolution masks, but they are often less accurate. They are based on embedding loss functions or pixel affinity learning. The most used pixel embedding is the distance to the centroid of the instance object. It has been used in [1], [2], [14]. It is usually learned thanks to a regression loss function with direct supervision. For each pixel, these models predict the offset vector that has to be added to its coordinates to point towards the object centroid. As explained in [2], a post-processing step is needed to cluster the pixels of an instance together, but

Neven et al. managed to avoid this step by using a hinged-loss function that maximises the intersection-over-union of the instance mask [1].

In the recent years, multi-task learning has attracted a lot of attention. Indeed, solving multiple tasks with one architecture has two advantages. First, it reduces the number of trainable parameters and thus, the learning and processing time. Then, it can also improve the results compared to when a task is solved alone, as demonstrated in [2]. In their work, Kendall et al. proposed a multi-task architecture to solve supervised depth prediction, semantic segmentation and instance segmentation. They showed that the naive approach that consists in equally weighting the loss functions of the tasks was suboptimal. Instead, they proposed to use the homoscedastic uncertainty which is the task-dependent uncertainty in Bayesian modelling. The weights are parameters that the model needs to learn. Sener and Koltun agree on the fact that equal weights are suboptimal, especially when tasks compete with each other. Instead of combining the different losses into one objective, they cast multi-task learning as multi-objective optimisation with the overall objective of finding a Pareto optimal solution [14]. To achieve this, they rely on algorithms developed in the gradient-based multi-objective optimisation literature. They tested their approach on several tasks among which, the same as those solved in [2]. Liu et al. proposed a multi-task architecture which allows learning of task-specific feature-level attention [15]. Their Multi-Task Attention Network (MTAN) is made of two parts. The first one is a shared network that takes care of computing shared features while the second part consists of task-specific soft-attention modules that use the shared features to compute task-specific features. MTAN is a flexible architecture, the soft-attention modules can be attached to any feedforward network. In addition to the task specific modules, they also proposed a novel weighting technique called Dynamic Weight Average (DWA) and based on the rate of change of the loss of each task. By doing so, MTAN does not need to learn additional parameters. They also applied the homoscedastic weighting to their architecture, but reported better results with DWA. They tested MTAN on depth prediction and semantic segmentation as well as other combinations of tasks.

III. Method

Our work draws inspiration from some of the approaches presented in Section II. To solve depth estimation and instance segmentation at the same time, we use the soft-attention modules proposed by Liu et al. [15]. The remainder of this Section describes our adaptation of MTAN to solve both tasks. For an extensive description of the attention modules, we refer to the original publication [15].

A. Unsupervised Depth Estimation

To deal with unsupervised depth estimation, we relied heavily on Monodepth2, the model of Godard et al. regarding monocular depth prediction [9]. Fig. 1 shows a global scheme of our model. We adopt the same representation symbols as in [15]. The core of our architecture is an encoder-decoder pair in which the encoder is a Resnet18 and the decoder is a succession of five deconvolutional blocks with skip connections. Contrary to Godard et al., we do not use a separate network to solve pose estimation. Instead, we add soft-attention modules on top of our core architecture. On the encoder side, four attention encoders, named ae_{di}, are dedicated to computing depth-related features and four other attention encoders, termed ae_{pi}, are in charge of computing pose-specific features. On the decoder side, since depth and pose outputs have distinctly different shapes, we take another approach. In addition to the attention decoders, we also add four convolutional blocks C with non linear activation functions. For the depth estimation, we add five soft-attention decoders ad_{di} while for the pose estimation part, we use the same decoder as in Monodepth2. Our choices give two advantages to our model:

- First, our network is lighter than Monodepth2 since we avoid the use of a separate pose encoder;
- Second, using soft-attention decoders for depth estimation enables to stack other attention modules dedicated to any other task.

