Efficient Sparse-to-Dense Optical Flow Estimation using a Learned Basis and Layers: Supplementary Material

Jonas Wulff Michael J. Black
Max Planck Institute for Intelligent Systems, Tübingen, Germany
{jonas.wulff,black}@tue.mpg.de

June 22, 2017

Contents

1 Parameter values

This section contains the parameter values we used for our experiments for both algorithms, PCA-Flow and PCA-Layers. All parameters were determined to mimimize the average endpoint error on the respective training sets.

1.1 PCA-Flow

Description	Symbol	Value	
		Sintel	KITTI
Noise estimation for robust function	σ	1.0	0.6
Regularization	λ	0.2	0.4

1.2 PCA-Layers

Description	Symbol	Value	
		Sintel	KITTI
Noise estimation for robust function	σ	0.1	0.3
Regularization	λ	0.002	0.3
Weight of color unary cost	γ_c	3.0	3.0
Weight of location unary cost	γ_l	9.0	40.0
Weight of pairwise cost	γ	450	250
Scaling in warping energy	σ_w	3.0	0.7
Scaling in feature distance cost	σ_l	15	

2 Experiment: Number of principal components

Fig. 1 shows the average endpoint error across the whole training set as a function of the number of principal components that were used. The projected ground truth is given in green, the estimation results using the learned basis is given in blue, and the estimation results using the DCT is given in red. Here we plot the results of `PCA-Flow`, i.e. only a single layer, since this is the part of our pipeline that is most directly affected by the maximum number of principal components.

Note that with very few principal components, the projected ground truth result is worse than the estimated results. The reason for this is that the projection minimizes the distance of the ground truth flow field to the projected ground truth field on the optical flow subspace. This does not necessarily minimize the average endpoint error.

While the results using the DCT basis are very close to the results using our learned basis, they are consistently slightly worse. Therefore, we prefer our basis.

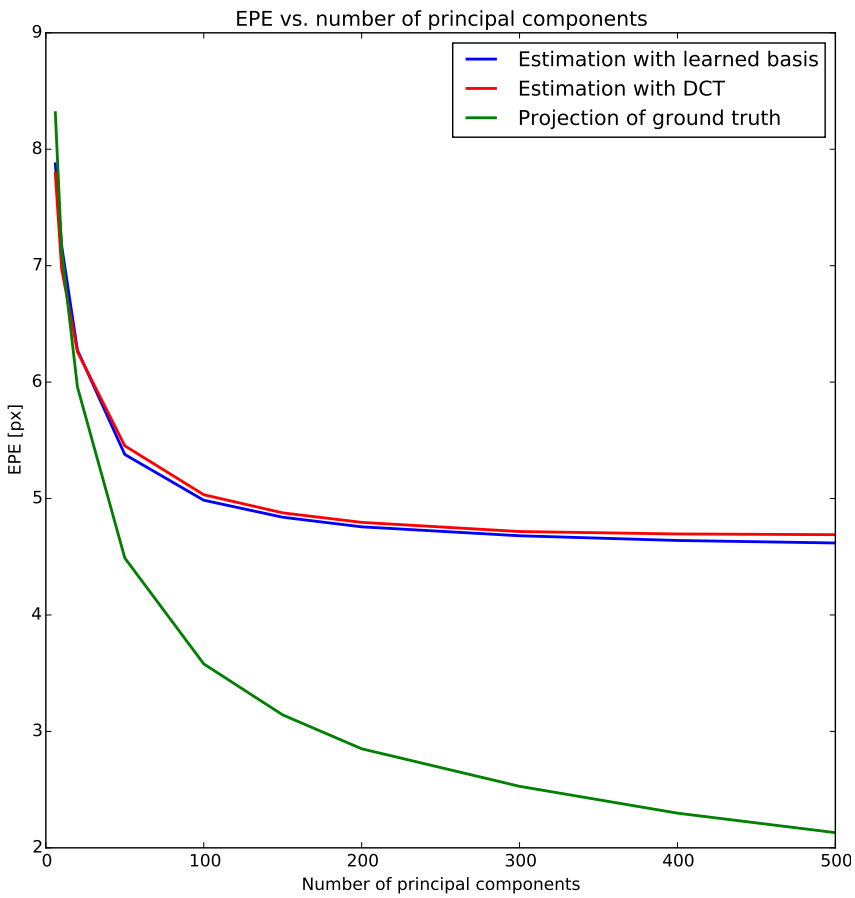


Figure 1: Average endpoint error as a function of the number of principal components

Fig. 2 shows the same results, split between the final pass (dotted) and the clean pass (dashed). Due to the more accurate feature matching, the results on the clean pass are much better. The shape of the curves, however, is not significantly different from Fig. 1.

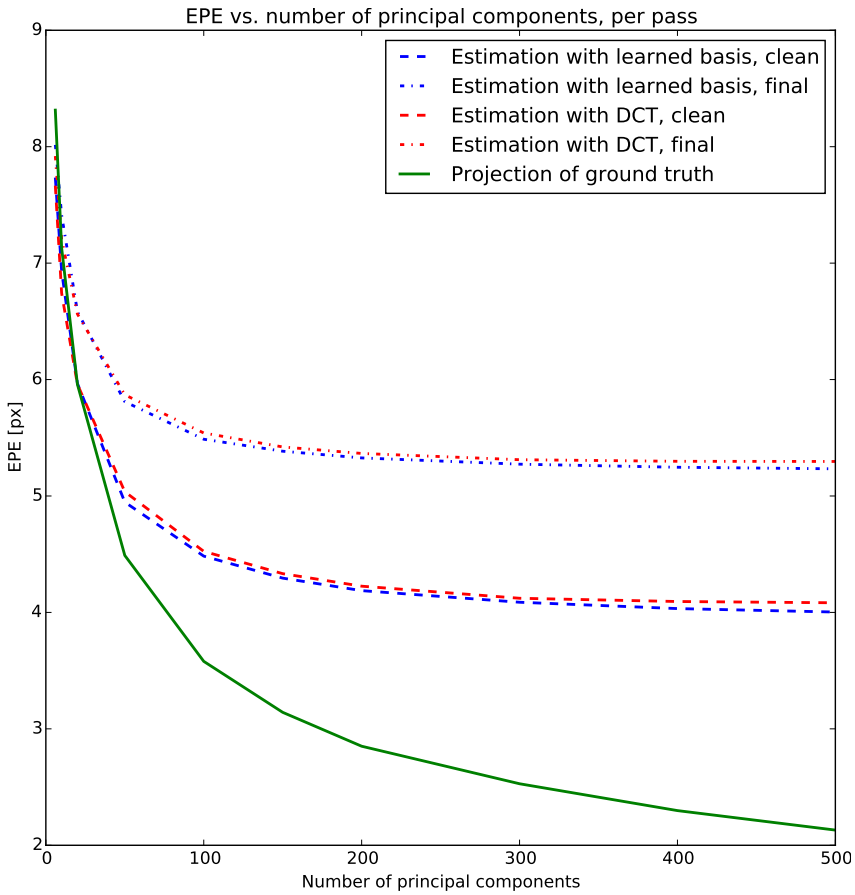


Figure 2: Average endpoint error as a function of the number of principal components, split by pass

3 Experiment: Feature quality

One important source of error are incorrect or insufficient feature matches. Here, we show two experiment analyzing this effect, first the influence of errors in the feature matching, and second the influence of insufficient (but potentially very good) feature matches.

3.1 Feature errors and endpoint error

Fig. 3 shows the average endpoint error within a frame, as computed using PCA-Flow, compared to the average error of all features founds in that frame.

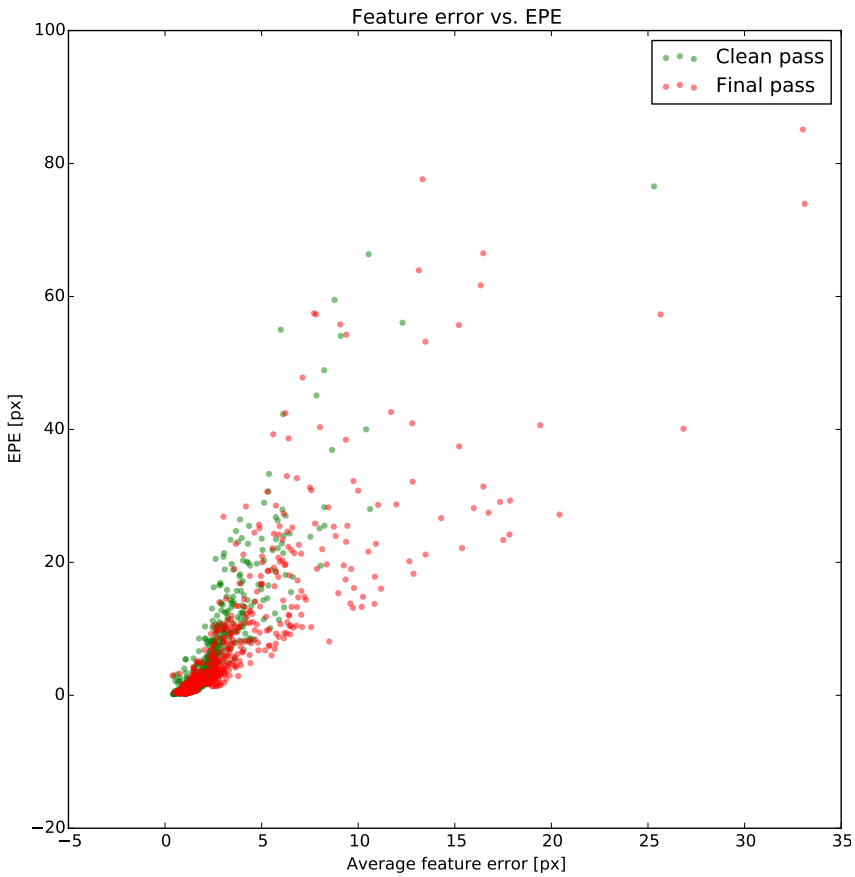


Figure 3: The feature error and the computed endpoint errors are very correlated. The feature errors are higher on the final pass.

3.2 Feature density and endpoint error

Fig. 4 and Fig. 5 shows the relationship between the number of features found in a single frame and the average endpoint error across this frame. Here, blue points correspond to the found feature matches. Yellow points correspond to the *ground truth matches* at the locations of the found matches. To obtain those ground truth matches, we first compute the matches, and then replace them with the ground truth flow at the detected feature locations.

The more features are found, the better the reconstruction generally is. Reversely, if only few features are found, they usually do not sufficiently cover the image, causing high errors. Additionally, the matching quality in

frames with lower feature density is also lower, since those frames do not contain enough structure everywhere to reliably match features.

An additional source of errors are motion and camera blur and atmospheric effects. The final pass (shown in Fig. 5) contains such artifacts, while the clean pass (Fig. 4) does not. Consequently, the feature match quality is generally lower (in addition to more frames with only very few available features), leading to a higher error rate (See Table 1).

