SfM-Net: Learning of Structure and Motion from Video

Sudheendra Vijayanarasimhan*
svnaras@google.com

Susanna Ricco*
ricco@google.com

Cordelia Schmid†*
cordelia.schmid@inria.fr

Rahul Sukthankar*
sukthankar@google.com

Katerina Fragkiadaki‡
katef@cs.cmu.edu

published in arXiv preprint, 25/07/2017

Seminar: Recent Advances in 3D Computer Vision

Presented by Oliver Lengwinat, 04/07/2018

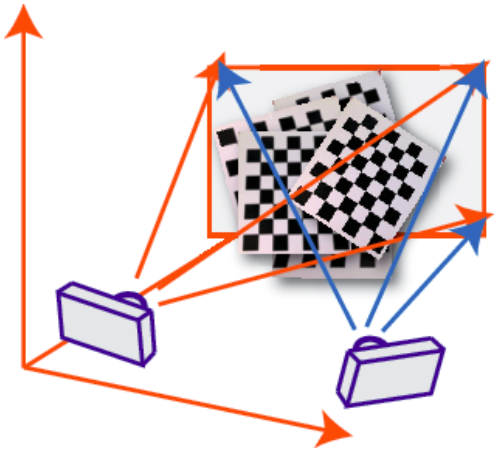
Agenda

- Motivation and previous approaches
- Proposed solution
- Results
- Conclusion
- Outlook

Depth and camera motion estimation in videos

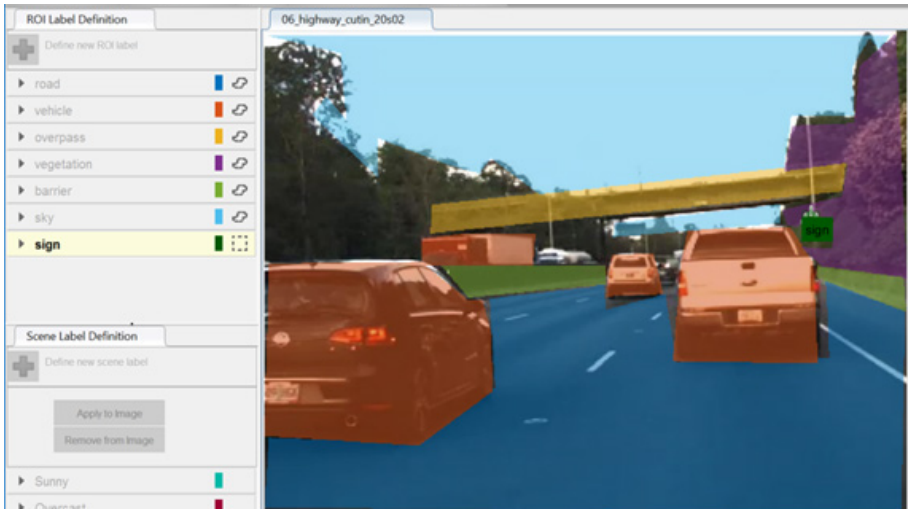
Previous approaches:

require extensive engineering



or

require expensive training sets



Images taken from mathworks.com

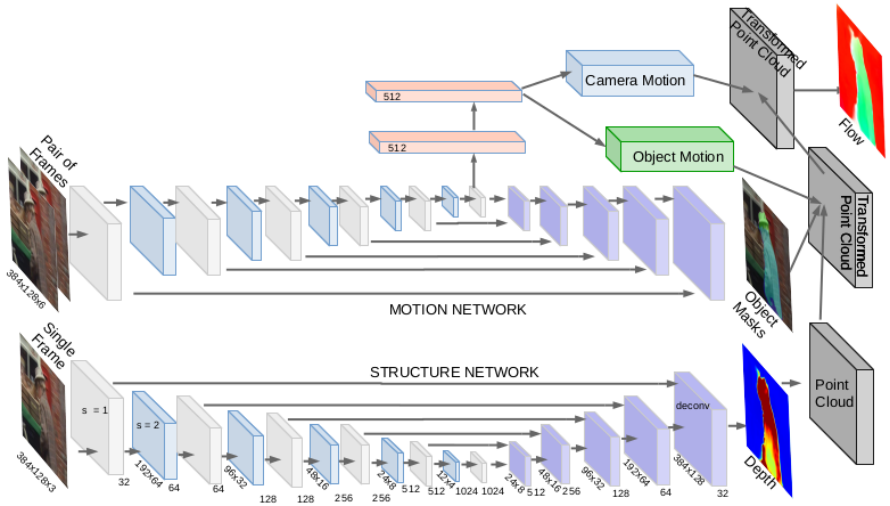
Proposed solution in ‘Unsupervised Learning of Depth and Ego-Motion from Video’ (Zhou et al., 2017)

Problem: moving objects

Proposed solution in this paper

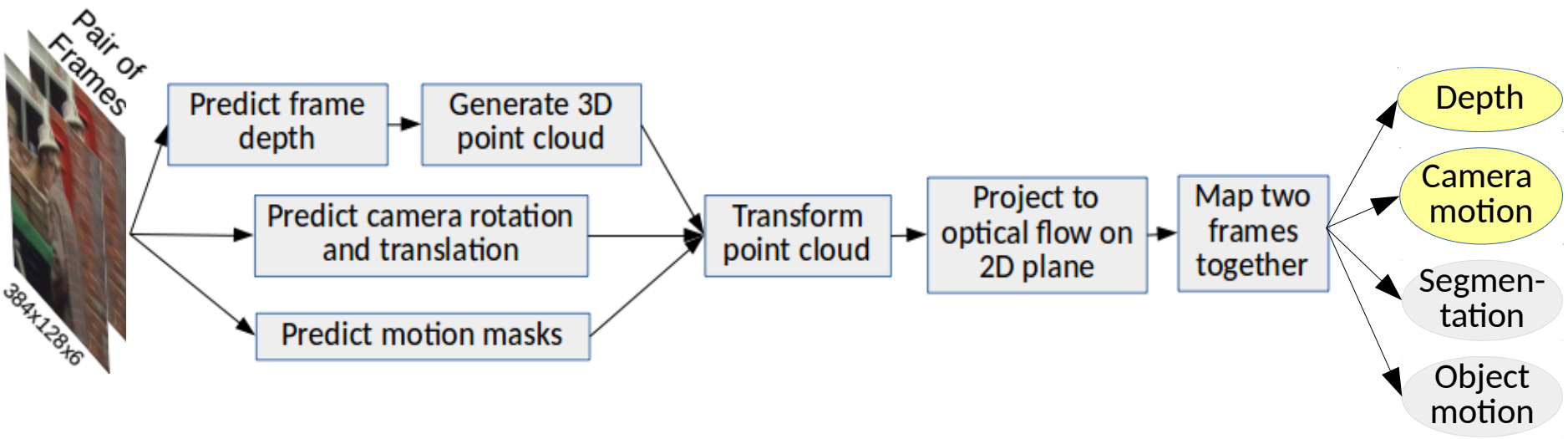
- Learning depth and segmentation in two separate sub-networks
- Model both camera and object motion

