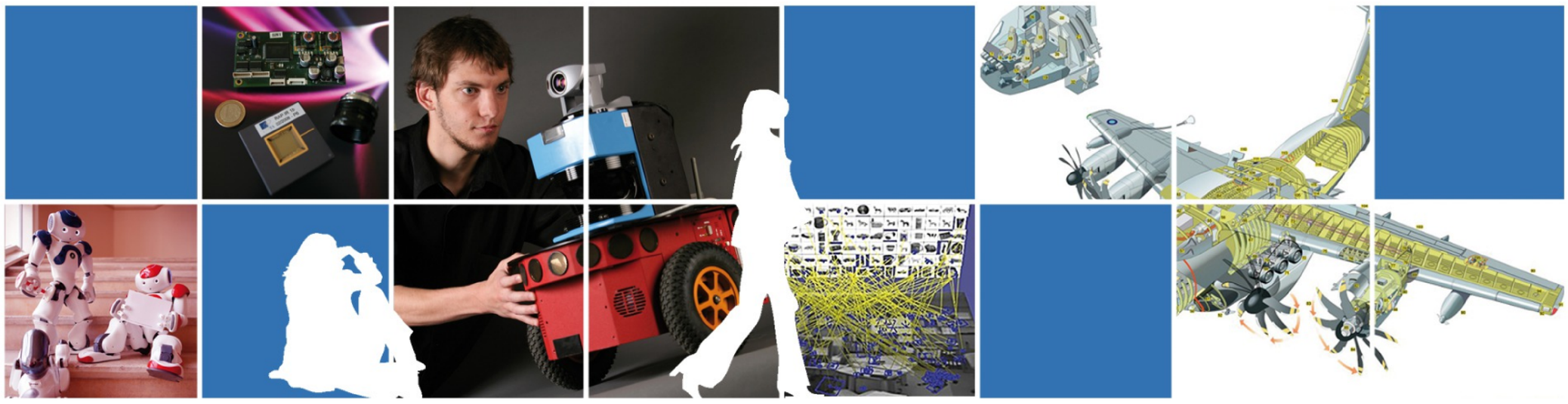


Multi range Real-time depth inference from a monocular stabilized footage using a Fully Convolutional Neural Network

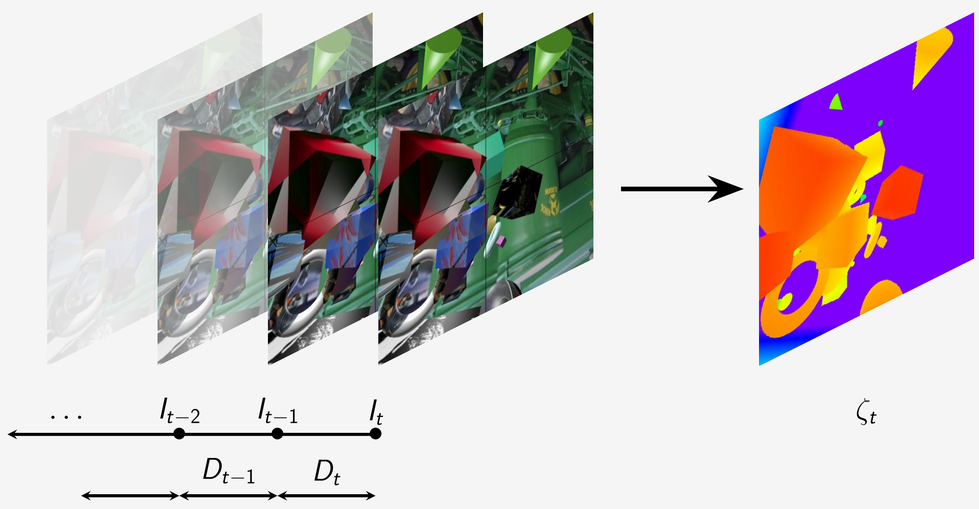
Clément Pinard[a,b] Laure Chevalley[a] Antoine Manzanera[b] David Filliat[b]

[a]Parrot, Paris, France
(clement.pinard, laure.chevalley)@parrot.com

[b]U2IS, ENSTA ParisTech, Université Paris-Saclay, Palaiseau, France
(clement.pinard, antoine.manzanera, david.filliat)@ensta-paristech.fr

Introduction

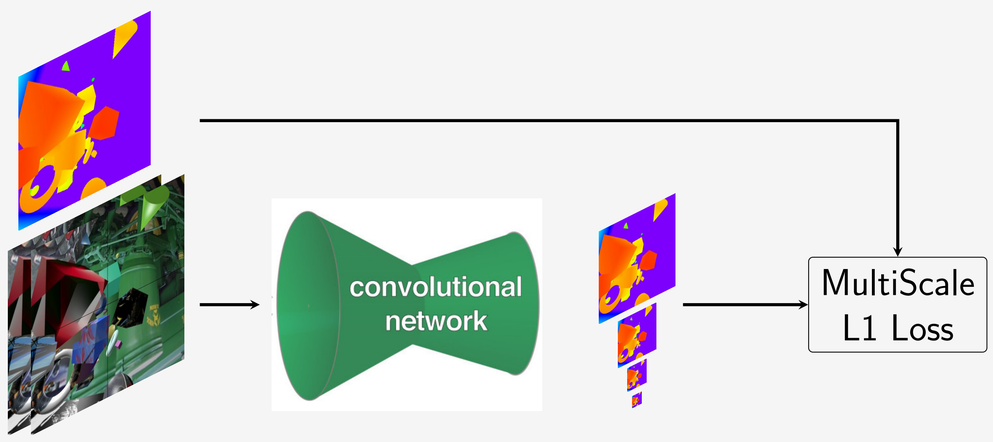
This research focuses on computing for every frame a dense depth-map ζ from a monocular **stabilized** footage using previous frames I_t and displacement D_t in a rigid scene.



