Monocular Piecewise Depth Estimation in Dynamic Scenes by Exploiting Superpixel Relations

Yan Di[1], Henrique Morimitsu[1], Shan Gao[2] and Xiangyang Ji[1]

[1]Department of Automation, Tsinghua University, Beijing, China
[2]Unmanned System Research Institute, Northwestern Polytechnical University, Xi'an, China

[*]{diy17@mails, hmorimitsu@mail}.tsinghua.edu.cn gaoshan@nwpu.edu.cn xyji@tsinghua.edu.cn

Abstract

In this paper, we propose a novel and specially designed method for piecewise dense monocular depth estimation in dynamic scenes. We utilize spatial relations between neighboring superpixels to solve the inherent relative scale ambiguity (RSA) problem and smooth the depth map. However, directly estimating spatial relations is an ill-posed problem. Our core idea is to predict spatial relations based on the corresponding motion relations. Given two or more consecutive frames, we first compute semi-dense (CPM) or dense (optical flow) point matches between temporally neighboring images. Then we develop our method in four main stages: superpixel relations analysis, motion selection, reconstruction, and refinement. The final refinement process helps to improve the quality of the reconstruction at pixel level. Our method does not require per-object segmentation, template priors or training sets, which ensures flexibility in various applications. Extensive experiments on both synthetic and real datasets demonstrate that our method robustly handles different dynamic situations and presents competitive results to the state-of-the-art methods while running much faster than them.

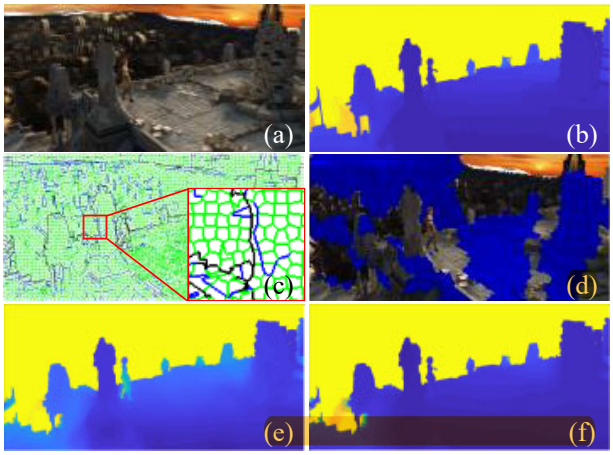


Figure 1. Results of our method for two-frame reconstruction. (a) is the input image pair. (b) is the ground truth depth map. (c) demonstrates the predicted spatial relations between neighboring superpixels. Green lines represent *coplanar*, blue lines represent *hinge*, black lines represent *crack*. (d) demonstrates the motion selection process. Reliable static superpixels (labelled blue) are used to identify camera motion. Furthermore, their scales are fixed and they are used as references to estimate the scales of remaining superpixels. (e) demonstrates the result of MVG [9]. (f) is the final result of our method.

1. Introduction

Dense monocular depth estimation in complex dynamic scenes has been a popular but challenging topic in computer vision for many years. It is widely adopted as an important step in many practical applications such as robot navigation [6], scene understanding [8], saliency detection [22], etc. However, real-world scenes usually consist of complex motion models, including rigid background, moving vehicles, non-rigid pedestrians and so on. Traditional structure-from-motion (SFM) methods [9, 23] fail to reconstruct moving objects due to the inherent RSA problem [16], as explained in Figure 2. Therefore, efficient and effective frameworks that can handle dynamic scenes are of great need.

Recently, two methods called DMDE [25] and S.Soup [16] achieved state-of-the-art performance on depth estimation on MPI Sintel [2] and KITTI datasets [6]. DMDE first applies motion segmentation to the optical flow field and then solves the RSA problem with an ordering constraint which captures the assumption that dynamic objects occlude the static environment. However, accurately and densely segmenting moving objects or parts is not easy in many cases (e.g. traffic scenes in KITTI). And low-

	MVG [9]	DT [15]	DMDE [25]	S.Soup [16]	Ours
Input frames	2	1	2	2	≥ 2
Requirements	OF	-	OF	OF	PM/OF
	-	Train	Seg	-	-
Robust to outliers	$\times$	$-$	$\surd$	$\surd$	$\surd$
Dynamic scenes	$\times$	$\surd$	$\surd$	$\surd$	$\surd$
Speed	$600s$	$100s$	$60s$	$660s$	$5s$

Table 1. Main features of our method compared to some baseline methods. DT is a single-frame method, so we don't evaluate its robustness to outliers. Abbreviation of requirements: OF - optical flow, Train - training set, Seg - motion segmentation, PM - semi-dense point matches. Our method is fast, robust and flexible in dynamic depth estimation.

Motion relations	Criterion
Coplanar	$\sum_{p\in S_i\cup S_j} \lvert H_iP - H_jP\rvert \approx 0$
Hinge	$\sum_{p\in B_{ij}} \lvert H_iP - H_jP\rvert \approx 0$
Crack	Otherwise

Spatial relations	Criterion
Coplanar	$\sum_{p\in S_i\cup S_j} \lvert \theta_iP - \theta_jP\rvert \approx 0$
Hinge	$\sum_{p\in B_{ij}} \lvert \theta_iP - \theta_jP\rvert \approx 0$
Crack	Otherwise

Table 2. Criteria of motion and spatial relations. P is the homogeneous form of pixel p. B_{ij} denotes pixels on the shared boundary of S_i and S_j. H_i and θ_i denote homography and plane parameter of S_i respectively.

quality motion segmentation directly leads to low-quality reconstruction, which deteriorates the robustness of DMDE. S.Soup [16] goes further by incorporating an ARAP (as rigid as possible) term in its energy function under the assumption that the transformation between two frames is locally piecewise-rigid and globally as rigid as possible. This term avoids motion segmentation and helps to solve the inherent RSA problem. However, the optimization of the non-convex ARAP term is time-consuming, which limits its practical applications.

