

Joint Optical Flow and Temporally Consistent Semantic Segmentation

Junhwa Hur, Stefan Roth

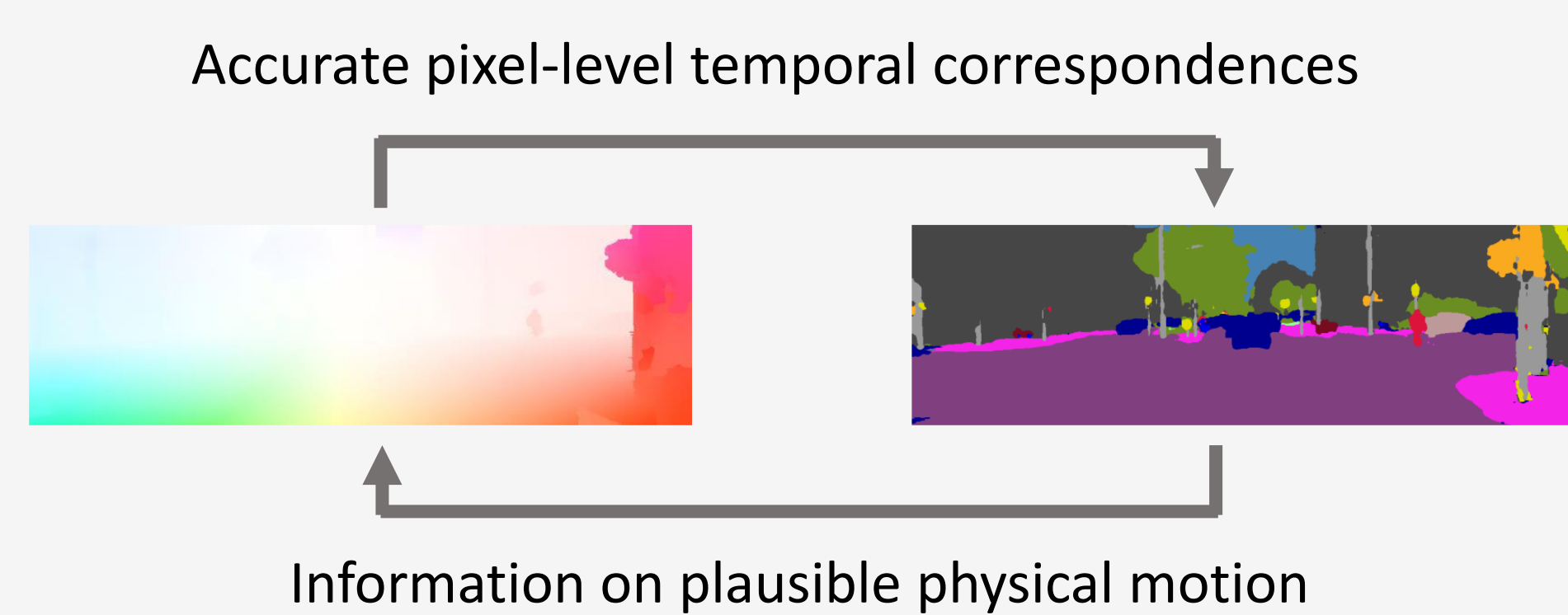
Department of Computer Science, TU Darmstadt



