TransFlow: Unsupervised Motion Flow by Joint Geometric and Pixel-level Estimation

Stefano Alletto*, Davide Abati, Simone Calderara, Rita Cucchiara
University of Modena and Reggio Emilia
Via P. Vivarelli 10, Modena, Italy
name.surname@unimore.it
Luca Rigazio*
Panasonic Silicon Valley Laboratory
10900 North Tantau Avenue, Suite 200, Cupertino, CA, USA
name.surname@us.panasonic.com

Abstract

We address unsupervised optical flow estimation for ego-centric motion. We argue that optical flow can be cast as a geometrical warping between two successive video frames and devise a deep architecture to estimate such transformation in two stages. First, a dense pixel-level flow is computed with a geometric prior imposing strong spatial constraints. Such prior is typical of driving scenes, where the point of view is coherent with the vehicle motion. We show how such global transformation can be approximated with an homography and how spatial transformer layers can be employed to compute the flow field implied by such transformation. The second stage then refines the prediction feeding a second deeper network. A final reconstruction loss compares the warping of frame X_t with the subsequent frame X_{t+1} and guides both estimates. The model, which we named TransFlow, performs favorably compared to other unsupervised algorithms, and shows better generalization compared to supervised methods with a 3x reduction in error on unseen data.

1 Introduction

In the last few years, we assisted to a growing interest from the computer vision and machine learning community towards the autonomous and assisted driving fields. In this context, optical flow estimation represents one of the most active research fields but, despite the efforts, is still an open problem. Indeed, optical flow estimation is particularly challenging in automotive scenarios due to large displacements, strong changes in lighting conditions and the predominance of car-centric motion often masking individual objects' motion patterns.
Furthermore, whereas deep network architectures push the state-of-the-art forward, they typically require a large amount of labeled examples; this represents a significant issue in general, and is even more critical in the automotive field, where pixel-level annotated datasets lack. Indeed, public datasets either lack precise optical flow ground truth [8], or have very few images [11]. Indeed, creating a labeled dataset for optical flow is a complex task that often requires dedicated hardware; for example, ground truth flow maps in the popular KITTI Flow benchmark [11] are computed by means of 2D-3D matching of point clouds acquired by a LIDAR sensor. Alternatively, computer graphics approaches are employed such as in [10] to obtain large datasets at the expense of photorealism. It is clear how, in such a scenario, unsupervised models could benefit from much larger video datasets featuring a car perspective, such as the raw KITTI sequences, or the more recent Cityscapes [8] and DR(eye)VE [2].

* Authors contributed equally

Submitted to: 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

Figure 1: Examples of motion flows obtained through the TransFlow network

While an unsupervised method seems appealing due to the fact that more training data is available, learning how to compute a high-dimensional transformation such as the optical flow without specific guidance can be hard. In fact, many recent unsupervised approaches to motion flow estimation either struggle in achieving competitive performance [23] or perform far from real-time [16].
In this paper, we estimate the optical flow as a pixel-level transformation between successive frames. Building on the recently proposed Spatial Transformer layer (ST) [14], we develop an architecture that learns, given a pair of frames X_t and X_{t+1}, the parameters ϕ of the transformation T_ϕ from X_t to X_{t+1}. Our model jointly leverages the representation power of modern deep networks and geometrical transformations in two steps: First, a shallow network provides an estimate of global ego-motion. Intuitively, this first module estimates a global image transformation and requires a significantly lower number of parameters compared to the generic, pixel-level flow between two frames. In our model, this transformation is estimated as an homography, which flow field is simply computed from the ST itself. Intuitively, this is possible because of the ego-centric perspective of the driving scenario. At a later stage, this prediction is refined to get a fine-grained optical flow. The reason for such choice is that bootstrapping a global coarse estimation and then refining it is a much easier task compared to learning the complete transformation from scratch. Moreover, the global transformation module introduces some beneficial spatial and geometrical constraints that are often overlooked by recent deep models, e.g. the lack of local geometrical consistency that convolutional architectures exhibit, especially in presence of large receptive fields. Finally, to get smooth flow vectors within object boundaries, a fully differentiable bilateral filtering layer is employed.

