Machine Learning Based approaches for Optical Flow and Depth Prediction

Antoine Manzanera

ENSTA Paris

M2 IMA - Computer Vision
December 2023

Learning to predict optical flow and 3d?

Limitations of analytical methods

Optical flow and 3d reconstruction are strongly ill-posed problems:

- Sensitive to untextured areas.
- Sensitive to displacement / distance ranges.
- Subject to numerical problems (FoE / epipole, Matched points precision, Noise, Outliers, . . .).

Learning-based methods

Learning based methods have the potential to exploit all motion / 3d cues, by jointly modelling different related concepts: geometric, photometric, dynamic, and semantic!

Presentation Outline

1. Reminder on Neural Networks
2. Learning-based Optical Flow prediction
    - Supervised methods
    - Self-supervised methods
3. Learning-based Depth prediction
    - Reminder on Depth Cues
    - Fully supervised methods
    - Self-supervised methods
4. Conclusion

Presentation Outline

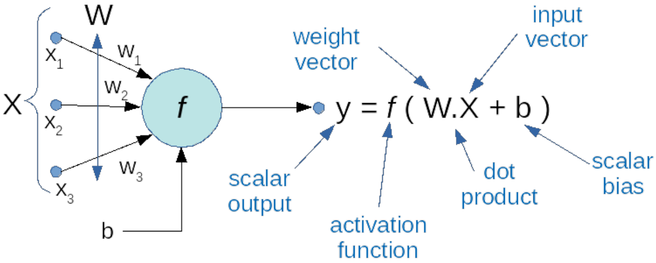
1. Reminder on Neural Networks

2. Learning-based Optical Flow prediction
   - Supervised methods
   - Self-supervised methods

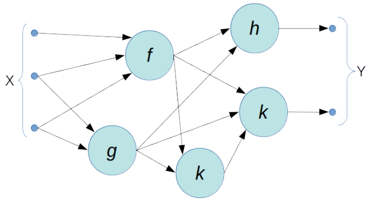
3. Learning-based Depth prediction
   - Reminder on Depth Cues
   - Fully supervised methods
   - Self-supervised methods

4. Conclusion

Formal Neuron Model



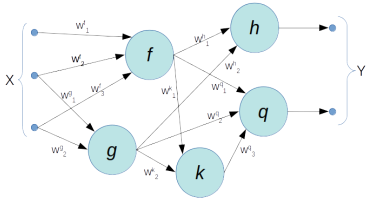
Neural Networks



A formal Neural Network is an oriented graph, where:

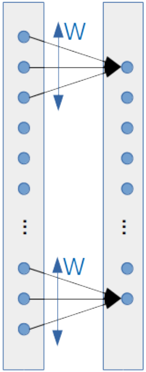
- The *source* nodes form the input vector X, that represents the data, or is the data itself (end-to-end learning).
- The *sink* nodes form the output vector Y, which is interpreted as the result of the classification or regression.

Neural Networks



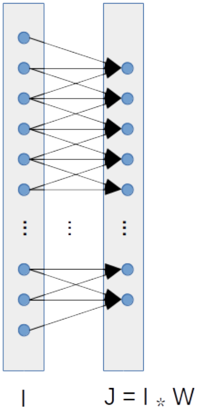
- The architecture (i.e. the graph), and the activation functions are generally defined *a priori* and *static*.
- The weights of the connexions W (and the bias values b) are *adaptive* and modelled by the learning process.

Convolutional Neural Networks



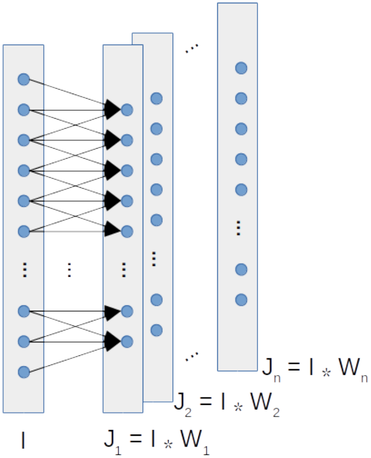
In a Convolutional Neural Networks (CNN), a same neuron (i.e. same weight vector and activation function) is used for all the parts of the input vector associated to each layer.

Convolutional Neural Networks



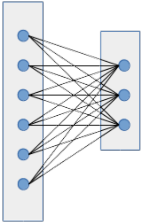
The operation performed between two layers I and J is then a translation invariant linear mapping, i.e. a convolution...

Convolutional Neural Networks



In fact, there are generally several neurons that are applied this way to each layer, which corresponds to a convolution filter bank...

Fully Connected Layers

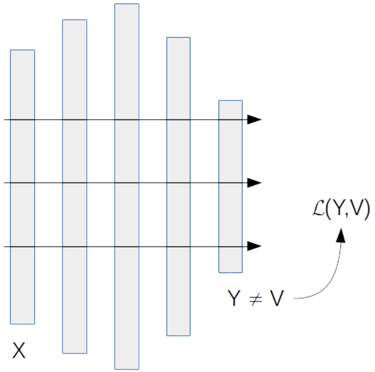


An important special case: when the size of the weight vectors is the same as the size of the input vector, this corresponds to a Fully Connected (FC) Layer.

Supervised Training of a NN

- A NN is trained from a learning set $\{X_i, V_i\}$ where X_i are the training data and V_i the expected outputs (*Ground Truth*).
- A loss function $\mathcal{L}(Y, V) \geq 0$ is designed, that measures the difference (error) between the predicted output Y and the expected output V.
- The objective of the training is to minimise the global error over the training set: $\displaystyle\sum_i \mathcal{L}(\mathcal{O}(X_i), V_i)$, where $\mathcal{O}(X)$ is the output predicted by the network on the input X.

Supervised Training of a NN (forward)



In the forward pass, the data X is submitted to the network, and its predicted output Y is compared to the expected output V using the loss function $\mathcal{L}(Y, V)$.

Forward Propagation

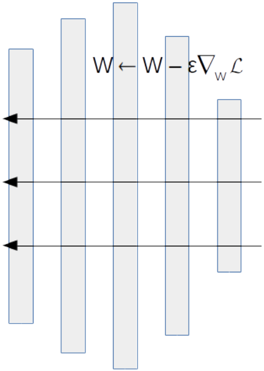
To simplify the notation, all activation functions are assumed equal to g. w_{kj} denotes the weight of the connection from neuron k to neuron j. The forward pass is then written:

Forward Propagation $Y \leftarrow \text{ForwardProp}(W, X)$

- For all neuron i from the input layer: $s_i = x_i$.
- For all following layer l:
    - For all neuron j from layer l: $s_j = g\left(\sum_k w_{kj} s_k + b_j\right)$.
- For all neuron j from the output layer: $y_j = s_j$.

Remark: In the following we denote $a_j = \left(\sum_k w_{kj} s_k + b_j\right)$ the *activation value* of neuron j, i.e. such that $s_j = g(a_j)$.

Training a NN (backward)



In the backward pass, the computed error $\mathcal{L}(Y, V)$ is back-propagated to all the neurons, and the connexion weights are adjusted, depending on their contribution to the error:

$$w_{ij} \leftarrow w_{ij} - \varepsilon \frac{\partial \mathcal{L}}{\partial w_{ij}}$$

where ε is the learning rate.

Gradient Backpropagation Algorithm

The backward pass algorithm is written (for a quadratic loss function $\mathcal{L}$):

Gradient Backpropagation $W \leftarrow \text{BackProp}(W, Y, V)$

- For all neuron j from the output layer:
    - compute the error $\Delta_j = (s_j - v_j) \times g'(a_j)$
- For all previous layer l:
    - For all neuron i from layer l :
        - compute the error $\Delta_i = \left(\sum_k \Delta_k w_{ik} \right) \times g'(a_i)$
- For all connection (i, j) of the network:
    - update the weight $w_{ij} \leftarrow w_{ij} - \varepsilon \times s_i \times \Delta_j$
    - update the bias $b_j \leftarrow b_j - \varepsilon \times \Delta_j$

Presentation Outline

1. Reminder on Neural Networks

2. Learning-based Optical Flow prediction
   - Supervised methods
   - Self-supervised methods

3. Learning-based Depth prediction
   - Reminder on Depth Cues
   - Fully supervised methods
   - Self-supervised methods

4. Conclusion

Learning-based Optical Flow prediction

Since 2015, end-to-end Deep Learning based methods are progressively outperforming the optical flow algorithms, providing results in fast and constant progression, both in terms of accuracy and computational efficiency:

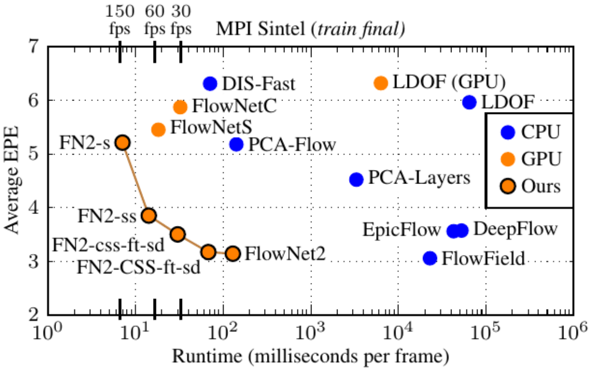
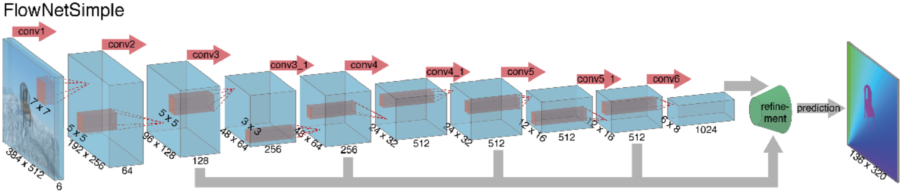
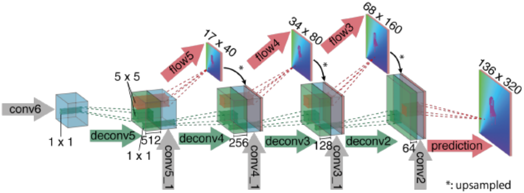