TECHNISCHE
UNIVERSITÄT
DARMSTADT

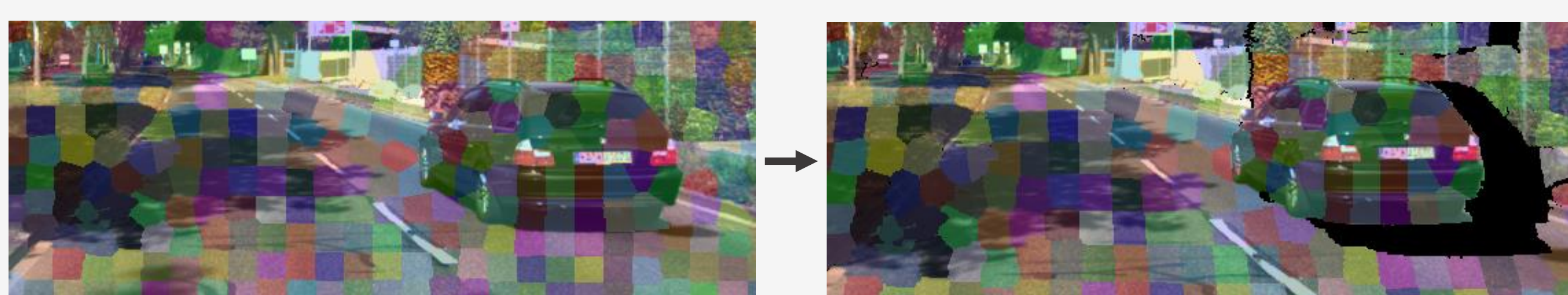
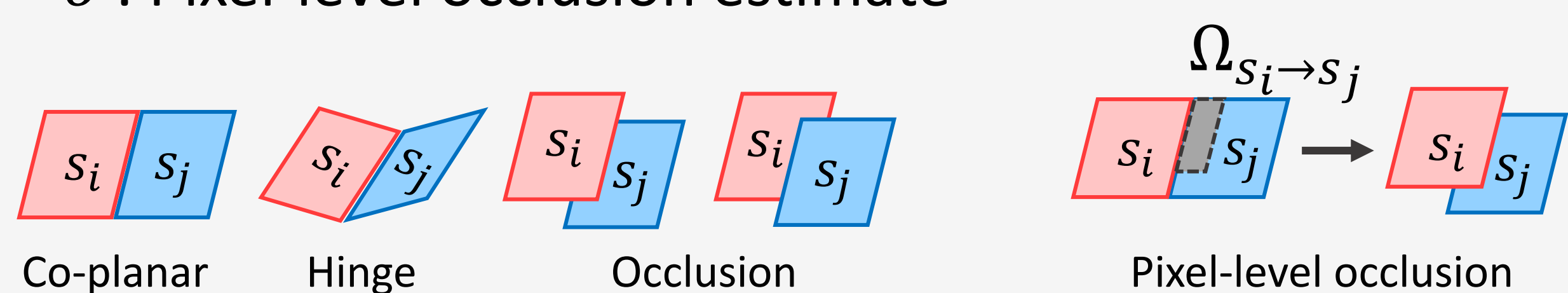
Introduction

- **Semantic segmentation and dense motion estimation** are two core components of visual scene understanding
- **Previous attempts** to bridge these two topics:
 - Using optical flow to enforce temporal consistency of semantic segmentation in a video sequence
 - Using both semantic information and segmentation to increase the accuracy of optical flow
- **Our objective:**
 - Jointly estimate optical flow and temporally consistent semantic segmentation
 - Closely connect these two problem domains and leveraging each other



Method

- **Piecewise rigid optical flow model (H, b, o)**
 - Superpixel-based formulation
 - $H_s \in \mathcal{R}^{3 \times 3}$: Parameterized homography motion
 - b : Occlusion boundary state between superpixels
 - o : Pixel-level occlusion estimate



- **Embedding semantic information (l)**
 - l : Labeling in subsequent frame (label constancy)
 - Applying the epipolar constraint (from the camera motion) on pixels of physically static objects

- **Approximate energy minimization with PMBP**

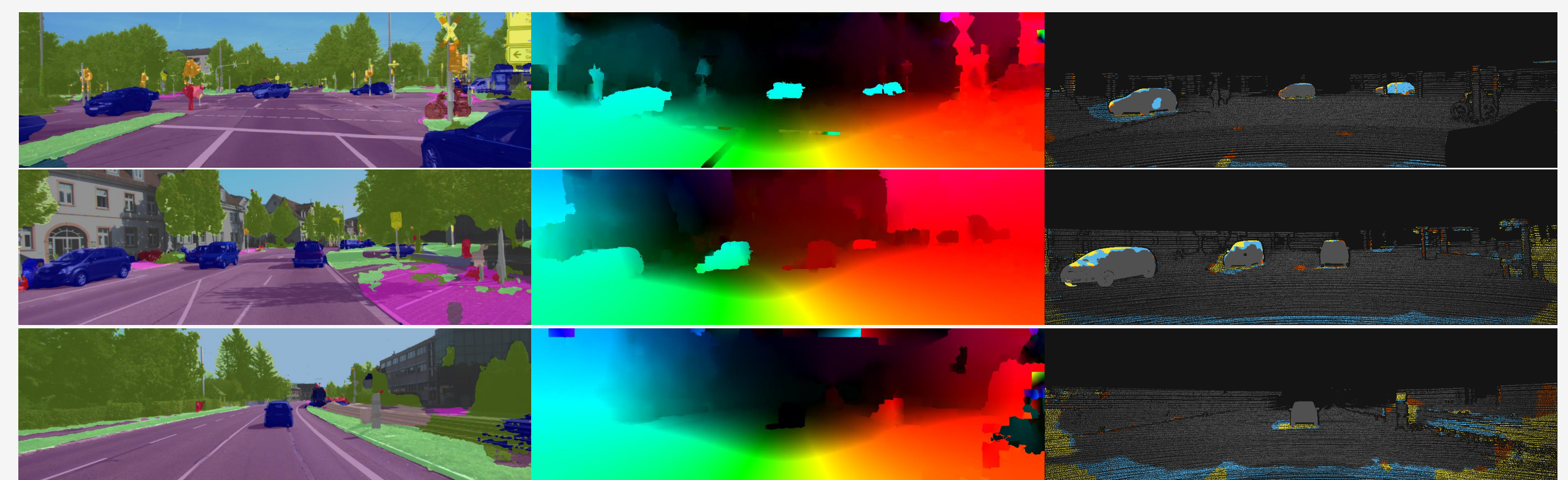
$$\begin{aligned}
 E(H, b, o, l) &= E_D(H, o) && \text{Data term: Truncated ternary census} \\
 &+ E_L(H, o, l) && \text{Label term: Label constancy} \\
 &+ E_P(H) && \text{Epipolar term: Label-dependent constraint} \\
 &+ E_C(H, b, o) && \text{Pairwise term: Motion smoothing} \\
 &+ E_B(b) && \text{Boundary term: Boundary relationship}
 \end{aligned}$$

