Structure from Motion

Master's Thesis Presentation

Andreas Weishaupt

Supervisor:

Prof. Pierre Vandergheynst

PhD Assistants:

Luigi Bagnato

Emmanuel D'Angelo

Signal Processing Laboratory

Section of Electrial Engineering – École polytechnique fédérale de Lausanne

February 3, 2010

Imagine what it would mean to…

… a car driver if he doesn't have to drive anymore

… an architect if he could automatically create a
3D model from photographs

… a surgeon if he had access to accurate 3D models
of his patient's organs

…

Enter the world of structure from motion!

1. TV-L^1 Optical Flow

A good point of departure

--->>> a brief demonstration <<<---

Optical flow $\mathbf{u} = (u_x, u_y)$

- relative movement of brightness pattern between two successive images
- inverse problem
- building block for many algorithms

1. TV-L^1 Optical Flow

TV-L^1 optical flow optimization problem:

$$\mathbf{u}^* = \arg\min_{\mathbf{u}} \int_{\Omega} \left\|\nabla \mathbf{u}\right\|_1 + \lambda \left\|\rho(\mathbf{u}, I_0, I_1)\right\|_1 \, d\Omega$$

Solution by Pock respectively Zach et al.:

- convex relaxation
- dual formulation of TV-norm
- soft-thresholding + Chambolle algorithm
- multi-scale resolution scheme
- real-time implementation on GPU

2. Traditional automatic 3D Reconstruction

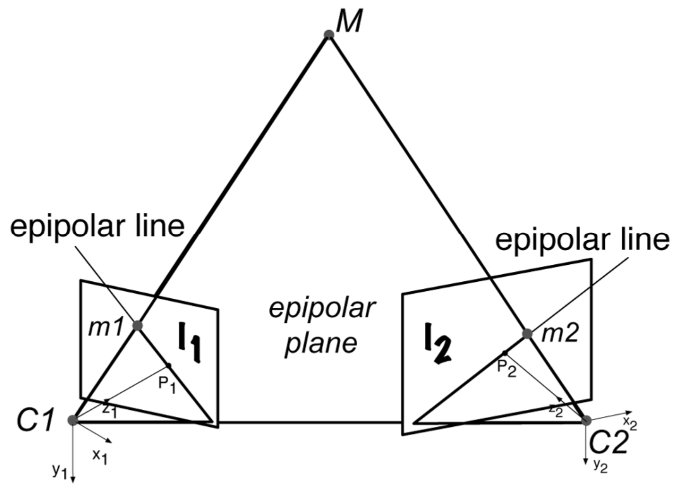
Objective: calculate **depth** of objects in the scene

How: exploit parallax given by two or more images

- Multi-view stereo methods
- Structure from motion

Problems:

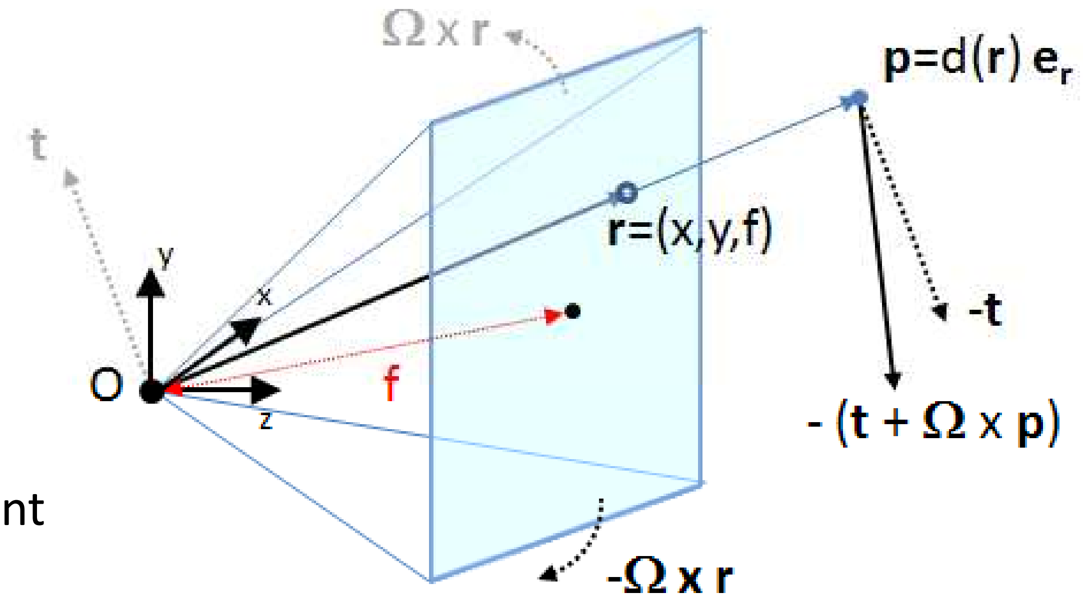
- Find correspondences
- Epipolar geometry