Our model learns novel view synthesis, i.e. it is trained to synthesise a target image I_t from several source images $I_{t'}$ taken from different points of view. The subscript t' either represents the frame at time $t-1$ or the one at time $t+1$ in a monocular sequence. To achieve novel view synthesis, it also needs to compute the relative pose for each source view $I_{t'}$ with respect to the target image I_t's pose, which we write $T_{t\to t'}$. With the adequate constraint, the model yields the disparity or depth map D_t corresponding to the view I_t. During training, it minimises the same photometric reprojection error $\mathcal{L}_p$ as in [9]:

$$\mathcal{L}_p = \sum_{t'} \min_{t'} \big(\mathrm{pe}\big(I_t, I_{t'\to t}\big)\big) \tag{1}$$

where,

$$\mathrm{pe}\big(I_a, I_b\big) = \frac{\alpha}{2}\big(1 - \mathrm{SSIM}\big(I_a, I_b\big)\big) + \big(1-\alpha\big)\,\|I_a - I_b\|_1 \tag{2}$$

and,

$$I_{t'\to t} = I_{t'}\big\langle \mathrm{proj}\big(D_t, T_{t\to t'}, K\big)\big\rangle \tag{3}$$

In (1) keeping the minimum rather than computing the mean photometric error over all the source images improves the prediction where occlusions and disocclusions occur in the monocular sequence. The photometric error, given by (2), uses the structural similarity index (SSIM) and the $\mathcal{L}_1$-norm [16]. The parameter α is set to 0.85. In (3), $I_{t\to t'}$

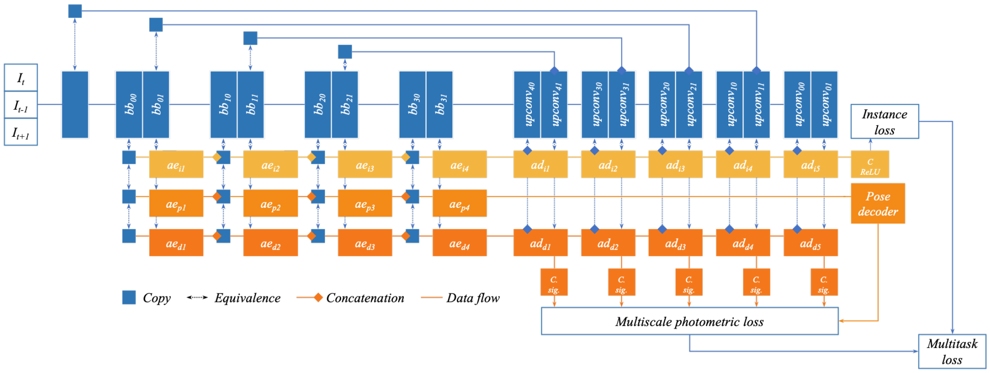


Figure 1: Architecture of our multi-task model.

is the synthesised view, $\langle \cdot \rangle$ designates the bilinear sampling operator and $\text{proj}(D_t, T_{t\to t'}, K)$ is the adequate constraint that forces the network to compute disparity. It represents the 2D coordinates of the projected depth D_t in $I_{t'}$, obtained thanks to the relative pose $T_{t\to t'}$ and the camera intrinsic parameters K. The latter are not learned by the model.

The two other improvements proposed in [9] are also implemented in our work. The first one consists in automatically masking pixels that violate the assumption of a moving camera in a static scene. Such pixels have similar appearances from one frame to the other and can easily be detected using this characteristic. More precisely, pixels belonging to moving objects or to textureless regions or even entire frames when the camera is static, are discarded from the loss thanks to binary masks. The second improvement consists in upsampling the intermediate low-resolution disparity predictions made by the intermediate soft-attention decoders to the resolution of the input frames. Indeed, computing the photometric loss at each of the decoder scale creates holes in large low-texture regions as well as texture copy artefacts. Instead, intermediate disparity maps are first upsampled, reprojected and then resampled to obtain a synthesised view with the resolution of the input.

Finally, as in [9], the depth loss function also encompasses an edge-aware smoothing term that helps against the ill-posed nature of the monocular depth estimation problem:

$$\mathcal{L}_s = \left|\partial_x d_t^*\right| e^{-|\partial_x I_t|} + \left|\partial_y d_t^*\right| e^{-|\partial_y I_t|} \tag{4}$$

where $d_t^* = d_t / \overline{d_t}$ is the mean normalised inverse depth introduced in [17] to discourage the predicted disparity from shrinking to zero. The resulting depth loss is given by:

$$\mathcal{L}_d = \sum_{sc} \left(\mu^{sc} \mathcal{L}_p^{sc} + \lambda \mathcal{L}_s^{sc}\right) \tag{5}$$

in which sc represents the different scales, μ designates the binary masks used to discard the violating pixels and λ is a constant parameter set to 0.001.