Considering the drawbacks of DMDE and S.Soup, we propose a new framework that is efficient and robust in monocular depth estimation in dynamic scenes. By applying superpixel over-segmentation to the image, we model each superpixel as a small plane (parameterized with a plane parameter and a scale parameter) in 3D space. We exploit two kinds of relations between neighboring superpixels: motion relations and spatial relations, each having three subcategories $\{coplanar, hinge, crack\}$. Spatial relations provide constraints on the spatial position of each superpixel and thus can be used to solve the RSA problem. However, spatial relations cannot be estimated directly since the plane parameter of each superpixel is also unknown. We instead predict spatial relations between neighboring superpixels based on their motion relations which can be jointly estimated with homographies according to input point matches. We observe that in most dynamic cases, motion and spatial relations correspond one-to-one. Based on this observation and the widely-used piecewise planar assumption, we design a unified framework that consists of four main stages: superpixel relations analysis, motion selection, reconstruction, and refinement. We demonstrate that our method achieves state-of-the-art performance on several popular datasets [2, 5, 7]. The main features of our method compared to the baseline methods [9, 15, 16, 25] are summarized in Table 1, and the result of each stage is demonstrated in Figure 1.

Our contributions are summarized as follows: 1) We propose a unified framework for monocular piecewise reconstruction in complex dynamic scenes, which achieves state-of-the-art performance on various benchmarks. 2) We demonstrate how to solve the RSA problem in dynamic reconstruction based on the assumption that motion and spatial relations between neighboring superpixels generally correspond one-to-one. 3) We further introduce an approach to improve the depth estimation quality by tracking superpixel relations between temporally neighboring frames. Experiments show that this approach successfully leverage multiple frames to output more accurate results without a noticeable reduction in speed.

2. Related works

Non-rigid structure from motion. Many methods have been proposed to deal with non-rigid reconstruction. Our approach is most related to piecewise reconstruction methods [29, 30, 32]. They typically formulate the reconstruction problem as a multiple model fitting problem where point matches belong to an unknown number of models. The point matches are divided into overlapping groups and the matches covered by multiple groups are used to align these groups. The assignment of point matches to multiple local models and the fitting of models to points are estimated simultaneously by minimizing an elaborate energy function with geometric terms and other optional terms like appearance term, minimum description length (MDL) term and so on. Russell *et al.* [30] go one step further by introducing multi-level segmentation to improve the performance. However, [29, 32] are not able to reconstruct the entire scene and [30] performs poorly when applied to complex scenes (e.g. on the MPI Sintel dataset [2]).

Optical flow-based methods. DMDE [25] and S.Soup [16] are two typical methods of this kind. DMDE segments the optical flow field into a set of motion models and then optimizes an energy function to reconstruct the scale-ambiguous foreground together with the surrounding environment. S.Soup goes further by incorporating an ARAP (as rigid as possible) term to avoid object-level motion seg-

mentation. Both methods provide favorable results on challenging datasets. However, their execution time is still far from practical requirements.

Learning-based methods. We focus on unsupervised learning methods [19, 38, 39, 40], since they demonstrate great potential in practical use, especially in complex scenes without sufficient ground truth data. They apply view synthesis as the only supervision and usually process multiple tasks simultaneously. Zhou *et al.* [40] adopt a depth net and a pose net to estimate depth and camera motion respectively, and a differentiable depth-image-based renderer to associate the depth and pose nets. Mahjourian *et al.* [19] exploit 3D geometric constraints on the basis of [40] and achieve significant improvement over [40] on challenging datasets. Yin *et al.* [39] jointly estimate depth, optical flow and ego-motion. However, unsupervised learning methods are usually ineffective at handling complex dynamic scenes, especially on MPI Sintel dataset.

Other methods. Scene flow methods [18, 20, 31, 33] characterize the 3D motion of points in the scene. They estimate a disparity map for the dynamic scene regardless of the RSA problem. Our method can be considered as a special kind of scene flow method. MRFlow [34] also analyzes the structure of the scene in its *Plane+Parallax* process. However, it depends on physical constraints as well as advanced learning-based techniques to segment foreground objects. Meanwhile, it only reconstructs the background. Yamaguchi *et al.* [35, 36] introduce superpixel relations as a powerful aid in epipolar flow estimation. They derive a slanted-plane MRF model which explicitly reasons about four kinds of superpixel relations to smooth the optical flow. In this paper, we expand their idea to reconstruct dynamic scenes.

3. Two-frame method

We propose a unified monocular dense depth estimation framework for dynamic scenes. We first introduce our method under a two-frame setting and then extend it to handle multiple frames.

We build our method upon the widely-used piecewise planar model. As in [16, 37], we model a generic dynamic scene with a set of non-overlapping rigid regions. We approximate the motion of a small region with an 8-dof homography model, which induces a 3D plane undergoing rigid motion.

Given two consecutive frames I_t and I_{t+1} as the current and next frame, we aim to identify camera motion parameters $\{R_0, t_0\}$ and estimate a dense depth map D_R for I_t.

In the preprocessing stage, we first compute semi-dense (CPM [11]) or dense (optical flow methods like [12]) point matches between I_t and I_{t+1}. We denote $\boldsymbol{M} = \{(p_i, p_j)|p_i \in I_t, p_j \in I_{t+1}\}$ as the set of point matches. $p_i \in \mathbb{R}^2$ and $p_j \in \mathbb{R}^2$ are coordinates of matched pixels.

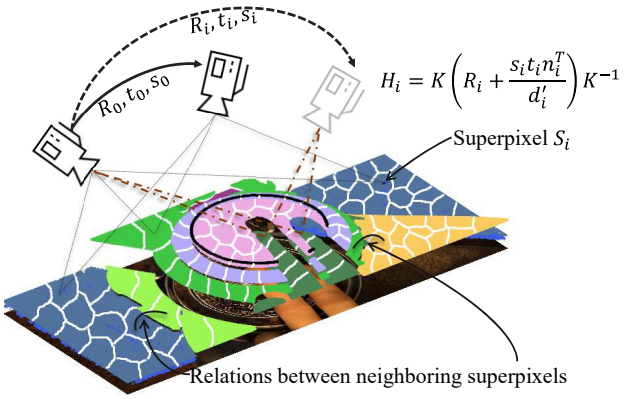


Figure 2. For each superpixel S_i, by decomposing H_i, we obtain its corresponding camera rotation R_i, translation t_i, plane norm n_i and depth d_i', up to a scale s_i [9]. K denotes camera intrinsic matrix. The scene is split into several rigid structures, each represented by a different color. Using SFM techniques (decomposing homographies) to solve the relative scale relations among different parts is ill-posed. This is defined as the RSA problem in dynamic reconstruction. Our method exploits relations between neighboring superpixels to solve the problem.