3. Projection Model for Planar Image Sensors

Optical flow/motion field <-> **camera movement+depth**

Pinhole camera model and camera movement:



f: focal

t: camera movment

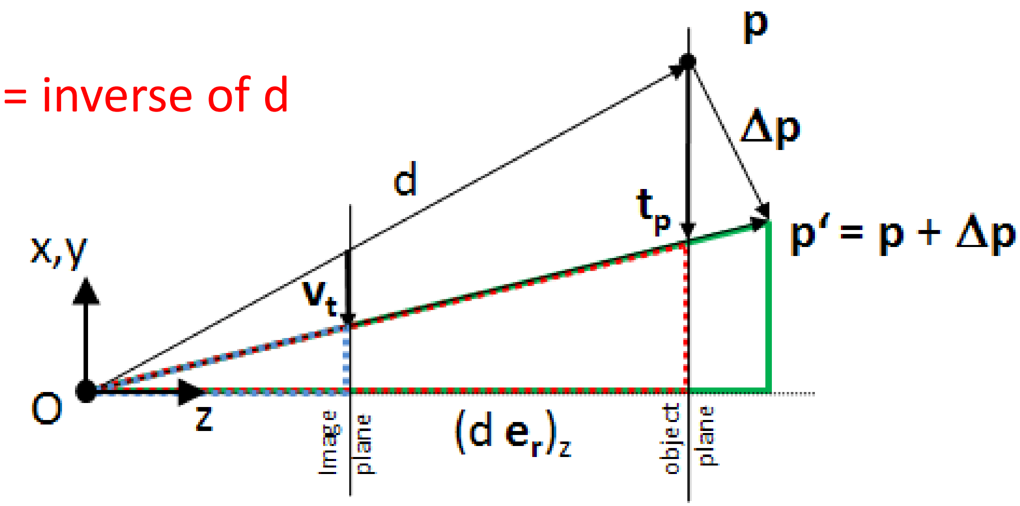
d: distance/depth

- Central projection
- Interchangeability camera <-> scene

3. Projection Model for Planar Image Sensors

In two steps from motion to optical flow:

Z: **depth map** = inverse of d



1. Parallel projection $\mathbf{t_p} = \frac{r_z}{\|\mathbf{r}\|Z} \cdot \frac{(\mathbf{r})_{x,y} + \|\mathbf{r}\|Z(\mathbf{t})_{x,y}}{(\mathbf{r})_z + \|\mathbf{r}\|Z(\mathbf{t})_z} - \frac{(\mathbf{r})_{x,y}}{\|\mathbf{r}\|Z}$
2. Optical flow $\mathbf{v_t} = r_z \cdot \frac{(\mathbf{r})_{x,y} + \|\mathbf{r}\|Z(\mathbf{t})_{x,y}}{(\mathbf{r})_z + \|\mathbf{r}\|Z(\mathbf{t})_z} - (\mathbf{r})_{x,y}$ → **nonlinear**

$\Leftrightarrow \boxed{\mathbf{v_t} = \|\mathbf{r}\|Z\mathbf{t_p}}$ → **linearization**

4. Camera Ego-Motion Estimation

Find camera motion parameters $\mathbf{t}$ and $\mathbf{\Omega}$

Image brightness constancy :

$$I_1 + (\|\mathbf{r}\| Z \mathbf{t_p})^T \nabla I_1 - I_0 = 0$$

Special case $\mathbf{t} = (t_x, t_y, 0)^T$ and $\mathbf{\Omega} = (0, 0, 0)^T$:

→ **linear: solve by linear least-squares**

General case:

→ nonlinear: e.g. use gradient descent

4. Camera Ego-Motion Estimation

Special case t = $(t_x, t_y, 0)^T$ and $\boldsymbol{\Omega} = (0, 0, 0)^T$:

Solve for $\mathbf{A(x)b} = \mathbf{c(x)}$ for $\mathbf{b} = (t_x, t_y)^T$ with

$$\mathbf{A(x)} = \begin{pmatrix} \sum_{\mathbf{x}\in D} \|\mathbf{r}\|^2 Z^2 (\partial_x I)^2 & \sum_{\mathbf{x}\in D} \|\mathbf{r}\|^2 Z^2 \partial_x I \partial_y I \\ \sum_{\mathbf{x}\in D} \|\mathbf{r}\|^2 Z^2 \partial_x I \partial_y I & \sum_{\mathbf{x}\in D} \|\mathbf{r}\|^2 Z^2 (\partial_y I)^2 \end{pmatrix}$$

$$\mathbf{c(x)} = \begin{pmatrix} -\sum_{\mathbf{x}\in D} \|\mathbf{r}\| Z \partial_t I \partial_x I \\ -\sum_{\mathbf{x}\in D} \|\mathbf{r}\| Z \partial_t I \partial_y I \end{pmatrix}$$

4. Camera Ego-Motion Estimation

General case $\mathbf{x} = (t_x, t_y, t_z, \Omega_x, \Omega_y, \Omega_z)^T$:

Gradient descent:

$$\mathbf{x}^{n+1} = \mathbf{x}^n + \gamma \nabla E(\mathbf{x}^n)$$

$$\frac{\partial E}{\partial x_i} = \sum_{\mathbf{x} \in D} \left(\partial_t I + \nabla I_1^T \mathbf{u} \right) \nabla I_1^T \frac{\partial \mathbf{u}}{\partial x_i}$$

$$\frac{\partial \mathbf{u}}{\partial x_i} = \frac{\partial}{\partial x_i} \left(\|\mathbf{r}\| Z \mathbf{t_p} \right)$$

5. TV-L^1 Depth from Motion Estimation

TV-L^1 depth from motion optimization problem:

$$Z^* = \arg\min_Z \int_\Omega \|\nabla Z\|_1 + \lambda \|\rho(Z, I_0, I_1)\|_1 \, \mathrm{d}\Omega$$