Figure from [Ilg 17]

Learning-based Optical Flow prediction



FlowNetSimple CNN **[Fischer 15]**

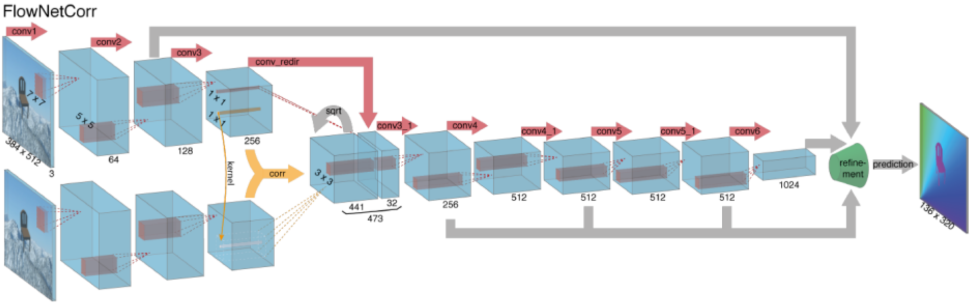


Decoder sub-network (refinement) **[Fischer 15]**

- The DNN performs a dense estimation of the OF by exploiting all possible cues of motion.
- With dense ground truth annotations, the loss function may be simply the $\mathcal{L}_2$ norm:

$$\mathcal{L} = ||\mathbf{v}_{GT} - \hat{\mathbf{v}}||_2$$

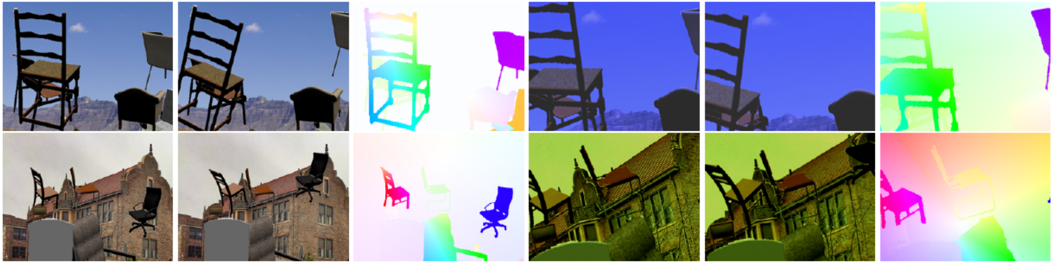
Learning-based Optical Flow prediction



FlowNetCorr CNN **[Fischer 15]**

- Unlike FlowNetSimple, FlowNetCorr learns the spatial features of fixed images, that are explicitly correlated in the 4d block "corr" (not learned!), which is then used as the input of an encoder sub-network.
- The distinction FlowNetSimple / FlowNetCorr related to the pre-processing or spatial selection of analytical methods.

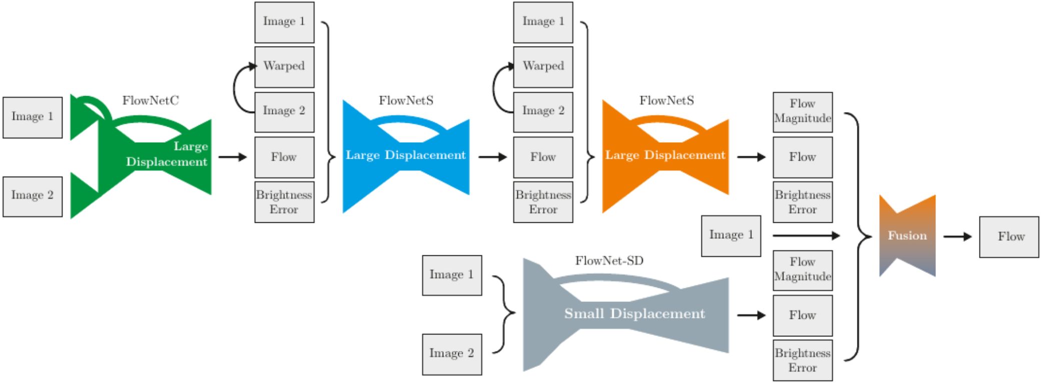
Which training dataset?



The FlyingChairs synthetic dataset **[Fischer 2015]** provides dense annotations on scenes integrating different level of typical optical flow difficulties: homogeneous areas, thin objects, holes, occlusions, large speed range, etc. Furthermore the data can be easily augmented (on the right).

Learning based methods: FlowNet 2.0

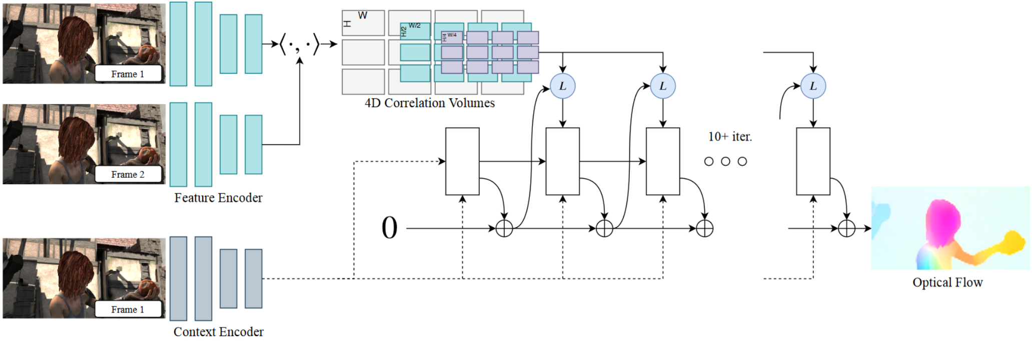
The complementary performances of FlowNetSimple and FlowNetCorr, and their limitations, e.g. their difficulty to address a wide range of displacements, have led to a modular trend in many DNN architectures.



FlowNet 2.0 Network [Ilg 17]

Learning based methods: RAFT

The RAFT (Recurrent All-Pairs Field Transforms) network imitates more directly the analytical methods, not only by reusing the principle of integral correlation of features through the 4d correlation volume, but also by mimicking the iterative mechanism of the optimisation methods, through the use of recursive blocks (GRUs):



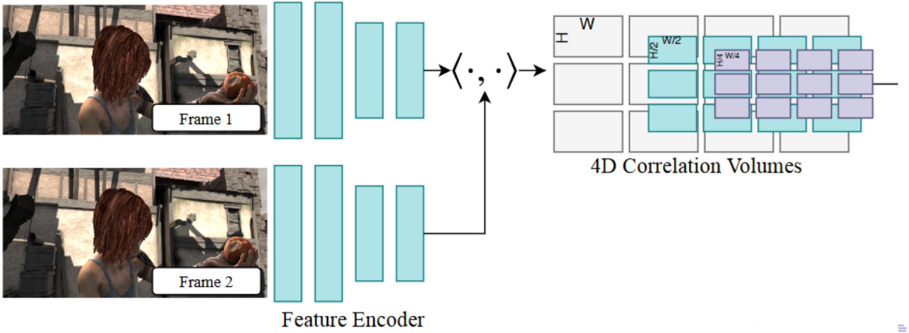
RAFT Network [Teed & Deng 20]

Learning based methods: RAFT

The first step of RAFT consists in computing – like FlowNetCorr – a collection of feature maps for each image of the pair, then to correlate the two resulting maps within a 4d volume:

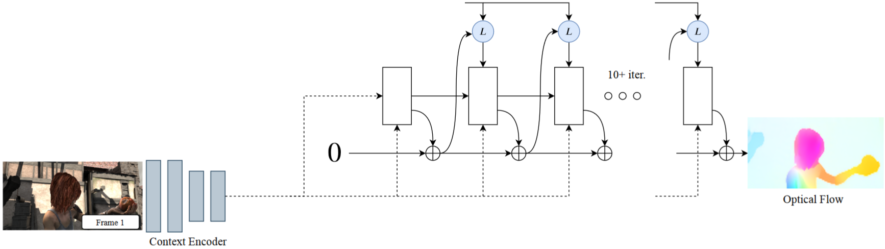
$$C(x,y,x',y') = F_1(x,y).F_2(x',y') = \sum_i F_1^i(x,y)F_2^i(x',y')$$

RAFT then forms a pyramid of 4d correlation $\{C_0, C_1, C_2, C_3\}$ where only the two last dimensions are quantized (The dimension of volume C_k is $W \times H \times W/2^k \times H/2^k$).



