Deep Learning for Augmented Reality

Vincent Lepetit

École des Ponts
ParisTech

monocular depth prediction

We show qualitative results on the DAVIS dataset.

Images were processed individually frame-by-frame.
No temporal information was used in any way.

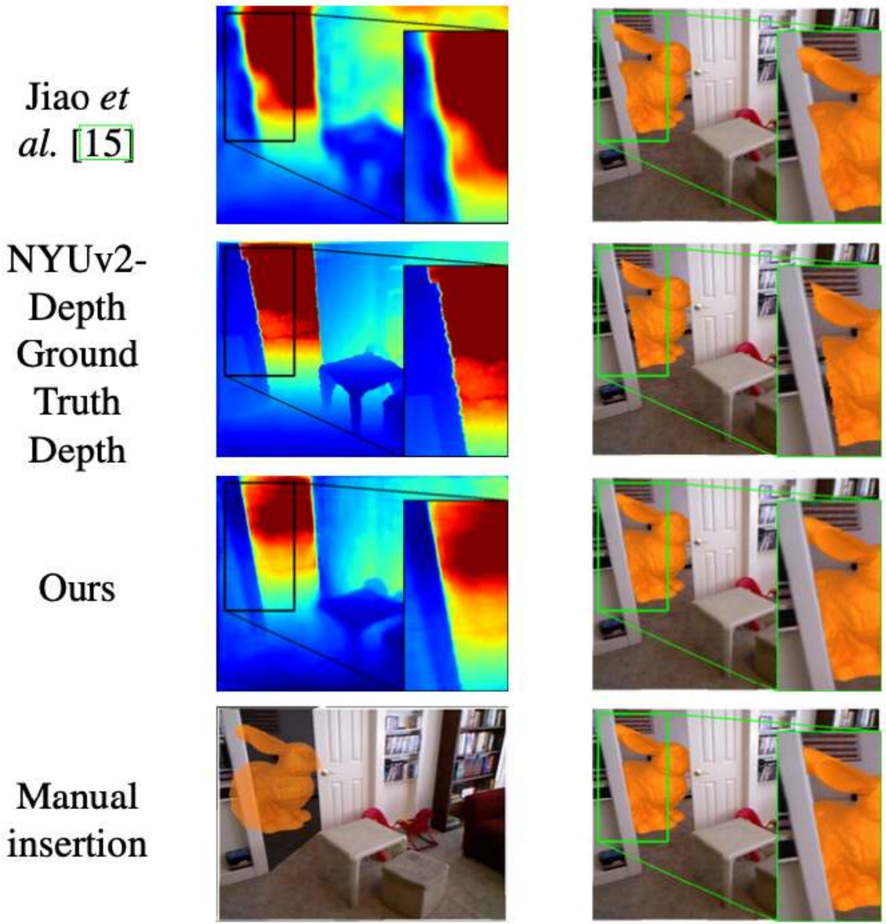
This is zero-shot cross-dataset transfer.
The DAVIS dataset was never seen during training.

MiDaS: Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-shot Cross-dataset Transfer. René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. PAMI 2020.

the problem

École des Ponts
ParisTech

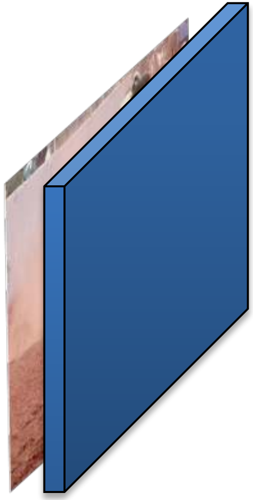
a simple application to AR



a possible architecture:
U-Net

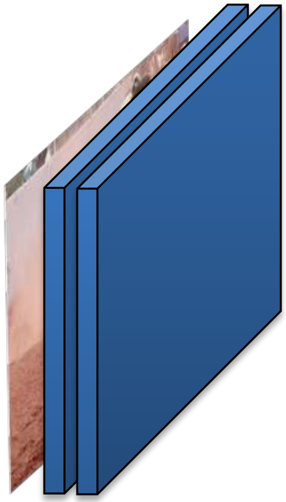
U-Net: Architecture

U-Net: Architecture



$$\mathbf{h}_1 = [g(\mathbf{f}_{1,1} * \mathbf{x}), \ldots, g(\mathbf{f}_{1,m} * \mathbf{x})]$$

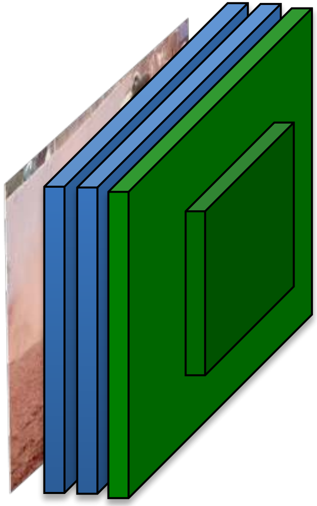
U-Net: Architecture



$$\mathbf{h}_1 = [g(\mathbf{f}_{1,1} * \mathbf{x}), \ldots, g(\mathbf{f}_{1,m} * \mathbf{x})]$$
$$\mathbf{h}_2 = [g(\mathbf{f}_{2,1} * \mathbf{h}_1), \ldots, g(\mathbf{f}_{2,m_2} * \mathbf{h}_1)]$$

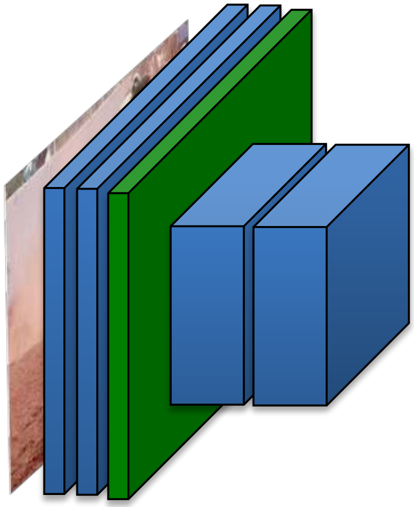
École des Ponts
ParisTech

U-Net: Architecture



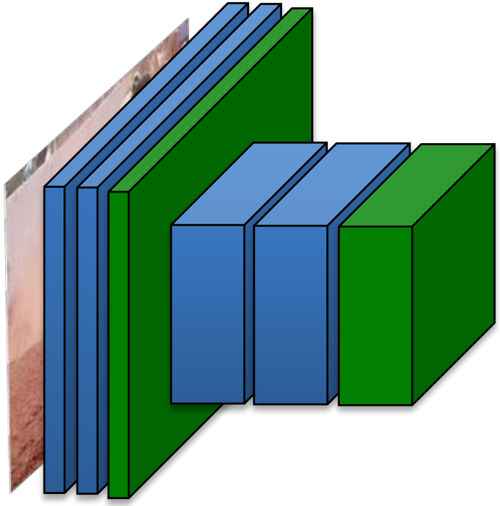
$$\mathbf{h}_1 = [g(\mathbf{f}_{1,1} * \mathbf{x}), \ldots, g(\mathbf{f}_{1,m} * \mathbf{x})]$$
$$\mathbf{h}_2 = [g(\mathbf{f}_{2,1} * \mathbf{h}_1), \ldots, g(\mathbf{f}_{2,m_2} * \mathbf{h}_1)]$$
$$\mathbf{h}_3 = \text{pooling}(\mathbf{h}_2)$$