→ allows to train on unrestricted videos

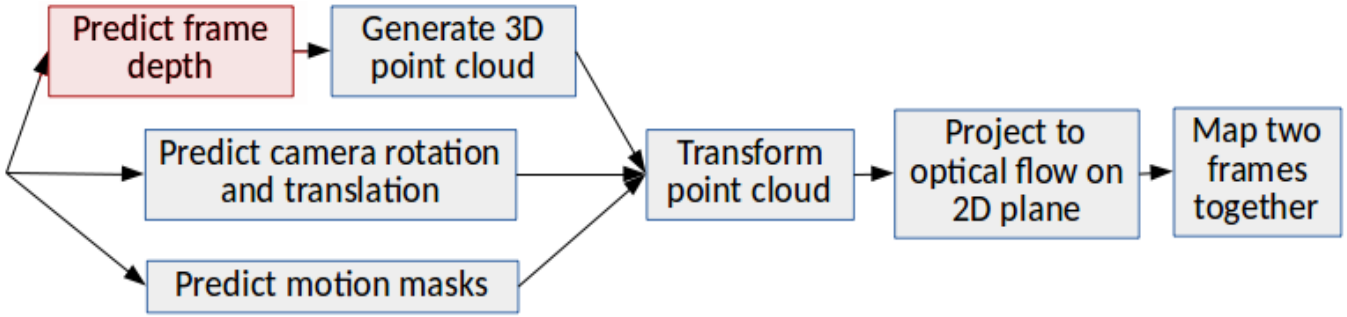


Two sub-networks for motion and for structure

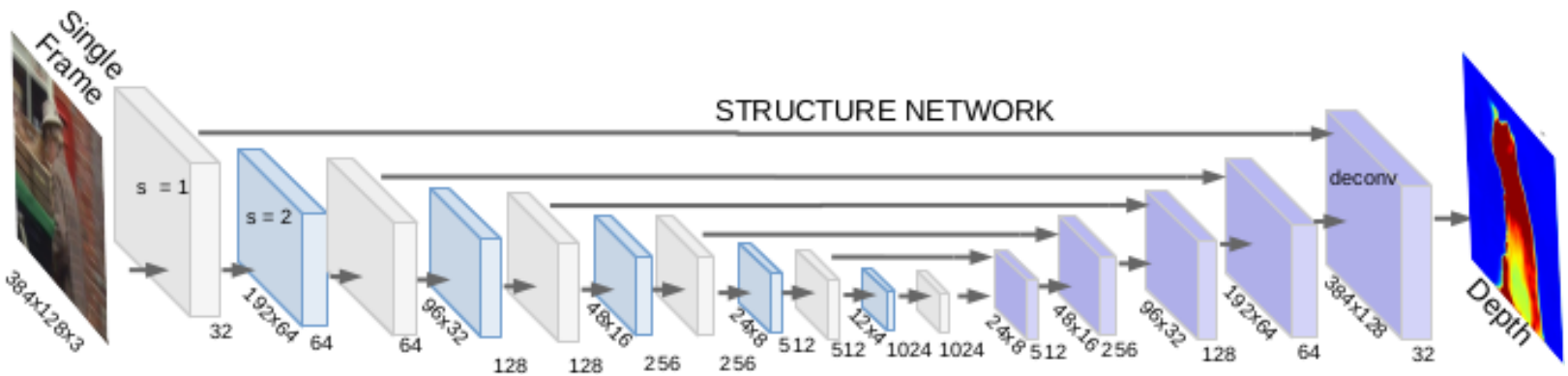
Approach in SfM-Net



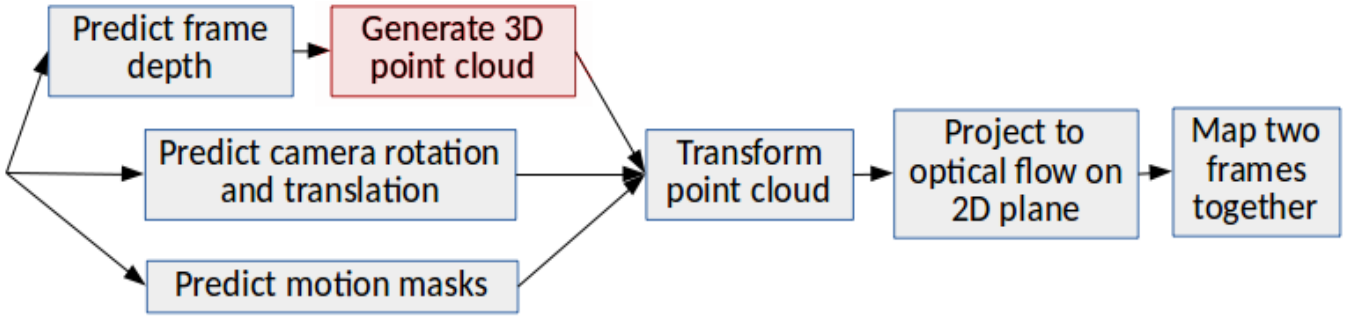
Approach in SfM-Net

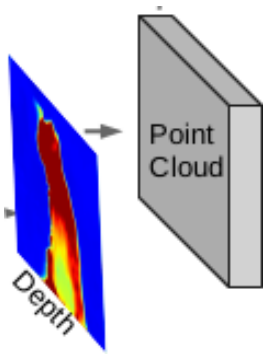


– Using unsupervised learning from single image