The paper is structured as follows: The next section discusses recent works on the topics of deep and unsupervised flow estimation. Sec. 3 discusses our approach for casting the unsupervised motion flow computation as a deep image transformation problem. Sec. 4 proposes a set of experiments that validate our proposal providing a quantitative and qualitative evaluation. Eventually, conclustions and further development strategies are discussed in Sec. 5.

2 Related work

Since its early days, optical flow estimation has been addressed from an image processing perspective, mainly adopting strategies that rely on the brightness constancy principle [12]. Models that solely rely on pixel brightness and motion smoothness majorly suffer of a high sensitivity to outliers, that often occur due to changes in illumination condition and large displacements. To this end, Brox *et al.* [5] propose a variational approach that deals with the shortcomings of previous methods by jointly accounting for brightness constancy and brightness gradient constancy, as long as introducing piecewise smoothness. On a different note, Chen *et al.* [7] approach large displacement optical flow estimation by first computing a nearest-neighbor field for each pixel, which is shown to coarsely approximate the flow transformation. Subsequently, the prediction is refined by estimating and successively transforming dominant motion patterns.
More recently, deep learning based models have outperformed traditional approaches on public benchmarks. In their seminal work, *FlowNet*, Fischer *et al.* [9] introduce one of the first end-to-end deep architectures for dense optical flow. Using a convolutional-deconvolutional autoencoder, they address the lack of large datasets by creating a synthetic image collection featuring random chairs flying over random landscapes. Authors show the surprising generalization capability of the proposed

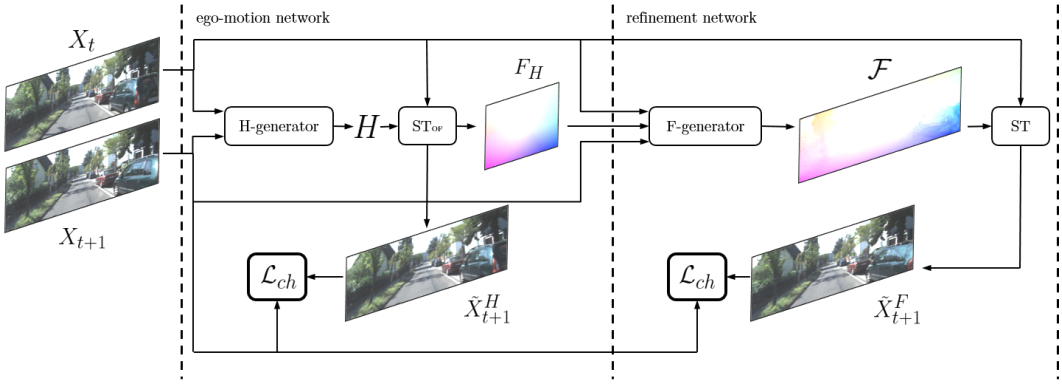


Figure 2: Architecture of the proposed model. Frames X_t and X_{t+1} are fed into a first module that estimates an homography matrix. A successive ST warps X_t according to the estimated projective transformation, resulting in the frame $\tilde{X}^H_{t+1}$. The global flow is then fed to a second network along with input frames, and its refinement $\mathcal{F}$ is used again to obtain a second reconstruction, $\tilde{X}^F_{t+1}$. Both reconstructions are guided by a Charbonnier Loss $\mathcal{L}_{ch}$ to approximate X_{t+1}. At test time, the output of the F-generator, $\mathcal{F}$, is the predicted optical flow.

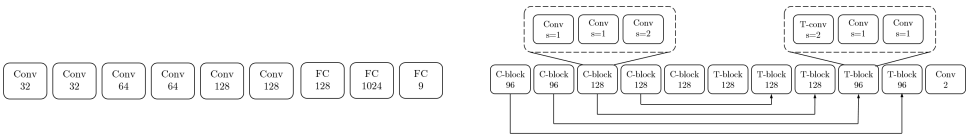
model when inference is performed over real-world sequences from a different domain. Building on their insights, a plethora of methods relying on deep neural networks to approach the problem have been proposed [4, 3, 19, 21, 22, 17, 13].