	`Sintel-clean`	`Sintel-final`	Average
Error of estimated features	1.83 px	2.67 px	2.25 px
Error over full frame, using estimated features	4.00 px	5.23 px	4.62 px
Error over full frame, using ground truth features	3.20 px	3.60 px	3.40 px

Table 1: Errors on MPI-Sintel for estimated and ground truth features

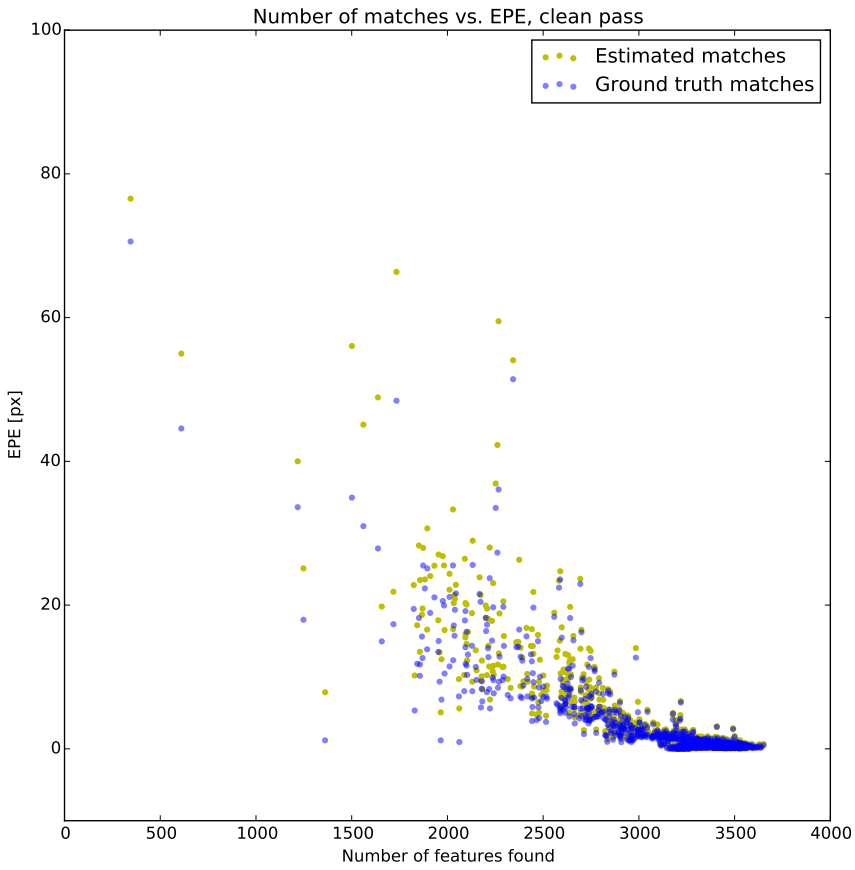


Figure 4: Clean pass

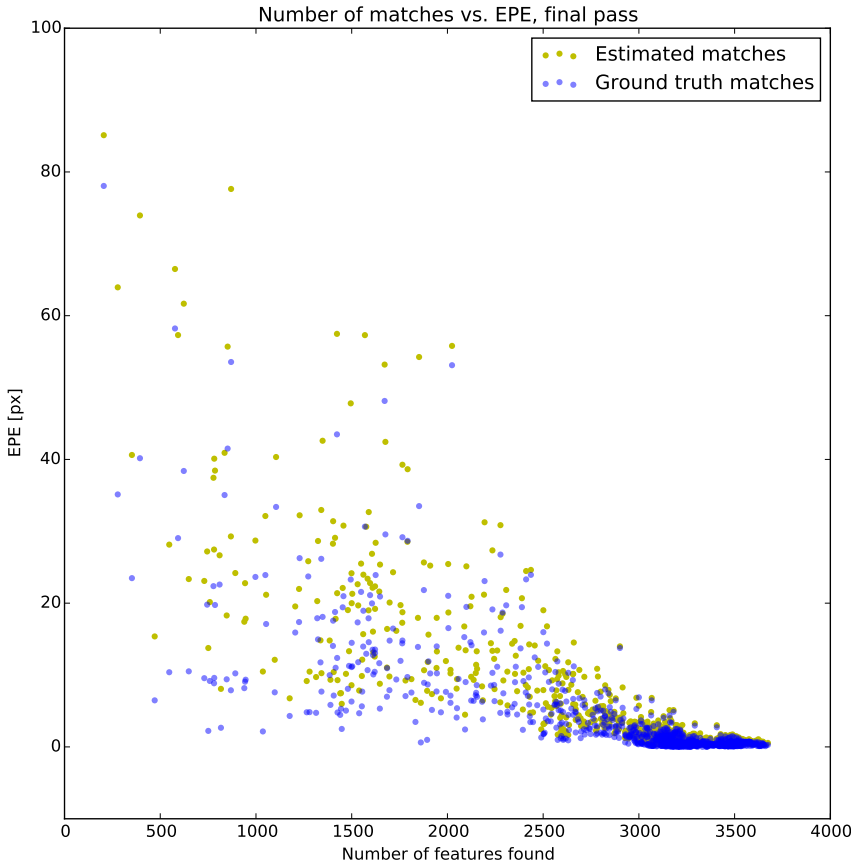


Figure 5: Final pass

4 Using a warping-based approach

As mentioned in the paper, it is also possible to use a warping-based approach to estimate the coefficients $\mathbf{w}$. Such an approach does not rely on feature matches, but instead iteratively rewarps the image to minimize the brightness constancy error [1]. It is commonly used in patch-based motion subspace methods, e.g. [2].

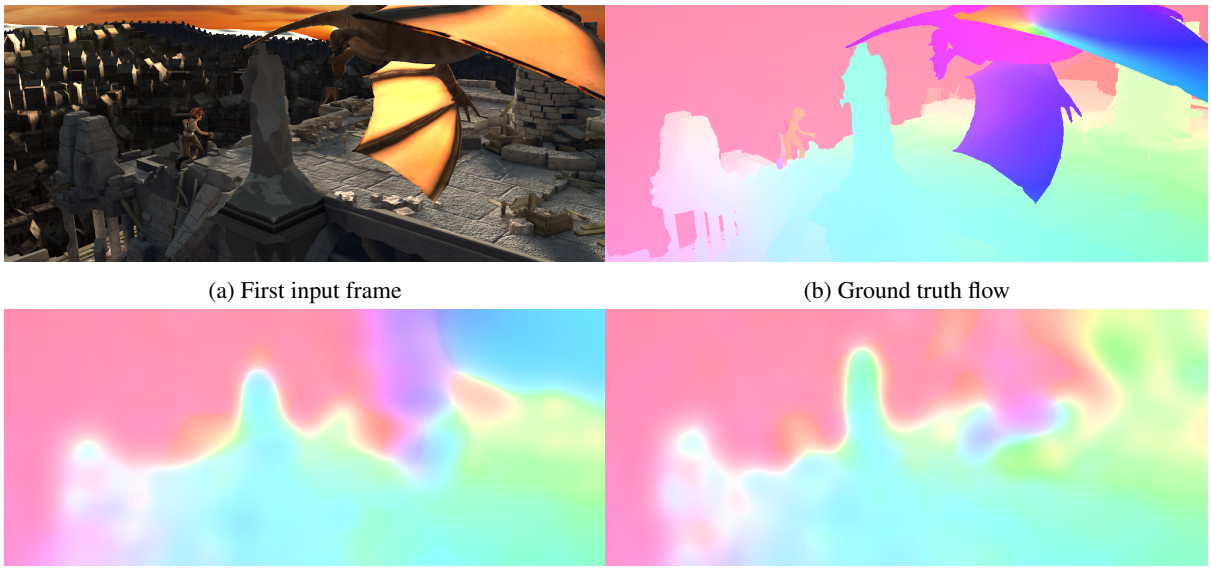
To be able to cope with large motions, a multiscale framework is usually used. Here, we use 7 pyramid levels, at a scale factor of 1.5 per pyramid level. Furthermore, the error term is robustified, and the same prior as in the feature-based approach is taken into account. We refer to this approach as `PCA-Warp`.

Table 2 shows the results for the feature-based `PCA-Flow` and the warping-based `PCA-Warp`. `PCA-Warp` results in significantly higher errors, mostly due to large motions. At the same time, it is much slower, and takes approximately 30 seconds per frame, as compared to 300 ms for `PCA-Flow`. An interesting observation is that when using `PCA-Warp`, the difference between the clean and the final pass is much smaller, since the increased difficulty of finding features in the final pass does not affect `PCA-Warp`.

	`Sintel-clean`	`Sintel-final`	Average
PCA-Flow	4.00 px	5.23 px	4.62 px
PCA-Warp	7.16 px	7.21 px	7.19 px

Table 2: Errors on MPI-Sintel for PCA-Flow and PCA-Warp.

The Figs. 6–11 show the *best* examples for `PCA-Warp` relative to `PCA-Flow`, that is, the frames for which the difference $\text{EPE}_{PCA-Warp} - \text{EPE}_{PCA-Flow}$ is lowest. We show three examples for the clean pass, and three for the final pass. The main advantages of `PCA-Warp` is that it does not rely on matched features, and hence is able to estimate motion in regions that are not sufficiently textured to extract feature points. Nevertheless, as shown in Table 2, the average error in the Sintel training set is significantly higher for `PCA-Warp` compared to `PCA-Flow`.



(a) First input frame (b) Ground truth flow

(c) PCA-Warp, EPE=13.6 (d) PCA-Flow, EPE=14.1

Figure 6: Clean pass, best result for `PCA-Warp` relative to `PCA-Flow`

(a) First input frame

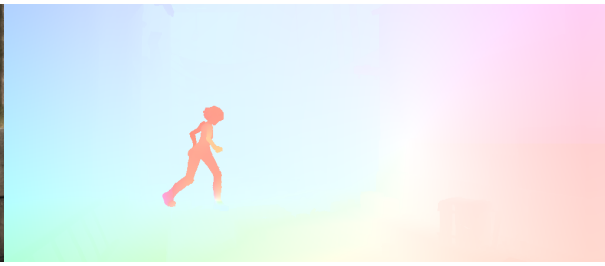
(b) Ground truth flow

(c) PCA-Warp, EPE=0.80

(d) PCA-Flow, EPE=1.03

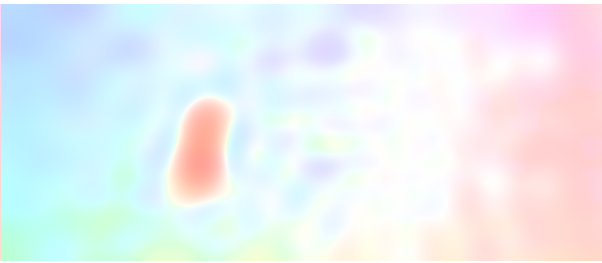
Figure 7: Clean pass, 2nd best result for `PCA-Warp` relative to `PCA-Flow`

