Supplementary Material
MegaSaM: Accurate, Fast and Robust Structure and Motion from Casual Dynamic Videos

Zhengqi Li[1], Richard Tucker[1], Forrester Cole[1], Qianqian Wang[1,2], Linyi Jin[1,3]
Vickie Ye[2], Angjoo Kanazawa[2], Aleksander Holynski[1,2], Noah Snavely[1]

[1]Google DeepMind [2]UC Berkeley [3]University of Michigan

1. Implementation Details

1.1. System Overview

Figure 1 shows an overview of our MegaSaM system. We separate the problem of camera and scene structure estimation into two stages, in the spirit of a conventional SfM pipeline [5, 6]. In particular, we first estimate camera poses $\hat{\mathbf{G}}$, focal length $\hat{f}$ and low-resolution disparity $\hat{\mathbf{d}}$ from the input monocular video through differentiable Bundle Adjustment (BA), where we initialize $\hat{\mathbf{d}}$ with monocular depth maps predicted from off-the-shelf models [4, 9]. In the second consistent video depth estimation phase, we fix estimated camera parameters and perform first-order optimization over video depth and uncertainty maps by enforcing flow and depth losses induced by pairwise 2D optical flows.

1.2. Framework and Architecture

We follow DROID-SLAM [7] for feature extraction, correlation feature construction, and perform iterative BA updates through flow, confidence, motion probability predictions. Each input to the model is a pair of video frames (I_i, I_j).

Feature extraction. We use context and feature encoders to encode each input video frame into two different low-resolution feature maps at $\frac{1}{8}$ resolution of the input image, as shown in Figure 3.

Correlation feature construction. The correlation layer constructs a 4D correlation volume from the features encoded from an image pair, and each entry of the volume contains inner product of one pairs of feature vectors from the image pair.

Iterative updates. During each iterative BA step k, we update camera parameters and low-resolution disparity through flow, confidence and motion probability prediction. In particular, we first pretrain F on synthetic video data (ego-motion pretraining in the main paper) to learn to predict flows and corresponding flow confidence, as shown by the gray blocks in Figure 2. In the second dynamic finetuning phase, we freeze the parameters of F and finetune the motion module F_m to predict extra object motion probability maps conditioned on the features from the ConvRGU, as shown in the blue blocks in Figure 2. Within the motion module, we first perform 2D spatial average pooling to provide the model with global spatial information; we then perform average pooling along the time axis to fuse information from I_i and all its neighboring keyframes I_j (where $j \in \mathcal{N}(i)$).

1.3. Consistent Video Depth Optimization

Recall, from Section 3.3 of our main paper, that we follow CasualSAM [11] to estimate consistent video depth by performing an additional first-order optimization on video disparity $\hat{D}_i$ along with per-frame aleatoric uncertainty maps $\hat{M}_i$. Instead of jointly optimizing camera parameters and scene structure as in CasualSAM, however, we fix camera parameters as done in conventional SfM pipelines like COLMAP [5, 6].

Our objective consists of three main cost functions:

$$\mathcal{C}_{\text{cvd}} = w_{\text{flow}}\mathcal{C}_{\text{flow}} + w_{\text{temp}}\mathcal{C}_{\text{temp}} + w_{\text{prior}}\mathcal{C}_{\text{prior}} \quad (1)$$

We treat object motion in the video as the heteroscedastic aleatoric uncertainty of the flow reprojection and depth consistency error [2], and assume the underlying noise is Laplacian [10]. Specifically, for each selected pair (I_i, I_j), flow reprojection loss $\mathcal{C}_{\text{flow}}$ compares l_1 loss weighted by the uncertainty $\hat{M}_i$ between flows $\text{flow}_{i\to j}$ from an off-the-shelf flow estimator [7] and the correspondences $\mathbf{u}_{ij}$ induced by our estimated camera motion and disparity through a multi-view constraint:

$$\mathcal{C}_{\text{flow}}^{i\to j} = \hat{M}_i ||\mathbf{u}_{ij} - \mathbf{p}_i, \text{flow}_{i\to j}(\mathbf{p}_i)||_1 + \log\left(\frac{1}{\hat{M}_i}\right), \quad (2)$$

$$\mathbf{u}_{ij} = \pi\left(\hat{\mathbf{G}}_{ij} \circ \pi^{-1}(\mathbf{p}_i, \hat{D}_i, K^{-1}), K\right) \quad (3)$$

