Depth estimation in the wild.

Alessio Tonioni – Postdoc at CVLAB, University of Bologna

Zurich, 08/05/2019

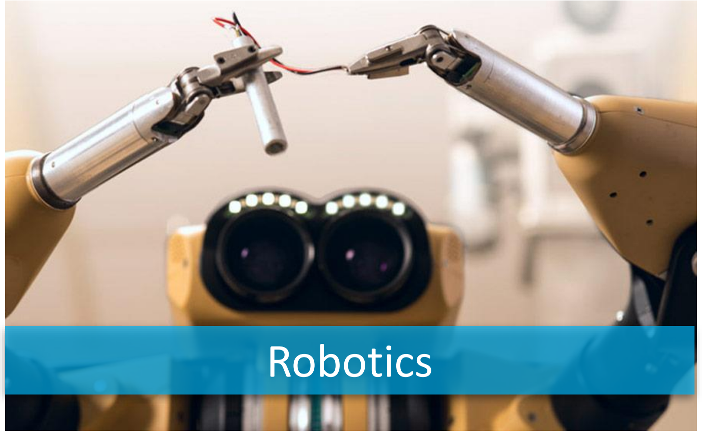
About Me

Personal Web Page

- Postdoc researcher in computer vision at the Computer Vision Lab of the University of Bologna with Professor Luigi Di Stefano.
- I received my Ph.D. in Computer Science and Engineering from the University of Bologna in April 2019.
- During my Ph.D. I have worked on deep learning applied to depth estimation from stereo and monocular cameras and on solutions for product detection and recognition in retail environments.
- I am continuing to work on depth estimation while starting to explore how to take advantage of the recent development of more general research subjects like domain adaptation and meta-learning.

Depth Estimation

Acquiring information about the 3D structure of an observed scene is a fundamental technology for more complex systems and applications.



Depth Estimation – Sensors

Active Sensors

Passive Sensors

Depth estimation from images

- State of the art algorithms for depth estimation from passive sensors are all based on some form of machine (deep) learning.
- A single CNN takes one (or two) images as input and directly regress a dense depth map as output. The training is performed using supervised regression losses.

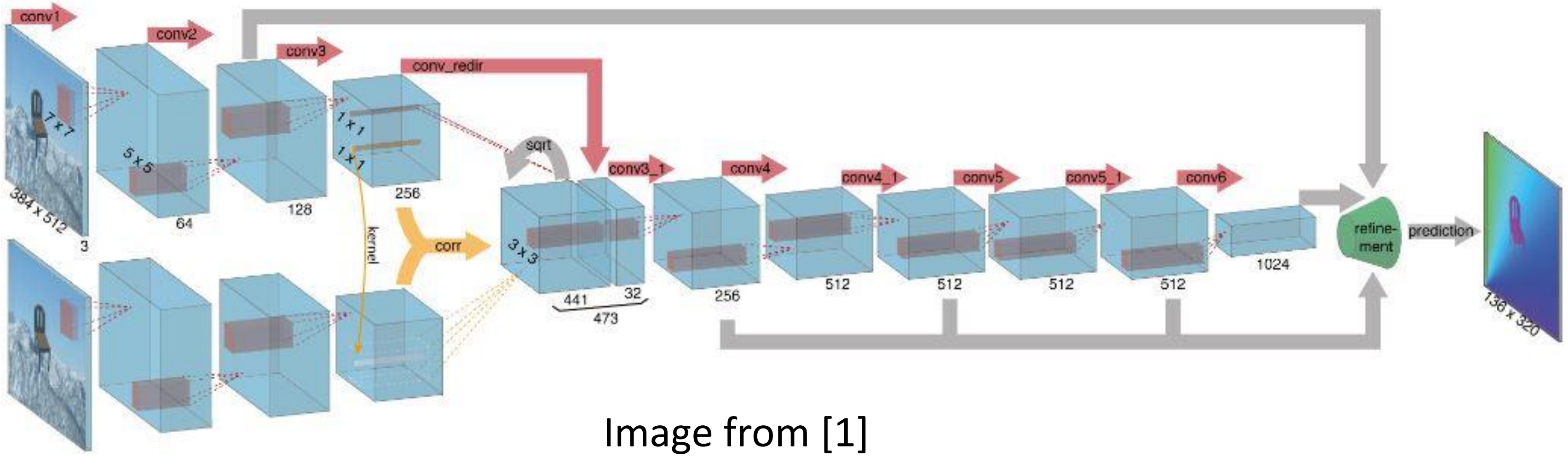


Image from [1]

1. Dosovitskiy, Alexey, et al. "Flownet: Learning optical flow with convolutional networks." *Proceedings of the IEEE international conference on computer vision*. 2015.

Deep Learning for Depth Estimation

Obtaining dense ground truth annotation for depth estimation is a quite challenging task per se. To overcome this issue:

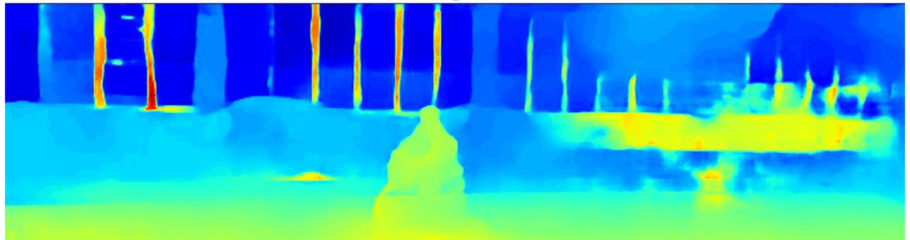
1. Use **rendered images** as the main training set with perfect depth information obtained freely as a by-product of the image creation process.
2. Fine tune the model on (*potentially few*) annotated data from the target environment.

Q: Do we really need the second step?