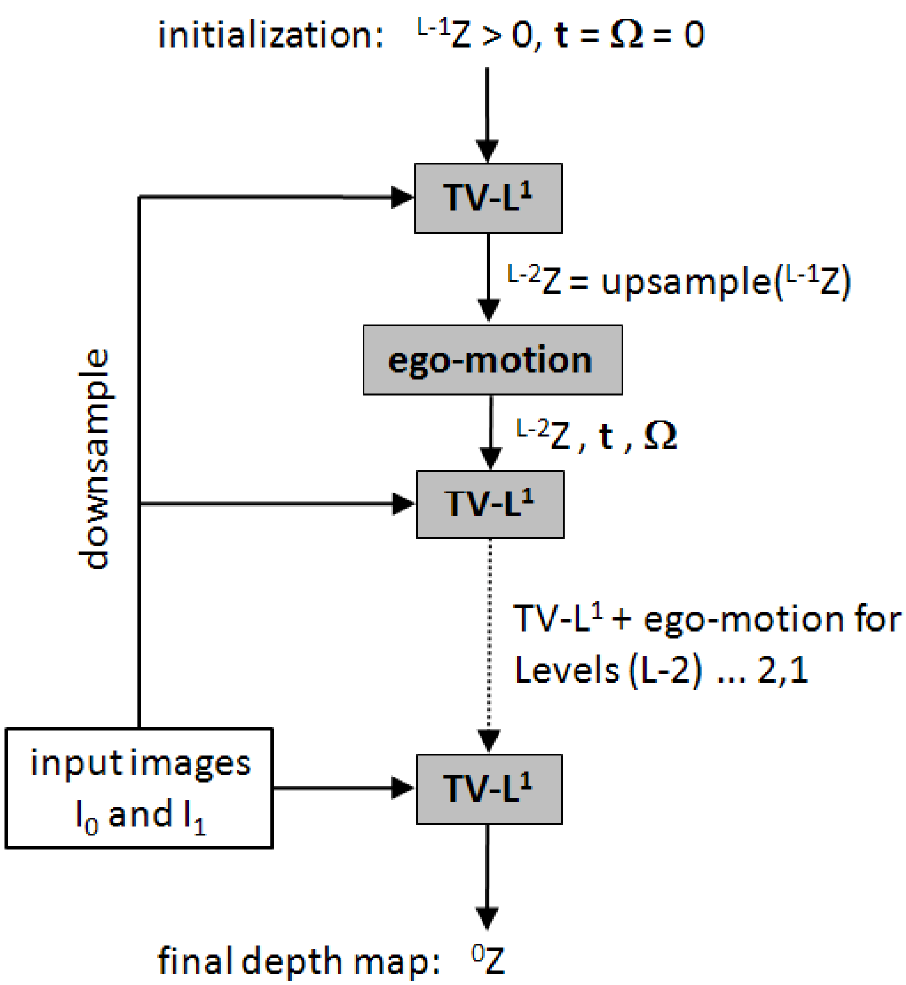
Differences to TV-L^1 DfM on the sphere:

→ linearized dependence on Z

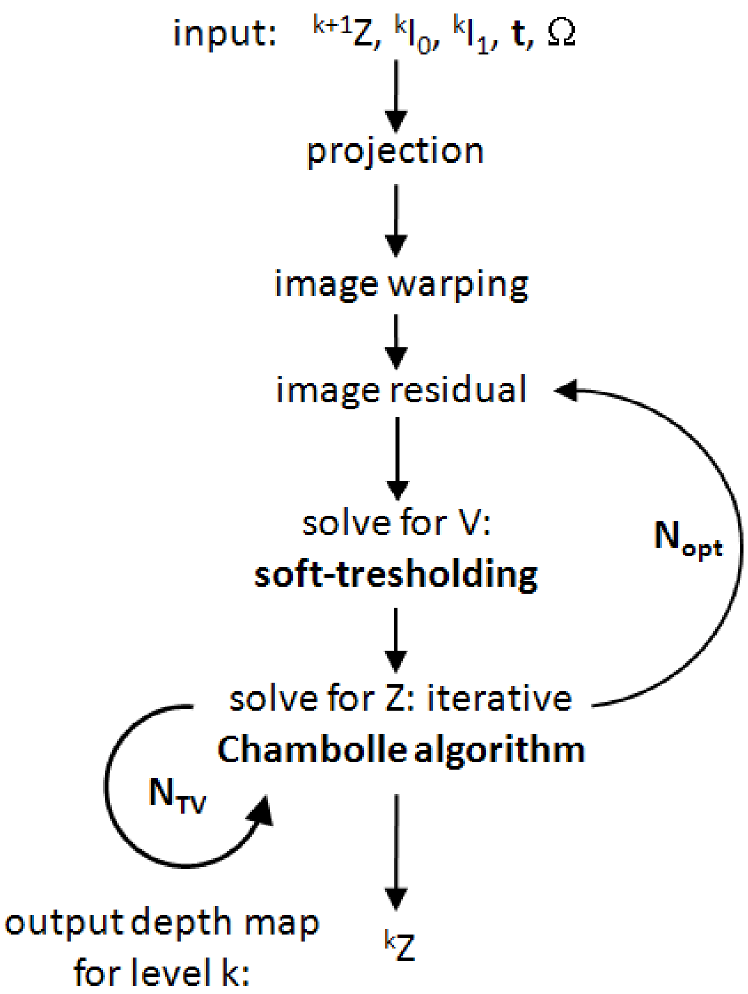
→ slight modification of soft-thresholding

