End-to-end depth from motion with stabilized monocular videos

Clément Pinard, Laure Chevalley
Antoine Manzanera, David Filliat

Parrot, ENSTA ParisTech

09/07/2017

Outline

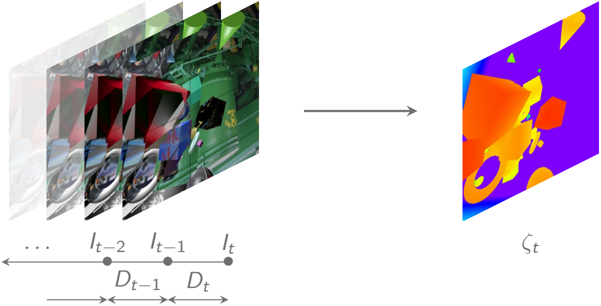
1. Motivations and Technological Context
   - Stabilized Footage
   - Datasets for supervised depth training
   - Introducing Still Box
2. Supervised Depth Training
3. Results

We assume a perfectly stabilized footage can be obtained from a drone, be it digital or mechanical.

On rigid scenes, this simplifies dramatically relation between depth, displacement and depth-map which can be then used for obstacle avoidance

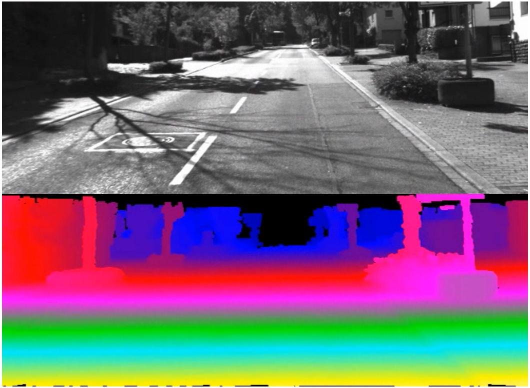
Initial problem

Our goal is to compute for every frame a dense depth-map ζ from a monocular footage using previous frames I_t and displacement D_t in a rigid scene



Datasets for supervised depth training

Some datasets with avalaible depth and displacement e.g. KITTI (Andreas Geiger et al. 2012)



Frames are not stabilized but orientations are provided for offline stabilization.

But...

- *a posteriori* warping is not ideal for information conservation
- Scenes are not always rigid
- Driving scenes are not as heterogeneous as drone scenes
- Movement is only forward/backard

In fact, driving scene structure are so predictible that depth from a single image is possible with a neural network! (Zhou et al. 2017)

These scenes are all taken from the same drone !

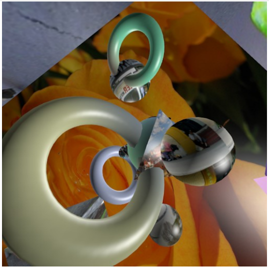


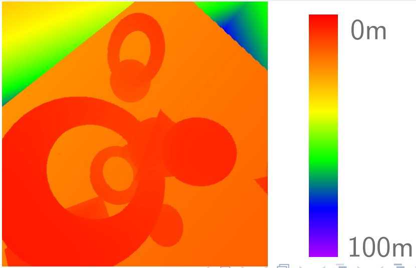



Introducing Still Box

Still Box aims at mimicking a typical drone video

- no rotation
- rigid scenes
- random orientation and speed direction
- random textures and shapes
- It is designed so that depth from a single image is impossible





Outline

1. Motivations and Technological Context

2. Supervised Depth Training
   - Flow Map vs Depth Map
   - Training on Still Box Dataset

3. Results

Definition

Disparity $\delta(\mathbf{P})$ is defined here by the norm of flow vector, $\mathbf{flow(P)} = \begin{pmatrix} du \\ dv \end{pmatrix}$ of a point $\mathbf{P} = \begin{pmatrix} u \\ v \end{pmatrix}$.

