Multi-body Depth and Camera Pose Estimation from Multiple Views

Andrea Porfiri Dal Cin
Politecnico di Milano
andrea.porfiridalcin@polimi.it

Giacomo Boracchi
Politecnico di Milano
giacomo.boracchi@polimi.it

Luca Magri
Politecnico di Milano
luca.magri@polimi.it

Abstract

Traditional and deep Structure-from-Motion (SfM) methods typically operate under the assumption that the scene is rigid, i.e., the environment is static or consists of a single moving object. Few multi-body SfM approaches address the reconstruction of multiple rigid bodies in a scene but suffer from the inherent scale ambiguity of SfM, such that objects are reconstructed at inconsistent scales. We propose a depth and camera pose estimation framework to resolve the scale ambiguity in multi-body scenes. Specifically, starting from disorganized images, we present a novel multi-view scale estimator that resolves the camera pose ambiguity and a multi-body plane sweep network that generalizes depth estimation to dynamic scenes. Experiments demonstrate the advantages of our method over state-of-the-art SfM frameworks in multi-body scenes and show that it achieves comparable results in static scenes. The code and dataset are available at https://github.com/andreadalcin/MultiBodySfM.

1. Introduction

Real-world environments often contain independently moving objects, and their reconstruction is fundamental for safe robot navigation and augmented reality in complex dynamic environments. Unfortunately, traditional and deep learning Structure-from-Motion (SfM) algorithms assume that the scene is *static* and treat moving objects as outliers, missing important information and producing suboptimal results when the scene is dynamic. In Fig. 1-left, the depth reconstructed by the state-of-the-art DeepSfM [36] shows artifacts in the image region of a moving object (in red).

In this work, we go one step further and enable consistent 3D reconstruction (Fig. 1-right) of a scene with multiple rigidly moving objects. This task, also known as Multi-Body Structure-from-Motion (MBSfM), is still open and far from being solved in practice. Traditional MBSfM attempts [1, 26] segment rigid motions to obtain partial reconstructions of individually moving objects in the scene. However, these methods are hampered by the *relative scale problem*, namely that the 3D structure of each independently moving object is estimated up to a similarity, so that each object is reconstructed in its own scale. Therefore, a unified reconstruction of all objects up to a common global scale factor is impossible without additional information. Similarly, deep methods for SfM that leverage explicit multi-view constraints [34, 36, 13, 29, 4] are affected by the relative scale problem and cannot regress a unified depth map of a multi-body scene since their underlying architecture cannot handle multiple motions.

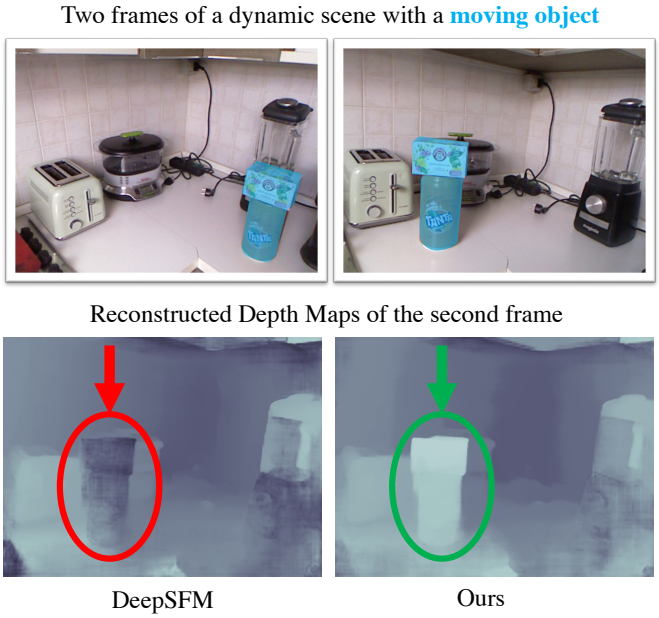

Figure 1: Two-views of a multi-body scene (*top-row*). Depth estimation networks (e.g. [36]) yield inconsistent depths (*bottom-left*) for moving objects. Our method explicitly accounts for moving objects to produce geometrically consistent depths (*bottom-right*).

Recent monocular depth estimators [9, 2] use learned priors to regress dense depth maps from images without exploiting multi-view constraints. Thus, monocular methods implicitly avoid the relative scale problem by regressing consistent depth maps across the entire image. However, monocular depth estimation is inherently ill-posed and these models do not provide the same generalization capabilities of multi-view methods when scenes are static. Fur-

thermore, the 3D structures inferred from different images of the same scene are not guaranteed to be consistent. Thus, this inconsistency eventually manifests in alignment problems that are not trivial to solve.

In this work, we present a novel deep learning framework for depth and camera pose estimation designed specifically for multi-body scenes. Following consolidated MBSfM pipelines, we perform motion segmentation to recover a set of relative camera poses up to a scale factor and the corresponding sparse 3D reconstructions of each moving object. Then, we depart from existing MBSfM approaches by directly solving the relative scale problem and regressing geometrically consistent depth maps and refined poses.

Specifically, our main contributions are twofold:

(1) a *robust scale estimation* to fix the scale ambiguity,

(2) a *multi-body plane sweep network* to regress refined camera poses and depth maps in multi-body scenes.

Our scale estimator uses monocular depth maps to unify the 3D structures and camera poses under a global scale factor. This is based on a robust voting scheme that mitigates the impact of inaccuracies in monocular depth estimates.

We adopt a deep neural network for multi-body depth estimation and pose refinement. The network adopts our novel *multi-body plane sweep* algorithm, which uses all the rigid motions in the scene, now devoid of scale ambiguity, to compute dense depth maps and refined camera poses. Notably, the estimated depth maps are geometrically consistent even in image regions covered by moving objects. Plane sweep has never been extended to support multiple rigid motions to the best of our knowledge. Our network also includes a pose estimation module that refines camera poses based on current depth estimates.

We demonstrate the effectiveness of our approach on both static and multi-body scenes. On static scenes, our method performs on par with traditional and deep SfM methods. On multi-body scenes, it significantly outperforms SOTA deep learning SfM methods in the depth estimation task and achieves comparable camera pose results.

2. Problem Formulation

Multi-body depth and camera pose estimation is framed as follows. The input is a set of n *unstructured* images $\{\mathbf{I}_i\}_{i=1}^n$ captured by a moving camera and depicting a *multi-body* scene in which μ objects $\mathcal{B} = \{\beta_i\}_{i=1}^{\mu}$ independently move according to their *rigid* motions. Without loss of generality, we consider the object β_1 to be static w.r.t. to the camera motion. Our goal is to estimate depth maps $\{\mathbf{D}_i\}_{i=1}^n$ for each image $\mathbf{I}_i$ and the absolute camera poses $\{\mathbf{R}_i, \mathbf{t}_i\}_{i=1}^n$ in the reference frame of β_1. We assume that: *i*) an upper bound M on the maximum number μ of moving objects in the scene is given, *ii*) the camera is *perspective* and its intrinsic parameters $\mathbf{K}$ are known.

Consider that our problem is different from non-rigid SfM [16, 14]. Non-rigid SfM handles a wider class of object deformations but does not assume unstructured input images, typically exploiting dense temporal information, and is restricted to orthographic cameras.

3. Related Work

Depth and camera pose estimation from unstructured images of *multi-body* scenes is largely unexplored, and most SfM methods either consider static scenes only ($\mu = 1$) or treat moving objects as outliers. This section focuses on the few SfM methods that address the multi-body scenario ($\mu > 1$) from unstructured images acquired by perspective cameras. Also, we discuss monocular depth estimation, a component of our solution, and relevant deep multi-view *static* SfM methods.