6. Joint Depth and Ego-motion Estimation

Multi-scale / pyramid resolution scheme



TV-L^1 depth from motion

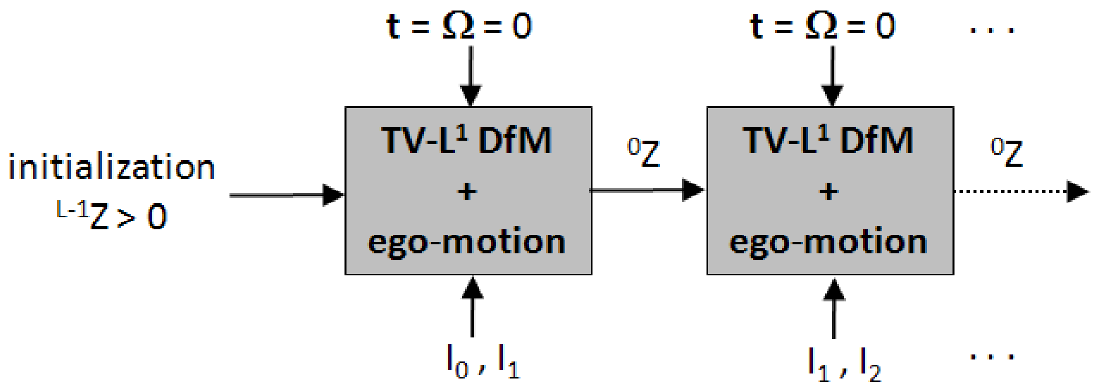


6. Joint Depth and Ego-motion Estimation

Multiple images / video sequence

→ propagate solution

→ initialize motion parameters with zero

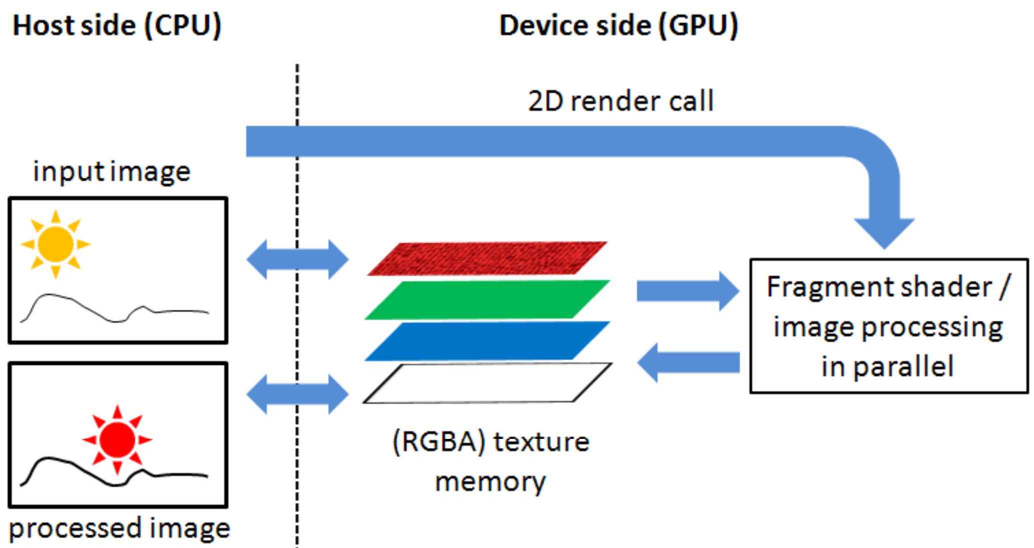


→ eventually predict solution

→ add tracker

7. GPU Implementation

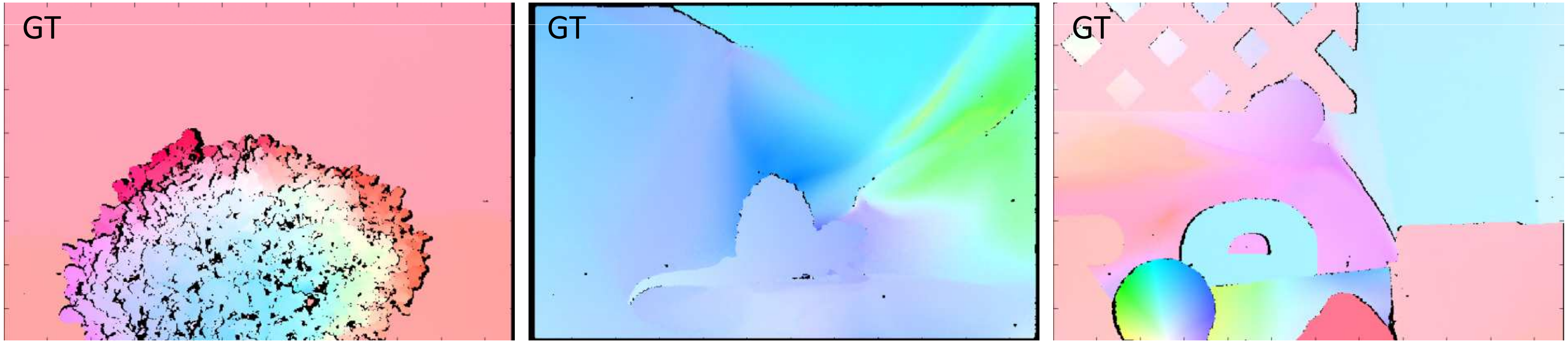
Simple image processing on the GPU:



1. Host : I/O + render calls
2. Device: **parallel processing** per pixel by fragment shading kernels

8. Results

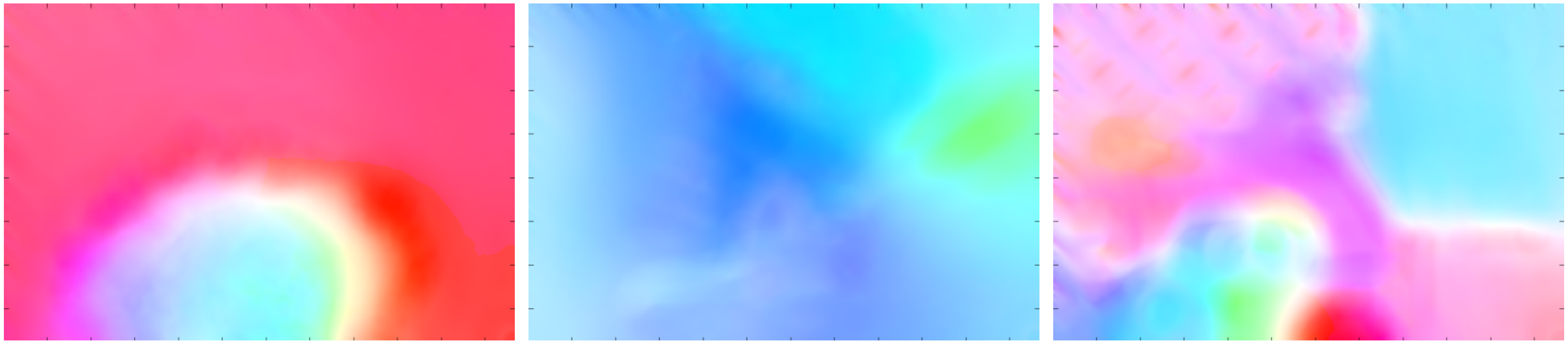
Optical flow : Middlebury training set (real sequences)