(a) First input frame



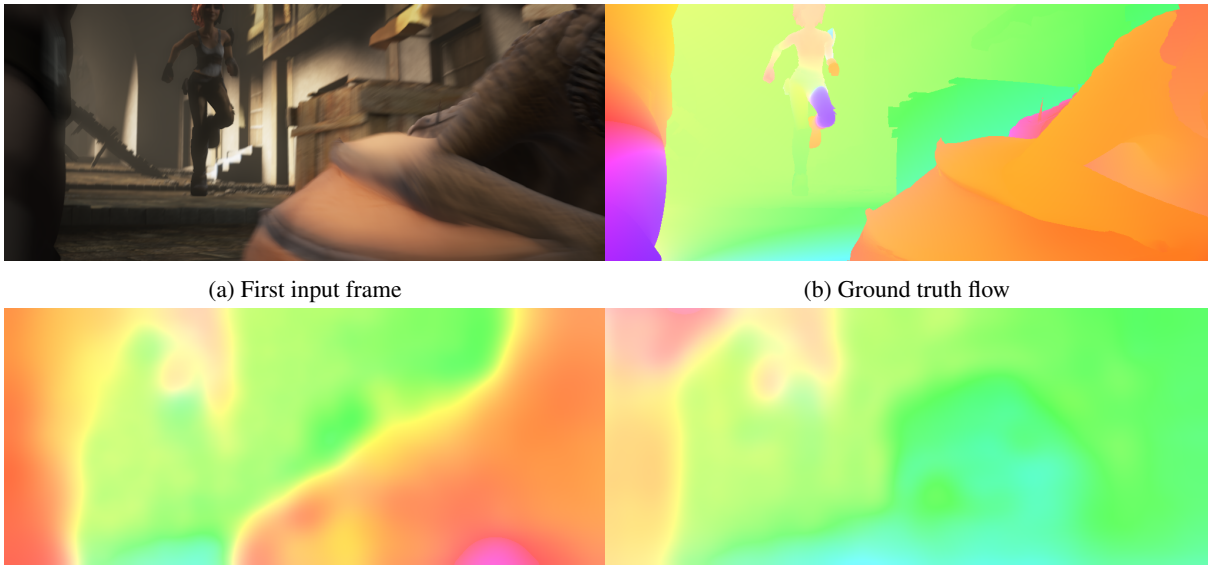
(b) Ground truth flow

(c) PCA-Warp, EPE=0.29



(d) PCA-Flow, EPE=0.49

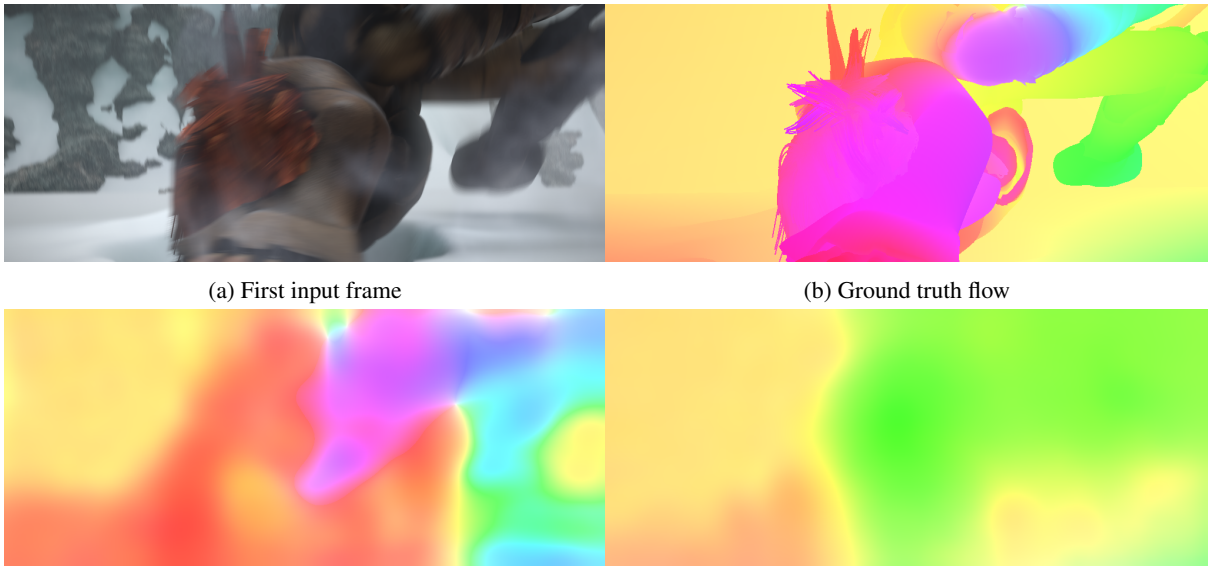
Figure 8: Clean pass, 3rd best result for `PCA-Warp` relative to `PCA-Flow`



(a) First input frame (b) Ground truth flow

(c) PCA-Warp, EPE=31.2 (d) PCA-Flow, EPE=49.3

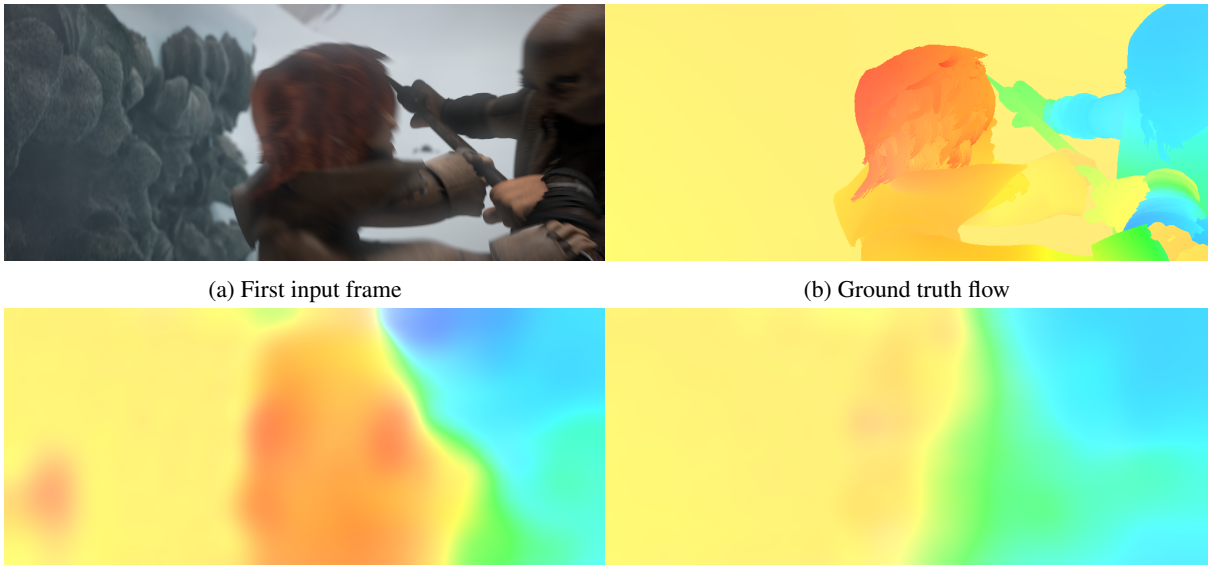
Figure 9: Final pass, best result for `PCA-Warp` relative to `PCA-Flow`



(a) First input frame (b) Ground truth flow

(c) PCA-Warp, EPE=35.5 (d) PCA-Flow, EPE=51.3

Figure 10: Final pass, 2nd best result for `PCA-Warp` relative to `PCA-Flow`



(a) First input frame

(b) Ground truth flow

(c) PCA-Warp, EPE=18.9

(d) PCA-Flow, EPE=26.0

Figure 11: Final pass, 3rd best result for `PCA-Warp` relative to `PCA-Flow`

5 Visual results

In the following section, we show additional results from the training sets of MPI-Sintel (both Clean and Final passes) and KITTI. For each example, we show:

- (a) The first of the two input frames.
- (b) The ground truth optical flow.
- (c) The model assignment at each pixel. This can be seen as a coarse motion segmentation.
- (d) The estimated optical flow.
- (e) The homography, robustly fitted to all matched features.
- (f) The result of PCA-Flow, added as an additional motion proposal.
- (g)-(l) The individual motion models computed by our hard EM algorithm; each model also shows which tracked feature point contributes to it. If too few features are assigned to a given model, it is removed from the estimation, and not shown here.

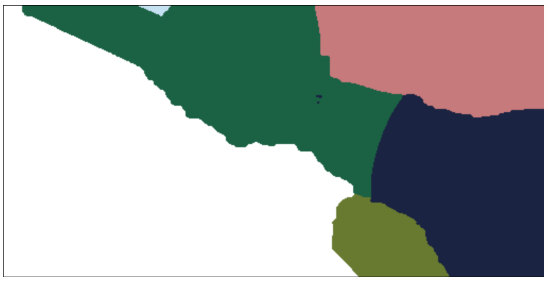
Note that all images were scaled to 512×256 pixel, since this is the resolution that our algorithms use internally.

5.1 Sintel clean

The following pages show examples from the "Clean" pass of MPI-Sintel.