Depth Estimation and Domain Shifts

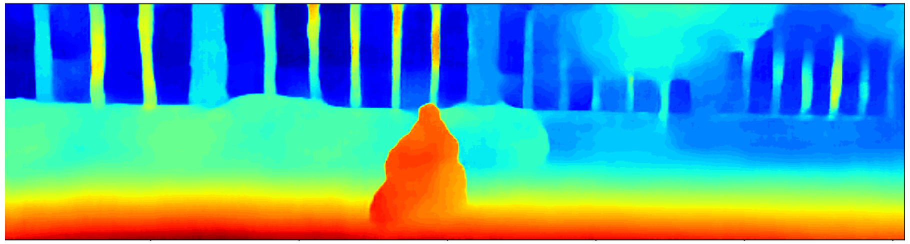
- Deep learning models for depth estimation, either monocular or stereo, struggle to generalize to unseen images due to the domain shifts between the train and test data.
- The second step of fine-tuning turns out crucial to regain good performance.

Disparity prediction obtained from a deep stereo network, hotter colors denotes points closer to the camera.



Trained only on synthetic data

Fine tuning



Fine tuned on few real data

Proxy labels for domain adaptation [a,b]

Observations:

- ML-based systems need to be fine-tuned to the target environment to get good performance.
- Producing annotated data is expensive and require ad hoc sensors and acquisition modalities.

Proposal:

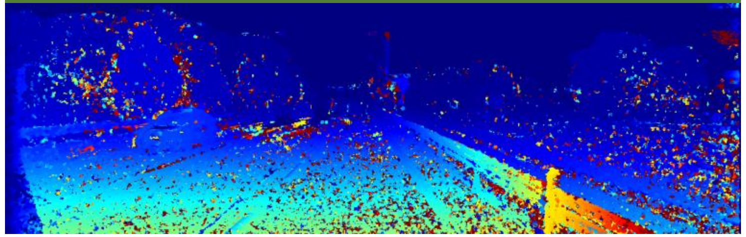
- Rely only on stereo images from the target environment.
- Use *traditional stereo algorithms* to produce a noisy proxy label for each pixel.
- Minimize a regression loss between the model prediction and the proxy labels weighting each reconstruction mistakes according to a *stereo confidence measure*.

[a] **Tonioni, Alessio** and Poggi, Matteo and Mattoccia, Stefano and Di Stefano, Luigi. "Unsupervised Adaptation for Deep Stereo." ICCV 2017.
[b] **Tonioni, Alessio** and Poggi, Matteo and Mattoccia, Stefano and Di Stefano, Luigi. " Unsupervised domain adaptation for depth prediction from images". Under review @ PAMI

Confidence Guided Regression [a]

Stereo RGB couples

Proxy labels from stereo alg. (D)

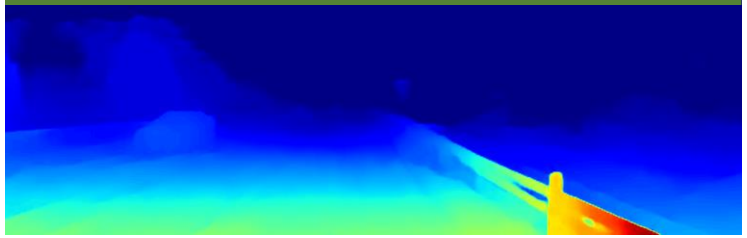


Confidence function (C)

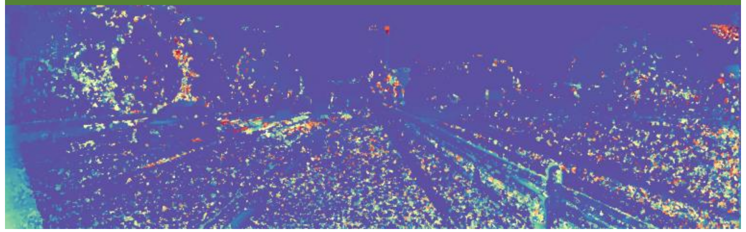
$-$

$*$

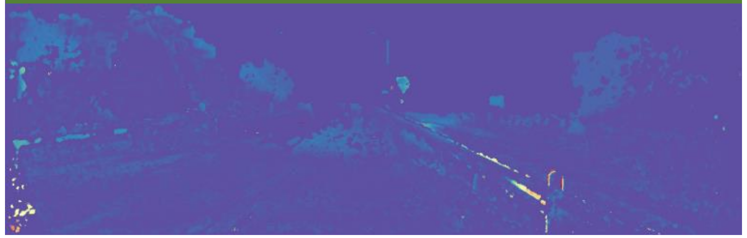
Network Prediction ($\hat{D}$)



L1-Regression Loss ($|\hat{D} - D|$)



Our Confidence Guided Loss (L_c)



