Supplementary Material

Anonymous ICCV submission

Paper ID 6575

1. Dataset

1.1. Statistics

As is mentioned in the paper, we use RealEstate10K (RE10K) [13], MannequinChallenge (MC) [3] and WSVD [11] for training and StereoBlur (SB) [12] for evaluation. Since the numbers of videos in these datasets are extremely unbalanced, for each epoch, we randomly choose a subset from each dataset. The statistics of these dataset is shown in Table 1.

Dataset	Depth supervision	Video count	Percentage for one epoch
SB	semi-dense	44	-
WSVD	semi-dense	1008	100%
RE10K	sparse	59380	2.5%
MC	sparse	1910	100%

Table 1: Statistics of datasets we use in experiments. *Percentage for one epoch* means the percent of data we use for one epoch during training. We use SB for evaluation only. Video counts may be different from the numbers reported officially because of the customized processing pipeline and some videos becoming unavailable

1.2. Data processing pipeline

Static scene RE10K and MC contain monocular videos exploring static scenes in the wild. They provide video links and frame indices for training as well as the corresponding camera poses, but no scene geometry is given. Thus we reconstruct a sparse point cloud model for each video clip using COLMAP [6, 7]. We initialize the camera extrinsic and intrinsic matrices using the provided poses and other parameters are set as default.

After reconstructing the sparse model, we filter out videos with frame numbers less than 12 and 3D points numbers less than 1000. We also compute the total travel distance of the camera and remove those less than 1.2. Finally we fit a global plane and compute the standard deviation (STD) of the distance of all points to the plane. We remove videos that has STD less than 0.6, since we find some videos try to construct all geometries in a common plane.

Dynamic scene WSVD contains YouTube stereo video links and frame indices for training. A closer look into the dataset we find the official splitting contains a certain number of clips with low quality. For example, some clips have negligible baselines so the left and right view are almost identical, some clips are too blur to compute accurate correspondence (due to motion blur or video compression) . Therefore, we use a customized data processing pipeline to filter out bad data and get temporal consistent semi-dense disparity map from raw data.

We start from detecting each individual scene by warping error[1] of two consecutive frames bigger than 0.25, or the average color is smaller than 0.15 (black frame). We also force the length of one clip to be between 0.5 and 10 seconds. For each frame, we compute its disparity in left (right) view by extracting the horizontal direction component of the optical flow to the right (left). We mask out those pixels with vertical flow bigger than 1.5 pixels or bidirectional flow consistency bigger than 1.5 pixels and yield a semi-dense disparity map. We decide that one video clip is a good clip only if: 1. The percentage of the pixels whose vertical flow bigger than 1.5 pixels is smaller than 30%; 2. The percentage of the pixels who do not pass the bidirectional flow consistency check is smaller than 40%; 3. The disparity of pixels ranked 90% and 10% has difference larger than 10 pixels. After this, we manually filter out more bad clips with duplicated content, wrong disparity or with severely flickering artifacts. Finally, we apply the occlusion aware temporal filtering to obtain temporal consistent disparity maps.

2. Implementation details

We now elaborate the implementation details and experiment settings.

2.1. Scale invariant depth loss

We describe the details of scale invariant depth loss. We first compute a scale factor by minimizing least square error

[1] We estimate all of the optical flows use RAFT algorithm [9]

in log space as in [10]:

$$scale = \exp\left(\frac{1}{|\mathbb{D}|}\sum_{i\in\mathbb{D}} \ln \frac{d_i}{\hat{d}_i}\right), \tag{1}$$

where the d_i and $\hat{d}_i$ is the ground truth and predicted disparity and $\mathbb{D}$ is the set of all the indices of ground truth. For semi-dense depth supervision, i is the pixel index and $\mathbb{D}$ is the set of all the pixels that has valid disparity value. For sparse depth supervision, i is the sparse point index and $\hat{d}_i$ is the bilinear sampling of $\hat{\mathbf{D}}$ at the 2D projection of the corresponding sparse point.

