

Optical Flow in Mostly Rigid Scenes

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

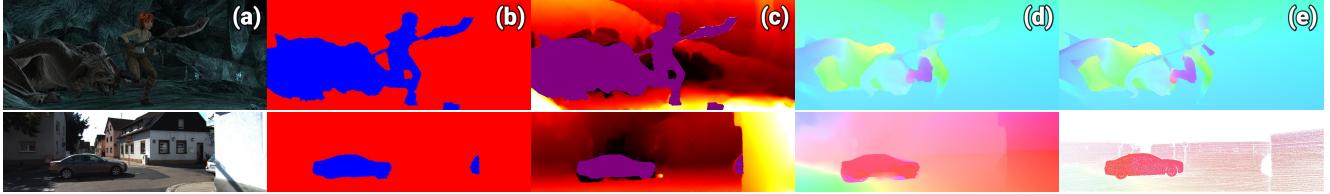


Figure 1: **Overview.** From three frames (a) our method computes a segmentation of the scene into static (red) and moving (blue) regions (b), the depth structure of the scene (c) , and the optical flow (d). (e) shows ground truth flow.

Abstract

The optical flow of natural scenes is a combination of the motion of the observer and the independent motion of objects. Existing algorithms typically focus on either recovering motion and structure under the assumption of a purely static world or optical flow for general unconstrained scenes. We combine these approaches in an optical flow algorithm that estimates an explicit segmentation of moving objects from appearance and physical constraints. In static regions we take advantage of strong constraints to jointly estimate the camera motion and the 3D structure of the scene over multiple frames. This allows us to also regularize the structure instead of the motion. Our formulation uses a Plane+Parallax framework, which works even under small baselines, and reduces the motion estimation to a one-dimensional search problem, resulting in more accurate estimation. In moving regions the flow is treated as unconstrained, and computed with an existing optical flow method. The resulting Mostly-Rigid Flow (MR-Flow) method achieves state-of-the-art results on both the MPI-Sintel and KITTI-2015 benchmarks.

1. Introduction

The world is composed of things that move and things that do not. The 2D motion field, which is the projection of the 3D scene motion onto the image plane, arises from observer motion relative to the static scene and the independent motion of objects. A large body of work exists on estimating camera motion and scene structure in purely static scenes,

generally referred to as Structure-from-Motion (SfM). On the other hand, methods that estimate general 2D image motion, or optical flow, make much weaker assumptions about the scene. Neither approach fully exploits the mixed structure of natural scenes. Most of what we see in such scenes is static - houses, roads, desks, etc.¹ Here, we refer to these static parts of the scene as the *rigid scene*, or *rigid regions*. At the same time, moving objects like people, cars, and animals make up a small but often important part of natural scenes. Despite the long history of both SfM and optical flow, no state-of-the art optical flow method synthesizes both into an algorithm that works on general scenes like those in the MPI-Sintel dataset [9] (Fig. 1). In this work, we propose such a method to estimate optical flow in video sequences of generic scenes that contain moving objects within a rigid scene.

For the rigid scene, the camera motion and depth structure fully determine the motion, which forms the basis of SfM methods. Modern optical flow benchmarks, however, are full of moving objects such as cars or bicycles in KITTI, or humans and dragons in Sintel. Assuming a fully static scene or treating these moving objects as outliers is hence not viable for optical flow algorithms; we want to reconstruct flow everywhere.

Independent motion in a scene typically arises from well defined objects with the ability to move. This points to a possible solution. Recently, convolutional neural networks (CNN) have achieved good performance on detecting and segmenting objects in images, and have been successfully

¹In KITTI-2015 and MPI-Sintel, independently moving regions make up only 15% and 28% of the pixels, respectively.

incorporated into optical flow methods [4, 34]. Here we take a slightly different approach. We modify a common CNN and train it on novel data to obtain a rigidity score from the labels, taking into account that some objects (*e.g.* humans) are more likely to move than others (*e.g.* houses). This score is combined with additional motion cues to obtain an estimate of rigid and independently moving regions.

After partitioning the scene into rigid and moving regions, we can deal with each appropriately. Since the motion of moving objects can be almost arbitrary, it is best computed using a classical unconstrained flow method. The flow of the rigid scene, on the other hand, is extremely restricted, and only depends on the depth structure and the camera motion and calibration. In theory, one could use an existing SfM algorithm to reconstruct the camera motion and the 3D structure of the scene, and project this structure back to obtain the motion of the rigid scene regions. Two factors make this hard in practice. First, the number of frames usually considered in optical flow is small; most methods only work on two or three consecutive frames. SfM algorithms, on the other hand, require tens or hundreds of frames to work reliably. Second, SfM algorithms require large camera baselines in order to reliably estimate the fundamental matrices. In video sequences, large baselines are rare, since the camera usually translates very little between frames. An exception to this are automotive scenarios such as the KITTI benchmark, where the recording car often moves rapidly and the frame rate is low.

Since full SfM is unreliable in general flow scenarios, we adopt the *Plane+Parallax* (P+P) framework [17, 18, 32]. In this framework, frames are registered to a common plane, which is aligned in all images after the registration. This removes the motion caused by camera rotation and simple intrinsic camera parameter changes, leaving parallax as the sole source of motion. Since all parallax is oriented towards or away from a common focus of expansion in the frame, computing the parallax is reduced to a 1D search problem and therefore easier than computing the full optical flow.

Here we show that using the P+P framework brings an additional advantage: the parallax can be factored into a structure component, which is independent of the camera motion and constant across time, and a temporally varying camera component, which is a *single number per frame*. We integrate the structure information across time; by definition, the structure of the rigid scene does not change. By combining the structure information from multiple frames, our algorithm generates a better structure component for all frames, and fills in areas that are unmatched in a single pair of frames due to occlusion.

Additionally, the relationship between the structure component and the parallax (and thus, the optical flow) enables us to regularize the flow in a physically meaningful way, since regularizing the structure implicitly regularizes the

flow. We use a robust second-order regularizer, which corresponds to a locally planar prior.

We integrate the regularization into a novel objective function measuring the photometric error across three frames as a function of the structure and camera motion. This allows us to optimize the structure and also to recover from poor initializations. We call the method MR-Flow for *Mostly-Rigid Flow* and show an overview in Fig. 2.

We test MR-Flow on MPI-Sintel [9] and KITTI 2015 [25] (Fig. 1). Among published monocular methods, at time of writing, we achieve the lowest error on MPI-Sintel on both passes; on KITTI-2015, our accuracy is second only to [4], a method specifically designed for automotive scenarios. Our code, the trained CNN, and all data is available at [1].

In summary, we present three main contributions. First, we show how to segment the scene into rigid regions and independently moving objects, allowing us to estimate the motion of each type of region appropriately. Second, we extend previous plane+parallax methods to express the flow in the rigid regions via its depth structure. This allows us to regularize this structure instead of the flow field and to combine information across more than two frames. Third, we formulate the motion of the rigid regions as a single model. This allows us to iterate between estimating the structure and to recover from unstable initializations.

2. Previous work

SfM and optical flow have both made significant, but mostly independent, progress. Roughly speaking, SfM methods require purely rigid scenes and use sparse point matches, wide baselines between frames, solve for accurate camera intrinsics and extrinsics, and exploit bundle adjustment to optimize over many views at once. In contrast, optical flow is applied to scenes containing generic motion, exploits continuous optimization, makes weak assumptions about the scene (*e.g.* that it is piecewise smooth), and typically processes only pairs of video frames at a time.

Combining optical flow and SfM. There have been many attempts to combine SfM and flow methods, dating to the 80’s [12]. For video sequences from narrow-focal-length lenses, the estimation of the camera motion is challenging, as it is easy to confuse translation with rotation and difficult to estimate the camera intrinsics [13].