Approach in SfM-Net

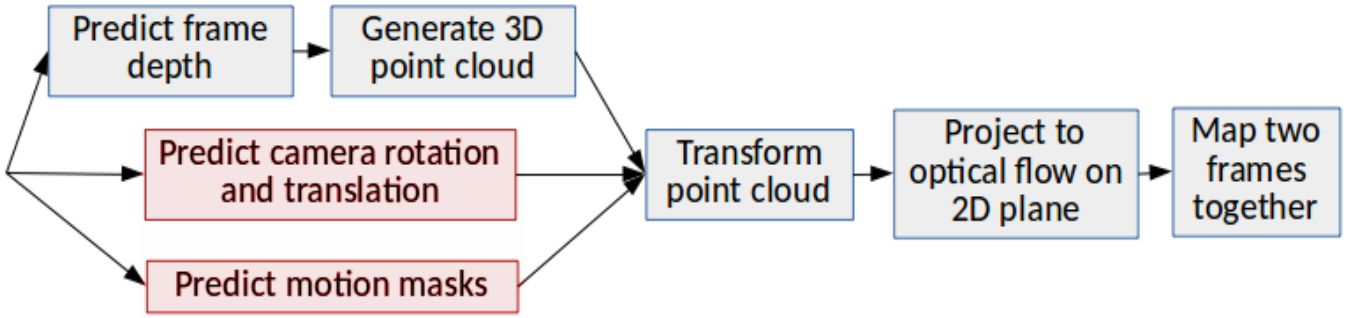


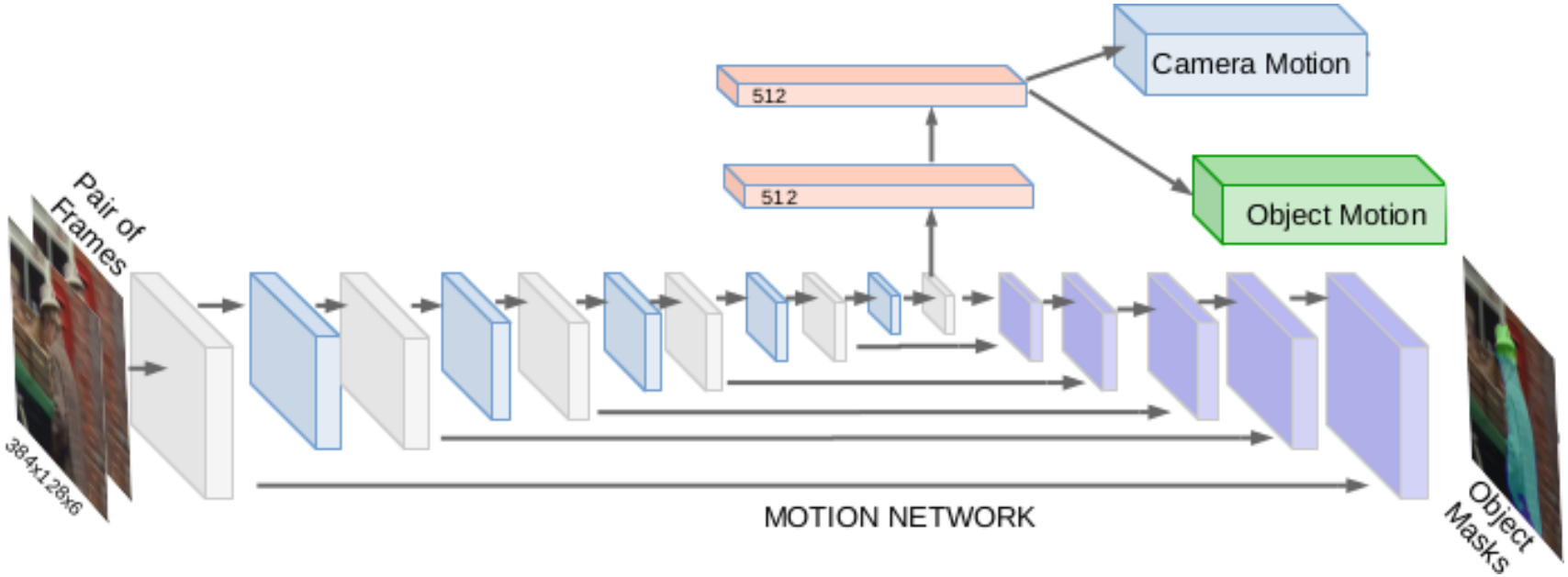


For each point *i*:

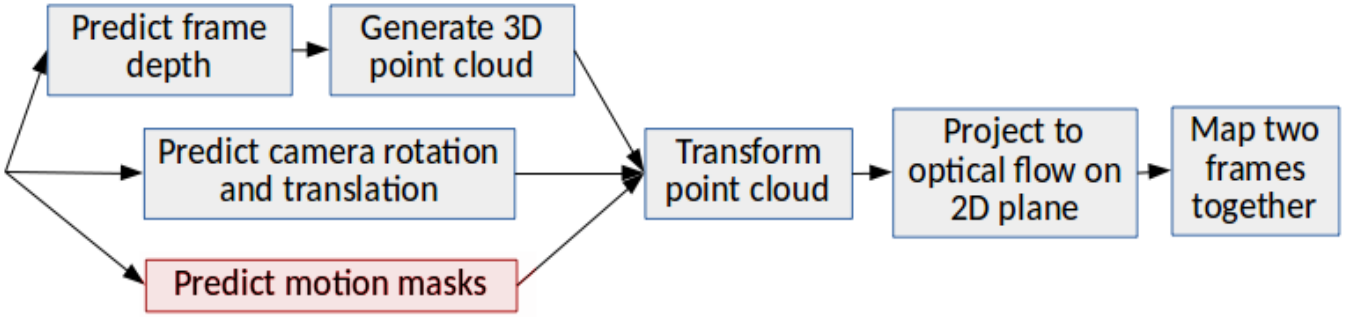
$$\mathbf{X}_t^i = \begin{bmatrix} X_t^i \\ Y_t^i \\ Z_t^i \end{bmatrix} = \frac{d_t^i}{f} \begin{bmatrix} \frac{x_t^i}{w} - c_x \\ \frac{y_t^i}{h} - c_y \\ f \end{bmatrix}$$