$$L_c = \frac{1}{|P_v|}\sum_{p\in P_v} C(p)\cdot|\hat{D}(p) - D(p)|$$

$$P_v = \{p \in P : C(p) > \tau\}$$

Learnable

Set of all pixels

Self supervision via photo consistency losses

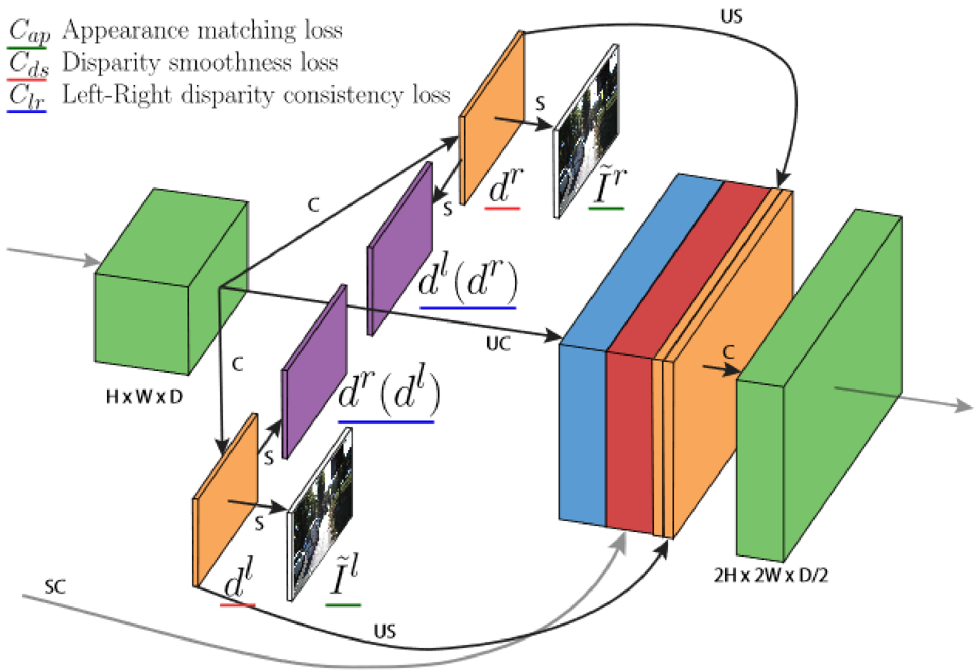


Image from [2]

- Photo-consistency loss computed warping the right frame on the left reference frame according to the predicted disparity values + adding some smoothness constraint.
- Similar ideas have been extensively used on many recent works for mono and stereo to train or fine tune depth estimation models, e.g. [2,3].

2. Clément Godard, Oisin Mac Aodha and Gabriel J Brostow. *'Unsupervised monocular depth estimation with left-right consistency'*. CVPR 2017
3. Y. Zhang, et al. *"Active stereonet: End-to-end self-supervised learning for active stereo systems"*. ECCV 2018

Results Stereo Depth Estimation

Loss	Target Domain bad3	Target Domain MAE	Similar Domains bad3	Similar Domains MAE
(a) No Adaptation	10.86	1.73	10.86	1.73
(b) GT Tuned (K12/15)	5.04	1.28	5.04	1.28
(c) Godard et. al. [56]	4.01	1.07	4.20	1.09
(d) Yinda et. al. [23]	3.59	1.00	5.15	1.14
(e) Tonioni et. al. [63]-AD	3.10	0.97	3.80	1.05
(f) *Masked-AD+Smooth.*	3.17	0.98	3.78	1.05
(g) Tonioni et. al. [63]-SGM	2.73	0.93	3.71	1.09
(h) *Masked-SGM+Smooth.*	2.79	1.01	3.63	1.09
(i) *Adaptation-AD* (τ=0.8)	2.96	0.96	3.66	1.04
(j) *Learned Adaptation-AD*	3.15	1.01	3.88	1.08
(k) *Adaptation-SGM* (τ=0.9)	**2.58**	**0.91**	3.39	**1.01**
(l) *Learned Adaptation-SGM*	2.84	0.99	3.75	1.07
(m) *Adaptation-AD-SGM*	2.61	0.92	**3.37**	**1.01**
(n) *Learned Adaptation-AD-SGM*	2.77	0.99	3.54	1.07

TABLE 2

Results obtained performing fine tuning of a pre-trained DispNetC network using different unsupervised strategy. All results are computed on the KITTI raw dataset using a 4-fold cross validation schema, best results highlighted in bold, our proposals in italic.

- Photo-Consistency losses (rows c and d) perform worse than our confidence guided regression (rows f and h).
- The two types of losses are complementary and the best performance can be obtained using them all together as shown by the results on row k.

Table from [b]

Results Mono Depth Estimation

The same considerations hold for depth from monocular camera models.

Table from [b]

Godard et al. [56]	ResNet50+pp	0.114	0.898	4.935	0.206	0.861	0.949	0.976
Masked-AD	ResNet50+pp	**0.109**	0.867	4.810	0.197	0.866	0.953	0.979
Adaptation-AD	ResNet50+pp	**0.109**	0.867	4.852	0.196	0.866	0.954	0.978
Learned Adaptation-AD	ResNet50+pp	0.110	0.864	4.953	0.195	0.858	0.948	0.976
Masked-SGM	ResNet50+pp	**0.109**	0.837	4.703	0.194	0.867	0.955	0.980
Adaptation-SGM	ResNet50+pp	**0.109**	**0.831**	**4.681**	**0.193**	0.867	**0.956**	**0.981**
Learned Adaptation-SGM	ResNet50+pp	0.111	0.880	4.820	0.196	0.864	0.954	0.980
Masked-AD-SGM	ResNet50+pp	0.110	0.866	4.775	0.195	0.867	0.955	0.980
Adaptation-AD-SGM	ResNet50+pp	0.110	0.891	4.809	0.196	**0.868**	**0.956**	**0.981**
Learned Adaptation-AD-SGM	ResNet50+pp	0.110	0.879	4.838	0.198	0.864	0.953	0.979

TABLE 3

Experimental results on the KITTI dataset [66] on the data split proposed by Eigen et al. [44]. On even conditions, the proposed adaptation scheme outperforms the supervision by Godard et al. [56].

Can we do it live?