U-Net: Architecture



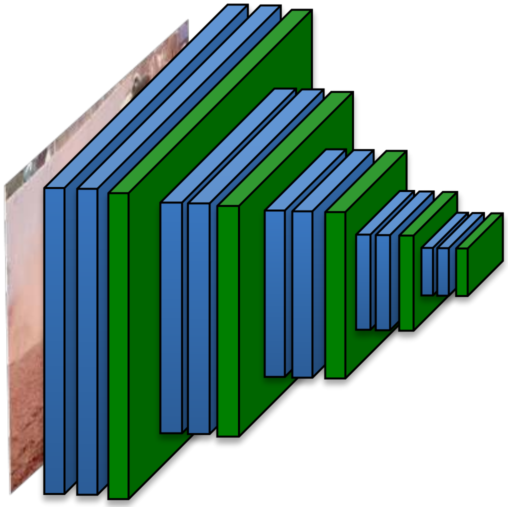
$$\mathbf{h}_1 = [g(\mathbf{f}_{1,1} * \mathbf{x}), \ldots, g(\mathbf{f}_{1,m} * \mathbf{x})]$$
$$\mathbf{h}_2 = [g(\mathbf{f}_{2,1} * \mathbf{h}_1), \ldots, g(\mathbf{f}_{2,m_2} * \mathbf{h}_1)]$$
$$\mathbf{h}_3 = \text{pooling}(\mathbf{h}_2)$$
$$\mathbf{h}_4 = [g(\mathbf{f}_{4,1} * \mathbf{h}_3), \ldots, g(\mathbf{f}_{4,m_4} * \mathbf{h}_3)]$$

U-Net: Architecture



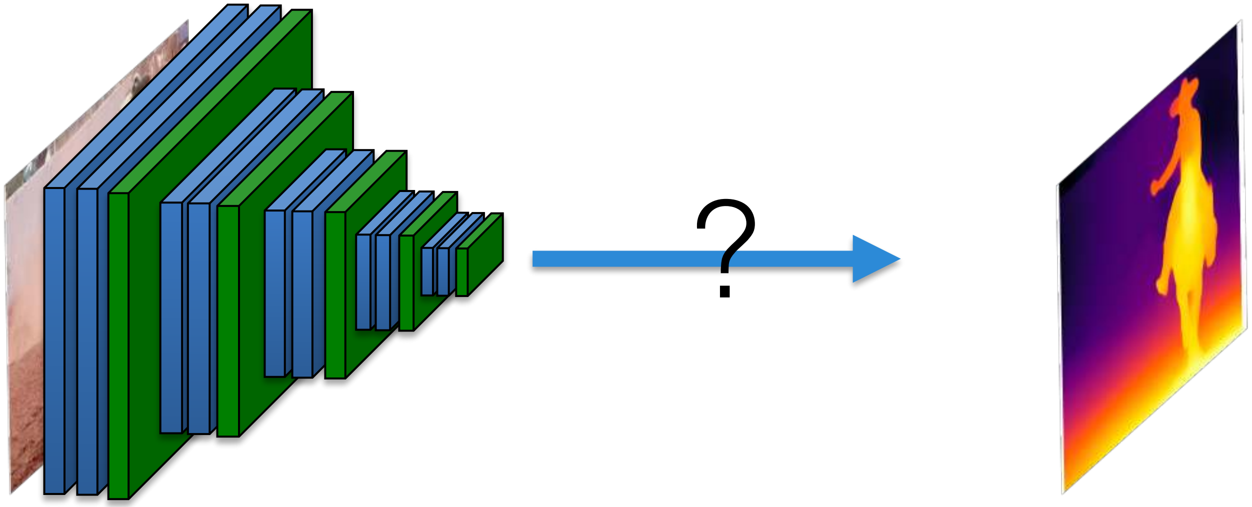
$$\mathbf{h}_1 = [g(\mathbf{f}_{1,1} * \mathbf{x}), \ldots, g(\mathbf{f}_{1,m} * \mathbf{x})]$$
$$\mathbf{h}_2 = [g(\mathbf{f}_{2,1} * \mathbf{h}_1), \ldots, g(\mathbf{f}_{2,m_2} * \mathbf{h}_1)]$$
$$\mathbf{h}_3 = \text{pooling}(\mathbf{h}_2)$$
$$\mathbf{h}_4 = [g(\mathbf{f}_{4,1} * \mathbf{h}_3), \ldots, g(\mathbf{f}_{4,m_4} * \mathbf{h}_3)]$$
$$\mathbf{h}_5 = [g(\mathbf{f}_{5,1} * \mathbf{h}_4), \ldots, g(\mathbf{f}_{5,m_5} * \mathbf{h}_4)]$$
$$\mathbf{h}_6 = \text{pooling}(\mathbf{h}_5)$$

U-Net: Architecture

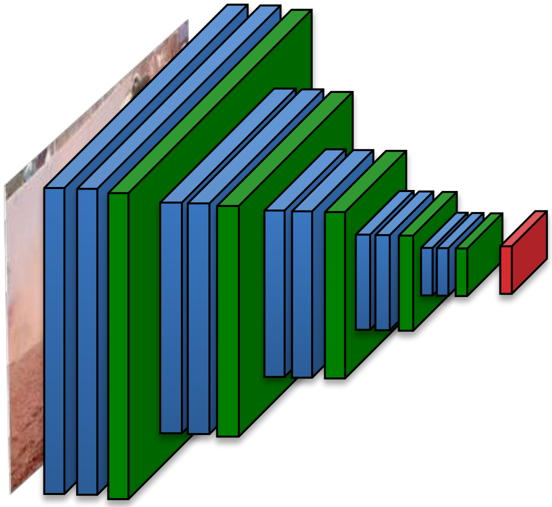


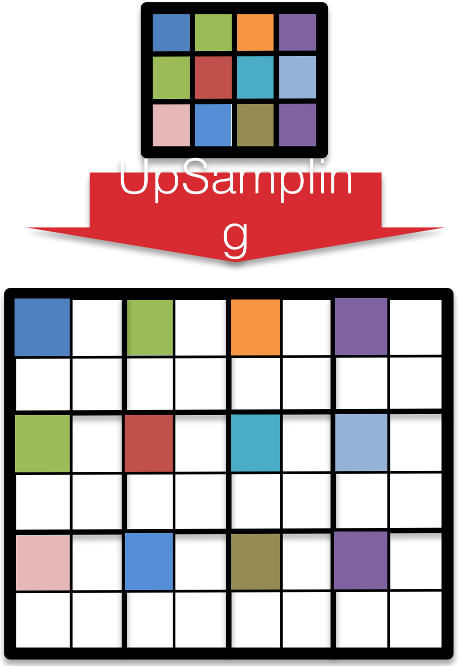
$$\mathbf{h}_{13} = [g(\mathbf{f}_{13,1} * \mathbf{h}_{12}), \ldots, g(\mathbf{f}_{13,m_{13}} * \mathbf{h}_{12})]$$
$$\mathbf{h}_{14} = [g(\mathbf{f}_{14,1} * \mathbf{h}_{13}), \ldots, g(\mathbf{f}_{14,m_{14}} * \mathbf{h}_{13})]$$
$$\mathbf{h}_{15} = \text{pooling}(\mathbf{h}_{14})$$