B. *Supervised Instance Segmentation*

To address the problem of instance segmentation, we treat it as the minimisation of an embedding loss. This approach is the most compatible with a feedforward network like the one we are using and avoids the addition of a proposal network in the architecture. To achieve instance segmentation in our multi-task framework, similarly to what is done for depth estimation, we connect four soft-attention encoders and five soft-attention decoders to our core architecture. They are respectively called ae_{ii} and ad_{ii} in Fig. 1

As in [2], [14], our model learns to predict a 2D offset vector o^p for each pixel p that belongs to an instance I so that, when the offset vector is added to the pixel coordinates, it points to the object's centroid C^I:

$$C_x^I = p + o_x^p \text{ and } C_y^I = p + o_y^p \tag{6}$$

The learning is supervised by a $\mathcal{L}_1$-loss:

$$\mathcal{L}_i = \|o - \overline{o}\|_1 \tag{7}$$

where $\overline{o}$ designates the ground truth 2D offset vectors map. Pixels that do not belong to any instance are masked in the loss. Due to the supervised nature of the approach, our model is only able to find instances of objects belonging to a limited number of classes, i.e. the ones for which instances are identified in the dataset. Contrary to what is done on the depth side, the loss is only computed on the last scale of the decoder features. Therefore, there is only one additional C block with a rectified linear unit (ReLU) function.

C. *The Multi-task Loss*

As demonstrated by Kendall et al., the naive strategy that uses equal weights when combining the various losses leads to suboptimal results [2]. Here, we explain how our model can use two existing weighting techniques to balance the losses of depth prediction and instance segmentation.

In [2], to improve the performances compared to equal weighting, they rely on the homoscedastic uncertainty which is the task-related uncertainty in Bayesian modelling. The weights derived from the homoscedastic uncertainty can be learned by the network. This weighting scheme is termed HUW in the remainder of the paper. Our model can use HUW to balance the different losses. It just needs to predict two additional parameters. The multi-task loss with HUW is given by:

$$\mathcal{L}_{HUW} = \frac{1}{2\sigma_d^2}\mathcal{L}_d + \frac{1}{2\sigma_i^2}\mathcal{L}_i + \log\sigma_d + \log\sigma_i \tag{8}$$

where σ_d and σ_i are the homoscedastic uncertainties of the depth estimation and the instance segmentation tasks respectively.

In MTAN, Liu et al. proposed a different solution called Dynamic Weighting Average (DWA). It is based on the rate of change of the loss of each task. The weights are therefore computed from the numeric values of the losses and the model does not have to learn any new parameter. The weight of task k among K tasks is updated as follows at time t:

$$\lambda_k(t) = \frac{K\exp\big(w_k(t-1)/T\big)}{\sum_i \exp\big(w_i(t-1)/T\big)} \tag{9}$$

with $w_k(t-1)$ given by:

$$w_k(t-1) = \frac{\mathcal{L}_k(t-1)}{\mathcal{L}_k(t-2)} \tag{10}$$

The parameter T in (9) is a temperature that controls the softness of task weighting. If T is large enough, tasks are weighted equally. In (10), $\mathcal{L}_k(t)$ is computed as the average of loss k in each epoch over several iterations to reduce the uncertainty from stochastic gradients descent and random training data selection. At times $t \in \{1, 2\}$, the weights are initialised arbitrarily. The multi-task loss resulting from the use of DWA on our two tasks is:

$$\mathcal{L}_{DWA} = \frac{1}{2}\big(\lambda_d\mathcal{L}_d + \lambda_i\mathcal{L}_i\big) \tag{11}$$

IV. Experiments

This Section describes our experiments with our multi-task architecture on two well-known datasets presented in Section IV-A: Kitti and Monodepth. In Section IV-B, we present the different configurations under test. Finally, we evaluate the performances of depth estimation and instance segmentation in Section IV-C and Section IV-D respectively.