More recently there have been attempts to combine SfM and optical flow [4, 28, 37, 39, 40]. The top monocular optical flow method on the KITTI-2012 benchmark estimates the fundamental matrix and computes flow along the epipolar lines [40]. This approach is limited to fully rigid scenes. Wedel *et al.* [39] compute the fundamental matrix and regularize optical flow to lie along the epipolar lines. If they detect independent motion, they revert to standard optical flow for the entire frame. In contrast, we segment static

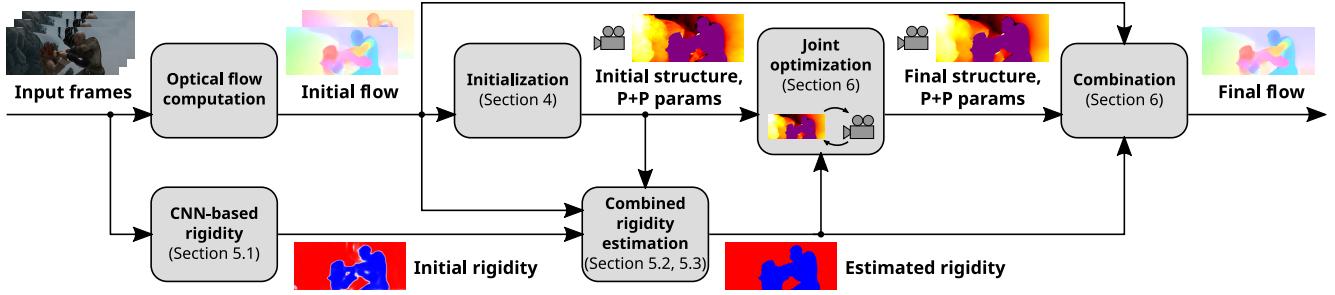


Figure 2: Algorithm overview. Given a triplet of frames, we first compute initial flow and an initial rigidity estimate based on a semantic segmentation CNN. The images are then aligned to a common plane, and the initial flow is converted to an estimate of the structure in the rigid scene using the Plane+Parallax framework. Where the P+P constraints are violated, the rigidity is refined, while at the same time the structure is refined using a variational optimization. To obtain the final flow estimate, the initial flow is used in moving regions, while the refined structure induces the flow in the rigid scene.

from moving *regions* and use appropriate constraints within each type of region. Roussos *et al.* [31] assume a known calibrated camera and solve for depth, motion and segmentation of a scene with moving objects. They perform batch processing on sequences of about 30 frames in length, making this more akin to SfM methods. While they have impressive results, they consider relatively simple scenes and do not evaluate flow accuracy on standard benchmarks.

Plane+Parallax. P+P methods were developed in the mid-90’s [17, 32]. The main idea is that stabilizing two frames with a planar motion (homography) removes the camera rotation and simplifies the geometric reasoning about structure [19, 36]. In the stabilized pair, motion is always oriented towards or away from the epipole and corresponds to parallax, which is related to the distance of the point from the plane in the 3D scene.

Estimating a planar homography can be done robustly and with more stability than estimating the fundamental matrix [18, 19]. While one is not able to estimate metric depth, the planar stabilization simplifies the matching process, turning the 2D optical flow estimation problem into a 1D problem that is equivalent to stereo estimation. Given the practical benefits, one may ask why P+P methods are not more prevalent in the leader boards of optical flow benchmarks. The problem is that such methods work only for rigid scenes. Making the P+P approach usable in general natural scenes is one of our main contributions.

Moving region segmentation. There have been several attempts to segment moving scenes into regions corresponding to independently moving objects by exploiting 3D motion cues and epipolar motion [2, 35, 38]. Several methods use the P+P framework to detect independent motions, but those methods typically only do detection and not flow estimation, and are often applied to simple scenes where there is a dominant motion like the ground plane and small moving objects [15, 33, 41]. Irani *et al.* [16] develop mosaic representations that include independently moving objects

but do not explicitly compute their flow. Given two frames as input, Ranftl *et al.* [29] segment a general moving scene into piecewise-rigid components and reason about the depth and occlusion relationships. While they produce impressive depth estimates, they rely on accurate flow estimates between the frames and do not refine the flow itself.

Combining multiple flow methods. There is also existing work on combining motion estimates from different algorithms into a single estimate [22, 23], but these do not attempt to fuse rigid and general motion. Bergen *et al.* [6] define a framework for describing optical flow problems using different constraints from rigid motion to generic flow, but do not combine these models into a single method.

Recent work combines segmentation and flow. Sevilla *et al.* [34] perform semantic segmentation and use different models for different semantic classes. Unlike them, we use semantic segmentation to estimate the rigid scene and then impose stronger geometric constraints in these regions. Hur and Roth [14] integrate semantic segmentation over time, leading to more accurate flow estimation for objects and better segmentation performance.

Most similar to our approach is [4], which first segments the scene into objects using a CNN. A fundamental matrix is then computed and used to constrain the flow within each object. Our work is different in a number of important ways. (i) Their approach is sequential and cannot recover from an incorrect fundamental matrix estimate. We propose a unified objective function where the parts of the solution inform and improve each other. (ii) [4] relies exclusively on the CNN to segment moving regions. While this works in specific scenarios such as automotive, it may not generalize to new scenes. We combine semantic segmentation and motion to classify rigid regions and thus require less accurate semantic rigidity estimates. This makes our algorithm both more robust and more general, as demonstrated by the fact that in contrast to [4] we evaluate on the challenging MPI-Sintel benchmark. (iii) [4] requires moving objects

to be rigid (*i.e.*, rigidly moving vehicles) and assumes a small rotational component of the egomotion. This works for KITTI-2015 but does not apply to more general scenes. (iv) [4] uses only two frames at a time and extrapolates into occlusions. Our model combines information across time, and thus it is able to compute accurate flow in occlusions.

3. Plane + Parallax background

The P+P paradigm has been used in rigid scene analysis for a long time. Since it forms the foundation of our algorithm, we briefly review the parts that are important for this work and refer the reader to [18, 32] for more details.

The core idea of P+P is to align two or more images to a common plane Π , so that

$$\mathbf{x} = \langle H\mathbf{x}'_h \rangle \quad \forall (\mathbf{x}, \mathbf{x}') \text{ on } \Pi \quad (1)$$

where \mathbf{x} and \mathbf{x}' represent a point in the reference frame and the corresponding point in another frame of the sequence, \mathbf{x}_h denotes \mathbf{x} in homogeneous coordinates, H is the homography mapping the image of Π between frames, and $\langle \mathbf{a} \rangle = (a_1/a_3, a_2/a_3)$ is the perspective normalization.

This alignment removes the effects of camera rotation and the effect of camera calibration change (such as a zoom) between the pair of frames [42]. Getting rid of rotation is especially convenient, since the ambiguity between rotation and translation in case of small displacements is a major source of numerical instabilities in the estimation of the structure of the scene.