Scale-invariance works in MC and RE10K, where the camera is calibrated so the inverse depth differs the disparity with only a scaling factor. However in WSVD, lots of stereo videos have one view shifted, whereas the shift can not be easily obtained [11]. Scale-and-shift-invariant loss [5] can be used but we find fitting an extra shift parameter makes training unstable and converge slow. So instead we use a simpler but more effective solution by computing a pseudo shift:

$$shift = \begin{cases} \max_{i\in\mathbb{D}}(d_i) + \epsilon & \text{left view} \\ \min_{i\in\mathbb{D}}(d_i) - \epsilon & \text{right view,} \end{cases} \tag{2}$$

where $\epsilon = 4$ is a constant for manually shifting back the corresponding view. Although this is not physically correct, we find this scheme greatly speed up the training process while the small error caused by manual shifting is negligible.

The final depth loss $\mathcal{L}_{depth}$ is defined as:

$$\mathcal{L}_{depth} = \frac{1}{|\mathbb{D}|}\sum_{i\in\mathbb{D}}\left(\ln \frac{d_i + shift}{\hat{d}_i \times scale}\right)^2. \tag{3}$$

2.2. Training

We train the entire pipeline end-to-end jointly using all the three aforementioned datasets. For each iteration we pick one video and randomly choose 5 consecutive frames with random skip between 1 and 5 frames. For static scene dataset we randomly select one frame among the chosen 5 frames as the ground truth novel view. For stereo video dataset we randomly pick one view as input and the other as ground truth novel view.

We use distributed training in 10 RTX2080Tis with total batch size 10. We use Adam Optimizer [1] for training and set the initial learning rate as 2e-4. We train the pipeline for 120 epochs and decay the learning rate by a factor of 0.5 for every 30 epochs. Other hyper-parameters of the Adam Optimizer are set as default.

3. Novel view synthesis metrics analysis

As is discussed in the paper, our method shows big improvements in perceptual similarity and flow magnitude, but small gaps in terms of *SSIM* and *PSNR*. We claim that this is because the *SSIM* and *PSNR* prefer blurry results to misaligned results. We show one extreme case in Figure 1, where our method demonstrates better *LPIPS* and *FMean* but worse *SSIM* and *PSNR*. We also visualize the pixel-wise L^1 error with the ground truth. We can see that small misalignment causes bigger pixel-wise error but still provide visual pleasant NVS result, while blurriness has smaller, more distributed error, but greatly decrease the visual quality.

(a) Result of **svMPI+svreg** (b) Result of Ours

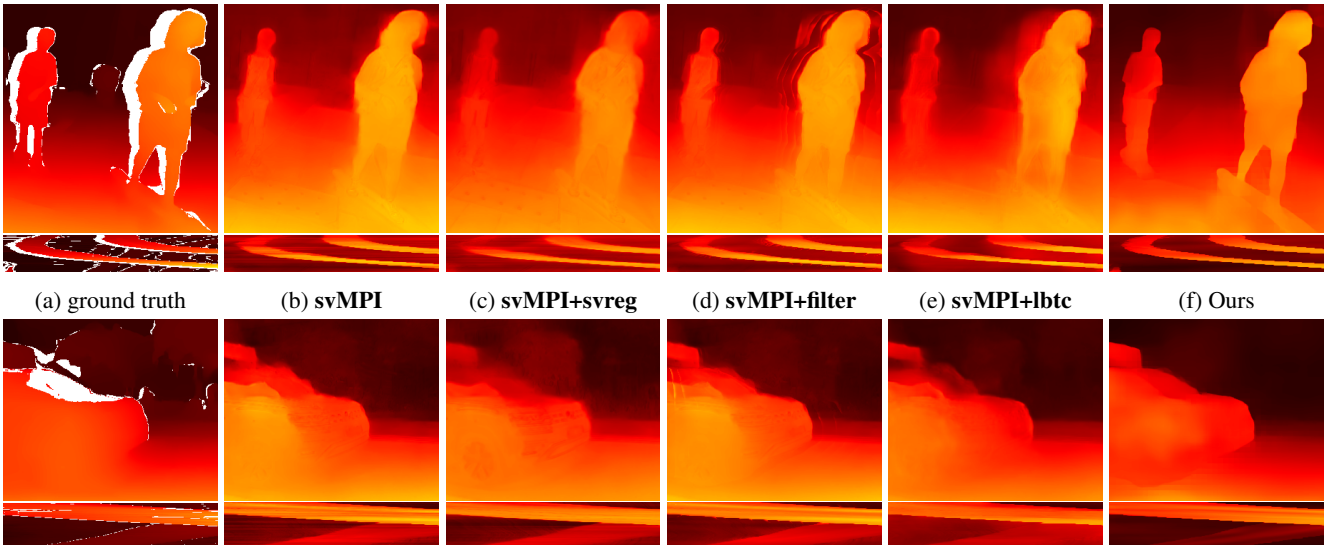
Figure 1: We show an extreme case that our method produces result with worse *SSIM* and *PSNR*, but better *LPIPS* and *FMean* than **svMPI+svreg**. We shown the NVS results in the first row and a visualization of the error map in the second row. We can see that *LPIPS* and *FMean* are more consistent with human perception.