- Boundary-label-dependent pairwise term

$$\frac{1}{|s_i \cup s_j|} \sum_{p \in s_i \cup s_j} \|H_{s_i} p - H_{s_j} p\|_1 + \frac{1}{|B_{s_i, s_j}|} \sum_{p \in B_{s_i, s_j}} \|H_{s_i} p - H_{s_j} p\|_1 + \sum_{p \in s_i \cup s_j} \{\lambda_{imp}[p \in \Omega_{s_i \rightarrow s_j}][o_p = 0] + \lambda_{imp}[p \notin \Omega_{s_i \rightarrow s_j}][o_p = 1]\}$$

Experiments

- Using bottom-up semantic segmentation from FCN [1] trained on the Cityscapes dataset [2]
- Using DiscreteFlow [3] to initialize estimation
- **Optical flow estimation results**



Bottom-up semantic evidence Optical flow results Comparison with DiscreteFlow

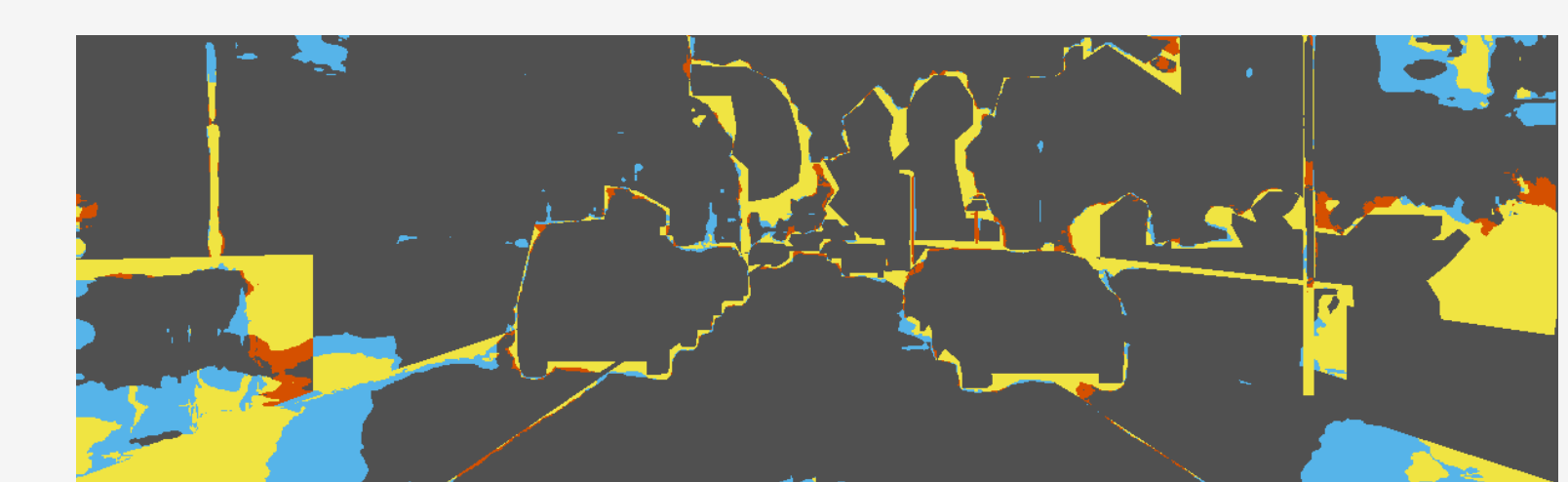
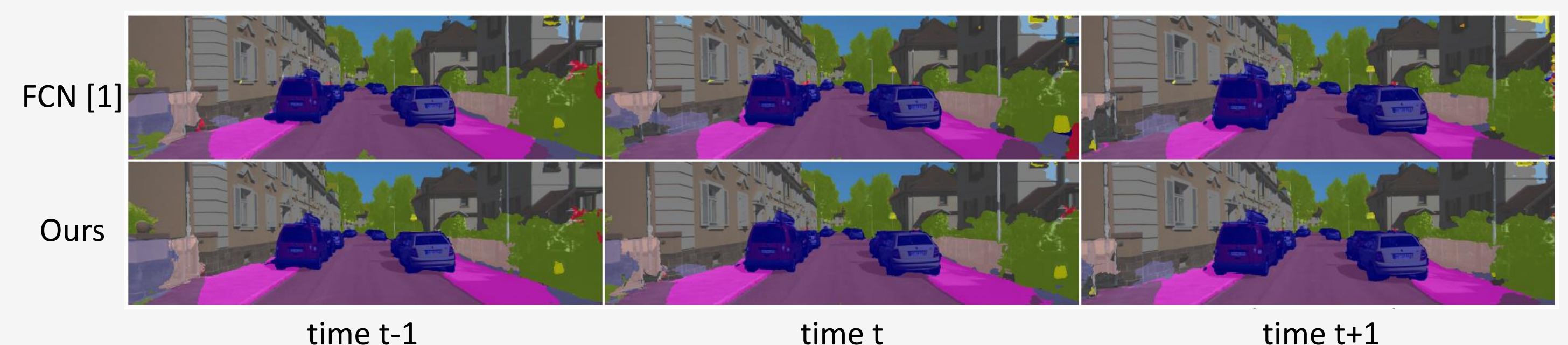
Flow Error	Non-occluded pixels			All pixels		
	Fl-bg	Fl-fg	Fl-all	Fl-bg	Fl-fg	Fl-all
MotionSLIC [4]	6.19 %	64.82 %	16.83 %	14.86 %	66.21 %	23.40 %
PatchBatch [5]	10.06 %	26.21 %	12.99 %	19.98 %	30.24 %	21.69 %
DiscreteFlow [3]	9.96 %	22.17 %	12.18 %	21.53 %	26.68 %	22.38 %
SOF [6]	8.11 %	23.28 %	10.86 %	14.63 %	27.73 %	16.81 %
Ours	7.85 %	18.66 %	9.81 %	15.90 %	22.92 %	17.07 %

KITTI Optical Flow 2015

Usage of terms		Non-occluded pixels			All pixels		
Label	Epipolar	Fl-bg	Fl-fg	Fl-all	Fl-bg	Fl-fg	Fl-all