3.1. Multi-body Structure-from-Motion (MBSfM)

A few works have addressed MBSfM from unstructured images $\{\mathbf{I}_i\}_{i=1}^n$ as a generalization of traditional SfM to multi-body scenes. We identify two main research directions. First, early factorization methods [32, 6] segment rigidly moving objects $\mathcal{B}$ and perform sparse 3D reconstruction in a single step. However, factorization methods rely on restrictive assumptions on camera models and require complete feature tracks, which are seldom available in practice. Due to its lack of robustness, factorization has yet to see practical applications beyond short, unrealistic sequences.

The second, more recent, research direction separates motion segmentation of objects $\mathcal{B}$ and their 3D reconstruction. [1, 26] cluster sparse correspondences based on their rigid motion and then recover the 3D structure of each object independently using a SfM pipeline [27]. As discussed in [23, 24, 17], the relative scale problem affects these methods because each object is reconstructed in its own scale. In SLAM, the relative scale problem has been addressed under specific assumptions: [23] assumes objects move in a one-parameter family of motions, and [17] works only with video input and continuous object motion. Our method also decouples motion segmentation and reconstruction. However, we explicitly address the relative scale problem using a learning-based approach without further assumptions on the input images or the motions in the scene.

3.2. Monocular Depth Estimation

Recent works [9, 2] recover a dense depth map $\mathbf{D}_1$ from a single image $\mathbf{I}_1$. These methods learn priors to regress depth maps, allowing consistent scale reconstruction of moving objects $\mathcal{B}$ without scale ambiguity. However, due to the ill-posed nature of monocular depth estimation, generalization is limited and multi-view approaches remain the benchmark for SfM in deep learning.

In [20, 38], monocular depth maps are estimated from video frames, and then a network is fine-tuned on the entire

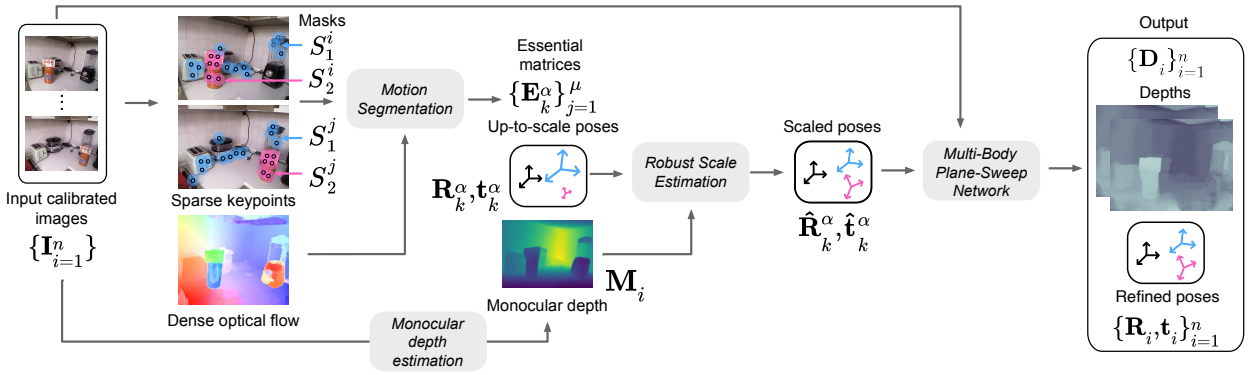

Figure 2: The three main steps of our method: i) *motion segmentation*: produces a set of essential matrices describing the rigid motion of objects between view pairs. ii) *robust scale estimation*: uses monocular depth cues to fix the scale ambiguity of each moving object. iii) *multi-body plane-sweep network* based on our novel multi-body plane-sweep regress refined depth maps and camera poses.

video sequence. These methods do not assume unstructured images and require many frames for depth and pose tracking and fine-tuning. In addition, depth estimation for moving objects still relies solely on depth priors, which can lead to inaccuracies as in standard monocular depth estimation.

3.3. Deep Learning for SfM

Deep learning methods for SfM from unstructured images regress camera poses and dense depth maps using multi-view stereo matching. DeMoN [33] works on image pairs using stacked encoder-decoder networks but lacks explicit multi-view constraints and has limited accuracy. More recent architectures [29, 4, 36, 34, 13, 31] replace the generic encoder-decoder architecture with layers that explicitly enforce multi-view constraints. Other works achieve better results by taking inspiration from traditional Bundle Adjustment: DeepSfM [36] introduces depth and pose cost volumes for iterative refinement, and [29] uses a BA-layer for regression from basis depth maps. Wang et al. [34] propose a scale-invariant network for depth and pose estimation for view pairs. Specifically, camera poses are estimated from essential matrices robustly fitted on fine-tuned optical flow matches, thus treating all moving objects as outliers. Then, depth maps are regressed end-to-end using a CNN similar to DeepSfM. Finally, [4] optimizes non-linear cost functions in SfM using recurring neural networks.

As the above deep methods embed explicit multi-view geometric constraints in their optimization layers, they are affected by the scale ambiguity of traditional MBSfM and cannot generalize to multi-body scenes, as shown in Fig. 1.

4. Method

Our method, illustrated in Fig. 2, is structured in three steps. First, given the unstructured images $\{\mathbf{I}_i\}_{i=1}^n$, we perform motion segmentation from image pairs α to estimate essential matrices $\{\mathbf{E}_k^\alpha\}_{j=1}^\mu$ that encode the μ rigid motions in the scene. Second, we fix the scale ambiguity of each $\mathbf{E}_k^\alpha$ by exploiting cues provided by monocular depth maps $\{\mathbf{M}_i\}_{i=1}^n$ in a robust scale estimation module. Finally, a multi-view depth and camera pose estimation network leverages our novel *multi-body* plane sweep algorithm to regress geometrically-consistent depth maps $\{\mathbf{D}_i\}_{i=1}^n$ and camera poses $\{\mathbf{R}_i, \mathbf{t}_i\}_{i=1}^n$ by considering all the μ motions.

4.1. Motion segmentation

The goal of motion segmentation is to recover the essential matrices $\{\mathbf{E}_k^\alpha\}_{k=1}^\mu$ that encode all rigid motions in the scene in any image pair $\alpha = (\mathbf{I}_i, \mathbf{I}_j)$. To this end, we combine traditional sparse motion segmentation, based on SIFT [19], with dense deep optical flow matches.

For each image $\mathbf{I}_i$, we extract $\{x_{i,u}\}_{u=1}^{p_i}$ SIFT keypoints and match them across image pairs. Then, for each pair α, we fit μ essential matrices $\{\mathbf{E}_k^\alpha\}_{k=1}^\mu$ to the matches using RPA [21], a multi-model fitting algorithm. RPA provides a labeling ℓ^α that assigns a matched keypoint to the motion it belongs to. Labels assigned to different image pairs may be inconsistent, thus we use permutation synchronization [1] to enforce a global labelling ℓ that assigns to keypoints belonging to the same motion the same label in all views.