4. More qualitative results

We show more view synthesis results in Figure 3, as well as the predicted disparity map in Figure 2. Besides, we also show several NVS results from DAVIS [4] dataset, which demonstrates the generality of our method. Please refer to the attached video for more intuitive results.

5. Details of Training LBTC modules

Learned Blind Video Temporal Consistency (LBTC) [2] use a ConvLSTM [8] structure to post process a sequence of frames $\mathbb{P} = \{\mathbf{P}_0, \mathbf{P}_1, ...\}$, which are preprocessed by some single frame algorithm from raw input $\mathbb{I} = \{\mathbf{I}_0, \mathbf{I}_1, ...\}$ in a frame-by-frame manner. It is capable of generating temporal consistent sequence $\mathbb{O} = \{\mathbf{O}_0, \mathbf{O}_1, ...\}$ without accurate dense correspondence. We follow the same idea but make some adjustments since the original implementation focuses on texture to texture transformation while in our scenario $\mathbf{O}$ lies in a totally different domain.



(a) ground truth (b) **svMPI** (c) **svMPI+svreg** (d) **svMPI+filter** (e) **svMPI+lbtc** (f) Ours

(g) ground truth (h) **svMPI** (i) **svMPI+svreg** (j) **svMPI+filter** (k) **svMPI+lbtc** (l) Ours

Figure 2: The visualization of predicted disparity maps and temporal consistency of baselines and our method. In the second and fourth rows, the vertical axis indicates timestamps while the horizontal axis indicates the spatial location.

We first increase the channel of each conv since $\mathbf{P}$ is a multi-plane alpha with relatively large channel number. We then redesign the losses for multi-plane alpha:

5.1. Content similarity Loss

[2] use perceptual (VGG) loss to constraint the $\mathbf{O}_t$ to be similar to $\mathbf{P}_t$. This can no longer be used since $\mathbf{P}_t$ is not an ordinary image. Therefore, we encourage the $\mathbf{O}_t$ to be consistent with $\mathbf{P}_t$ by L^1 norm as well as a multi-level gradient loss:

$$\mathcal{L}_p = \sum_{t=1}^{T} \|\mathbf{P}_t - \mathbf{O}_t\|_1 + \lambda_g \sum_{l=0}^{3} \|\mathcal{G}(\mathbf{D}_t^{(l)} - \hat{\mathbf{D}}_t^{(l)})\|_1, \quad (4)$$

where $\mathbf{D}_t$ and $\hat{\mathbf{D}}_t$ are the disparity maps computed from $\mathbf{P}_t$ and $\mathbf{O}_t$ [10], respectively, the superscript l means l-th level in the image pyramid, and T is the total number of frames we use in one training iteration. We follow the same notation as in the paper, where $\|\cdot\|_1$ is the L1 norm over all pixel positions and channels, and $\mathcal{G}(\cdot)$ is the per-pixel L1 norm of the gradient field. We empirically set $\lambda_g = 1$.