U-Net: Architecture



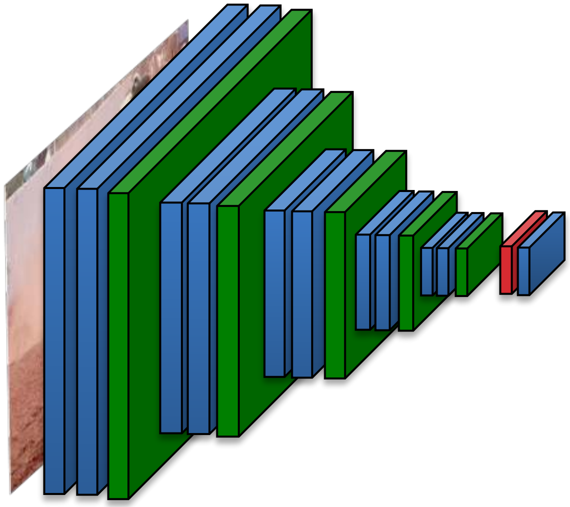
U-Net: Architecture

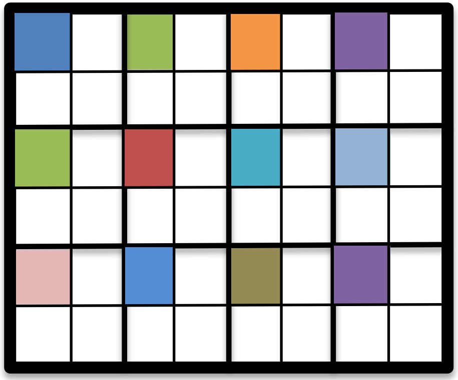




$$\mathbf{h}_{16} = \mathrm{UpSampling}(\mathbf{h}_{15})$$

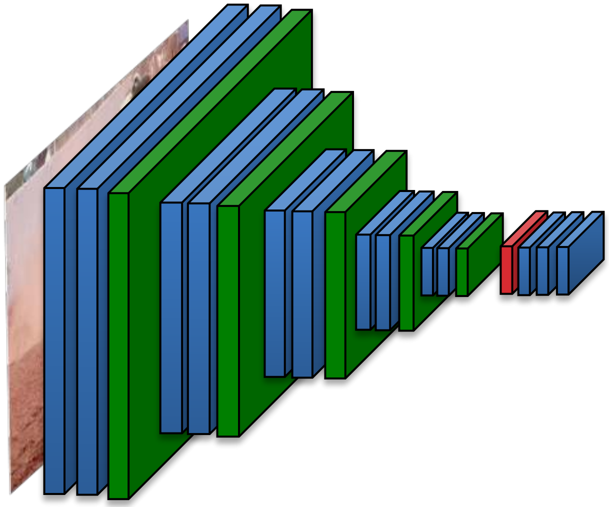
U-Net: Architecture





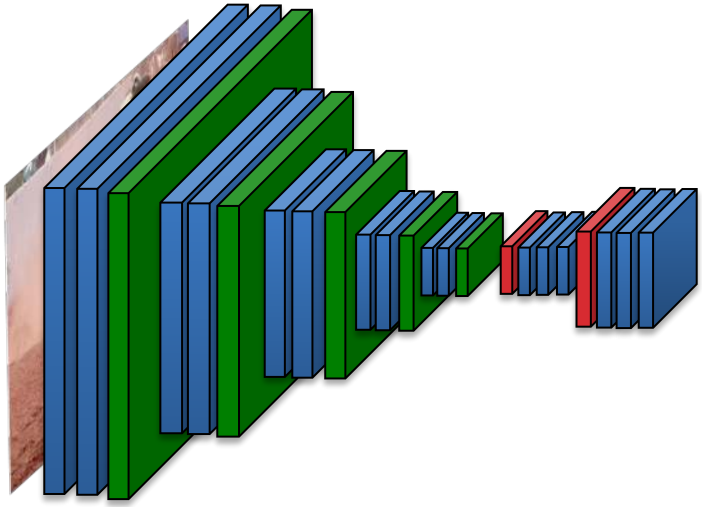
$$\mathbf{h}_{16} = \mathrm{UpSampling}(\mathbf{h}_{15})$$
$$\mathbf{h}_{17} = [g(\mathbf{f}_{17,1} * \mathbf{h}_{16}), \ldots, g(\mathbf{f}_{17,m_{17}} * \mathbf{h}_{16})]$$

U-Net: Architecture

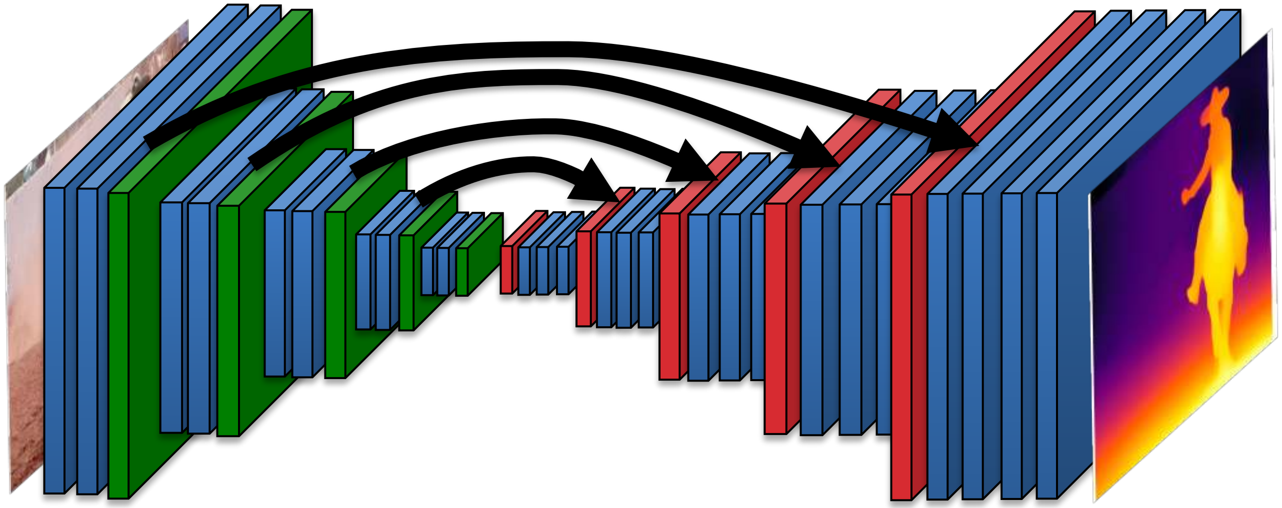


$$\mathbf{h}_{16} = \mathrm{UpSampling}(\mathbf{h}_{15})$$
$$\mathbf{h}_{17} = [g(\mathbf{f}_{17,1} * \mathbf{h}_{16}), \ldots, g(\mathbf{f}_{17,m_{17}} * \mathbf{h}_{16})]$$
$$\mathbf{h}_{18} = [g(\mathbf{f}_{18,1} * \mathbf{h}_{17}), \ldots, g(\mathbf{f}_{18,m_{18}} * \mathbf{h}_{17})]$$

U-Net: Architecture



U-Net: Skip Connections



The loss can be, for example (but see next slides):