When computing optical flow between aligned images, the flow of the pixels corresponding to points on the plane is zero². For an image point \mathbf{x} corresponding to a 3D point \mathbf{X} off the plane, the residual motion is given as [32]

$$\mathbf{u}_p(\mathbf{x}) = \frac{1}{1 - \frac{d(C_2)}{T_z} \frac{z}{d(\mathbf{X})}} (\mathbf{e} - \mathbf{x}), \quad (2)$$

where $d(C_2)$ is the distance of the second camera center to Π , z is the distance of point \mathbf{X} to the first camera, T_z is the depth displacement of the second camera, $d(\mathbf{X})$ is the distance from point \mathbf{X} to Π , and \mathbf{e} is the common focus of expansion that coincides with the epipole corresponding to the second camera. This representation has two main advantages. First, instead of an arbitrary 2D vector, each flow is confined to a line; therefore computing the optical flow is reduced to a 1D search problem. Second, when considering the flow of a pixel to different frames t which are registered to the same plane, Eq. (2) can be written as

$$\mathbf{u}_p(\mathbf{x}, t) = \frac{A(x)b_t}{A(x)b_t - 1} (\mathbf{e}_t - \mathbf{x}), \quad (3)$$

²Note that the plane does not have to correspond to a physical surface, but merely to a rigid, “virtual” plane.

where $A(x) = d(\mathbf{X})/z$ is the structural component of the flow field, which is *independent of t*. It is hence convenient to accumulate structure over time via A . $b_t = T_z/d(C_2)$, on the other hand, encodes the camera motion to frame t , and is a single number per frame. To simplify notation, we express the residual flow in terms of the *parallax field* $w(\mathbf{x}, t)$, so that

$$\mathbf{u}_p(\mathbf{x}) = w(\mathbf{x}, t) \frac{\mathbf{q}}{\|\mathbf{q}\|}, \quad w(\mathbf{x}, t) = \frac{A(x)b_t \|\mathbf{q}\|}{A(x)b_t - 1}, \quad (4)$$

with $\mathbf{q} = (\mathbf{e} - \mathbf{x})$. Here, w denotes the flow in pixels along the line towards \mathbf{e} .

We can thus parametrize the motion across multiple frames as a common structure component A and per-frame parameters $\theta_t = \{H_t, b_t, \mathbf{e}_t\}$. Since we use the center frame of a triplet of frames as the reference and compute the motion to the two adjacent frames, from here on we denote the two parameter sets as $\theta^+ = \{H^+, b^+, \mathbf{e}^+\}$ for the forward direction and θ^- for the backward direction.

4. Initialization

Given a triplet of images and a coarse, image-based rigidity estimation (described in Sec. 5.1), the goal of our algorithm is to compute (i) a segmentation into rigid regions and moving objects and (ii) optical flow for the full frame. We start by computing initial motion estimates using an existing optical flow method [26]. For a triplet of images $\{I^-, I, I^+\}$, we compute four initial flow fields, \mathbf{u}_0^+ from I to I^+ and \mathbf{u}_0^- from I to I^- , and their respective backwards flows $\bar{\mathbf{u}}_0^+$ and $\bar{\mathbf{u}}_0^-$. Due to the non-convex nature of our model (see Sec. 6) we need to compute good initial estimates for the P+P parameters $\hat{\theta}^+, \hat{\theta}^-, V^+, V^-$ denoting which pixels are visible and which are occluded in forward and backward directions, and an initial structure estimate \hat{A} .

Initial alignment and epipole detection. First we compute the planar alignments (homographies) between frames. Since P+P only holds in the rigid scene, in this section we only consider points that are marked as rigid by the initial semantic rigidity estimation. While computing a homography between two frames is usually easy, two factors make it challenging in our case: (i) when aligning multiple frames, the plane to which the frames are aligned has to be equivalent for each frame for P+P to work, and (ii) the 3D points corresponding to the four points used to estimate the homographies have to be coplanar for Eq. (3) to hold.

To compute homographies obeying these constraints, we use a two-stage process. First, we compute initial homographies \tilde{H}^+, \tilde{H}^- using RANSAC. In each iteration, the *same* random sample is used to fit both \tilde{H}^+, \tilde{H}^- , and a point is considered an inlier only when its reprojection error is low in both forward *and* backward directions. This ensures that the computed homographies belong to the same plane. If

a computed homography displaces the images corners by more than half the image size, it is considered invalid. If no valid homography is found, our method returns the initial flow field. This happens on average in 2% of the frames.

The second step is to ensure the coplanarity of the points inducing the homographies. For this, we can turn around Eq. (3), and simultaneously refine the homographies and estimate the epipoles $\mathbf{e}^{\{+,-\}}$ so that Eq. (3) holds. Let $\mathbf{u}_r = \langle H(\mathbf{x} + \mathbf{u}_0)_h \rangle - \mathbf{x}$ be the residual flow after registration with H . Each pair \mathbf{x}, \mathbf{u}_r defines a residual flow line, and in the noise-free case, the epipole \mathbf{e} is simply the intersection of these lines. Since the computed optical flow contains noise, we compute the epipole using the method described in [24], which we found to be sufficiently robust to noise. Therefore, \mathbf{e} is a function of the optical flow and of the computed homography. Enforcing coplanarity of the homographies is now equivalent to enforcing that the residual flow lines in both directions each pass through a common point as well as possible. The refined homographies are thus computed as

$$\hat{H}^+, \hat{H}^- = \arg \min_{H^+, H^-} \sum_{\mathbf{x}} \sum_{z \in \{+,-\}} \rho(o^z(\mathbf{x})), \quad (5)$$

with $o^z(\mathbf{x})$ defining the orthogonal distance of the residual flow line at \mathbf{x} to \mathbf{e}^z . While Eq. (5) is highly non-linear, we found that initializing with $\tilde{H}^{\{+,-\}}$ and using a standard non-linear minimization package such as LBFGS [27] produced results that greatly improved the final flow error compared to using the unrefined homographies $\tilde{H}^{\{+,-\}}$. Throughout the paper, we use the Lorentzian $\rho(x) = \sigma^2 \log(1 + x^2/\sigma^2)$ as the robust function, and compute the scaling parameter σ via the MAD [7]. The initial epipolar estimates $\hat{\mathbf{e}}^{\{+,-\}}$ are computed using $\hat{H}^{\{+,-\}}$.

To initialize b^+, b^- , we first compute the parallax fields by projecting \mathbf{u}_r onto the parallax flow lines,

$$w = \mathbf{u}_r^\top \mathbf{q} / \|\mathbf{q}\|. \quad (6)$$

Inserting (6) into (4) and solving for A , we get

$$A = w / (b(\|\mathbf{q}\| - w)). \quad (7)$$

Note that Eq. (3) contains a scale ambiguity between the structure A and the camera motion parameter b . Therefore, we can freely choose one of b^+, b^- , which only affects the scaling of A ; we choose \hat{b}^+ so that the initial forward structure A^+ defined by Eq. (7) has a MAD of 1. Since A^- is a function of b^- and should be as close as possible to A^+ , we obtain the estimate \hat{b}^- by solving

$$\hat{b}^- = \arg \min_{b^-} \sum_{\mathbf{x}} \rho \left(\hat{A}^+(\mathbf{x}) - A^-(\mathbf{x}) \right). \quad (8)$$

Using \hat{b}^- , we compute the initial backward structure \hat{A}^- using Eq. (7), and set the full sets of P+P parameters to $\theta^+ = \{\hat{H}^+, \hat{b}^+, \hat{\mathbf{e}}^+\}$, and θ^- accordingly.

Occlusion estimation. Pixels can become occluded in both directions. In occluded regions, we expect the flow to be wrong, since it can at best be extrapolated. Given the initial flow fields, we compute the visibility masks $V^+(\mathbf{x})$, $V^-(\mathbf{x})$ using a forward-backward check [20].

Initial structure estimation. Using the computed structure maps $\hat{A}^{\{+,-\}}$ and visibility maps $V^{\{+,-\}}$, the initial estimate for the full structure is

$$\hat{A}(\mathbf{x}) = \frac{1}{\max(1, V^+(\mathbf{x}) + V^-(\mathbf{x}))} \sum_{z \in \{+,-\}} V^z(\mathbf{x}) A^z(\mathbf{x}). \quad (9)$$

5. Rigidity estimation

Different cues provide different, complementary information about the rigidity of a region. The semantic category of an object tells us whether it is capable of independent motion, rigid scene parts have to obey the parallax constraint (3), and the 3D structure of rigid parts cannot change over time. We integrate all of them in a probabilistic framework to estimate a rigidity map of the scene, marking each pixel as belonging to the rigid scene or to a moving object.