A. Datasets

Kitti is the most used dataset to train models for depth prediction [18]. Created to encourage the development of autonomous driving systems, it contains 61 road sequences recorded in rural areas or on highways. Greyscales and RGB images were recorded by two stereo rigs on the car roof while depth ground truth was acquired by a 360° velodyne laser scanner. This ground truth is sparse and only 5% of the pixels have a known depth value. All these data are synchronised at 10 Hz. Among the frames, 400 are annotated for instance and semantic segmentation. The instances are divided in 8 classes: *'person', 'rider', 'car', 'truck', 'bus', 'train', 'motorcycle'* and *'bike'*.

Cityscapes was also recorded on the road in 50 different cities with a stereo rig but without any depth sensor [19]. Instead, the dataset contains disparity maps computed with the semi-global matching algorithm (SGM) [20]. Video sequences were recorded at 17 Hz and are 30-frames long. High quality pixel-level annotations of 5000 frames are available as well as a larger set of 20 000 weakly annotated ones. The instance classes are the same as those of Kitti. Cityscapes is known to be a much more realistic and challenging dataset than the previous one since it contains many moving objects.

For the depth prediction task, our model needs three successive frames I_i, $i \in \{t-1, t, t+1\}$, at training. Consequently, it also needs three successive instance ground truth masks M_i, $i \in \{t-1, t, t+1\}$ to be trained for the instance segmentation task at the same time. Nevertheless, none of the two datasets provides such labelled sequences. Therefore, to enable multi-task training, we generate the missing data with Mask R-CNN. More precisely, the latter is trained on Cityscapes and its prediction on each unlabelled frame of both datasets is used to complete the ground truth. A similar strategy has already been used in [21].

B. Training Details

For our experiments on Kitti, we use the so-called Eigen split introduced in [7]. As in [6] and [9], we remove static frames from the dataset. There remain 39 810 triplets for training and 4424 for validation. For Cityscapes, we randomly select as many training and validation triplets as for the first dataset.

To validate the benefits of multi-task learning, various configurations of our architecture are trained:

- **MT**$_d^d$ designates the multi-task setting with DWA. The temperature parameter is set to 2, as recommended by the authors;
- **MT**$_h^d$ refers to the multi-task setting with homoscedastic weighting;
- **MT**$_e^d$ is the multi-task setting with equal weighting;
- **ST**$_d^d$ is the configuration dedicated to depth and pose estimation;
- **ST**$_i^d$ designates the configuration that solves instance segmentation alone.

Attention modules are used even in the single-task configuration. The superscript d designates the dataset. It is either replaced by k or cs in the following.

Our models are implemented in PyTorch [22]. The encoder of the shared network is a pre-trained dilated Resnet18 modified to accept multiple frames as input. The remaining components are initialised randomly. The training pattern is similar to the one of Monodepth2. The models are trained for 20 epochs with batches of 5 samples with a resolution

TABLE I: QUANTITATIVE DEPTH EVALUATION ON KITTI AND CITYSCAPES. OUR DIFFERENT TRAINING CONFIGURATIONS ARE COMPARED TO MONODEPTH2. LOWER VALUES ARE BETTER FOR THE FOUR FIRST METRICS. MEDIAN SCALING IS APPLIED TO ALL THE RESULTS. PIXELS WITH A GROUND TRUTH DEPTH HIGHER THAN 80 m ARE IGNORED. BEST RESULTS ON KITTI ARE HIGHLIGHTED IN BLUE WHILE BEST RESULTS ON CITYSCAPES ARE HIGHLIGHTED IN GREEN.