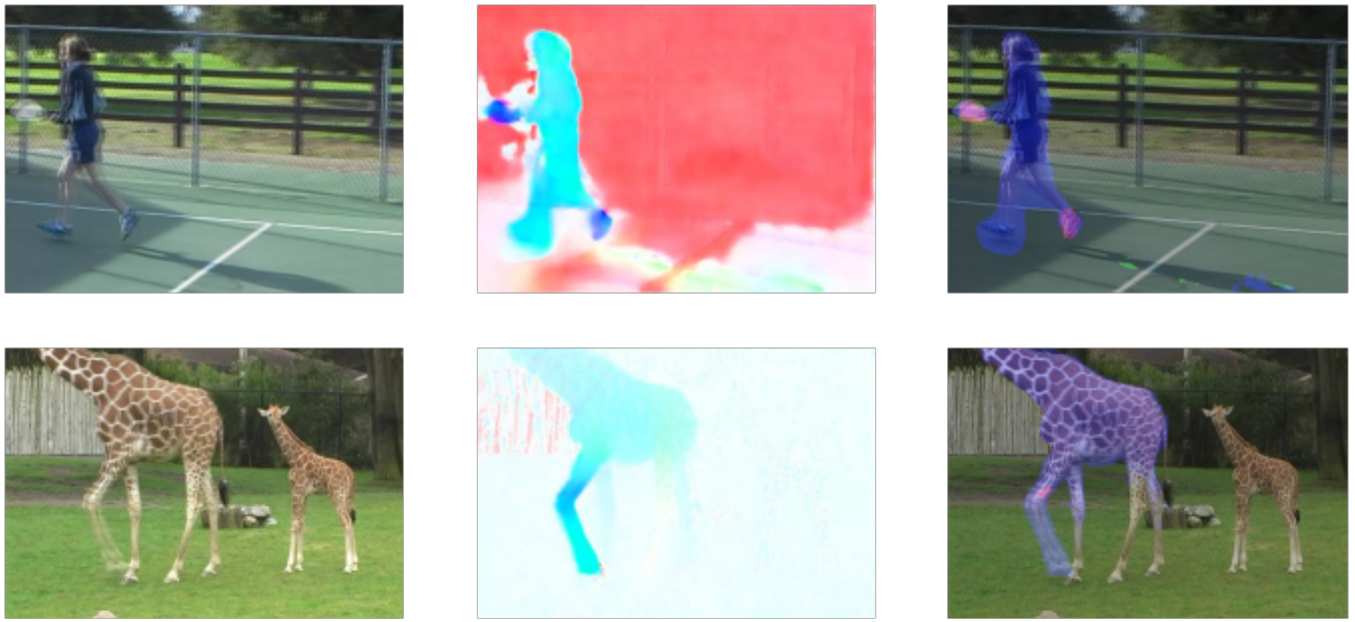
Approach in SfM-Net



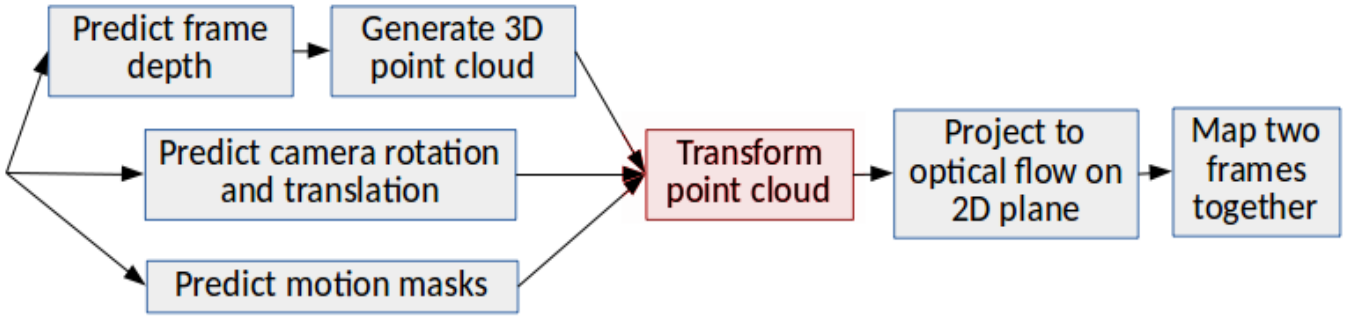


Approach in SfM-Net





Approach in SfM-Net



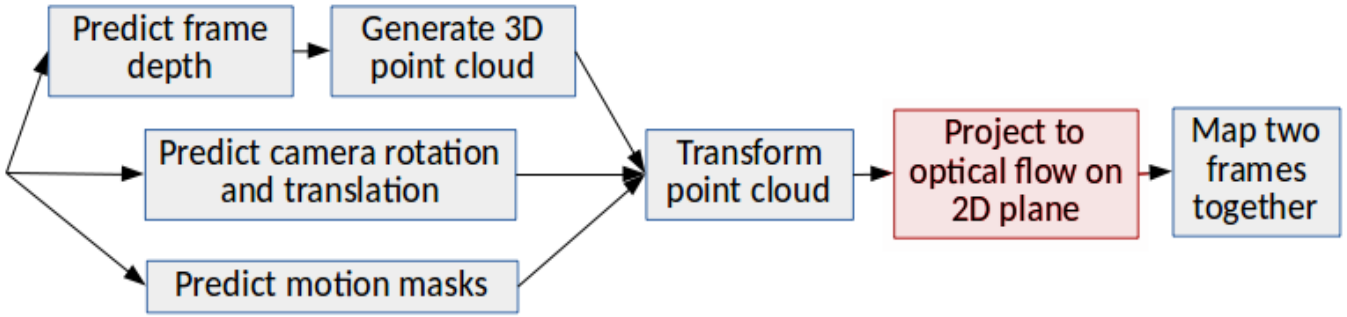
- First, apply object transformations:

$$\mathbf{X}'_t = \mathbf{X}_t + \sum_{k=1}^{\mathrm{K}} \mathrm{m}_t^k(i)(R_t^k(\mathbf{X}_t - p_k) + t_t^k - \mathbf{X}_t)$$