5.1. Semantic rigidity estimation

We leverage the recent progress of CNNs for semantic segmentation to predict rigid and independently moving regions in the scene. In short, we model the relationship between an object’s appearance and its ability to move.

Obviously object appearance alone does not fully determine whether something is moving independently. A car may be moving, if driving, or static, if parked. However, for the purpose of motion estimation, not all errors are the same. Assuming an object is static when in reality it is not imposes false constraints that hurt the estimation of the global motion, while assuming a rigid region is independently moving does little harm. Thus, when in doubt, we predict a region to be independently moving.

The main optical flow benchmarks, KITTI-2015 and MPI-Sintel, provide different training data. While the essence of our model is the same for both, our training process varies to adapt to the available data. In both cases we start with the DeepLab architecture [10], pre-trained on the 21 classes of Pascal VOC [11], substitute all fully connected layers with convolutional layers, and densify the predictions [34]. Both networks produce a rigidity score between 0 and 1 which we call the semantic rigidity probability p_s .

MPI-Sintel contains many objects that are not contained in Pascal VOC, such as dragons. Thus using the CNN to predict a semantic segmentation is not possible. Also, no ground truth semantic segmentation is provided, so training a CNN to recognize these categories is not possible. However, the dataset provides ground truth camera calibration, depth and optical flow for the training set. With these we

estimate rigidity maps that we take as ground truth. We do this by computing a fully rigid motion field, using the depth and camera calibration, and comparing it with the ground truth flow field. Pixels are classified as independently moving if these two fields differ by more than a small amount. We make this data publicly available [1].

We modify the last layer of the CNN to predict 2 classes, rigid and independently moving, instead of the original 21. We train using the last 30 frames of each sequence in the training set, and validate on the first 5 frames of each sequence. Sequences shorter than 50 frames are only included in the validation set. At test time, the probability of being rigid is computed at each pixel and then thresholded. Examples of the estimated rigidity maps can be seen in Fig. 3.

In **KITTI 2015**, some independently moving objects (*e.g.* people) are masked out from the depth and flow ground truth. Therefore, the approach we followed for MPI-Sintel cannot be used. The objects in KITTI, however, appear in standard datasets like the enriched Pascal VOC. We modify the last layer of the network to predict the 22 classes that may be present in KITTI (*e.g.* person or road) similar to [34]. We then classify an object as moving if it has the ability to move independently (*e.g.* cars, or buses) and as rigid otherwise. Training details appear in the Sup. Mat. [1].

Note that the same approach we use for KITTI can be used for general video sequences by using a generic pre-trained semantic segmentation network together with a definition of which semantic classes can move and which are static. This allows our method to directly benefit from advances in semantic segmentation and novel, fine-grained semantic segmentation datasets.

5.2. Physical rigidity estimation

For objects that have not been seen previously or that exhibit phenomena like motion blur, the semantic rigidity may be wrong. Hence, we use two additional cues, motion direction and temporal consistency of the structure.

Moving regions from motion direction. A simple approach to classify a pixel as rigid or independently moving is to test whether its parallax flow points to the epipole [15]. Here, we employ a probabilistic framework for this classification. Due to space limitations, we just present the final result here; for the derivation, please see the Sup. Mat. [1].

For a given point \mathbf{x} , our model assumes the measured corresponding point $\mathbf{x}' = \mathbf{x} + \mathbf{u}_r$ to have a Gaussian error distribution around the true correspondence with covariance matrix $\Sigma = \sigma_d^2 \mathbf{I}$. Let $c = \|\mathbf{u}_r\|$ and α be the angle between \mathbf{u}_r and the line connecting \mathbf{x} to \mathbf{e} . Assuming a uniform distribution of motion directions for moving objects, the likelihood of a point being rigid is then given as

$$p(\mathbf{x} \text{ is rigid}) = \frac{\exp(-2t \sin^2(\alpha))}{\exp(-t) \mathbb{I}_0(t) + \exp(-2t \sin^2(\alpha))} \quad (10)$$

with $t = c^2/(4\sigma_d^2)$ and $\mathbb{I}_0(x)$ the modified Bessel function of the first kind. Solving for both forward and backward directions yields the direction-based rigidity probabilities p_d^+ and p_d^- . These are then combined into the final direction-based rigidity probability using the visibility maps

$$p_d = \begin{cases} \frac{1}{V^+ + V^-} \sum_{z \in +,-} V^z p_d^z & \text{if } V^- + V^+ > 0 \\ 1/2 & \text{otherwise.} \end{cases} \quad (11)$$

Moving regions from structure consistency. Another cue for rigidity is the temporal consistency of the structure. This is particularly helpful where semantics and motion direction cannot disambiguate the rigidity, for example when an object such as a car moves parallel to the observer's motion.

Recall that according to the P+P framework the structure of the rigid scene is independent of time. In rigid regions that are visible in all frames, we assume the forward and backward structure A^+ and A^- to be close to each other. A structure based rigidity estimate p_s can thus be computed as

$$p_s = \begin{cases} \exp(-(A^+ - A^-)^2 / \sigma_s^2) & \text{if } V^- V^+ = 1 \\ 1/2 & \text{otherwise.} \end{cases} \quad (12)$$

Combined rigidity probability from motion. The motion-based probabilities p_d , p_s can be seen as orthogonal. Surfaces that move independently along the parallax direction are considered to be rigid according to p_d , while surfaces that move by small amounts orthogonal to the parallax direction are considered to be rigid according to p_s . Hence, for a region to be considered *actually* rigid, we require both p_d and p_s to be high. The final motion-based rigidity probability p_m is

$$p_m = \begin{cases} p_d p_s & \text{if } V^+ V^- = 1 \\ (p_d + p_s)/2 & \text{otherwise.} \end{cases} \quad (13)$$

5.3. Combining rigidity estimates

The previously computed rigidity probabilities p_c , p_m yield per-pixel rigidity probabilities. To combine those into a coherent estimate, we first compute a rigidity unary

$$p_r = \lambda_{r,c} p_c + (1 - \lambda_{r,c}) p_m \quad (14)$$

and the corresponding energy

$$E_r(R, \mathbf{x}) = \begin{cases} -\log p_r(\mathbf{x}) & \text{if } R(\mathbf{x}) = 1 \\ -\log(1 - p_r(\mathbf{x})) & \text{otherwise,} \end{cases} \quad (15)$$

with $R(\mathbf{x}) = 1$ if \mathbf{x} is rigid, and 0 otherwise. Since we expect the rigidity to be spatially coherent, we estimate the full labelling by solving $\hat{R} =$

$$\arg \min_R \sum_{\mathbf{x}} E_r(R, \mathbf{x}) + \lambda_{r,p} \sum_{y \in \mathcal{N}(\mathbf{x})} w_{\mathbf{x},\mathbf{y}} [R(\mathbf{x}) \neq R(\mathbf{y})] \quad (16)$$

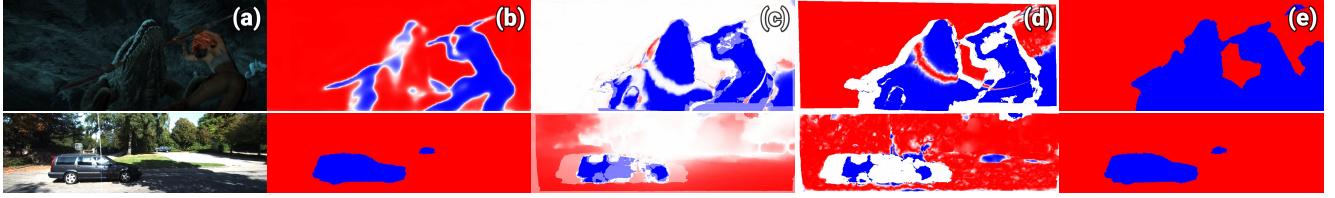


Figure 3: Results of rigidity estimation on the test sets of MPI-Sintel and KITTI-2015. From an image (a), we estimate a semantic rigidity (b) and combine it with the direction-based rigidity (c) and the structure-based rigidity (d) to obtain the final estimate (e). Likely rigid regions are red, likely moving regions are blue.

where $w_{\mathbf{x}, \mathbf{y}}$ is the image-based Potts modulation from [30] and $\mathcal{N}(\mathbf{x})$ is the 8-connected neighborhood of \mathbf{x} . Eq. (16) is solved using TRWS [3].