$\mathcal{C}_{\text{temp}}$ is an uncertainty weighted temporal depth loss that encourages pixel disparity to be temporally consistent ac-

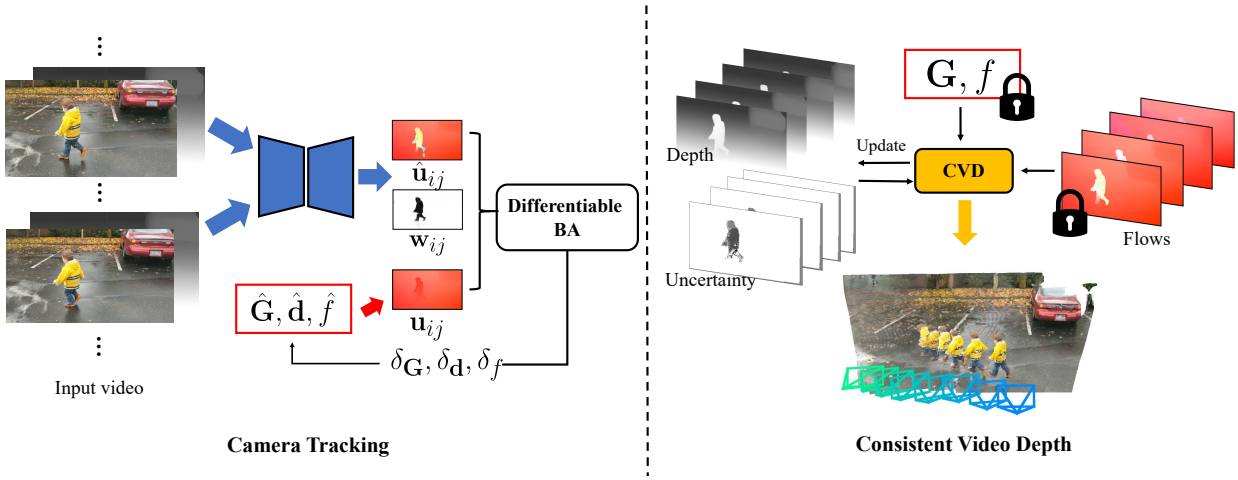

Figure 1. **System overview. Left**: we estimate camera poses, focal length and low-resolution disparity maps from the input monocular video through differentiable Bundle Adjustment (BA): the network iteratively updates these state variables by learning to predict low-resolution flow $\hat{\mathbf{u}}_{ij}$, confidence, and movement probability maps $\mathbf{w}_{ij}$ and minimize weighted reprojection error between predicted flow $\hat{\mathbf{u}}_{ij}$ and flow induced by ego-motion $\mathbf{u}_{ij}$. We also initialize estimated disparity with mono-depth predicted from off-the-shelf models [4, 9]. **Right**: we fix estimated camera parameters and perform first-order global optimization over video depth and corresponding uncertainty parameters by minimizing flow and depth losses through pairwise 2D optical flows.

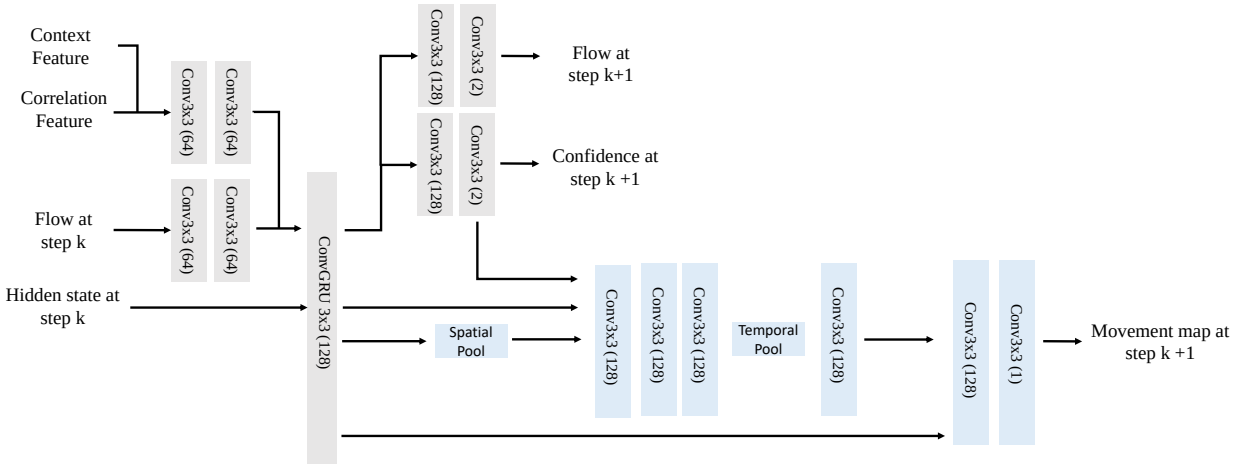

Figure 2. Architecture of flow, confidence and movement map predictor. The gray blocks belong to the network F for flow and confidence prediction, and the blue blocks belong to the network F_m for object movement map prediction. In the first stage, we perform ego-motion pretraining for F. In the second stage, we perform dynamic fine-tuning for F_m while fixing the parameters of F.

cording to estimated 2D optical flow:

$$\mathcal{C}_{\text{temp}}^{i\to j} = \hat{M}_i \delta\left(\mathbf{P}_z^{i\to j}, \hat{D}_j(\mathbf{p} + \text{flow}_{i\to j}(\mathbf{p}))\right) + \log\left(\frac{1}{\hat{M}_i}\right)$$

$$\delta(a,b) = ||\max(\frac{a}{b}, \frac{b}{a})||_1$$

$$\mathbf{P}_z^{i\to j} = \left(D_i(\mathbf{p})\mathbf{R}_{i\to j}\mathbf{K}^{-1}\mathbf{p} + \mathbf{t}_{i\to j}\right)_{[z]} \quad (4)$$

$\mathbf{R}_{i\to j}$ and $\mathbf{t}_{i\to j}$ are relative camera rotation and translation between I_i and I_j; $_{[z]}$ is an operator that retrieve the third component of the 3D point vector (i.e. z value).

$\mathcal{C}_{\text{prior}}$ is a depth prior loss that stops the final estimated video disparity from drifting too much from the initial estimate from the mono-depth network, and it consists of three losses:

$$\mathcal{C}_{\text{prior}} = \mathcal{C}_{\text{si}} + w_{\text{grad}}\mathcal{C}_{\text{grad}} + w_{\text{normal}}\mathcal{C}_{\text{normal}} \quad (5)$$

The scale-invariant depth loss $\mathcal{C}_{\text{si}}$ computes the mean square error of the difference among all pairs between optimized log-disparity $\log \hat{D}_i$ and initial log-disparity from the

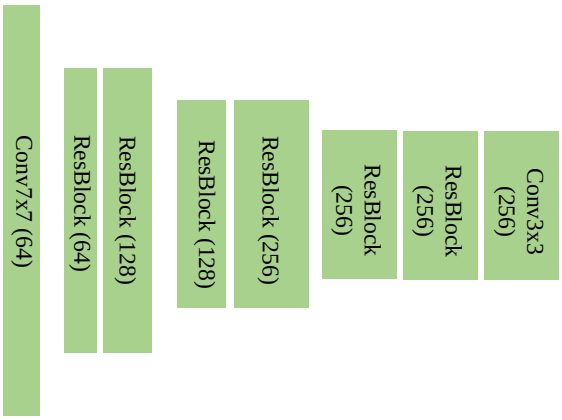

Figure 3. Architecture of the feature and context encoders. Both encoders extract low-resolution features from input video frames at $\frac{1}{8}$ of the original resolution.

metric-aligned mono-depth prediction $\log D_i^{\text{align}}$.

$$\mathcal{C}_{\text{si}} = \frac{1}{n}\sum_{(\mathbf{p})}(R(\mathbf{p}))^2 - \frac{1}{n^2}\left(\sum_{(\mathbf{p})} R(\mathbf{p})\right)^2$$
$$R_i = \log(\hat{D}_i) - \log(D_i^{\text{align}}). \tag{6}$$