P_i is the homogeneous form of p_i. Then we over-segment I_t into n non-overlapping superpixels $\boldsymbol{S} = \{S_1, S_2, ..., S_n\}$ with SLIC [1]. We construct a superpixel-level undirected graph $G_s = (\boldsymbol{S}, \boldsymbol{E}, \boldsymbol{\omega_s})$, where $\boldsymbol{E} = \{(i,j)|S_i, S_j$ are neighboring superpixels $\}$ and $\boldsymbol{\omega_s}$ denotes the weight of each edge in $\boldsymbol{E}$. As in [10, 27], we use structure edge detection (SED [4]) to generate a cost map, and $w_s(S_i, S_j)$ is the geodesic distance from the center of S_i to the center of S_j on the cost map.

After preprocessing, we develop our method in four main stages: superpixel relations analysis, motion selection, reconstruction, and refinement. In the first stage, we estimate an 8-dof homography model H_i for each superpixel S_i and determine motion relations $\boldsymbol{R_e} = \{(i,j,r)|r \in \{coplanar, hinge, crack\}, (i,j) \in \boldsymbol{E}\}$ between neighboring superpixels. For convenience, we use $\{co, hi, cr\}$ to denote the three kinds of relations respectively. We predict the spatial relations $\boldsymbol{R_s}$ based on $\boldsymbol{R_e}$. In the motion selection stage, we simultaneously select a set of reliable static superpixels $\boldsymbol{S_t}$ that belong to the background and identify the camera motion $\{R_0, t_0\}$. In reconstruction stage, we estimate a three-dimensional plane parameter $\theta_i \in \mathbb{R}^3$ and a scale parameter s_i for each superpixel S_i. For pixel p_j in S_i, $\theta_i P_j = d(p_j)$, where $d(p_j)$ is the inverse depth of p_j. Then we obtain the reconstructed depth map D. In the final refinement process, we improve the quality of depth map D at pixel level and output the refined result D_R. In this paper, λ_* are weighting parameters of different energy terms. τ_* are thresholds. The flow diagram of our method is demonstrated in Figure 3.

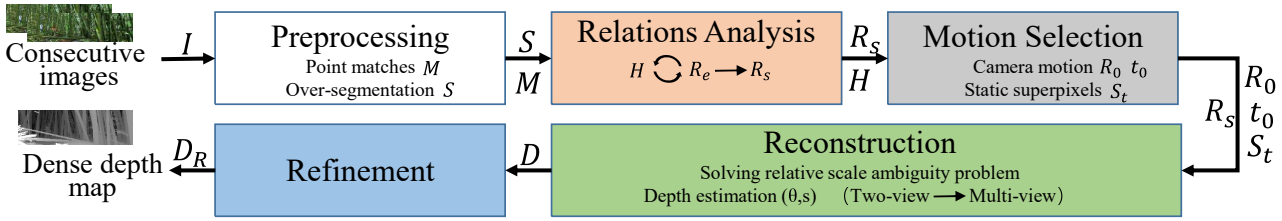


Figure 3. The pipeline of the proposed method for two-frame reconstruction. In this diagram, H, θ, s denote homography models, plane parameters and scale parameters for superpixels in $\boldsymbol{S}$ respectively.

Algorithm 1 Optimization of Eq. 1

Input:
Input image I_t and I_{t+1};
Superpixels $\boldsymbol{S}$;
Matches $\boldsymbol{M}$;
Output:
$\boldsymbol{H}$, $\boldsymbol{R_e}$;

iter=0.
Minimize Eq.2 for $\boldsymbol{H}$ with fast propagation [10].
while iter < 3 **do**
  Fix $\boldsymbol{H}$, minimize Eq.4+Eq.5 for $\boldsymbol{R_e}$.
  Fix $\boldsymbol{R_e}$, minimize Eq.2+Eq.4 for $\boldsymbol{H}$ with a variant of fast propagation [10].
  iter=iter+1.
end while
return $\boldsymbol{H}$, $\boldsymbol{R_e}$.

3.1. Superpixel relations analysis

The concept of superpixel relations is successfully used in [35, 36] to smooth optical flow, while we follow and develop this idea to solve the RSA problem. We subdivide superpixel relations into two categories: motion relations and spatial relations, each having three subcategories $\{coplanar, hinge, crack\}$. Motion relations $\boldsymbol{R_e}$ are determined by homographies $\boldsymbol{H}$, and spatial relations $\boldsymbol{R_s}$ are determined by plane parameters $\boldsymbol{\theta}$. We summarize the criteria of motion relations and spatial relations in Table 2. If two superpixels S_i and S_j are coplanar, they agree in all pixels in them. If they form a hinge, they agree only in the shared boundary between them. Otherwise, they form a crack.

$\boldsymbol{R_e} \rightarrow \boldsymbol{R_s}$. The RSA problem is a major challenge in dynamic reconstruction since traditional SFM theory [9] provides no constraints on the scale of each superpixel. As shown in Figure 2, after over-segmentation, each superpixel S_i corresponds to a virtual camera that undergoes rotation R_i and translation t_i in 3D space. And the scale s_i of each superpixel is unconstrained. We observe that spatial relations $\boldsymbol{R_s}$ can be used to solve the RSA problem since $\boldsymbol{R_s}$ provides constraints on the depth of neighboring superpixels. However, $\boldsymbol{R_s}$ is hard to be estimated directly, because plane parameters $\boldsymbol{\theta}$ are also unknown.

In this paper, we predict spatial relations $\boldsymbol{R_s}$ based on motion relations $\boldsymbol{R_e}$. $\boldsymbol{R_e}$ is determined by homographies, which can be easily estimated with point matches or optical flow. Although the real relationship between $\boldsymbol{R_e}$ and $\boldsymbol{R_s}$ is hard to describe exactly, we observe that $\boldsymbol{R_e}$ and $\boldsymbol{R_s}$ correspond one-to-one in most dynamic scenes. In other words, we assume $\boldsymbol{R_s} = \boldsymbol{R_e}$. Notice that a superpixel usually has multiple neighbors that may provide conflicting or wrong spatial constraints to the reference superpixel. It is not straightforward to decide how to leverage the information provided by all neighboring superpixels and obtain a reasonable reconstruction. Our unified framework selects reliable relations to reconstruct each superpixel by optimizing an energy function, which ensures accuracy and robustness in dynamic reconstruction.