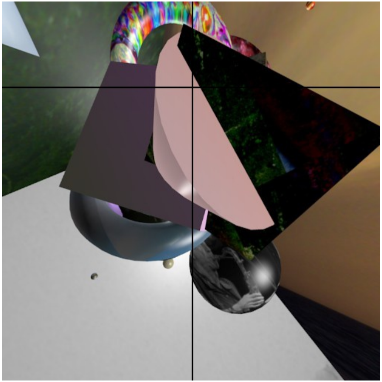
$$\forall \mathbf{P} = \begin{pmatrix} u \\ v \end{pmatrix}, \delta(\mathbf{P}) = \|\mathbf{flow(P)}\|$$

Definition

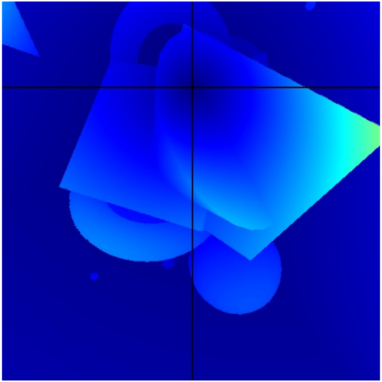
Focus of Expansion is defined by the point Φ where each flow vector $\mathbf{flow(P)} = \begin{pmatrix} du \\ dv \end{pmatrix}$ of a point $\mathbf{P} = \begin{pmatrix} u \\ v \end{pmatrix}$ is headed from.

$$\forall \mathbf{P} = \begin{pmatrix} u \\ v \end{pmatrix}, det\left(\overrightarrow{\mathbf{P}\Phi}, \mathbf{flow(P)}\right) = 0$$

FOE Φ is the center of the cross, (and is perfectly known)



Input Images



Disparity Map δ

Around Φ , disparity δ is approaching 0

Theorem

For a random rotation-less displacement of norm V of a pinhole camera, with a focal length f, depth $\zeta(\mathbf{P})$ is an explicit function of disparity $\delta(\mathbf{P})$, focus of expansion Φ and optical center $\mathbf{P}_0$

$$\forall \mathbf{P}, \zeta(\mathbf{P}) = \frac{Vf}{\sqrt{f^2 + \left\|\overrightarrow{\mathbf{P}_0\Phi}\right\|^2}} \left(\frac{\left\|\overrightarrow{\mathbf{P}\Phi}\right\|}{\delta(\mathbf{P})} - 1 \right)$$

This will be undefined when approching Φ ! Problematic since it's where the drone is going. A simple Optical flow network CNN will not be sufficient for our problem.

training

We train a CNN to output direct DepthMap from an image pair instead of Optical Flow called **DepthNet**. Displacement is supposed to be constant (at D_0), depth is compensated according to this statement

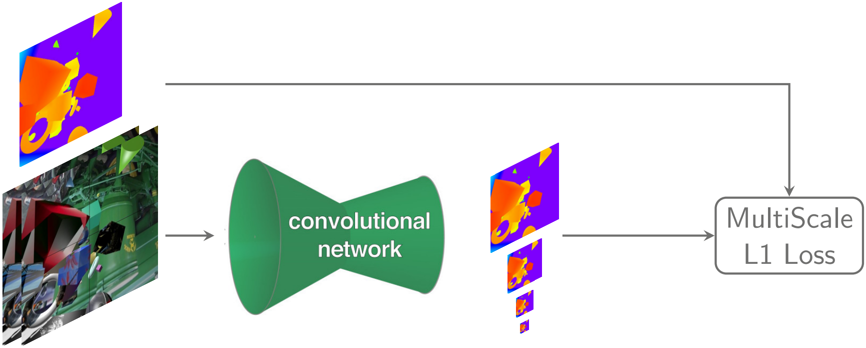
$$\zeta_i' = \frac{D_i}{D_0}\zeta_i \tag{1}$$

D_i and ζ_i are known and we want

$$DepthNet(I_{i-1}, I_i) = \zeta_i' \tag{2}$$

Direction ?