Among others, several works address flow estimation from an unsupervised perspective. Long *et al.* [16] reformulate the problem as image interpolation and matching. Training an autoencoder to learn the interpolation between two frames – i.e. to obtain image X_t from X_{t-1} and X_{t+1} – they show how performing per-pixel backpropagation at test time results into sensitivity maps from where pixel level matches can be obtained. However this method suffers of high computational cost: despite relying on modern GPU architectures, authors show how obtaining the flow for an image pair requires n/s backpropagations, where n is the total number of image pixels and s is an arbitrary stride controlling the sparseness of the resulting map. In [23] another unsupervised model is illustrated, employing a similar autoencoder but embedding into its loss function brightness constancy and motion smoothness constraints, first introduced in [12]. Surprisingly, this method achieves state of the art performance on the KITTI dataset when compared to other unsupervised approaches, showing the benefits of leveraging the representation power of deep learning and traditional image approaches. A disadvantage of many of the aforementioned approaches is their algorithmic complexity. In fact, many rely on a set of different techniques bundled together such as as object segmentation, background-foreground separation, Markov random field inference or smoothing. Conversely, our approach employs one end-to-end trainable network that limits the problem complexity by forcing the flow to resemble a geometric warping between consecutive frames while still preserving the capability of a deep architecture to refine coarse predictions by exploiting deep features and transformer layers.

3 Optical flow as a pixel-level image transformation

Estimating a dense flow between two adjacent frames X_t, X_{t+1} is an inherently difficult problem. In fact, for each pixel in X_t, the two components u, v of the flow that form a transformation between X_t and X_{t+1} have to be computed, resulting in a transformation with $2 \times h \times w$ parameters, where h, w represent the height and width of the two frames. While traditional supervised methods estimate this transformation directly by guiding the optimization using ground-truth information, here we propose an unsupervised model that can be applied when such a ground-truth guidance is not possible.

Our initial model computes a simple perspective trasformation of the X_t frame (e.g. a global, rigid geometrical transformation) that is used for bootstrapping the computation of the dense, $2 \times h \times w$ motion flow using a deep convolutional model, Fig. 2. We constrained the initial transformation to be geometrically consistent by using a ST. The main insight is that in scenarios such as autonomous



(a) (b)

Figure 3: (a) H-generator network. All convolutional layers have stride $s = 2$ and all layers feature leaky ReLU activations, except for the top linearly activated fully connected module. (b) F-generator network. Layers in the same block share the number of output features. All activations are leaky ReLU. Convolutional layers feature a 3x3 kernel (both in H-generator and F-generator), whereas transposed convolutions feature 4x4 kernel. Best viewed on screen and zoomed in.

driving there are two main types of motion: the motion of the car (often referred to as *ego-motion*) and different motion patterns for moving objects[1].

3.1 Flow computation as an unsupervised reconstruction task

Before introducing the two components of our architecture, we describe how estimating the motion flow with a neural network architecture in an unsupervised fashion can be seen as an image transformation problem.

Given a generic transformation T_ϕ between two adjacent frames X_t, X_{t+1} obtained as the output of a deep model, we use a ST to compute $T(X_t)$. The ST module, being fully differentiable, allows us to backpropagate the gradient of the loss function to the module that regresses the geometric transformation parameters e.g. the deep model itself. For optical flow computation the estimated T_ϕ are the optical flow vectors while $\tilde{X}_{t+1} = T_\phi(X_t)$ results in the estimate of frame X_{t+1}, Fig. 2. By applying such an architecture it is possible to constrain every flow estimation model to learn the best transformation disregarding the optical flow ground truths by optimizing a Charbonnier reconstruction penalty:

$$\mathcal{L}_{ch} = \sqrt{(\tilde{X}_{t+1} - X_{t+1})^2 + \epsilon}, \tag{1}$$

where ϵ is a small constant (fixed to 0.1 in our experiments). The Charbonnier penalty, a differentiable version of the l_1 norm, penalizes the deviation of the prediction from the ground truth sub-band residuals. Notice how, regardless of the choice of the architecture and subsequently of the resulting transformation, the training of such network can be performed with no direct supervision other than the frame X_{t+1} itself.

3.2 Ego-motion estimation

As the initial step of the TransFlow net, a shallow architecture is used to estimate the initial geometric warping H approximating the ego-motion of a car. H is contrained to be a perspective homographic transformation that acts on the whole image plane and generates, through a modified ST (ST_{OF} in Fig. 2), a coarse flow model. The projective transformation H is particularly suitable since it has a limited number of parameters ($H \in \mathbb{R}^9$) and effectively allows us to employ the ST_{OF} module to warp X_t into X_{t+1}.