Joint Estimation. We jointly estimate homography models $\boldsymbol{H}$ and determine motion relations $\boldsymbol{R_e}$ with point matches $\boldsymbol{M}$. The energy function is defined as follows,

$$\begin{aligned} E_{sra}(\boldsymbol{H}, \boldsymbol{R_e}) = & \sum_{S_i \in \boldsymbol{S}} E_{data}(H_i) \\ & + \lambda_{s_1} \sum_{(i,j) \in \boldsymbol{E}} E_o(R_e(i,j)) \\ & + \lambda_{s_2} \sum_{(i,j) \in \boldsymbol{E}} E_{pair}(H_i, H_j, R_e(i,j)), \end{aligned} \tag{1}$$

where $R_e(i,j)$ is the motion relation of neighboring superpixels S_i and S_j. E_{data} is applied as the data term to estimate homography models $\boldsymbol{H}$. To define E_{data}, we first initialize the local neighboring matches of each superpixel as its support matches as in [10]. The support matches of the superpixels labelled with blue dots are demonstrated in Figure 4 (d) and (e). We denote $S_u(S_i)$ to represent the support matches of S_i. Then E_{data} is defined as follows,

$$E_{data} = \frac{1}{|Z_i|} \sum_{p_l \in S_u(S_i)} \omega_c(S_i, p_l) \cdot min(|H_i \cdot P_l - P_k|, \tau_g), \tag{2}$$

where $|Z_i|$ is the adaptive normalization parameter, $Z_i = \sum_{p_l \in S_u(S_i)} \omega_c(S_i, p_l)$, $(p_l, p_k) \in M$. The weight ω_c is defined as,

$$\omega_c(S_i, p_l) = exp(-\omega_s(S_i, S_m)/\gamma), p_l \in S_m \tag{3}$$

where γ is a constant.

The pairwise term E_{pair} encapsulates homography models $\boldsymbol{H}$ and motion relations $\boldsymbol{R_e}$,

$$E_{pair}(H_i, H_j, R_e(i,j)) = \begin{cases} 0 & R_e(i,j) = cr \\ \frac{1}{|B_{ij}|}\sum_{p\in B_{ij}} |H_i \cdot P - H_j \cdot P| & R_e(i,j) = hi \\ \frac{1}{|S_i \cup S_j|}\sum_{p\in S_i\cup S_j} |H_i \cdot P - H_j \cdot P| & R_e(i,j) = co. \end{cases} \quad (4)$$

where $|B_{ij}|$ denotes the number of pixels on the shared boundary between superpixel S_i and S_j.

E_o is the *Occam's Razor* [35, 36], which prefers simpler explanations of the scene, i.e., *coplanar* over *hinge*, *hinge* over *crack*. E_o is defined as follows,

$$E_o(R_e(i,j)) = \begin{cases} \lambda_{crack} & R_e(i,j) = cr \\ \lambda_{hinge} & R_e(i,j) = hi \\ \lambda_0 & R_e(i,j) = co \end{cases}, \quad (5)$$

where $\lambda_{crack} > \lambda_{hinge} > \lambda_0 = 0$.

Optimization. The minimization of Eq. 1 is not trivial. We follow [36] to design an efficient block coordinate descent inference method. We alternate to estimate the homography models and determine motion relations between neighboring superpixels. Firstly, we estimate $\boldsymbol{H}$ only with the E_{data} term. We adopt the fast propagation algorithm proposed in [10] to optimize E_{data}. Then we fix the homography model of each superpixel and determine the relations by minimizing $E_{pair} + E_o$ in a closed form. Next, we fix the relations between neighboring superpixels and minimize $E_{data} + E_{pair}$ to refine the homographies by applying an improved variant of the fast propagation method [10]. We introduce the method in detail in Sup.Mat. Then we repeat the last two steps to iteratively refine the motion relations and homographies. At the end, we use our assumption $\boldsymbol{R_s} = \boldsymbol{R_e}$ to predict spatial relations $\boldsymbol{R_s}$. We summarize the optimization process in Algorithm 1.

3.2. Motion selection

The output of the superpixel relations analysis step is a set of homographies $\boldsymbol{H}$ and spatial relations $\boldsymbol{R_s}$. We follow [9] to decompose each homography model and generate $2n$ hypotheses of camera rotation $\boldsymbol{R}$, translation $\boldsymbol{t}$, plane norm $\boldsymbol{n}$ and inverse depth $\boldsymbol{d}$, up to scale. To handle pure rotation cases, we follow [26] to compute the distance between the identity matrix I and the matrix HH^T with the metric $\Phi 4$ proposed in [13]. We consider only the rotation component for homographies whose $\Phi 4$ distance lie below a threshold.

In this step, we follow [26] to jointly identify camera motion $\{R_0, t_0\}$ and select a set of static superpixels $\boldsymbol{S_t}$ belonging to the background in a PEaRL framework [14]. For fast running speed, we also design an intuitive method to reduce the dimension of the label set. Details of the motion selection procedure appear in the Sup. Mat.

Notice that our method does not require accurate object segmentation in this step. Instead, we only select several reliable superpixels that belong to the background. These superpixels will serve as references to estimate the scales of the remaining superpixels. For this purpose, we fix the scales of superpixels in $\boldsymbol{S_t}$ in the reconstruction process and adjust the scales of remaining ones based on spatial relations.

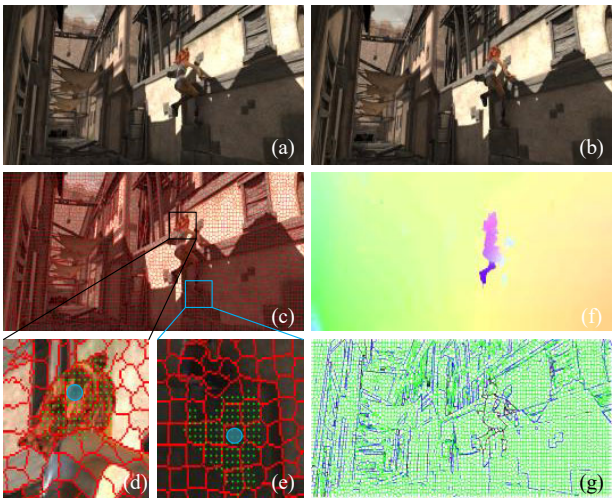


Figure 4. Processing flow of the superpixel relations analysis step. (a) and (b) are current frame I_t and next frame I_{t+1}. (c) is the result of over-segmentation. (d) and (e) demonstrate the support matches of the superpixels labeled with blue dots. (f) demonstrates the flow field indicated by the homography models. (g) demonstrates motion relations. The meanings of the color lines in (g) are explained in Figure 1.