Model	Test Set	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Monodepth2	K	0.115	0.903	4.863	0.193	0.877	0.959	0.981
	Cs	0.186	4.798	8.298	0.236	0.815	0.929	0.963
ST$_d^k$	K	0.126	1.113	5.196	0.203	0.860	0.954	0.978
	Cs	0.175	2.153	9.375	0.261	0.728	0.906	0.967
ST$_d^{cs}$	K	0.212	3.821	7.137	0.274	0.749	0.916	0.964
	Cs	0.181	4.109	8.406	0.246	0.804	0.923	0.961
MT$_d^k$	K	0.116	0.815	4.771	0.190	0.869	0.960	0.983
	Cs	0.167	1.887	9.162	0.251	0.744	0.915	0.972
MT$_h^k$	K	0.119	0.864	4.877	0.192	0.865	0.957	0.982
	Cs	0.164	1.932	9.244	0.252	0.752	0.911	0.969
MT$_e^k$	K	0.118	0.835	4.774	0.190	0.866	0.960	0.983
	Cs	0.167	1.881	9.046	0.248	0.748	0.917	0.972
MT$_d^{cs}$	K	0.185	1.549	6.657	0.259	0.726	0.913	0.968
	Cs	**0.127**	**1.504**	**6.915**	**0.189**	**0.855**	**0.956**	**0.984**
MT$_h^{cs}$	K	0.183	1.504	6.384	0.254	0.732	0.918	0.971
	Cs	0.141	1.931	7.436	0.203	0.838	0.949	0.980
MT$_e^{cs}$	K	0.183	1.433	6.348	0.256	0.728	0.914	0.969
	Cs	0.133	1.678	7.228	0.195	0.847	0.953	*0.983*

of 640×192 for the input and the output as well. The learning rate is set to 10^{-4} for the first 15 epochs and to 10^{-5} afterwards.

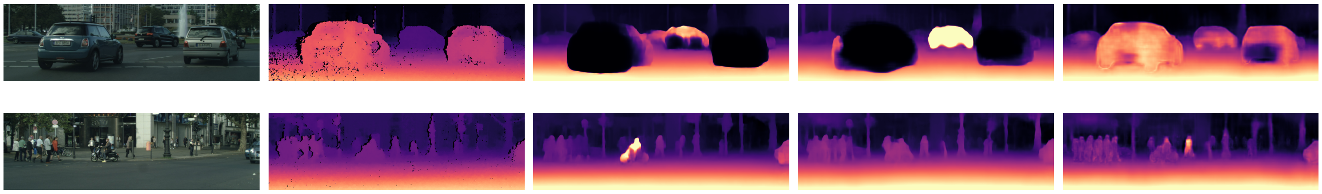
C. Depth Evaluation

We report the values of the 7 most commonly used metrics in terms of depth evaluation in Table I. For Kitti, the test set is that of Eigen which contains 697 frames. The predictions are cropped, according to common practice, and compared to the Lidar recordings provided in the dataset. They are sparse and only cover around 5% of the pixels. For Cityscapes, we use the 1525 test images provided by the authors [19]. This time, the predictions are compared to the disparity computed with SGM [20]. In both cases, depth is capped to 80 m, according to common practice. As our model is monocular and is only able to predict up-to-scale depth maps, we apply the median scaling proposed in [6] when computing the metrics. The results of Monodepth2 on Kitti are those of the original publication [9] while those on Cityscapes are from our own training.

From these results, it appears that our single-task configuration performs worse than Monodepth2, regardless of the training set. The better performances of Monodepth2 are likely due to their use of a separate pose network. Moreover, the model trained on Kitti gives better results compared to the one trained on Cityscapes which could be related to the latter being more challenging than Kitti.

On the contrary, the multi-task configuration leads to results that match Monodepth2 or even outperform it. This demonstrates that letting a model know that some pixels belong to certain objects can improve the depth estimation. Regarding the loss weighting strategy, we do not observe any improvement when using HUW, but the DWA approach appears to lead to the best results as observed in Table I.

Fig. 2 shows two qualitative examples of our results on Cityscapes. In the first frame, two cars are moving in the same direction as the camera. They violate the assumption of staticity and both Monodepth2 and the single task configuration of our model fail to recover their depth. The estimated disparity is equal to zero which corresponds to an infinite depth. On the contrary, we observe that the depth estimation is improved when the model is trained to solve instance segmentation at the same time, which demonstrates the benefit of introducing instance knowledge in the process. However, some artefacts still remain in the car on the right. The second frame shows a group of pedestrians crossing the road on the left. Again, we can see that our model recovers their shape more accurately than Monodepth2.



Input　Ground truth　Monodepth2　ST$_d^{cs}$　MT$_d^{cs}$

Figure 2: Qualitative results of depth prediction on Cityscapes.

D. *Instance Segmentation Evaluation*

To achieve instance segmentation, our model uses the instance masks computed with Mask R-CNN for the seven classes present in each dataset. More preciesely, it relies on ground truth embeddings computed from these masks. The embeddings are normalised so that they are within the interval $[0;1[$. Background pixels are set to 1 to easily mask them in the loss computation as explained in Section III-B.