Learning based methods: RAFT

The second step of RAFT consists in the iterated application of a GRU (Gated Recurrent Unit) convolutional cell, that iteratively estimates the residual flow Δf and sums it to the current estimate: $f_k^t = f_{k-1}^t + \Delta f$, as a function of the "memory" state m_k, the "input" state i_k, and the "hidden" state h_k, from which Δf is computed. W_m, W_i and W_h are the learned weights of convolutions.

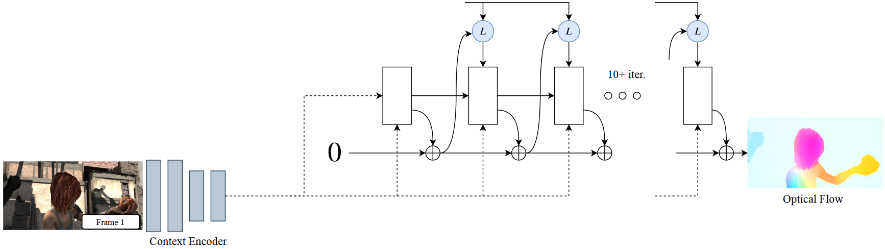


RAFT Network [Teed & Deng 20]

Learning based methods: RAFT

- the input vector x_k^t is formed, for each pixel $\mathbf{p}$, by the concatenation of its estimated flow f_k^t, and its associated Correlation C^t and Context K^t features, i.e. around $\mathbf{p} + f_k^t$ (Lookup L).

$$m_k = \sigma\left(W_m \star [h_{k-1}, x_k^t]\right)$$
$$i_k = \sigma\left(W_i \star [h_{k-1}, x_k^t]\right)$$
$$\hat{h}_k = \tanh\left(W_h \star [i_k \odot h_{k-1}, x_k^t]\right)$$
$$h_k = (1 - m_k) \odot h_{k-1} + m_k \odot \hat{h}_k$$



RAFT Network **[Teed & Deng 20]**

Learning based methods: RAFT

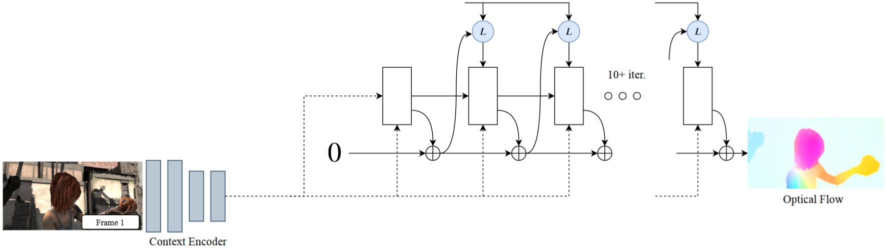
$$m_k = \sigma\left(W_m \star [h_{k-1}, x_k^t]\right)$$

$$i_k = \sigma\left(W_i \star [h_{k-1}, x_k^t]\right)$$

$$\hat{h}_k = \tanh\left(W_h \star [i_k \odot h_{k-1}, x_k^t]\right)$$

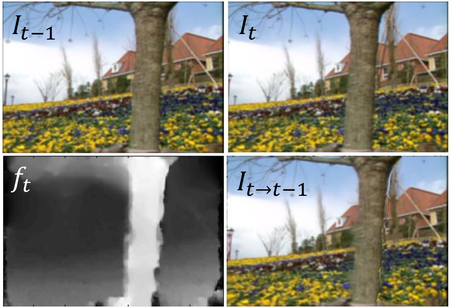
$$h_k = (1 - m_k) \odot h_{k-1} + m_k \odot \hat{h}_k$$

- σ and tanh are the activation functions, $[,]$ stands for the concatenation, and $\odot$ for the Hadamard product.



RAFT Network **[Teed & Deng 20]**

Self-supervised Learning the Optical Flow



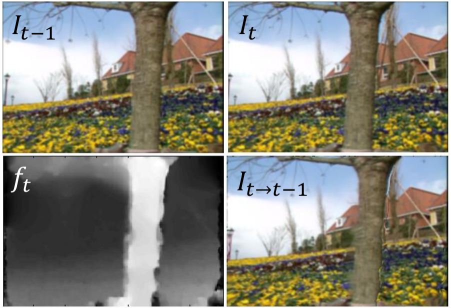
Self-supervised learning (or fine tuning) can be made on real images by using a photometric loss function, that quantifies the difference between an image and its prediction based on the optical flow:

$$\mathcal{L}_{ph} = ||I_{t-1} - I_{t\to t-1}||,$$

with:

$$I_{t\to t-1}(\mathbf{x}) = I_t\left(\mathbf{x} + f_t(\mathbf{x})\right).$$

Self-supervised Learning the Optical Flow



However, additional difficulties occur:

- Homogeneous areas
- Occlusion areas

This implies - among other - a finer modelling of the loss function, for example:

$$\mathcal{L}_{zh} = ||\left(I_{t-1} - I_{t\to t-1}\right)||\nabla I_{t-1}||||$$

$$\mathcal{L}_{occ} = \min\left(||I_{t-1} - I_{t\to t-1}||, ||I_t - I_{t-1\to t}||\right)$$

Presentation Outline

1. Reminder on Neural Networks

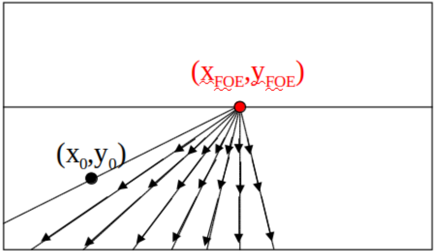
2. Learning-based Optical Flow prediction
   - Supervised methods
   - Self-supervised methods

3. Learning-based Depth prediction
   - Reminder on Depth Cues
   - Fully supervised methods
   - Self-supervised methods

4. Conclusion

3d reconstruction: Limitations of analytical methods

- Estimation strongly relies on local structure (texture), then depth estimation on textureless areas depends on complicated regularization methods.
- Depth calculation depends on the apparent displacement (speed) of a point with respect to the epipole (i.e. the Focus of Expansion FoE, that indicates the translation direction of the camera). Such calculation turns undetermined when the point gets close to the FoE.

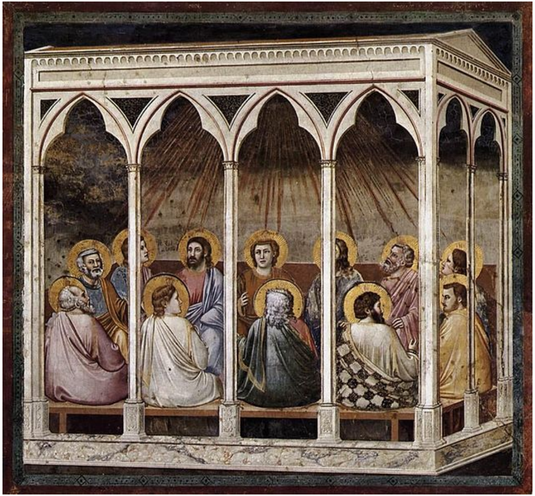


DNN for 3d reconstruction

- Like Optical Flow, Depth can benefit from Deep Networks dense prediction capabilities.
- Training can be easily done on *synthetic* or *real RGB-d* data, and loss function is also relatively straightforward.
- One determining benefit of DNN is their ability to exploit potentially *all the depth indices:* parallax, perspective, size and texture gradients, shading,...

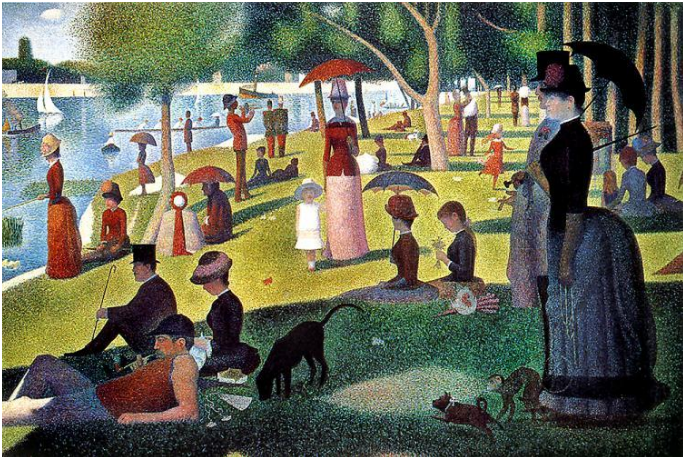
Monocular Depth Cues? Occlusions!

Giotto - Pentecoste
(*circa* 1305)



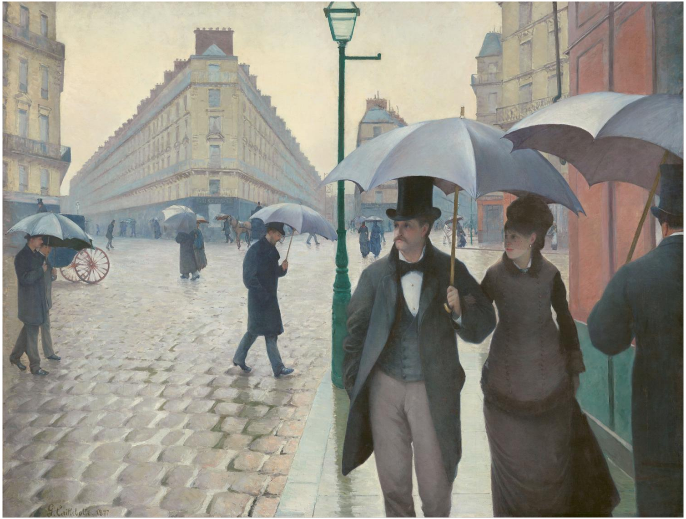
Monocular Depth Cues? Object sizes!