Hydrangea
AEE = 0.89, AAE = 12.5°

Dimetodron
AEE = 0.52, AAE = 10.2°

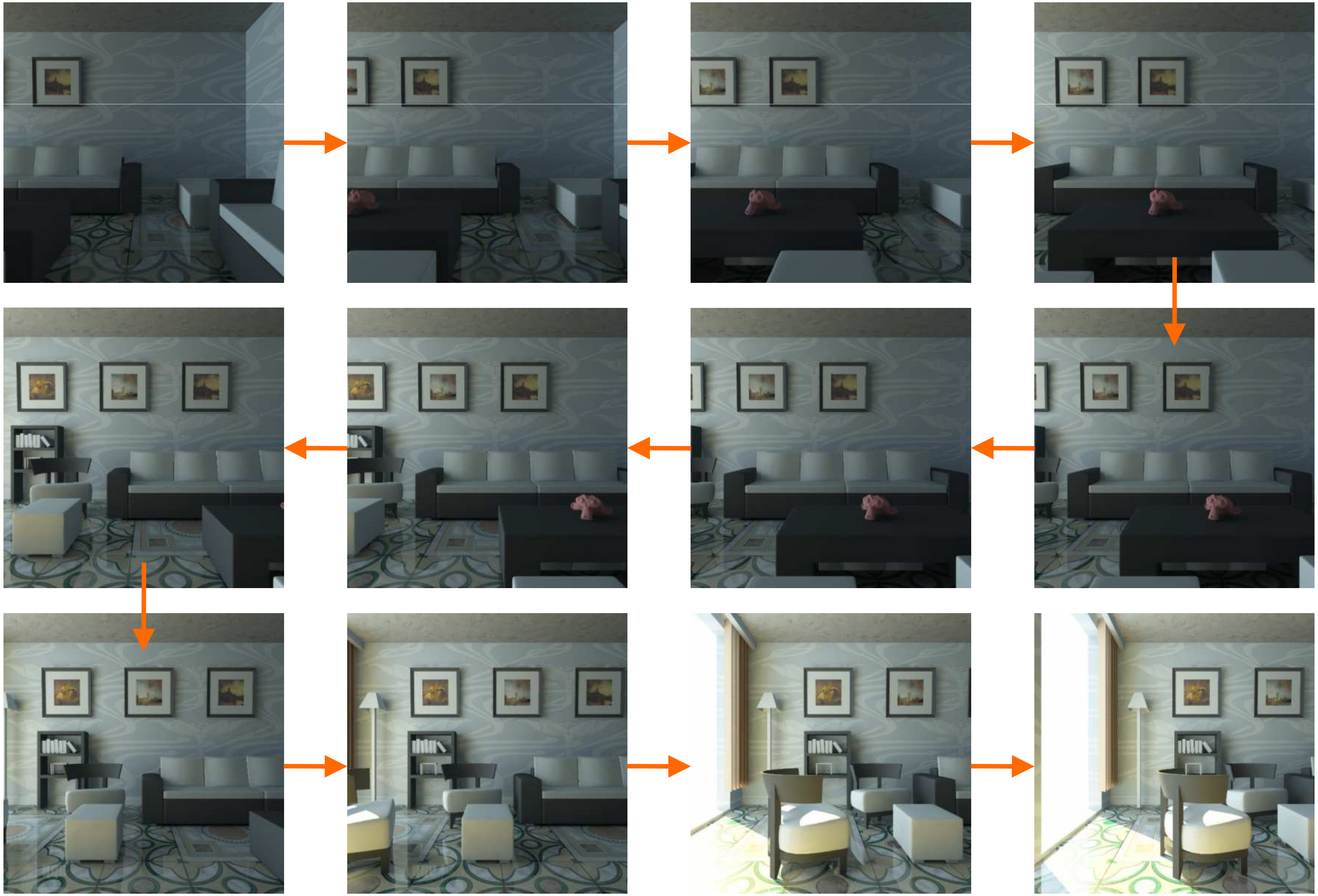
Rubberwhale
AEE = 0.62, AAE = 21.3°



Used parameters: 5 resolution levels, 33 global iterations, 3 Chambolle iterations, λ=1.0, θ=1.0

8. Results

(the synthetic room sequence – **Camera moving left**)



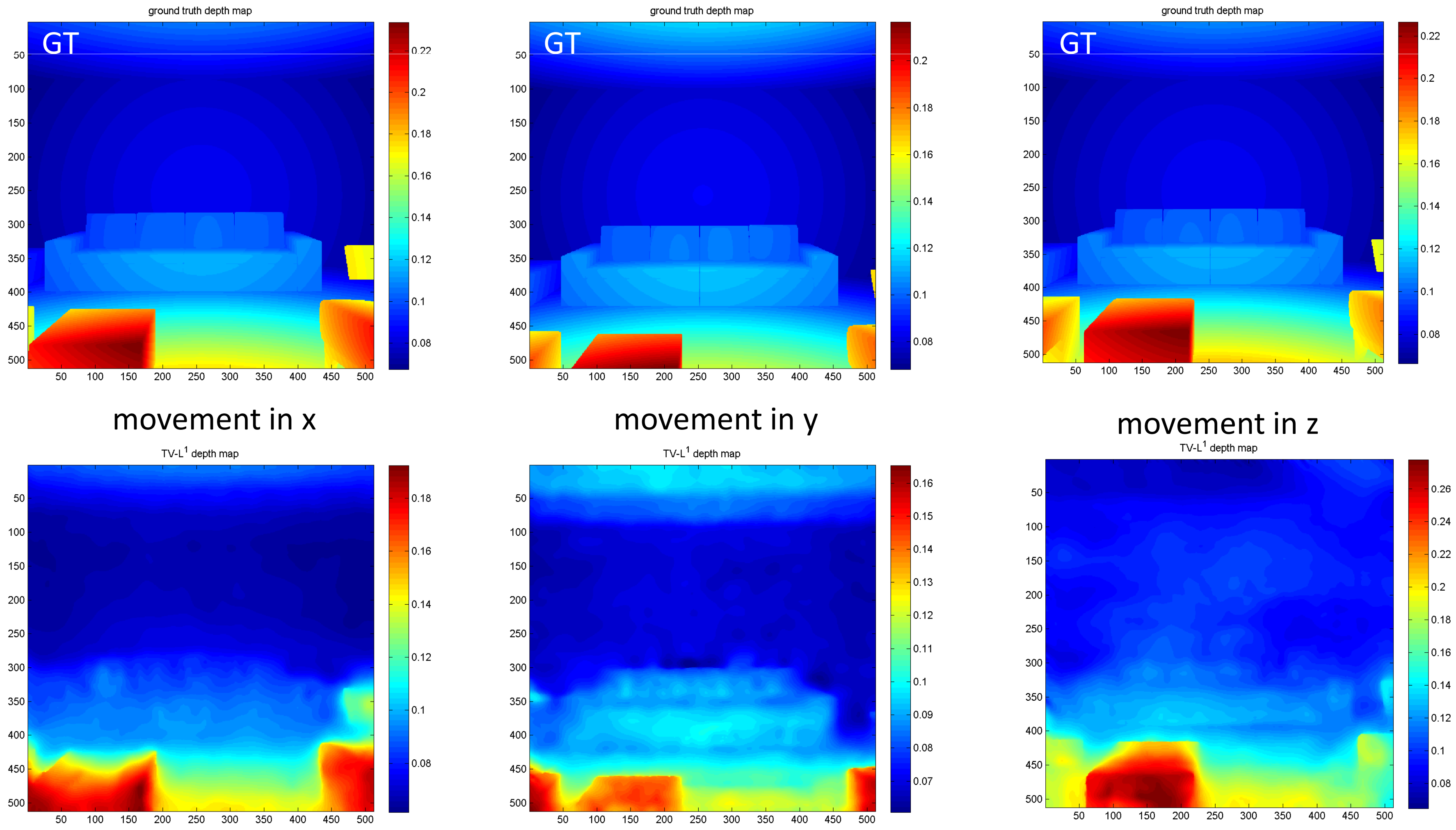
8. Results

Ego-motion estimation

Sequence resp. Camera movement	True t = (t_x,t_y,t_z)	median(t)
X	(-1, 0, 0)	(-1.00, 0.03,-0.13)
Y	(0,-1, 0)	(0.02,-1.00,-0.40)
Z	(0, 0,-1)	(0.05,-0.05,-1.00)
X+Z	(-1, 0,-1)	(-0.95, 0.23,-1.00)
Y+Z	(0,-1,-1)	(0.10,-1.00,-0.09)
Y+Y+Z	(-1,-1,-1)	(-0.36,-1.00,-0.45)