The predicted embeddings could be used with a clustering algorithm such as OPTICS, as in [2]. Nevertheless, our model is unaware of the class of the detected objects. Therefore, we could not compute the traditional evaluation metrics without getting the class ID from the ground truth. Instead, similarly to [14], we report the Mean Squared Error of the embeddings in Table II. For Kitti, we use the 200 frames dedicated to instance segmentation while for Cityscapes, we use again the 1525 test frames. As none of these two test sets contains ground truth data, we rely on the prediction of Mask R-CNN to compute the MSE. We also report the masked MSE which is obtained by masking the background pixels. If both errors are close, then, it means the model finds the same instances as in the ground truth. Otherwise, it either misses some instances or finds more than it should.

The best results on Kitti are obtained with $\mathbf{ST}_i^{cs}$, the single task configuration trained for instance segmentation on Cityscapes. It is closely followed by $\mathbf{MT}_h^k$, the multi-task configuration trained on Kitti with HUW. On Cityscapes, $\mathbf{ST}_i^{cs}$ and $\mathbf{MT}_h^{cs}$ are the best performing models. This demonstrates that depth estimation does not necessarily help instance segmentation within our framework. We also observe that the DWA strategy has an opposite effect on the instance segmentation task compared to the depth estimation.

Fig. 3 shows an example of segmentation with $\mathbf{MT}_h^{cs}$. The first row shows the input frame, the embeddings and the result of OPTICS on these latter. The second row shows the output of Mask R-CNN and the corresponding ground truth. Edges of the clusters computed from the ground truth embeddings are poorly recovered, but this is purely due to a wrong choice of preprocessing steps'order. Indeed, ground truth embeddings are computed on the full frame resolution before being resized to 640 × 192. Inverting the two step is likely to solve that issue.

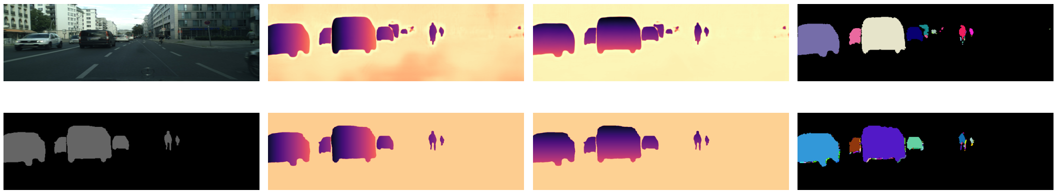
TABLE II: MEAN SQUARED ERROR FOR THE INSTANCE SEGMENTATION TASK ON KITTI AND CITYSCAPES.

Model	Test Set	MSE	Masked MSE
$\mathbf{ST}_i^k$	K	0.010	0.007
	Cs	0.024	0.016
$\mathbf{ST}_i^{cs}$	K	0.006	0.001
	Cs	**0.012**	**0.006**
$\mathbf{MT}_d^k$	K	0.105	0.064
	Cs	0.109	0.081
$\mathbf{MT}_h^k$	K	0.008	0.001
	Cs	0.023	0.012
$\mathbf{MT}_e^k$	K	0.022	0.025
	Cs	0.048	0.025
$\mathbf{MT}_d^{cs}$	K	0.013	0.011
	Cs	0.014	0.010
$\mathbf{MT}_h^{cs}$	K	0.012	0.011
	Cs	**0.012**	0.007
$\mathbf{MT}_e^{cs}$	K	0.009	0.006
	Cs	0.013	0.008

V. CONCLUSION

In this work, we presented a multi-task architecture dedicated to unsupervised monocular depth estimation and supervised instance segmentation. Our model is made of a shared feedforward architecture and soft-attention modules connected along the latter. The shared network computes general features while the soft-attention modules refine them to create task-specific features that help to solve each task. For the depth estimation task, the model learns novel view synthesis which enables it to learn depth as a by-product and avoids the need of ground truth depth. For the instance segmentation task, we use an embedding loss as it is the approach that is the most compatible with feedforward networks.