Georges Seurat - Un après-midi à l'île de la Grande Jatte (1884-1886)



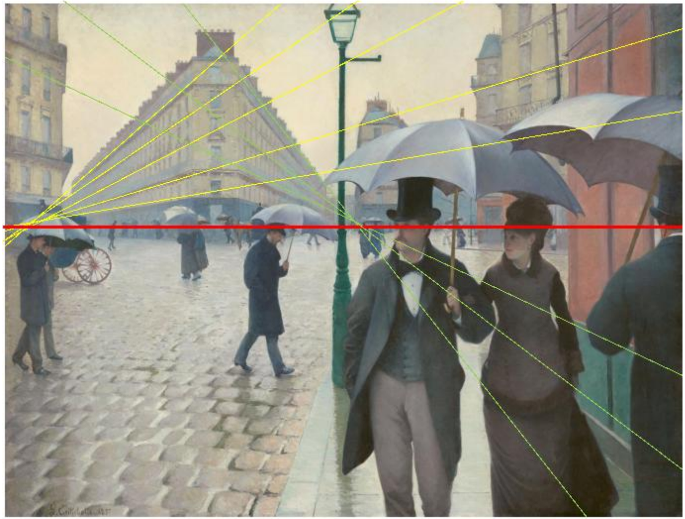
Monocular Depth Cues? Object sizes, Perspective, and Texture Gradients!

Gustave Caillebotte - Rue de Paris, temps de pluie (1877)

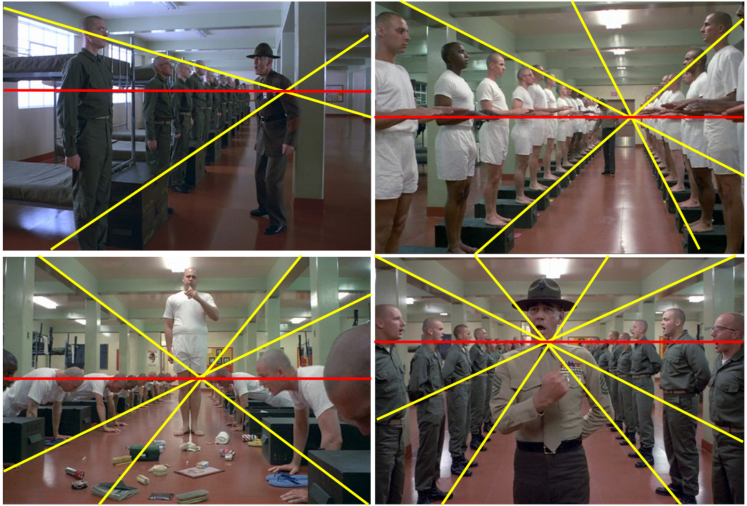


Monocular Depth Cues? Perspective, Horizon and Vanishing Points!

Gustave Caillebotte - Rue de Paris, temps de pluie (1877)

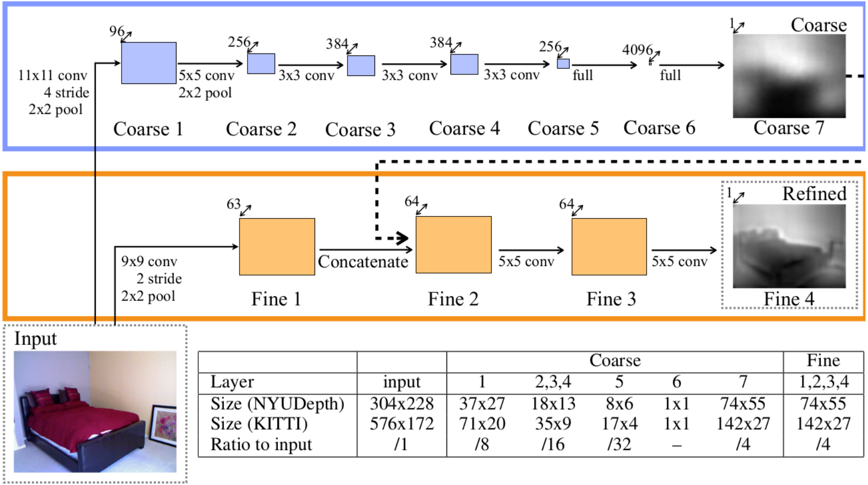


Monocular Depth Cues? Horizon and Camera Pose!



Stanley Kubrick – Full Metal Jacket (1987)

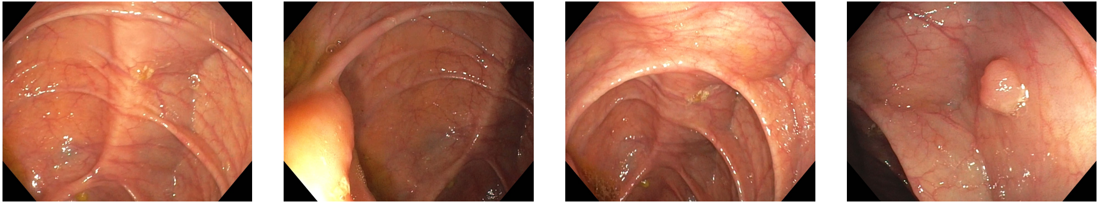
Depth inference from single view!



		Coarse					Fine
Layer	input	1	2,3,4	5	6	7	1,2,3,4
Size (NYUDepth)	304x228	37x27	18x13	8x6	1x1	74x55	74x55
Size (KITTI)	576x172	71x20	35x9	17x4	1x1	142x27	142x27
Ratio to input	/1	/8	/16	/32	–	/4	/4

CNN based Depth estimation from single view **[Eigen 14]** works well on a particular context!

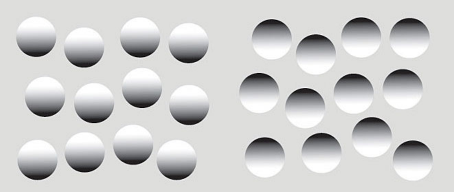
One very particular context...

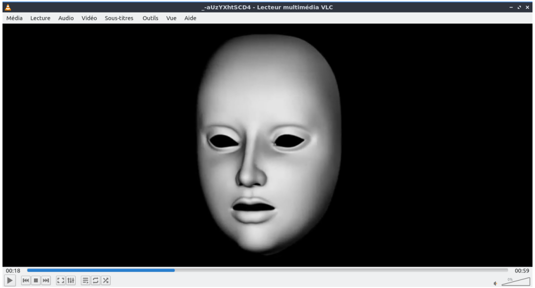


Colonoscopy images **[Ruano 22]**

Monocular Depth Cues? Shading!

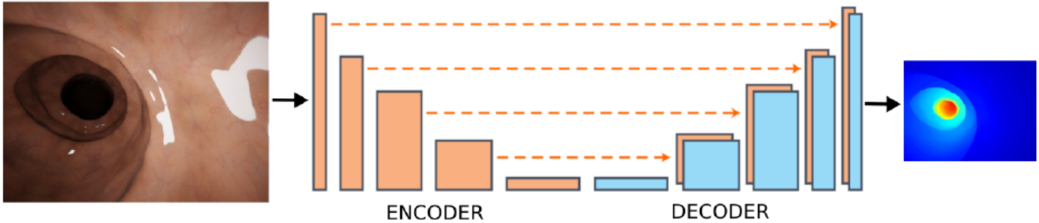
Self shadowing is a strong but ambiguous depth cue (light source position *vs* concavity). Without shape prior, the concavity is determined by a prior of top lighting (right image).





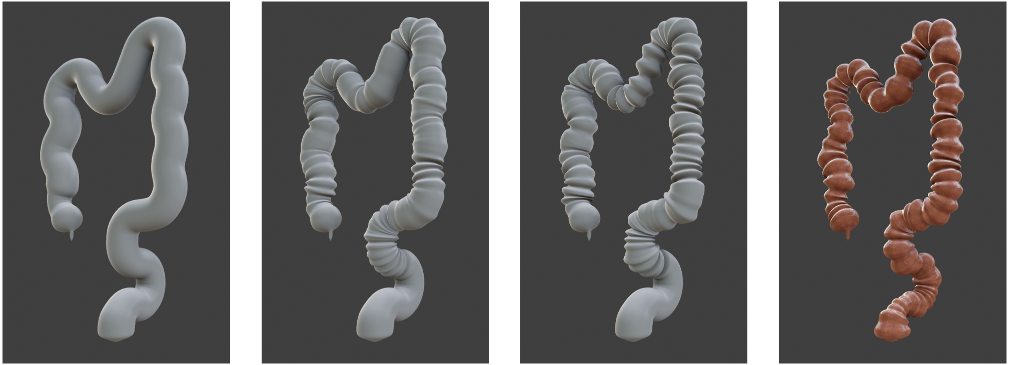
When the shape prior is strong (face then convex), the concavity prior dominates the lighting prior (top-down effect, animation on the left).