3.3. Reconstruction

After superpixel relations analysis and motion selection, we obtain a set of floating superpixels in 3D space whose scales are undetermined. In this step, we aim to solve the RSA problem and provide an accurate and smooth depth map for the dynamic scene.

Energy function. We model each superpixel S_i with a plane parameter $\theta_i = [\theta_{i1}, \theta_{i2}, \theta_{i3}]^T$ and a scale parameter s_i. The reconstruction pipeline is formulated as an energy optimization problem with the following energy function,

$$\begin{aligned} E_{re}(\boldsymbol{\theta}, \boldsymbol{s}) &= \lambda_{r_1} \sum_{S_i \in \boldsymbol{S}} E_{fit}(\theta_i, s_i) + \sum_{(j,k)\in \boldsymbol{E}} E_{rel}(\theta_j, \theta_k) \\ &+ \lambda_{r_2} \sum_{S_i \in \boldsymbol{S}} E_{occ}(s_i) + \lambda_{r_3} \sum_{(j,k)\in \boldsymbol{E_c}} E_{pri}(\theta_j, \theta_k), \\ \boldsymbol{s.t.} \quad & s_i = 1, \forall S_i \in \boldsymbol{S_t}, \end{aligned} \quad (6)$$

Let $\boldsymbol{S_r}$ be the set of superpixels which are either in $\boldsymbol{S_t}$ or connect to $\boldsymbol{S_t}$. On graph G_s, we define that a superpixel S_i ($S_i \notin \boldsymbol{S_t}$) connects to $\boldsymbol{S_t}$ if there exists a path connecting S_i to a superpixel in $\boldsymbol{S_t}$ such that all spatial relations along the path are either coplanar or hinge. Then we define $\boldsymbol{E_c} = \{(i,j) | (i,j) \in \boldsymbol{E}, S_i \in \boldsymbol{S_r}, S_j \in \boldsymbol{S} \setminus \boldsymbol{S_r}\}$. The E_{rel} term

is defined as follows,

$$E_{rel}(\theta_j, \theta_k) = \begin{cases} 0 & R_s(j,k) = cr \\ \frac{\omega_{hi}}{|B_{jk}|} \sum_{p \in B_{jk}} |\theta_j \cdot P - \theta_k \cdot P| & R_s(j,k) = hi \\ \frac{\omega_{co}}{|S_k \cup S_j|} \sum_{p \in S_k \cup S_j} |\theta_j \cdot P - \theta_k \cdot P| & R_s(j,k) = co, \end{cases} \quad (7)$$

where ω_{hi}, ω_{co} are weighting constants and $\omega_{hi} < \omega_{co}$.

The E_{pri} term is adopted to ensure that the plane parameter of each superpixel is fully constrained. We force every superpixel to be in S_t or connect to S_t based on the assumption that moving regions are supported by the surrounding environment, which is similar to the ordering constraint in [25]. Then the E_{pri} term is defined as follows,

$$E_{pri}(\theta_j, \theta_k) = \frac{\omega_{cr}}{|B_{jk}|} \sum_{p \in B_{jk}} |\theta_j \cdot P - \theta_k \cdot P| + \delta(\theta_j \cdot \bar{P} \geq \theta_k \cdot \bar{P}), \quad (8)$$

where $\bar{P} = 1/|B_{jk}| \sum_{p \in B_{jk}} P$, ω_{cr} is a weighting constant and $\delta(\cdot)$ is an indicator function defined as follows,

$$\delta(c) = \begin{cases} 0 & c \ is \ false \\ 1 & c \ is \ true. \end{cases} \quad (9)$$

The E_{fit} term fits a plane for each superpixel,

$$E_{fit}(\theta_i, s_i) = \sum_{p_i \in S_i} min(|\theta_i \cdot P_i - s_i d_i \boldsymbol{n_i}^T K^{-1} P_i|, \tau_r) \quad (10)$$

and the E_{occ} term encourages simple explanations to the scene and is defined as $E_{occ}(s_i) = \delta(s_i \neq 1)$.

Optimization. Similar to the optimization of Eq. 1, we use a block coordinate descent algorithm to minimize Eq. 6. We first process superpixels in S_r by minimizing $E_{fit} + E_{rel} + E_{occ}$. Plane parameters θ and scales s are propagated among spatially neighboring superpixels with the improved fast propagation method. Note that for superpixels in S_t, their scales are fixed to 1. Then we process the superpixels that are not in S_r by minimizing the E_{pri} term. Finally, all superpixels are optimized together by minimizing $E_{fit} + E_{rel} + E_{occ} + E_{pri}$ with the improved fast propagation method. We provide detailed explanations of the optimization in Sup.Mat.

3.4. Refinement

A pixel-level refinement serves as the final step of our pipeline. Since superpixels may not adhere tightly to boundaries, we use the fast smoothing method proposed in [21] to improve the quality of the depth map.

4. From two-frame to multi-frame

Given a video with a sequence of frames $\boldsymbol{I} = \{I_1, I_2, ..., I_t, I_{t+1}\}$, we aim to estimate a dense depth map for I_t. Since a multi-frame setting helps in many aspects (e.g. consistency checking [41]) in the reconstruction of dynamic scenes, we mainly study the effect of tracking superpixels between temporally neighboring frames to achieve a more accurate estimation of motion relations. Prior methods on temporally consistent superpixels [3, 17] have achieved satisfying performance in challenging environments. We follow [17] to track superpixels, and then for I_t, we get a prior of motion relations $\boldsymbol{E_p} = \{(i,j)\}$, $\boldsymbol{R_p} = \{(i,j,r)\}$, $\boldsymbol{w_p} = \{(i,j,w_{ij}^m)\}$, where S_i and S_j are temporally consistent [17] and spatially neighboring superpixels. $r \in \{coplanar, hinge\}$ and w_{ij}^m records how many frames the relations hold. Then the total energy function to jointly estimate homographies and determine motion relations is defined as follows,

$$E_m(\boldsymbol{H}, \boldsymbol{R_e}) = E_{sra} + \lambda_m \sum_{(i,j) \in E_p} E_{mul}(R_e(i,j)). \quad (11)$$

The E_{mul} term is defined as follows,

$$E_{mul}(R_e(i,j)) = min(w_{ij}^m, \tau_m) \delta(R_e(i,j) \neq R_p(i,j)). \quad (12)$$

The optimization process of Eq.11 is similar to Eq.1. Other stages are the same as two-frame reconstruction pipeline.