Regarding depth estimation, our results show the benefit of combining both tasks. Indeed, making the model learn where to find objects in the scene helps it to improve its depth prediction abilities. Our model also gives satisfying instance segmentation. Nevertheless, it could benefit from semantic knowledge that would help it know the class the objects belong to. This could be brought by the embedding loss function presented in [1] or by adding a semantic segmentation task.



Input - Mask R-CNN x-embeddings y-embeddings OPTICS clusters

Figure 3: Qualitative results of instance segmentation on Cityscapes with $\mathbf{MT}_h^{cs}$.

References

[1] Davy Neven, Bert De Brabandere, Marc Proesmans, et al. “Instance Segmentation by Jointly Optimizing Spatial Embeddings and Clustering Bandwidth”. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2019.

[2] Alex Kendall, Yarin Gal, and Roberto Cipolla. “Multi-task learning using uncertainty to weigh losses for scene geometry and semantics”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 7482–7491.

[3] Ambroise Moreau, Matei Mancas, and Thierry Dutoit. “Depth prediction from 2D images: A taxonomy and an evaluation study”. In: *Image and Vision Computing* (2019), p. 103825.

[4] Ravi Garg, Vijay Kumar BG, Gustavo Carneiro, et al. “Unsupervised cnn for single view depth estimation: Geometry to the rescue”. In: *European Conference on Computer Vision*. Springer. 2016, pp. 740–756.

[5] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. “Unsupervised monocular depth estimation with left-right consistency”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 270–279.

[6] Tinghui Zhou, Matthew Brown, Noah Snavely, et al. “Unsupervised learning of depth and ego-motion from video”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 1851–1858.

[7] David Eigen, Christian Puhrsch, and Rob Fergus. “Depth map prediction from a single image using a multi-scale deep network”. In: *Advances in neural information processing systems*. 2014, pp. 2366–2374.

[8] Reza Mahjourian, Martin Wicke, and Anelia Angelova. “Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 5667–5675.

[9] Clément Godard, Oisin Mac Aodha, Michael Firman, et al. “Digging into self-supervised monocular depth estimation”. In: *arXiv preprint arXiv:1806.01260* (2018).

[10] Ross Girshick, Jeff Donahue, Trevor Darrell, et al. “Rich feature hierarchies for accurate object detection and semantic segmentation”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pp. 580–587.

[11] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, et al. “Selective search for object recognition”. In: *International journal of computer vision* 104.2 (2013), pp. 154–171.

[12] Shaoqing Ren, Kaiming He, Ross Girshick, et al. “Faster r-cnn: Towards real-time object detection with region proposal networks”. In: *Advances in neural information processing systems*. 2015, pp. 91–99.

[13] Kaiming He, Georgia Gkioxari, Piotr Dollár, et al. “Mask r-cnn”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2961–2969.

[14] Ozan Sener and Vladlen Koltun. “Multi-task learning as multi-objective optimization”. In: *Advances in Neural Information Processing Systems*. 2018, pp. 527–538.

[15] Shikun Liu, Edward Johns, and Andrew J Davison. “End-to-end multi-task learning with attention”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 1871–1880.

[16] Zhou Wang, Alan C Bovik, Hamid R Sheikh, et al. “Image quality assessment: from error visibility to structural similarity”. In: *IEEE transactions on image processing* 13.4 (2004), pp. 600–612.

[17] Chaoyang Wang, José Miguel Buenaposada, Rui Zhu, et al. “Learning depth from monocular videos using direct methods”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 2022–2030.

[18] Andreas Geiger, Philip Lenz, and Raquel Urtasun. “Are we ready for autonomous driving? the kitti vision benchmark suite”. In: *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2012, pp. 3354–3361.

[19] Marius Cordts, Mohamed Omran, Sebastian Ramos, et al. “The cityscapes dataset for semantic urban scene understanding”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 3213–3223.

[20] Heiko Hirschmuller. “Stereo processing by semiglobal matching and mutual information”. In: *IEEE Transactions on pattern analysis and machine intelligence* 30.2 (2007), pp. 328–341.

[21] Vincent Casser, Soeren Pirk, Reza Mahjourian, et al. “Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 2019, pp. 8001–8008.

[22] Adam Paszke, Sam Gross, Francisco Massa, et al. “PyTorch: An Imperative Style, High-Performance Deep Learning Library”. In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, et al. Curran Associates, Inc., 2019, pp. 8024–8035. URL: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.