Learning Shape from Shading for Automated Colonoscopy



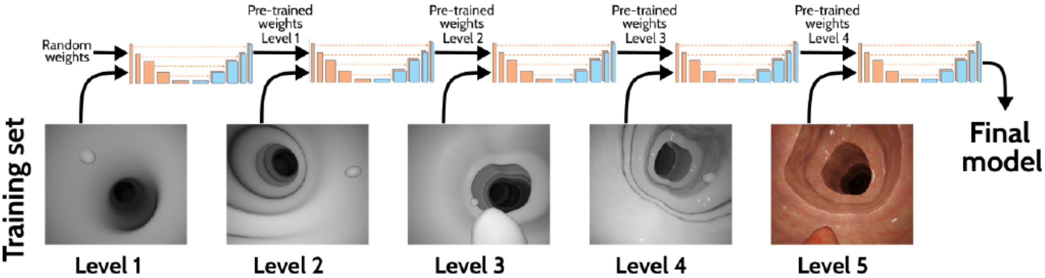
Images from synthetic videos are used to train a CNN using a loss function based on the ground truth depthmap **[Ruano 22]**

Curriculum Learning Shape from Shading for Automated Colonoscopy



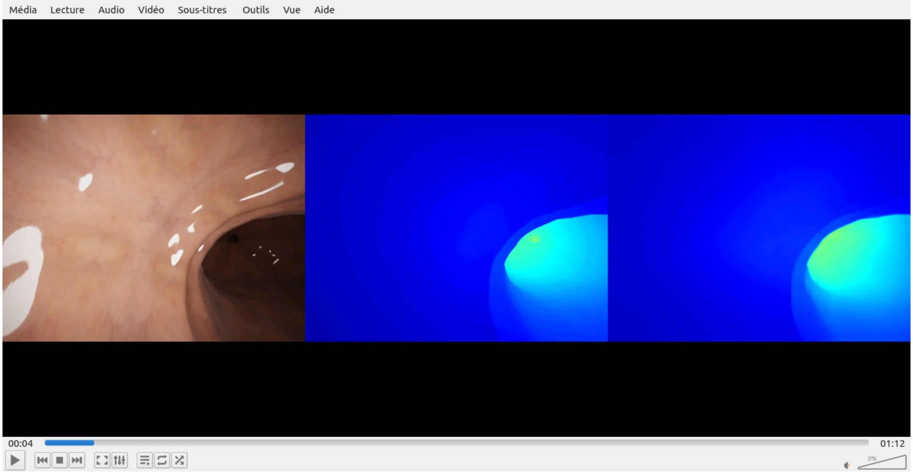
Synthetic exploration videos are created from a hierarchy of synthetic colons of increasing complexity **[Ruano 22]**

Curriculum Learning Shape from Shading for Automated Colonoscopy


Random weights
Pre-trained weights Level 1
Pre-trained weights Level 2
Pre-trained weights Level 3
Pre-trained weights Level 4
Training set
Level 1
Level 2
Level 3
Level 4
Level 5
Final model

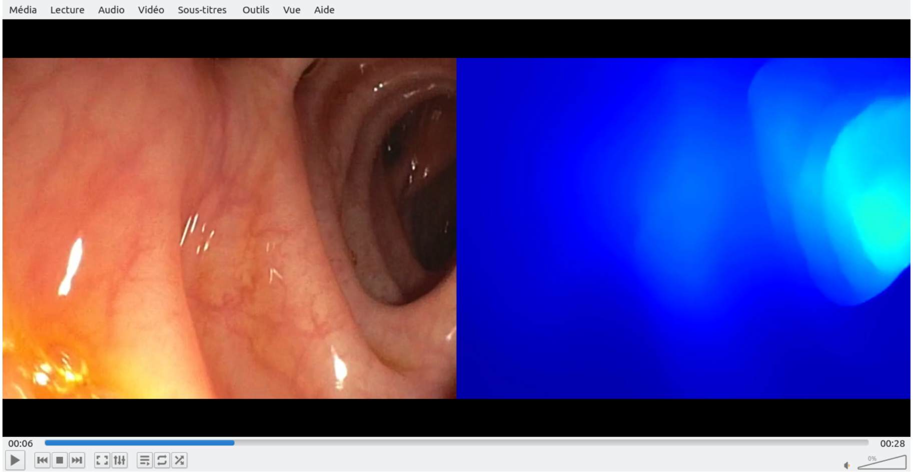
The training is performed with progressive complexity **[Ruano 22]**

SfSNet on Synthetic Videos



ShapeFromShadingNet on Synthetic Test Videos **[Ruano 22]**

SfSNet on Real Videos



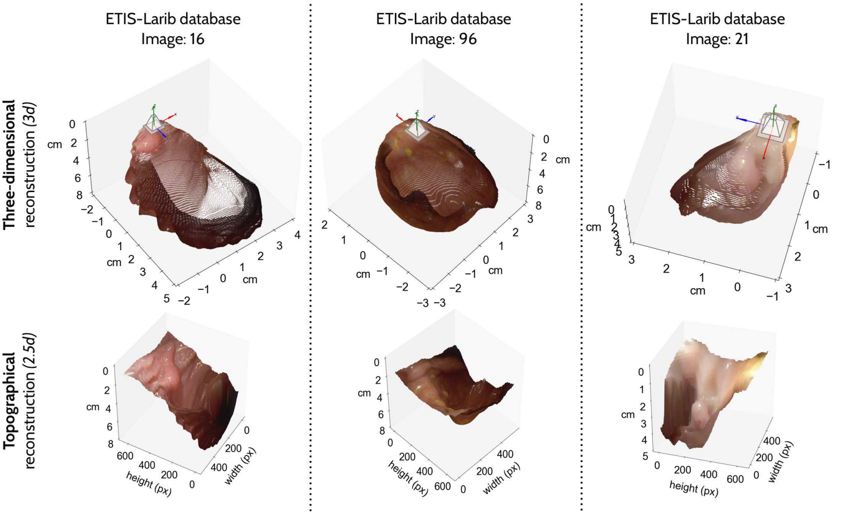
ShapeFromShadingNet on Real Videos [Ruano 22]. Single images seem to be sufficient in such particular context!

3d reconstruction from depth maps

Back-projection from the depth map Z:

$$M = Z(m)\mathbf{K}^{-1}m$$

[Ruano 23]



ETIS-Larib database
Image: 16

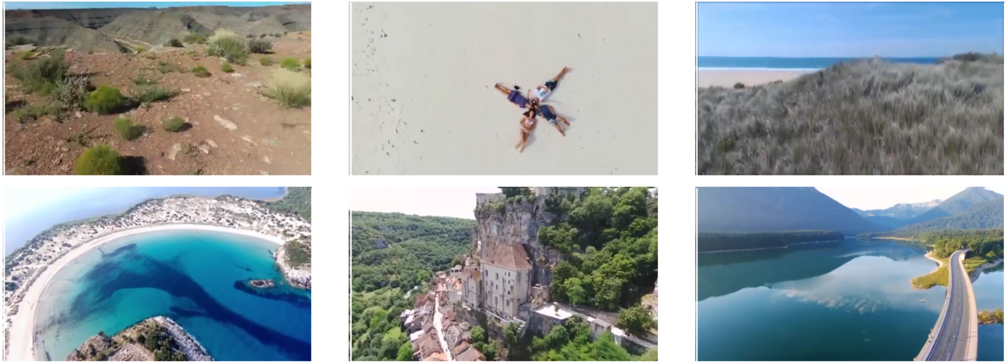
ETIS-Larib database
Image: 96

ETIS-Larib database
Image: 21

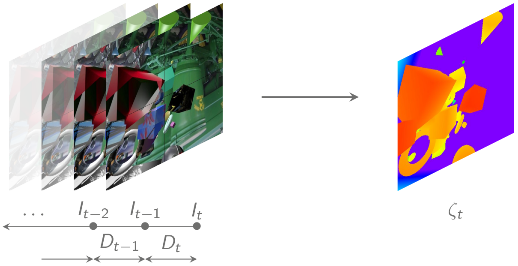
Three-dimensional reconstruction (3d)
Topographical reconstruction (2.5d)
cm
height (px)
width (px)

What about UAV's context?

These scenes are all taken from the same drone !



Non photorealistic synthesis for learning SfM



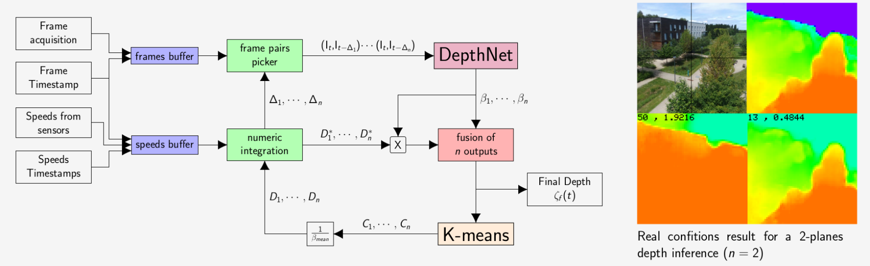
Supervised learning of depth from synthetic sequences

[Pinard 17a]

- Network is based on FlowNet_S
- Unrealistic scenes ↔ Abstraction of the context
- Focus on geometry / motion, not on appearance /context
- Trained on rotationless movement, at a constant speed

Baseline adaptation using multiple image pairs

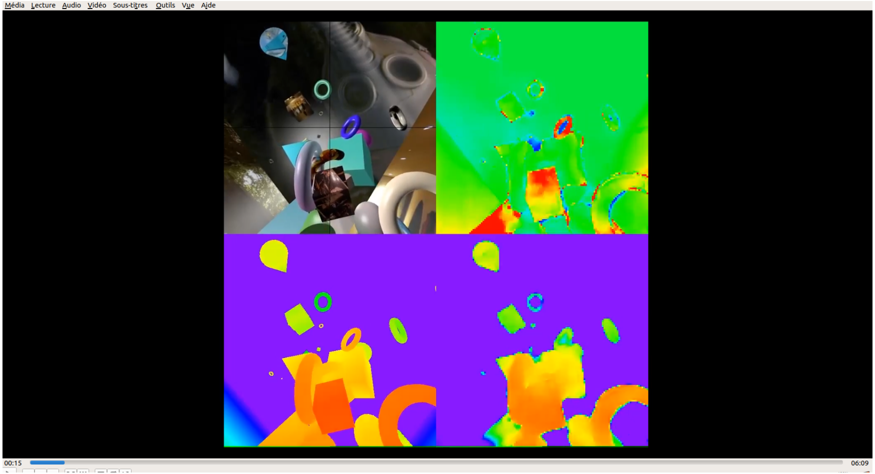
- At the inference time, the depth which is relative to the trained speed, is scaled with respect to the actual velocity.
- Adaptable precision is achieved by dynamically adapting the image pairs (baselines) to the depth distribution.



Real confitions result for a 2-planes depth inference ($n = 2$)

Adaptation of the baselines to the depth distribution [Pinard 17b]

Supervised DepthNet



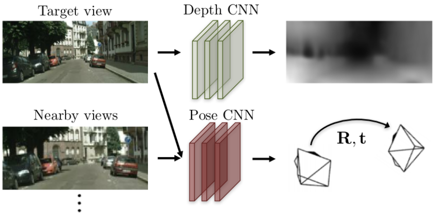
Supervised DepthNet results [Pinard 17a]: See

https://perso.ensta-paris.fr/~manzaner/Download/ECMR2017/DepthNetResults.mp4

Unsupervised depth estimation CNN

- Re-training on real/operative context is still essential.
- But data are rarely annotated.
- Self-supervised learning is then necessary.
- *Photometric loss function* can be used, that compares a pair of registered images, knowing the depth and the camera pose.
- Camera pose then needs to be known, or predicted!

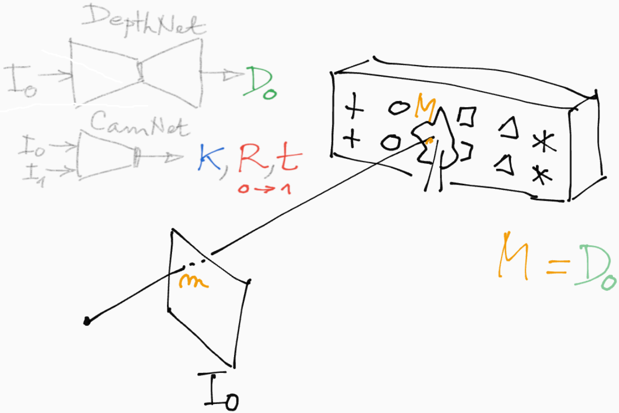
(a) Training: unlabeled video clips.



(b) Testing: single-view depth and multi-view pose estimation.

[Zhou 17]

Photometric Loss (1): Back-projection from first image

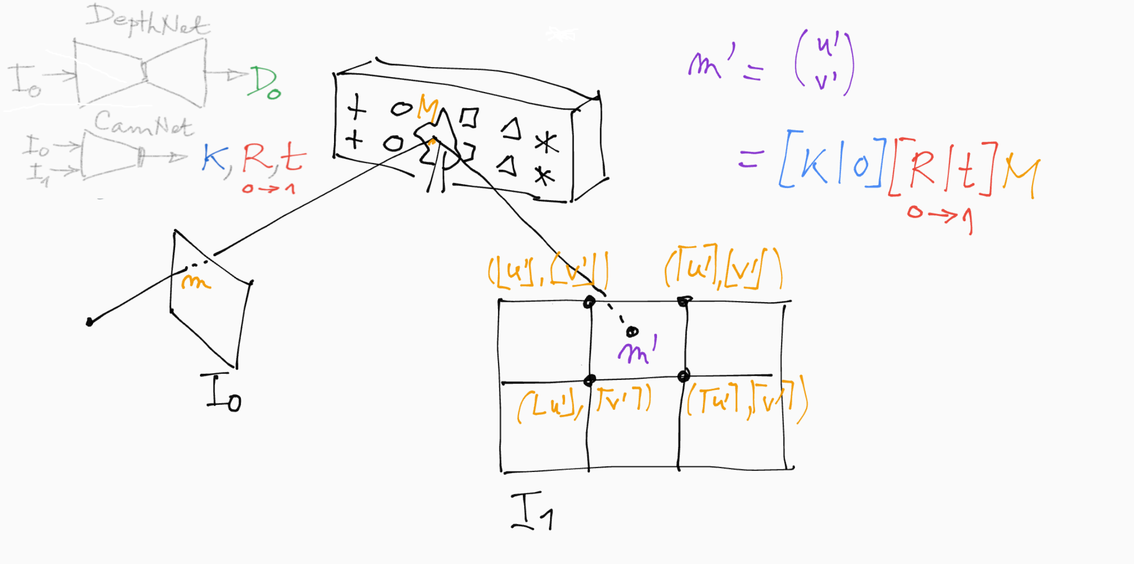


$$m = \begin{pmatrix} u \\ v \end{pmatrix}$$

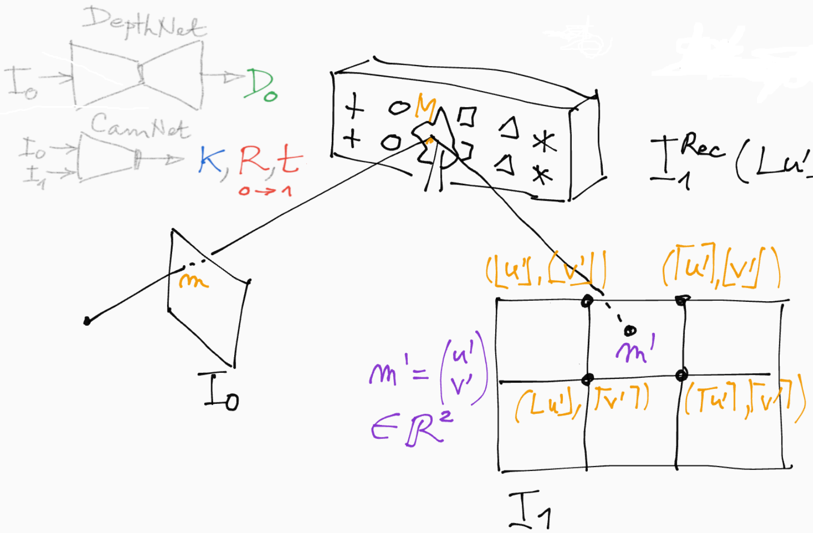
$$M = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

$$M = D_0(m) \times K^{-1} m$$

Photometric Loss (2): Re-projection onto second image


DepthNet
I_0
D_0
CamNet
I_0
I_1
K, R, t
$0 \to 1$
M
m
I_0
$m' = \begin{pmatrix} u' \\ v' \end{pmatrix}$
$= [K|0][R|t]M$
$0 \to 1$
$(\lfloor u' \rfloor, \lfloor v' \rfloor)$
$(\lceil u' \rceil, \lfloor v' \rfloor)$
m'
$(\lfloor u' \rfloor, \lceil v' \rceil)$
$(\lceil u' \rceil, \lceil v' \rceil)$
I_1

Photometric Loss (3): Interpolation within second image


DepthNet
I_0
D_0
CamNet
I_0
I_1
K, R, t
$0 \to 1$
M
m
I_0
$m' = \begin{pmatrix} u' \\ v' \end{pmatrix} \in \mathbb{R}^2$
$(\lfloor u' \rfloor, \lfloor v' \rfloor)$
$(\lceil u' \rceil, \lfloor v' \rfloor)$
m'
$(\lfloor u' \rfloor, \lceil v' \rceil)$
$(\lceil u' \rceil, \lceil v' \rceil)$
I_1

$$I_1^{Rec}(\lfloor u' \rfloor, \lfloor v' \rfloor) += I_0(m) \times (1 - (u' - \lfloor u' \rfloor)) \times (1 - (v' - \lfloor v' \rfloor))$$