Fig. 3.2 (a) reports the details of the architecture of the H-generator network. Notice how the network itself is small sized, featuring 6 convolutional layers and 3 fully connected layers resulting in $600k$ parameters overall. The H-generator processes the two frames stacked on the channel dimension and outputs the 9 parameters of the transformation, which are fed to an instance of the ST that produces the warped $\tilde{X}^H_{t+1}$. Furthermore, the modified ST module, ST_{OF} outputs the dense flow F_H between its input and output, *i.e.* the motion vector of each pixel warped from X_t to X^H_{t+1}. More precisely, given a uniform sampling grid G holding columnwise pixel homogeneous coordinates, the ST applies the H transfomation to obtained the warped grid $\tilde{G} = H \cdot G$. Since $\tilde{G}$ holds the warped coordinates of each pixel, the flow field can easily be computed as $F_H = \tilde{G} - G$. All the operation on G in ST_{OF} are differentiable thus it is possible to seamlessly backpropagate through the layer.

[1]Please note that landscape and static objects are subjected to the ego-motion component.

3.3 Flow refinement

The H-generator produces a dense and yet global flow which captures the overall motion of the car but lacks the details of individual objects. To obtain a fine-grained flow of the scene, we employ a second, deeper network inspired by the model in [9]. Fig 1 (b) reports the details of the architecture: the encoder is composed of five convolutional blocks, where each of them includes three 3×3 convolutional layers with leaky ReLU activations. All convolutions in a block have stride 1 except for the last one, which has stride 2. The transposed-convolution blocks mirror the architecture, and a top 2-channel convolutional layer with tanh activation function produces the final flow in the range $[-1, 1]$.

During training, the F-generator processes a channel-wise concatenation of X_t, X_{t+1} and F_H, and outputs a dense flow map $\mathcal{F} \in \mathbb{R}^{2\times w\times h}$. Hence, the top ST produces $\tilde{X}^F_{t+1}$ estimate of X_{t+1} and, by backpropagating the loss of Eq. 1, the F-generator network learns to improve the flow accounting for moving objects and fine details neglected by the ego-motion network, while keeping the flow geometrically consistent.

3.4 Edge aware smoothing

The two components of our flow described in Sec. 3.2 and 3.3 are over-smooth and over-sharp respectively. We choose to add an edge aware filtering layer for making inference. Edge aware smoothing has been demonstrated to be beneficial in optical flow estimation, as it allows to get uniform flows within object boundaries, which often correspond to motion boundaries [18].
In particular, we rely on the lattice-based implementation of high dimensional gaussian filters by Adams *et al.* [1]. Calling $\{z_i\}_{i=1}^N \subset \mathbb{R}^2$ the set of pixel flow components and $\{f_i\}_{i=1}^N \subset \mathbb{R}^d$ the set of pixel features in a d-dimensional space, the output of a high dimensional gaussian filter is given by:

$$\tilde{z}_i = \frac{\sum_{j=1}^N z_j K(f_i, f_j)}{\sum_{j=1}^N K(f_i, f_j)}, \quad \text{with} \quad K(f_i, f_j) = e^{-(||f_i - f_j||^2)} \tag{2}$$

Depending on the set of features employed, different types of filters can be achieved. Here, we rely on the bilateral filter, in which $f_i = (\frac{x_i}{\sigma_s}, \frac{y_i}{\sigma_s}, \frac{r_i}{\sigma_c}, \frac{g_i}{\sigma_c}, \frac{b_i}{\sigma_c})$ where x_i and y_i represent the pixel's location within the image, r_i, g_i and b_i encode the color of the pixel in the target image and σ_s and σ_c are hyperparameters tuning the sparseness of spatial and color features. This way, the flow components get smoothed using the target image as guidance.

Since we aim at keeping our solution fully differentiable and end-to-end trainable, this kind of filter is particularly suitable as $\delta C/\delta \tilde{z}$ can be propagated to the filter input z by filtering the gradient itself with the same feature image. We refer the reader to [24] for further details.