- Then, apply camera transformation:

$$\mathbf{X}''_t = R_t^c(\mathbf{X}'_t - p_t^c) + t_t^c$$

Approach in SfM-Net



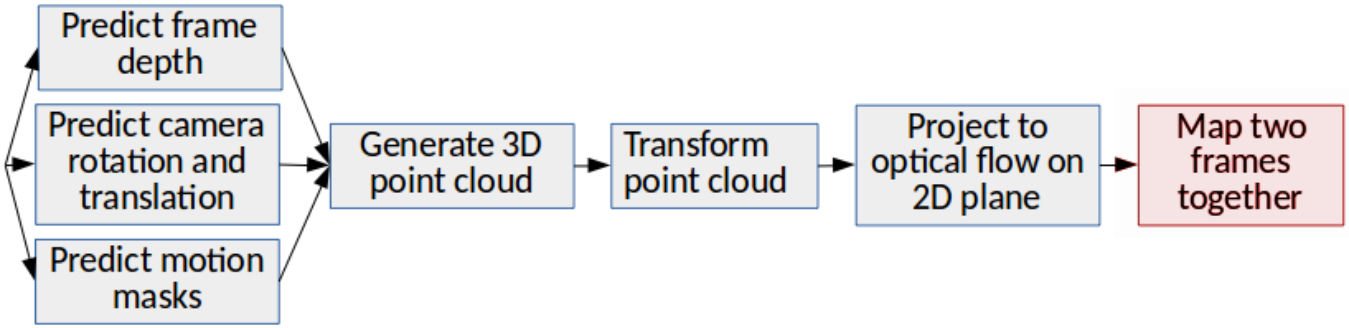
- Projecting each Point $\mathbf{X}_t'' = (X_t'', Y_t'', Z_t'')$ back to the image plane:

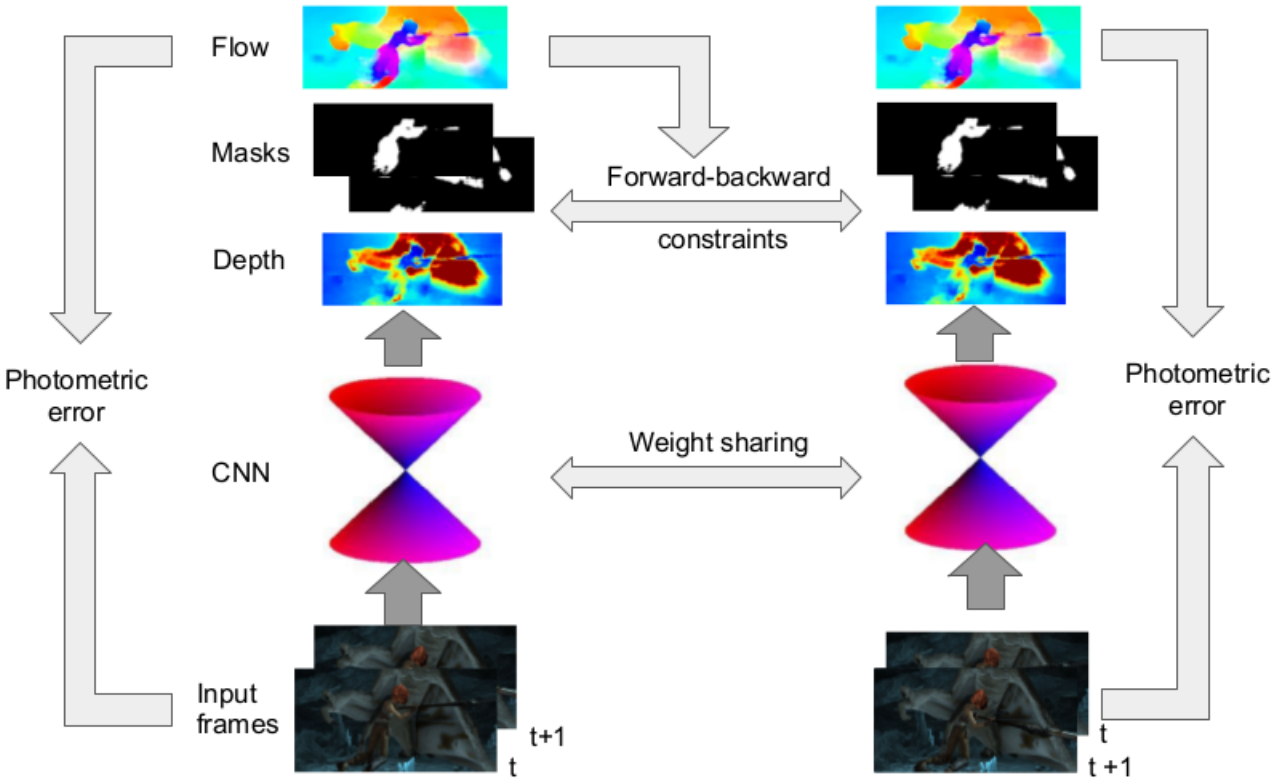
$$\begin{bmatrix} \frac{x_{t+1}^i}{w} \\ \frac{y_{t+1}^i}{h} \end{bmatrix} = \frac{f}{Z_t''} \begin{bmatrix} X_t'' \\ Y_t'' \\ f \end{bmatrix} + \begin{bmatrix} c_x \\ c_y \end{bmatrix}$$

- The flow *U,V* between the two frames at pixel *i* is

$$(U_t(i), V_t(i)) = (x_{t+1}^i - x_t^i, y_{t+1}^i - y_t^i)$$

Approach in SfM-Net





Computing Loss in SfM-Net

- Photometric error:

$$\mathcal{L}_t^{\text{color}} = \frac{1}{w\,h} \sum_{x,y} \|I_t(x,y) - I_{t+1}(x',y')\|_1$$

- Forward-backward consistency constraints:

$$\mathcal{L}_t^{FB} = \frac{1}{w\,h} \sum_{x,y} |\,(\mathrm{d}_t(x,y) + W_t(x,y)) - \mathrm{d}_{t+1}(x + U_t(x,y), y + V_t(x,y))|$$

- Spatial smoothness as in 'Unsupervised Learning of Depth and Ego-Motion from Video'