In multi-body scenes, moving objects can be supported by only a few keypoints. Therefore, to refine the essential matrices, we augment the set of sparse matches with dense ones computed from the optical flow. Similarly to [34], we consider the dense matches at keypoint locations and, thus, avoid the typical inaccuracies of the optical flow caused by illumination changes and texture-less regions. In particular, given an image pair $\alpha = (\mathbf{I}_i, \mathbf{I}_j)$, for each object $\beta_k \in \mathcal{B}$, we consider $U_k^i = \{x_{i,u} : \ell(x_{i,u}) = k\}$ the sparse keypoints in $\mathbf{I}_i$ labelled as k. Around each keypoint $x \in U_k^i$ we consider a squared neighborhood $N(x)$ of 3×3 pixels. The union of neighborhoods $S_k^i = \bigcup_{x \in U_k^i} N(x)$ defines a mask in $\mathbf{I}_i$ as illustrated in Fig. 2. We refine $\mathbf{E}_k^\alpha$ by RANSAC equipped with the 5-point algorithm [22, 18] on the set of all

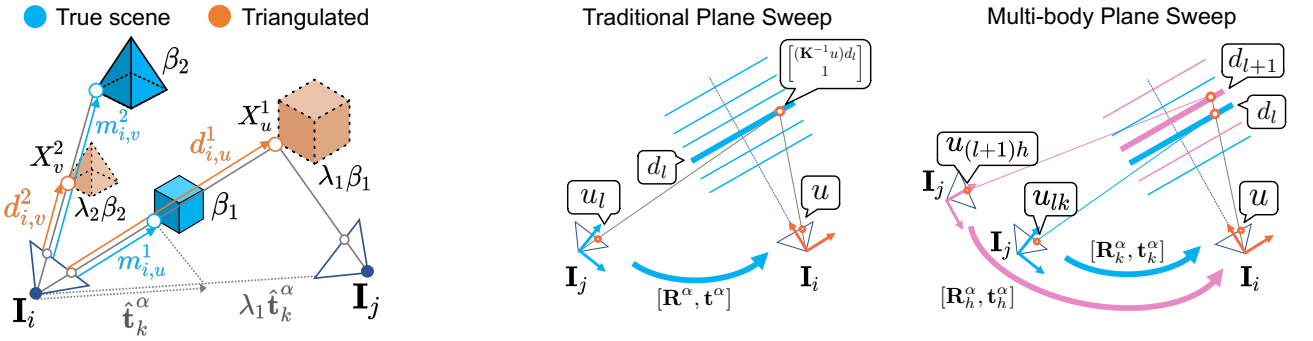

Figure 3: (*left*) In a *multi-body* scene, the scale of triangulated structures (orange) differs from the true object scale (cyan), leading to inconsistent relative object scales, i.e., cube bigger than pyramid. The tentative scale factor $\lambda^1_{i,u}$ of $X^1_u \in \beta_1$ is the ratio of the monocular depth $m^1_{i,u}$ and its triangulated depth $d^1_{i,u}$. (*right*) Traditional plane sweep [5] vs. our novel multi-body plane sweep.

the dense matches corresponding to the mask S^i_k. The optical flow is computed using the DICL-Flow network [35].

4.2. Robust Scale Estimation

We factorize each essential matrix $\mathbf{E}^\alpha_k = [\tilde{\mathbf{t}}^\alpha_k]_\times \tilde{\mathbf{R}}^\alpha_k$ to obtain the relative camera pose as a rotation $\tilde{\mathbf{R}}^\alpha_k$ and translation $\tilde{\mathbf{t}}^\alpha_k$. Since each $\tilde{\mathbf{t}}^\alpha_k$ is recovered only up to an unknown scale-factor λ_k, each moving object in a *multi-body* scene is reconstructed in its own arbitrary scale λ_k. Specifically, given an image pair α, we can triangulate from matched image points a set of 3D points $\{X^k_u\}_{u=1}^{p_k}$ belonging to object β_k from the up-to-scale camera pose $\{\tilde{\mathbf{R}}^\alpha_k, \tilde{\mathbf{t}}^\alpha_k\}$. However, these 3D points are expressed in the scale λ_k of $\tilde{\mathbf{t}}^\alpha_k$, meaning that different objects refer to inconsistent scales. Fig. 3-left depicts a *multi-body* scene where the scale ambiguity manifests as an inconsistent relative scale between objects $\lambda_1\beta_1$ and $\lambda_2\beta_2$ in the triangulated scene (orange) w.r.t. the actual 3D scene (cyan). Thus, the cube $\lambda_1\beta_1$ appears larger than the pyramid $\lambda_2\beta_2$ in the reconstruction.

We estimate the unknown scale factor λ_k for each object β_k by first regressing a monocular depth map $\mathbf{M}_i$ from each image $\mathbf{I}_i$ using AdaBins [2]. This network processes each $\mathbf{I}_i$ individually and, thus, is not affected by the multi-view scale ambiguity. We denote by $d^k_{i,u}$ and by $m^k_{i,u}$ the triangulated and monocular depths respectively of the u-th 3D point X^k_u w.r.t. $\mathbf{I}_i$. Then, for each object β_k and image $\mathbf{I}_i$, we compute the *point-wise* scale factor $\lambda^k_{i,u} = d^k_{i,u}/m^k_{i,u}$ that approximates the real λ_k according to the triangulated and monocular depths of X^k_u.

As monocular depth maps generally lack in accuracy, we devise a voting scheme to robustly aggregate the scale factors from all images $\{\mathbf{I}_i\}_{i=1}^n$ and unify all the μ camera poses under a global scale factor λ. First, we partition the point-wise scale factors computed from all the images into μ sets $\Lambda_k = \{\lambda^k_{i,u}\}^{n,p_k}_{i=1,u=1}$ according to their object β_k. Then, to promote robustness, we adopt a Kernel Density Estimator and derive a probability density function ϕ_k from each Λ_k. We use a Gaussian Kernel $\mathcal{K}$ with bandwidth h set at 5% of the median of the elements in Λ_k and define the kernel ϕ_k as follows:

$$\phi_k(x) = \sum_{\lambda^k_{i,u} \in \Lambda_k} \frac{\mathcal{K}(\lambda^k_{i,u} - x)}{h}. \tag{1}$$

The highest peak of ϕ_k is denoted by $\hat{\lambda}_k$ and represents the estimate of the k-th object scale factor λ_k. The estimated $\{\hat{\lambda}_k\}_{k=1}^\mu$ are used to scale the pairwise relative camera poses $[\tilde{\mathbf{R}}^\alpha_k, \hat{\lambda}^{-1}_k \tilde{\mathbf{t}}^\alpha_k]$ by multiplying the translation component. Now, camera poses are aligned to a global scale factor λ. As depths $\mathbf{M}_i$ are regressed in real-world unit of measurement, $\lambda \approx 1$, i.e., camera poses should be approximately in their real-world scale. We denote the correctly scaled camera poses as $[\hat{\mathbf{R}}^\alpha_k, \hat{\mathbf{t}}^\alpha_k]$ and use this unified camera configuration as the initialization for the next step.

4.3. Multi-body Plane Sweep Network

Although the poses $\{\hat{\mathbf{R}}^\alpha_k, \hat{\mathbf{t}}^\alpha_k\}$ are now aligned to a single scale factor, the depths are only estimated from monocular maps $\mathbf{M}_i$ which ignore the multi-view constraints. In principle, it could be possible to use camera poses to initialize a multi-view iterative scheme where depths and poses are refined, as in [36], but this would regress inconsistent depths in regions corresponding to moving objects. Thus, we design a novel multi-body depth and pose estimation network that takes the unordered images $\{\mathbf{I}_i\}_{i=1}^n$ and the scale-consistent poses $\{\hat{\mathbf{R}}^\alpha_k, \hat{\mathbf{t}}^\alpha_k\}$ to jointly estimates geometrically consistent depth maps $\{\mathbf{D}_i\}_{i=1}^n$ and refined poses $\{\mathbf{R}_i, \mathbf{t}_i\}_{i=1}^n$ in the reference frame of object β_1. As shown in Fig. 4, we include two separate branches for depth and pose estimation, which are used alternatively in an iterative scheme using the output of the other branch as input.

4.3.1. Multi-body Depth Estimation Branch

As depicted in Fig. 4, our depth estimation branch is based on our novel multi-body plane sweep and concatenates the following modules.