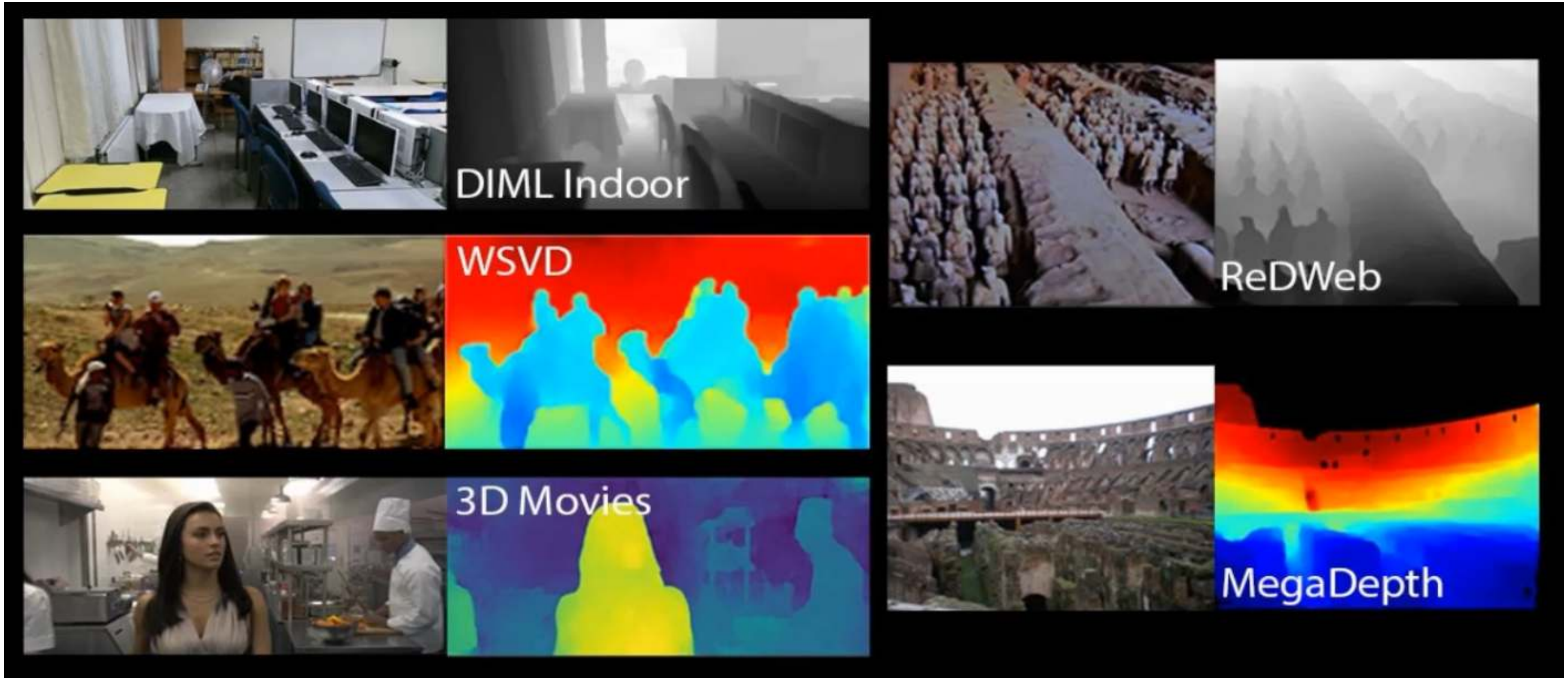
$$\mathcal{L}(\Theta) = \|D - \hat{D}\|^2$$

$$\mathcal{L}(\Theta) = |D - \hat{D}|$$

$$\mathcal{L}(\Theta) = \sum_i |\log D_i - \log \hat{D}_i|$$ (depth defined up to a scale factor)

École des Ponts ParisTech

dataset used by MiDaS

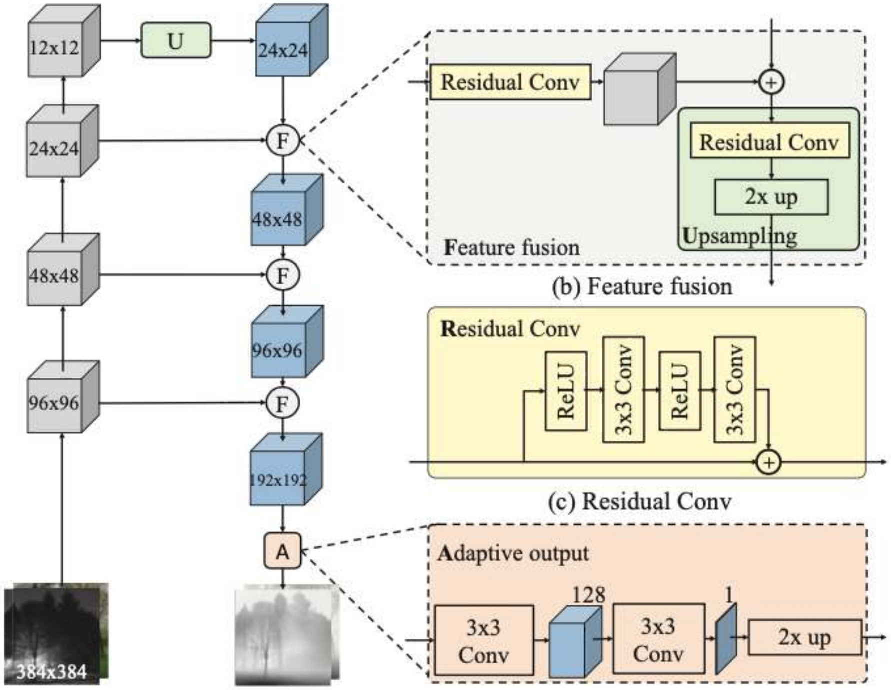


École des Ponts
ParisTech

dataset used by MiDaS

Dataset	Indoor	Outdoor	Dynamic	Video	Dense	Accuracy	Diversity	Annotation	Depth	# Images
DIML Indoor [31]	✓			✓	✓	Medium	Medium	RGB-D	**Metric**	220K
MegaDepth [11]		✓	(✓)		(✓)	Medium	Medium	SfM	No scale	130K
ReDWeb [32]	✓	✓	✓		✓	Medium	**High**	Stereo	No scale & shift	3600
WSVD [33]	✓	✓	✓	✓	✓	Medium	**High**	Stereo	No scale & shift	1.5M
3D Movies	✓	✓	✓	✓	✓	Medium	**High**	Stereo	No scale & shift	75K
DIW [34]	✓	✓	✓			Low	**High**	User clicks	Ordinal pair	496K
ETH3D [35]	✓	✓			✓	**High**	Low	Laser	**Metric**	454
Sintel [36]	✓	✓	✓	✓	✓	**High**	Medium	Synthetic	(Metric)	1064
KITTI [28], [29]		✓	(✓)	✓	(✓)	Medium	Low	Laser/Stereo	**Metric**	93K
NYUDv2 [30]	✓		(✓)	✓	✓	Medium	Low	RGB-D	**Metric**	407K
TUM-RGBD [37]	✓		(✓)	✓	✓	Medium	Low	RGB-D	**Metric**	80K

architecture used by MiDaS



(a) Proposed network

(b) Feature fusion

(c) Residual Conv

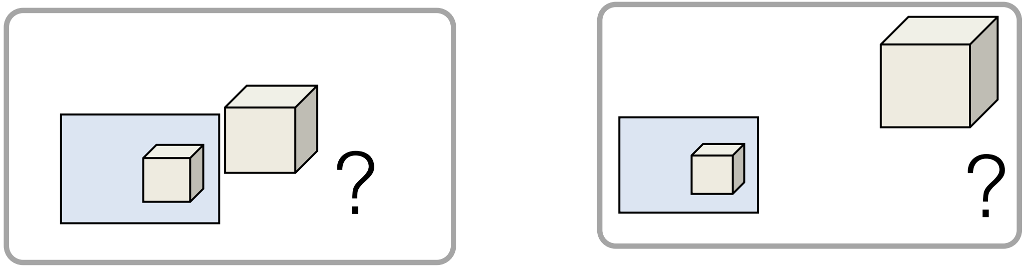
(d) Adaptive output

Monocular Relative Depth Perception with Web Stereo Dat. Supervision. Ke Xian, Chunhua Shen, Zhiguo Cao, Hao Lu, Yang Xiao, Ruibo Li, Zhenbo Luo. CVPR 2018.

MiDaS

Uses the disparity: $\mathbf{d} = \frac{1}{D}$

Depth is predicted up to a scale factor.