4 Experimental results

4.1 Datasets and training

Due to the unsupervised nature of our method, we do not require ground truth flow information to train our network. For this reason, we are able to employ large-scale automotive datasets such as KITTI raw [11] and DR(eye)VE [2]. In particular, the KITTI raw dataset includes 44,000 frames acquired in the city of Karlsruhe, while DR(eye)VE features 555,000 frames including steep changes in image conditions due to transitions between day and night, sun, rain and a significant variance in scenarios such as highway, downtown or countryside. Notice how the size of the aforementioned datasets is suitable for training a neural network; on the other hand, the biggest real-world automotive database including ground truth flow information nowadays is the KITTI Flow 2012[11], that combines less that 400 annotated pairs split between training and testing. Recently, a synthetic automotive dataset inspired by KITTI has been released [10]. The Virtual Kitti dataset features more than 21,000 frames fully annotated with optical flow, semantic segmentation, depth and object bounding boxes. Due to its recent release, a state of the art on the Virtual Kitti dataset is not yet established, nonetheless we include it in our evaluation to show the generalization capabilities of our method when challenged with different automotive scenarios.

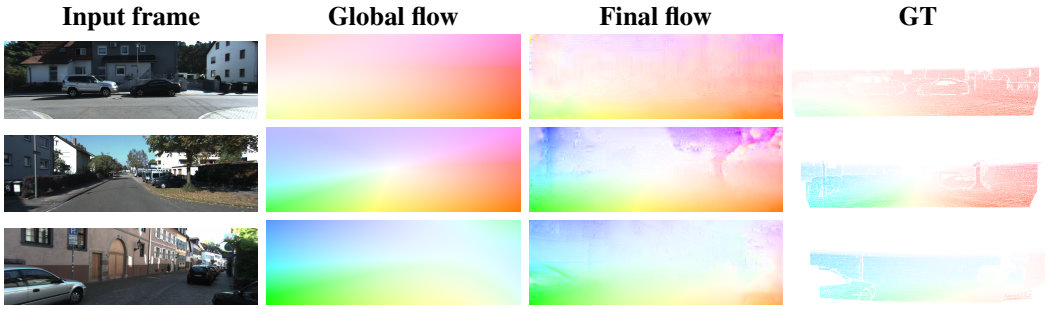


Figure 4: Qualitative results on the Kitti 2012 dataset.

Table 1: Performance comparison on the KITTI Flow 2012 dataset, non occluded pixels (NOC). Note that not all the methods report results on the public leaderboards (testing set). The table is divided in three sections: hand-crafted and supervised methods, unsupervised, our proposal. Execution times are reported on the testing set.

Method	**Training**		**Testing**		
	Acc@5	**APE**	**Acc@5**	**APE**	**Time (s)**
HoG	0.455	9.68	-	-	-
KLT	0.702	8.16	-	-	-
FlowNetS [9]	-	-	0.630	5.0	0.08
DeepFlow [22]	-	-	0.927	1.5	17.0
EpicFlow [17]	-	-	0.912	1.5	15.0
Long *et al.* [16]	0.716	4.70	-	-	486
Yu *et al.* [23]	-	4.30	0.652	4.60	0.03
TransFlow	0.857	3.335	0.692	3.90	0.14
TransFlow+Bilat	0.866	3.132	0.705	3.60	0.15

To train our network, we build a set of image pairs sampled from the KITTI raw dataset; we also report experiments where the network has been finetuned on the different datasets employed, namely DR(eye)VE, KITTI 2012 and Virtual Kitti. We train the network using the Adam optimizer [15] with β_1 set to 0.5 and minibatch size of 16, 1000 batches per epoch. The training is stopped after 250 epochs. The loss function of Eq. 1 is employed during the training by weighting the errors of the H-generator and F-generator as follows:

$$\mathcal{L}_{ch} = \mathcal{L}_{ch}(\tilde{X}^H_{t+1}, X_{t+1}) \times \alpha + \mathcal{L}_{ch}(\tilde{X}^F_{t+1}, X_{t+1}) \times \beta \tag{3}$$

where α and β are set to 0.5 and 1 respectively. The source code is released and available on the github page of the project[2].

4.2 Evaluation: Kitti 2012

Following standard flow evaluation benchmarks, to evaluate the performance of our method we adopt the following metrics: Accuracy@5, meaning the ratio of motion vectors with end point error lower than 5 pixels, and APE which is the average point error of all motion vectors.