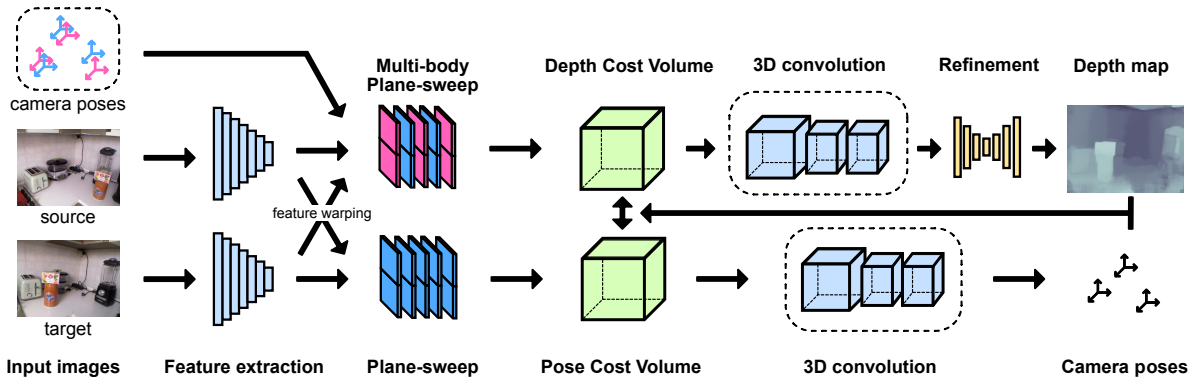

Figure 4: **Multi-body Plane Sweep network.** We extract CNN features from input images and build a cost volume using our *multi-body* plane sweep algorithm. Depth and camera poses are regressed from separate cost volumes with a series of 3D CNN and refinement layers.

Feature Extraction. A multi-scale CNN with spatial pyramid pooling [12] extracts features $\{\mathbf{F}_i\}_{i=1}^n$ from images $\{\mathbf{I}_i\}_{i=1}^n$. Each $\mathbf{F}_i$ has dimension C $\times$ W $\times$ H, where C, W, H are feature channels, width and height respectively.

Multi-body Plane Sweep for Cost Volume Construction. Let us briefly recall how the traditional plane sweep works before describing our contributions. The traditional plane sweep algorithm [5] takes as input an image pair α of a *source* $\mathbf{I}_i$ and *target* $\mathbf{I}_j$ and their relative pose $[\mathbf{R}^\alpha, \mathbf{t}^\alpha]$. The algorithm samples a set of virtual planes parallel to $\mathbf{I}_j$ at fixed depths $\{d_l\}_{l=1}^L$. Then, for each depth d_l, the source $\mathbf{I}_i$ is warped onto the target $\mathbf{I}_j$ through the homography induced by the plane at d_l and the relative camera pose $[\mathbf{R}^\alpha, \mathbf{t}^\alpha]$, as in Fig. 3-right. A dense depth map is obtained by selecting, for each pixel, the depth d_l that minimizes a pixel-wise *photometric* dissimilarity between the target $\mathbf{I}_j$ and the warped source $\mathbf{I}_i$. Deep methods [13, 36, 34] revisited this framework, but measure *featuremetric* consistency between latent features $\{\mathbf{F}_i\}_{i=1}^n$ instead of photometric dissimilarity. To this end, warped source and target features are aggregated in a single *depth cost* volume from which depths are regressed through convolutional layers. Unfortunately, both traditional and deep plane sweep methods assume static scenes with a single relative camera pose $[\mathbf{R}^\alpha, \mathbf{t}^\alpha]$ between the pair α. As a result, any point belonging to a moving object has an inconsistent depth.

We propose instead a *multi-body* plane sweep that considers *all* the μ relative poses $\{\hat{\mathbf{R}}_k^\alpha, \hat{\mathbf{t}}_k^\alpha\}_{k=1}^\mu$ to warp a source feature $\mathbf{F}_i$ onto a target $\mathbf{F}_j$ and construct the depth cost volume. Given a source-target pair $\alpha = (\mathbf{I}_i, \mathbf{I}_j)$, we uniformly sample L 3D virtual planes at depths $\{d_l\}_{l=1}^L$ parallel to the source $\mathbf{I}_i$. Then, we sequentially assign each depth plane to one of the μ relative camera poses by cycling through the $\{\hat{\mathbf{R}}_k^\alpha, \hat{\mathbf{t}}_k^\alpha\}_{k=1}^\mu$. Thus, the individual features $\mathbf{F}_i$ of the object β_k are warped onto $\mathbf{F}_j$ in a geometrically consistent manner when considering any depth plane assigned to the camera pose $[\hat{\mathbf{R}}_k^\alpha, \hat{\mathbf{t}}_k^\alpha]$ relative to β_k. This is depicted in Fig. 3-right, utilizing color to illustrate the cyclic assignment of the k-th and h-th rigid motions to planes at d_l and d_{l+1}, respectively, and to subsequent planes. Finally, each point u in the source feature $\mathbf{F}_i$ is first back-projected onto the virtual planes at d_l and then re-projected as $\tilde{u}_{lk}$ on the target $\mathbf{F}_j$ as follows:

$$\tilde{\mathbf{F}}_{il}^k(u) = \mathbf{F}_i(\tilde{u}_{lk}), \quad \tilde{u}_{lk} \sim \mathbf{K}\left[\hat{\mathbf{R}}_k^\alpha | \hat{\mathbf{t}}_k^\alpha\right] \begin{bmatrix} (\mathbf{K}^{-1}u)d_l \\ 1 \end{bmatrix}, \qquad (2)$$

where $\tilde{\mathbf{F}}_{il}^k$ is the warped source feature through the homography induced by the plane at d_l and the motion $[\hat{\mathbf{R}}_k^\alpha, \hat{\mathbf{t}}_k^\alpha]$.

Cost volume Construction. Given the set of virtual depth planes $\{d_l\}_{l=1}^L$, for each d_l, we obtain a warped source feature $\tilde{\mathbf{F}}_{il}^k$ using Eq. 2 and concatenate it to the target feature $\mathbf{F}_j$. This yields a set of L feature blocks, each sized 2C $\times$ W $\times$ H, that we arrange into a 4D depth cost volume sized 2C $\times$ L $\times$ W $\times$ H, as illustrated in Fig. 5. Since the number of virtual planes is significantly higher than the number of motions, *all* the μ motions are assigned to some virtual planes. Thus, our cost volume jointly incorporates all the motions in the scene, unlike traditional plane-sweep methods that assume a single motion.

Cost volume Regularization. Regularizing the cost volume is fundamental to cope with imperfect latent feature matching, which is especially common in image regions corresponding to texture-less objects or uniform pixel intensity. To this end, we use a sequence of 3D convolutional layers consisting of $L/2$ filters of size $3\times3\times3$ with stride 1, and residual connections. Then, we apply a 3D convolution with a single $3 \times 3 \times 3$ filter to obtain a 3D cost volume of size $L\times$W$\times$H. Since the order of the stacked features in the cost volume follows the correct virtual plane sequence, 3D convolutional layers can capture the 3D dependencies in the scene to enable consistent depth map estimation. Finally, we apply edge-preserving filtering to the resulting cost volume as in [13, 11]. In the case of multiple source-target image pairs, the final 3D cost volume is obtained by aver-

aging the 3D cost volumes from each pair. Further details on the cost volume can be found in the supplementary.

Depth Regression. We denote as c_l each slice of the aggregated cost volume, which corresponds to a plane at d_l. Then, we convert the aggregated cost volume to a probability volume as in [15] and use the softmax operation σ to normalize the probability volume across the depth dimension. We estimate the depth map $\mathbf{D}_i$ for a view $\mathbf{I}_i$ as follows:

$$\mathbf{D}_i = \frac{L \times d_{\text{min}}}{\tilde{l}}, \; \tilde{l} = \sum_{l=1}^{L} l \times \sigma(c_l), \tag{3}$$

where $\tilde{l}$ is the map corresponding to the average of virtual plane depths $l \in \{1, \dots, L\}$, weighted by the probabilities $\sigma(c_l)$. In (3), d_{min} denotes the depth of the closest plane to the source image.