MiDaS

Uses the disparity: $\mathbf{d} = \dfrac{1}{D}$

Or, more exactly, the scale- and shift-independent disparities:

$$(s, t) = \arg\min_{s,t} \sum_{i=1}^{M} (s\mathbf{d}_i + t - \mathbf{d}_i^*)^2 \ ,$$

$$\hat{\mathbf{d}} = s\mathbf{d} + t, \quad \hat{\mathbf{d}}^* = \mathbf{d}^*,$$

or (more robust):

$$t(\mathbf{d}) = \text{median}(\mathbf{d}), \quad s(\mathbf{d}) = \frac{1}{M}\sum_{i=1}^{M} |\mathbf{d} - t(\mathbf{d})|$$

$$\hat{\mathbf{d}} = \frac{\mathbf{d} - t(\mathbf{d})}{s(\mathbf{d})}, \quad \hat{\mathbf{d}}^* = \frac{\mathbf{d}^* - t(\mathbf{d}^*)}{s(\mathbf{d}^*)}$$

MiDaS

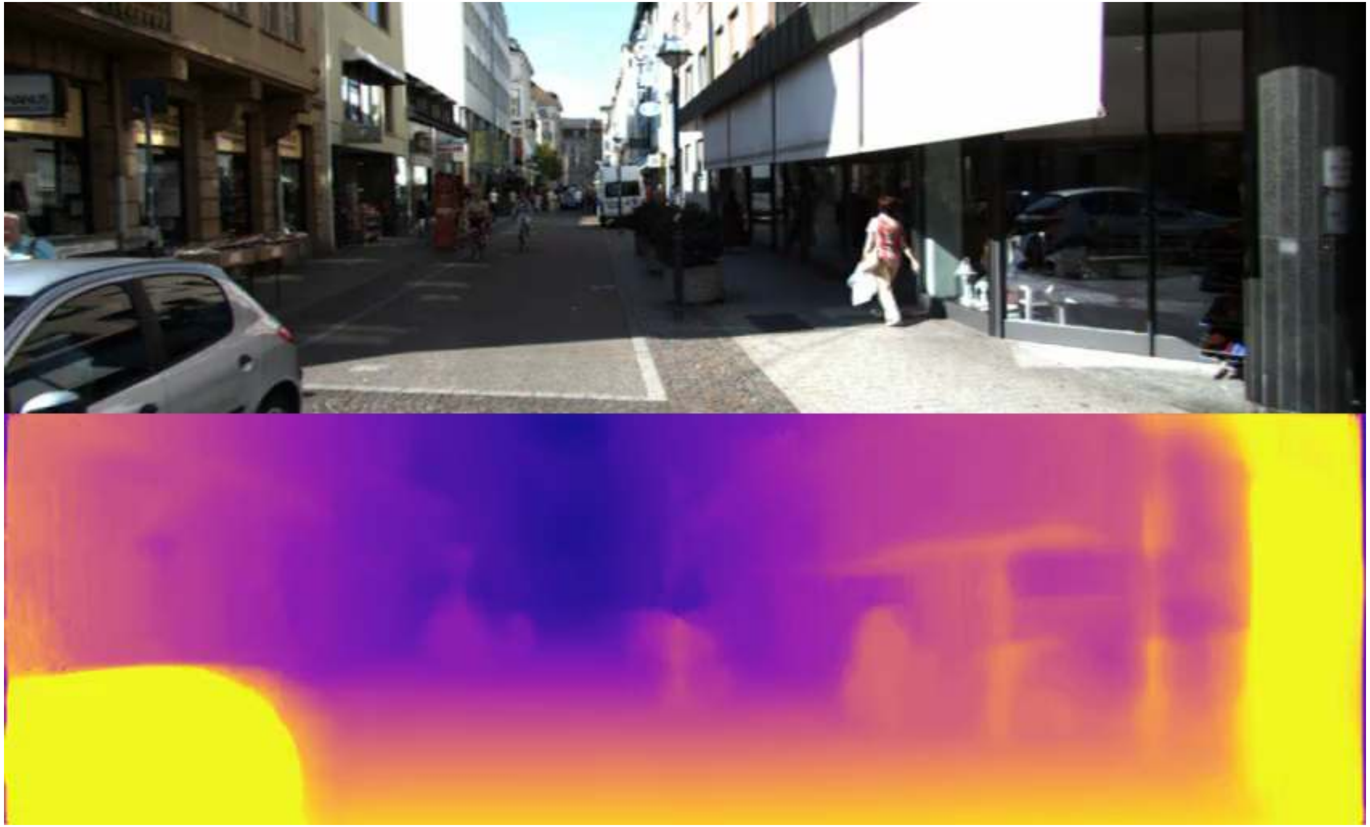
Loss: $$\mathcal{L}_{ssi}(\hat{\mathbf{d}}, \hat{\mathbf{d}}^*) = \frac{1}{2M} \sum_{i=1}^{M} \rho\left(\hat{\mathbf{d}}_i - \hat{\mathbf{d}}_i^*\right)$$

(and variants)

Regularization: $$\mathcal{L}_{reg}(\hat{\mathbf{d}}, \hat{\mathbf{d}}^*) = \frac{1}{M} \sum_{k=1}^{K} \sum_{i=1}^{M} \left(|\nabla_x R_i^k| + |\nabla_y R_i^k|\right) \qquad \text{where } R_i = \hat{\mathbf{d}}_i - \hat{\mathbf{d}}_i^*$$

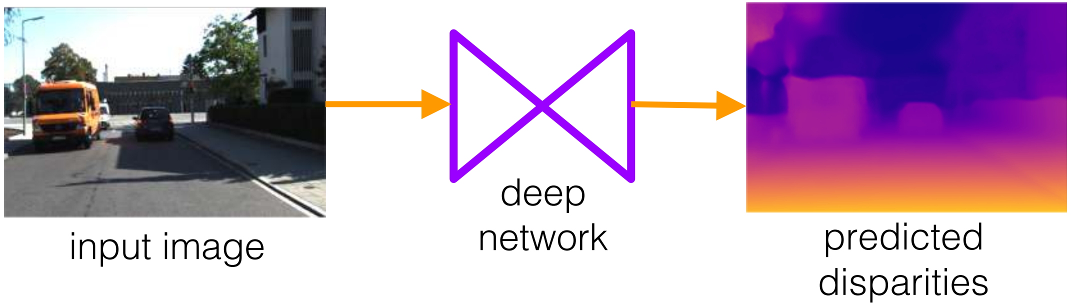
some failure cases

unsupervised depth prediction

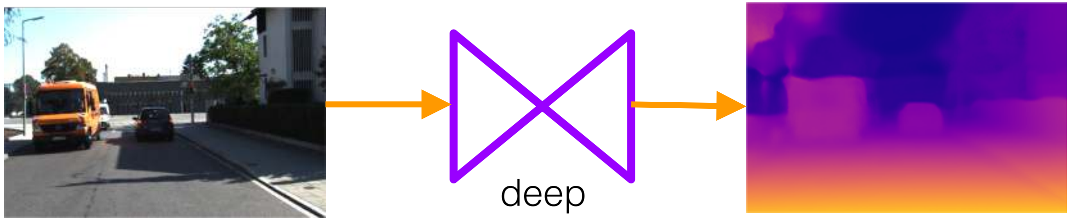


Unsupervised Monocular Depth Estimation with Left-Right Consistency. Clément Godard Oisin Mac Aodha Gabriel J. Brostow. CVPR 2017.