First, we evaluate our algorithm against two popular hand-crafted methods, KLT [20] and HoG [6]. We also include in our evaluation two recent deep-learning based approaches computing the flow in an unsupervised fashion, namely [16, 23]. Finally, we report the results of TransFlow in two variants, namely with and without the bilateral refinement described in Sec. 3.

Table 1 reports the results of this evaluation. In particular, notice how TransFlow compares favorably against both hand-crafted methods and recent unsupervised approaches. Compared to recent supervised approaches, TransFlow shows competitive performance and while not reaching the average

[2]https://github.com/stefanoalletto/TransFlow

Table 2: Performance comparison on the Virtual Kitti dataset

Method	Acc@5	APE
DeepFlow	0.450	20.179
FlowNet2	0.420	20.883
EpicFlow	0.458	17.948
TransFlow	0.745	7.627
TransFlow+FT	0.777	6.770

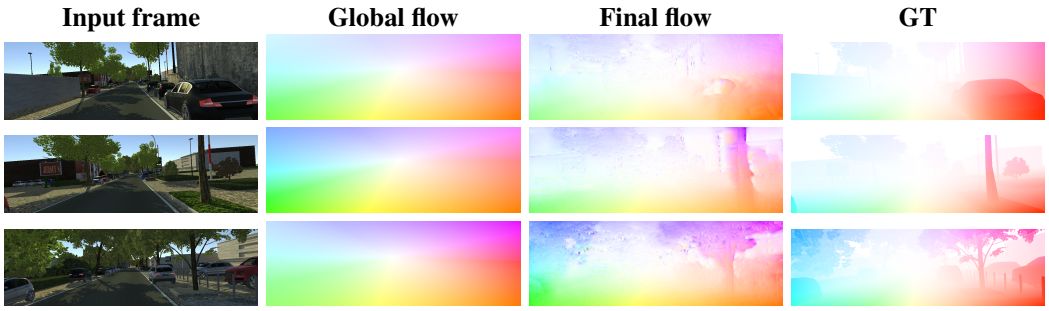


Figure 5: Qualitative assessment of our the proposed model on the Virtual Kitti dataset.

point error results of DeepFlow [22] or EpicFlow [17] , is two orders of magnitude faster, requiring $0.15sec/pair$ compared to the $17sec/pair$ and $15sec/pair$ of DeepFlow and EpicFlow respectively.

4.3 Evaluation: Virtual Kitti

To further evaluate our method, we experiment on the Virtual Kitti dataset. While still featuring the typical automotive perspective, Virtual Kitti presents some noteworthy differences from the other datasets, the most significant being the presence of the typical artifacts of computer rendered scenes. Nonetheless, it is currently the biggest dataset providing ground truth optical flow for automotive, and is hence worth evaluating the aforementioned methods performance on it. Due to the dataset being very recent, no results are publicly available and we evaluate DeepFlow, EpicFlow and FlowNetv2 relying on the provided source codes and pretrained models. In all the experiments, the methods have been used with their default parameters, and no finetunig has been performed on the Virtual Kitti dataset (TransFlow reported results are obtained with the model trained on the Kitti raw sequences only). Table 2 provides the results of this evaluation. In particular, notice how all three methods perform rather poorly on this new dataset, mainly due to the synthetic nature of the rendered scenes that significantly differs from what the methods have been trained on, showing a lack of generality. Arguably, supervised method in this context would likely require a finetuning step on this new dataset in order to adapt the learned feature to the differences that are due to the computer graphics generated sequences. On the other hand, notice how TransFlow results in a significantly lower average point error. This behavior can be attributed to the fact that, aiming at learning an image transformation instead of trying to approximate a ground truth flow, TransFlow learned features have better generalization capabilities. Further confirmation of this derives from the fact that finetuning the network on the Virtual Kitti dataset (in Table 2 indicated by the row TransFlow+FT), while still improving the results, does not significantly alter the performance of the method. Figure 4.2 displays a qualitative evaluation of the results on the Virtual Kitti dataset.

Table 3: Performance comparison on the DR(eye)ve dataset. The evaluation is performed on the *Downtown* sequences of the dataset.