$\mathcal{C}_{\text{grad}}$ is a multi-scale scale-invariant gradient matching term [3], which computes l_1 difference between estimated log disparity gradients and initial log-disparity gradients

$$\mathcal{C}_{\text{grad}} = \frac{1}{n}\sum_{s} w_{\nabla}^{s}(\mathbf{p}) \sum_{\mathbf{p}} \left(|\nabla_x R^s(\mathbf{p})| + |\nabla_y R^s(\mathbf{p})|\right)$$
$$w_{\nabla}^{s}(\mathbf{p}) = 1 - \exp\left(-\beta_{\nabla}(\nabla_x R^s(\mathbf{p}) + \nabla_y R^s(\mathbf{p}))\right) \tag{7}$$

where $R^s(\mathbf{p})$ is log-depth difference map at pixel position $\mathbf{p}$ and scale s. In other words, we only apply multi-scale gradient matching loss to pixels where the current estimated disparity deviates significantly from the original mono-depth.

$\mathcal{C}_{\text{normal}}$ is a surface normal loss that encourages that normal $\hat{\mathbf{N}}(\mathbf{p})$ derived from estimated disparity to be close to the surface normal $\mathbf{N}^{\text{align}}$ derived from the initial metric-aligned monocular disparity:

$$\mathcal{C}_{\text{normal}} = \sum_{\mathbf{p}} 1 - \hat{\mathbf{N}}(\mathbf{p}) \cdot \mathbf{N}^{\text{align}}(\mathbf{p}) \tag{8}$$

We set $w_{\text{grad}} = 1, w_{\text{normal}} = 4, \beta_{\nabla} = 5$ throughout our experiments. We simply choose image pairs (I_i, I_j) from a set of fixed intervals following prior work [11]: $j \in (i+1, i+2, i+4, i+8, i+15)$. During optimization, we initialize the disparity variables from the metric-aligned monocular depth by combining estimates from off-the-shelf modules as described in the main paper [4, 9], and we initialize the uncertainty map with object motion probability maps predicted from our camera tracking module. The optimization first conducts a "warm-up" phase for 100 steps by fixing the video disparity variables and optimizing the per-frame uncertainty map, per-frame scale, shift variables using the aforementioned losses. The disparity maps and uncertainty maps are then optimized together under the aforementioned losses for another 400 steps.

1.4. Additional Details

Training Losses. We supervise our network using a combination of pose loss and flow loss. The flow loss is applied to pairs of adjacent frames. We compute the optical flow induced by the predicted depth and poses and the flow induced by the ground truth depth and poses. The loss is taken to be the average l2 distance between the two flow fields.

Given a set of ground truth poses $\{\mathbf{T}_i\}_{i=1}^N$ and predicted poses $\{\mathbf{G}_i\}_{i=1}^N$, the pose loss is taken to be the distance between the ground truth and predicted poses, $\mathcal{L}_{pose} = \sum_i ||\text{Log}_{SE(3)}(\mathbf{T}_i^{-1} \cdot \mathbf{G}_i)||_2$. We apply the losses to the output of every BA iteration with exponentially increasing weight using $\gamma = 0.9^k$, where k indicates the k^{th} BA iterations.

Training and Inference Details In our two-stage training scheme, we first pretrain our model on synthetic data of static scenes, which include 163 scenes from TartanAir [8] and 5K videos from static Kubric [1]. In the second stage, we finetune motion module F_m on 11K dynamic videos from Kubric [1]. Each training example consists of a 7-frame video sequence. We first precompute a distance matrix between each pair of video frame based on the average ego-motion induced flow magnitude. We then dynamically generate a training sequence according to the constructed distance matrix, we randomly sample each frame such that average flow between them is between 0.5px and 64px.

Within the camera tracking module, we normalize video disparity $\hat{\mathbf{d}}$ such that its 98 percentile is 2; we also normalize focal length by dividing it by the input image resolution within every the bundle adjustment stage.

2. Limitations

Despite excellent performance on a variety of in-the-wild videos, we observe that our approach can fail in extremely challenging scenarios, similar to findings from prior work [11]. For instance, camera tracking fails if moving objects dominate the entire image or if there is nothing for the system to track reliably, as shown in the first row of Fig. 4. Furthermore, our approach also struggles on dynamic videos where camera motion and object motion are colinear, as shown in the second row of Fig. 4.