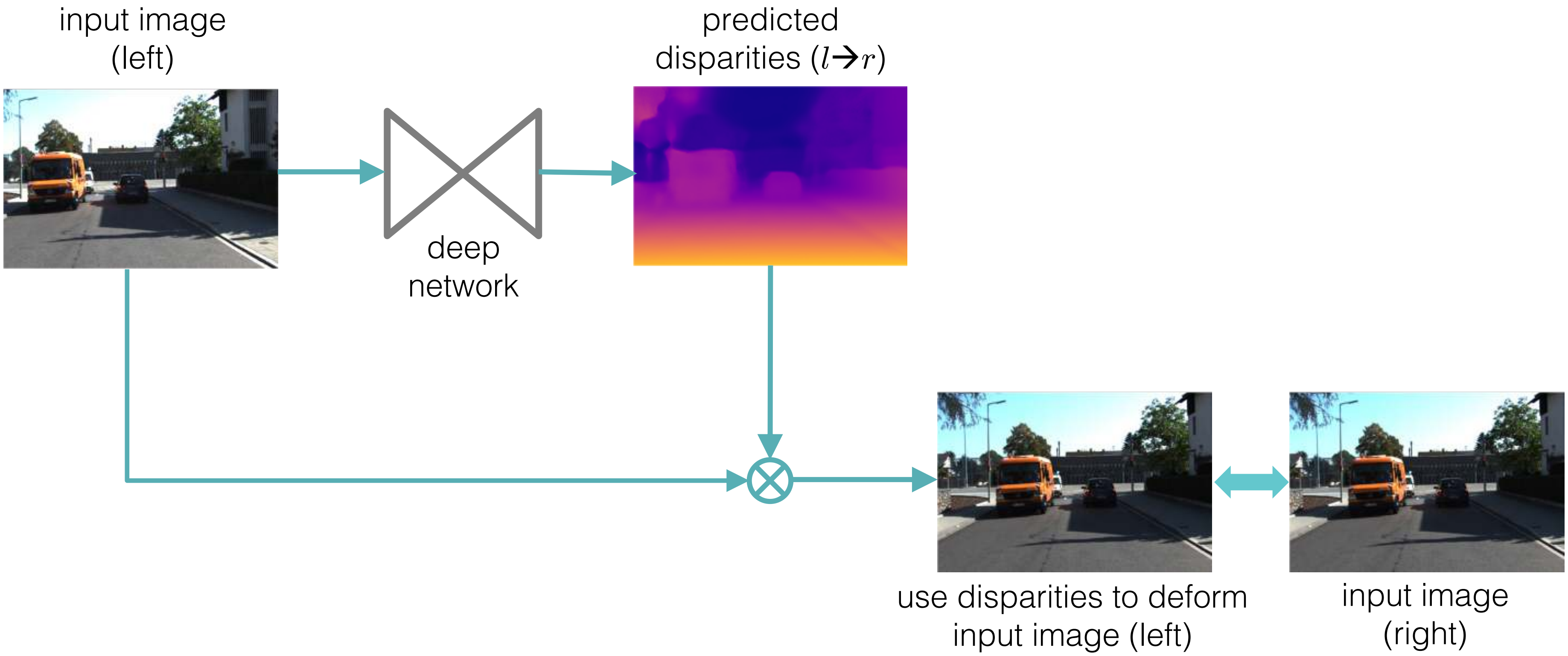
Unsupervised depth estimation



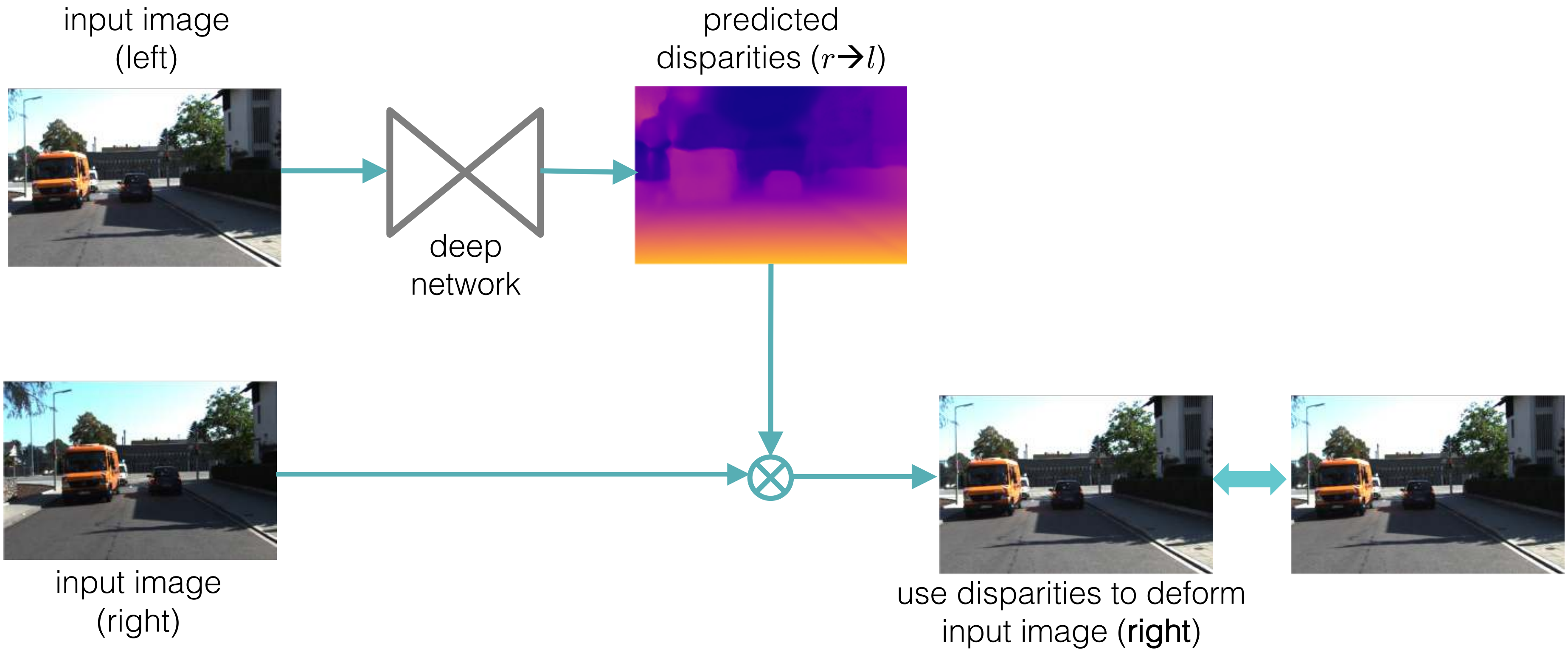
Unsupervised depth estimation



unsupervised depth estimation (naive)