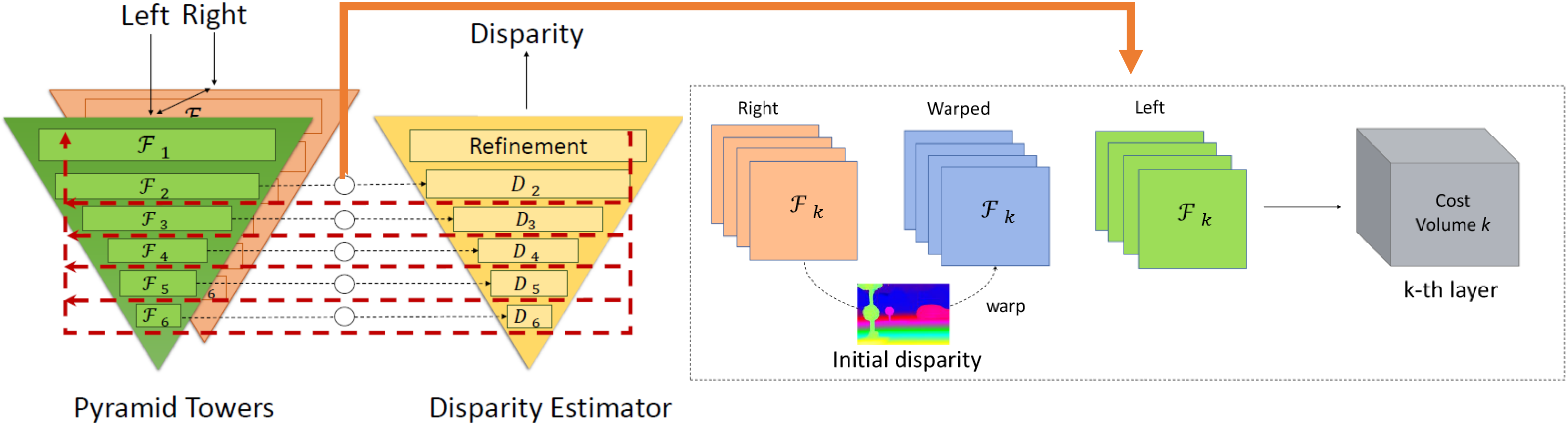
- All the previous solutions require the preliminary acquisition of data from the target environment and a long offline fine-tuning before deployment
- For many practical applications this is cumbersome and/or unfeasible (e.g., autonomous driving).

Q: Can we obtain the same performance adapting the network to a target environment online as soon as new frames are acquired?

- Photometric consistency losses are fast to compute and provide a training signal based only on stereo frames and model predictions: focus on stereo depth estimation.
- The adaptation process requires training and should be as fast as possible: development of a lightweight stereo model (**MADNet**) and a fast approximated training strategy (**MAD**).

MADNet [c]

We design MADNet a new CNN for disparity estimation with speed and modularity as core design principles. Performance comparable with Dispnet but running at 40FPS.



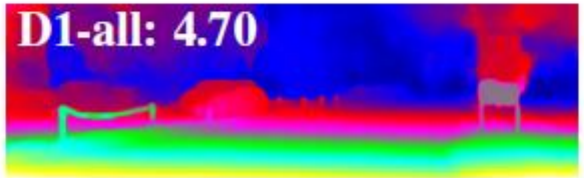
[c] **Alessio Tonioni**, Fabio Tosi, Matteo Poggi, Stefano Mattoccia and Luigi Di Stefano. “Real-time self-adaptive deep stereo”. CVPR2019 Oral.

MADNet [c]

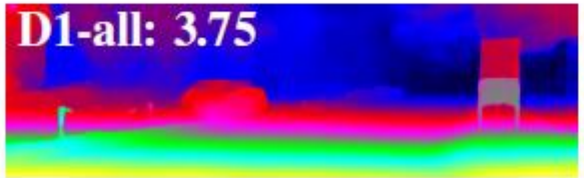
Model	D1-all	Runtime
DispNetC [19]	4.34	0.06
StereoNet [16]	4.83	0.02
MADNet (Ours)	4.66	0.02

- State of the art performance for fast architecture for stereo depth estimation.
- For an input resolution of 400X1200 the network can run at 40FPS on a 1080Ti GPU and 4FPS on a jetson TX2.

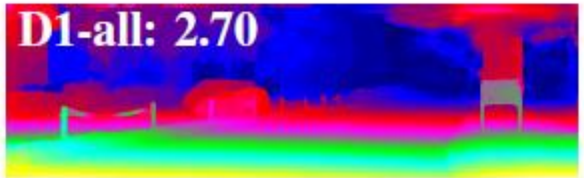

RGB



Dispnet



StereoNet



MADNet

Continous Online Adaptation [c]

- We propose to use self/proxy supervised losses to continuously fine-tune the network to the current domain, even at deployment time. Among the different losses we choose the reprojection loss as it is the fastest to compute.

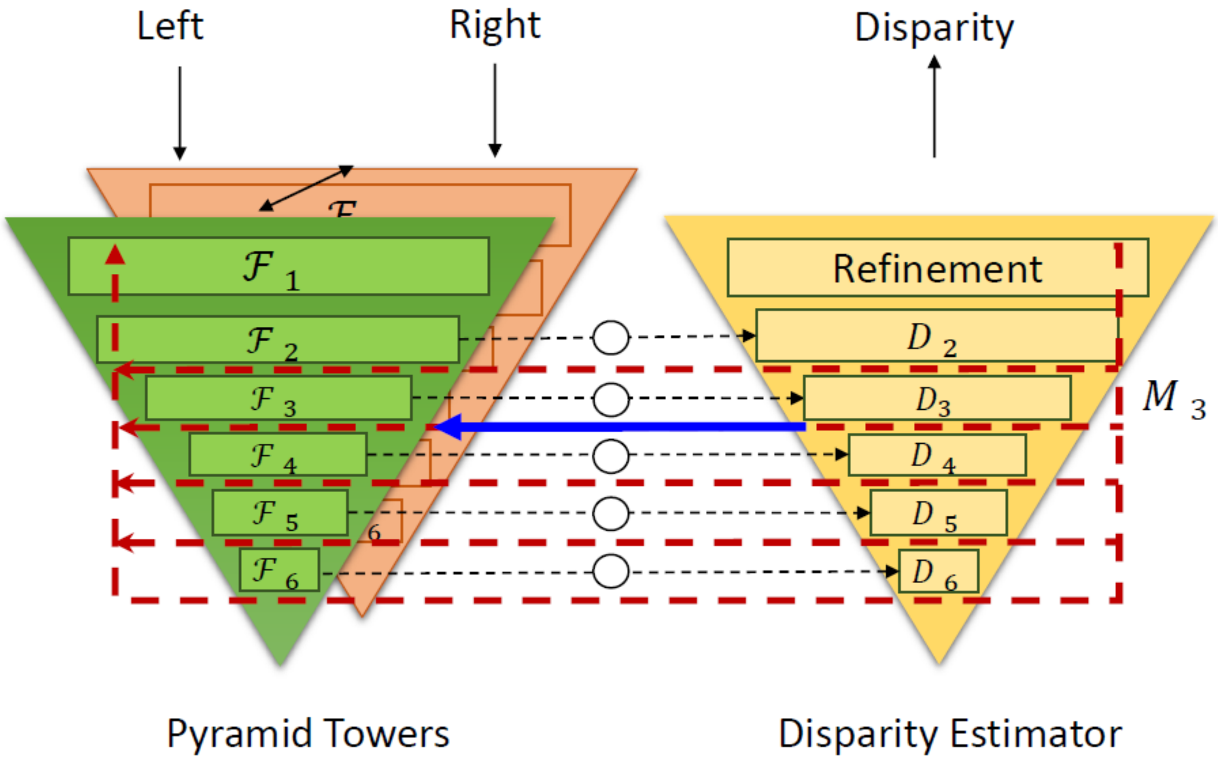
- **No clear distinction between train and test phases**, the network is always in training mode. Similar to [4] but here we wish to perform only fine tuning, not training from scratch.