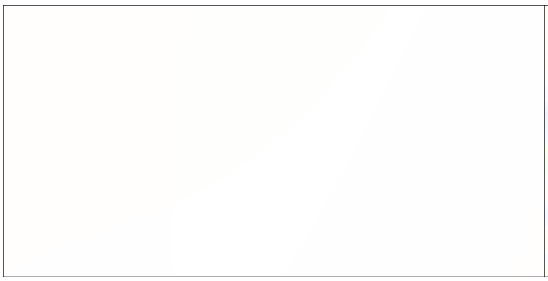
(a) Frame

(b) Ground truth flow

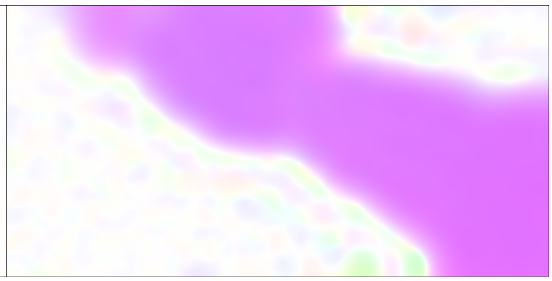


(c) Computed segmentation

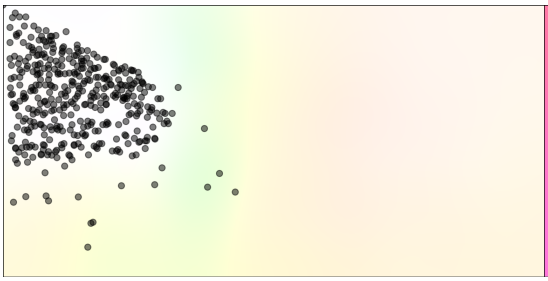
(d) Computed optical flow, EPE = 0.82



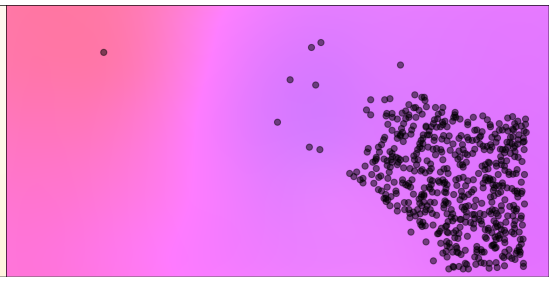
(e) Homography



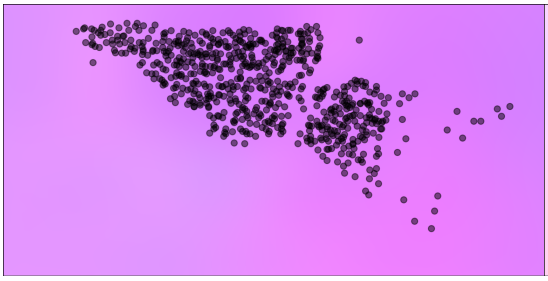
(f) PCA-Flow, EPE = 2.48



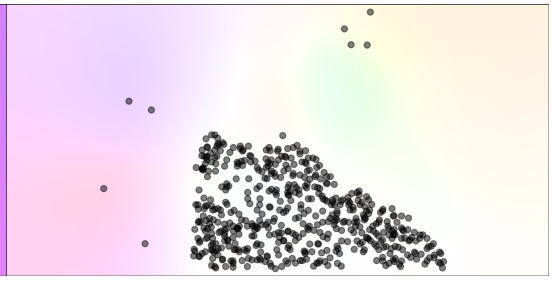
(g) Model 1



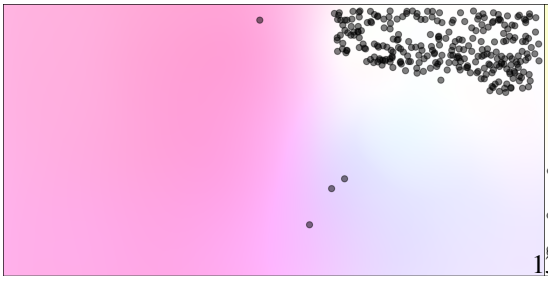
(h) Model 2



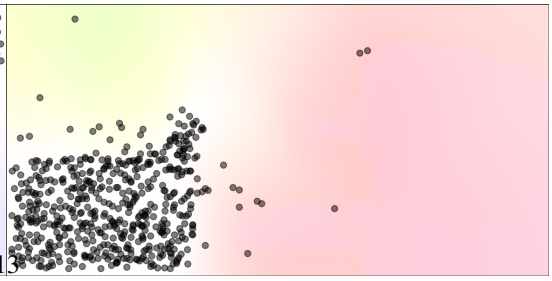
(i) Model 3



(j) Model 4



(k) Model 5



(l) Model 6

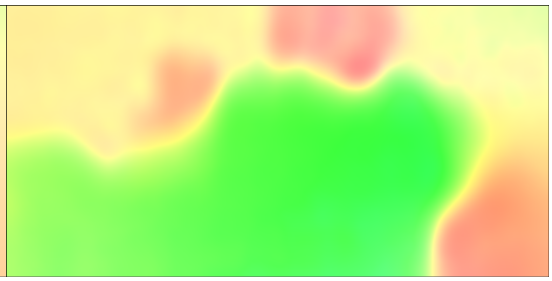
(a) Frame

(b) Ground truth flow

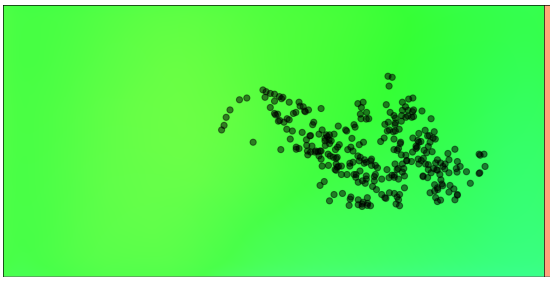
(c) Computed segmentation

(d) Computed optical flow, EPE = 5.30

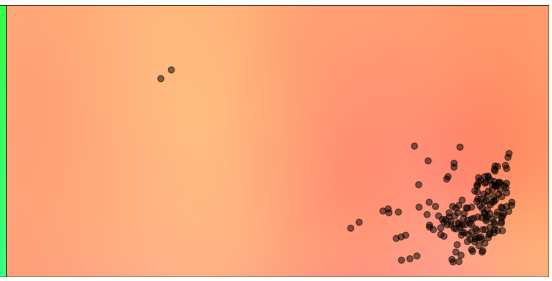
(e) Homography



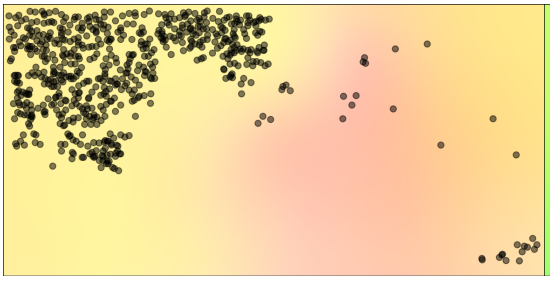
(f) PCA-Flow, EPE = 28.0



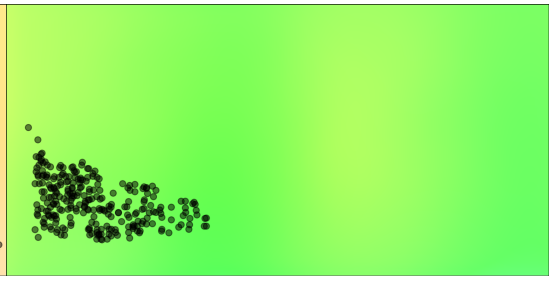
(g) Model 1



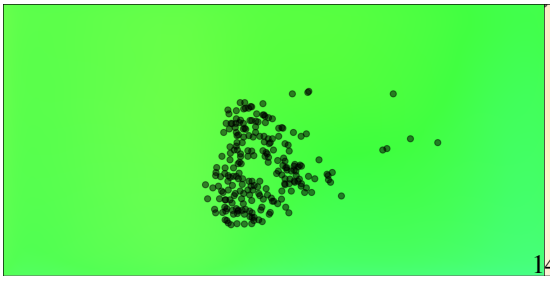
(h) Model 2



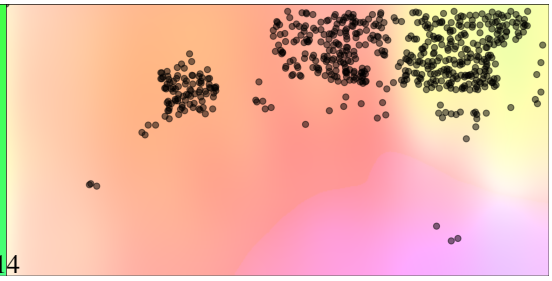
(i) Model 3



(j) Model 4



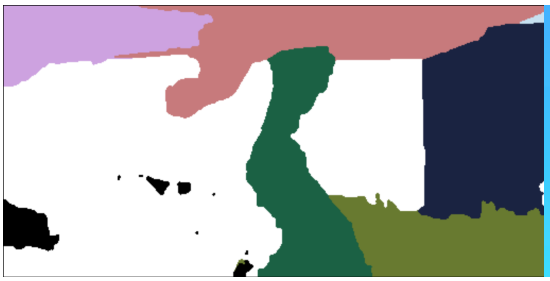
(k) Model 5



(l) Model 6

(a) Frame

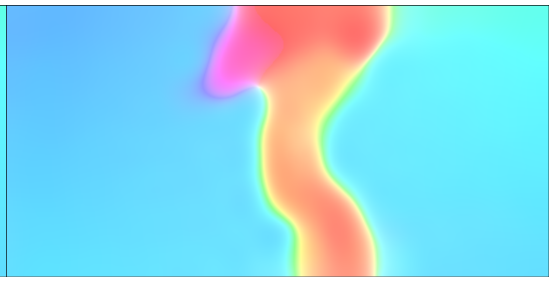
(b) Ground truth flow



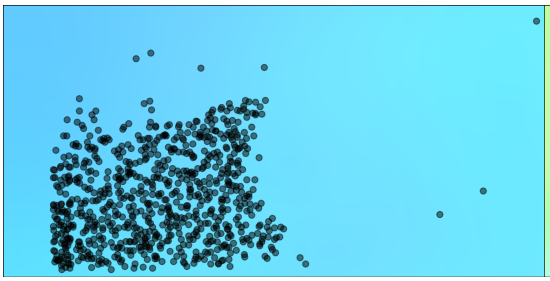
(c) Computed segmentation

(d) Computed optical flow, EPE = 7.04

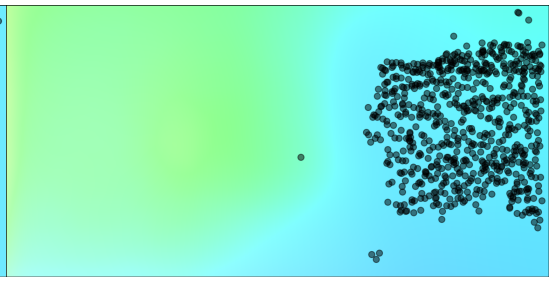
(e) Homography



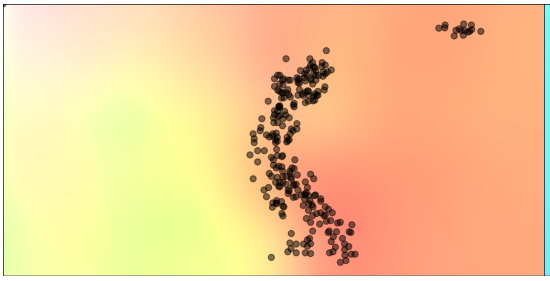
(f) PCA-Flow, EPE = 16.5



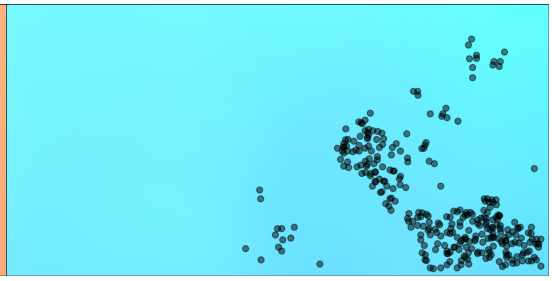
(g) Model 1



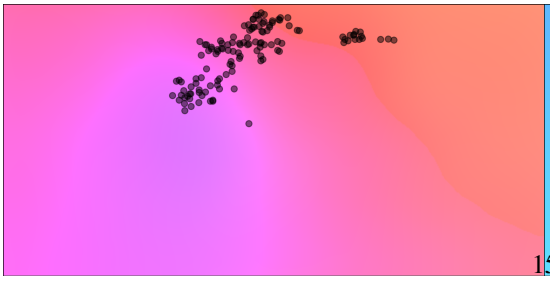
(h) Model 2



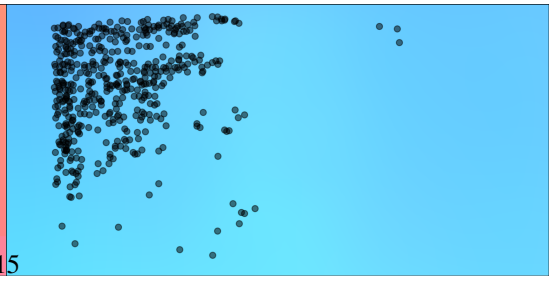
(i) Model 3



(j) Model 4



(k) Model 5



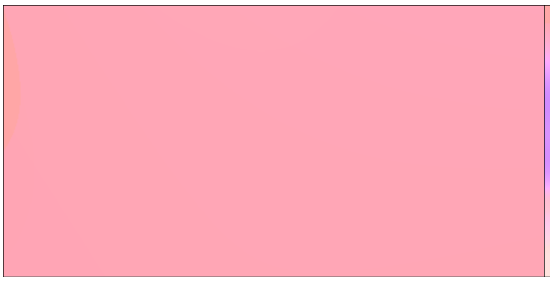
(l) Model 6

(a) Frame

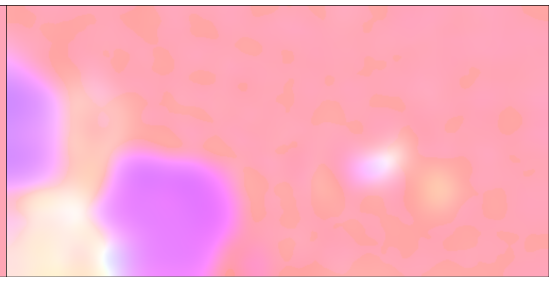
(b) Ground truth flow

(c) Computed segmentation

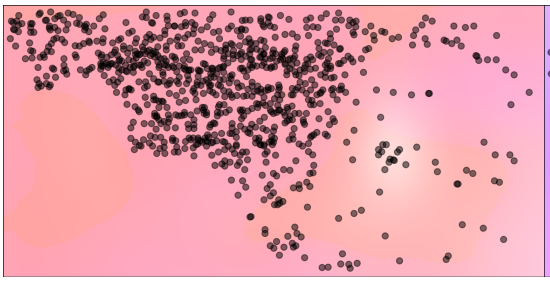
(d) Computed optical flow, EPE = 1.33



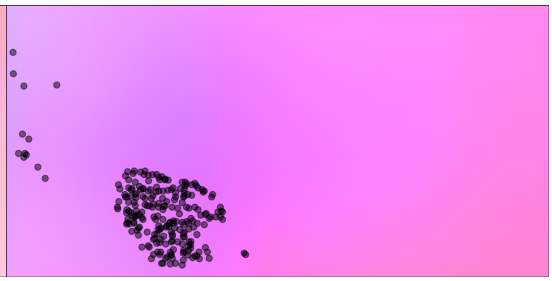
(e) Homography



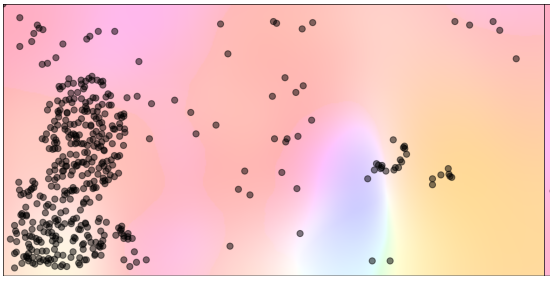
(f) PCA-Flow, EPE = 2.10



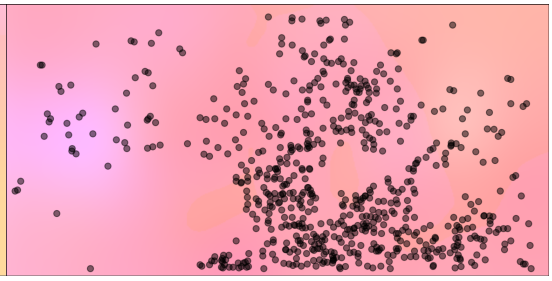
(g) Model 1



(h) Model 2

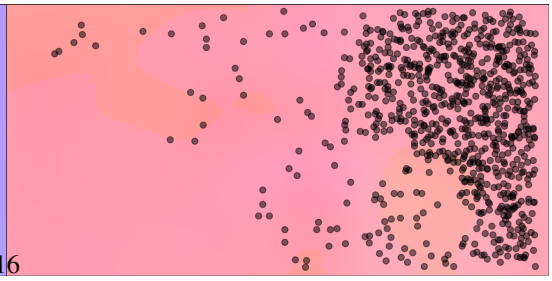


(i) Model 3



(j) Model 4

(k) Model 5



(l) Model 6

5.2 Sintel final

The following pages show examples from the “Final” pass of MPI-Sintel.