4.3.2. Pose Estimation Branch

We describe our pose estimation branch, which optimizes current camera pose estimates through feature-metric consistency. To this purpose, we use a network architecture akin to the one in [36]. The network receives a pair $\alpha = (\mathbf{I}_i, \mathbf{I}_j)$, the depth map $\mathbf{D}_i$ of $\mathbf{I}_i$ and $[\hat{\mathbf{R}}_1^\alpha, \hat{\mathbf{t}}_1^\alpha]$, the current camera pose estimate in the reference frame of β_1, to regress a refined $[\mathbf{R}_1^\alpha, \mathbf{t}_1^\alpha]$. We uniformly sample P tentative camera poses $\{\hat{\mathbf{R}}_{1p}^\alpha, \hat{\mathbf{t}}_{1p}^\alpha\}_{p=1}^P$ by perturbing the rotation and the translation of the initial poses $[\hat{\mathbf{R}}_1^\alpha, \hat{\mathbf{t}}_1^\alpha]$. The P poses are used to construct a *pose cost volume* by warping each point u in the source feature $\mathbf{F}_i$ to a point $\tilde{u}_p$ on the target feature $\mathbf{F}_j$ as follows:

$$\tilde{u}_p \sim \mathbf{K} \left[\hat{\mathbf{R}}_{1p}^\alpha \mid \hat{\mathbf{t}}_{1p}^\alpha\right] \begin{bmatrix} (\mathbf{K}^{-1}u)\mathbf{D}_i(u) \\ 1 \end{bmatrix}. \tag{4}$$

Finally, a sequence of 3D convolutional layers is applied to the pose cost volume to regress the camera pose $[\mathbf{R}_1^\alpha, \mathbf{t}_1^\alpha]$.

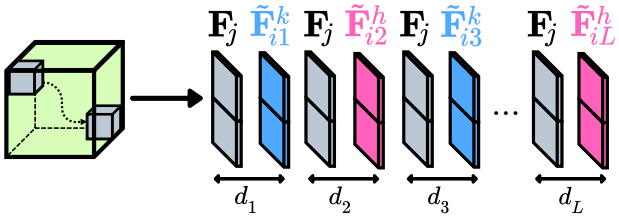

Figure 5: **Cost volume construction by multi-body plane sweep.** For each virtual depth plane at $d_l \in \{d_l\}_{l=1}^L$, we generate a $2C \times W \times H$ feature map by concatenating the target feature $\mathbf{F}_j$ (gray) to the warped source feature $\tilde{\mathbf{F}}_{il}^\gamma$ (in color) at depth d_l using motion $[\hat{\mathbf{R}}_\gamma^\alpha, \hat{\mathbf{t}}_\gamma^\alpha]$. The 4D cost volume is constructed through concatenation of these L feature blocks. $\tilde{\mathbf{F}}_{il}^k$ (in blue) indicates a source feature $\mathbf{F}_i$ warped onto $\mathbf{F}_j$ through the homography induced by the generic plane at d_l and its relative camera motion $[\hat{\mathbf{R}}_k^\alpha, \hat{\mathbf{t}}_k^\alpha]$. Instead, $\tilde{\mathbf{F}}_{il}^h$ is induced by the motion $[\hat{\mathbf{R}}_h^\alpha, \hat{\mathbf{t}}_h^\alpha]$ (in pink).

	view	Method	lower is better ↓ Abs Rel	Sq Rel	RMSE	RMSE_{log}	higher is better ↑ δ_1	δ_2	δ_3
Eigen SfM (s)	sv	DORN [9]	0.067	0.295	2.929	0.108	0.949	0.988	0.995
		AdaBins [2]	0.054	0.182	2.341	0.087	0.966	0.995	0.999
	mv	DeepV2D [31]	0.050	0.212	2.483	0.089	0.973	0.992	0.997
		Wang et al. [34]	**0.034**	**0.103**	**1.919**	**0.057**	**0.989**	**0.998**	0.999
		Ours	0.044	0.148	2.125	0.068	0.974	0.996	0.999
Eigen (s & m)	sv	DORN [9]	0.072	0.307	2.727	0.120	0.932	0.984	0.994
		AdaBins [2]	0.058	0.190	2.360	0.088	0.964	**0.995**	**0.999**
	mv	SfMLearner [39]	0.208	1.768	6.856	0.283	0.678	0.885	0.975
		CCNet [25]	0.140	1.070	5.326	0.217	0.826	0.941	0.975
		BANet [29]	0.083	-	3.640	0.134	-	-	-
		DeepV2D [31]	0.064	0.350	2.946	0.120	0.946	0.982	0.991
		Wang et al. [34]	0.055	0.224	2.273	0.091	0.956	0.984	0.993
		Ours	**0.054**	**0.186**	**2.269**	**0.086**	**0.966**	0.991	0.996
Eigen MB (m)	sv	DORN [9]	0.072	0.307	2.727	0.120	0.932	0.984	0.994
		AdaBins [2]	0.062	0.201	2.372	0.092	0.959	**0.994**	**0.999**
	mv	DeepV2D [31]	0.089	0.401	3.278	0.152	0.931	0.965	0.989
		Wang et al. [34]	0.063	0.242	2.347	0.109	0.948	0.971	0.991
		Ours	**0.057**	**0.197**	**2.301**	**0.090**	**0.962**	0.991	0.994

Table 1: **Depth evaluation on KITTI Depth.** Results from single-view (sv) and multi-view (mv) depth estimation methods evaluated under two-view SfM setting. Best results in **bold**.

4.3.3. Training and Inference

We train the feature extractor, 3D convolutions and regression layers in a end-to-end supervised manner. We denote as $\mathbf{D}$ the depth regressed from the refined cost volume, and as $\mathbf{D}_{\text{gt}}$ the ground-truth depth. To improve the learning process, we also regress a depth map $\tilde{\mathbf{D}}$ from the cost volume before applying edge-preserving filtering. The loss function is defined as $\mathcal{L}_{\text{depth}} = \sum_i \lambda H(\tilde{\mathbf{D}}, \mathbf{D}_{\text{gt}}) + H(\mathbf{D}, \mathbf{D}_{\text{gt}})$, where $\lambda = 0.7$ is a weight and $H(\cdot)$ is the Huber loss. The pose losses $\mathcal{L}_{\text{rot}}$ and $\mathcal{L}_{\text{trans}}$ are defined as the ℓ_1 distance between the predicted and ground-truth absolute poses. Our final loss is then: $\mathcal{L} = \lambda_r \mathcal{L}_{\text{rot}} + \lambda_t \mathcal{L}_{\text{trans}} + \lambda_d \mathcal{L}_{\text{depth}}$. All the weights are specified in the supplementary.

The *multi-body* plane sweep component of the network has no learnable parameters and is not trained. Our network does not require any form of motion annotation for depth estimation, even in multi-body scenes. For this reason, we train our model on static frames from KITTI [10] and DeMoN [33]. The training and inference are performed in an iterative fashion in four steps, where the regressed depth maps and camera poses are fed to the next iteration.

5. Experiments

We evaluate our method on *static* (s) and *multi-body* (m) datasets for depth and pose estimation against state-of-the-art single-view (sv) and multi-view (mv) methods. In Sec. 5.4 we discuss the limitations of our method and possible countermeasures. Additional experiments and implementation details are provided in the supplementary.

5.1. Datasets, Competitors and Metrics

KITTI Depth (s & m) [10] is designed for depth evaluation in autonomous driving and contains several image se-

Figure 6: **Multi-body Unstructured Extract.** Three view pairs extracted from our proposed dataset for illustration purposes. The dataset contains indoor sequences that depict varied scenarios in which objects of several different sizes move in the scene.