- Supervising depth (optional):

$$\mathcal{L}_t^{depth} = \frac{1}{w\,h} \sum_{x,y} \mathrm{dmask}_t^{GT}(x,y) \cdot \|\mathrm{d}_t(x,y) - \mathrm{d}_t^{GT}(x,y)\|_1$$

- Supervising camera motion (optional):

$$\mathcal{L}_t^{c_{\text{trans}}} = \|t_t^{\text{err}}\|_2$$

$$\mathcal{L}_t^{c_{\text{rot}}} = \arccos\left(\min\left(1, \max\left(-1, \frac{\mathrm{trace}(R_t^{err}) - 1}{2}\right)\right)\right)$$

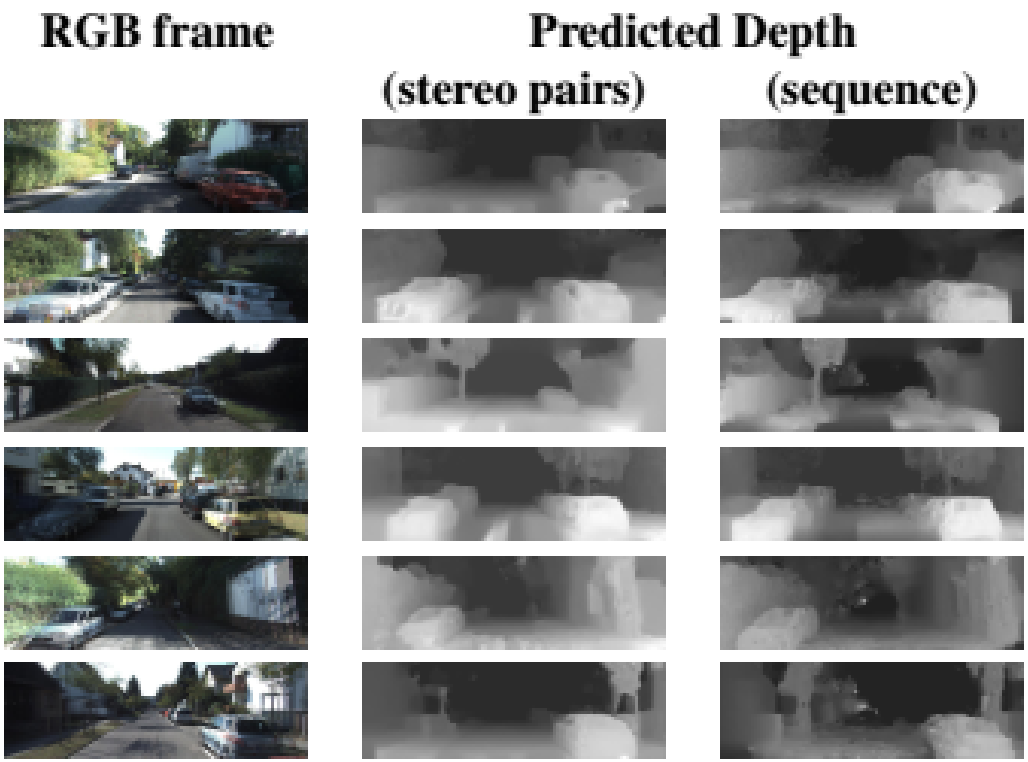
Results

- Depth prediction
- Segmentation
- Camera pose

Prediction: Depth

On KITTI 2012 and 2015:

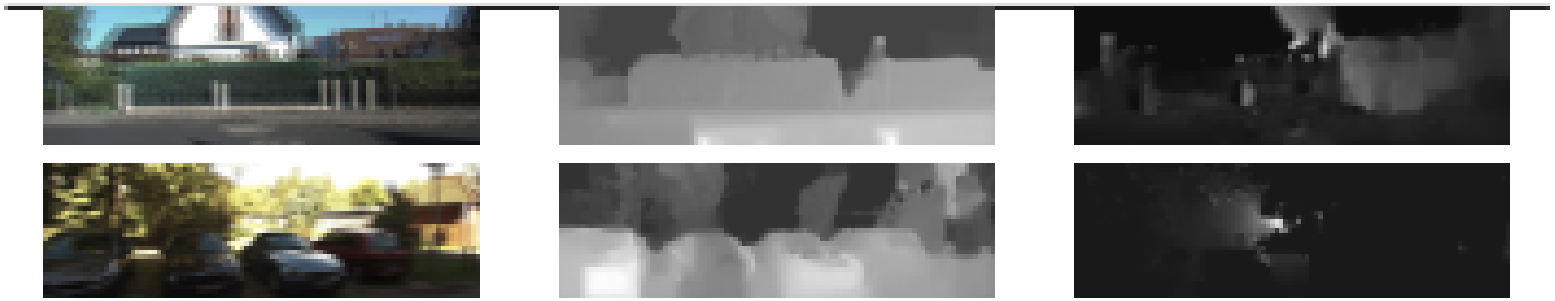
Approach	Log RMSE	
	KITTI 2012	KITTI 2015
with stereo pairs	0.31	0.34
seq. with motion masks	0.45	0.41
seq. without motion masks	0.77	1.25



Prediction: Depth

On KITTI 2012 and 2015:

Approach	Log RMSE	
	KITTI 2012	KITTI 2015
with stereo pairs	0.31	0.34
seq. with motion masks	0.45	0.41
seq. without motion masks	0.77	1.25



Prediction: Segmentation

On MoSeg:

- Includes many non-rigid motions

Intersection over Union (IoU):

ours:
0.29
benchmark (4 masks) :
0.30
best:
0.57

Prediction:
Relative camera pose

On RGB-D SLAM Freiburg I dataset

Seq.	transl [27]	rot [27]	transl. ours	rot ours
360	0.099	0.474	0.009	1.123
plant	0.016	1.053	0.011	0.796
teddy	0.020	1.14	0.0123	0.877
desk	0.008	0.495	0.012	0.848
desk2	0.099	0.61	0.012	0.974

Discussion

Scientific contributions

- Framework to learn depth and camera motion
- Less limitations on videos for training than previous approaches
- More than just depth and camera motion
- Can be used with several degrees of supervision
- Can improve further if more training is done