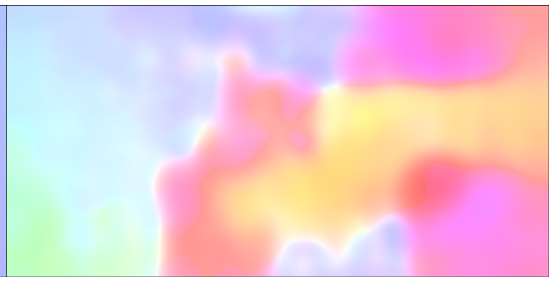
(a) Frame

(b) Ground truth flow

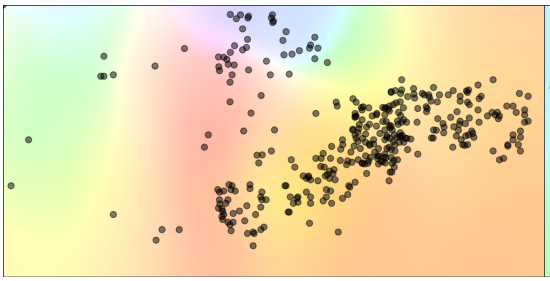
(c) Computed segmentation

(d) Computed optical flow, EPE = 1.88

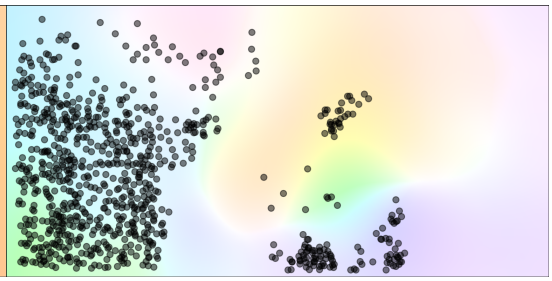
(e) Homography



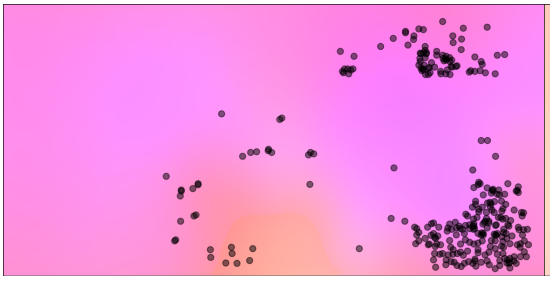
(f) PCA-Flow, EPE = 5.58



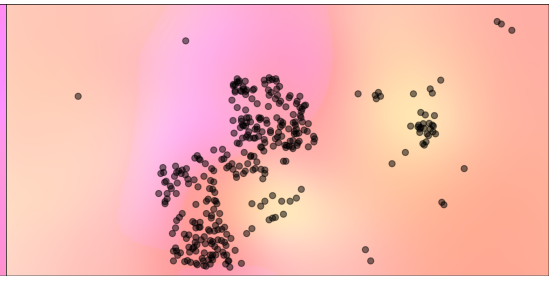
(g) Model 1



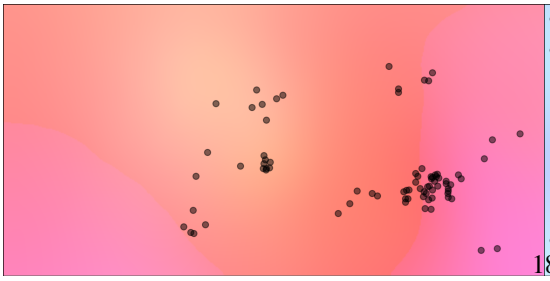
(h) Model 2



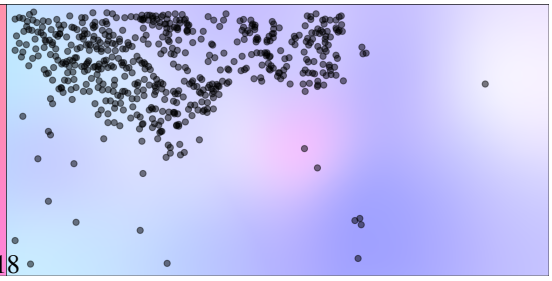
(i) Model 3



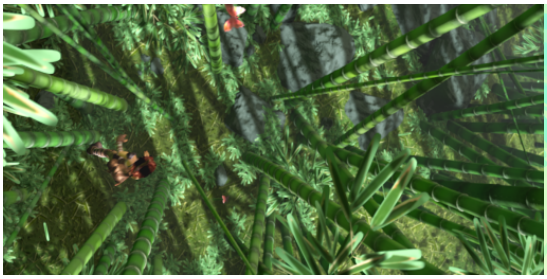
(j) Model 4



(k) Model 5

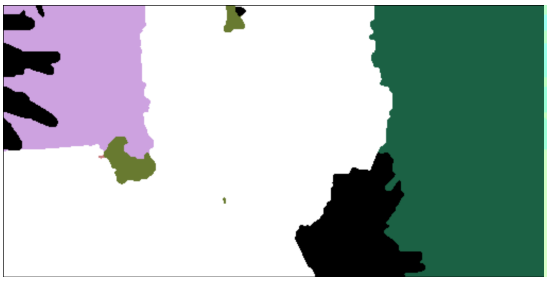


(l) Model 6



(a) Frame

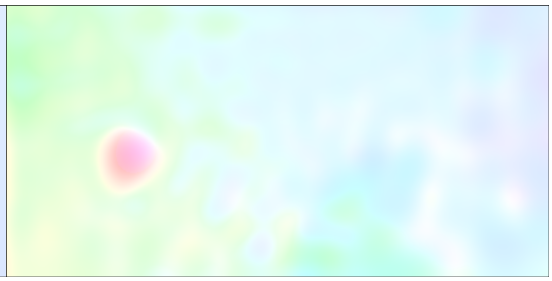
(b) Ground truth flow



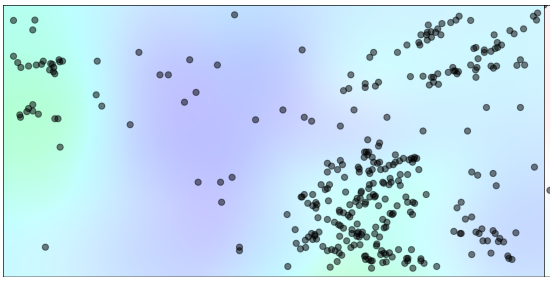
(c) Computed segmentation

(d) Computed optical flow, EPE = 0.76

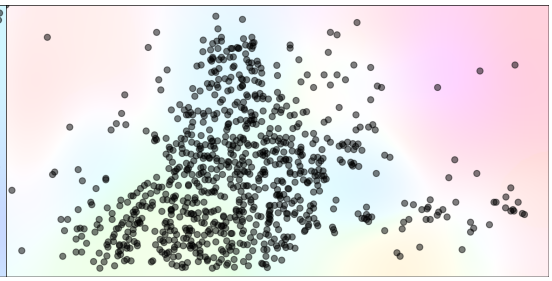
(e) Homography



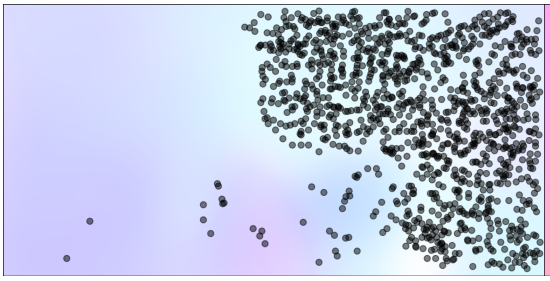
(f) PCA-Flow, EPE = 0.86



(g) Model 1

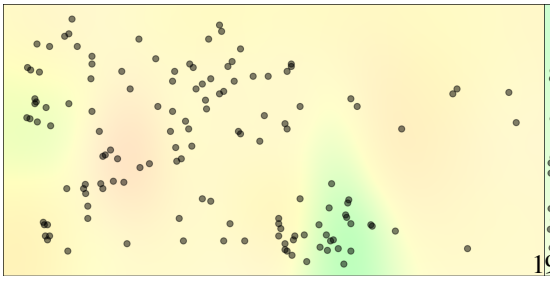


(h) Model 2

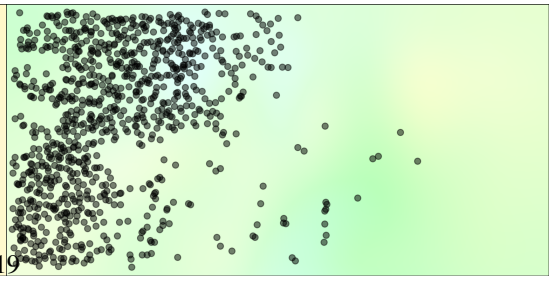


(i) Model 3

(j) Model 4



(k) Model 5



(l) Model 6

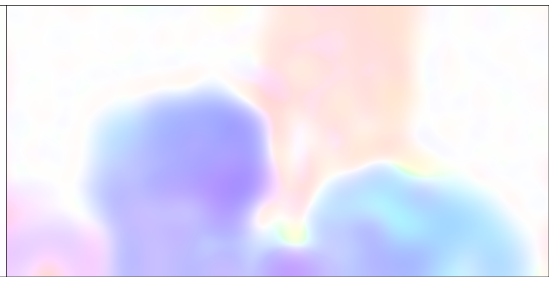
(a) Frame

(b) Ground truth flow

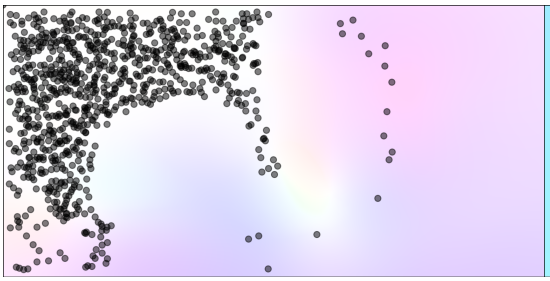
(c) Computed segmentation

(d) Computed optical flow, EPE = 0.88

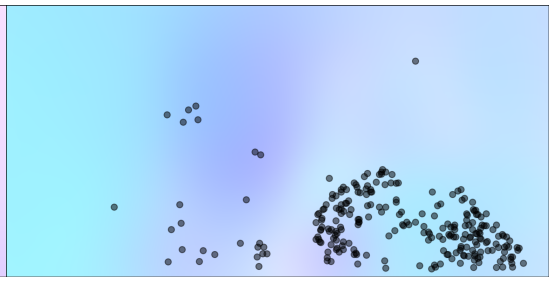