- Continuously performing back propagation is computationally expensive. Experimentally we measured that a network performing online adaptation is roughly 3 times slower than the same network performing only inference.

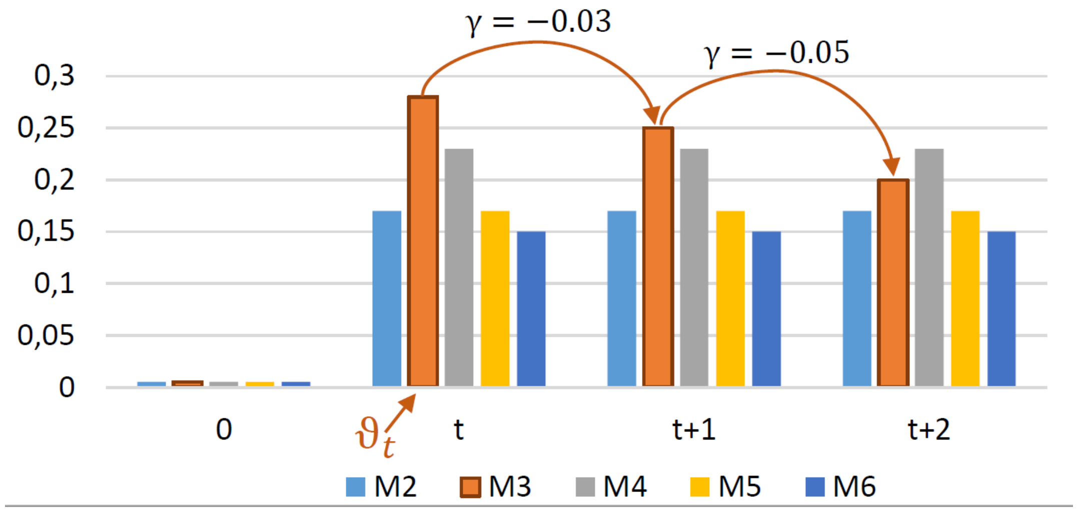
- <u>A fast network can help, but we need something more!</u>