Scientific contributions

To do this

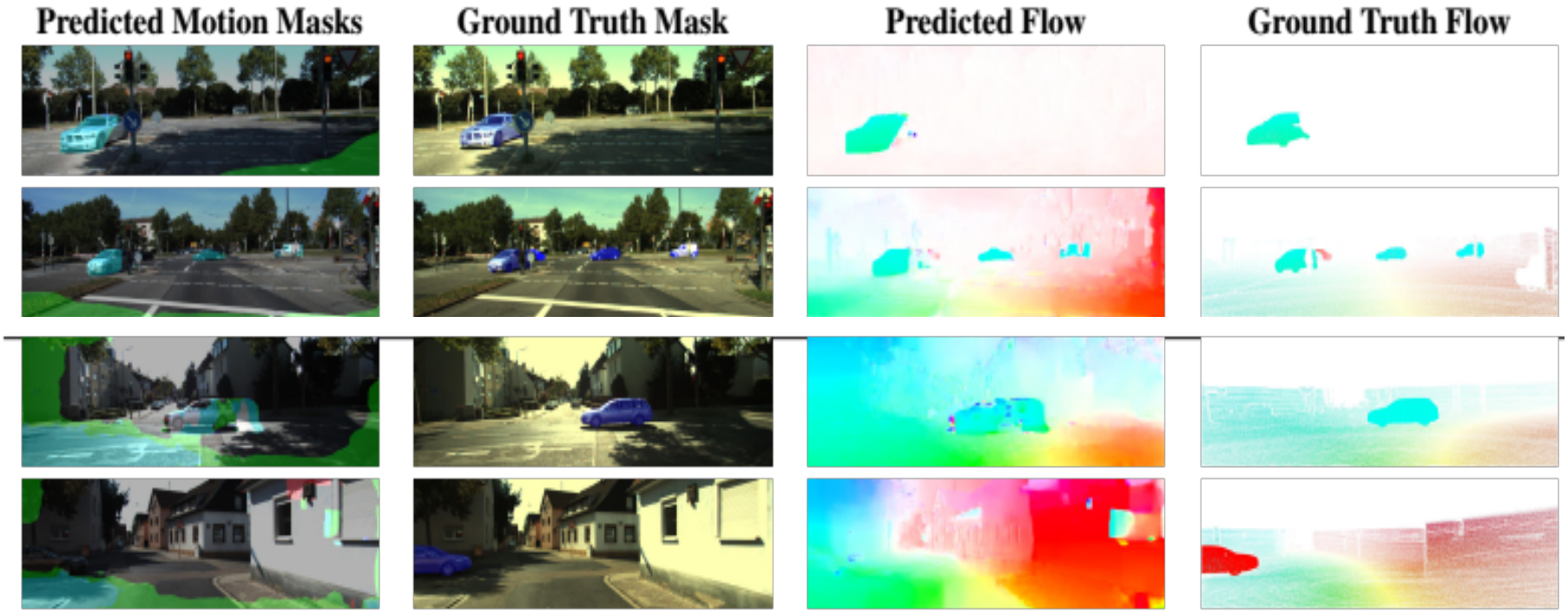
- the geometry of image formation is learned
- pixel-wise depth is predicted from one image
- camera motion, segmentation and object motions are predicted from a pair of images
- forward-backward constraints are applied to learn a *consistent* 3D structure

Discussion

- Mostly combination of these papers:
  - **Unsupervised Learning of Depth and Ego-Motion from Video**
  - **Motion Cooperation: Smooth Piece-Wise Rigid Scene Flow from RGB-D Images**
  - **Unsupervised monocular depth estimation with left-right consistency**
- Limited to only 3 objects
- Only works when camera is moved
- Neglecting small objects (see next slide)

Discussion

Object masks tend to miss very small, distant moving objects.



Segmentation and flow compared with ground truth of KITTI2015

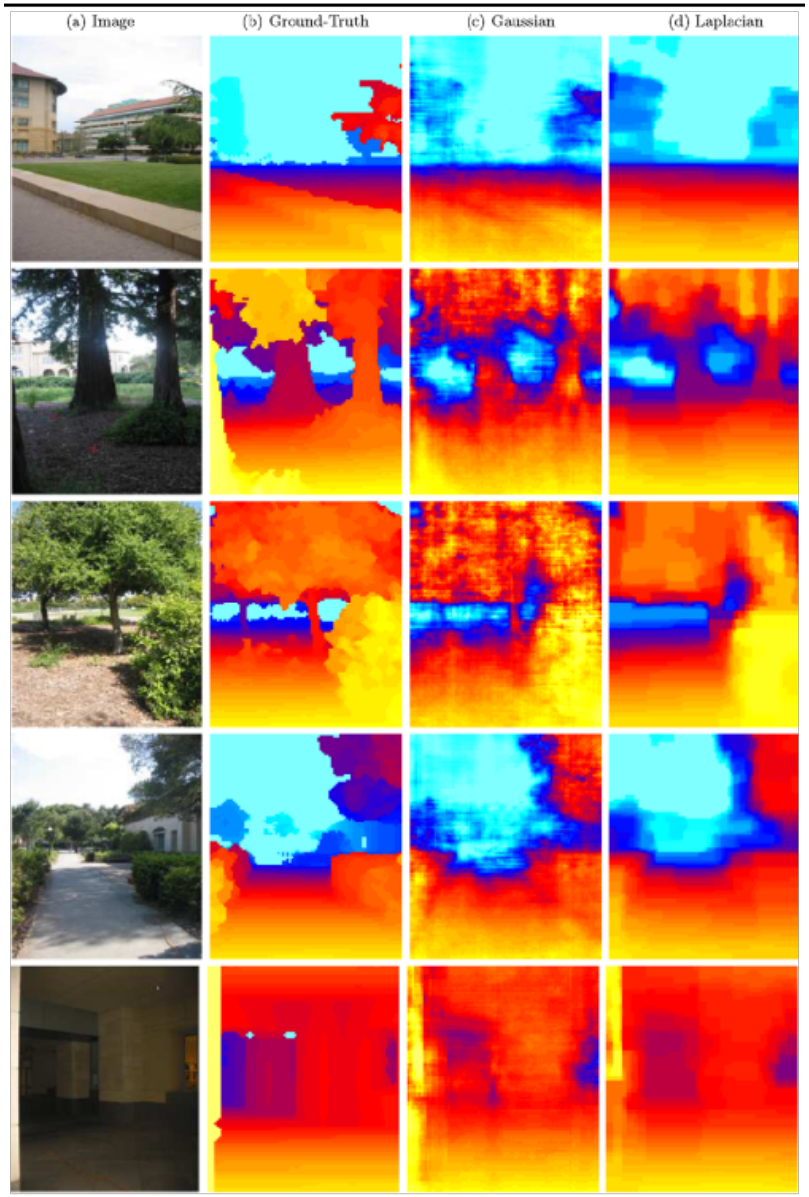
Outlook

- Possible further research:
  - Extend forward-backward constraints
  - Inference problem
  - Methods for pre-training
  - Find a good solution for number of motion masks

Thank you!