Figure 3 (top) shows the importance of combining different cues to recover from errors and accurately estimate the rigidity. The semantic estimation (b) misses a large part of the dragon’s head, while both the direction-based (b) and structure-based estimations misclassify different segments of the scene. Combining cues yields a good estimate (e).

6. Model and optimization

Model. The final structure should fulfill a number of criteria. First, as in the classical flow approach, warping the images using the flow induced by the structure should result in a low photometric error. Second, we assume that our initial flow fields are reasonable, hence, the final structure should be similar to the structures defined by the initial forward and backward flow. Third, the structure directly corresponds to the surface structure of the world, and thus we can regularize it using a locally planar model. This implicitly regularizes the flow in a more geometrically meaningful way than traditional priors on the flow.

Under these considerations, the full model for the motion of the rigid parts of the scene is defined as $E(A, \theta^+, \theta^-) =$

$$\sum_{\mathbf{x}} \hat{R}(\mathbf{x}) (E_d + \lambda_c E_c + \lambda_{1st} E_{1st} + \lambda_{2nd} E_{2nd}). \quad (17)$$

E_d is the photometric error, modulated by the estimated visibilities in forward and backward directions:

$$E_d = V^+(\mathbf{x}) \rho(I_a^+(s(\mathbf{x}, A, \theta^+)) - I_a(\mathbf{x})) + V^-(\mathbf{x}) \rho(I_a^-(s(\mathbf{x}, A, \theta^-)) - I_a(\mathbf{x})), \quad (18)$$

where I_a^-, I_a, I_a^+ are augmented versions of I^-, I, I^+ , i.e. stacked images containing the respective grayscale images and the gradients in x and y directions. The warping function $s(\mathbf{x}, A, \theta)$ defines the correspondence of \mathbf{x} according to the structure A and the P+P parameters θ ,

$$s(\mathbf{x}, A, \theta) = \left\langle H^{-1} \left(\mathbf{x} + \frac{A(\mathbf{x})b}{A(\mathbf{x})b - 1} (\mathbf{e} - \mathbf{x}) \right)_h \right\rangle. \quad (19)$$

The consistency term E_c encourages similarity between A and $A^{\{+, -\}}$.

$$E_c = V^+ \rho_c (A - A^+) + V^- \rho_c (A - A^-). \quad (20)$$

To ensure a constant error for all $A \in [A^-, A^+]$, we use the Charbonnier function as the robust penalty ρ_c .

The locally-planar regularization uses a 2nd order prior,

$$E_{2nd} = w_x \rho(\nabla_{xx} A) + w_y \rho(\nabla_{xy} A) + w_y \rho(\nabla_{yy} A). \quad (21)$$

Here, w_x, w_y are again the modulation terms from [30], and, using a slight abuse of notation, $\nabla_{xx}, \nabla_{xy}, \nabla_{yy}$ are the second derivative operators. Since the second order prior by itself is highly sensitive to noise, we add a first order prior

$$E_{1st} = w_x \rho(\nabla_x A) + w_y \rho(\nabla_y A), \quad (22)$$

where ∇_x, ∇_y are the first derivative operators in the horizontal and vertical direction respectively.

Optimization. To minimize the energy (17) we employ an iterative scheme, and alternate between optimizing for A with $\theta^{\{+, -\}}$ fixed, and for $\theta^{\{+, -\}}$ with A fixed. When optimizing A , we use a standard warping-based variational optimization [8] with 1 inner and 5 outer iterations and no downscaling. To optimize for θ , we first optimize for H, b using L-BFGS and then recompute e as described in Sec. 4. We use two iterations, since we found that more do not decrease the error significantly. This yields the final estimates $\bar{A}, \bar{\theta}^+, \bar{\theta}^-$ for the structure and the P+P parameters.

Due to the non-convex nature of (17), a global optimum is not guaranteed. However, in practice we found that our initializations are close to a good optimum, and hence our optimization procedure works well.

Final flow estimation. Finally, we convert the estimated structure \bar{A} into an optical flow field

$$\mathbf{u}_s(\mathbf{x}) = s(\mathbf{x}, \bar{A}, \bar{\theta}^+) - \mathbf{x}. \quad (23)$$

In the moving regions, we use the initial forward flow \mathbf{u}_0^+ , and compose the full flow field as

$$\mathbf{u}(\mathbf{x}) = \hat{R}(\mathbf{x}) \mathbf{u}_s + (1 - \hat{R}(\mathbf{x})) \mathbf{u}_0^+. \quad (24)$$

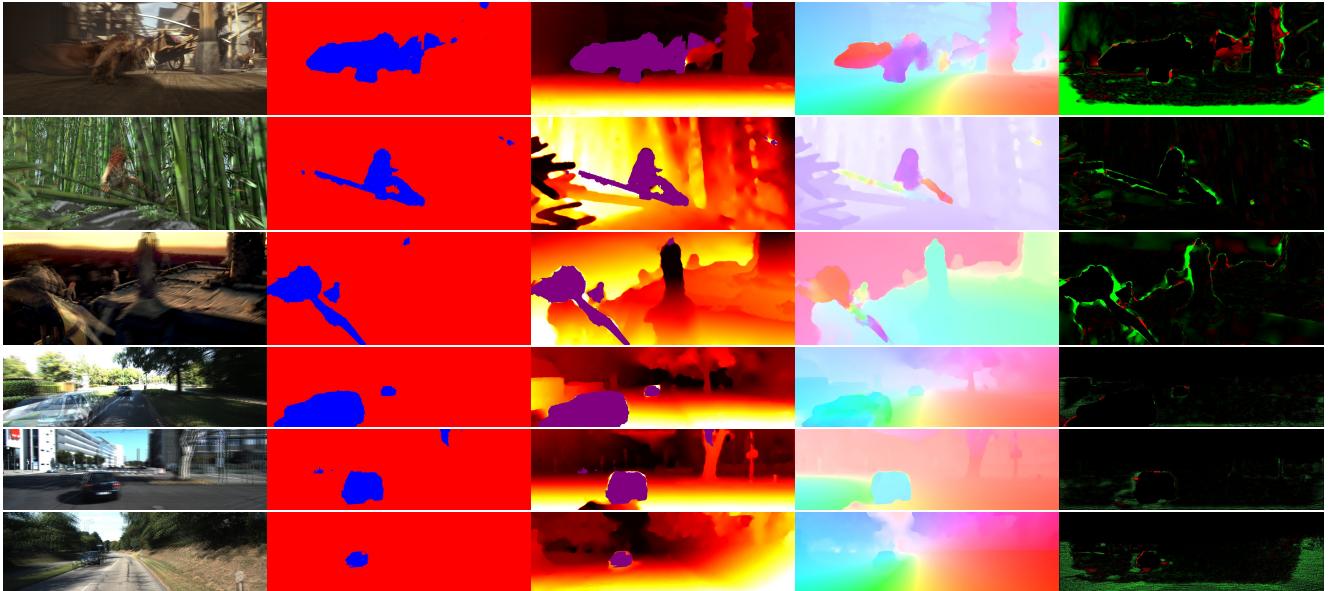


Figure 4: Results on MPI-Sintel and KITTI. From left to right: Overlaid input images, rigidity estimation, estimated structure (moving regions are masked in purple), estimated optical flow, comparison to initial flow (green areas denote improvements).

| | Sintel | | KITTI 2015 | | Train | Test | | |
|---------|-------------|-------------|-------------|-------------|---------------|---------------|--|--|
| | Clean | | Final | | | | | |
| | Train | Test | Train | Test | | | | |
| DF [26] | 1.96 | 3.57 | 3.80 | 6.08 | 23.09% | 21.57% | | |
| FF+ [5] | - | 3.10 | - | 5.71 | - | - | | |
| SDF [4] | - | - | - | - | 12.14% | 11.01% | | |
| MR-Flow | 1.83 | 2.53 | 3.59 | 5.38 | 14.09% | 12.19% | | |

Table 1: Errors on Sintel (EPE) and KITTI (%incorrect).

