

Supplemental Material to CVPR Submission 1941: Optical Flow for Mostly Rigid Scenes

April 24, 2017

Contents

1	Additional details	2
1.1	Semantic rigidity estimation: CNN architecture details and training procedure .	2
1.2	Derivation of direction-based rigidity likelihood	3
2	Ablation study	5
3	Additional results	7
3.1	Sintel, clean pass	8
3.2	Sintel, final pass	14
3.3	KITTI	20
4	Failure cases	24



Figure 1: Results of rigidity estimation on the MPI-Sintel test set. Top: Original frame; Bottom: probability map of rigidity (white is likely rigid, black likely moving independently).

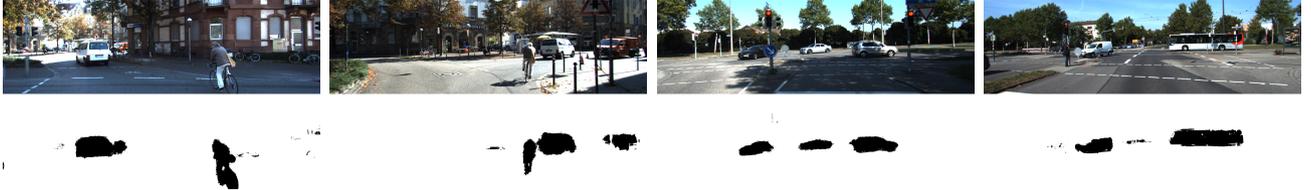


Figure 2: Results of rigidity estimation on KITTI 2015. Top: Original frame; Bottom: probability map of rigidity.

1 Additional details

Due to space restrictions, some details had to be omitted from the main paper; we present those details here. Sec. 1.1 describes the learning procedure and our CNN architectures in greater detail than in the paper and provides more semantic rigidity estimation results. Sec. 1.2 provides the derivation of the motion-based rigidity formulation (11) in the paper.

1.1 Semantic rigidity estimation: CNN architecture details and training procedure

As described in our paper, each of the two datasets provide different data, and thus we used slightly different procedures to estimate rigidity on each of the datasets. Here we describe the details. Additionally, Fig. 1 and Fig. 2 show some more examples of the outputs of the networks on images that were not seen during test time.

KITTI 2015: We followed the same procedure as previous work on semantic segmentation on KITTI [4], since it was shown to be successful. We used the DeepLab architecture [1] modifying the output layer to classify 22 classes: aeroplane, bicycle, bird, boat, building, bus, car, cat, cow, dog, floor, grass, horse, motorbike, road, sheep, sidewalk, sky, train, water, person and background, where background includes anything that is not one of the other 21 classes. We initialized the weights with the VGG model trained on Pascal VOC and then fine-tuned it on our categories using a fixed momentum of 0.9, weight decay of 0.0005 and learning rate of 0.0001 for the first 100K iterations, reduced by 0.1 after every 50K steps, during 200K iterations. We used a dense CRF [2] on top, where the unaries are the CNN output at each pixel and the pairwise potentials are position and a bilateral kernel with both position and RGB values. The inference in the dense CRF model is performed using 10 steps of mean-field approximate inference. At test time we obtain a probability over the classes. We estimate rigidity by choosing the class with the highest probability, and classifying the pixel as rigid or non-rigid based on whether an object in the class is capable of moving independently (for example, car) or not (for example, building). The accuracy of rigidity classification on the training set is 96.09%, where rigid parts are correctly classified 96.93% of the time, and independent moving parts are correctly classified 91.51% of the time.

MPI-Sintel: In this dataset there is no previous work on estimating rigidity. Thus, we used one of the latest released versions of the DeepLab architecture, called DeepLab-Coco-LargeFov,

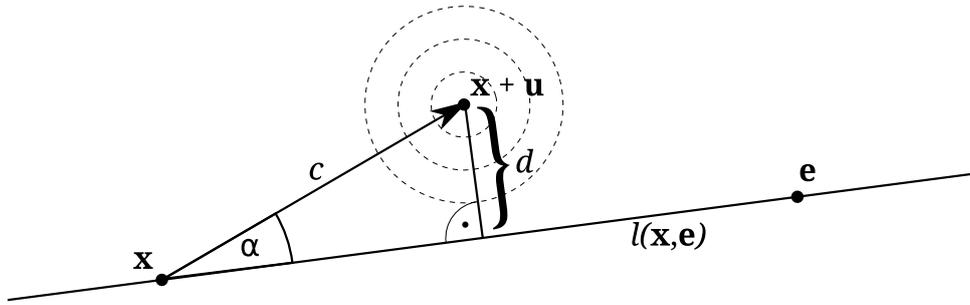


Figure 3: Illustration

which is pretrained including extra annotations from the MS-COCO dataset¹. We modified the output layer to classify pixels as rigid or nonrigid. We fine-tuned all layers using the same parameters as before for 1.4K iterations. This small number of iterations was selected to avoid overfitting. At test time, we obtain a probability of rigidity, and we compute the final estimate of rigidity by thresholding at 0.5. The accuracy of the estimation on a validation set is 94.2%.

1.2 Derivation of direction-based rigidity likelihood

While the accuracy of the CNN-based rigidity estimation is surprisingly high, there are still occasions when it fails. This may happen for example when there is strong motion blur in the rigid regions, causing these to be classified as independently moving, which is a false positive. On the other hand, a noisy or over-saturated appearance of objects can cause the semantic segmentation to fail and result in a false negative. Therefore, we combine the CNN-based rigidity estimation with a motion-based estimation, described in this section.

For a given point \mathbf{x} , our model assumes the measured corresponding point $\mathbf{x}' = \mathbf{x} + \mathbf{u}$ to have a Gaussian error distribution around the true correspondence with covariance matrix $\Sigma = \sigma^2 \mathbf{I}$. For a given rigid point (denoted by the conditioning on $r = 1$) and a given focus of expansion \mathbf{e} , the probability of the true correspondence pointing towards \mathbf{e} is then given as

$$p(\mathbf{x}' | r = 1, \mathbf{e}) = \int_{y \in l(\mathbf{x}, \mathbf{e})} \frac{1}{2\pi\sigma_m^2} \exp\left(-\frac{1}{2}\mathbf{y}^T \Sigma^{-1} \mathbf{y}\right) dy, \quad (1)$$

where $l(\mathbf{x}, \mathbf{e})$ denotes the line that goes through \mathbf{x} and \mathbf{e} . Figure 3 shows an illustration.

Since the error distribution is Gaussian, every marginal is also Gaussian. Therefore, the line integral in (1) is given as

$$p(\mathbf{x}' | r = 1, \mathbf{e}) = \frac{1}{\sqrt{2\pi\sigma_m^2}} \exp\left(-\frac{d^2}{2\sigma_m^2}\right), \quad (2)$$

where d denotes the distance of \mathbf{x}' to $l(\mathbf{x}, \mathbf{e})$, as shown in Figure 3.

In the following, it will be convenient to express the correspondence point \mathbf{x}' in (2) in terms of the angle α and the displacement magnitude $c = \|\mathbf{u}\|$. Then,

$$p(\alpha, c | r = 1, \mathbf{e}) = \frac{1}{\sqrt{2\pi\sigma_m^2}} \exp\left(-\frac{c^2 \sin^2(\alpha)}{2\sigma_m^2}\right). \quad (3)$$

We can thus drop \mathbf{e} from the equation.

¹Further details can be found on their webpage <http://ccvl.stat.ucla.edu/software/deeplab/deeplab-coco-largefov/>

The likelihood of a point being rigid given the measurements α, c is now

$$\begin{aligned} p(r = 1|\alpha, c) &= \frac{p(\alpha|r = 1, c)p(r = 1)}{p(\alpha|r = 0, c)p(r = 0) + p(\alpha|r = 1, c)p(r = 1)} \\ &= \frac{\frac{1}{2}p(\alpha, c|r = 1)p(r = 1)}{p(\alpha|r = 0, c)p(r = 0) + \frac{1}{2}p(\alpha, c|r = 1)p(r = 1)}. \end{aligned} \quad (4)$$

Abbreviating the prior for rigidity $p(r = 1)$ as p_1 and setting $p(\alpha|r = 0, c) = \frac{1}{2\pi}$ (the motion of independently moving regions is supposed to be uniformly distributed), we get

$$p(r = 1|\alpha, c) = \frac{p_1 p(\alpha, c|r = 1)}{\frac{Z(1-p_1)}{2\pi} + p_1 p(\alpha, c|r = 1)} \quad (5)$$

and for uninformative priors $p_1 = 0.5$

$$p(r = 1|\alpha, c) = \frac{p(\alpha, c|r = 1)}{\frac{Z}{2\pi} + p(\alpha, c|r = 1)}. \quad (6)$$

What remains is to compute Z . Using Eq. (3), it can be computed as

$$\begin{aligned} Z &= p(c|r = 1) = \int_0^{2\pi} p(\alpha, c|r = 1) d\alpha \\ &= \frac{1}{\sqrt{2\pi\sigma_m^2}} \int_0^{2\pi} \exp\left(-\frac{c^2 \sin^2(\alpha)}{2\sigma_m^2}\right) d\alpha \\ &= \frac{1}{\sqrt{2\pi\sigma_m^2}} \int_0^{2\pi} \exp(-t(1 - \cos(2\alpha))) d\alpha \quad \left[t = \frac{c^2}{4\sigma_m^2}, \sin^2(x) = \frac{1}{2}(1 - \cos(2x))\right] \\ &= \frac{1}{\sqrt{2\pi\sigma_m^2}} \exp(-t) \int_0^{2\pi} \exp(t \cos(2\alpha)) d\alpha \\ &= \frac{1}{\sqrt{2\pi\sigma_m^2}} \exp(-t) \frac{1}{2} \int_0^{4\pi} \exp(t \cos(\beta)) d\beta \quad [\beta = 2\alpha] \\ &= \frac{1}{\sqrt{2\pi\sigma_m^2}} \exp(-t) \int_0^{2\pi} \exp(t \cos(\beta)) d\beta \\ &= \frac{1}{\sqrt{2\pi\sigma_m^2}} \exp(-t) 2\pi \mathbb{I}_0(t) \\ &= \frac{\sqrt{2\pi}}{\sigma_m} \exp(-t) \mathbb{I}_0(t), \end{aligned} \quad (7)$$

with $\mathbb{I}_0(x)$ the modified Bessel function of the first kind. Inserting Eq. (7) and Eq. (3) into Eq. 6 yields Eq. (11) from the main paper:

$$\begin{aligned} p(\mathbf{x} \text{ is rigid}) &= p(r = 1|\alpha, c) \\ &= \frac{\exp(-2t \sin^2(\alpha))}{\exp(-t)\mathbb{I}_0(t) + \exp(-2t \sin^2(\alpha))}. \end{aligned} \quad (8)$$

2 Ablation study

This section provides an additional ablation study that had to be omitted from the paper due to space limitations. Here, we test how different subcomponents of our algorithm impact the end result². To assess the impact in different regions of the frame, we provide the errors both on the full frames and only in the ground truth rigid regions.

For the ablation study, we successively switch on four steps: *occlusion reasoning*, *coplanarity refinement*, *nonlinear initialization of \hat{b}^-* , and *spatial priors*.

- **Occlusion reasoning** refers to the estimation of the visibility maps V^+, V^- using the forward-backward consistency, as described in Section 4 in the paper. If this step is switched off, we set $V^- = V^+ = 1$ everywhere, and therefore do not explicitly exclude occluded pixels from subsequent computations.
- **Coplanarity refinement** refers to the second part of the initial alignment (Eq. (5) in the main paper)³. This step refines the initial homographies \bar{H}^-, \bar{H}^+ to ensure that after registration all residual flow vectors meet in the two epipoles $\mathbf{e}^-, \mathbf{e}^+$. The optimization yields \hat{H}^-, \hat{H}^+ . If this step is switched off, we simply set $\hat{H}^+ = \bar{H}^+, \hat{H}^- = \bar{H}^-$.
- **Nonlinear initialization of \hat{b}^-** refers to initializing \hat{b}^- using Eq.(8), *i.e.* choosing \hat{b}^- so that the resulting backward \hat{A}^- structure is as similar as possible under a robust error norm to the initial forward structure \hat{A}^+ . Note that, without loss of generality, \hat{b}^+ is always chosen so that the MAD of \hat{A}^+ is 1.

If this step is switched off, use Eq. (7) to equate \hat{A}^+ and \hat{A}^- . This produces an estimate of \hat{b}^- per pixel, of which we take the median to arrive at the global estimate of \hat{b}^- .

$$\hat{b}^- = \underset{\mathbf{x}}{\text{median}} \frac{1}{A^+(\mathbf{x})} \frac{w^-(\mathbf{x})}{\|\mathbf{e}^- - \mathbf{x}\| - w^-(\mathbf{x})} \quad (9)$$

- **Spatial priors** refer to the 1st- and 2nd-order spatial smoothness regularizers in our objective function (17). To disable those, we set $\lambda_{1st} = \lambda_{2nd} = 0$.

	occlusion reasoning	coplanarity refinement	nonlinear b^- initialization	spatial priors	EPE rigid	EPE all
Baseline	□	□	□	□	1.859	3.798
+occlusions	■	□	□	□	1.733	3.705
+coplanarity	■	■	□	□	1.695	3.671
+nonlin-init	■	■	■	□	1.619	3.628
Full	■	■	■	■	1.602	3.614

Table 1: Errors when successively switching on parts of the algorithm

Table 1 shows the improvement when successively switching on more parts of the algorithm. The occlusion reasoning has the largest positive impact on the error, since it allows the algorithm to properly merge the flow in both directions from the reference frame. Following this, the most important parts are the nonlinear initialization and ensuring the coplanarity. The spatial priors serve mostly to remove flow noise near boundaries. This improves the result visually, but has a fairly small numerical impact.

²The results in this section were obtained using a reduced version of the MPI-Sintel training set containing every 4th frame, and only the `final` pass.

³All equation numbers refer to equations in the main paper.

	occlusion reasoning	coplanarity refinement	nonlinear b^- initialization	spatial priors	EPE rigid	EPE all
no-occlusions	□	■	■	■	1.809	3.759
no-coplanarity	■	□	■	■	1.642	3.645
no-nonlin-init	■	■	□	■	1.677	3.656
no-spatial-priors	■	■	■	□	1.619	3.628
Full	■	■	■	■	1.602	3.614

Table 2: Errors when disabling individual parts of the algorithm

Table 2 shows the impact when turning off individual components, but leaving all others intact. Again, we can observe that disabling the occlusion reasoning has the largest negative impact, followed by the nonlinear initialization and ensuring the coplanarity.

	Data term	1st order regularization	2nd order regularization	EPE rigid	EPE all
No-opt	□	□	□	1.6124	3.6214
No-spatial-priors	■	□	□	1.6194	3.6285
No-1st	■	□	■	1.6025	3.6138
No-2nd	■	■	□	1.6183	3.6274
Full	■	■	■	1.6024	3.6138

Table 3: Influence of regularization terms

Table 3 shows the impact of the different terms of the variational refinement (Eq. (17)). The cases are as described above. In addition, table 3 includes a complete omission of the variational refinement (*no-opt*), which uses only the initial structure estimate as described by Eq. (9) in the paper, and selective disabling of the individual regularizers ($\lambda_{1st} = 0$ for *no-1st* and $\lambda_{2nd} = 0$ for *no-2nd*).

When using the data term only (*no-spatial-priors*), the error is higher than when not using any optimization. Due to effects in the Sintel final pass such as motion blur, fog, vignetting etc. this is to be expected. Using the 2nd order regularization improves the results; interestingly, however, the impact of the 1st order regularization is negligible.

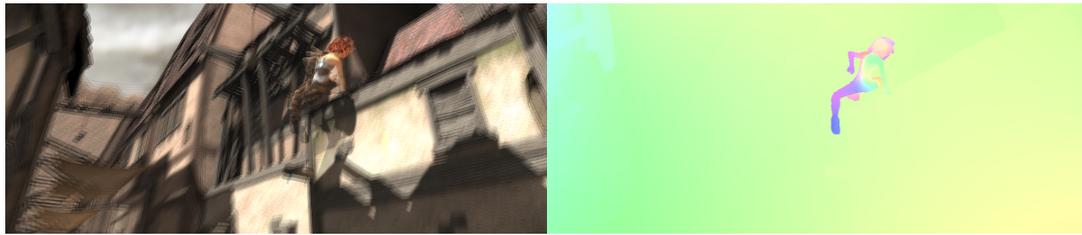
3 Additional results

This section shows more visual results on Sintel (clean + final pass) and KITTI. In all subsequent results,

- (a) shows the three unaligned input frames as overlays.
- (b) shows the ground truth optical flow
- (c) shows the estimated rigid regions, where red corresponds to rigid regions of the scene, and blue corresponds to moving regions.
- (d) shows the estimated structure. Since the estimated structure in the moving regions usually corresponds to extreme values in the structure estimate, we mask the estimated moving regions out; they are shown in purple.
- (e) shows the final, estimated optical flow
- (f) shows the change compared to the initial optical flow. In green regions, MR-Flow improves upon the initial flow method; in red regions, MR-Flow makes the results worse.

Note that for KITTI the ground truth flow is only given for a sparse sampling of pixels and does not include some objects such as moving people. In the result shown in p.22, bottom, for example, the flow estimated by our method (e) looks very different from the GT flow (b). However, this is merely an artifact of the GT visualization showing unlabelled pixels in white; in this example, all humans and the tram are not included in the ground truth flow, and the flow in the rigid parts of the scene is sampled very sparsely. For this reason, the background appears almost white and therefore different from the flow estimated by our method, while in reality, it is only unlabelled.

3.1 Sintel, clean pass



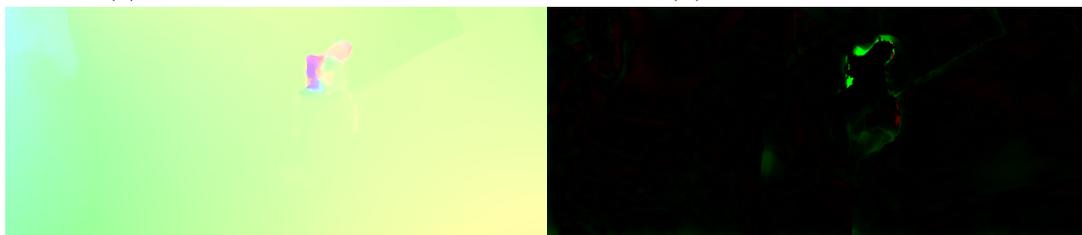
(a) Input images

(b) Ground truth flow



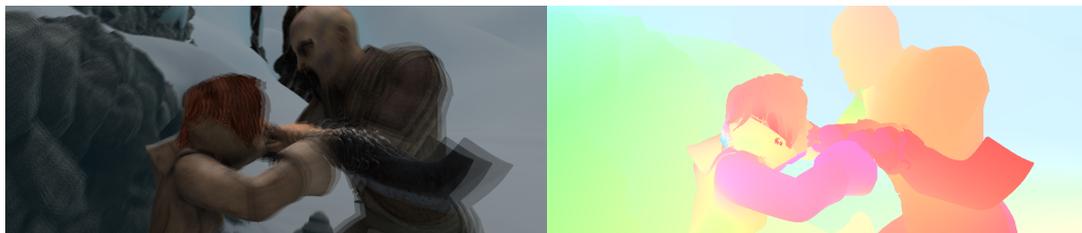
(c) Estimated rigid regions

(d) Estimated structure



(e) Estimated flow

(f) Comparison to initial flow



(a) Input images

(b) Ground truth flow



(c) Estimated rigid regions

(d) Estimated structure



(e) Estimated flow

(f) Comparison to initial flow



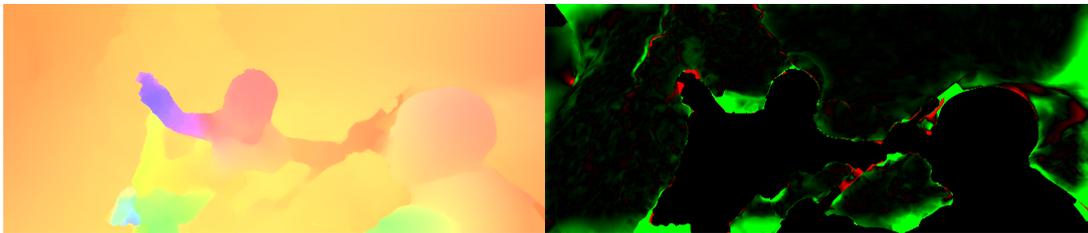
(a) Input images

(b) Ground truth flow



(c) Estimated rigid regions

(d) Estimated structure



(e) Estimated flow

(f) Comparison to initial flow



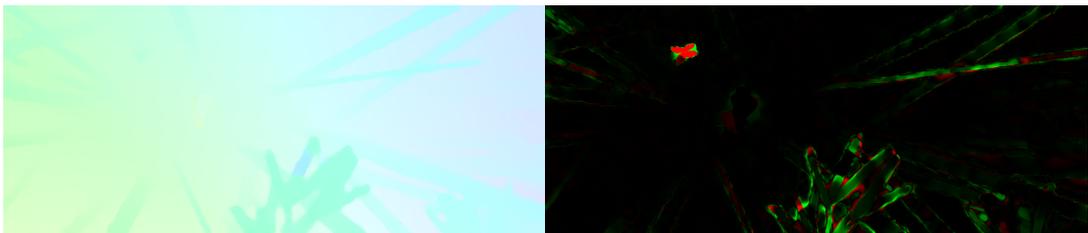
(a) Input images

(b) Ground truth flow



(c) Estimated rigid regions

(d) Estimated structure



(e) Estimated flow

(f) Comparison to initial flow



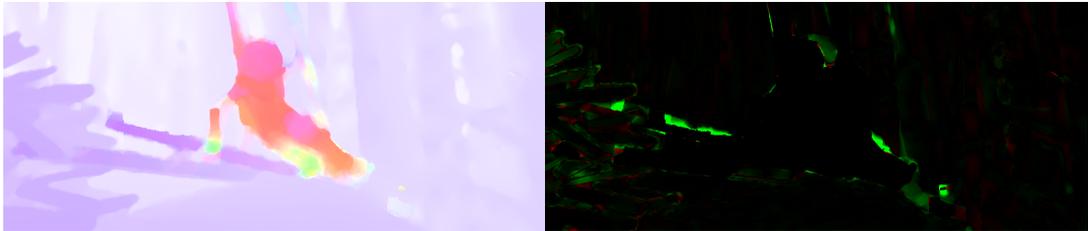
(a) Input images

(b) Ground truth flow



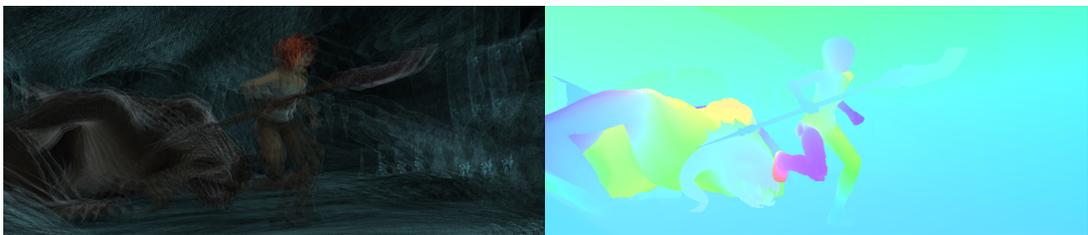
(c) Estimated rigid regions

(d) Estimated structure



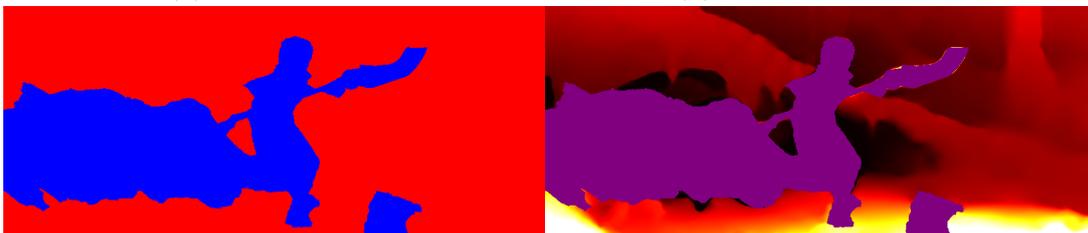
(e) Estimated flow

(f) Comparison to initial flow



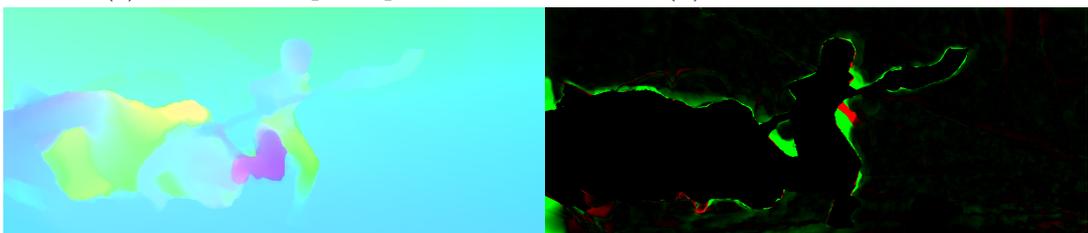
(a) Input images

(b) Ground truth flow



(c) Estimated rigid regions

(d) Estimated structure



(e) Estimated flow

(f) Comparison to initial flow



(a) Input images

(b) Ground truth flow



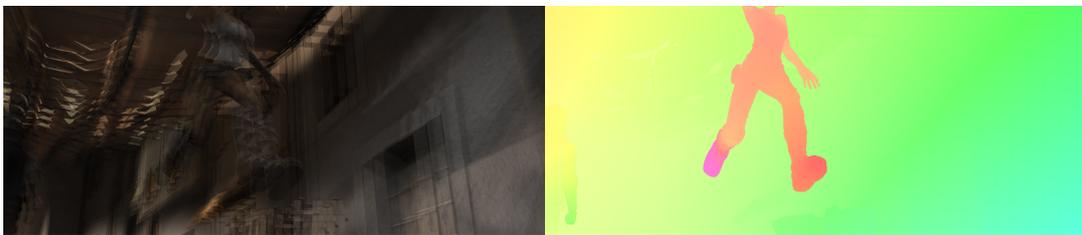
(c) Estimated rigid regions

(d) Estimated structure



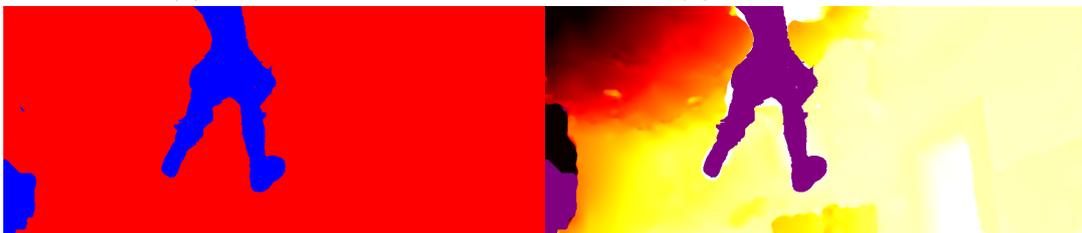
(e) Estimated flow

(f) Comparison to initial flow



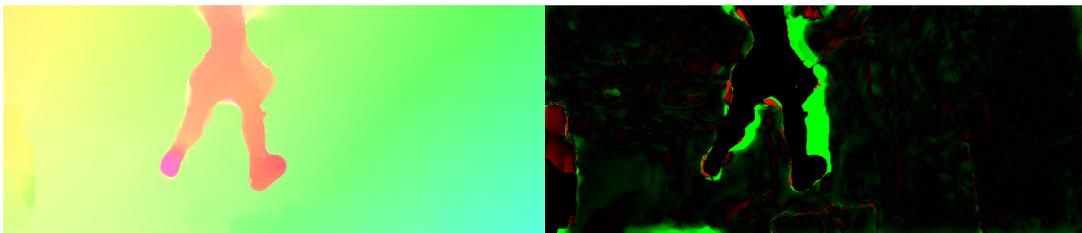
(a) Input images

(b) Ground truth flow



(c) Estimated rigid regions

(d) Estimated structure



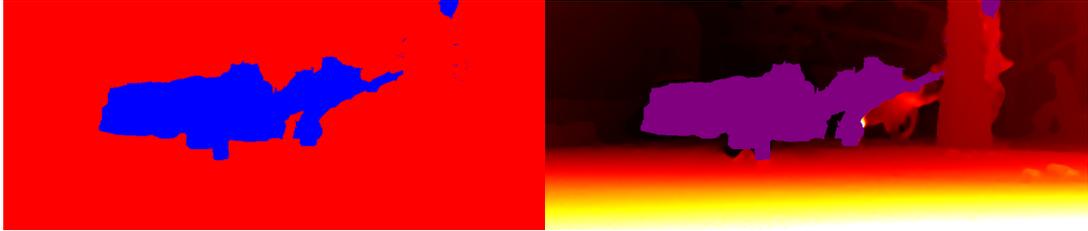
(e) Estimated flow

(f) Comparison to initial flow



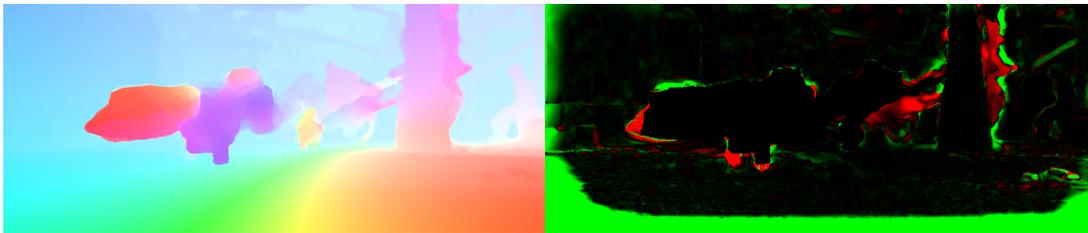
(a) Input images

(b) Ground truth flow



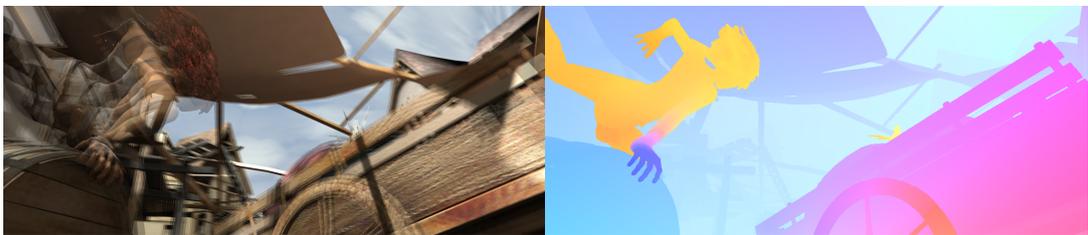
(c) Estimated rigid regions

(d) Estimated structure



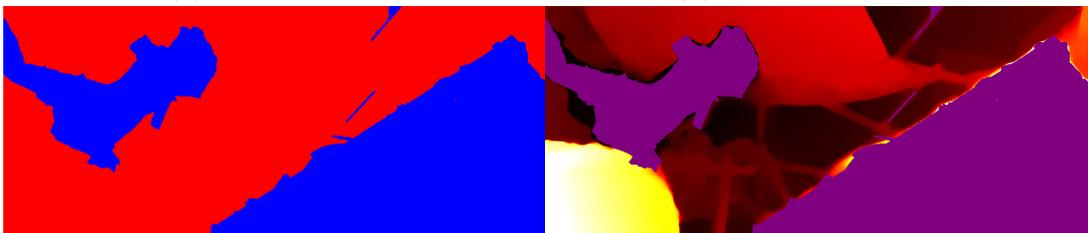
(e) Estimated flow

(f) Comparison to initial flow



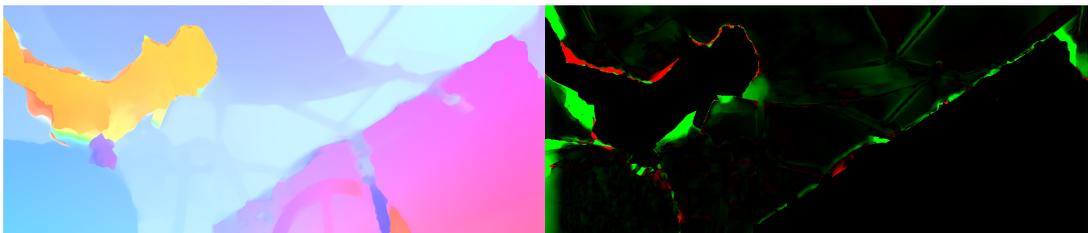
(a) Input images

(b) Ground truth flow



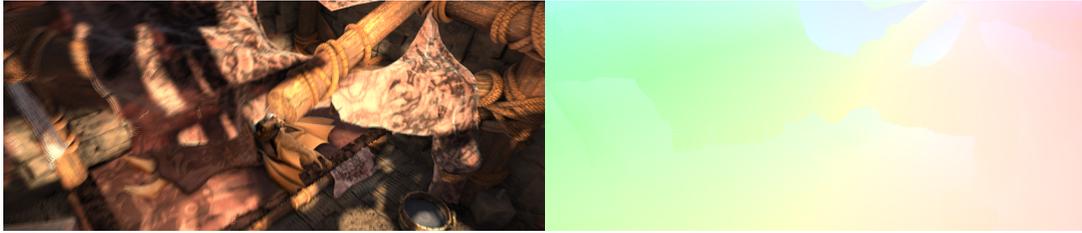
(c) Estimated rigid regions

(d) Estimated structure



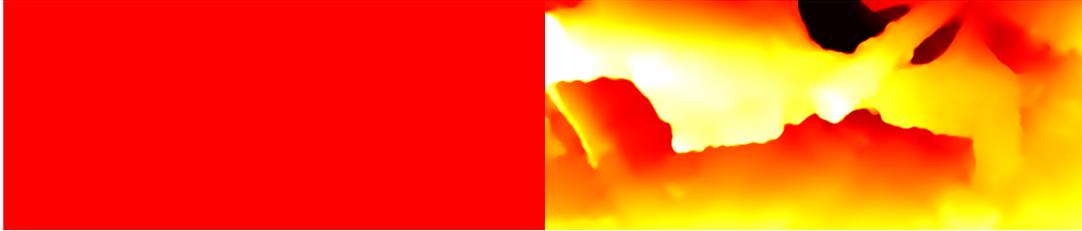
(e) Estimated flow

(f) Comparison to initial flow



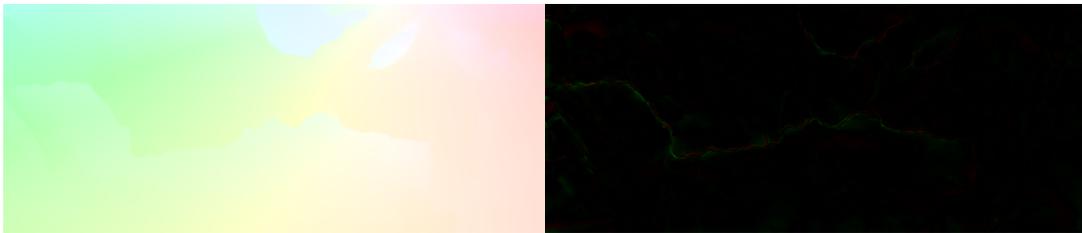
(a) Input images

(b) Ground truth flow



(c) Estimated rigid regions

(d) Estimated structure



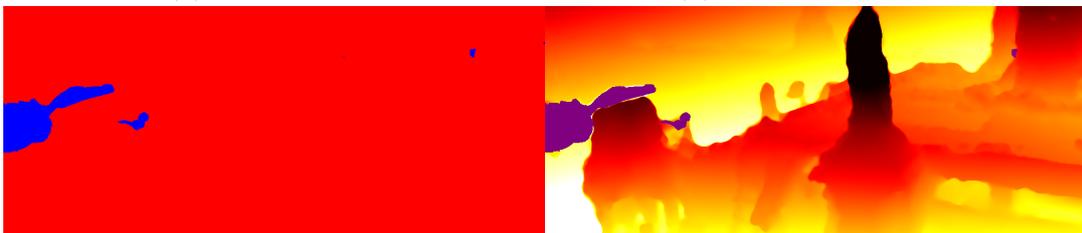
(e) Estimated flow

(f) Comparison to initial flow



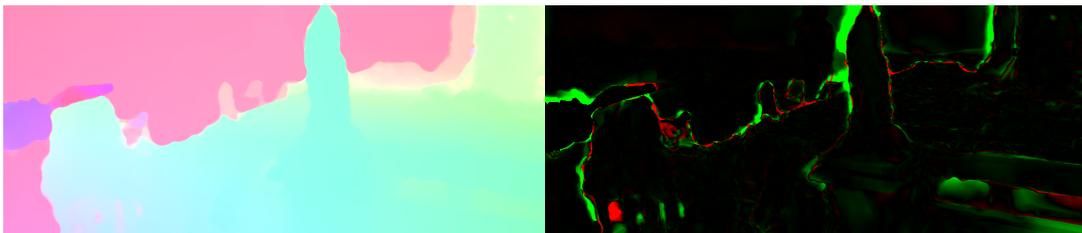
(a) Input images

(b) Ground truth flow



(c) Estimated rigid regions

(d) Estimated structure



(e) Estimated flow

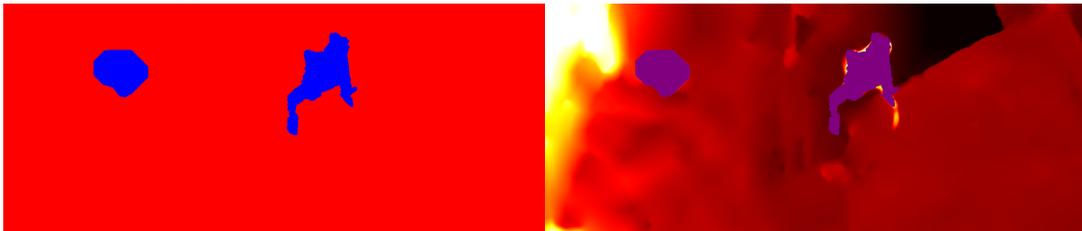
(f) Comparison to initial flow

3.2 Sintel, final pass



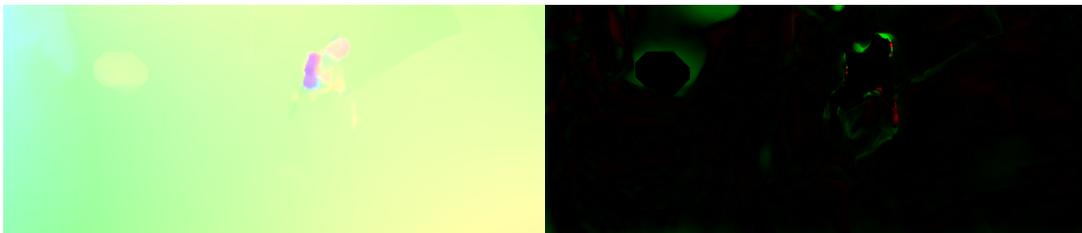
(a) Input images

(b) Ground truth flow



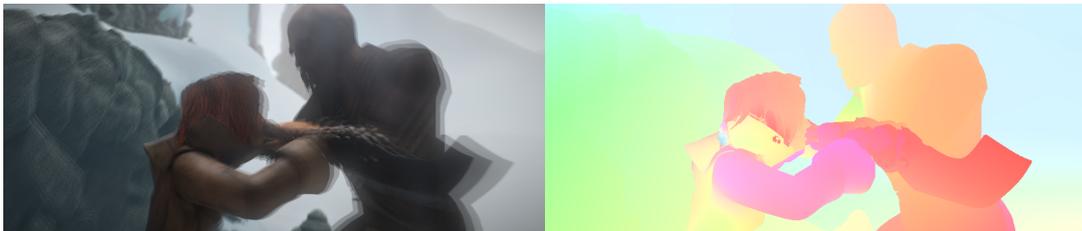
(c) Estimated rigid regions

(d) Estimated structure



(e) Estimated flow

(f) Comparison to initial flow



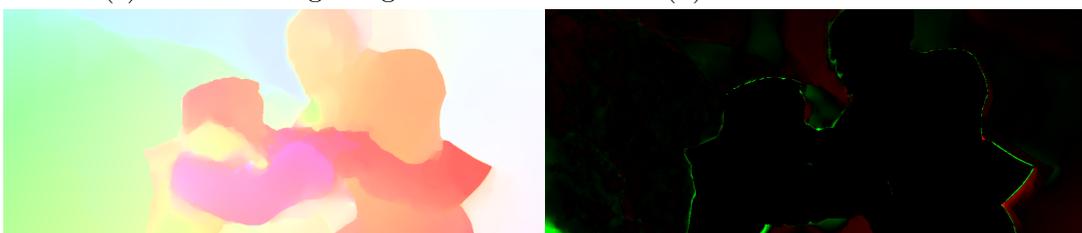
(a) Input images

(b) Ground truth flow



(c) Estimated rigid regions

(d) Estimated structure



(e) Estimated flow

(f) Comparison to initial flow



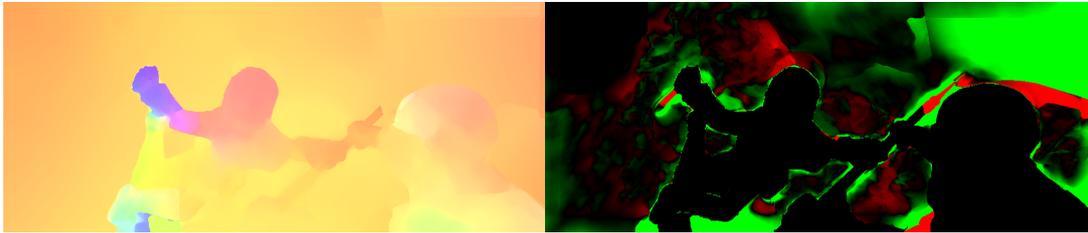
(a) Input images

(b) Ground truth flow



(c) Estimated rigid regions

(d) Estimated structure



(e) Estimated flow

(f) Comparison to initial flow



(a) Input images

(b) Ground truth flow



(c) Estimated rigid regions

(d) Estimated structure



(e) Estimated flow

(f) Comparison to initial flow



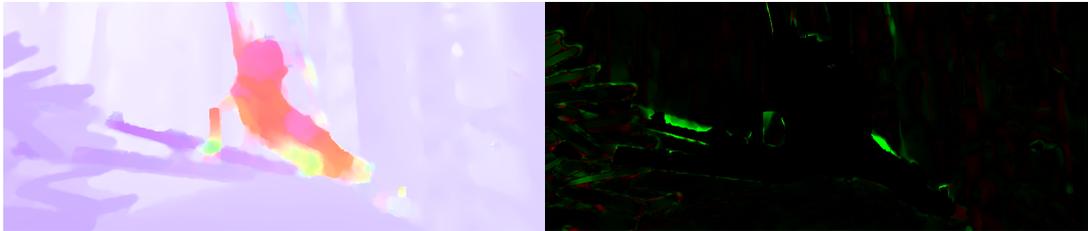
(a) Input images

(b) Ground truth flow



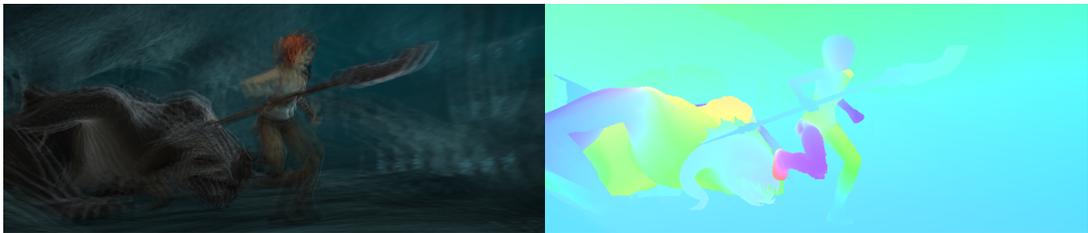
(c) Estimated rigid regions

(d) Estimated structure



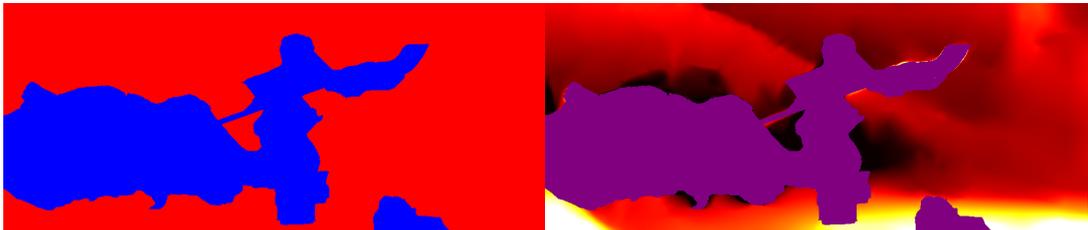
(e) Estimated flow

(f) Comparison to initial flow



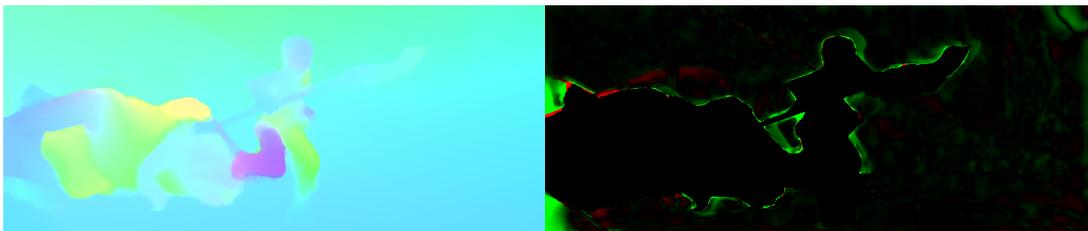
(a) Input images

(b) Ground truth flow



(c) Estimated rigid regions

(d) Estimated structure



(e) Estimated flow

(f) Comparison to initial flow



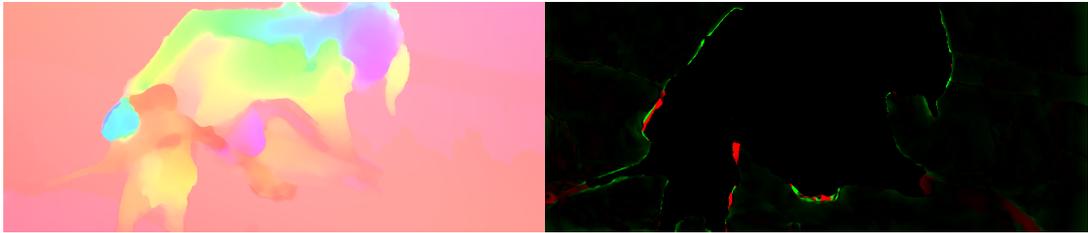
(a) Input images

(b) Ground truth flow



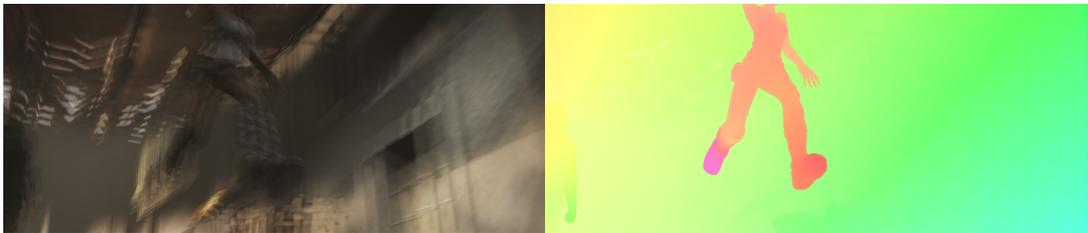
(c) Estimated rigid regions

(d) Estimated structure



(e) Estimated flow

(f) Comparison to initial flow



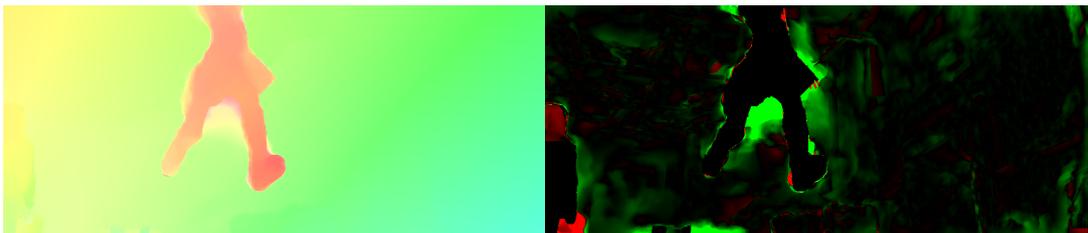
(a) Input images

(b) Ground truth flow



(c) Estimated rigid regions

(d) Estimated structure



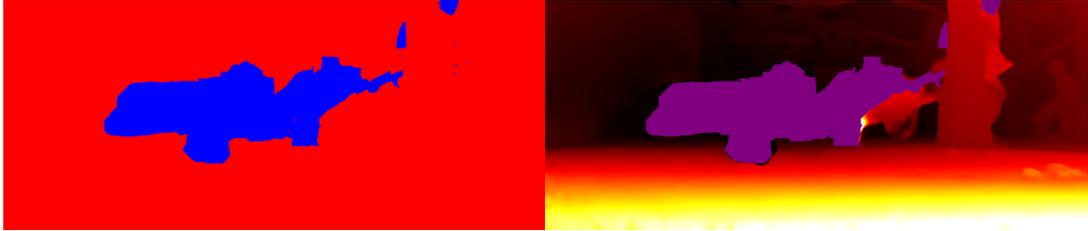
(e) Estimated flow

(f) Comparison to initial flow



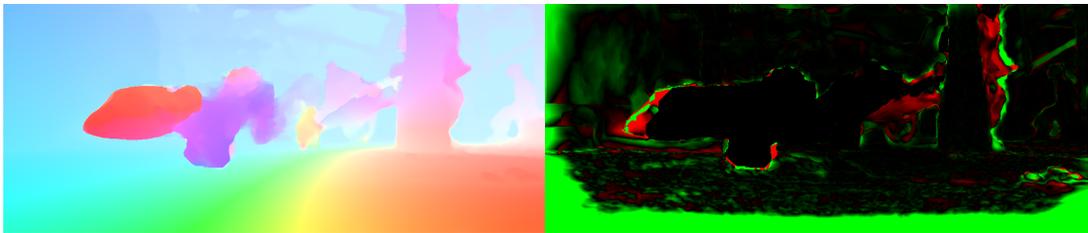
(a) Input images

(b) Ground truth flow



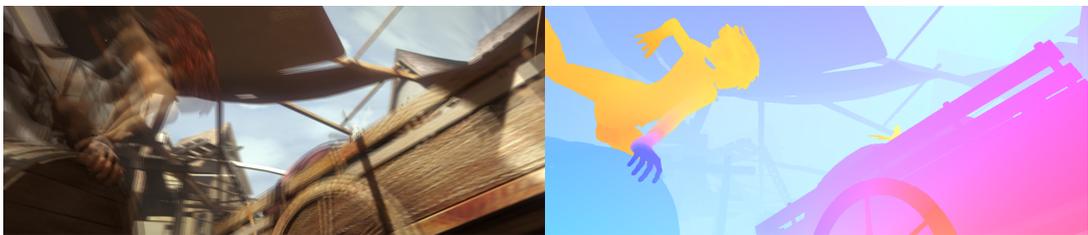
(c) Estimated rigid regions

(d) Estimated structure



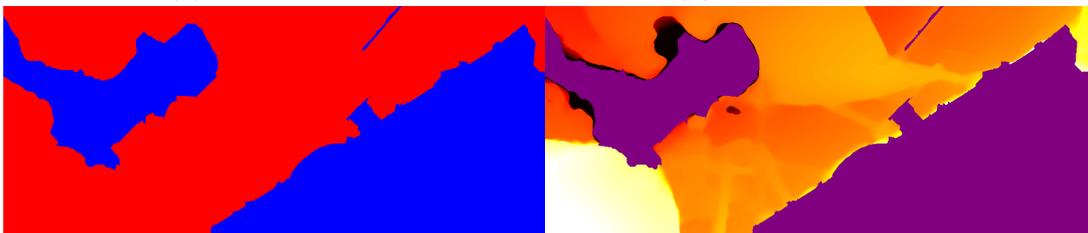
(e) Estimated flow

(f) Comparison to initial flow



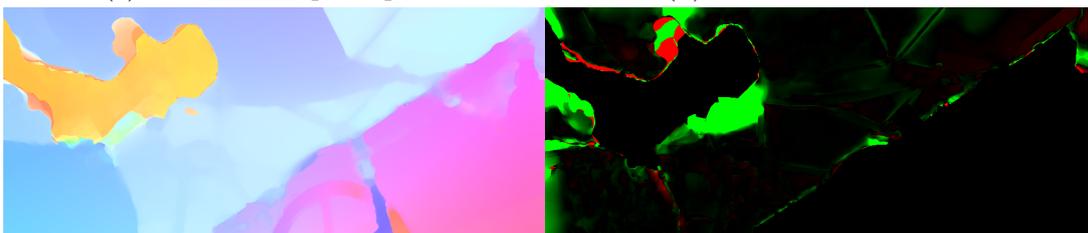
(a) Input images

(b) Ground truth flow



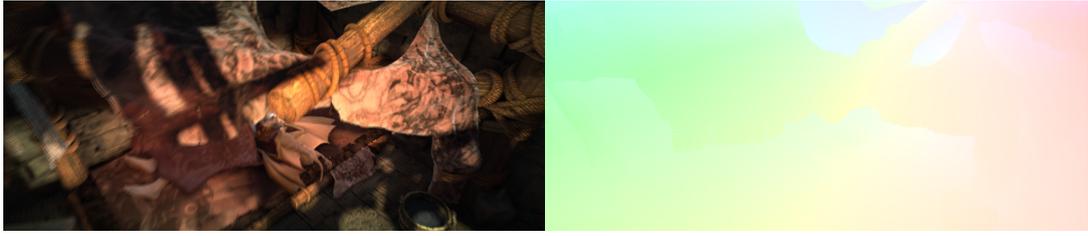
(c) Estimated rigid regions

(d) Estimated structure



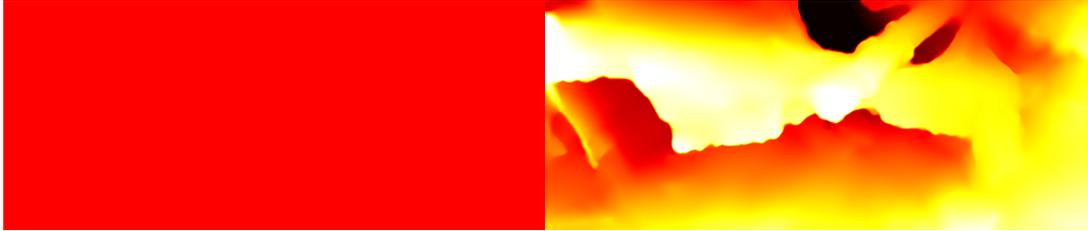
(e) Estimated flow

(f) Comparison to initial flow



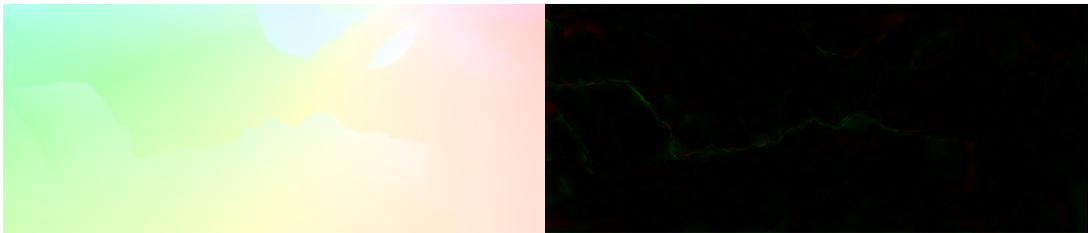
(a) Input images

(b) Ground truth flow



(c) Estimated rigid regions

(d) Estimated structure



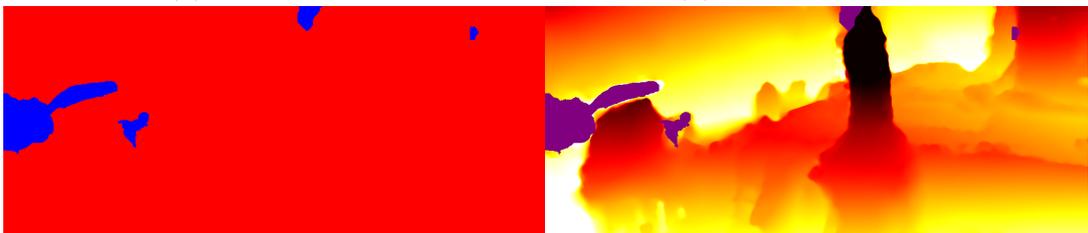
(e) Estimated flow

(f) Comparison to initial flow



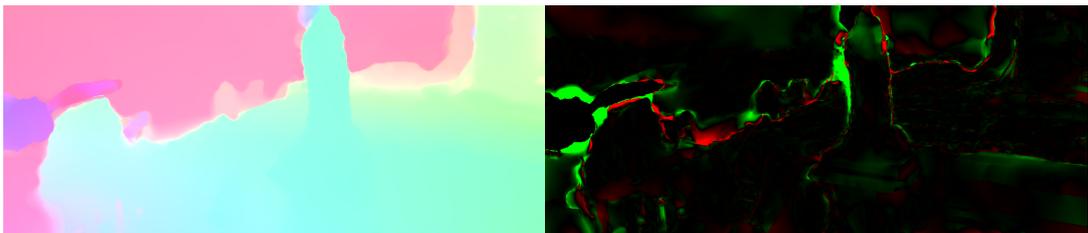
(a) Input images

(b) Ground truth flow



(c) Estimated rigid regions

(d) Estimated structure



(e) Estimated flow

(f) Comparison to initial flow

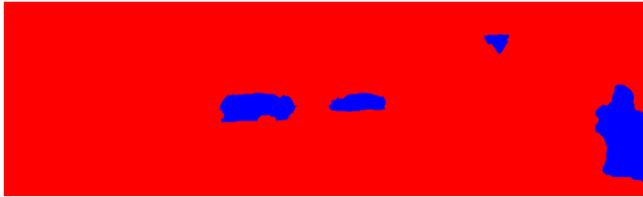
3.3 KITTI



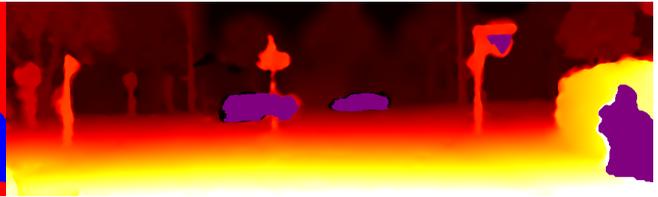
(a) Input images



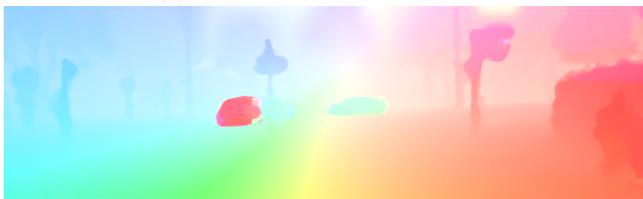
(b) Ground truth flow



(c) Estimated rigid regions



(d) Estimated structure



(e) Estimated flow



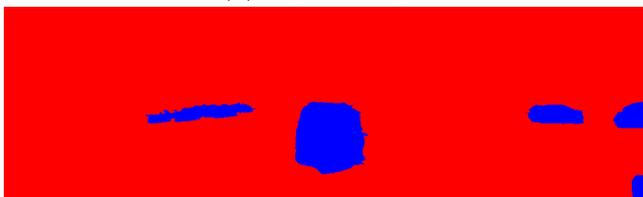
(f) Comparison to initial flow



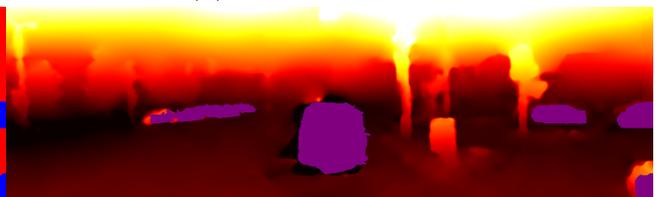
(a) Input images



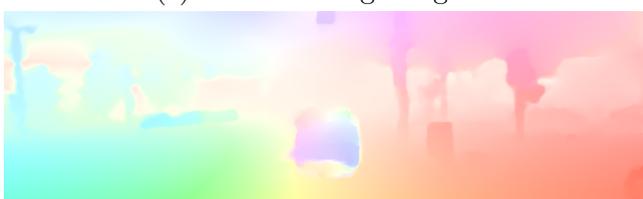
(b) Ground truth flow



(c) Estimated rigid regions



(d) Estimated structure



(e) Estimated flow



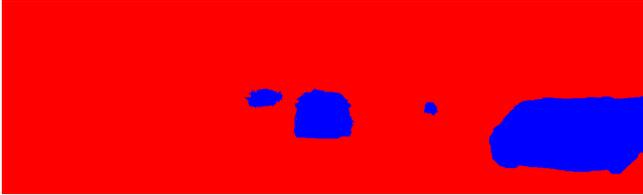
(f) Comparison to initial flow



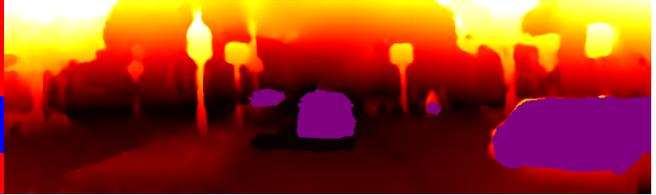
(a) Input images



(b) Ground truth flow



(c) Estimated rigid regions



(d) Estimated structure



(e) Estimated flow



(f) Comparison to initial flow



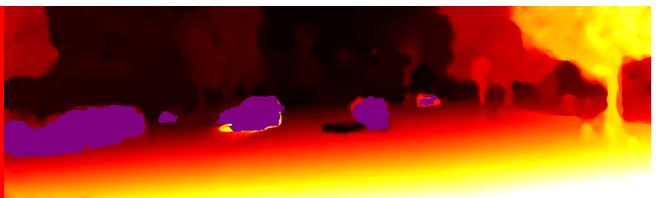
(a) Input images



(b) Ground truth flow



(c) Estimated rigid regions



(d) Estimated structure



(e) Estimated flow



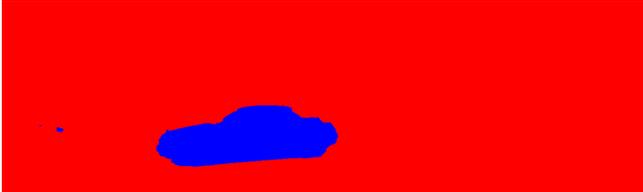
(f) Comparison to initial flow



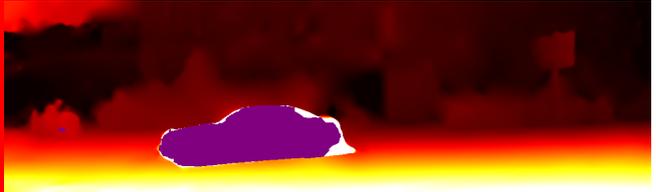
(a) Input images



(b) Ground truth flow



(c) Estimated rigid regions



(d) Estimated structure



(e) Estimated flow



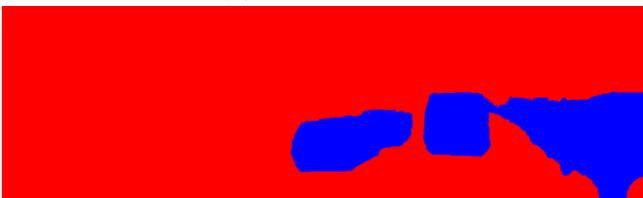
(f) Comparison to initial flow



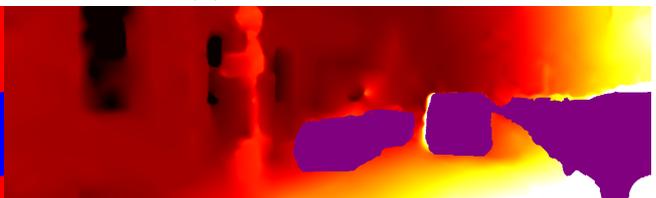
(a) Input images



(b) Ground truth flow



(c) Estimated rigid regions



(d) Estimated structure



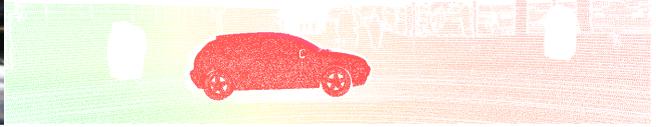
(e) Estimated flow



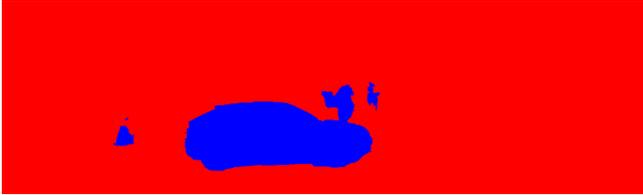
(f) Comparison to initial flow



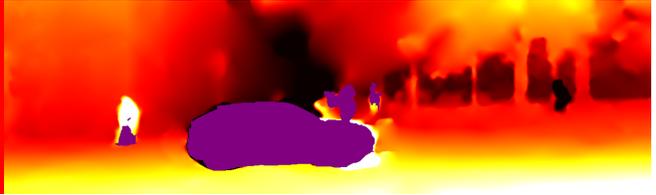
(a) Input images



(b) Ground truth flow



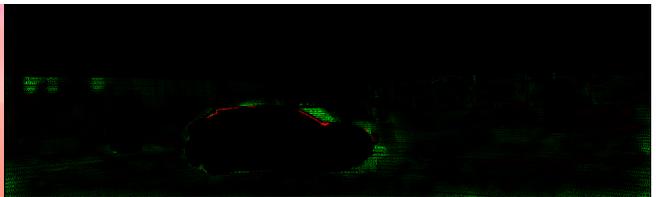