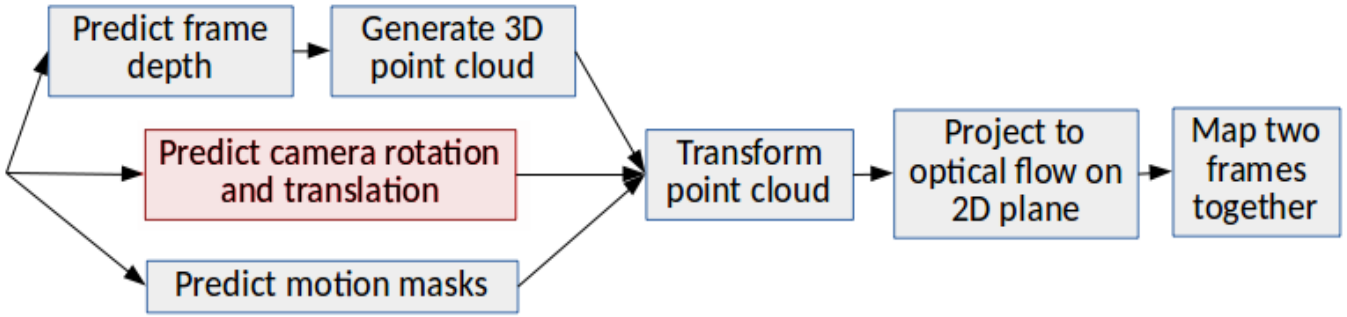
Backup

Depth from Mono



3-D Depth Reconstruction from a Single Still Image, Saxena et al., 2007

Approach in SfM-Net



$$R_t^{cx}(\alpha) = \begin{pmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

$$R_t^{cy}(\beta) = \begin{pmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{pmatrix},$$

$$R_t^{cz}(\gamma) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\gamma & -\sin\gamma \\ 0 & \sin\gamma & \cos\gamma \end{pmatrix},$$

Calculating camera motion

Let $\{R_t^c, t_t^c\} \in SE3$ denote the 3D rotation and translation of the camera from frame I_t to frame I_{t+1} (*relative* camera pose across consecutive frames). We represent R_t^c using an Euler angle representation as $R_t^{cx}(\alpha)R_t^{cy}(\beta)R_t^{cz}(\gamma)$ where

$$R_t^{cx}(\alpha) = \begin{pmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

$$R_t^{cy}(\beta) = \begin{pmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{pmatrix},$$

$$R_t^{cz}(\gamma) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\gamma & -\sin\gamma \\ 0 & \sin\gamma & \cos\gamma \end{pmatrix},$$

and α, β, γ are the angles of rotation about the x, y, z-axes respectively. The fully-connected layers are used to predict translation parameters t^c, the pivot points of the camera rotation $p_c \in \mathbb{R}^3$ as in [5], and $\sin\alpha, \sin\beta, \sin\gamma$. These last three parameters are constrained

to be in the interval $[-1, 1]$ by using RELU activation and the minimum function.

Calculating spatial smoothness

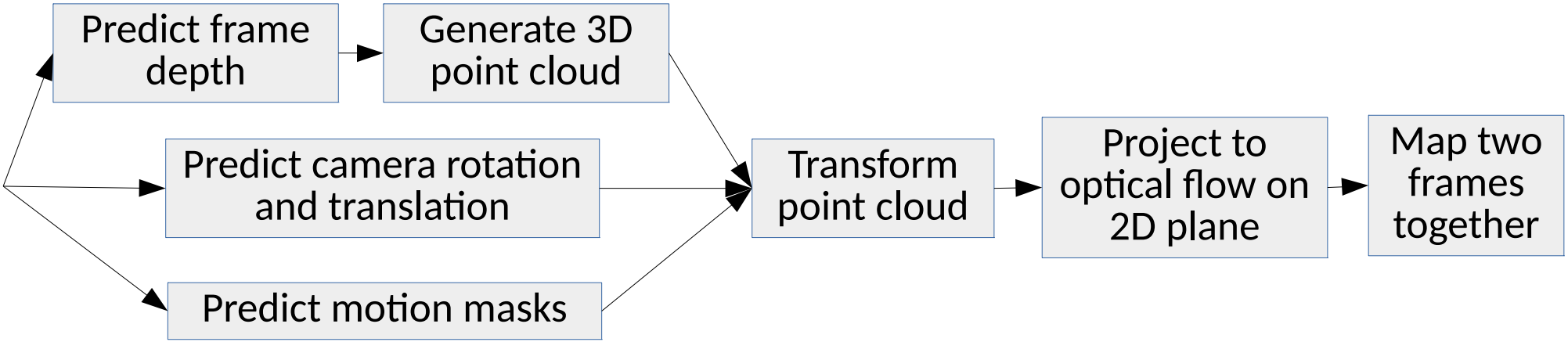
$$E_{\text{spat}}(D_t) = \sum_{\mathbf{x}} s_t^x(\mathbf{x})\rho(\nabla_x D_t(\mathbf{x})) + s_t^y(\mathbf{x})\rho(\nabla_y D_t(\mathbf{x}))$$

The weighting functions sxt and syt control the smoothness of the estimated depth map. It allows higher smoothing influence where contours do not arise in the image, so that the discontinuities are kept where the contours arise.

(from ‘Intrinsic Depth: Improving Depth Transfer with Intrinsic Images’ by Naejin Kong and Michael J. Black)

Approach in SfM-Net

Workflow:



Approach in SfM-Net

Workflow: