Structure and Motion from Casual Videos

Zhoutong Zhang[1,2], Forrester Cole[2], Zhengqi Li[2], Noah Snavely[2,3],
Michael Rubinstein[2], and William T. Freeman[1,2]

[1] MIT CSAIL [2]Google Research [3] Cornell

Abstract. Casual videos, such as those captured in daily life using a hand-held camera, pose problems for conventional structure-from-motion (SfM) techniques: the camera is often roughly stationary (not much parallax), and a large portion of the video may contain moving objects. Under such conditions, state-of-the-art SfM methods tend to produce erroneous results, often failing entirely. To address these issues, we propose CasualSAM, a method to estimate camera poses and dense depth maps from a monocular, casually-captured video. Like conventional SfM, our method performs a joint optimization over 3D structure and camera poses, but uses a pretrained depth prediction network to represent 3D structure rather than sparse keypoints. In contrast to previous approaches, our method does not assume motion is rigid or determined by semantic segmentation, instead optimizing for a per-pixel motion map based on reprojection error. Our method sets a new state-of-the-art for pose and depth estimation on the Sintel dataset, and produces high-quality results for the DAVIS dataset where most prior methods fail to produce usable camera poses.

Keywords: Structure from motion, depth estimation, casual video

1 Introduction

Structure-from-motion (SfM) and related methods for 3D reconstruction from monocular video are considered a relatively mature technology. They work quite reliably for predominantly stationary scenes involving large camera motions, such as a video of a walkthrough of a house, or a video taken from a car driving down a street. Videos taken under “casual” conditions, however, often violate these assumptions. The operator is often standing roughly stationary, capturing moving subjects such as people and pets, and the video may only be a few seconds long. Under these conditions, state-of-the-art SfM systems often fail. Worse, when SfM does fail, it tends to fail spectacularly and produce useless results (Fig. 1).

In this paper, we introduce a new method for dense depth and 3D camera pose estimation designed for casual videos. Our method performs a joint optimization of cameras and 3D structure over the video, similar to methods based on bundle adjustment [33, 1]. But in contrast to sparse feature matches typically used in such approaches, our method optimizes dense 3D correspondences, following recent work that fine-tunes a pre-trained depth prediction network on the input video [18, 37]. While powerful, previous fine-tuning methods require known camera poses, and simply adding camera poses to the optimization produces poor results [16]. We reexamine this approach and show that with several careful modifications,

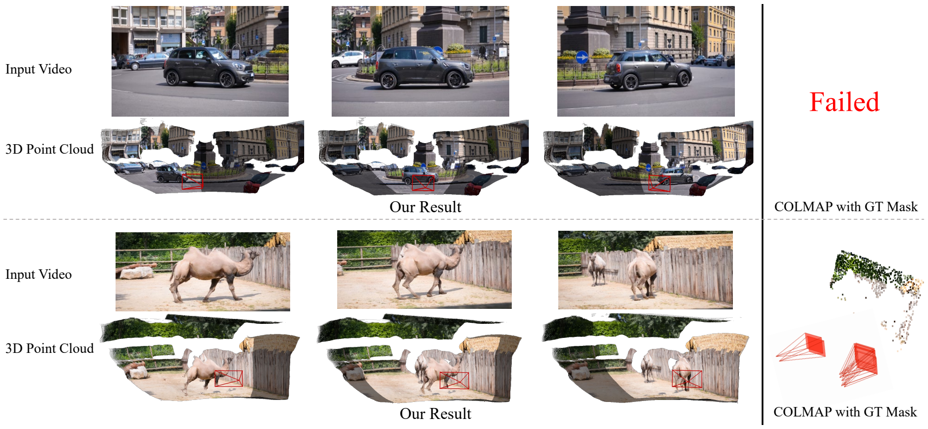


Fig. 1. Structure and motion for casual videos. Given a casually shot video with moving objects, our method estimates the structure and motion of the scene and the camera poses. Above: input video frame. Below: 3D reconstruction of estimated stationary points for the entire video and estimated moving points for the current frame. On these videos, the conventional SfM system COLMAP either fails to create a single camera track (top right), or produces incorrect results (bottom right). Note that COLMAP is provided GT motion masks for these examples.

camera poses and the depth network can be successfully optimized together. The pre-trained depth estimates disambiguate camera motion when parallax is small, and the fine-tuning process produces sharp, temporally-consistent depth maps, leading to previously unachievable quality on real-world videos (Fig 1).

One key innovation is to optimize for per-pixel movement maps that effectively modulate the training loss in moving areas. Unlike masks based on semantic segmentation [16, 6], the movement maps capture only the parts of the scene that are *currently* moving. These movement maps, for example, allow multi-view constraints to be enforced for stationary people and vehicles. They also help relax the multi-view constrains for moving objects that semantic segmentation may miss, such as unusual animals and moving background "stuff" (such as trees swaying in the wind).

This method, which we name CasualSAM, achieves a new state-of-the-art for simultaneous depth prediction and camera pose estimation in dynamic scenes, while remaining competitive with prior methods on traditional SfM scenarios featuring stationary scenes. Our method is simple, involving a single, joint optimization with one reprojection loss term and one depth prior loss term. And, importantly, unlike conventional SfM systems that can fail catastrophically, our method gracefully degrades in performance on especially difficult videos. We benchmark the approach on the Sintel [5] and TUM RGBD [28] datasets, and show convincing qualitative results on DAVIS [24].

2 Background

Structure from motion and visual SLAM. Estimating camera poses and scene structure from monocular video is a long-standing problem in computer vision and related areas. This problem is variously called structure from motion (SfM) or visual SLAM, depending on the precise inputs and constraints [8, 26, 21] (e.g., SLAM methods often operate in real time).

The classic approach to SfM is to compute sparse 2D correspondence across the input views, and optimize for the camera poses and sparse 3D point locations that minimize a reprojection error given these 2D observations [32]. Some visual SLAM methods instead seek to solve for dense or semi-dense depth per frame, and optimize directly based on image intensities rather than sparse correspondence [23, 14, 11, 10]. However, SfM methods based on reprojection or photometric error fundamentally depend on a static scene assumption. For dynamic scenes, the underlying epipolar constraints break down, leading to errors or failures. In addition, these methods require parallax, and face ambiguities between rotational and translational motion in the face of small camera motion.Previous work [36] recovers sparse depth measures from accidental motion, but is limited in reconstruction quality and focuses on static scenes only When faced with videos that feature a combination of small camera motion and dynamic scene motion, classic methods often produce completely erroneous results.

A variety of methods have integrated learned components into classic SfM or SLAM methods. Previous works [38, 20] explored jointly optimizing depth maps, camera poses and confidence masks for weighting the photo-metric loss during training. These methods focus on mostly static scenes, where the confidence mask is either intended for excluding out-of-view pixels, or based on a tuned prior that is sensitive for joint optimization. Notable recent examples include DROID-SLAM, which makes use of a learned bundle adjustment layer to update camera poses and dense depth maps [30], and D3VO, which leverages modules for learned prediction of depth, pose, and uncertainty within a bundle adjustment framework [35]. Like D3VO and other recent methods like CodeSLAM [3], our method leverages monocular depth prediction as a prior. However, unlike our method, these prior works tend to be designed for and evaluated on standard SLAM benchmarks (such as KITTI), that feature large camera motions and dynamic content that occupies a small fraction of the field of view (if it exists at all). We show our method, with its explicit handling of motion, leads to better performance on datasets like TUM that feature more dramatic scene motion.

Some methods attempt to directly model dynamic scene motion. One class of methods, exemplified by DynamicFusion, uses explicit depth sensing (e.g., with a Kinect sensor) to reduce the ill-posedness of the problem [22]. Other *non-rigid structure from motion* methods leverage monocular video and fit a low-dimensional model to the dynamic scene [31], but have trouble scaling to arbitrary video featuring arbitrary motion. Our method works from standard RGB videos and more robustly handles scene motion.

Test-time refinement of depth estimation. Recently, methods that finetune a pretrained depth prediction network on an input video have been shown to produce high-quality, temporally-consistent depth maps [7, 6, 18, 37]. In these methods, the optimized variables are the weights of a deep network that predicts the unknowns, rather than the unknowns themselves. A chief advantage of these methods is their ability to combine a monocular depth prior with an optimization across the entire video. These approaches can roughly be divided into methods

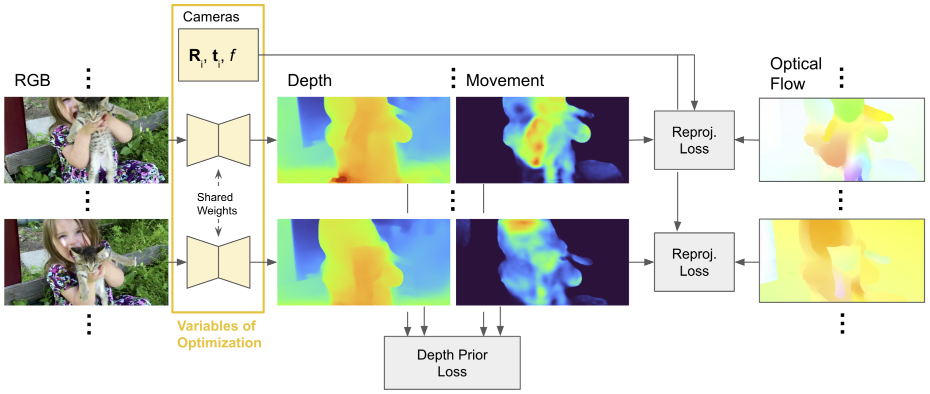


Fig. 2. System overview. Our method performs a global optimization over a video, optimizing for camera poses $\mathbf{R}_i$, $\mathbf{t}_i$, focal length f, and the weights of a monocular depth and movement prediction network. The principal loss (Reprojection Loss) compares the reprojection of the depth map with the observed optical flow, while an auxiliary loss (Depth Prior Loss) constrains the optimization to stay close to the initial depth estimates. Both losses are weighted by the estimated movement map. Only a subset of the weights of the network are optimized (see Sec. 3.2).

that use a self-supervised loss to learn depth, optical flow, and camera posing [7, 6], and methods that assume flow and camera poses and aim for high-quality depth map reconstruction [18, 37].

Because the test-time loss is still based on triangulation, moving objects remain an issue for these methods. CVD [18] assumes a mostly stationary scene, relying on the depth prior to avoid errors due to object motion. The method of [6] segments the video and proposes separate motion models for each segment, while GLNet [7] derives motion from optical flow and estimated depth. Dynamic Video Depth [37] explicitly models scene flow with an additional neural network, but relies on accurate initial camera poses to initialize the scene flow network from initial depth estimates. In contrast to these approaches, we model movement as uncertainty of the reprojection accuracy. This approach is robust to camera misalignment and does not require semantic segmentation.

Most related to our work is Robust Consistent Video Depth Estimation (RCVD) [16]. Like our approach, RCVD optimizes for camera poses and depth maps in a dynamic scene using a learned depth prior, driven by observed optical flow. Unlike our method, moving objects are assumed to be masked out using semantic segmentation. Further, RCVD does not finetune the depth network, opting instead to refine depth using a spline-based warp. The reason given is that finetuning the depth network requires an alternating optimization between stochastic gradient descent for the network and global optimization for the camera poses, which is unstable. We show that this alternating optimization can be avoided by collecting a full gradient at each iteration before updating the network weights (GD vs. SGD, Sec. 3.2), and demonstrate significantly improved quality as a result (Tab. 1).

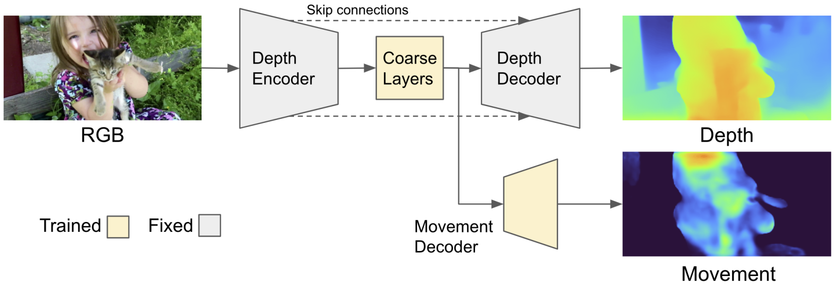


Fig. 3. Network design and trainable parameters. Our network consists of a monocular depth prediction network [25] with an additional movement prediction branch. Only the weights of the coarse layers of the depth network and the movement decoder weights are optimized. The depth prediction network is a U-net architecture with skip connections, but the movement branch is a CNN connected only to the coarse layers of the depth network.

3 Method

Our method takes a RGB video as input and performs a joint estimation of camera poses, focal length, dense depth maps, and dense movement maps (Fig. 2). There are two key design objectives for our method: (1) robustness to camera movement featuring small translational motion and (2) robustness to significant dynamic object motion. In Sec. 3.1, we explain how our method address these two challenges. We also introduce a two-stage optimization process that aims to jointly optimize networks weights and camera poses over all input frames, similar to global bundle adjustment.

3.1 Problem Formulation

Given images of a video sequence $I_1, I_2, ..., I_n$, we aim to recover the corresponding camera rotations $\mathbf{R}_1, \mathbf{R}_2, ..., \mathbf{R}_n$, translations $\mathbf{t}_1, \mathbf{t}_2, ..., \mathbf{t}_n$, dense depth maps $D_1, D_2, ..., D_n$, and movement maps $M_1, M_2, ..., M_n$. The movement maps are not binary masks, but instead scalar fields whose magnitude correlates with object motion magnitudes. We assume a pinhole camera model whose projection center is in the middle of the image, and also aim to recover the focal length f.

Our optimization runs over a collection of pairs of frames. This collection is different between the initialization stage and the joint optimization stage. To illustrate our formulation, we start with a single pair of frames, I_i and I_j. As in geometric bundle adjustment, the major signal for driving the optimization is to make estimated depth and camera poses agree with image space pixel correspondence. In our case, the correspondence is estimated using an off-the-shelf optical flow algorithm. We denote the optical flow between I_i and I_j as $\text{flow}_{i\to j}$. If we denote camera relative transformations from I_i to I_j as $\mathbf{R}_{i\to j} = \mathbf{R}_j \mathbf{R}_i^T$ and $\mathbf{t}_{i\to j} = \mathbf{t}_j - \mathbf{t}_i$, we can write the objective as:

$$\text{Loss}_{\text{flow}}^{i\to j}(\mathbf{x}) = L(\pi(D_i(\mathbf{x})\mathbf{K}\mathbf{R}_{i\to j}\mathbf{K}^{-1}\hat{\mathbf{x}} + \mathbf{K}\mathbf{t}_{i\to j}) - \mathbf{x}, \text{flow}_{i\to j}(\mathbf{x})), \quad (1)$$

where L is some loss function, $\mathbf{x}$ is the pixel coordinate, $\hat{\mathbf{x}}$ is $\mathbf{x}$ in homogeneous coordinates, $\pi(\cdot)$ is the projection operation $(x, y, z) \to (x/z, y/z)$, and $\mathbf{K}$ is the camera intrinsics matrix derived from focal length f. Note that this formulation is equivalent to bundle adjustment using a geometric reprojection error.

However, when camera motion is dominated by rotation, as in many casual videos, Equation 1 is less effective for optimizing D_i due to the limited parallax. This can be seen by rewriting it using disparity (1/depth) instead of depth:

$$\mathrm{Loss}_{\mathrm{flow}}^{i\to j}(\mathbf{x}) = L(\pi(\mathbf{K}\mathbf{R}_{i\to j}\mathbf{K}^{-1}\hat{\mathbf{x}} + \frac{1}{D_i(x)}\mathbf{K}\mathbf{t}_{i\to j}) - \mathbf{x}, \mathrm{flow}_{i\to j}(\mathbf{x})). \tag{2}$$

When $\mathbf{t}_{i\to j}$ is small(relative to the depth D_i), the loss term's gradient w.r.t. D_i will be small due to the $\mathbf{K}\mathbf{t}_{i\to j}$ term. In such cases, another constraint is needed that keeps the depth to be consistent according to optical flow:

$$\mathrm{Loss}_{\mathrm{depth}}^{i\to j}(\mathbf{x}) = L(d(D_i(\mathbf{x})\mathbf{R}_{i\to j}\mathbf{K}^{-1}\hat{\mathbf{x}} + \mathbf{t}_{i\to j}), D_j(\mathbf{x} + \mathrm{flow}_{i\to j}(\mathbf{x}))), \tag{3}$$

where $d(\cdot)$ denotes the depth of a 3D point in camera coordinates $(x, y, z) \to z$. We use both Equation 1 and Equation 3 for our optimization.

Note that the combination of these two objectives are found effective by CVD [18] and RCVD [16], each with their specific choice of the loss function L. For our implementation, we use L_1 for $\mathrm{Loss}_{\mathrm{flow}}$ and the ratio loss introduced in RCVD for $\mathrm{Loss}_{\mathrm{depth}}$. The ratio loss has the form:

$$L(a, b) = \left| \frac{\max(a, b)}{\min(a, b)} - 1 \right| \tag{4}$$

Handling movement. Until now, our formulation only addresses the problem of small camera translation, and not the problem of object motion. Most SfM methods that handle scene motion do so through a required input motion mask that specifies the moving regions, since pixels that belong to moving objects are outliers in both Equation 1 and Equation 3, contaminating the optimization process. In many cases, semantic segmentation is used as an approximation, where objects belonging to classes that tend to move are excluded from the optimization [16, 2]. As shown in Figure 8, semantic segments can be problematic. For example, pixels corresponding to stationary people are often excluded, but they are in fact helpful for depth and camera triangulation.

Instead of semantic segmentation, we aim to estimate object movement as part of the joint optimization. We adopt the machinery of Bayesian deep learning [13] and treat movement as the *heteroscedastic aleatoric uncertainty* of the reprojection, or in other words, a spatially-varying estimate of the noise of the depth and camera predictions. Instead of a Gaussian noise model as in [13], we use a Cauchy distribution as we empirically find it more robust. Treating the movement map M_i as the γ of a zero-mean Cauchy distribution, taking the negative-log-likelihood and simplifying, the error function is:

$$\mathrm{C}(\mathbf{x}, \mathrm{Loss}) = \log(M_i(\mathbf{x}) + \frac{\mathrm{Loss}(\mathbf{x})^2}{M_i(\mathbf{x})}). \tag{5}$$

The full reprojection loss is then:

$$L^{i\to j}_{\text{reproj}} = \frac{1}{N}\sum_{\mathbf{x}} \mathrm{C}(\mathbf{x}, \mathrm{Loss}^{i\to j}_{\text{flow}}) + \mathrm{C}(\mathbf{x}, \mathrm{Loss}^{i\to j}_{\text{depth}}). \tag{6}$$

As in [13], the uncertainty M_i is learned and allows the optimization to reduce M_i where the loss can be minimized effectively, and increase M_i where it cannot. Intuitively, M_i becomes a measure of how far an outlier the optical flow at $\mathbf{x}$ is from the expected ego-flow, which is in turn an estimate of how much object movement is present.

While the above method encourages an accurate movement map M_i, by design it does not penalize inaccurate depth estimates where M_i is large. Where M_i is large and the reprojection loss is unreliable, we fall back to the depth prior. Specifically, we propose to constrain the depth estimate D to the initial depth estimate D^{init} using a movement-weighted version of the scale invariant loss [9]:

$$L^{i}_{\text{prior}} = \frac{1}{N}\sum_{\mathbf{x}} M_i(\mathbf{x})(\log\frac{D_i(\mathbf{x})}{D^{\text{init}}_i(\mathbf{x})} + \alpha)^2, \quad \alpha = \frac{1}{N}\sum_{\mathbf{x}} \log\frac{D^{\text{init}}_i(\mathbf{x})}{D_i(\mathbf{x})} \tag{7}$$

as a depth prior loss. Finally, the total loss function we use for optimizing a pair of images is:

$$L^{i,j}_{\text{total}} = L^{i\to j}_{\text{reproj}} + \lambda L^{i}_{\text{prior}}, \tag{8}$$

where we use $\lambda = 1$ through out the experiments. To optimize over a collection of pairs, we average $L^{i,j}_{\text{total}}$ over all pairs for the total Loss L_{total}.

3.2 Two-Stage Optimization

Initialization. Since the depth maps from the depth prediction network are scale and shift invariant, we need to roughly align them before the joint optimization. Empirically, we find that calibrating only the scale is sufficient for the optimization. Specifically, for each initial depth map D^{init}_i, we assign a scale variable s_i and let $D_i = s_i D^{\text{init}}_i$ when optimizing L_{total}. During this phase, the weights of the depth network are fixed while the remaining variables are optimized.

The collection of image pairs is defined by a sliding window of 5 frames, from the beginning of the image sequence to the end. We use all pairs of frames within the sliding window, and optimize L_{total} for 600 iterations.

Full Optimization. After initialization, the camera poses are roughly aligned but not yet sufficiently accurate (Table 1). We then fix the scale factors s_i and optimize the weights of the depth network, letting $D_i = s_i\mathrm{DepthNet}(I_i)$ while optimizing L_{total}. In contrast to the initialization stage, in this stage we optimize L_{total} over all the frames in the video, with a collection of images pairs covering the entire set of images. Empirically, we find the pair sampling strategy by CVD is simple and effective; image pair i, j is sampled if $|i - j|$ is a power of 2. Note that we compute a full gradient for each step and perform gradient descent (GD) for both network weights and camera parameters, using an adaptive first-order optimizer (Adam [15]). The per-parameter weight tuning of Adam is sufficient to deal with the widely varying gradient magnitudes between the camera parameters and network weights.

Implementation Details. We use RAFT [29] to estimate optical flow and MiDaS [25] as the depth prediction network. Since we perform a full gradient descent over all sampled pairs of frames during the full optimization, we only optimize the coarse layers (4 refinement layers) of the MiDaS decoder to make computation costs manageable. The movement maps M_i are generated by a small CNN decoder that takes as input the MiDaS encoder output (Fig. 3). The decoder is composed of 8 convolution layers with two bi-linear upsampling layers. More details of the networks can be found in the supplementary material.

The camera poses are represented with camera-to-world translations and rotations. Rotations are represented in Lie Algebra $\mathfrak{so}(3)$, and translations as 3d vectors. Focal length f is initialized as 55mm through out all the experiments.

We use a learning rate of 1e-3 for the movement map decoder, 1e-4 for the coarse layers of MiDaS network. For the full optimization, we accumulate gradients over batches of 8 pairs of frames to perform full gradient descent. We take 1800 iterations of full gradient descent for all our experiments. Since Eq. 5 is ill-defined when $M_i = 0$, we add a fixed bias of 0.5 to M_i when computing the error function.

4 Results

We evaluate CasualSAM both quantitatively and qualitatively on the MPI Sintel dataset [5], dynamic sequences from the TUM RGB-D dataset [28], and the DAVIS video dataset [24]. We evaluate both camera pose and depth maps on Sintel, which contains fast object and camera motion with ground truth annotations. We evaluate camera pose accuracy on the TUM dynamic sequences, where motion is limited but ground truth camera poses are provided. Since no ground-truth depth or poses are available for DAVIS, we evaluate consistency between our predictions and optical flow, and show results for 3D reconstruction.

Baselines. We compare with two state-of-the-art learning-based methods: DROID-SLAM (DSLAM) [30] and Robust CVD (RCVD) [16]. DSLAM is a robust SLAM system that focuses on camera localization for almost static scenes and requires camera intrinsics as an input. RCVD is a camera localization and depth estimation system aimed at video clips. It optimizes camera focal length within its system and uses an off-the-shelf semantic segmentation as an approximation for movement masks. We also compare with COLMAP [26, 27] as a non-learning based baseline to demonstrate the limitations of conventional SfM.

4.1 Camera Pose and Depth Accuracy on Sintel

Camera Pose Evaluation. We compare camera pose quality against RCVD and DSLAM. Metrics used are Absolute Translation Error(ATE), Relative Translation Error(RTE) and Relative Rotation Error(RRE). Since camera tracks in the Sintel dataset have very different lengths (from less than 1 meter to larger than 100 meters), simply averaging over all sequences introduces bias towards long

Table 1. Pose and depth accuracy on Sintel. We compare camera pose accuracy on Sintel using normalized ATE and RTE (fraction of total path length) for translation, and RRE for rotation in degrees.

Method	Pose Error ↓			Depth Error, Rel. L1 ↓			Avg. Depth Accuracy ↑		
	ATE	RRE	RTE	Avg.	Dynamic	Static	$\delta < 1.5$	$\delta < 1.5^2$	$\delta < 1.5^3$
DSLAM† (GT focal)[30]	0.077	1.605	0.043	–	–	–	–	–	–
Ours (GT focal)	**0.036**	**0.190**	**0.008**	0.440	1.151	0.191	0.651	0.792	0.863
RCVD‡ (Opt. focal)[16]	0.164	1.151	0.057	0.847	1.505	0.427	0.543	0.718	0.806
Ours (Init. only)	0.122	0.449	0.025	0.697	1.468	0.305	0.526	0.719	0.825
Ours (No uncertainty)	0.134	0.573	0.026	0.779	2.295	0.284	0.527	0.717	0.829
Ours (Opt. focal, full)	**0.089**	**0.410**	**0.015**	**0.484**	**1.267**	**0.227**	**0.626**	**0.775**	**0.853**

† We use the original code to run all experiments. ‡ we used the results provided by the authors.

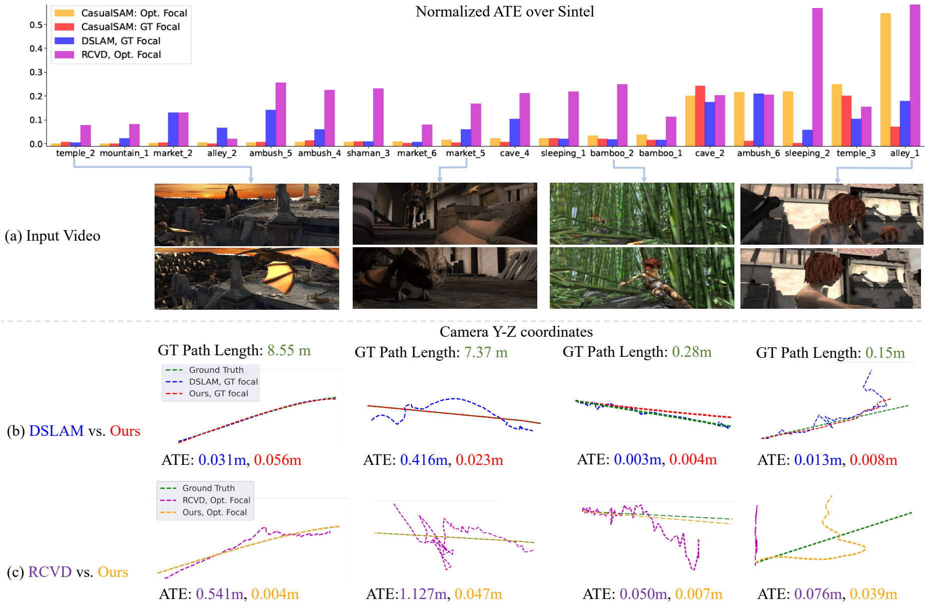


Fig. 4. Results of camera localization on Sintel. In the top row, we plot ATE results of CasualSAM, RCVD and D-SLAM over all the Sintel sequences, sorted by ATE of CasualSAM without ground truth focal length. We then select 4 sequences from low ATE to high ATE shown in (a). We plot the Y-Z coordinates of the camera locations of each method against ground truth (b-e).

trajectories. Therefore, before calculating the metrics, ground truth trajectories are normalized to unit length. For all methods, we align the predicted results to the normalized ground truth tracks using Umeyama [34] alignment with scale calibration. Five sequences are excluded from ATE and RTE calculation because the cameras are stationary. The overall results are reported in Table 1 and per-track statistics in Figure 4. Our method achieves 53% smaller ATE than DSLAM given the ground truth camera intrinsics, and 46% smaller ATE than RCVD when optimizing for focal length.

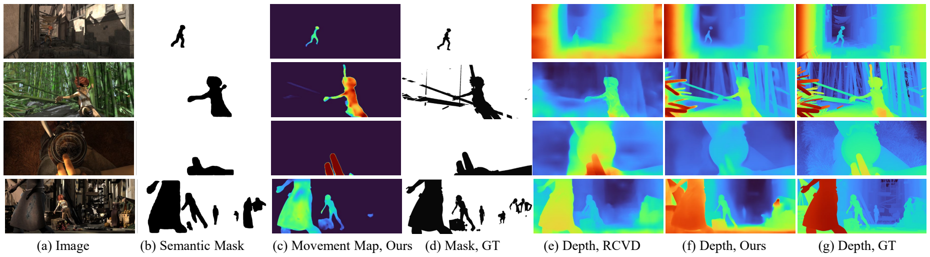


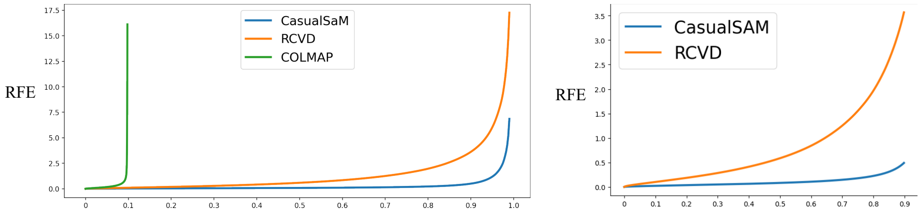
Fig. 5. Qualitative results of depth and movement map prediction on Sintel. Given an image (a), we show semantic masks (b) used by RCVD as movement segmentation proxies. Our movement map estimation is shown in (c) and ground truth masks in (d). We also show depth maps of RCVD, CasualSAM, and ground truth in (e)-(f). CasualSAM has more accurate estimate of both movement and depth maps.

Depth Map Evaluation. We evaluate the depth map quality against RCVD. Because DSLAM a slam system that focuses on camera localization, it is excluded from depth evaluation. We adopt the standard depth metrics: Absolute Relative Error and Delta accuracy measures; for Absolute Relative Error, we report results for static and dynamics regions separately, in addition to the average error. We follow the standard evaluation protocol by excluding points that are further than 80 meters. Median alignment is applied for all the metric calculations. We report quantitative results in Table 1 and qualitative results in Figure 5. Ours method produces more accurate depth maps than RCVD for dynamics and static part of the scenes, qualitatively and quantitatively.

Ablations We perform an ablation study on the sintel dataset, quantifying the contribution of uncertainty map estimation and known intrinsics. Without uncertainty map estimation, where the reprojection and prior term are replaced with plain L1 loss, both camera pose quality and depth map quality degrades, as all the moving objects are treated as if being static. Without known intrinsics, the camera poses degrades compared to results using known intrinsics. However, in the per-track statistics shown in Figure 4, this degradation is largely due to three tracks in the entire dataset. A more detailed ablation, comparing our uncertainty map estimation against previous works[38, 20] can be found in the supplementary material.

Table 2. ATE on TUM dynamic sequences. Absolute Translation Error (ATE) in meters of estimated camera poses for dynamic sequences in the TUM RGBD dataset [28].

Method	s_halfsphere	s_rpy	s_static	s_xyz	w_halfsphere	w_rpy	w_static	w_xyz
DROID-SLAM (GT focal) [30]	0.079	0.065	**0.005**	0.009	**0.023**	0.144	0.006	0.016
Ours (GT focal)	**0.045**	**0.019**	**0.005**	**0.008**	0.080	**0.032**	**0.005**	**0.012**
RCVD(Opt. K)	0.254	0.058	0.026	0.206	0.241	0.171	0.020	0.205
Ours(Opt. K)	**0.096**	**0.033**	**0.008**	**0.009**	**0.088**	**0.082**	**0.007**	**0.024**



(a) Residual Flow Error over all pixels

(b) Residual Flow Error, 90% quantile

Fig. 6. Residual Flow Error Over DAVIS. (a) We show the Residual Flow Error statistics over all the valid pixels in the DAVIS dataset. Even with GT segmentation, COLMAP[27] only generates valid depth values for 10% pixels over the entire dataset. (b) 90% of the errors of CasualSAM are below 0.5 pixels.

4.2 Camera Pose Accuracy on TUM benchmarks

We further evaluate camera pose quality of CasualSAM on the dynamic subset of the TUM benchmarks. This subset contains 8 tracks, capturing two people sitting and walking in front of an office desk. In addition to dynamic objects, this sequence is challenging due to versatile camera motions. Our method is better or on par with Droid-SLAM for 7 out 8 sequences with known intrinsics, and is better than RCVD in all sequences when intrinsics is not known.

4.3 Depth and Pose Quality on DAVIS

The DAVIS video dataset [24] is a set of 90 short videos captured with handheld cameras and containing moving objects. Most of the videos are less than 10 seconds long. Though intended as a segmentation benchmark, the dataset provides an excellent range of casual videos to test our method.

Residual Flow Error. Since no ground-truth camera poses were captured, we instead evaluate how well the ego-flow induced by the camera motion and the estimated depth map agree with optical flow as measured by RAFT [29]. Since DAVIS provides segmentation masks for foreground objects, we report the differences between ego-flow and optical flow outside the mask, which is suppose to be static through out the video. We refer to this metric as Residual Flow Error(RFE). To give an accurate picture of the distribution of results, we plot the cumulative distributions of errors across all pixels of the DAVIS videos in Figure 6. Note that the RFE of CasualSAM is 53% of RCVD and 90% of the error is less than 0.5 pixels. We show qualitative results of the RFE in Figure 7.

We also compare with COLMAP [27, 26] under RFE. COLMAP's SfM pipeline either failed or produced multiple tracks for 63 out of 90 videos, even when provided the DAVIS GT segmentation as a movement mask. For the remaining 27 sequences, we run the COLMAP's MVS pipeline for per-frame depth estimates, which provides valid depth values for only 10% of all pixels. The RFE for the majority of valid pixels is low, but still higher than the bottom 10% of pixels from our method (Figure 6).

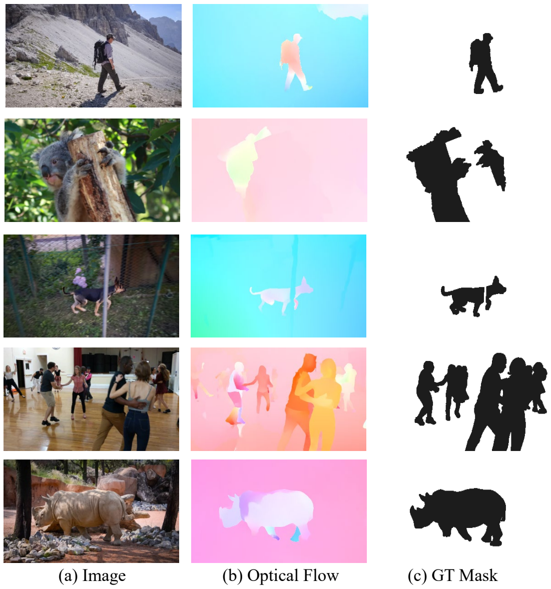


Fig. 7. Residual flow results. Given input image (a), we show its optical flow towards next frame (b) and DAVIS GT segmentation (c). Residual flow of RCVD and ours is shown in (d) and (e) respectively. GT segmentation is shown in grey and pixels outside the mask should show small errors.

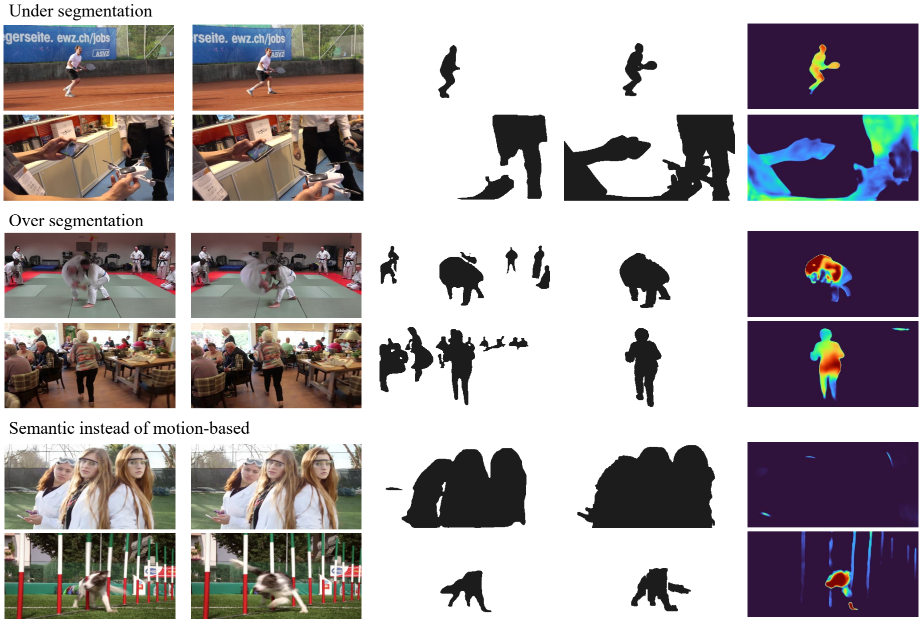
Movement Mask Comparison. The movement maps M_i are more sensitive and specific than using semantic segmentation as a proxy for motion segmentation, as shown in Fig. 8. Since the semantic segmentation is ignorant of motion cues, it may exhibit different types of failures compared to ours.

Depth and reprojection. We show qualitative results on depth and 3D reconstruction in Fig. 9. For 3D reconstruction, we use the predicted depths and camera poses as input to KinectFusion[12] to generate a mesh. To exclude moving objects from this mesh, we threshold our estimated movement map at 0.5, and use semantic segmentation for RCVD as described by the authors [16]. Our depth maps are more plausible than RCVD. Our fusion results are cleaner as well, suggesting our estimated camera poses and depth maps agree with each other better than RCVD.

5 Discussion and Limitations

Our method provides high-quality camera poses and dense depth maps for a broad range of casual videos. Compared with previous work, the method is simple and robust: it does not require semantic labeling of moving regions, handles videos with large and small camera motion.

There are several avenues to improve results. One is in the depth prior itself: if the prediction of the depth CNN is particularly poor, the optimization cannot recover (Fig. 10(b)). The MiDaS network is very powerful, but is vulnerable to



(a) Frame 1 (b) Frame 2 (c) Semantic Mask (d) DAVIS GT Mask (e) Our Movement Map

Fig. 8. Learned movement map vs. alternatives. From an RGB input (a)(b), the movement prediction network produces a map of moving regions (e), blue: not moving, red: moving). We show three cases where semantic segmentation (c) is not accurate: under segmentation (top), over segmentation (middle), and segmentations (or lack thereof) of objects that are not actually moving (or moving) in the scene (bottom).

errors for unusual camera angles such as the roll present in the TUM dataset. A more robust depth prior would similarly make our method more robust.

Currently, the camera model is a pinhole camera with a constant focal length for the entire video. However, wide-angle, zoom, and even fisheye lenses are common in casual videos (including the DAVIS dataset; see supplemental material). While unconstrained optimization for per-frame, multi-parameter intrinsics may be unstable, recent work proposed CNNs for intrinsics prediction [19, 4, 17] that could optimized similarly to our depth prior.

When the camera moves very rapidly, or when a moving object covers most of the frame, our method can lose tracking (Fig. 10(a)). Improved uncertainty map estimation, possibly using semantic features, may allow the optimization maintain a consistent track across such interruptions.

Finally, our method relies on the depth prior for depth estimation in moving regions, which may be inaccurate (Fig. 10(c)). Dynamic Video Depth [37] has shown improved depth maps in moving regions by using an explicit estimate of sceneflow and using it to apply multi-view constraints to moving objects, and this approach could likely be integrated with ours.

6 Acknowledgement

The authors would like to thank Jian-bin Huang for providing the official results of RCVD[16].

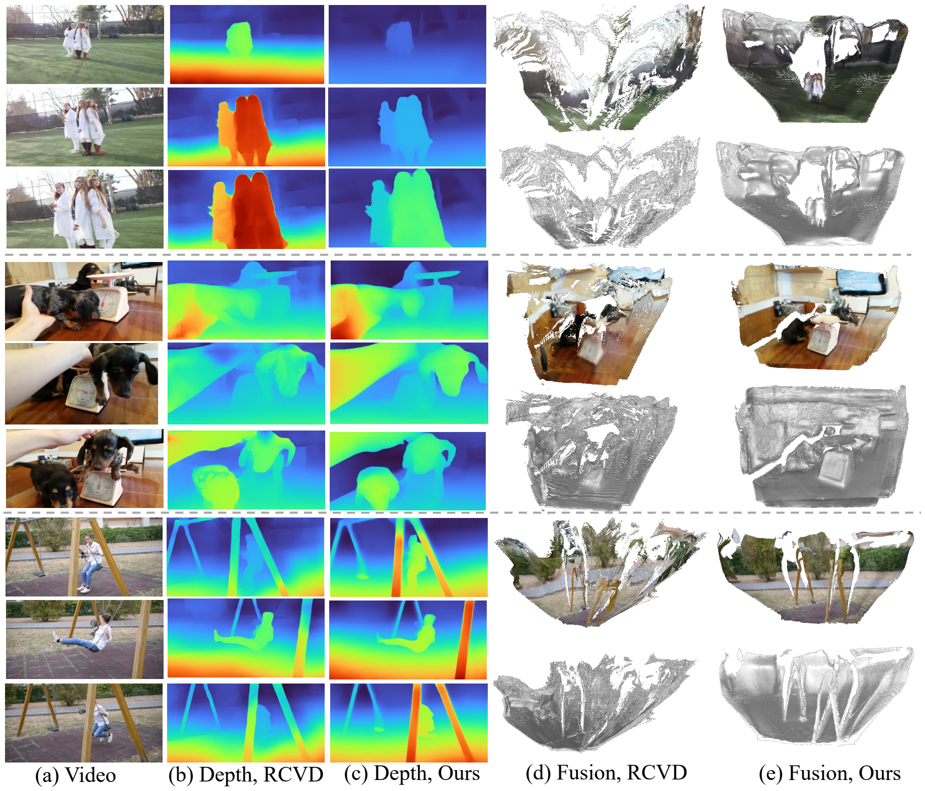


Fig. 9. Qualitative Results on Davis. (a) Input video. (b) Depth maps from Robust CVD [16]. (c) Depth maps from CasualSAM (ours). (d) KinectFusion results [12] using RCVD estimated depth maps, camera poses and motion masks (semantic segmentation). (e) KinectFusion results using our estimated depth maps, camera poses and movement maps.

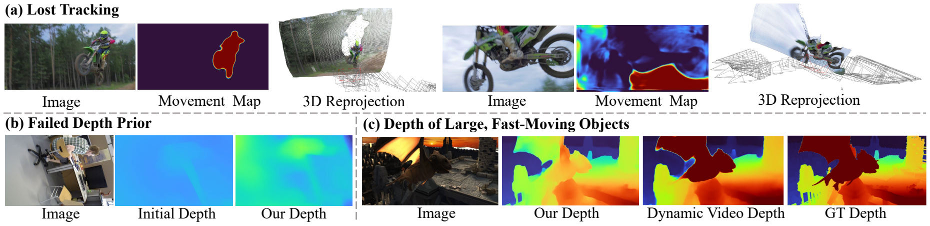


Fig. 10. Limitations of our method. (a) movement estimation may fail and camera tracking may be lost if a moving object dominates the frame (note movement map switches between bike and background). (b) Initial depth may fail due to an unusual camera angle, such as heavy roll. (c) Depth of moving objects may be inaccurate, as our method relies on the depth prior in those cases.

References

1. Agarwal, S., Snavely, N., Seitz, S.M., Szeliski, R.: Bundle adjustment in the large. In: European conference on computer vision. pp. 29–42. Springer (2010)
2. Bescos, B., Fácil, J.M., Civera, J., Neira, J.: Dynaslam: Tracking, mapping, and inpainting in dynamic scenes. IEEE Robotics and Automation Letters **3**(4), 4076–4083 (2018)
3. Bloesch, M., Czarnowski, J., Clark, R., Leutenegger, S., Davison, A.: CodeSLAM—Learning a compact, optimisable representation for dense visual SLAM. In: CVPR (2018)
4. Bogdan, O., Eckstein, V., Rameau, F., Bazin, J.C.: Deepcalib: a deep learning approach for automatic intrinsic calibration of wide field-of-view cameras. In: Proceedings of the 15th ACM SIGGRAPH European Conference on Visual Media Production (2018)
5. Butler, D.J., Wulff, J., Stanley, G.B., Black, M.J.: A naturalistic open source movie for optical flow evaluation. In: A. Fitzgibbon et al. (Eds.) (ed.) European Conf. on Computer Vision (ECCV). pp. 611–625. Part IV, LNCS 7577, Springer-Verlag (Oct 2012)
6. Casser, V.M., Pirk, S., Mahjourian, R., Angelova, A.: Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos. In: AAAI (2019)
7. Chen, Y., Schmid, C., Sminchisescu, C.: Self-supervised learning with geometric constraints in monocular video: Connecting flow, depth, and camera. In: ICCV. pp. 7063–7072 (2019)
8. Davison, A.J., Reid, I.D., Molton, N.D., Stasse, O.: MonoSLAM: Real-Time Single Camera SLAM. TPAMI **29**(6) (2007)
9. Eigen, D., Puhrsch, C., Fergus, R.: Depth map prediction from a single image using a multi-scale deep network. arXiv preprint arXiv:1406.2283 (2014)
10. Engel, J., Koltun, V., Cremers, D.: Direct sparse odometry. TPAMI (2018)
11. Engel, J., Schöps, T., Cremers, D.: LSD-SLAM: Large-scale direct monocular SLAM. In: ECCV (2014)
12. Izadi, S., Kim, D., Hilliges, O., Molyneaux, D., Newcombe, R., Kohli, P., Shotton, J., Hodges, S., Freeman, D., Davison, A., Fitzgibbon, A.: Kinectfusion: Real-time 3d reconstruction and interaction using a moving depth camera. In: UIST '11 Proceedings of the 24th annual ACM symposium on User interface software and technology. pp. 559–568. ACM (October 2011)
13. Kendall, A., Gal, Y.: What uncertainties do we need in bayesian deep learning for computer vision? In: Proceedings of the 31st International Conference on Neural Information Processing Systems. p. 5580–5590. NIPS'17, Curran Associates Inc., Red Hook, NY, USA (2017)
14. Kerl, C., Sturm, J., Cremers, D.: Dense visual slam for rgb-d cameras. In: IROS (2013)
15. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. International Conference on Learning Representations (12 2014)
16. Kopf, J., Rong, X., Huang, J.B.: Robust consistent video depth estimation. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (2021)
17. Li, X., Zhang, B., Sander, P.V., Liao, J.: Blind geometric distortion correction on images through deep learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4855–4864 (2019)
18. Luo, X., Huang, J., Szeliski, R., Matzen, K., Kopf, J.: Consistent video depth estimation. In: ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH) (2020)

19. López, M., Marí, R., Gargallo, P., Kuang, Y., Gonzalez-Jimenez, J., Haro, G.: Deep single image camera calibration with radial distortion. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 11809–11817 (2019). https://doi.org/10.1109/CVPR.2019.01209
20. Mahjourian, R., Wicke, M., Angelova, A.: Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5667–5675 (2018)
21. Mur-Artal, R., Tardós, J.D.: ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras. IEEE Transactions on Robotics **33**(5) (2017)
22. Newcombe, R.A., Fox, D., Seitz, S.M.: DynamicFusion: Reconstruction and tracking of non-rigid scenes in real-time. In: CVPR (2015)
23. Newcombe, R.A., Lovegrove, S.J., Davison, A.J.: DTAM: Dense tracking and mapping in real-time. In: ICCV (2011)
24. Perazzi, F., Pont-Tuset, J., McWilliams, B., Van Gool, L., Gross, M., Sorkine-Hornung, A.: A benchmark dataset and evaluation methodology for video object segmentation. In: CVPR (2016)
25. Ranftl, R., Lasinger, K., Hafner, D., Schindler, K., Koltun, V.: Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) (2020)
26. Schönberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: CVPR (2016)
27. Schönberger, J.L., Zheng, E., Pollefeys, M., Frahm, J.M.: Pixelwise view selection for unstructured multi-view stereo. In: ECCV (2016)
28. Sturm, J., Engelhard, N., Endres, F., Burgard, W., Cremers, D.: A benchmark for the evaluation of rgb-d slam systems. In: Proc. of the International Conference on Intelligent Robot Systems (IROS) (Oct 2012)
29. Teed, Z., Deng, J.: Raft: Recurrent all-pairs field transforms for optical flow. In: European Conference on Computer Vision. pp. 402–419. Springer (2020)
30. Teed, Z., Deng, J.: DROID-SLAM: Deep Visual SLAM for Monocular, Stereo, and RGB-D Cameras. In: NeurIPS (2021)
31. Torresani, L., Hertzmann, A., Bregler, C.: Nonrigid structure-from-motion: Estimating shape and motion with hierarchical priors. TPAMI **30**(5), 878–892 (2008)
32. Triggs, B., McLauchlan, P.F., Hartley, R.I., Fitzgibbon, A.W.: Bundle adjustment - a modern synthesis. In: Proc. Int. Workshop on Vision Algorithms: Theory and Practice (1999)
33. Triggs, B., McLauchlan, P.F., Hartley, R.I., Fitzgibbon, A.W.: Bundle adjustment—a modern synthesis. In: International workshop on vision algorithms. pp. 298–372. Springer (1999)
34. Umeyama, S.: Least-squares estimation of transformation parameters between two point patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence **13**(4), 376–380 (1991). https://doi.org/10.1109/34.88573
35. Yang, N., von Stumberg, L., Wang, R., Cremers, D.: D3VO: Deep depth, deep pose and deep uncertainty for monocular visual odometry. In: CVPR (2020)
36. Yu, F., Gallup, D.: 3d reconstruction from accidental motion. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3986–3993 (2014)
37. Zhang, Z., Cole, F., Tucker, R., Freeman, W.T., Dekel, T.: Consistent depth of moving objects in video. ACM Transactions on Graphics (TOG) **40**(4), 1–12 (2021)
38. Zhou, T., Brown, M., Snavely, N., Lowe, D.G.: Unsupervised learning of depth and ego-motion from video. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1851–1858 (2017)