Network and Training Specification

An intermediate problem is solved with a Fully Convolutional Neural Network, Depth Map from a stabilized image pair.

- Displacement is assumed to be the same
- Displacement direction is **NOT** given to the network.
- **Supervised** training is done using a synthetic drone-like dataset



from DepthNet to real depth

Knowing UAV speed, DepthNet output β is compensated to match real depth

$$\zeta(t) = \beta(I_t, I_{t-\Delta t})\frac{D(t,\Delta t)}{D_0} \quad with \quad D(t,\Delta t) = \left\| \int_{t-\Delta t}^{t} V(\tau) d\tau \right\| \tag{1}$$

From the centroids $C_1, \cdots, C_n$ of ζ, we set optimized distances $D_1^*, \cdots, D_n^*$ for DepthNet to compute depth precisely around C_k, $D_k^* = {}^{C_k}/_{\beta_{mean}}$. Δ_k is then found from flight history to match D_k^*

weighted fusion

Each plane output pixel is weighted from its proximity to target mean depth β_{mean}. Final Depth is then the pixel-wise weighted sum of the planes

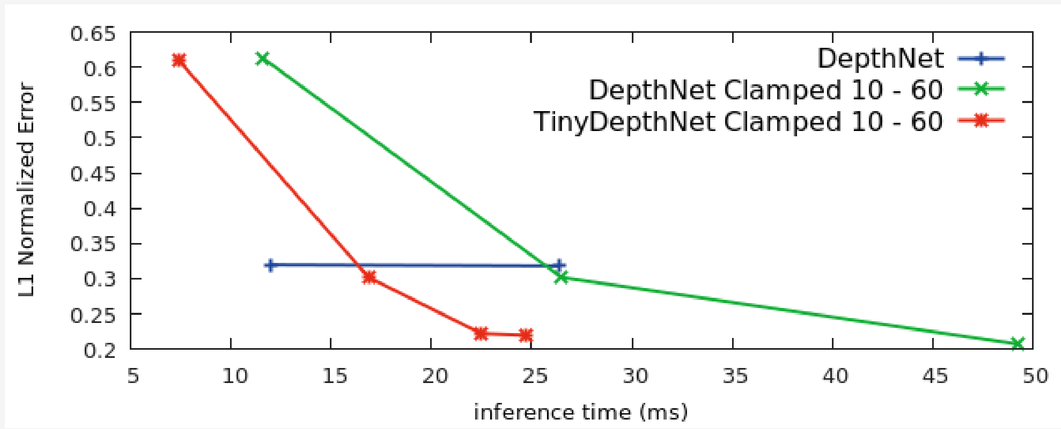
$$w_{ijk} = \epsilon + f(\beta(I_t, I_{t-\Delta_i}))$$

$$f : x \mapsto \begin{cases} 0 & \text{if } x < \beta_{min} \\ \frac{x-\beta_{min}}{\beta_{mean}-\beta_{min}} & \text{if } \beta_{min} \leq x < \beta_{mean} \\ \frac{\beta_{max}-x}{\beta_{max}-\beta_{mean}} & \text{if } \beta_{mean} \leq x < \beta_{max} \\ 0 & \text{if } x \geq \beta_{max} \end{cases} \tag{2}$$

$$\zeta_i(t) = D_i(t)\beta(I_t, I_{t-\Delta_i})$$

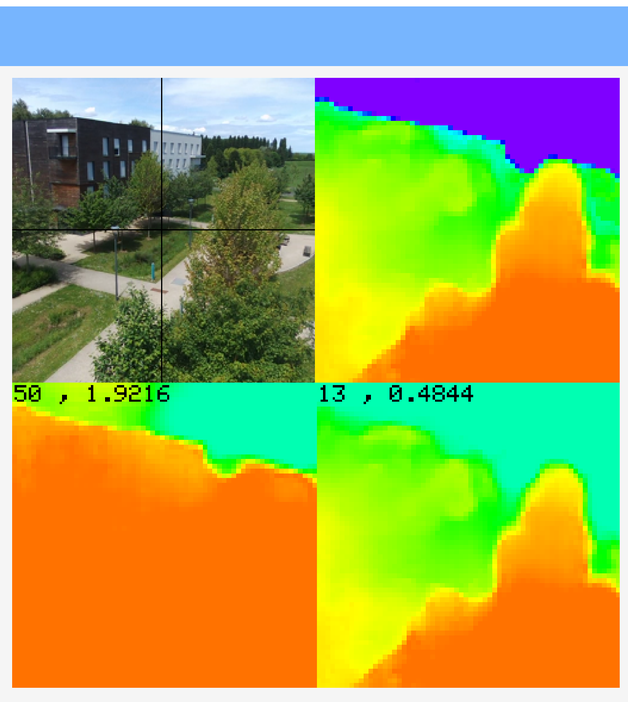
$$\forall (j,k) \in [\![0, W]\!] \times [\![0, H]\!], \zeta_f(t)_{jk} = \frac{\sum_i w_{ijk}\zeta_{ijk}(t)}{\sum_i w_{ijk}} \tag{3}$$

Validation on synthetic dataset



Results for synthetic 256x256 scenes with noisy orientation (1^omax)

Final Workflow and result



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