(e) Homography



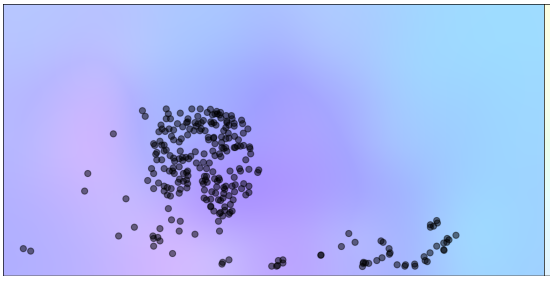
(f) PCA-Flow, EPE = 1.13



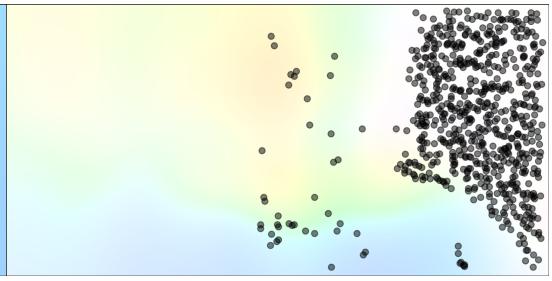
(g) Model 1



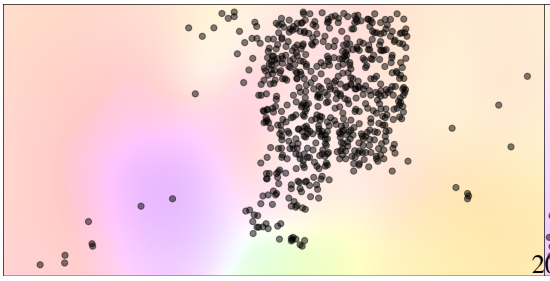
(h) Model 2



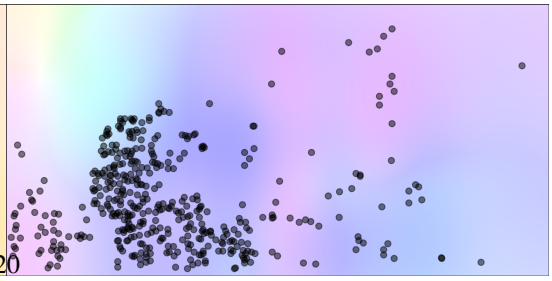
(i) Model 3



(j) Model 4



(k) Model 5



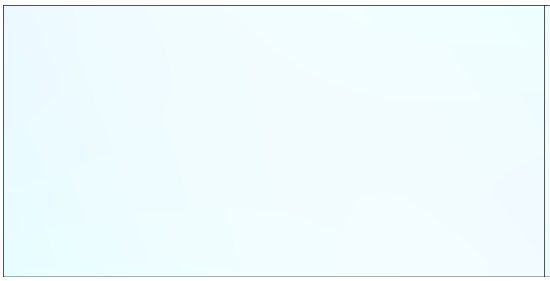
(l) Model 6

(a) Frame

(b) Ground truth flow

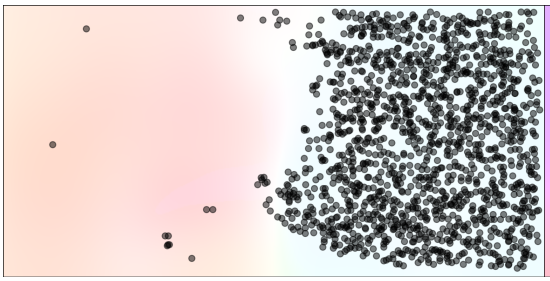
(c) Computed segmentation

(d) Computed optical flow, EPE = 0.42

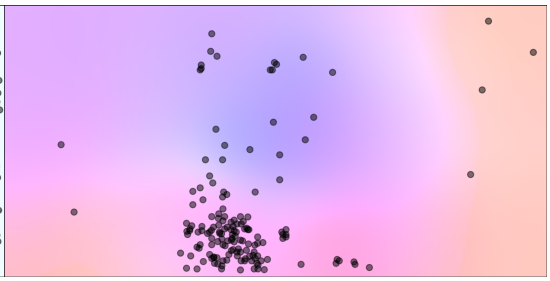


(e) Homography

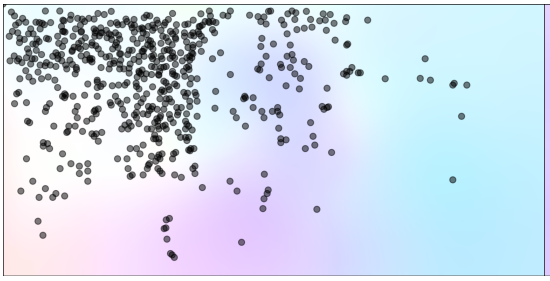
(f) PCA-Flow, EPE = 0.61



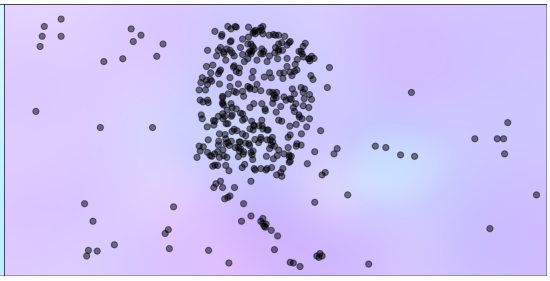
(g) Model 1



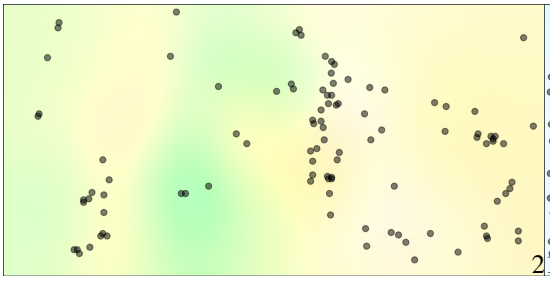
(h) Model 2



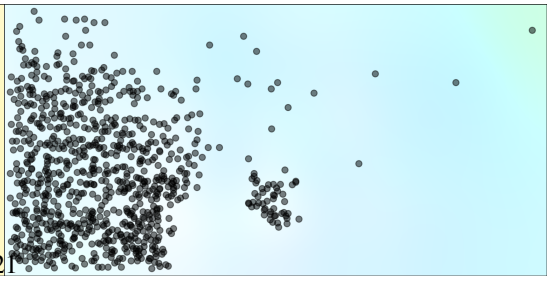
(i) Model 3



(j) Model 4



(k) Model 5

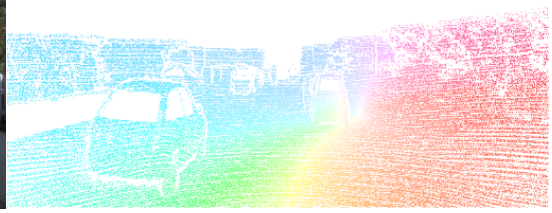


(l) Model 6

5.3 KITTI

The following pages show examples from KITTI. Since the motion in KITTI is inherently low dimensional, it is fairly well captured by the pure PCA-Flow approach; using multiple models does not improve the accuracy in terms of endpoint error. It does, however, reduce the number of "wrong" pixels with an error larger than 3 pixel. Additionally, as shown in the following examples, it increases the accuracy near motion boundaries.