(c) Estimated rigid regions



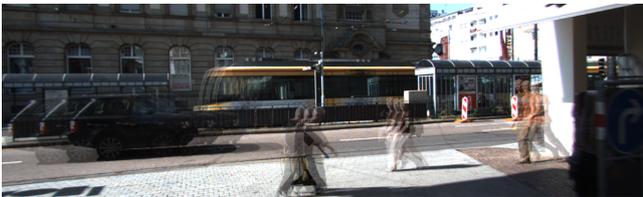
(d) Estimated structure



(e) Estimated flow



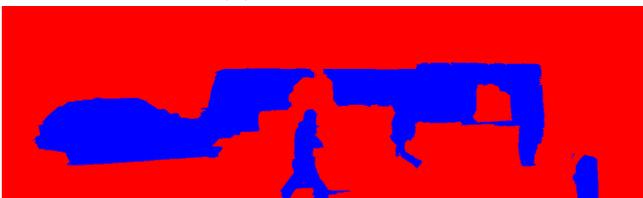
(f) Comparison to initial flow



(a) Input images



(b) Ground truth flow



(c) Estimated rigid regions



(d) Estimated structure



(e) Estimated flow



(f) Comparison to initial flow

4 Failure cases

This section gives examples of when our algorithm fails. We consider our algorithm to fail when the flow computed by our algorithm is worse than the initial flow [3]. For each of Sintel clean, Sintel final, and KITTI we give one failure example. All examples are among the worst overall in the respective training sets. In our training set, we observe two primary sources of error, segmentation failures and alignment failures.

Figures 36 and Figure 37 show examples for the first type of error, segmentation failures. In these cases, moving regions are mistaken as parts of the rigid background, such as the car in Fig. 36 or the girl’s head in Fig. 37. These failures occur if the CNN does not pick up a region strongly enough and if, at the same time, the motion of the object is consistent with the motion of the rigid scene. In Fig. 36, the CNN picks up only the frontal part of the car. Since in this example the camera is not moving, the focus of expansion is mistakenly determined by the few parts of the frame that move (*i.e.* the car), and the motion-based rigidity estimation cannot correct the mistake made by the CNN.

In Fig. 37, the camera pans to the left, and at the same time, the head moves to the right. Since both directions are approximately parallel, the head is considered to be rigid. Note how in this case the estimated flow has a very similar hue to the ground truth flow, even in most of