References

[1] Klaus Greff, Francois Belletti, Lucas Beyer, Carl Doersch, Yilun Du, Daniel Duckworth, David J Fleet, Dan Gnanapragasam, Florian Golemo, Charles Herrmann, Thomas Kipf, Ab-

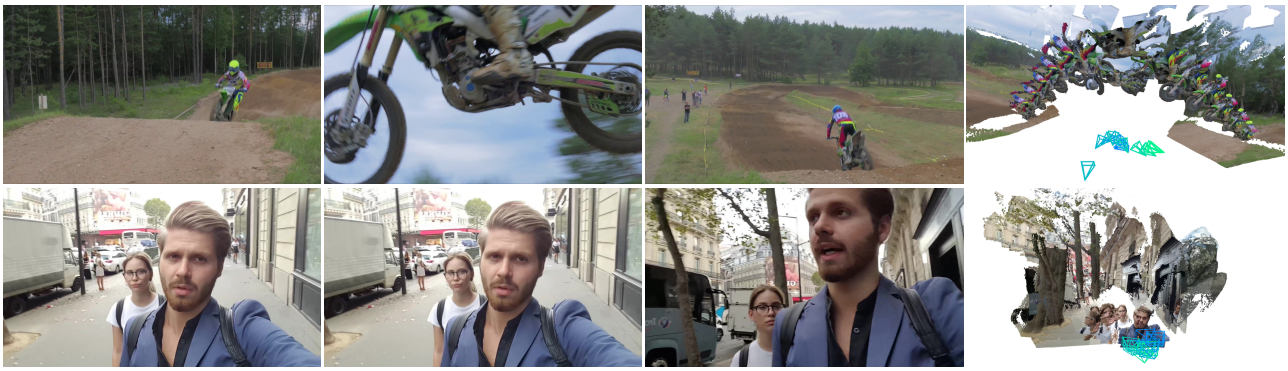

Figure 4. **Limitations.** We visualize three reference video frames on the left and their corresponding estimated camera paths and reconstruction on the right. Our method can lose tracks in cases where a moving object dominates the entire videos (top row). Our approach can also struggle in cases where object motion and camera motion are colinear, such as the selfie video in the bottom row.

hijit Kundu, Dmitry Lagun, Issam Laradji, Hsueh-Ti (Derek) Liu, Henning Meyer, Yishu Miao, Derek Nowrouzezahrai, Cengiz Oztireli, Etienne Pot, Noha Radwan, Daniel Rebain, Sara Sabour, Mehdi S. M. Sajjadi, Matan Sela, Vincent Sitzmann, Austin Stone, Deqing Sun, Suhani Vora, Ziyu Wang, Tianhao Wu, Kwang Moo Yi, Fangcheng Zhong, and Andrea Tagliasacchi. Kubric: a scalable dataset generator. 2022. 3

[2] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in neural information processing systems*, 30, 2017. 1

[3] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2041–2050, 2018. 3

[4] Luigi Piccinelli, Yung-Hsu Yang, Christos Sakaridis, Mattia Segu, Siyuan Li, Luc Van Gool, and Fisher Yu. UniDepth: Universal monocular metric depth estimation. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2024. 1, 2, 3

[5] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016. 1

[6] Johannes L Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14*, pages 501–518. Springer, 2016. 1

[7] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 402–419. Springer, 2020. 1

[8] Wenshan Wang, Delong Zhu, Xiangwei Wang, Yaoyu Hu, Yuheng Qiu, Chen Wang, Yafei Hu, Ashish Kapoor, and Sebastian Scherer. Tartanair: A dataset to push the limits of visual slam. 2020. 3

[9] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *CVPR*, 2024. 1, 2, 3

[10] Nan Yang, Lukas von Stumberg, Rui Wang, and Daniel Cremers. D3vo: Deep depth, deep pose and deep uncertainty for monocular visual odometry. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, pages 1281–1292, 2020. 1

[11] Zhoutong Zhang, Forrester Cole, Zhengqi Li, Michael Rubinstein, Noah Snavely, and William T Freeman. Structure and motion from casual videos. In *European Conference on Computer Vision*, pages 20–37. Springer, 2022. 1, 3