5.2. Short term temporal loss

We use the same occlusion-aware warping error as in [2] to constraint the short term temporal consistency:

$$\mathcal{L}_{st} = \sum_{t=2}^{T} \|\mathbf{M}_{t\to t-1} * |\hat{\mathbf{D}}_t - \mathcal{W}(\mathbf{F}_{t\to t-1}, \hat{\mathbf{D}}_{t-1})|_1\|_1 \quad (5)$$

where $|\cdot|$ is the pixel-wise L1 norm, $\mathcal{W}$ is the backward warping function, and $\mathbf{M}_{t\to t-1} = exp(-\alpha|\mathbf{I}_t - \mathcal{W}(\mathbf{F}_{t\to t-1}, \mathbf{I}_{t-1})|_1)$ is the visibility mask. We set α to 50 as in [2].

5.3. Long term temporal loss

The long term temporal consistent loss is also the same as the original implementation:

$$\mathcal{L}_{lt} = \sum_{t=2}^{T} \|\mathbf{M}_{t\to 1}|\hat{\mathbf{D}}_t - \mathcal{W}(\mathbf{F}_{t\to 1}, \hat{\mathbf{D}}_1)|_1\|_1 \quad (6)$$

The final loss is a weighted sum of all the terms:

$$\mathcal{L} = \lambda_p \mathcal{L}_p + \lambda_{st}\mathcal{L}_{st} + \lambda_{lt}\mathcal{L}_{lt}. \quad (7)$$

In the experiment we set $\lambda_p = 10$, $\lambda_{st} = \lambda_{lt} = 100$. Other settings are the same as [2].

6. Practical Aspect

Our compact representation $\mathbf{R}$ can achieve real-time novel view synthesis. All parameters in $\mathbf{R}$ are bounded between $(0, 1)$ by setting the last activation as Sigmoid. We can then quantify parameter maps $\{\mathbf{D}_{fg}, \mathbf{T}_{fg}, \mathbf{D}_{bg}, \mathbf{T}_{bg}\}$ to UINT8 and stack all the maps together with the foreground and background image $\mathbf{I}$ and $\mathbf{B}$. The stacked image sequence can then be encoded and compressed using existing video compression algorithms like H.264. During rendering we implement a customized shader to render the layered MPI in OpenGL and achieve the rendering speed of over $1K$ fps at 800×448 resolution in a commercial laptop.

Since we explicitly estimate a temporal consistent geometry, other interesting applications can be easily achieved