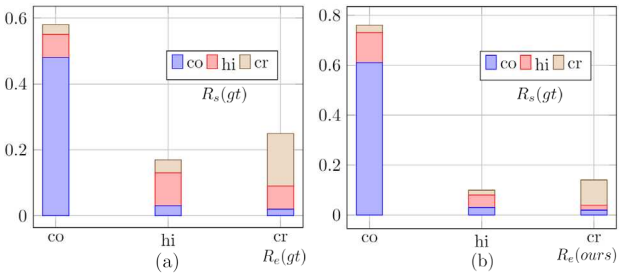


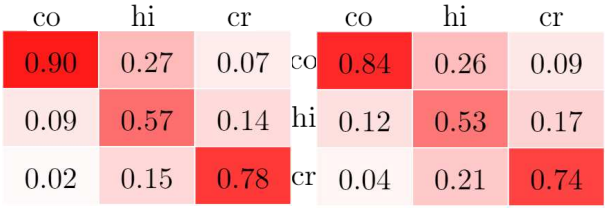
Figure 5. Distribution of motion and corresponding spatial relations. The height of each bar reflects the percentage of current type of motion relations in total motion relations. And inside each bar, we show the corresponding ground truth distribution of spatial relations. For example, in (a), considering the first bar, coplanar relations account for about 58% of total motion relations. And in this bar, considering the corresponding spatial relations, coplanar relations occupy about 83%, hinge relations 12% and crack relations 5%. (a) and (b) verify that our assumption $\boldsymbol{R_s} = \boldsymbol{R_e}$ is reasonable in most dynamic cases.

5. Experiments

Implementation details. In our experiment, we set superpixel size to be about 150 pixels per superpixel. We set $\{\lambda_{s_1}, \lambda_{s_2}, \gamma, \lambda_{r_1}, \lambda_{r3}, \omega_{cr}, \omega_{hi}, \omega_{co}\} = \{1, 1, 0.1, 1, 1, 1, 0.5, 2\}$. Other Parameters were adjusted differently for each dataset. We use a small split of the datasets to optimize the parameters and evaluate our method on the remaining parts. For multi-frame depth reconstruction, we use five consecutive frames from a video and reconstruct the fourth frame. For learning-based methods [39, 40], we provide training details in the Sup.Mat.

Method	DT [15]	MVG [9]	DMDE [25]	Superpixel Soup[16]	SFMLeaner [40]	Geonet [39]	Ours+ MirrorFlow	Ours+ CPM	Ours+ CPM
Settings	S	T	T	T	M	M	T	T	M
MPI Sintel	0.4903	0.3327	0.2970*	0.1669*	0.4733	0.4398	0.2017	0.2070	**0.1632**
Virtual KITTI	0.2911	0.2432	-	0.1045*	0.1532	0.1430	**0.0930**	0.1276	0.1010
KITTI	0.2217	0.2907	0.1480*	0.1268*	0.1817	0.1630	**0.1023**	0.1340	0.1232
SYNTHIA	0.1910	0.1973	-	-	0.1537	0.1321	**0.1123**	0.1337	0.1323
Running Time	100s	600s	$(60s)^*$	$700s^*$	$(0.02s)$	$(0.02s)$	600s	5s	5s

Table 3. Performance comparison of depth accuracy and time consumption. The table lists the MRE on different datasets. We adopt results of DMDE [25] and S.Soup [16] (labelled with *) from their papers, since the authors haven't released the source code. Running times of [16, 25, 40] are tested on GPU. It is clearly shown that our method achieves state-of-the-art performance on various benchmarks in a reasonable time. In summary, Ours+MirrorFlow (T) performs best in traffic scenes (Virtual KITTI, KITTI, SYNTHIA) and Ours+CPM (M) performs best in other scenes (MPI Sintel).



(a) $R_e(ours) - R_e(gt)$ (b) $R_s(ours) - R_s(gt)$

Figure 6. Confusion matrices (a) and (b) demonstrate the accuracy of our method in estimating motion relations $\boldsymbol{R_e}$ and spatial relations $\boldsymbol{R_s}$. The horizontal axis shows superpixel relations predicted by our method and the vertical axis shows ground truth relations.

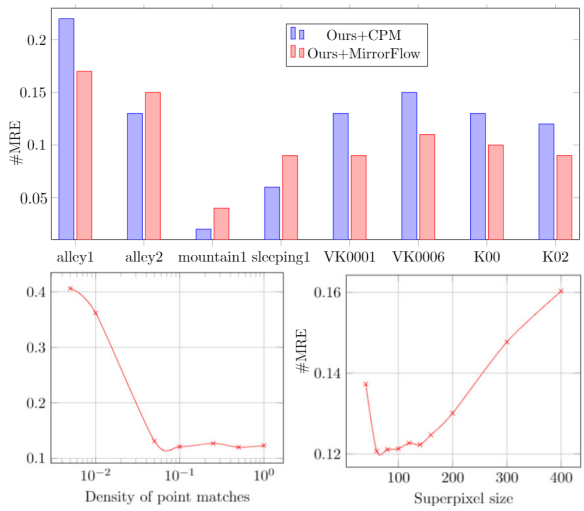


Figure 7. Parameter analysis. **Upper**: Effect of different optical flow estimation methods. We test our two-frame method with CPM [11] (semi-dense matches) and MirrorFlow [12] (dense flow field). **Lower Left**: The effect of density of point matches on reconstruction quality. **Lower Right**: The trend of MRE as the superpixel size increases.

In this section, we evaluate the performance of our method both qualitatively and quantitatively on challenging datasets containing moving objects, including synthetic

	MPI Sintel	KITTI	SYNTHIA
DT [15]	9.7%	24.9%	21.1%
MVG [9]	37.4%	45.7%	33.2%
Geonet [39]	20.1%	52.2%	43.7%
Ours+MirrorFlow	44.0%	**67.7%**	**59.3%**
Ours+CPM (T)	39.6%	56.3%	48.3%
Ours+CPM (M)	**50.1%**	59.7%	55.5%

Table 4. Performance comparison under depth inlier rate.