4. Zhong, Yiran, Hongdong Li, and Yuchao Dai. "Open-world stereo video matching with deep rnn." *ECCV* 2018.

MAD

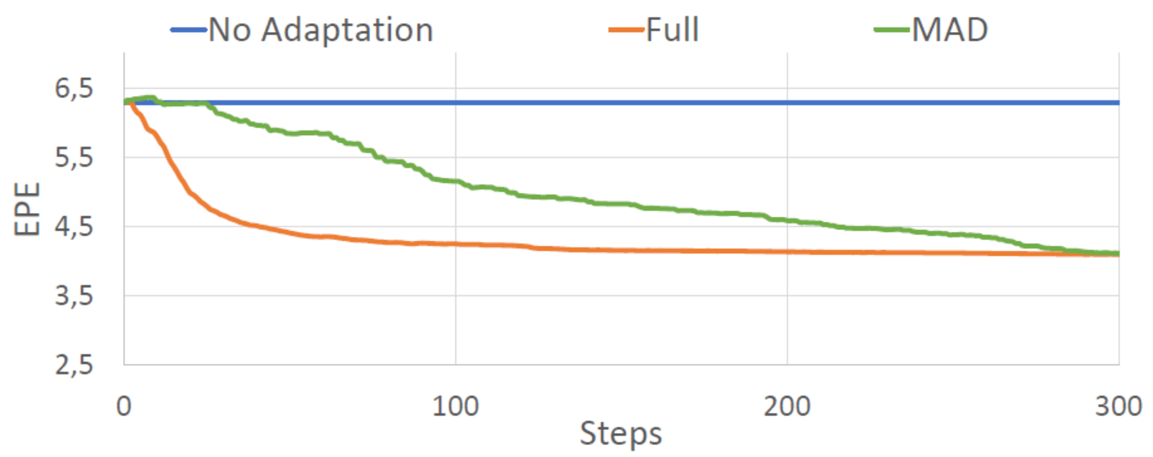


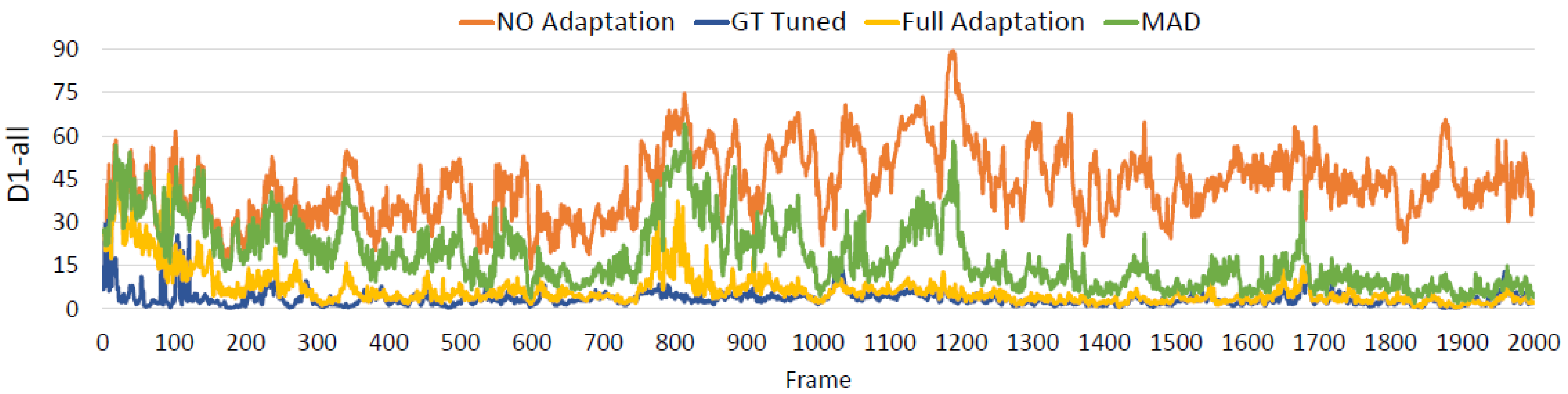
- At each iteration stochastic update of a separate portion of the network selected according to a reinforcement-based heuristic.
- Roughly 1/5 of the network is updated (1/5 of backprop) for each incoming stereo frame.



MAD Results

Model	Adapt.	D1-all(%)	EPE	FPS
DispNetC	No	9.09	1.58	15.85
DispNetC	Full	3.45	1.04	5.22
DispNetC-GT	No	4.40	1.21	15.85
MADNet	No	38.84	11.65	39.48
MADNet	Full	2.17	0.91	14.26
MADNet	*MAD*	3.37	1.11	25.43
MADNet-GT	No	2.67	0.89	39.48





Better safe than sorry

- Continuous online adaptation is very effective for deep stereo models but requires quite a lot of optimization steps before starting to improve the model.
- For some practical applications, the few seconds required by the online adaptation are still too much.

Learning to adapt for stereo [d]

Q: Can we train a deep stereo network to be more suitable to be adapted online?

- We propose **L2A** a *meta-learning* algorithm to find a good initial weight configuration suitable for online adaptation.
- We simulate at training time several online adaptations to different scenarios and optimize the initial weight configuration to obtain good performance across all domains.
- L2A is general and applicable to any deep stereo network.
- L2A find a good weight initialization, therefore, it does not affect online adaptation speed.

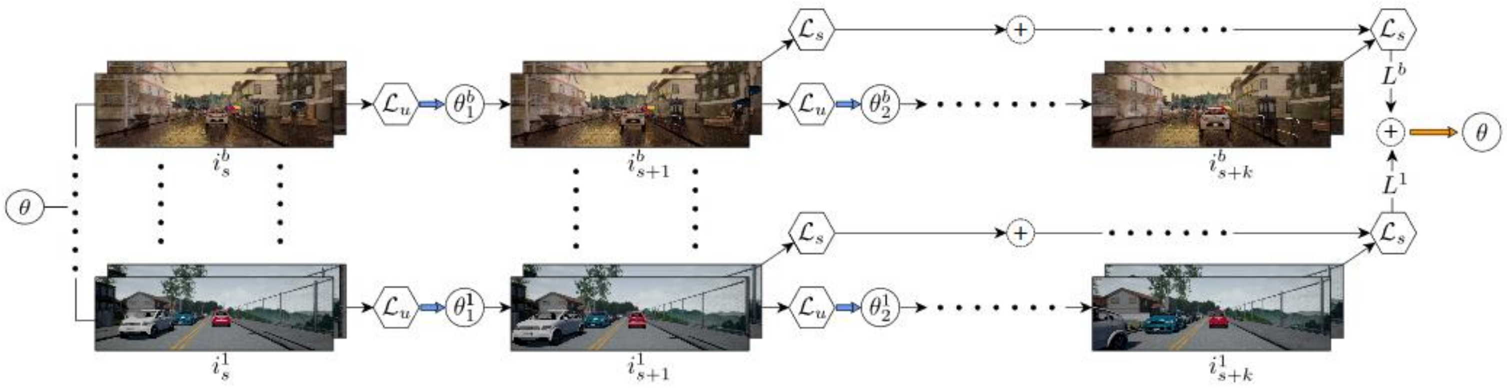
[d] **Alessio Tonioni**, Oscar Rahnama, Thomas Joy, Luigi Di Stefano, Thalaiyasingam Ajanthan, and Philip HS Torr, "*Learning to Adapt for Stereo*" CVPR 2019

Learning to Adapt via meta learning

Algorithm 1 Adaptation at training time for sequence $\mathcal{V}^\tau$

Require: $\theta, \mathcal{V}^\tau = [i_1^\tau, \dots, i_n^\tau]$

$\theta_0^\tau \leftarrow \theta$ ▷ Parameter initialization
for $t \leftarrow 1, \dots, n-1$ **do**
    $\theta_t^\tau \leftarrow \theta_{t-1}^\tau - \alpha \nabla_{\theta_{t-1}^\tau} \mathcal{L}_u \left(\theta_{t-1}^\tau, i_t\right)$ ▷ Adaptation
    $\mathcal{L}_s \left(\theta_t^\tau, i_{t+1}^\tau\right)$ ▷ Supervised evaluation