7. Experiments

To quantify our method, we evaluate on the MPI-Sintel and KITTI-2015 flow benchmarks. The parameters are chosen to minimize errors on the training sets, and are set to $\{\sigma_d, \sigma_s, \lambda_{r,c}, \lambda_{r,p}, \lambda_c, \lambda_{1st}, \lambda_{2nd}\} = \{0.75, 2.5, 0.1, 1.1, 0, 0.1, 5e3\}$ for Sintel and $\{1.0, 0.25, 0.5, 1.1, 0.01, 1, 5e4\}$ for KITTI. Table 1 shows the errors for our method, our initialization (DF), and for top performing methods on MPI-Sintel (FF+) [5] and KITTI-2015 (SDF) [4]. Both evaluate only on one dataset; in contrast, our method achieves high accuracy on both datasets. Figure 4 visualizes results; for more results see [1].

On **MPI-Sintel**, our method currently outperforms all published works. In particular, the structure estimation gives flow in occluded regions, producing the *lowest errors in the unmatched regions* of any published or unpublished work. On a 2.2 GHz i7 CPU, our method takes on average 2 minutes per triplet of frames without the initial flow computa-

tation, 74s for the initialization and rigidity estimation, and 46s for the optimization.

In **KITTI-2015** the scenes are simpler and contain only automotive situations; however, the images suffer from artifacts such as noise and overexposures. Among published monocular methods, MR-Flow is second after [4], which is designed for automotive scenarios and not tested on Sintel.

8. Conclusion

We have demonstrated an optical flow method that segments the scene and improves accuracy by exploiting rigid scene structure. We combine semantic and motion information to detect independently moving regions, and use an existing flow method to compute the motion of these regions. In rigid regions of the scene, the flow is directly constrained by the 3D structure of the world. This allows us to implicitly regularize the flow by constraining the underlying structure to a locally planar model. Furthermore, since the structure is temporally coherent, we combine information from multiple frames. We argue that this uses the right constraints in the right place and produces accurate flow in challenging situations and competitive results on Sintel and KITTI.

This opens several directions for future work. First, the rigidity estimation could be improved using better inference algorithms and training data. Jointly refining the foreground flow with the rigid flow estimation could improve performance. Our method could also use longer sequences, and enforce temporal consistency of the rigidity maps.

Acknowledgements. JW and LS were supported by the Max Planck ETH Center for Learning Systems.

References

- [1] <http://mrfow.is.tue.mpg.de>. 2, 6, 8
- [2] G. Adiv. Determining three-dimensional motion and structure from optical flow generated by several moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-7(4):384–401, July 1985. 3
- [3] K. Alahari, P. Kohli, and P. H. S. Torr. Reduce, reuse & recycle: Efficiently solving multi-label mrf's. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, June 2008. 7
- [4] M. Bai, W. Luo, K. Kundu, and R. Urtasun. Exploiting semantic information and deep matching for optical flow. In B. Leibe, J. Matas, N. Sebe, and M. Welling, editors, *Computer Vision – ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VI*, pages 154–170, Cham, 2016. Springer International Publishing. 2, 3, 4, 8
- [5] C. Bailer, B. Taetz, and D. Stricker. Flow fields: Dense correspondence fields for highly accurate large displacement optical flow estimation. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 4015–4023, Dec 2015. 8
- [6] J. Bergen, P. Anandan, K. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *Computer Vision ECCV'92*, volume LNCS 588, pages 237–252. Springer, 1992. 3
- [7] M. J. Black and G. Sapiro. Edges as outliers: Anisotropic smoothing using local image statistics. In M. Nielsen, P. Johansen, O. F. Olsen, and J. Weickert, editors, *Scale-Space Theories in Computer Vision: Second International Conference, Scale-Space'99 Corfu, Greece, September 26–27, 1999 Proceedings*, pages 259–270, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg. 5
- [8] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In T. Pajdla and J. Matas, editors, *Computer Vision - ECCV 2004: 8th European Conference on Computer Vision, Prague, Czech Republic, May 11–14, 2004. Proceedings, Part IV*, pages 25–36, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg. 7
- [9] D. Butler, J. Wulff, G. Stanley, and M. Black. A naturalistic open source movie for optical flow evaluation. In A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, editors, *Computer Vision - ECCV 2012*, volume 7577 of *Lecture Notes in Computer Science*, pages 611–625. Springer Berlin Heidelberg, 2012. 1, 2
- [10] 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. 5, 10
- [11] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88(2):303–338, jun 2010. 5
- [12] D. Heeger and A. Jepson. Subspace methods for recovering rigid motion I: Algorithm and implementation. *IJCV*, 7(2):95–117, 1992. 2
- [13] B. K. P. Horn and E. J. Weldon. Direct methods for recovering motion. *Int. Journal of Computer Vision*, 2(1):51–76, June 1988. 2
- [14] J. Hur and S. Roth. Joint optical flow and temporally consistent semantic segmentation. In G. Hua and H. Jégou, editors, *Computer Vision – ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8–10 and 15–16, 2016, Proceedings, Part I*, pages 163–177, Cham, 2016. Springer International Publishing. 3
- [15] M. Irani and P. Anandan. A unified approach to moving object detection in 2d and 3d scenes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(6):577–589, Jun 1998. 3, 6
- [16] M. Irani, P. Anandan, J. Bergen, R. Kumar, and S. Hsu. Efficient representations of video sequences and their applications. In *Signal Processing: Image Communication*, pages 327–351, 1996. 3
- [17] M. Irani, P. Anandan, and M. Cohen. Direct recovery of planar-parallax from multiple frames. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(11):1528–1534, Nov 2002. 2, 3
- [18] M. Irani, P. Anandan, and D. Weinshall. From reference frames to reference planes: Multi-view parallax geometry and applications. In H. Burkhardt and B. Neumann, editors, *Computer Vision – ECCV'98*, volume 1407 of *Lecture Notes in Computer Science*, pages 829–845. Springer Berlin Heidelberg, 1998. 2, 3, 4
- [19] M. Irani, B. Rousso, and S. Peleg. Recovery of ego-motion using region alignment. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(3):268–272, Mar. 1997. 3
- [20] Z. Kalal, K. Mikolajczyk, and J. Matas. Forward-backward error: Automatic detection of tracking failures. In *2010 20th International Conference on Pattern Recognition*, pages 2756–2759, Aug 2010. 5
- [21] P. Krähenbühl and V. Koltun. Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials. In *Advances in Neural Information Processing Systems*, pages 109–117, 2011. 10
- [22] V. Lempitsky, S. Roth, and C. Rother. Fusionflow: Discrete-continuous optimization for optical flow estimation. In *CVPR*, pages 1–8, June 2008. 3
- [23] O. Mac Aodha, A. Humayun, M. Pollefeys, and G. J. Brostow. Learning a confidence measure for optical flow. *PAMI*, 35(5):1107–1120, May 2013. 3
- [24] W. J. MacLean. Removal of translation bias when using subspace methods. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 2, pages 753–758 vol.2, 1999. 5
- [25] M. Menze and A. Geiger. Object scene flow for autonomous vehicles. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2015*, pages 3061–3070. IEEE, June 2015. 2
- [26] 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. 4, 8, 13