	MVS					Scenes11					SUN3D				
	Depth			Pose		Depth			Pose		Depth			Pose	
Method	L1-inv	Sc-inv	L1-rel	$\mathbf{R}_{\text{err}}$	t_{err}	L1-inv	Sc-inv	L1-rel	$\mathbf{R}_{\text{err}}$	t_{err}	L1-inv	Sc-inv	L1-rel	$\mathbf{R}_{\text{err}}$	t_{err}
AdaBins [2]	0.048	0.236	0.291	-	-	0.020	0.382	0.270	-	-	0.021	0.127	0.161	-	-
DeMoN [33]	0.047	0.202	0.305	5.156	14.447	0.019	0.315	0.248	0.809	8.918	0.019	0.114	0.172	1.801	18.811
LS-Net [4]	0.051	0.221	0.311	4.653	11.122	0.010	0.410	0.210	4.653	8.210	0.015	0.189	0.650	1.521	14.347
BA-Net [29]	0.030	0.150	0.080	3.499	11.238	0.080	0.210	0.130	3.499	10.370	0.015	0.110	0.060	1.729	13.260
DeepSfM [36]	0.021	0.129	0.079	2.824	9.881	0.007	0.112	0.064	0.403	5.828	0.013	0.093	0.072	1.704	13.107
Wang et al. [34]	**0.015**	**0.102**	0.068	**2.417**	**3.878**	0.005	**0.097**	0.058	**0.276**	**2.041**	**0.010**	**0.081**	**0.057**	**1.391**	**10.757**
Ours ($M = 1$)	0.016	0.107	0.068	2.538	4.538	0.005	0.099	0.058	0.321	3.649	0.011	0.084	0.061	1.470	12.018
Ours ($M = 2$)	0.019	0.121	0.073	2.681	7.340	0.007	0.102	0.069	0.401	4.619	0.013	0.092	0.071	1.625	13.402
Ours ($M = 3$)	0.025	0.132	0.075	2.892	9.530	0.009	0.124	0.078	0.542	5.782	0.014	0.095	0.079	1.769	15.231
Ours ($M = 4$)	0.026	0.134	0.076	2.931	9.741	0.009	0.128	0.081	0.571	5.803	0.016	0.107	0.084	1.830	15.904

Table 2: **Depth and pose evaluation on MVS, Scenes 11, SUN3D. (s)**. For all metrics lower is better. Best results are in **bold**.

quences where objects move rigidly. We consider the Eigen [7] split (697 frames) and the Eigen SfM split [34] (256 frames), a subset of the Eigen split without moving objects. In addition, we introduce the Eigen MB split, a subset of the Eigen split that includes 90 frames in which dynamic objects appear and camera motions are well-conditioned.

MVS, Scenes11, SUN3D. (s) MVS [33] includes outdoor sequences with large baselines. Scenes11 [33] contains fairly realistic frames from rendered scenes with accurate ground truth depth and camera poses. SUN3D [37] is an indoor dataset with sometimes inaccurate ground truth depth and camera poses. For SUN3D, we consider the split by [33] to exclude frames with high photoconsistency errors.

ETH3D SLAM. (m) [28] contains SLAM sequences with ground truth depth and camera poses. We consider the dynamic sequences *motion1-2-3-4* and, specifically, a 32 frames split where camera motions are well-conditioned.

Multi-body Unstructured. (m) We capture multi-body indoor sequences using a Kinect with ground truth depth. The dataset is meant for evaluation in varied conditions, for instance, when image sets are unstructured or when both large and big objects move in the scene. We use RGB-D SLAM [8] to fuse visual information and data from the Kinect IMU and annotate accurate camera poses. The dataset includes 42 frames with either one or two moving objects. Samples scenes are reported in Fig.6 and additional details are provided in the supplementary.

Competitors. We compare against state-of-the-art single-view methods (sv) DORN [9] and AdaBins [2], and multi-view methods (mv) DeMoN [33], LS-Net [3], CCNet [25], BANet [29], DeepV2D [31] and Wang et al. [34]. DeMoN estimates depth and poses from image pairs, CCNet and SfMLearner mask moving objects before depth and pose estimation, LS-Net, BANet and DeepV2D optimize multi-view differentiable cost functions to minimize photo- or feature-metric errors, and Wang et al. [34] proposes a deep, scale-invariant estimator for depth and camera poses that robustly identify the dominant background motion β_1.

Metrics. To evaluate the quality of depth maps in KITTI Depth, we adopt the metrics in [7], i.e., the depth absolute (Abs Rel) and squared (Sq Rel) relative difference, the RMSE, the $\text{RMSE}_{\log}$, and the thresholds $\{\delta_i\}_{i=1}^{3}$. For other datasets, we adopt the metrics in [33], i.e., scale-invariant depth error (sc-inv), relative error to ground truth depth (L1-rel), relative error w.r.t. inverse depth (L1-inv). The definitions of these metrics are in the supplementary. As for pose estimation, we report the angle (in deg) between predicted and ground truth translations (t_{err}) and rotations ($\mathbf{R}_{\text{err}}$) of the cameras with respect to the background motions β_1.

Implementation Details. We fine-tune the optical flow network on KITTI Depth for a fair comparison to [34]. Otherwise, we train the optical flow on synthetic scenes, as in [35], and let the proposed framework refine initial camera poses. We implement our framework using PyTorch. The batch size is set to 32. The learning rate is set to 1×10^{-4} and is halved after 2 epochs. For KITTI Depth, the feature extractor is initialized with pre-trained weights, which are frozen for the first epoch. For other datasets, we train the network from randomly initialized weights. Training takes approximately 3 days on 2 NVIDIA RTX A6000 GPUs. Further details are reported in the supplementary.

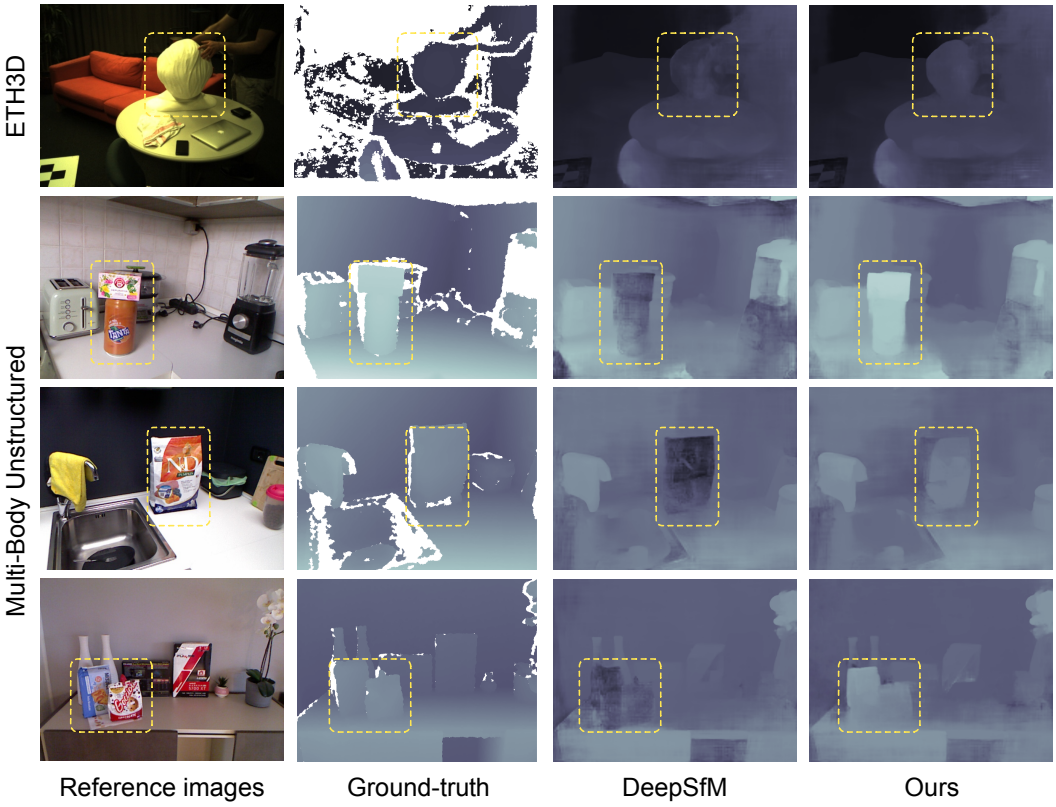

Figure 7: **Qualitative comparisons on ETH3D and Multi-body Unstructured.** (m) The yellow boxes highlight moving objects reconstructed by our method but not by DeepSfM [36].