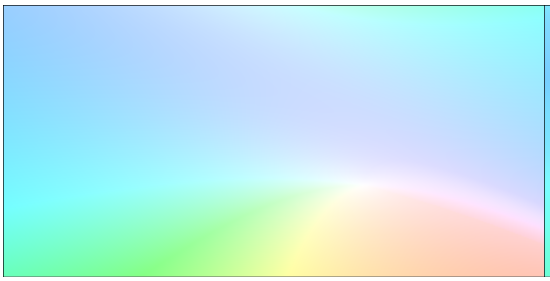
(a) Frame



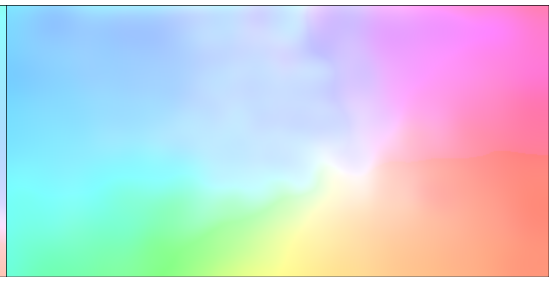
(b) Ground truth flow

(c) Computed segmentation

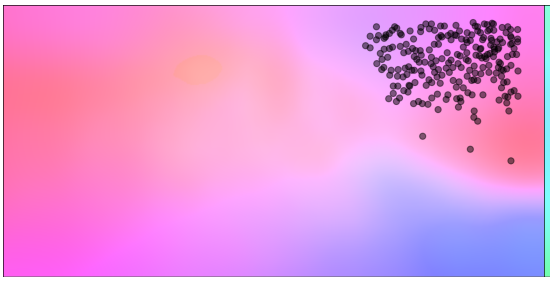
(d) Computed optical flow, EPE = 1.07



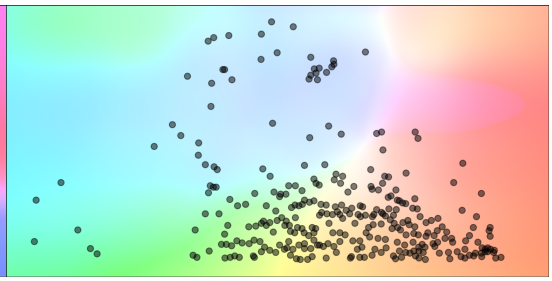
(e) Homography



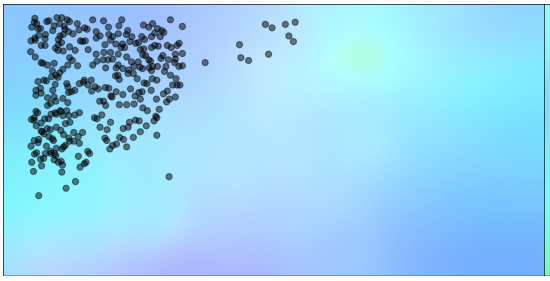
(f) PCA-Flow, EPE = 1.34



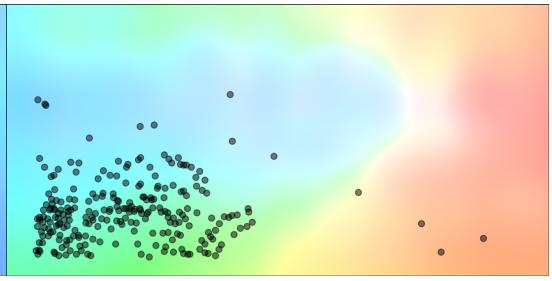
(g) Model 1



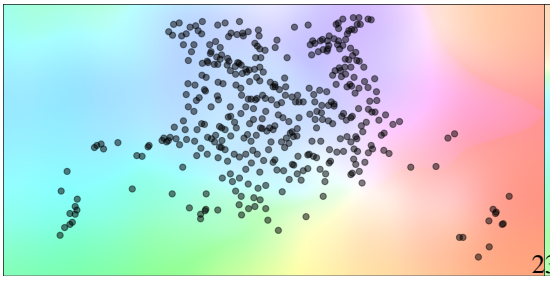
(h) Model 2



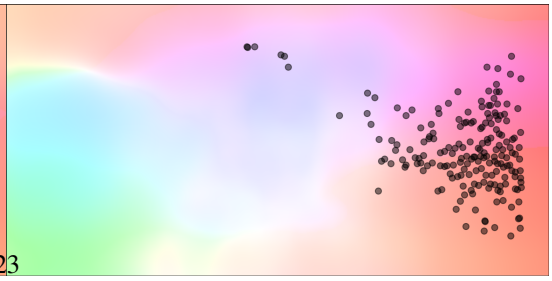
(i) Model 3



(j) Model 4

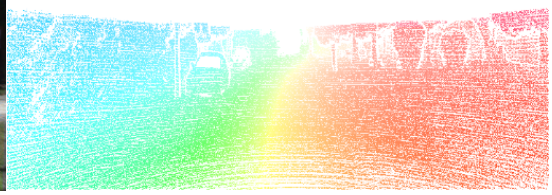


(k) Model 5



(l) Model 6

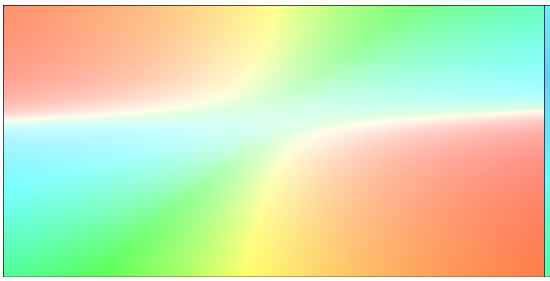
(a) Frame



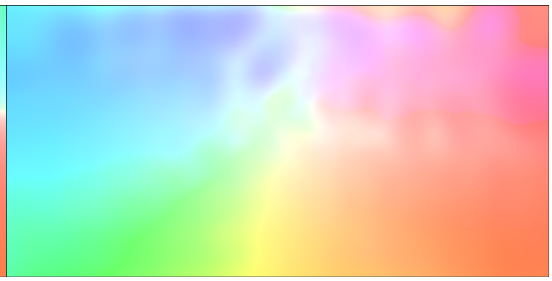
(b) Ground truth flow

(c) Computed segmentation

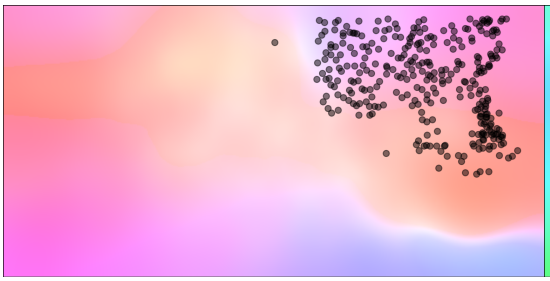
(d) Computed optical flow, EPE = 2.69



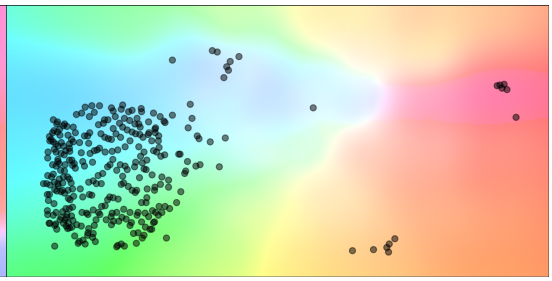
(e) Homography



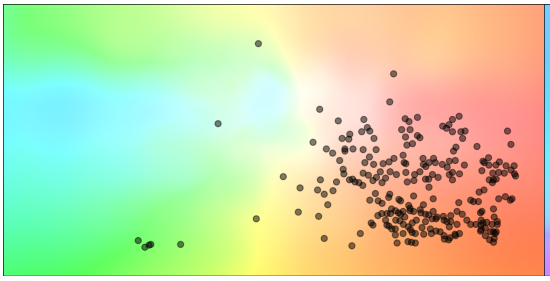
(f) PCA-Flow, EPE = 4.88



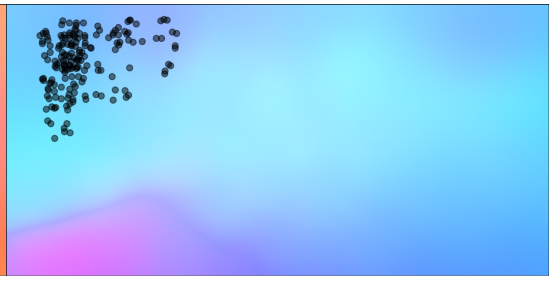
(g) Model 1



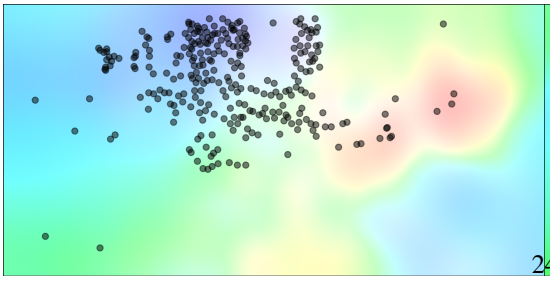
(h) Model 2



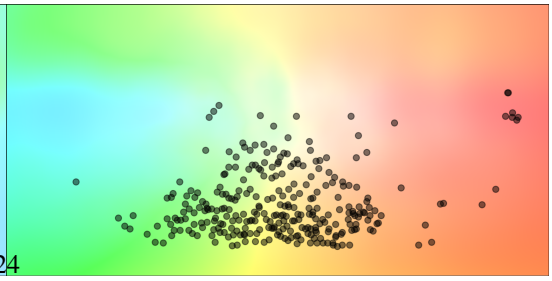
(i) Model 3



(j) Model 4

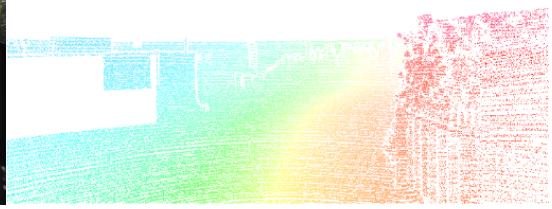


(k) Model 5



(l) Model 6

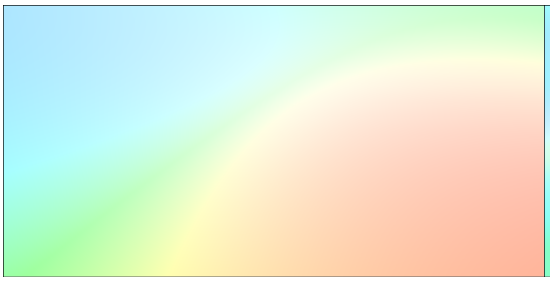
(a) Frame



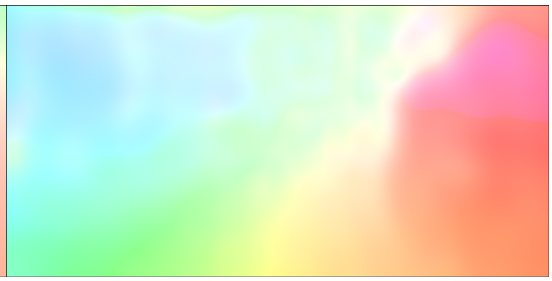
(b) Ground truth flow

(c) Computed segmentation

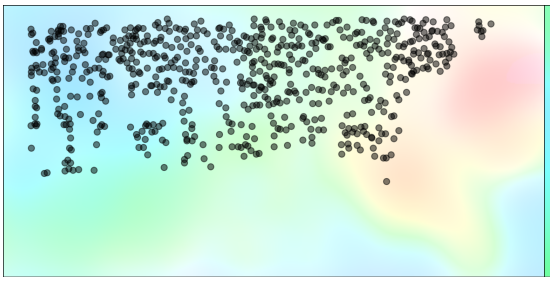
(d) Computed optical flow, EPE = 2.08



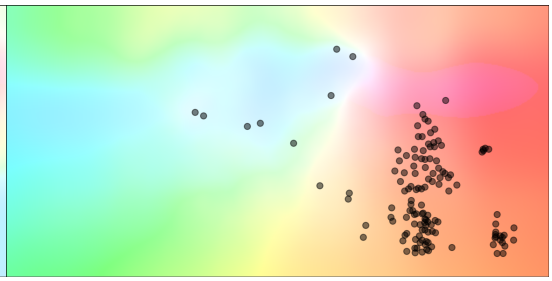
(e) Homography



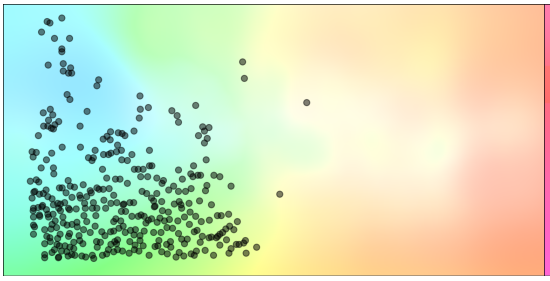
(f) PCA-Flow, EPE = 2.71



(g) Model 1

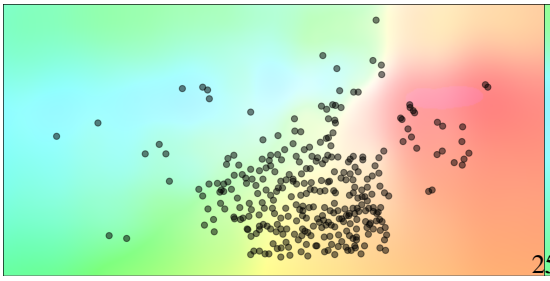


(h) Model 2

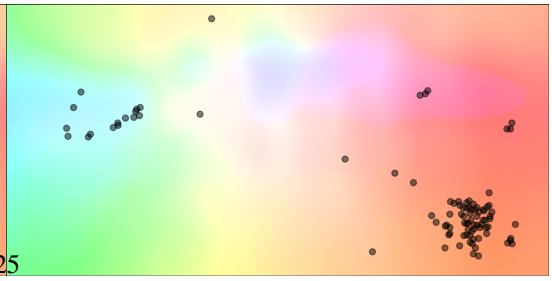


(i) Model 3

(j) Model 4

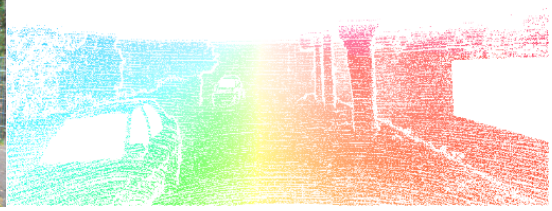


(k) Model 5



(l) Model 6

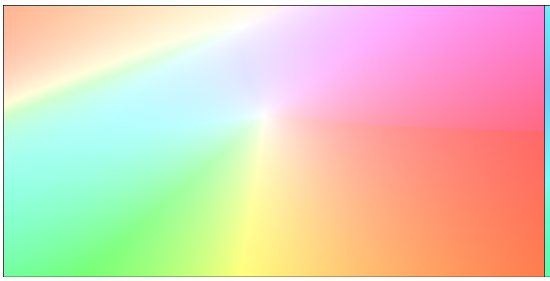
(a) Frame



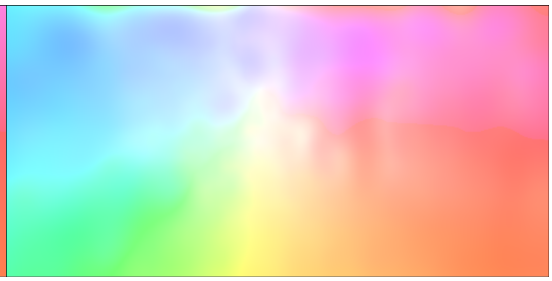
(b) Ground truth flow

(c) Computed segmentation

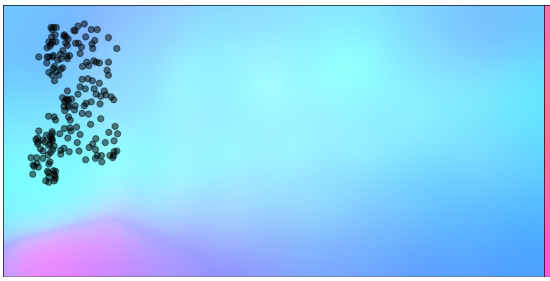
(d) Computed optical flow, EPE = 3.90



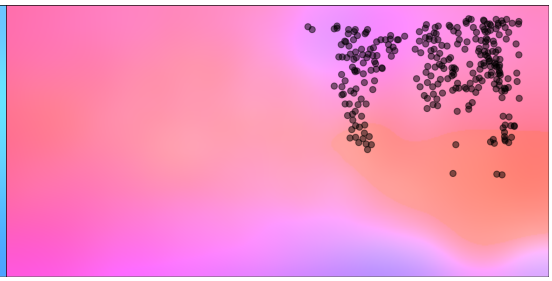
(e) Homography



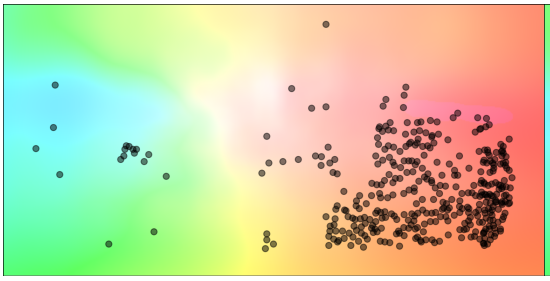
(f) PCA-Flow, EPE = 4.42