- [27] J. Nocedal. Updating quasi-newton matrices with limited storage. *Mathematics of computation*, 35(151):773–782, 1980. 5
- [28] L. Oisel, E. Memin, L. Morin, and C. Labit. Epipolar constrained motion estimation for reconstruction from video sequences. *Proc. SPIE*, 3309:460–468, 1998. 2
- [29] R. Ranftl, V. Vineet, Q. Chen, and V. Koltun. Dense monocular depth estimation in complex dynamic scenes. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 3
- [30] C. Rother, V. Kolmogorov, and A. Blake. "grabcut": Interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.*, 23(3):309–314, Aug. 2004. 7
- [31] A. Roussos, C. Russell, R. Garg, and L. Agapito. Dense multibody motion estimation and reconstruction from a handheld camera. In *IEEE Intl Symposium on Mixed and Augmented Reality (ISMAR 2012)*, 2012. 3
- [32] H. Sawhney. 3d geometry from planar parallax. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94., 1994 IEEE Computer Society Conference on*, pages 929–934, Jun 1994. 2, 3, 4
- [33] H. Sawhney, Y. Gao, and R. Kumar. Independent motion detection in 3D scenes. *PAMI*, 22(10):1191–1199, Oct. 1999. 3
- [34] 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, 3, 5, 6, 10
- [35] W. B. Thompson and T.-C. Pong. Detecting moving objects. *IJCV*, 4:39–57, 1990. 3
- [36] B. Triggs. Plane + parallax, tensors and factorization. In *European Conf. on Computer Vision (ECCV)*, volume LNCS 1842, pages 522–538. Springer, 2000. 3
- [37] L. Valgaerts, A. Bruhn, and J. Weickert. A variational model for the joint recovery of the fundamental matrix and the optical flow. In *DAGM*, 2008. 2
- [38] J. Weber and J. Malik. Rigid body segmentation and shape description from dense optical flow under weak perspective. *PAMI*, 19(2):139–143, Feb. 1997. 3
- [39] A. Wedel, D. Cremers, T. Pock, and H. Bischof. Structure-and motion-adaptive regularization for high accuracy optic flow. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1663–1668, Sept 2009. 2
- [40] K. Yamaguchi, D. McAllester, and R. Urtasun. Robust monocular epipolar flow estimation. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 1862–1869, June 2013. 2
- [41] C. Yuan, G. Medioni, J. Kang, and I. Cohen. Detecting motion regions in the presence of a strong parallax from a moving camera by multiview geometric constraints. *PAMI*, 29(9):1627–1641, Sept. 2007. 3
- [42] L. Zelnik-Manor and M. Irani. Multi-frame estimation of planar motion. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(10):1105–1116, Oct 2000. 4

A. Supplemental Material

A.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. 5 and Fig. 6 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 [34], since it was shown to be successful. We used the DeepLab architecture [10] 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 [21] 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, which is pretrained including extra annotations from the MS-COCO dataset³. 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%.

³Further details can be found on their webpage <http://ccvl.stat.ucla.edu/software/deeplab/deeplab-coco-largefov/>



Figure 5: 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).

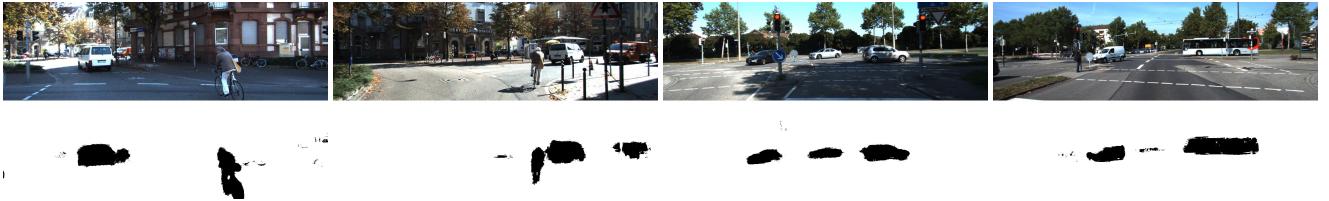


Figure 6: Results of rigidity estimation on KITTI 2015. Top: Original frame; Bottom: probability map of rigidity.

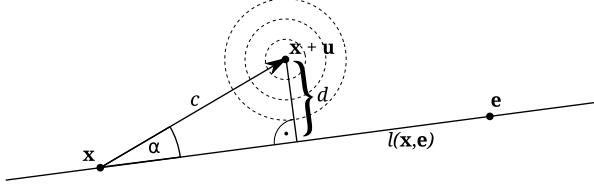


Figure 7: Illustration

A.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

$$p(\mathbf{x}'|r=1, \mathbf{e}) = \int_{y \in l(\mathbf{x}, \mathbf{e})} \frac{1}{2\pi\sigma_m^2} \exp\left(-\frac{1}{2}\mathbf{y}^T \Sigma^{-1} \mathbf{y}\right) dy, \quad (25)$$

where $l(\mathbf{x}, \mathbf{e})$ denotes the line that goes through \mathbf{x} and \mathbf{e} . Figure 7 shows an illustration.

Since the error distribution is Gaussian, every marginal is also Gaussian. Therefore, the line integral in (25) is given as

$$p(\mathbf{x}'|r=1, \mathbf{e}) = \frac{1}{\sqrt{2\pi\sigma_m^2}} \exp\left(-\frac{d^2}{2\sigma_m^2}\right), \quad (26)$$

where d denotes the distance of \mathbf{x}' to $l(\mathbf{x}, \mathbf{e})$, as shown in Figure 7.

In the following, it will be convenient to express the correspondence point \mathbf{x}' in (26) in terms of the angle α and the displacement magnitude $c = \|\mathbf{u}\|$. Then,

$$p(\alpha, c|r=1, \mathbf{e}) = \frac{1}{\sqrt{2\pi\sigma_m^2}} \exp\left(-\frac{c^2 \sin^2(\alpha)}{2\sigma_m^2}\right). \quad (27)$$

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

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

$$\begin{aligned} p(r=1|\alpha, c) &= \\ &= \frac{p(\alpha|r=1, c)p(r=1)}{p(\alpha|r=0, c)p(r=0) + p(\alpha|r=1, c)p(r=1)} \\ &= \frac{\frac{1}{Z} p(\alpha, c|r=1)p(r=1)}{p(\alpha|r=0, c)p(r=0) + \frac{1}{Z} p(\alpha, c|r=1)p(r=1)}. \end{aligned} \quad (28)$$

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

$$p(r=1|\alpha, c) = \frac{p_1 p(\alpha, c|r=1)}{\frac{p_1}{Z(1-p_1)} + p_1 p(\alpha, c|r=1)} \quad (29)$$

and for uninformative priors $p_1 = 0.5$

$$p(r = 1|\alpha, c) = \frac{p(\alpha, c|r = 1)}{\frac{Z}{2\pi} + p(\alpha, c|r = 1)}. \quad (30)$$