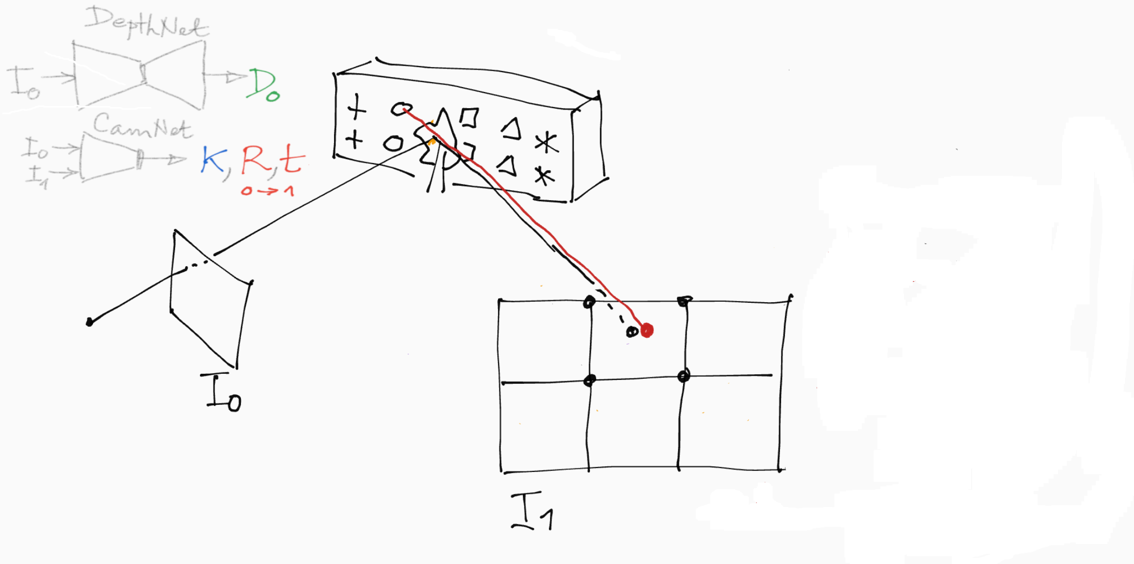
... etc.

Photometric Loss: Summary and formula

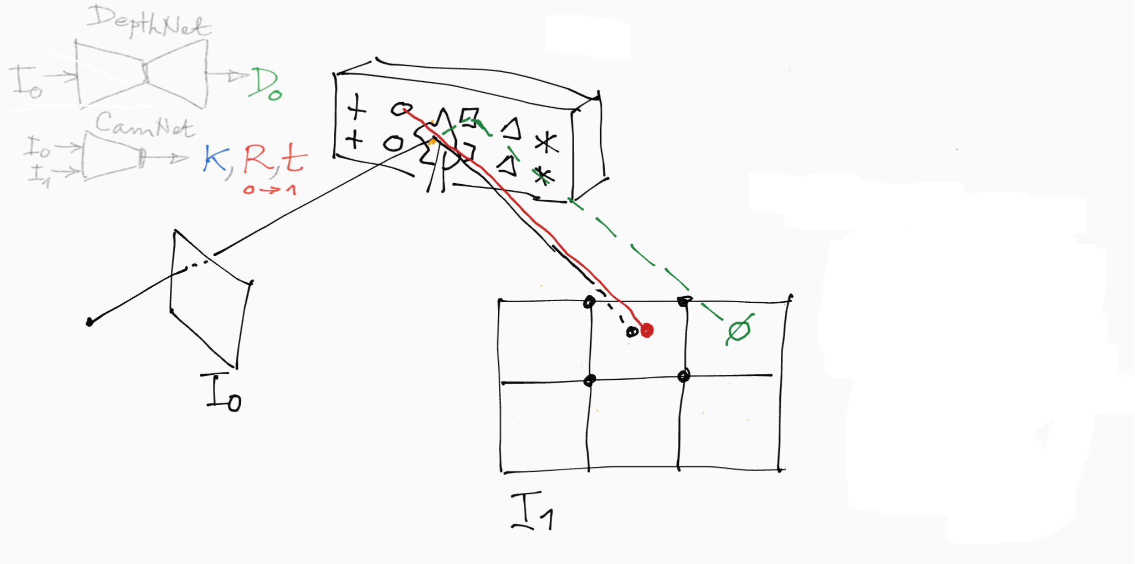
The photometric loss provides a self-supervision signal by comparing the observed image with the reconstructed image from the previous view, based on predicted depth map and odometry:

$$\begin{aligned}\mathcal{L}_{\mathrm{photo}}^{\mathrm{depth,odometry}} &= \| I_1 - I_1^{\mathrm{Rec}} \| \\ &= \sum_{\mathbf{m}'} \left(I_1(\mathbf{m}') - I_0(\mathbf{m}) \right)^2, \text{ with } \mathbf{m}' \simeq \left([\mathbf{K}|\mathbf{O}_4]\,[\mathbf{R}|\mathbf{t}]\, D_0(\mathbf{m}) \times \mathbf{K}^{-1}\mathbf{m} \right)\end{aligned}$$

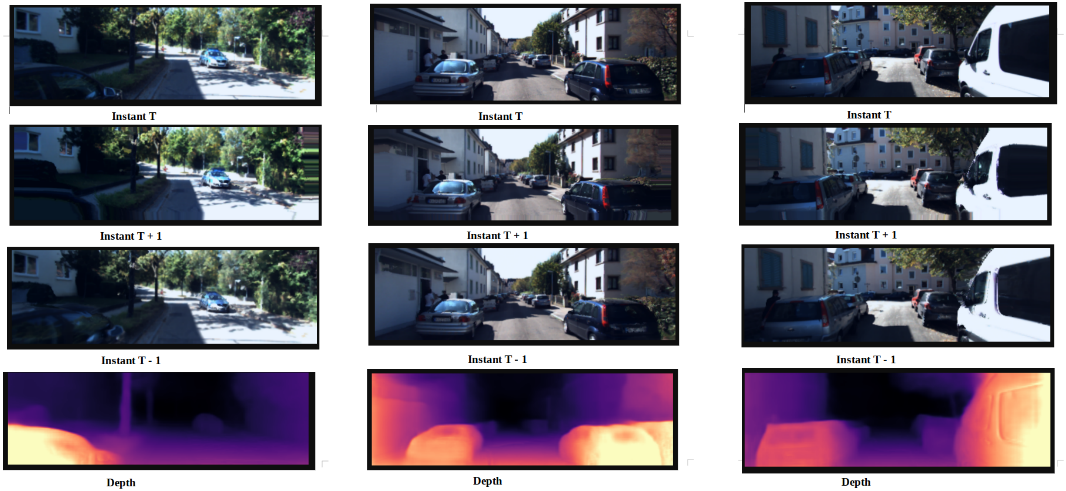
Photometric Loss: Occlusion issue



Photometric Loss: Un-occlusion issue



Examples of reprojected images



Instant T

Instant T + 1

Instant T - 1

Depth

Instant T

Instant T + 1

Instant T - 1

Depth

Instant T

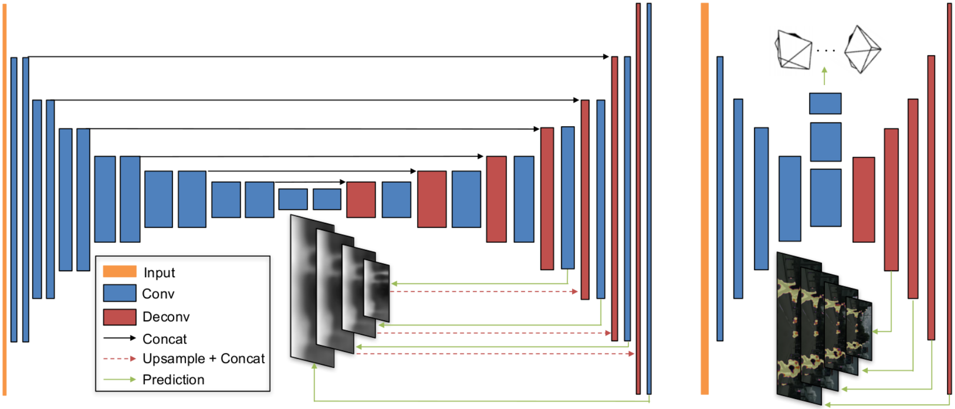
Instant T + 1

Instant T - 1

Depth

[PhD Marwane Hariat]

Unsupervised depth estimation CNN

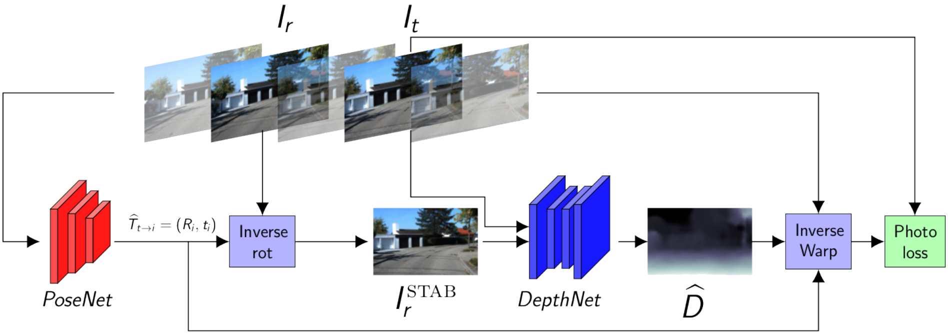


(a) Single-view depth network

(b) Pose/explainability network

[Zhou 17]