Figure 8. Results on Youtube-objects dataset [24].

datasets MPI Sintel [2], Virtual KITTI [5], SYNTHIA [28], real-world datasets KITTI [6, 7] and Youtube-objects [24]. We choose MVG [9], DT [15], FGI [21], DMDE [25], S.Soup [16], SFMLearner [40] and Geonet [39] as the baseline methods. Note that MVG is a traditional SFM pipeline implemented by ourselves, in which we use MirrorFlow [12] to estimate dense correspondences between two consecutive frames and then directly apply triangulation to reconstruct the scene. For convenience, we define three abbreviations to represent the setting of each method, "S" for single-frame, "T" for two-frame and "M" for multi-frame. We provide the results of three variants of our method: ours + CPM (T), ours + MirrorFlow (T) and ours + CPM (M). Since the authors of DMDE and S.Soup haven't released the source codes, we simply show the evaluation results posted in their papers.

Assumption $\boldsymbol{R_s = R_e}$. We use Virtual KITTI dataset to verify that our assumption is effective and reasonable in dynamic scenes. In Figure 5, to obtain the ground truth distribution of motion and spatial relations, we directly compute homographies and plane parameters on ground truth optical flow and depth data and determine superpixel rela-

tions with the criteria in Table 2. Figure 5 (a) reflects that the assumption $\boldsymbol{R_s} = \boldsymbol{R_e}$ is applicable in most dynamic cases. Figure 5 (b) demonstrates the distribution of motion relations estimated by ours + CPM (T) method and the corresponding ground truth spatial relations, which reflects that the assumption is also effective when using the homographies estimated by our method.

Parameter analysis. Figure 7 studies the effect of point matches estimation methods, the density of point matches and superpixel size over the quality of two-frame depth reconstruction. We select 120 pairs of images from MPI Sintel, Virtual KITTI, and SYNTHIA datasets and use MRE as the evaluation metric. Let Z^e denote the estimated depth map and Z^{gt} denote the ground truth.

$$MRE = \frac{1}{M_t} \sum_{p \in I_t} |Z_p^e - Z_p^{gt}| / Z_p^{gt}, \tag{13}$$

where M_t is the number of pixels in image I_t. In Figure 7, we can see that introducing other constraints (MirrorFlow introduces symmetries) helps to improve the quality of the depth map in most cases. For the density of point matches, we observe that the quality of the depth map improves quickly and finally achieves a stable result as the density increases. For superpixel size, we observe that the MRE of results keeps stable when superpixel size increases from 50 to 150 pixels per superpixel.

Result analysis. Figure 6 (a) and (b) are confusion matrices that demonstrate the accuracy of the estimated motion relations $\boldsymbol{R_e}$ and spatial relations $\boldsymbol{R_s}$ on Virtual KITTI dataset. It is shown clearly that our method correctly estimates most of the relations. The major errors come from the hinge relation. Our method sometimes mistakes hinge relation for coplanar relation. But in practical reconstruction experiments, our pipeline still outputs reasonable results since each superpixel has many neighboring superpixels and we can select reliable relations for reconstruction by minimizing the energy function Eq. 6.

Table 3 provides a statistical comparison between our method and other competing methods over depth quality. In order to compare our method with state-of-the-art methods DMDE [25] and S.Soup [16], we use the same experiment setup as [16] and follow [25] and [16] to use MRE as the evaluation metric. For learning-based methods [39, 40], we directly evaluate the public training models provided by the authors on Virtual KITTI, KITTI and SYNTHIA datasets, while fine-tuning the models before testing on MPI Sintel dataset. In Table 4, we also provide results comparison with the inlier rate as the metric. In this experiment, the inlier rate is defined as, *inlier rate* $= \frac{1}{M_t} \sum_{p \in I_t} \delta(|Z_p^{gt} - Z_p^e| < 10\% \cdot Z_p^{gt})$, and higher is better.

Next, we introduce the performance of our methods on different datasets in detail.

MPI Sintel, derived from an open source 3D animated short film, is a famous dataset for the evaluation of optical flow, depth, segmentation and so on. For depth reconstruction, this dataset is very challenging due to irregular deformation of objects, significant illumination changes, and complex scene structures. On this dataset, our two-frame method outperforms other competing methods except for S.Soup. Our methods may output low-quality depth maps when the foreground objects are too large and do not connect to the background. More details can be found in Sup.Mat. When using multiple frames, we get superior results to other methods since the estimation of superpixel relations improves.

KITTI, a novel challenging real-world computer vision benchmark for multiple tasks, can be used to test depth reconstruction methods since it provides sparse LiDAR measurements as the ground truth depth. It is shown clearly in Table 3 that the results of ours+MirrorFlow and Ours+CPM (M) outperforms all other methods. And although ours+CPM (T) is inferior to S.Soup, our method runs much faster. On this dataset, we observe that using MirrorFlow to estimate homographies greatly improves the reconstruction results since MirrorFlow exploits symmetries and generally outputs more accurate homographies in traffic scenes.

Virtual KITTI and SYNTHIA. The two datasets provide synthetic traffic videos with perfect ground truth depth. In Table 3, the results of ours+Mirrorflow and ours+CPM (M) outperform competitors. And Ours+CPM (T) provides comparable results to other methods but runs much faster.

We provide qualitative depth estimation results on Youtube-objects dataset [24] in Figure 8. More results are demonstrated in Sup.Mat.

6. Conclusion

We propose a unified framework for dense monocular depth estimation in complex dynamic scenes with two or more frames. We build our method based on the piecewise planar assumption and our observation that motion relations indicate spatial relations in most dynamic cases. This observation provides a new insight into solving the RSA problem in dynamic reconstruction. Our results on popular public benchmarks demonstrate clearly that our method successfully handles various dynamic scenes and achieves superior performance to state-of-the-art methods at a much faster speed. We believe that combining our method with other advanced techniques (e.g. deep learning) will lead to further improvements in the quality of depth map.

References

[1] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, Sabine Süsstrunk, et al. SLIC superpixels compared to state-of-the-art superpixel methods. *TPAMI*, 34(11):2274–2282, 2012.

[2] Daniel J. Butler, Jonas Wulff, Garrett B. Stanley, and Michael J. Black. A naturalistic open source movie for optical flow evaluation. In A. Fitzgibbon et al. (Eds.), editor, *ECCV*, Part IV, LNCS 7577, pages 611–625. Springer-Verlag, Oct. 2012.

[3] Jason Chang, Donglai Wei, and J. W. Fisher. A video representation using temporal superpixels. In *CVPR*, 2013.

[4] Piotr Dollar and C Lawrence Zitnick. Fast edge detection using structured forests. *TPAMI*, 37(8):1558–1570, 2015.