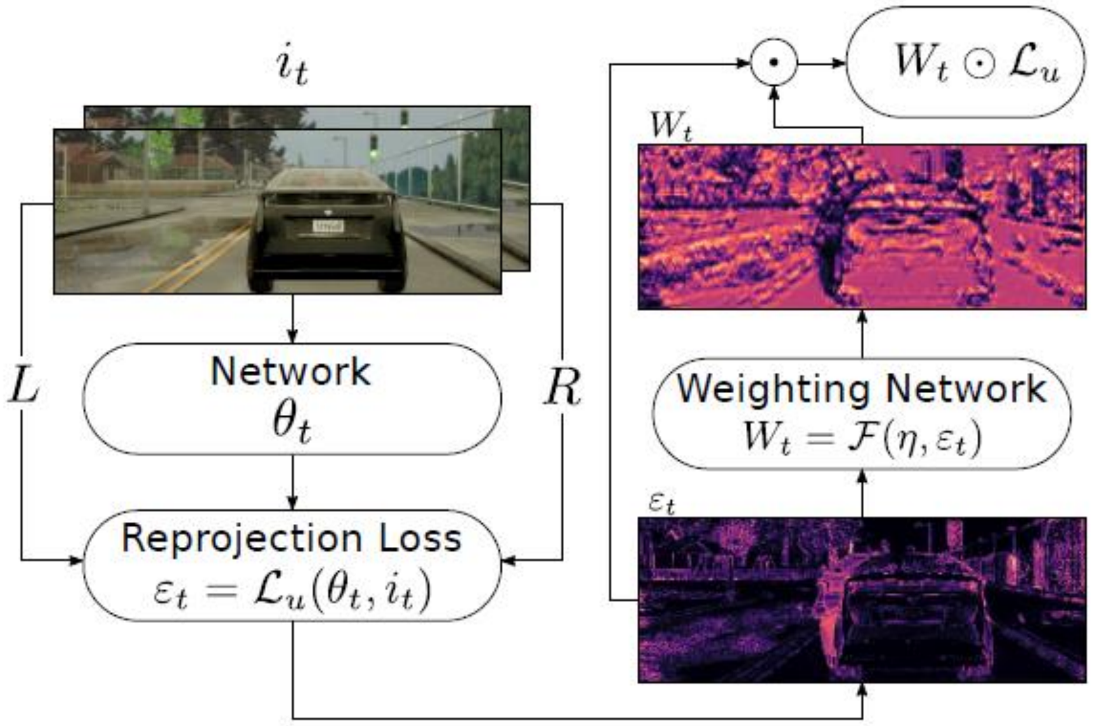
Algorithm 2 Learning to Adapt for Stereo

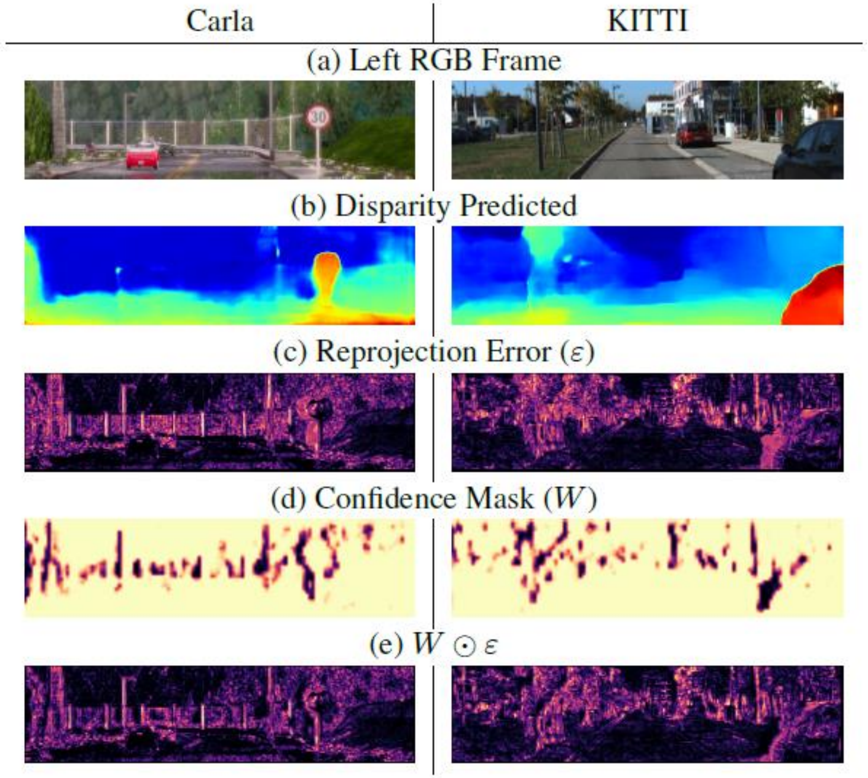
Require: Training set $\mathcal{D}_s$, and hyper-parameters α, β, k, b

Initialize θ
while *not done* **do**
    $\mathcal{D}^b \sim \mathcal{D}_s$ ▷ Sample a batch of sequences
    for all $\mathcal{V}^\tau \in \mathcal{D}^b$ **do**
        $\theta^\tau \leftarrow \theta$ ▷ Initialize model
        $L^\tau \leftarrow 0$ ▷ Initialize accumulator
        $[i_s, \dots, i_{s+k}] \sim \mathcal{V}^\tau$ ▷ Sample k frames
        for $t \leftarrow s, \dots, s+k-1$ **do**
            $\theta^\tau \leftarrow \theta^\tau - \alpha \nabla_{\theta^\tau} \mathcal{L}_u(\theta^\tau, i_t)$ ▷ Adaptation
            $L^\tau \leftarrow L^\tau + \mathcal{L}_s(\theta^\tau, i_{t+1})$ ▷ Evaluation
    $\theta \leftarrow \theta - \beta \nabla_\theta \sum_{\mathcal{V}^\tau \in \mathcal{D}^b} L^\tau$ ▷ Optimization

