Supplementary Materials for Learning Rigidity in Dynamic Scenes with a Moving Camera for 3D Motion Field Estimation

Zhaoyang Lv[1]*, Kihwan Kim[2], Alejandro Troccoli[2], Deqing Sun[2],
James M. Rehg[1], Jan Kautz[2]

[1] Georgia Institute of Technology, Atlanta, U.S.
{zhaoyang.lv,rehg}@gatech.edu
[2] NVIDIA, Santa Clara, U.S.
{kihwank,atroccoli,deqings,jkautz}@nvidia.com

1 Visualization for Qualitative Evaluation

Color coding for flow vectors We visualize the flow vectors in 2D following the color encoding in [1], for optical flow $\delta\mathbf{u}^{of}_{0\to1}$, egomotion flow $\delta\mathbf{u}^{cm}_{0\to1}$ and projected scene flow $\delta\mathbf{u}^{sf}_{0\to1}$. The central white color means there is no motion. Hue represents the flow vector direction, and color intensity represents the magnitude. All the flow vectors are normalized to the range [0,1] during visualization, shown in Figure 1. Thus, an accurate estimation of flow should have minimal difference w.r.t. ground truth flow visualization both in hue and intensity.

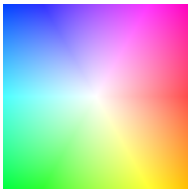


Fig. 1: **Flow color encoding** in all qualitative visualizations. The central white color means there is no motion. Hue represents the flow vector direction and color intensity represents the magnitude. All the flow vectors are normalized to the range of 0-1 during visualization.

We visualize the 3D dense scene flow following the same color encoding in 2D, simply by using the corresponding projected scene flow $\delta\mathbf{u}^{sf}_{0\to1}$ per-pixel. Such color encoding in projected image space can alleviate the noisy estimation for depth close to infinity, which usually has huge uncertainty in scale and thus affects the magnitude normalization.

* This work started during an internship that the author did at NVIDIA.

Table 1: **Optical flow validation comparison** (EPE) on SINTEL[1] set (all images) using different datasets as training from scratch, validated at different number of training iterations. The same PWC-net [2] architecture is use in all training.

SINTEL EPE (clean/final)	6K	30K	60K	90K	120K
FlyingChairs[3]	6.87/7.58	4.27/5.22	3.75/4.66	3.42/4.50	3.36/4.43
FlyingThings3D [4]	8.98/9.89	6.14/7.11	5.57/6.63	5.26/6.28	5.13/6.11
REFRESH (Ours)	5.82/6.54	4.27/5.23	3.85/4.78	3.45/4.48	3.42/4.46

2 Training Optical Flow with REFRESH Dataset

We evaluate our optical flow model [2] trained on REFRESH dataset and compare it against models trained on FlyingChairs [3] and FlyingThings3D [4]. This evaluation serves as a sanity check of our dataset, and more importantly, an indication of its usefulness for scene flow.

Admittedly, the comparison with FlyingChairs [3] is not apple-to-apple. First, the FlyingChairs dataset is for 2D optical flow because it does not provide information such as depth, foreground masks, and camera ego-motion. More critically, the dataset has been tuned to match the statistics of the synthetic SINTEL dataset. However, it is important to check how valid our new dataset is for 2D optical flow, which is a sub-task of scene flow. As discussed earlier, the FlyingThings3D dataset is the only training dataset that satisfies the requirements for training scene flow models[3].

As shown in Table 1, REFRESH dataset converges significantly faster and achieves better results on SINTEL than the FlyingThings3D dataset. The model trained on the REFRESH dataset also has similar performance as the one trained on the FlyingChairs dataset.

3 Test Generalization to the Outdoor Domain

A fair quantitive evaluation on the KITTI dataset is challenging because: (1) the available ground truth depth from LIDAR is sparse for our method, and (2) the portion of moving regions is smaller. However, as an interest to see how our method and the data perform in a completely different domains with above domain discrepancies, we performed a qualitative evaluation on KITTI using the same RTN network trained on our dataset and dense depth calculated from PSMnet [5] output.

The rigidity results show that the RTN can generalize to KITTI reasonably well despite the domain gap and imperfect depth. We find the errors are more likely to happen in regions where the input depth uncertainty is higher and the surfaces are rigid planar, or textureless, which are not covered in our current generated data. This observation may inspire us to generate a mixture of nonrigid and rigid moving objects to improve the dataset diversity.

[3] The Sintel dataset is held for validation and contains much fewer sequences to train scene flow models from scratch.