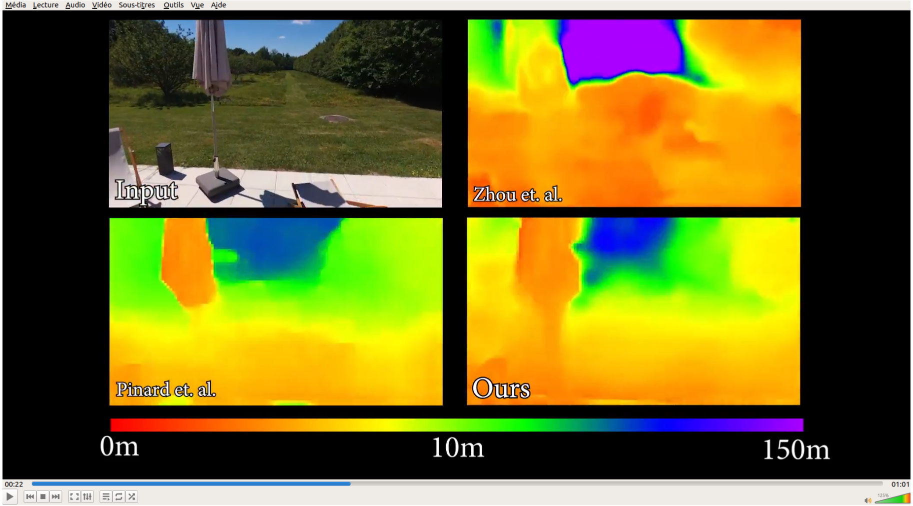
Unsupervised DepthNet



$$\forall i,\ t_i^{\mathrm{NORM}} = t_i \frac{T_0}{\epsilon + \|t_r\|}$$

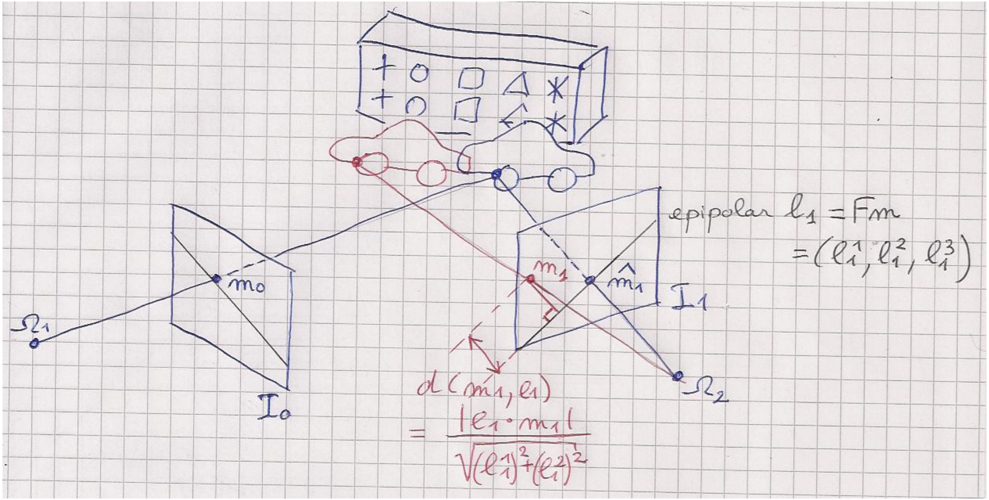
Unsupervised re-learning of Structure from Motion with adaptive baseline **[Pinard 18]**

Unsupervised DepthNet

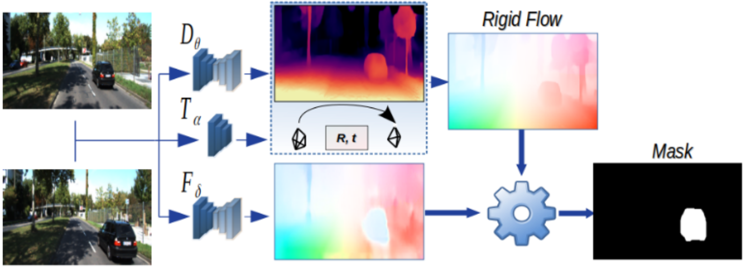


Unsupervised DepthNet real fly demo **[Pinard 18]**: See `https://www.youtube.com/watch?v=ZDgWAWTwU7U`

Photometric Loss: Moving objects issue



CoopNet: Joint training of Optical Flow, Odometry and Depth

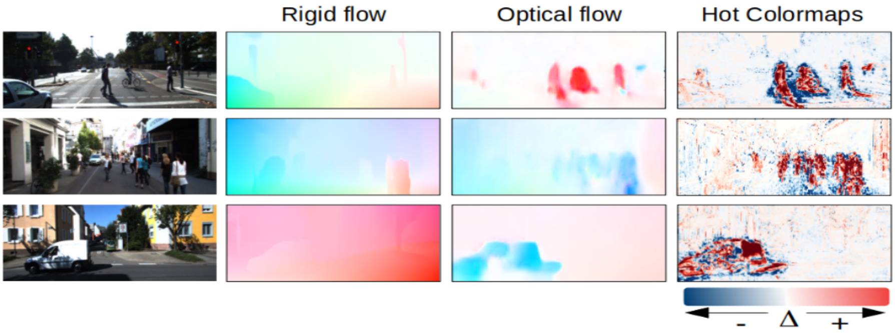


CoopNet **[Hariat 23]**

By estimating (or predicting) the optical flow, moving objects can also be predicted by comparing the optical flow with the *rigid flow*, which is the apparent velocity field under rigid assumption scene (i.e. only due to camera motion), defined as:

$$[\mathbf{K}|\mathbf{O}_4]\,[\mathbf{R}|\mathbf{t}]\,D_0(\mathbf{m}) \times \mathbf{K}^{-1}\mathbf{m} - \mathbf{m}$$

CoopNet: Joint training of Optical Flow, Odometry and Depth



CoopNet [Hariat 23]

The CoopNet network is trained based on the difference between the photometric losses from the optical flow and from the depth networks:

$$\Delta(\mathbf{m}) = \mathcal{L}_{\text{photo}}^{\text{depth,odometry}} - \mathcal{L}_{\text{photo}}^{\text{flow}}$$

Presentation Outline

1. Reminder on Neural Networks

2. Learning-based Optical Flow prediction
   - Supervised methods
   - Self-supervised methods

3. Learning-based Depth prediction
   - Reminder on Depth Cues
   - Fully supervised methods
   - Self-supervised methods

4. Conclusion

Conclusion on Learning-based methods

- Learning optical flow and depth from videos has many advantages:
  - Globally addressing the context
  - Multi-cues depth inference
  - Natural regularization of ill-posed problem
- The main issues to adress are the hard dependence to the learned context, and the difficulties inherent to online learning. The current work perspectives are:
  - Domain adaptation: ground robotics, medical robotics,...
  - Incremental and online learning...
  - Explainability and Reliability...

Contributors for this lecture

- **Matthieu Garrigues**: PhD student 2012-2016
- **Clément Pinard**: PhD student (CIFRE ANRT Parrot) 2016-2019
- **Josué Ruano Balseca**: PhD student (w. UNAL Bogotá) 2018-
- **Marwane Hariat**: PhD student 2021-

References (1)

- **[Fischer 2015]** P. Fischer, A. Dosovitskiy, E. Ilg, P. Häusser, C. Hazırbaş, V. Golkov, P. van der Smagt, D. Cremers, T. Brox
  FlowNet: Learning Optical Flow with Convolutional Networks
  Proceedings of the International Conference on Computer Vision (ICCV), Dec. 2015, pp 2758–2766

- **[Ilg 2017]** E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, T. Brox
  FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks
  Conference on Computer Vision and Pattern Recognition (CVPR), 2017

- **[Teed & Deng 2020]** Z Teed, J Deng
  A simple and efficient rectification method for general motion
  Proceedings of 16th European Conference, ECCV 2020

References (2)

[Garrigues 17] M. Garrigues and A. Manzanera
Fast Semi Dense Epipolar Flow Estimation
IEEE Winter Conf. on Applications of Computer Vision (WACV). Sta Rosa, CA, pp.1-8, 2017

[Eigen 14] D. Eigen and C. Puhrsch and R. Fergus
Depth map prediction from a single image using a multi-scale deep network
Advances in neural information processing systems (NIPS), pp.2366–2374, 2014

[Ruano 22] J. Ruano Balseca and M. Gómez and E. Romero and A. Manzanera
SfSNet: Learning Shape-from-Shading for recovering the geometry of the colon wall from monocular colonoscopy
Research Report, 2022

[Zhou 17] T. Zhou and M. Brown and N. Snavely and D.G. Lowe
Unsupervised learning of depth and ego-motion from video
Computer Vision and Pattern Recognition (CVPR), 2017.

References (3)

[Pinard 17a] C. Pinard and L. Chevalley and A. Manzanera and D. Filliat
End-to-end depth from motion with stabilized monocular videos
Int. Conf. on Unmanned Aerial Vehicles in Geomatics (UAV-g) Bonn, pp. 67-74, 2017

[Pinard 17b] C. Pinard and L. Chevalley and A. Manzanera and D. Filliat
Multi range Real-time depth inference from a monocular stabilized footage using a Fully Convolutional Neural Network
European Conference on Mobile Robotics (ECMR), Palaiseau, 2017

[Pinard 18] C. Pinard and L. Chevalley and A. Manzanera and D. Filliat
Learning structure-from-motion from motion
European Conf. on Computer Vision Workshops (ECCV-W), pp.363-376, 2018

[Hariat 23] M. Hariat and A. Manzanera and D.Filliat
Rebalancing gradient to improve self-supervised co-training of depth, odometry and optical flow predictions
IEEE Winter Conf. on Applications of Computer Vision (WACV). Waikoloa, 2023