Yes	Yes	8.27 %	17.40 %	9.83 %	16.44 %	20.02 %	16.98 %
Yes	No	8.45 %	16.97 %	9.90 %	16.73 %	19.61 %	17.17 %
No	Yes	8.20 %	17.82 %	9.84 %	16.35 %	20.41 %	16.99 %
No	No	8.51 %	17.21 %	10.00 %	16.84 %	19.86 %	17.31 %

Effectiveness of semantic-related terms (KITTI flow 2015 training dataset)

- **Temporally consistent semantic segmentation results**



Performance gain/loss over bottom-up semantic segmentation

IoU (%)	sky	building	road	sidewalk	fence	vegetation	pole	car	sign	pedestrian	cyclist	mean
FCN [1]	69.35	78.53	73.75	38.19	33.33	68.37	23.68	77.60	31.27	20.11	21.42	48.69
Ours	71.80	79.97	77.99	41.01	36.27	69.21	16.44	78.58	39.05	23.50	25.44	50.84

Temporally consistent semantic segmentation results on a sequence from the KITTI dataset

Conclusions

- Jointly estimating optical flow and temporally consistent semantic segmentation can successfully leverage each other
- A piecewise optical flow model with PMBP inference builds the basis and itself already achieves competitive results
- Embedding semantic information through label consistency and epipolar constraints further boosts the performance
- We achieve **state-of-the-art** optical flow results, and outperform all published algorithms by a large margin on challenging, but crucial dynamic objects