What remains is to compute Z . Using Eq. (27), it can be computed as

$$\begin{aligned} Z &= p(c|r = 1) = \int_0^{2\pi} p(\alpha, c|r = 1) d\alpha \\ &= \frac{1}{\sqrt{2\pi\sigma_m^2}} \int_0^{2\pi} \exp\left(-\frac{c^2 \sin^2(\alpha)}{2\sigma_m^2}\right) d\alpha \\ &\quad [t = \frac{c^2}{4\sigma_m^2}, \sin^2(x) = \frac{1}{2}(1 - \cos(2x))] \\ &= \frac{1}{\sqrt{2\pi\sigma_m^2}} \int_0^{2\pi} \exp(-t(1 - \cos(2\alpha))) d\alpha \\ &= \frac{1}{\sqrt{2\pi\sigma_m^2}} \exp(-t) \int_0^{2\pi} \exp(t \cos(2\alpha)) d\alpha \\ &\quad [\beta = 2\alpha] \\ &= \frac{1}{\sqrt{2\pi\sigma_m^2}} \exp(-t) \frac{1}{2} \int_0^{4\pi} \exp(t \cos(\beta)) d\beta \\ &= \frac{1}{\sqrt{2\pi\sigma_m^2}} \exp(-t) \int_0^{2\pi} \exp(t \cos(\beta)) d\beta \\ &= \frac{1}{\sqrt{2\pi\sigma_m^2}} \exp(-t) 2\pi I_0(t) \\ &= \frac{\sqrt{2\pi}}{\sigma_m} \exp(-t) I_0(t), \end{aligned} \quad (31)$$

with $I_0(x)$ the modified Bessel function of the first kind. Inserting Eq. (31) and Eq. (27) into Eq. 30 yields Eq. (11) from the main paper:

$$\begin{aligned} p(\mathbf{x} \text{ is rigid}) \\ &= p(r = 1|\alpha, c) \\ &= \frac{\exp(-2t \sin^2(\alpha))}{\exp(-t) I_0(t) + \exp(-2t \sin^2(\alpha))}. \end{aligned} \quad (32)$$

B. 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 result⁴. 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*.

⁴The 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.

- **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)⁵. 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}^- .

$$\hat{b}^- = \underset{\mathbf{x}}{\text{median}} \frac{1}{A^+(\mathbf{x})} \frac{w^-(\mathbf{x})}{\|\mathbf{e}^- - \mathbf{x}\| - w^-(\mathbf{x})} \quad (33)$$

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

Table 2 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.

Table 3 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.

Table 4 shows the impact of the different terms of the variational refinement (Eq. (17)). The cases are as described above. In addition, table 4 includes a complete omission of the variational refinement (*no-opt*), which uses only the

⁵All equation numbers refer to equations in the main paper.

| | 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 2: Errors when successively switching on parts of the algorithm

| | 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 3: Errors when disabling individual parts of the algorithm

| | 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 4: Influence of regularization terms

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.

C. 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 [26]. 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 8 and Figure 9 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. 8 or the girl’s head in Fig. 9. 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. 8, 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. 9, 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

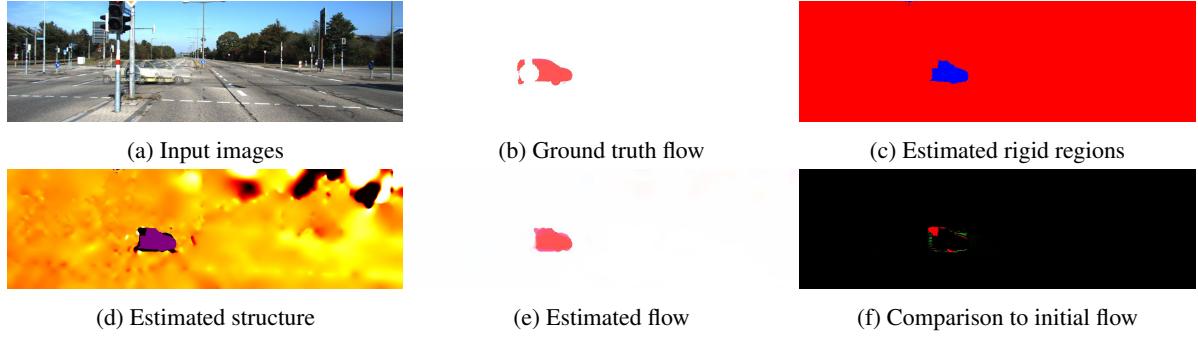


Figure 8: Failure case KITTI: The car on the left is wrongly detected as rigid.

Perc. wrong initialization: 3.62%. Perc. wrong MR-Flow: 4.48%.

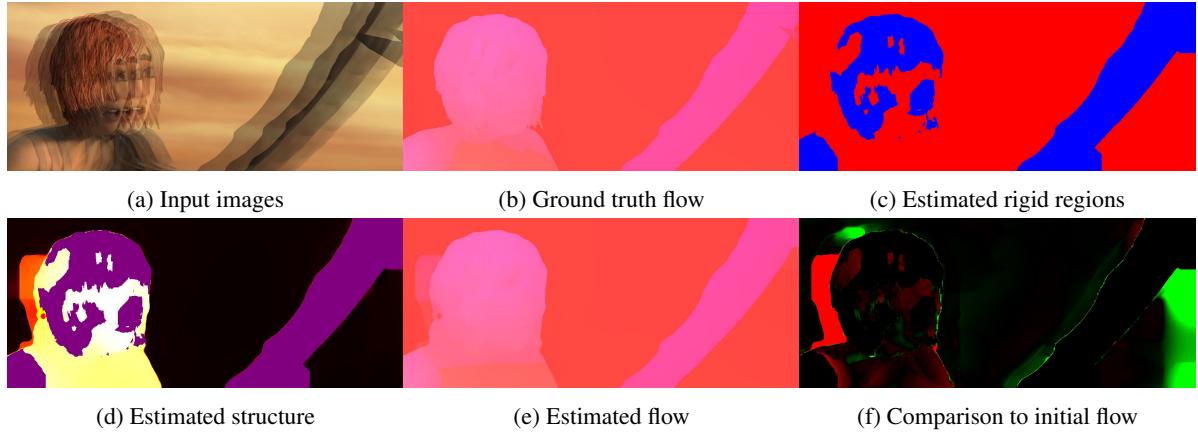


Figure 9: Failure case Sintel clean: Moving regions are wrongly detected as rigid.

EPE initialization: 0.93. EPE MR-Flow: 1.59.

the error over the initialization.

Figure 10 shows the second type of error, a failure to align the images. As can be seen in Fig. 10(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.

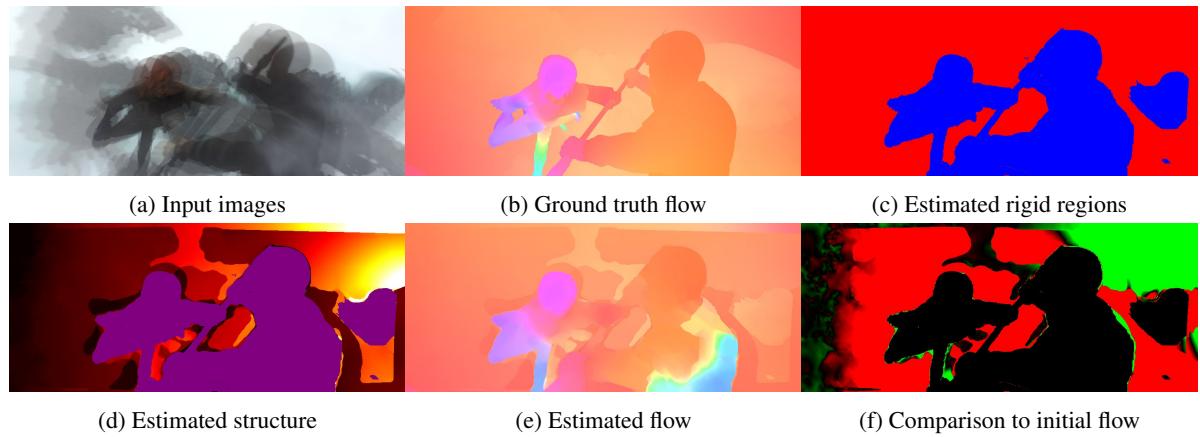


Figure 10: Failure case Sintel final: Strong motion blur destroys the alignment.
EPE initialization: 11.17. EPE MR-Flow: 12.21.