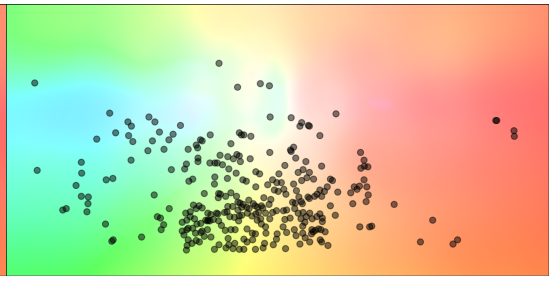
(g) Model 1



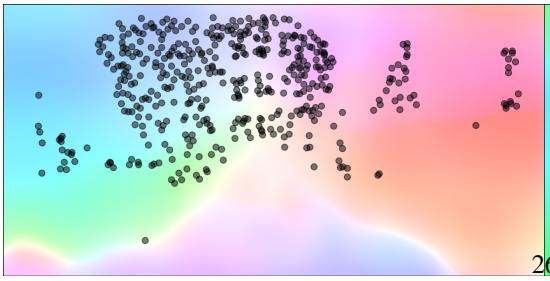
(h) Model 2



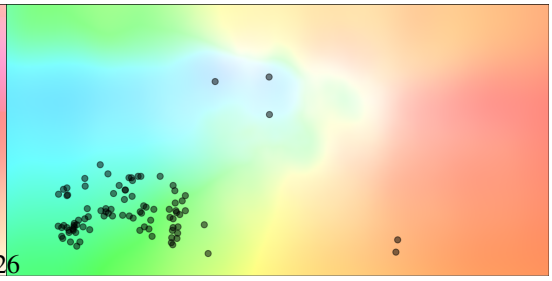
(i) Model 3



(j) Model 4



(k) Model 5

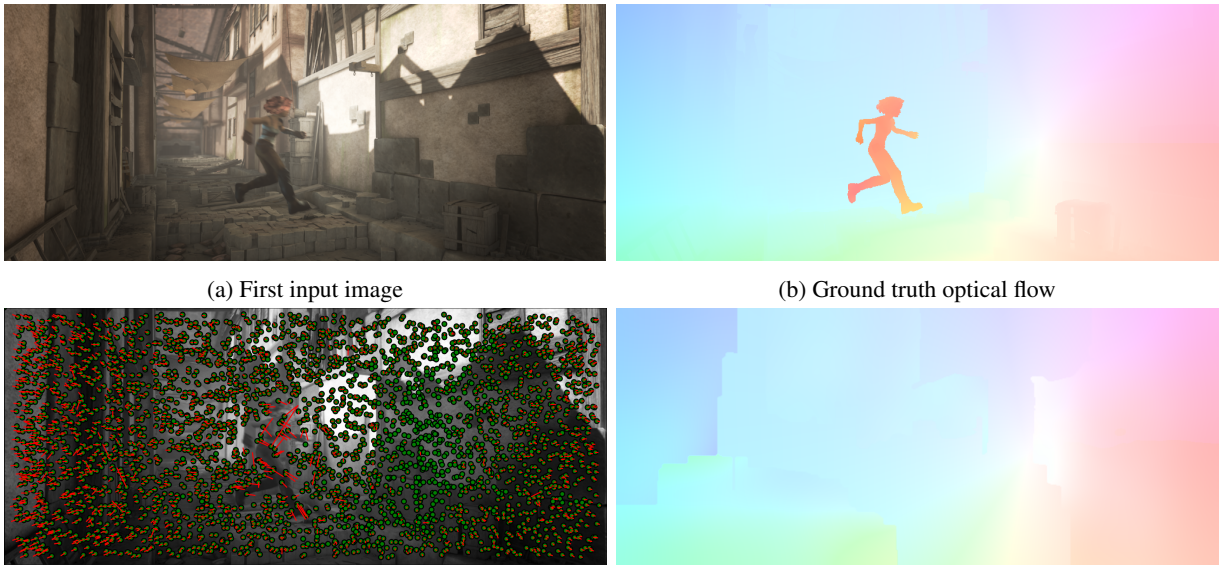


(l) Model 6

6 Failure cases

Our algorithm fails primarily for two reasons, missing or wrong features, and large unstructured regions. Figs. 24 and 25 show examples for both cases.

In Fig. 24, the girl is absent from the estimated optical flow field. As can be seen from Fig. 24(c), due to motion blur, not many features are detected on her body, especially on her legs. Even worse, the few features that are detected are very noisy, and are eliminated by the robust estimation. Better features can help improve the performance in cases like this; nevertheless, such features often come at higher computational cost. Whether they should be used or not is therefore dependent on the application; here, we decided against it.



(a) First input image

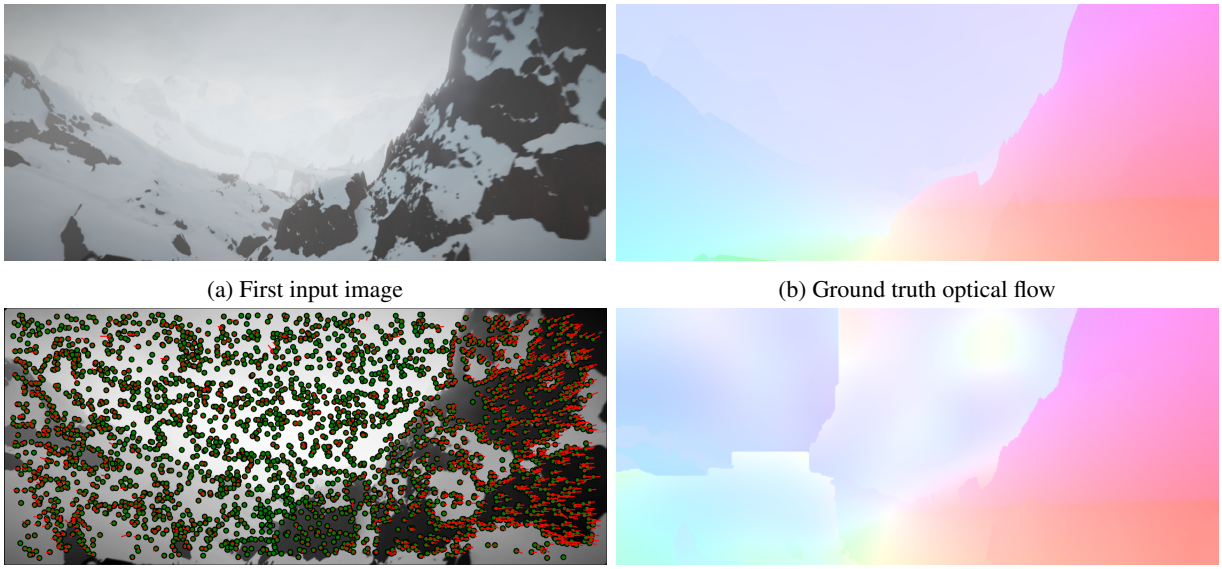
(b) Ground truth optical flow

(c) Matched features

(d) Estimated optical flow

Figure 24: Failure case: Missing or incorrect features.

In Fig. 25, one can see a wrong, "blocky" structure in the background. While many features are found, they are not all assigned to the same model. In such a case and in the absence of other image cues, the MRF tends to create blocky assignments, causing artifacts at the seams. One way to fix this would be to use a better inference scheme than the simple pairwise MRF we currently use; for example a densely connected CRF. We leave this for future work.



(a) First input image

(b) Ground truth optical flow

(c) Matched features

(d) Estimated optical flow

Figure 25: Failure case: Artifacts in weakly structured regions.

References

[1] S. Baker and I. Matthews. Lucas-kanade 20 years on: A unifying framework. *International Journal of Computer Vision*, 56(3):221–255, 2004. 8

[2] M. J. Black, Y. Yacoob, A. D. Jepson, and D. J. Fleet. Learning parameterized models of image motion. In *IEEE Conf. on Computer Vision and Pattern Recognition, CVPR-97*, pages 561–567, Puerto Rico, June 1997. 8