Method	Avg.	Night	Rain	Day
DeepFlow	48.10	36.53	52.93	54.84
FlowNet2	48.72	35.56	54.03	56.56
EpicFlow	49.77	36.65	55.60	57.05
TransFlow	4.38	2.71	4.91	5.52

4.4 Evaluation: DR(eye)VE

To evaluate the proposed approach on the DR(eye)VE dataset, which lacks ground truth flow information, we rely on our initial assumption that the optical flow can be computed as the transformation warping two consecutive frames. It is hence possible to quantitatively evaluate the flow performance by estimating the average reconstruction error (ARE) of the second frame. That is, given a frame X_t and the optical flow transformation T_ϕ, we use our spatial transformer layer to obtain the reconstructed frame $\tilde{X}^F_{t+1}$ and compute the ARE as $\frac{|\tilde{X}^F_{t+1} - X_{t+1}|}{N}$ where N is the number of pixels in the image. Table 3 reports the results of such evaluation where no finetuning has been performed on the DR(eye)VE dataset. Notice how, similarly to the Virtual Kitti scenario, TransFlow generalizes better than any of the supervised approaches, obtaining a significantly lower reconstruction error despite the lack of specific training. Figure 4.4 reports qualitative results of this analysis, showing how the evaluated methods tend to overestimate the motion flow on sequences where the average motion is fairly limited. In turn, this results in a flow field at the edges of the image with an excessive intensity, that leads the reconstructed pixel to end outside the frame. Explanation for this behavior is the overfitting of supervised methods on higher intensity flow sequences, while TransFlow is shown to be able to effectively estimate the correct amount of motion. To analyze the impact of different environmental conditions on the flow computation, Table 3 also reports results divided by sequence type. The slightly lower ARE across all methods during the Night sequences is due to the lower intensity of the images, effectively resulting in a lower error. Beside that, the experiment shows that environmental conditions do not have a significant impact on optical flow computation.

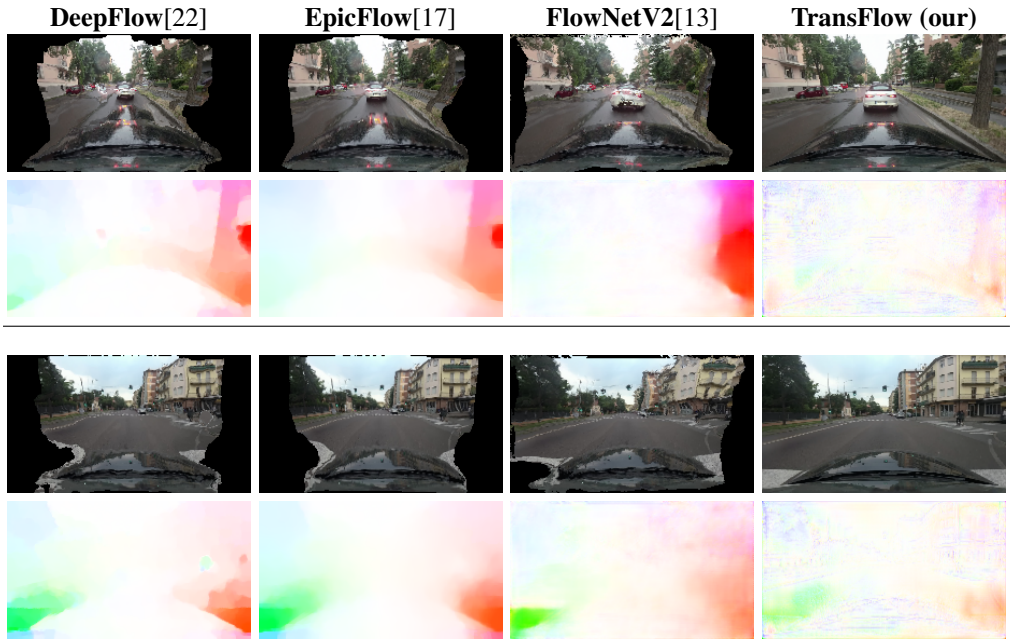


Figure 6: Qualitative assessment of our the proposed model on reconstructing the frame.

5 Conclusions

In this paper, we showed an unsupervised approach to optical flow estimation that jointly accounts for the geometric cues in a scene, and for the pixel-level motion patterns of different objects. In particular, we address the complexity of unsupervised training by first estimating the global motion of the car using a lightweight network that, in turn, serves as initialization for a more complex, dense, pixel-level transformation. Our experimental evaluation shows how the proposed approach outperforms recent unsupervised methods while maintaining the advantages in terms of simplicity and speed of an end-to-end forward only neural network. In fact, we strongly believe that, especially in motion flow estimation, obtaining ground truth information can be prohibitive and unsupervised models will in turn become a key component of any computer vision pipeline relying on flow information. In fact,

our evaluation shows that our method express significantly better generalization capabilities compared to supervised approaches, where switching dataset is possible without requiring a finetuning step.