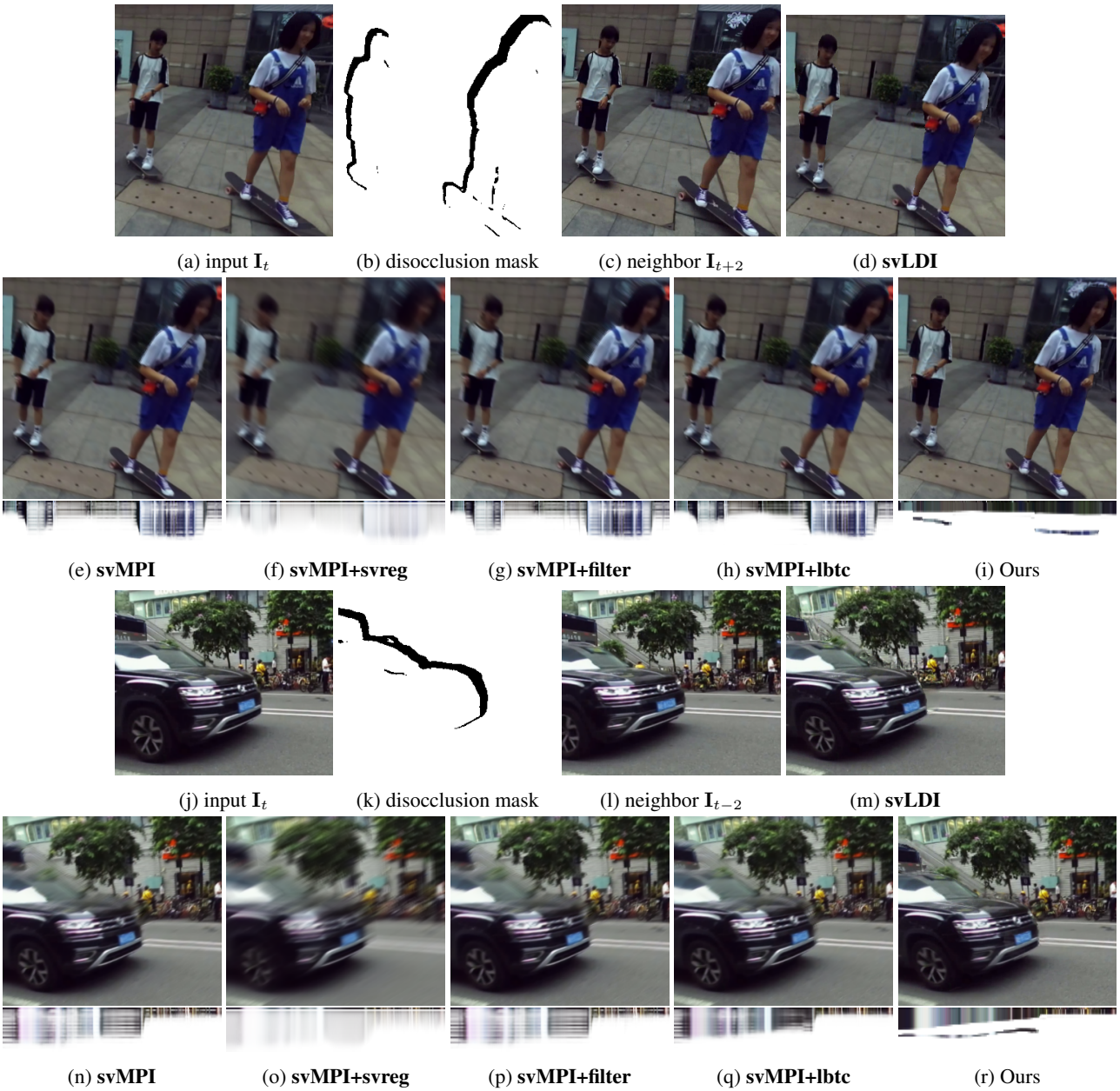


Figure 3: NVS results and MPI visualization of baselines and ours. The visualization is the same as the paper. Note how **svMPI** based methods produce blurry results and **svLDI** generates the inconsistent textures with neighbor frames.

like dolly zoom effect and refocus. Please refer to the attached video for visualization.

References

[1] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. 2

[2] Wei-Sheng Lai, Jia-Bin Huang, Oliver Wang, Eli Shechtman, Ersin Yumer, and Ming-Hsuan Yang. Learning blind video temporal consistency. In *European Conference on Computer Vision*, 2018. 2, 3