5.2. Static evaluation (s)

KITTI Depth. The Eigen SfM block of Tab. 1 reports quantitative results for static depth estimation. Our method beats the single-view methods [9, 2] and also the multi-view DeepV2D [31] in all metrics and with a clear margin. Yet Wang et al. [34] achieves better performance on this static scenes, although it recovers only up-to-scale depth maps.

MVS, Scenes11, SUN3D. Quantitative results are shown in Tab. 2. Our method beats the single-view [9, 2] and multi-view [33, 4, 29] in most metrics, but achieves results that are on par or slightly worse compared to DeepSfM [36], since we rely on monocular depth cues that are less accurate than the DeMoN depths used in DeepSfM. The best results are achieved by [34]. In the supplementary, we show that our method beats geometric SfM baselines. By setting $M = 1$ and leveraging the static scene assumption, we outperform [36] and are similar to [34]. However, when the scene is static and $M \geq 2$ motions are considered, our method produces outlying essential matrices that may negatively impact depth estimation. Nonetheless, we achieve good results for $M \geq 2$ and even outperform [36] in the multi-body configuration $M = 2$. For $M \geq 2$, the performance drop is limited, indicating that M can be larger without affecting depth accuracy substantially.

5.3. Multi-body evaluation (m)

KITTI Depth. The Eigen MB and Eigen blocks of Tab. 1 report quantitative results for depth estimation in the *multi-body* setting. As opposed to the static case, our method beats its multi-view competitors in both the considered splits. Specifically, we observe the most significant margin in the Eigen MB split, in which dynamic objects negatively affect the quality of static multi-view depth estimators [30, 34]. This is evident in the highlighted portion of Fig. 8, where our method accurately reconstructs the moving vehicle, whereas the state-of-the-art Wang et al. [34] exhibits significant artifacts. As in the static evaluation, our framework beats monocular approaches [2, 9] in most metrics, showing it effectively leverage multi-view constraints for improved depth estimation accuracy. Less strict thresholds δ_2, δ_3 generally see better results for single-view methods compared to multi-view due to better handling of occlusions and image regions with uniform pixel intensity. However, monocular methods perform worse on the most strict threshold δ_1 and the other accuracy-oriented metrics.

ETH3D and Multi-Body Unstructured. Results on these multi-body datasets are reported in Tab. 3. Our method produces more accurate depth and camera pose estimates than its competitors in all metrics. Specifically, our method out-

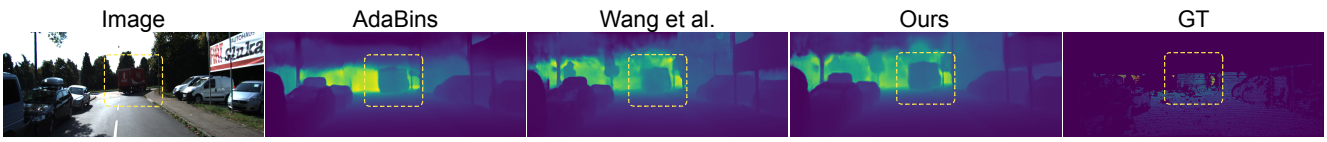

Figure 8: **Qualitative results on KITTI (m).** The yellow box highlights a moving vehicle that is captured more accurately by our method.

	ETH3D					Multi-body Unstructured				
	Depth			Pose		Depth			Pose	
Method	L1-inv	Sc-inv	L1-rel	$\mathbf{R}_{err}$	t_{err}	L1-inv	Sc-inv	L1-rel	$\mathbf{R}_{err}$	t_{err}
DeMoN[33]	0.38	0.41	0.33	5.63	12.04	0.52	0.71	0.41	4.90	9.26
AdaBins[2]	0.35	0.39	0.27	-	-	0.48	0.63	0.38	-	-
DeepSfM[36]	0.12	0.14	0.10	**2.80**	9.64	0.22	0.22	0.18	2.38	5.03
Wang et al.[34]	0.21	0.19	0.17	3.26	9.04	0.27	0.22	0.19	2.79	7.67
Ours (M=1)	0.11	0.14	0.10	2.98	9.58	0.21	0.22	0.18	2.21	5.01
Ours (M=2)	0.09	0.12	0.08	2.91	8.53	0.19	0.20	0.17	1.99	4.91
Ours (M=3)	**0.08**	**0.11**	**0.08**	2.94	8.28	**0.17**	**0.19**	**0.16**	**1.94**	4.72
Ours (M=4)	0.09	**0.11**	0.09	2.94	**8.23**	0.17	0.20	0.17	**1.94**	**4.70**

Table 3: **Depth and pose evaluation on multi-body datasets (m).** For all metrics, lower is better. Best results are in **bold**.

performs DeMoN [33] by a significant margin and, when compared to DeepSfM [36], attains better results in depth estimation in both datasets. This is shown by the qualitative results of Fig. 7, where depth maps from DeepSfM exhibit noticeable break-ups in the areas corresponding to the moving objects. Our pose estimation is also more accurate, except for the $\mathbf{R}_{\text{err}}$ over the ETH3D dataset [28]. Overall, we attribute this difference to: *i*) DeepSfM suffers from poor initialization from DeMoN, which is not accurate in multi-body scenes and hinders the overall quality of the reconstruction, *ii*) the plane sweep algorithm adopted by DeepSfM constructs cost volumes that do not satisfy the geometric constraints for the moving objects. Our method also beats Wang et al. [34] in all metrics by a clear margin. As expected, the performance of Wang et al. suffers significantly in multi-body scenes. The moving objects are treated as outliers and negatively impact the accuracy of the estimated camera poses, which, in turn, undermine the quality of depth estimation. By segmenting the rigid motions in the scene before computing camera poses, our method exploits and is robust to the moving objects and achieves state-of-the-art performance in multi-body depth estimation. The effectiveness of multi-body plane sweep (PS) is evidenced by the uplift in performance for $M \geq 2$ in Tab. 3. Note that for $M = 1$, a single motion is considered and MBPS specializes to traditional PS, with similar results to [36].

5.4. Discussion and Limitations

Our method inherits some known limitations of the multi-view approaches. In this section we discuss their possibile mitigations.

Small camera motions may hinder initial pose estimates. Our pose refinement network mitigates poor initializations, but it may not recover in particularly challenging cases.

Occlusions are mitigated, as in [13, 36], by aggregating cost volumes from multiple image pairs, if available. However, with only two images, occlusions result in noisy depth maps in the region of the occluded objects, as seen in the second sequence of Fig. 7.

Monocular depth inaccuracies may hinder the estimation of object scale factors. However, we observed our KDE voting scheme can isolate outlying measurements and return accurate results when enough (~ 10) factors are computed.

Non-rigid scenes. Our method is not designed for non-rigid scenes, for which different assumptions are used, e.g. temporal coherence or complete reliance on monocular depth for moving objects. Nevertheless, as shown in the supplementary, our method can reconstruct articulated motions, provided that a sufficient number of motions is considered.

Virtual depth planes. Our multi-body plane sweep assigns L/M depth planes to each motion, meaning that the more motions considered, the fewer depth planes are assigned to each motion. As stated in the supplementary, we consider up to four motions in our evaluation, as increasing the number of depth planes worsens training and inference times.

Inference times are on average $\sim 20\%$ higher in KITTI Depth with respect to [34] due to the required motion segmentation step, as discussed in the supplementary.

6. Conclusion

In this paper, we have addressed the problem of multi-body depth and camera pose estimation. We overcome the scale-ambiguity problem typical of all the MBSfM approaches by introducing a learning-based robust voting scheme to unify the scales of all independently moving objects in the scene. In addition, we overcome the static-scene assumption of deep SfM approaches thanks to a novel multi-body plane sweep network that explicitly supports the additional multi-view geometric constraints derived from the multiple bodies. Extensive experiments on multi-body datasets show that our method outperforms state-of-the-art deep learning methods qualitatively and quantitatively.