the regions that are misclassified. This confirms that the direction of the flow is approximately consistent with the motion of the rigid parts of the scene; however, since the head still moves slightly out of the rigidity constraints, our method increases the error over the initialization.

Figure 38 shows the second type of error, a failure to align the images. As can be seen in

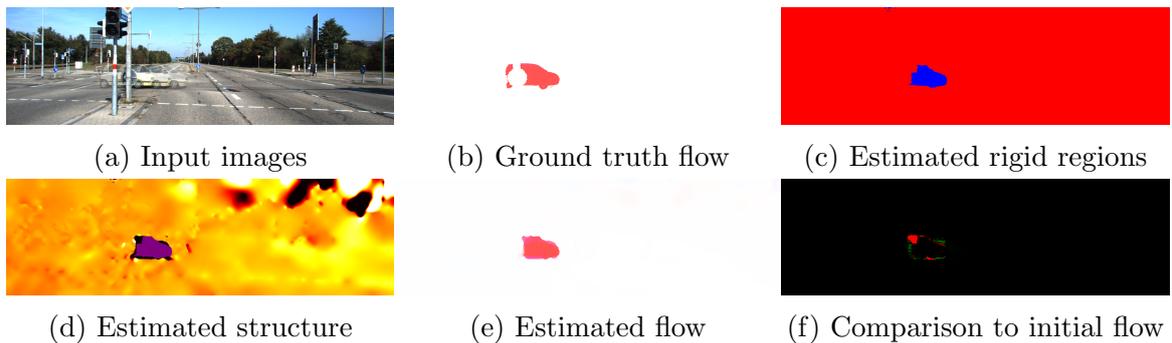


Figure 36: Failure case KITTI: The car on the left is wrongly detected as rigid.
 Perc. wrong initialization: 3.62%. Perc. wrong MR-Flow: 4.48%.

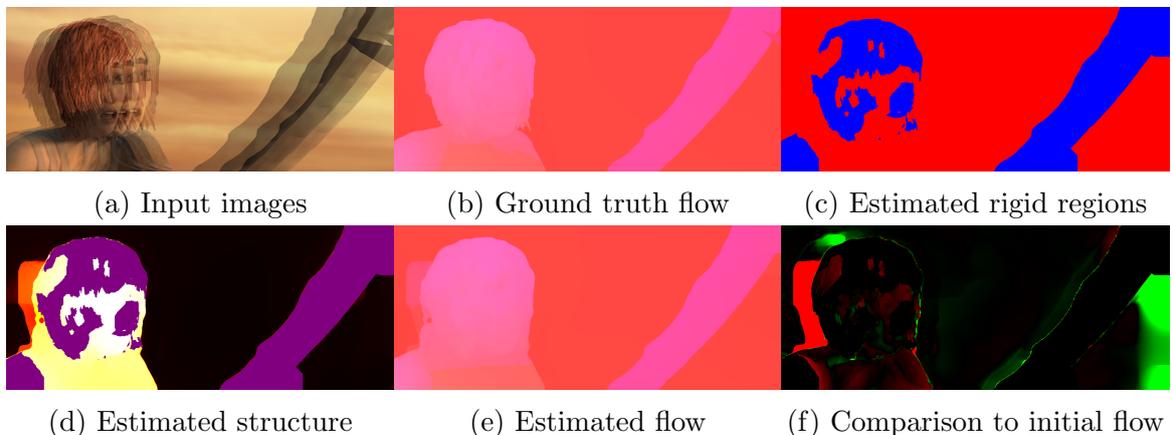


Figure 37: Failure case Sintel clean: Moving regions are wrongly detected as rigid.
 EPE initialization: 0.93. EPE MR-Flow: 1.59.

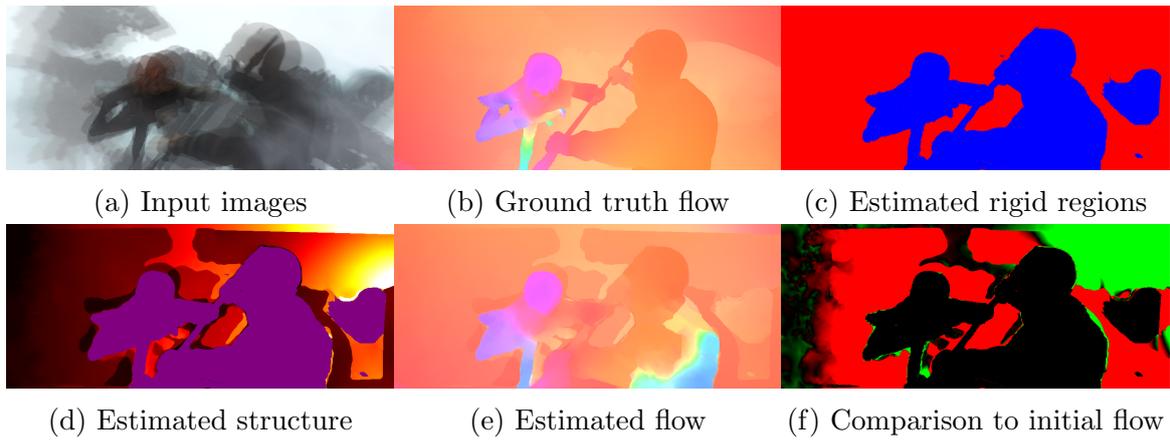


Figure 38: Failure case Sintel final: Strong motion blur destroys the alignment.
 EPE initialization: 11.17. EPE MR-Flow: 12.21.

Fig. 38(a), the background in this sequence contains heavy motion blur and a slight vignetting. Together, these two effects cause a high uncertainty of the initial optical flow in the background regions, which in turn causes our initial alignment procedure to fail.

References

- [1] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *CoRR*, abs/1412.7062, 2014. 2
- [2] P. Krähenbühl and V. Koltun. Efficient nonlocal regularization for optical flow. pages I:356–369, 2012. 2
- [3] M. Menze, C. Heipke, and A. Geiger. Discrete optimization for optical flow. In *German Conference on Pattern Recognition (GCPR)*, volume 9358, pages 16–28. Springer International Publishing, 2015. 24
- [4] L. Sevilla-Lara, D. Sun, V. Jampani, and M. J. Black. Optical flow with semantic segmentation and localized layers. In *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*, 2016. 2