Learning to mask reprojection artifacts

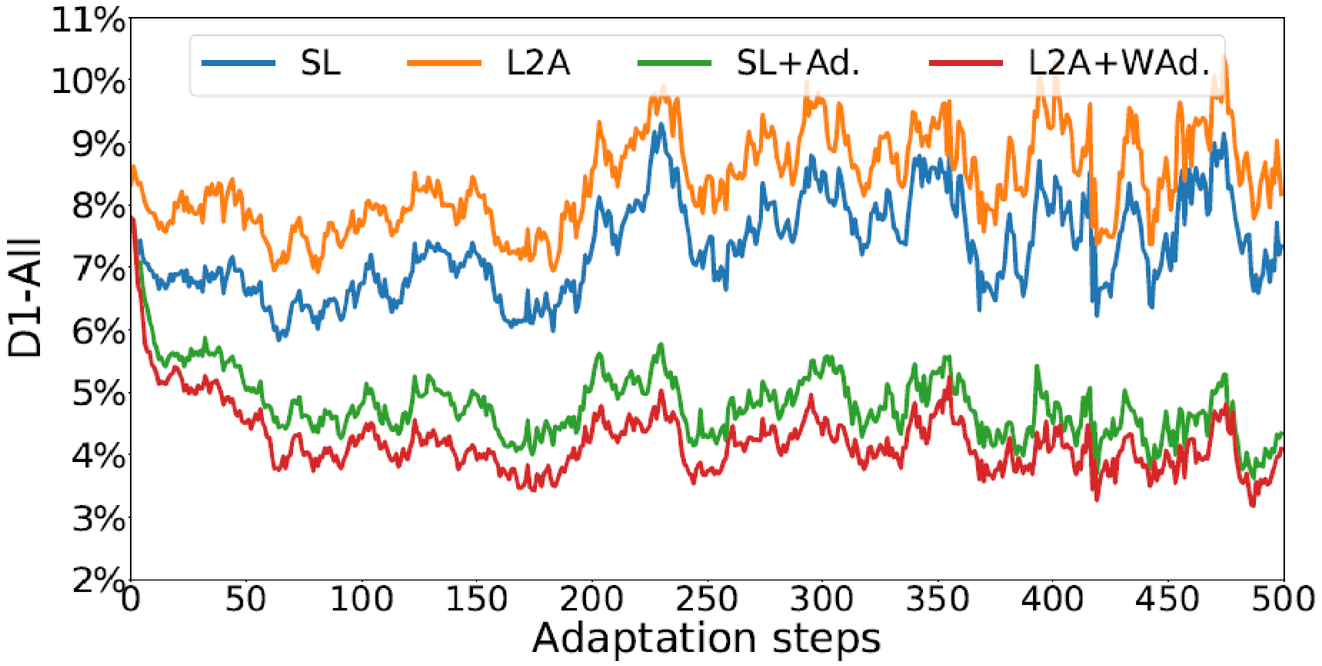
- Thanks to the meta-learning formulation we can additionally learn without supervision a confidence function that detects mistakes of the reprojection loss and mask them to improve online adaptation.





Results

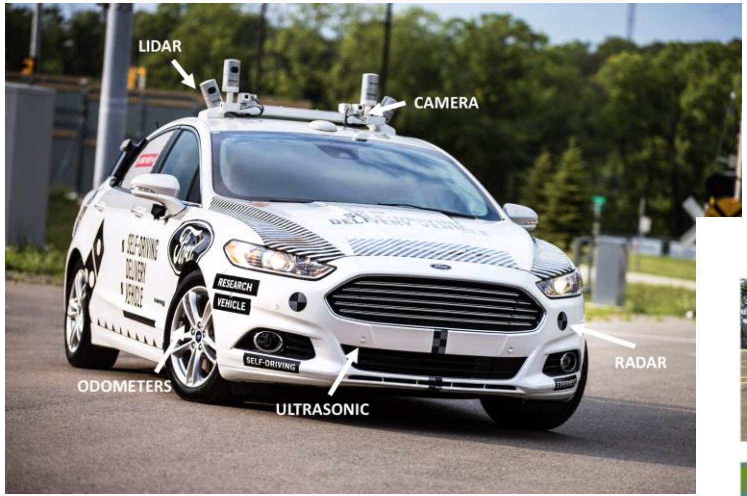
	Method	Training set	D1-all (%)	EPE	ΔD1	ΔEPE
	(a) **SL**	-	9.43	1.62	-	-
	(b) **SL+Ad**	-	7.81	1.44	-1.62	-0.18
	(c) **SL**	Carla	7.46	1.48	-	-
	(d) **SL+Ad**	Carla	5.26	1.20	-2.20	-0.28
	(e) **SL**	Synthia	8.55	1.51	-	-
	(f) **SL+Ad**	Synthia	5.33	1.19	-3.22	-0.32
Ours	(g) **L2A**	Carla	8.41	1.51	-	-
Ours	(h) **L2A+WAd**	Carla	**4.49**	**1.12**	**-3.92**	**-0.39**
Ours	(i) **L2A**	Synthia	8.22	1.50	-	-
Ours	(j) **L2A+WAd**	Synthia	4.65	1.14	-3.57	-0.36
	(k) **SL** (ideal)	KITTI	4.26	1.12	-	-

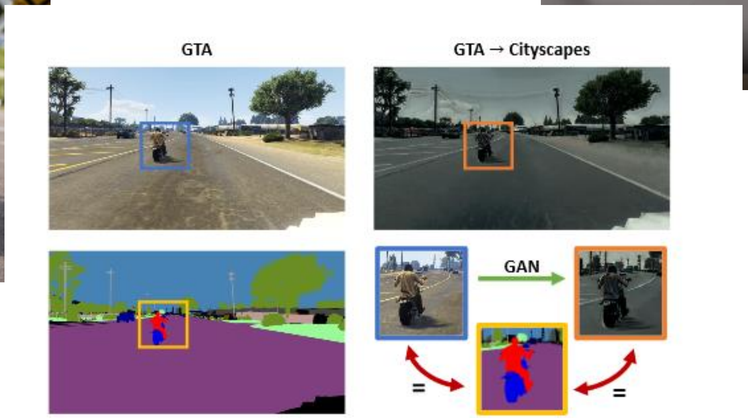


Conclusions & Future works

- Domain shifts can severally affects the performance of depth estimation models, but thanks to stereo geometry adaptation can be successfully performed without the need of annotation (and potentially online).

What's Next?





Thanks and Q&A


Matteo Poggi


Stefano Mattoccia


Luigi Di Stefano


Philip Torr


Thomas Joy


Fabio Tosi


Filippo Aleotti


Oscar Rahnama


Thalaiyasingam
Ajanthan