References

[1] Andrew Adams, Jongmin Baek, and Myers Abraham Davis. Fast high-dimensional filtering using the permutohedral lattice. In *Computer Graphics Forum*, volume 29, pages 753–762. Wiley Online Library, 2010.

[2] Stefano Alletto, Andrea Palazzi, Francesco Solera, Simone Calderara, and Rita Cucchiara. Dr(eye)ve: a dataset for attention-based tasks with applications to autonomous and assisted driving. In *IEEE Internation Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2016.

[3] Min Bai, Wenjie Luo, Kaustav Kundu, and Raquel Urtasun. Exploiting semantic information and deep matching for optical flow. In *European Conference on Computer Vision*, pages 154–170. Springer, 2016.

[4] Christian Bailer, Kiran Varanasi, and Didier Stricker. Cnn-based patch matching for optical flow with thresholded hinge loss. *arXiv preprint arXiv:1607.08064*, 2016.

[5] Thomas Brox, Andrés Bruhn, Nils Papenberg, and Joachim Weickert. High accuracy optical flow estimation based on a theory for warping. In *European conference on computer vision*, pages 25–36. Springer, 2004.

[6] Thomas Brox and Jitendra Malik. Large displacement optical flow: descriptor matching in variational motion estimation. *IEEE transactions on pattern analysis and machine intelligence*, 33(3):500–513, 2011.

[7] Zhuoyuan Chen, Hailin Jin, Zhe Lin, Scott Cohen, and Ying Wu. Large displacement optical flow from nearest neighbor fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2443–2450, 2013.

[8] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[9] Philipp Fischer, Alexey Dosovitskiy, Eddy Ilg, Philip Häusser, Caner Hazırbaş, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. *arXiv preprint arXiv:1504.06852*, 2015.

[10] A Gaidon, Q Wang, Y Cabon, and E Vig. Virtual worlds as proxy for multi-object tracking analysis. In *CVPR*, 2016.

[11] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[12] Berthold KP Horn and Brian G Schunck. Determining optical flow. *Artificial intelligence*, 17(1-3):185–203, 1981.

[13] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. *arXiv preprint arXiv:1612.01925*, 2016.

[14] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Advances in Neural Information Processing Systems*, pages 2017–2025, 2015.

[15] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[16] Gucan Long, Laurent Kneip, Jose M Alvarez, Hongdong Li, Xiaohu Zhang, and Qifeng Yu. Learning image matching by simply watching video. In *European Conference on Computer Vision*, pages 434–450. Springer, 2016.

[17] Jerome Revaud, Philippe Weinzaepfel, Zaid Harchaoui, and Cordelia Schmid. EpicFlow: Edge-Preserving Interpolation of Correspondences for Optical Flow. In *Computer Vision and Pattern Recognition*, 2015.

[18] Jerome Revaud, Philippe Weinzaepfel, Zaid Harchaoui, and Cordelia Schmid. Epicflow: Edge-preserving interpolation of correspondences for optical flow. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1164–1172, 2015.

[19] Tal Schuster, Lior Wolf, and David Gadot. Optical flow requires multiple strategies (but only one network). *arXiv preprint arXiv:1611.05607*, 2016.

[20] Carlo Tomasi and Takeo Kanade. Detection and tracking of point features. 1991.

[21] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Deep end2end voxel2voxel prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 17–24, 2016.

[22] Philippe Weinzaepfel, Jerome Revaud, Zaid Harchaoui, and Cordelia Schmid. DeepFlow: Large displacement optical flow with deep matching. In *IEEE Intenational Conference on Computer Vision (ICCV)*, Sydney, Australia, December 2013.

[23] Jason J Yu, Adam W Harley, and Konstantinos G Derpanis. Back to basics: Unsupervised learning of optical flow via brightness constancy and motion smoothness. *arXiv preprint arXiv:1608.05842*, 2016.

[24] Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip HS Torr. Conditional random fields as recurrent neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1529–1537, 2015.