Information on displacement direction (and thus FOE Φ) is **NOT** given



- Training and Network Fully Convolutionnal architecture are both inspired from FlowNetS (Fischer et al. 2015), minimizing a multiscale absolute error
- Training takes about a day on a single Nvidia GTX 980Ti

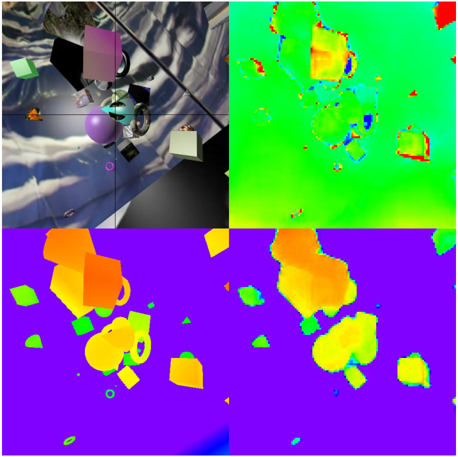
Outline

1. Motivations and Technological Context
2. Supervised Depth Training
3. Results
   - Raw results
   - Varying Speed usecase

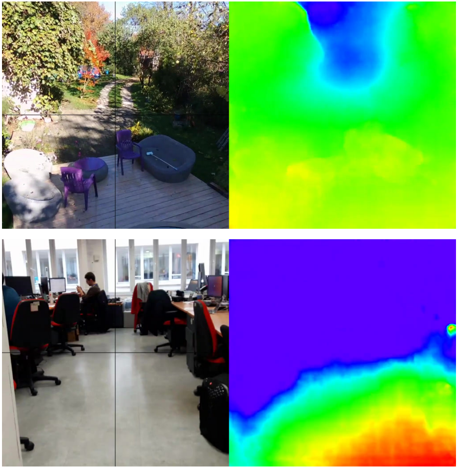
quantitative results

Numerical results

- Error is less than 2.50m for the validation dataset on values ranging from 0 to 100m on 512 × 512 px image pairs
- **10fps** on a TX1 for 512 × 512 px image pairs, **40fps** for 256 × 256



qualitative raw results



real footage

Drone video and handheld stabilized guimbal with unknown speed (assumed constant)

Varying Speed usecase

Compensating depth

Knowing Displacement from a real footage we can deduce real depth map

$$\zeta(t) = \frac{D_t}{D_0} DepthNet(I_t, I_{t-1}) \tag{3}$$

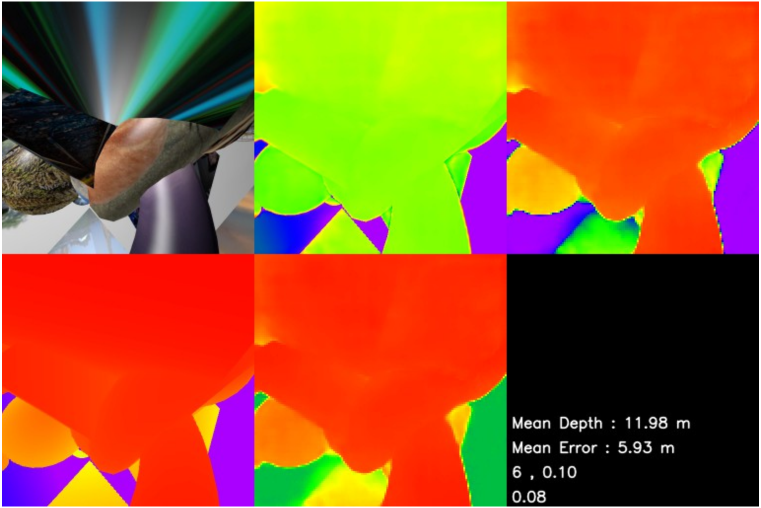
Optimal temporal shift

In order to have an optimal frame pair, we can change the shift to keep DepthNet's output within its typical range (0 to 100m)

$$\Delta_{t+1} \;\; such\ that \;\; D_{\Delta_{t+1}} = D_{\Delta_t} \frac{E_\zeta}{E_0}$$

where E_0 is an ideal mean (here 50m), and E_ζ is the mean of precedent output

Input Error Raw Output



Ground Truth Post processed Statistics

Conclusion and future work

- Getting a dense quality depth map from an image pair is possible solely with convolutions
- The FOE dead zone is solved, allowing obstacle avoidance applications
- Fine tuning on real videos might be to consider

- Still Box Dataset avalaible to download soon !
- Obstacle avoidance proof of concept available on demand, featuring a bebop and a laptop

Thank You !