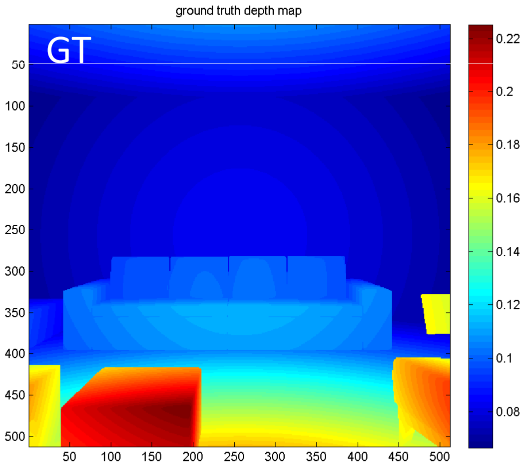
8. Results

Depth from motion : simple camera movement

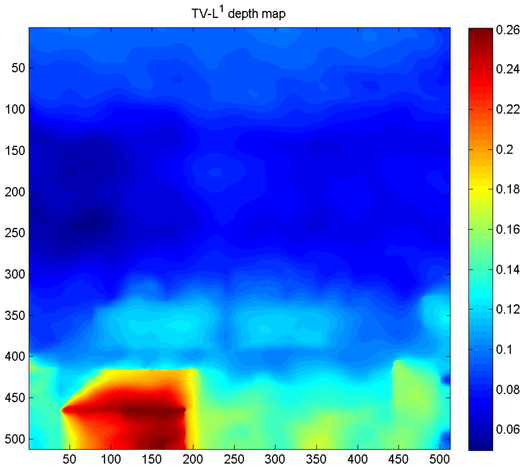


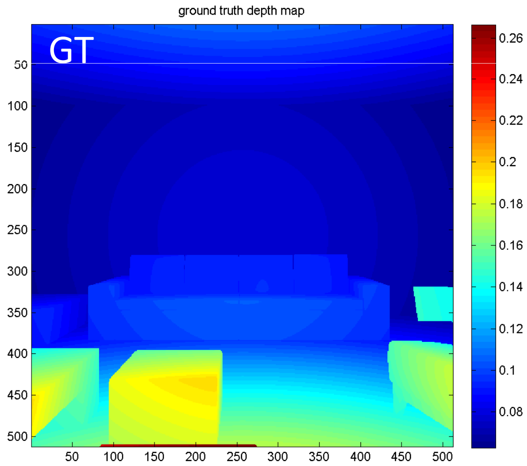
8. Results

Depth from motion : **“combined”** camera movement

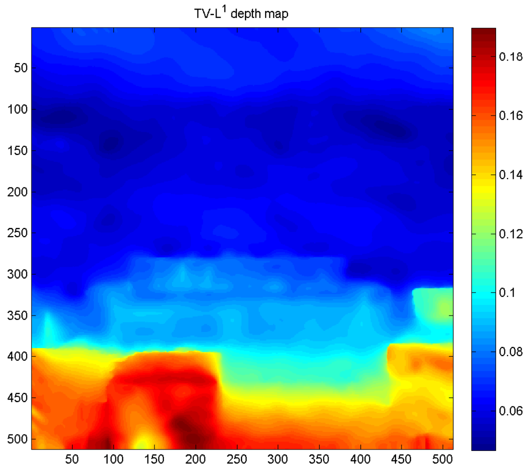


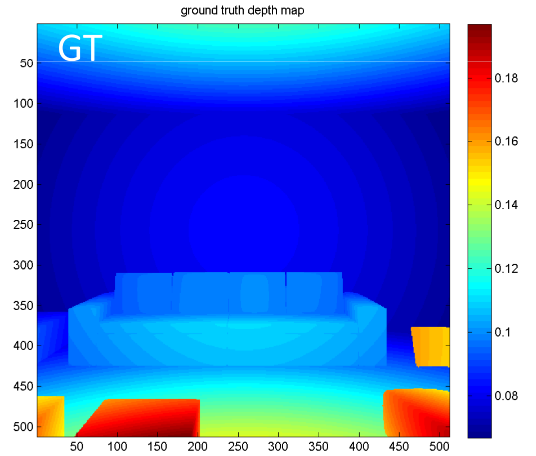
movement in x+z



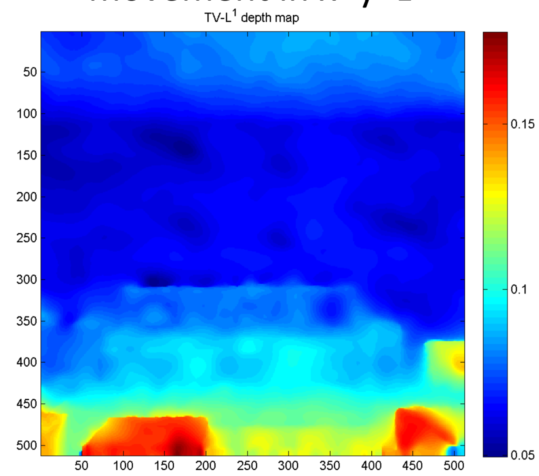


movement in y+z



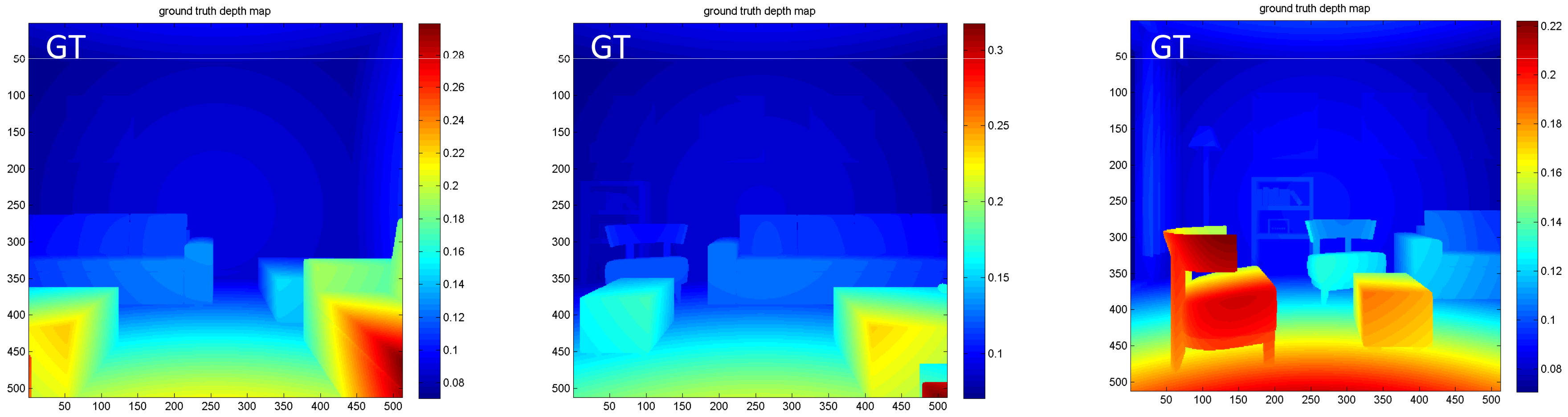


movement in x+y+z

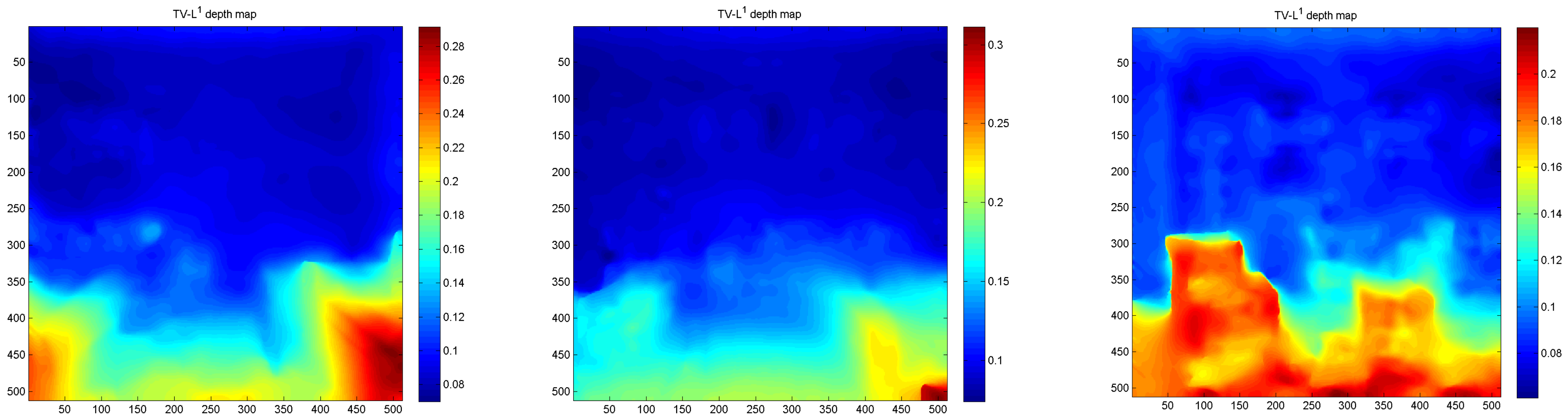


8. Results

Joint ego-motion and depth from motion estimation



Camera movement in x for all frames



9. Future Prospects and Conclusion

- The proposed algorithm for planar SfM
  - promising
  - slow with respect to convergence if $\theta \rightarrow 0$
  - no camera rotation
  - only for synthetic images
  - fails when ego-motion fails
- Future work
  - rotation
  - robust tracking for ego-motion
  - camera calibration
  - GPU implementation

Thank you for your attention

Any questions?

→ andreas.weishaupt@epfl.ch ←

References

1. L. Bagnato, P. Vandergheynst, and P. Frossard. A Variational Framework for Structure from Motion in Omnidirectional Image Sequences. IEEE Transations on Image Processing, 2009.
2. C. Zach, T. Pock, and H. Bischof. A duality based approach for realtime TV-L1 optical flow. In DAGM-Symposium, pp.214–223, 2007
3. S. Baker, D. Scharstein, J. Lewis, S. Roth, M. J. Black, and R. Szeliski. A database and evaluation methodology for optical flow. Microsoft Corporation, 2009.
4. B. K. P. Horn. Robot Vision. The MIT Press, 1986.
5. M. Pharr and R. Fernando. GPU Gems 2: Programming Techniques for High-Performance Graphics and General-Purpose Computation. Addison-Wesley Professional, 2005.