Acknowledgements. This paper is supported by FAIR (Future Artificial Intelligence Research) project, funded by the NextGenerationEU program within the PNRR-PE-AI scheme (M4C2, Investment 1.3, Line on Artificial Intelligence). We also gratefully acknowledge *NVIDIA Corporation* for the GPUs donated within the Academic and Applied Research Program to *Giacomo Boracchi* and *Luca Magri*

References

[1] Federica Arrigoni, Elisa Ricci, and Tomas Pajdla. Multi-frame motion segmentation by combining two-frame results. *International Journal of Computer Vision*, 130(3):696–728, 2022. 1, 2, 3

[2] Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. Adabins: Depth estimation using adaptive bins. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4009–4018, 2021. 1, 2, 4, 6, 7, 8, 9

[3] Ronald Clark, Michael Bloesch, Jan Czarnowski, Stefan Leutenegger, and Andrew J Davison. Learning to solve nonlinear least squares for monocular stereo. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 284–299, 2018. 7

[4] Ronald Clark, Michael Bloesch, Jan Czarnowski, Stefan Leutenegger, and Andrew J Davison. Ls-net: Learning to solve nonlinear least squares for monocular stereo. *arXiv preprint arXiv:1809.02966*, 2018. 1, 3, 7, 8

[5] Robert T Collins. A space-sweep approach to true multi-image matching. In *Proceedings CVPR IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 358–363. Ieee, 1996. 4, 5

[6] Joao Costeira and Takeo Kanade. A multi-body factorization method for motion analysis. In *Proceedings of IEEE International Conference on Computer Vision*, pages 1071–1076. IEEE, 1995. 2

[7] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *Advances in neural information processing systems*, 27, 2014. 7

[8] Felix Endres, Jürgen Hess, Jürgen Sturm, Daniel Cremers, and Wolfram Burgard. 3-d mapping with an rgb-d camera. *IEEE transactions on robotics*, 30(1):177–187, 2013. 7

[9] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2002–2011, 2018. 1, 2, 6, 7, 8

[10] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012. 6

[11] Kaiming He, Jian Sun, and Xiaoou Tang. Guided image filtering. *IEEE transactions on pattern analysis and machine intelligence*, 35(6):1397–1409, 2012. 5

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 37(9):1904–1916, 2015. 5

[13] Sunghoon Im, Hae-Gon Jeon, Stephen Lin, and In So Kweon. Dpsnet: End-to-end deep plane sweep stereo. *arXiv preprint arXiv:1905.00538*, 2019. 1, 3, 5, 9

[14] Sebastian Hoppe Nesgaard Jensen, Mads Emil Brix Doest, Henrik Aanæs, and Alessio Del Bue. A benchmark and evaluation of non-rigid structure from motion. *International Journal of Computer Vision*, 129(4):882–899, 2021. 2

[15] Alex Kendall, Hayk Martirosyan, Saumitro Dasgupta, Peter Henry, Ryan Kennedy, Abraham Bachrach, and Adam Bry. End-to-end learning of geometry and context for deep stereo regression. In *Proceedings of the IEEE international conference on computer vision*, pages 66–75, 2017. 6

[16] Suryansh Kumar. Non-rigid structure from motion: Prior-free factorization method revisited. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 51–60, 2020. 2

[17] Abhijit Kundu, K Madhava Krishna, and CV Jawahar. Real-time multibody visual slam with a smoothly moving monocular camera. In *2011 International Conference on Computer Vision*, pages 2080–2087. IEEE, 2011. 2

[18] Hongdong Li and Richard Hartley. Five-point motion estimation made easy. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 1, pages 630–633. IEEE, 2006. 3

[19] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004. 3

[20] Xuan Luo, Jia-Bin Huang, Richard Szeliski, Kevin Matzen, and Johannes Kopf. Consistent video depth estimation. *ACM Transactions on Graphics (ToG)*, 39(4):71–1, 2020. 2

[21] Luca Magri and Andrea Fusiello. Multiple structure recovery via robust preference analysis. *Image and Vision Computing*, 67:1–15, 2017. 3

[22] David Nistér. An efficient solution to the five-point relative pose problem. *IEEE transactions on pattern analysis and machine intelligence*, 26(6):756–770, 2004. 3

[23] Kemal Egemen Ozden, Kurt Cornelis, Luc Van Eycken, and Luc Van Gool. Reconstructing 3d trajectories of independently moving objects using generic constraints. *Computer Vision and Image Understanding*, 96(3):453–471, 2004. 2

[24] Kemal E Ozden, Konrad Schindler, and Luc Van Gool. Multibody structure-from-motion in practice. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(6):1134–1141, 2010. 2

[25] Anurag Ranjan, Varun Jampani, Lukas Balles, Kihwan Kim, Deqing Sun, Jonas Wulff, and Michael J Black. Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12240–12249, 2019. 6, 7

[26] Muhamad Risqi U Saputra, Andrew Markham, and Niki Trigoni. Visual slam and structure from motion in dynamic environments: A survey. *ACM Computing Surveys (CSUR)*, 51(2):1–36, 2018. 1, 2

[27] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

[28] Thomas Schops, Torsten Sattler, and Marc Pollefeys. Bad slam: Bundle adjusted direct rgb-d slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 134–144, 2019. 7, 9

[29] Chengzhou Tang and Ping Tan. Ba-net: Dense bundle adjustment network. *arXiv preprint arXiv:1806.04807*, 2018. 1, 3, 6, 7, 8

[30] Zachary Teed and Jia Deng. Deepv2d: Video to depth with differentiable structure from motion. *arXiv preprint arXiv:1812.04605*, 2018. 8

[31] Zachary Teed and Jia Deng. Deepv2d: Video to depth with differentiable structure from motion. In *International Conference on Learning Representations*, 2020. 3, 6, 7, 8

[32] Carlo Tomasi and Takeo Kanade. Shape and motion from image streams under orthography: a factorization method. *International journal of computer vision*, 9(2):137–154, 1992. 2

[33] Benjamin Ummenhofer, Huizhong Zhou, Jonas Uhrig, Nikolaus Mayer, Eddy Ilg, Alexey Dosovitskiy, and Thomas Brox. Demon: Depth and motion network for learning monocular stereo. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5038–5047, 2017. 3, 6, 7, 8, 9

[34] Jianyuan Wang, Yiran Zhong, Yuchao Dai, Stan Birchfield, Kaihao Zhang, Nikolai Smolyanskiy, and Hongdong Li. Deep two-view structure-from-motion revisited. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8953–8962, 2021. 1, 3, 5, 6, 7, 8, 9

[35] Jianyuan Wang, Yiran Zhong, Yuchao Dai, Kaihao Zhang, Pan Ji, and Hongdong Li. Displacement-invariant matching cost learning for accurate optical flow estimation. *Advances in Neural Information Processing Systems*, 33:15220–15231, 2020. 4, 7

[36] Xingkui Wei, Yinda Zhang, Zhuwen Li, Yanwei Fu, and Xiangyang Xue. Deepsfm: Structure from motion via deep bundle adjustment. In *European conference on computer vision*, pages 230–247. Springer, 2020. 1, 3, 4, 5, 6, 7, 8, 9

[37] Jianxiong Xiao, Andrew Owens, and Antonio Torralba. Sun3d: A database of big spaces reconstructed using sfm and object labels. In *Proceedings of the IEEE international conference on computer vision*, pages 1625–1632, 2013. 7

[38] Zhoutong Zhang, Forrester Cole, Zhengqi Li, Michael Rubinstein, Noah Snavely, and William T Freeman. Structure and motion from casual videos. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIII*, pages 20–37. Springer, 2022. 2

[39] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1851–1858, 2017. 6