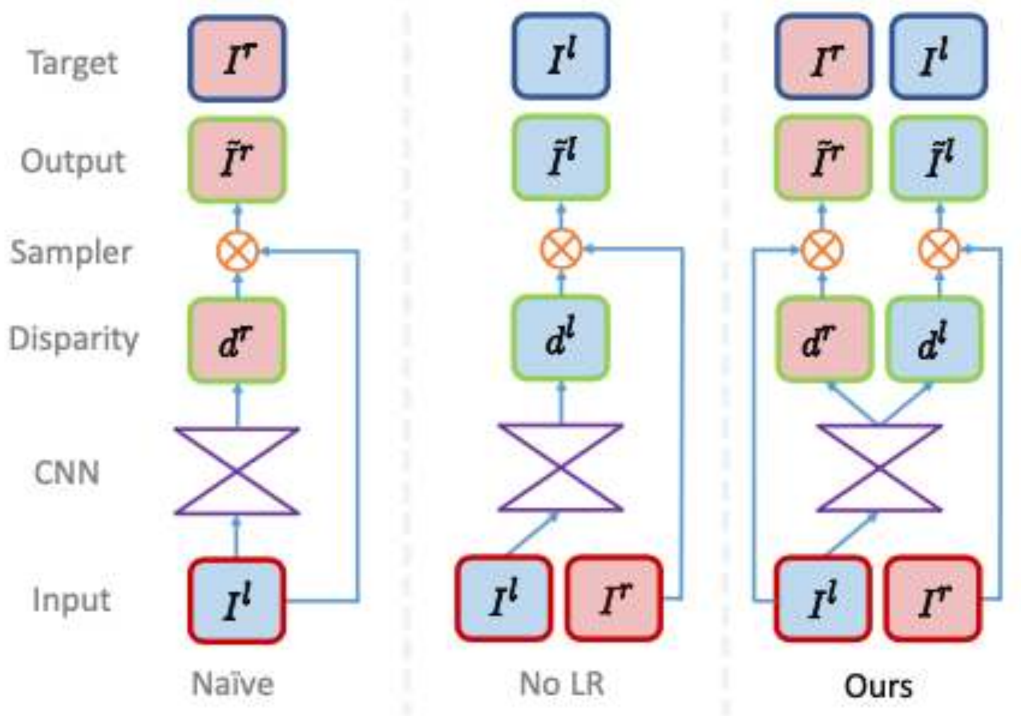


unsupervised depth estimation



École des Ponts

unsupervised depth estimation



École des Ponts
ParisTech

unsupervised depth estimation

$$C_s = \alpha_{ap}(C^l_{ap} + C^r_{ap}) + \alpha_{ds}(C^l_{ds} + C^r_{ds}) + \alpha_{lr}(C^l_{lr} + C^r_{lr})$$

$$C^l_{ap} = \frac{1}{N}\sum_{i,j}\alpha\frac{1 - \mathrm{SSIM}(I^l_{ij}, \tilde{I}^l_{ij})}{2} + (1-\alpha)\left\|I^l_{ij} - \tilde{I}^l_{ij}\right\|$$

$$C^l_{ds} = \frac{1}{N}\sum_{i,j}\left|\partial_x d^l_{ij}\right|e^{-\left\|\partial_x I^l_{ij}\right\|} + \left|\partial_y d^l_{ij}\right|e^{-\left\|\partial_y I^l_{ij}\right\|}$$

$$C^l_{lr} = \frac{1}{N}\sum_{i,j}\left|d^l_{ij} - d^r_{ij+d^l_{ij}}\right|$$

$$\mathrm{SSIM}(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

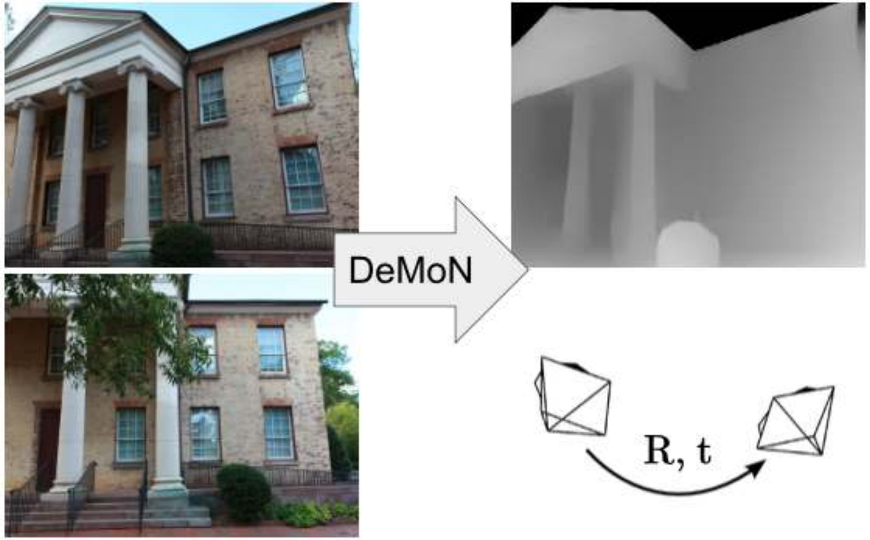
$$\mathrm{SSIM}(x,y) = \left[l(x,y)^\alpha \cdot c(x,y)^\beta \cdot s(x,y)^\gamma\right]$$

$$l(x,y) = \frac{2\mu_x\mu_y + c_1}{\mu_x^2 + \mu_y^2 + c_1}$$

$$c(x,y) = \frac{2\sigma_x\sigma_y + c_2}{\sigma_x^2 + \sigma_y^2 + c_2}$$

$$s(x,y) = \frac{\sigma_{xy} + c_3}{\sigma_x\sigma_y + c_3}$$

two-image 3D geometry recovery



DeMoN: Depth and Motion Network for Learning Monocular Stereo. Benjamin Ummenhofer, Huizhong Zhou, Jonas Uhrig, Nikolaus Mayer, Eddy Ilg, Alexey Dosovitskiy, Thomas Brox. CVPR 2017.

École des Ponts
ParisTech

DeMoN: Depth and Motion Network for Learning Monocular Stereo

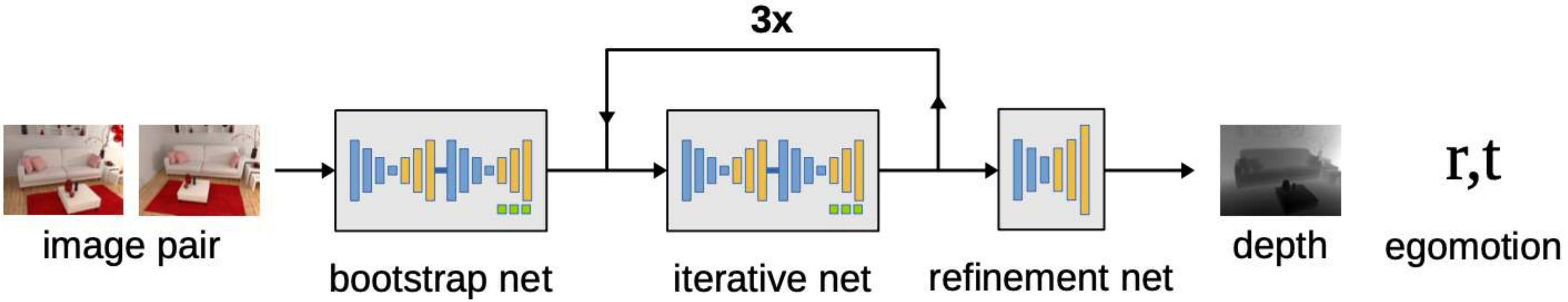
Benjamin Ummenhofer[*,1] Huizhong Zhou[*,1] Jonas Uhrig[1,2]

Nikolaus Mayer[1] Eddy Ilg[1] Alexey Dosovitskiy[1] Thomas Brox[1]