[5] Adrien Gaidon, Qiao Wang, Yohann Cabon, and Eleonora Vig. Virtual worlds as proxy for multi-object tracking analysis. In *CVPR*, 2016.

[6] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The KITTI dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

[7] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *CVPR*, 2012.

[8] Christian Hane, Christopher Zach, Andrea Cohen, Roland Angst, and Marc Pollefeys. Joint 3D scene reconstruction and class segmentation. In *CVPR*, pages 97–104, 2013.

[9] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[10] Yinlin Hu, Yunsong Li, and Rui Song. Robust interpolation of correspondences for large displacement optical flow. In *CVPR*, 2017.

[11] Yinlin Hu, Rui Song, and Yunsong Li. Efficient coarse-to-fine patchmatch for large displacement optical flow. In *CVPR*, pages 5704–5712, 2016.

[12] Junhwa Hur and Stefan Roth. Mirrorflow: Exploiting symmetries in joint optical flow and occlusion estimation. In *ICCV*, pages 312–321, 2017.

[13] Du Q Huynh. Metrics for 3D rotations: Comparison and analysis. *Journal of Mathematical Imaging and Vision*, 35(2):155–164, 2009.

[14] Hossam Isack and Yuri Boykov. Energy-based geometric multi-model fitting. *IJCV*, 97(2):123–147, 2012.

[15] Kevin Karsch, Ce Liu, and Sing Bing Kang. Depth transfer: Depth extraction from video using non-parametric sampling. *TPAMI*, 36(11):2144–2158, 2014.

[16] Suryansh Kumar, Yuchao Dai, and Hongdong Li. Monocular dense 3D reconstruction of a complex dynamic scene from two perspective frames. In *ICCV*, pages 4649–4657, 2017.

[17] Se Ho Lee, Won Dong Jang, and Chang Su Kim. Temporal superpixels based on proximity-weighted patch matching. In *ICCV*, 2017.

[18] Zhaoyang Lv, Chris Beall, Pablo F Alcantarilla, Fuxin Li, Zsolt Kira, and Frank Dellaert. A continuous optimization approach for efficient and accurate scene flow. In *ECCV*, pages 757–773. Springer, 2016.

[19] Reza Mahjourian, Martin Wicke, and Anelia Angelova. Unsupervised learning of depth and ego-motion from monocular video using 3D geometric constraints. In *CVPR*, pages 5667–5675, 2018.

[20] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *CVPR*, 2015.

[21] Dongbo Min, Sunghwan Choi, Jiangbo Lu, Bumsub Ham, Kwanghoon Sohn, and Minh N Do. Fast global image smoothing based on weighted least squares. *TIP*, 23(12):5638–5653, 2014.

[22] Houwen Peng, Bing Li, Weihua Xiong, Weiming Hu, and Rongrong Ji. Rgbd salient object detection: a benchmark and algorithms. In *ECCV*, pages 92–109. Springer, 2014.

[23] Matia Pizzoli, Christian Forster, and Davide Scaramuzza. Remode: Probabilistic, monocular dense reconstruction in real time. In *ICRA*, 2014.

[24] Alessandro Prest, Christian Leistner, Javier Civera, Cordelia Schmid, and Vittorio Ferrari. Learning object class detectors from weakly annotated video. In *CVPR*, pages 3282–3289. IEEE, 2012.

[25] Rene Ranftl, Vibhav Vineet, Qifeng Chen, and Vladlen Koltun. Dense monocular depth estimation in complex dynamic scenes. In *CVPR*, pages 4058–4066, 2016.

[26] Carolina Raposo and Joao P Barreto. π match: Monocular vSLAM and piecewise planar reconstruction using fast plane correspondences. In *ECCV*, pages 380–395. Springer, 2016.

[27] Jerome Revaud, Philippe Weinzaepfel, Zaid Harchaoui, and Cordelia Schmid. EpicFlow: Edge-preserving interpolation of correspondences for optical flow. In *CVPR*, pages 1164–1172, 2015.

[28] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M. Lopez. The SYNTHIA dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *CVPR*, June 2016.

[29] Chris Russell, Joao Fayad, and Lourdes Agapito. Energy based multiple model fitting for non-rigid structure from motion. In *CVPR*, pages 3009–3016. IEEE, 2011.

[30] Chris Russell, Rui Yu, and Lourdes Agapito. Video pop-up: Monocular 3D reconstruction of dynamic scenes. In *ECCV*, pages 583–598. Springer, 2014.

[31] Tatsunori Taniai, Sudipta N Sinha, and Yoichi Sato. Fast multi-frame stereo scene flow with motion segmentation. In *CVPR*, pages 3939–3948, 2017.

[32] Aydin Varol, Mathieu Salzmann, Engin Tola, and Pascal Fua. Template-free monocular reconstruction of deformable surfaces. In *ICCV*, pages 1811–1818. IEEE, 2009.

[33] Christoph Vogel, Konrad Schindler, and Stefan Roth. 3D scene flow estimation with a piecewise rigid scene model. *IJCV*, 115(1):1–28, 2015.

[34] Jonas Wulff, Laura Sevilla-Lara, and Michael J Black. Optical flow in mostly rigid scenes. In *CVPR*, volume 2, page 7. IEEE, 2017.

[35] Koichiro Yamaguchi, David McAllester, and Raquel Urtasun. Robust monocular epipolar flow estimation. In *CVPR*, pages 1862–1869. IEEE, 2013.

[36] Koichiro Yamaguchi, David McAllester, and Raquel Urtasun. Efficient joint segmentation, occlusion labeling, stereo and flow estimation. In *ECCV*, pages 756–771. Springer, 2014.

[37] Jiaolong Yang and Hongdong Li. Dense, accurate optical flow estimation with piecewise parametric model. In *CVPR*, pages 1019–1027, 2015.

[38] Zhenheng Yang, Peng Wang, Yang Wang, Wei Xu, and Ram Nevatia. LEGO: Learning edge with geometry all at once by watching videos. In *CVPR*, pages 225–234, 2018.

[39] Zhichao Yin and Jianping Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *CVPR*, volume 2, 2018.

[40] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *CVPR*, volume 2, page 7, 2017.

[41] Yuliang Zou, Zelun Luo, and Jia-Bin Huang. DF-Net: unsupervised joint learning of depth and flow using cross-task consistency. In *ECCV*, pages 36–53, 2018.