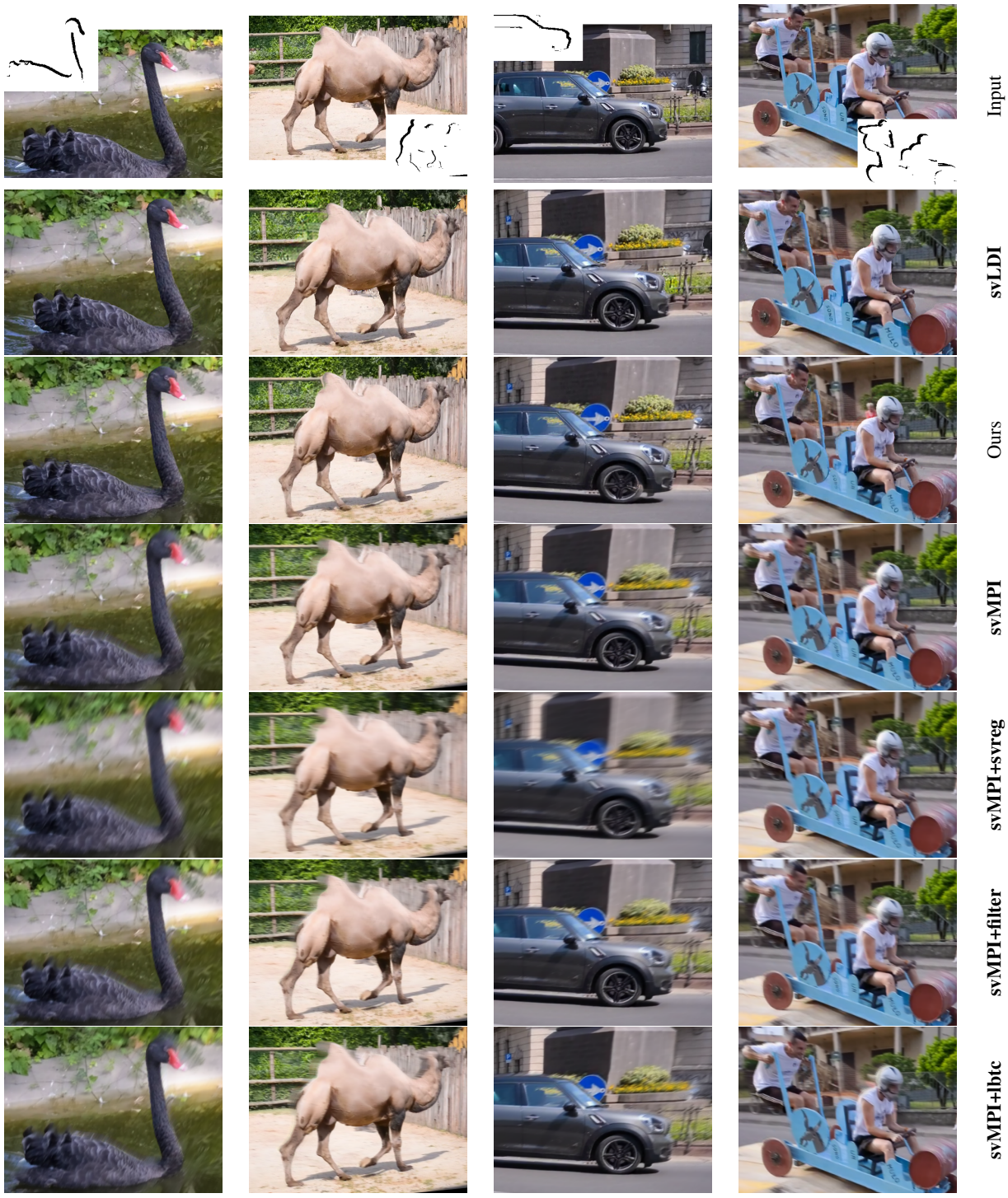


Figure 4: We show NVS results of videos in the wild (from DAVIS [4] dataset). In the first row we visualize the occlusion mask alongside the input frames.

[3] Zhengqi Li, Tali Dekel, Forrester Cole, Richard Tucker, Noah Snavely, Ce Liu, and William T Freeman. Learning the depths of moving people by watching frozen people. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2019. 1

[4] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alexander Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv:1704.00675*, 2017. 2, 5

[5] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2020. 2

[6] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1

[7] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016. 1

[8] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 802–810, Cambridge, MA, USA, 2015. MIT Press. 2

[9] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European Conference on Computer Vision*, pages 402–419. Springer, 2020. 1

[10] Richard Tucker and Noah Snavely. Single-view view synthesis with multiplane images. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2, 3

[11] Chaoyang Wang, Simon Lucey, Federico Perazzi, and Oliver Wang. Web stereo video supervision for depth prediction from dynamic scenes, 2019. 1, 2

[12] Shangchen Zhou, Jiawei Zhang, Wangmeng Zuo, Haozhe Xie, Jinshan Pan, and Jimmy Ren. Davanet: Stereo deblurring with view aggregation. In *CVPR*, 2019. 1

[13] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *ACM Trans. Graph.*, 37(4), July 2018. 1