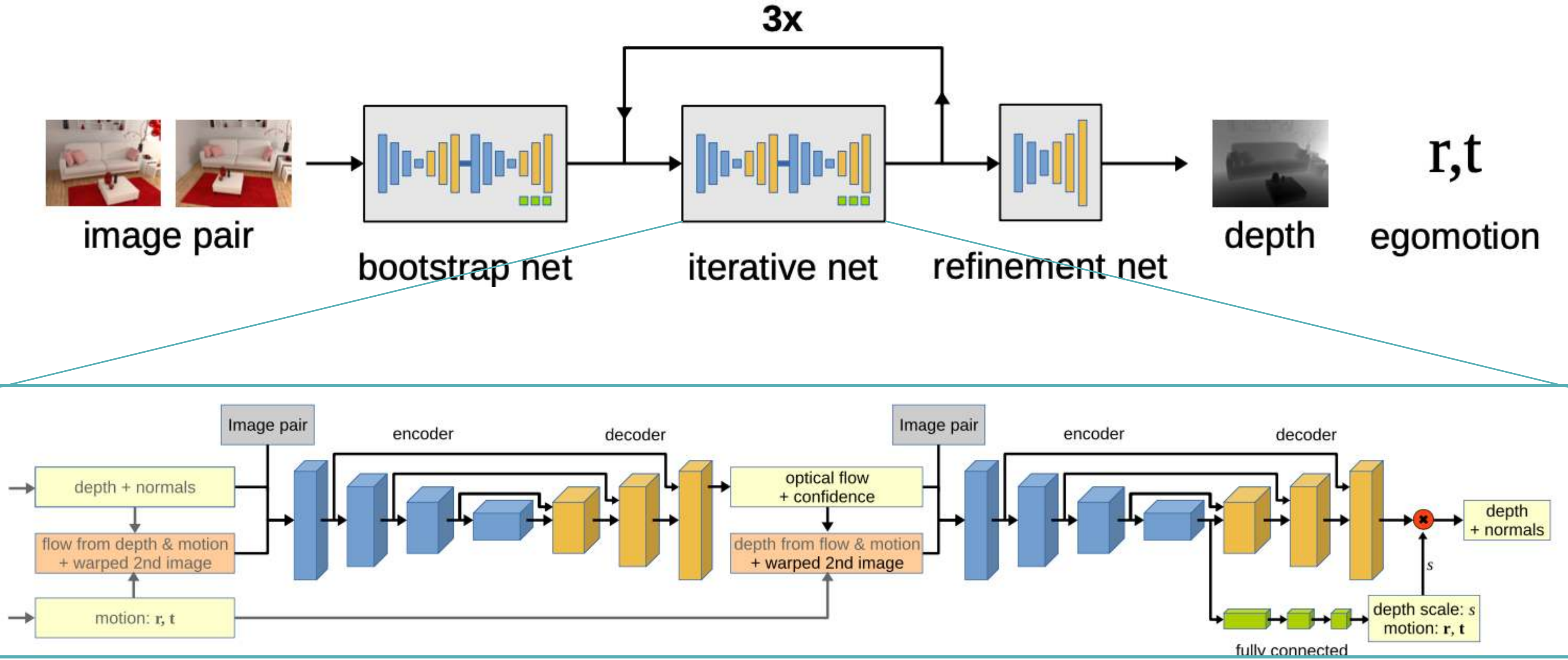
[1]University of Freiburg [2]Daimler AG R&D

*equal contribution

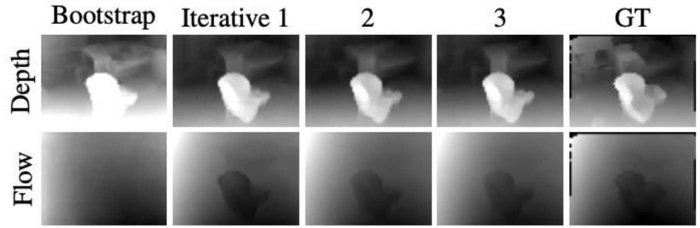
DeMoN architecture

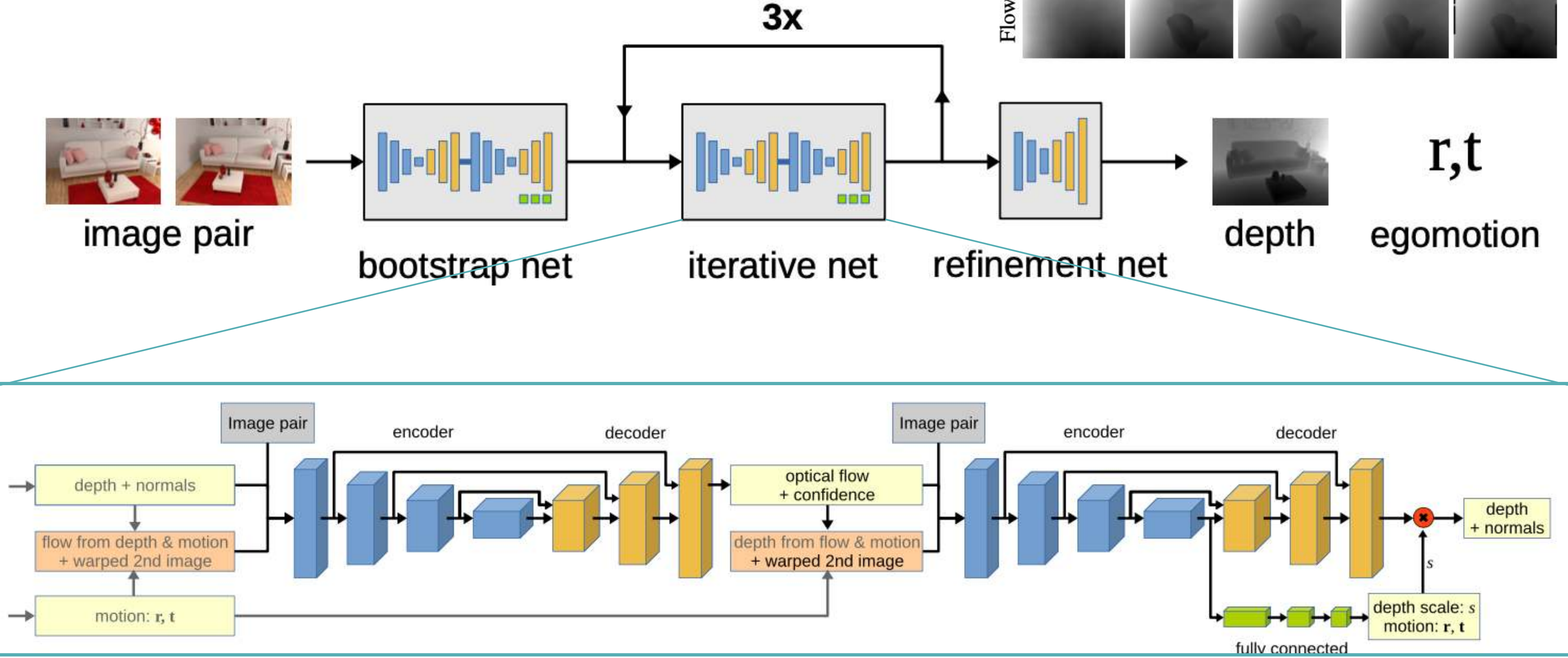


DeMoN architecture



DeMoN architecture





DeMoN loss function

$$\mathcal{L}_{\text{depth}} = \sum_{i,j} |s\xi(i,j) - \hat{\xi}(i,j)|$$

$$\mathcal{L}_{\text{normal}} = \sum_{i,j} \|\mathbf{n}(i,j) - \hat{\mathbf{n}}(i,j)\|_2$$
$$\mathcal{L}_{\text{flow}} = \sum_{i,j} \|\mathbf{w}(i,j) - \hat{\mathbf{w}}(i,j)\|_2$$

$$\mathcal{L}_{\text{rotation}} = \|\mathbf{r} - \hat{\mathbf{r}}\|_2$$
$$\mathcal{L}_{\text{translation}} = \|\mathbf{t} - \hat{\mathbf{t}}\|_2$$

$$\mathcal{L}_{\text{grad}\,\xi} = \sum_{h \in \{1,2,4,8,16\}} \sum_{i,j} \left\| \mathbf{g}_h[\xi](i,j) - \mathbf{g}_h[\hat{\xi}](i,j) \right\|_2$$

$$\mathbf{g}_h[f](i,j) = \left(\frac{f(i+h,j)-f(i,j)}{|f(i+h,j)|+|f(i,j)|}, \frac{f(i,j+h)-f(i,j)}{|f(i,j+h)|+|f(i,j)|} \right)^\top$$

École des Ponts
ParisTech