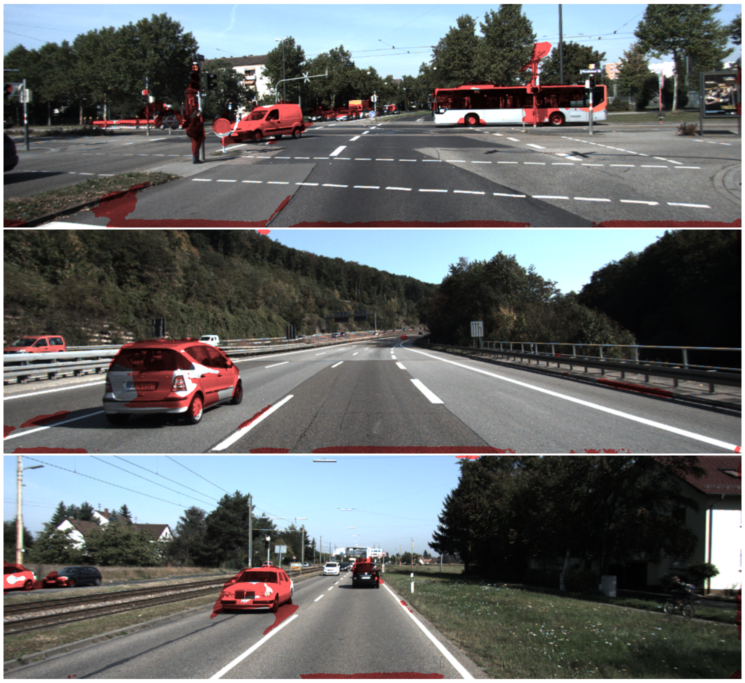


Fig. 2: Rigidity on KITTI with network trained on our REFRESH dataset. There is no finetuning on KITTI data.

4 REFRESH Datasets

4.1 Dataset rendering details

The whole dataset creation is done using Blender 2.78[4], fully automated with python scripts without any GUI interaction, which scales well to the creation of the entire dataset. We separately render the background 3D meshes and foreground nonrigid humans, which allows us to speed up the rendering process. Since we use the raw color image as the background image and only use the geometry ground truth from multi-pass rendering (depth, flow, and segmentation), lighting does not affect background rendering with or without the foreground. Such separation can significantly boost the dataset creation speed. With a 28-core CPU server, we can finish the entire rendering process using BundleFusion [6] 3D scenes in two days.

Background Static Mesh Rendering Since we do not use the rendered color images in any process, we use a simplified setting for background rendering without ray-tracing, tile size as 512 × 512. The rendering time depends on the size of 3D mesh size. In average, we render one frame in $(< 1s)$ in CPU, and we can finish the frame-by-frame rendering of 8 scenes of BundleFusion in 10 hours.

[4] Blender: https://www.blender.org/

Foreground Nonrigid Human Rendering We create the human bodies following SURREAL [7] with synthetic textures (772 clothes textures and 158 CAESAR textures). The illuminated textures are used as the appearance of humans in our composted dynamic scenes. We use spherical harmonics with nine coefficients [8], with ambient illumination coefficient randomly sampled from [0.5, 1.5] and other coefficients randomly sampled from [-0.7, 0.7]. We implement this part by refactoring over [7], by extending SURREAL to arbitrary humans bodies with random textures and actions.

We split the camera trajectory into multiple clips. Each clip is a continuous 100-frame sequence, with randomly loaded human models and actions. There are two major motivations to rendering the outputs in clips rather than an entire trajectory: 1. We can load different random human bodies and motions for different clips in the same trajectory, which increase the motion diversity both in action and appearance; 2. There are numerous human models generated along the entire trajectory, which composes complex meshes in 3D and slow for rendering. Rendering individual clip with several human models is much faster in execution. We can render multiple pass image ground truth with an average of 3 seconds per frame.

Create Ground Truth We use Blender Cycles rendering passes to extract the per-pixel ground truth. We use the *Vector* node to retrieve the 2D vectors giving the frame by frame motions towards to the next and previous frame positions in pixel space, which are denoted as the forward/backward optical flow. Note that we currently do not retrieve the 3D motion vector representation of scene flow from Blender as one pass, which can be an extension to the current dataset in the future work.

We use the rendered depth from 3D scenes instead of the raw 3D scene depth for all the training. Compared to the raw depth, the rendered depth is less noisy and contains less missing measurements and has a per-pixel correspondence to the other ground truth, e.g., optical flow. However, the rendered depth does not guarantee a valid per-pixel value due to the incomplete 3D reconstruction from raw measurements. We marked the projected pixels from incomplete regions (holes in 3D reconstruction) as *invalid* region, and exclude them from the training on-the-fly.

4.2 Dataset statistics

We rendered dataset using the optimized camera trajectory during 3D reconstruction as the camera extrinsic setting. Since the camera movement during 3D acquisition is small and stable between frames, we also use the sampled keyframes from the camera trajectory during rendering. We name the sub-sample trajectory based on their frame interval n as *keyframe n*: $keyframe1$ represents that we use every frame along the trajectory during dataset creation and $keyframe10$ represents we use every ten frames. We list the number of static scene frames with varying keyframes in Table 2.

Figure 3 shows the histogram distributions of our outputs in optical flow, depth, and rigidity from the rendered REFRESH dataset. We show the histogram distribution independently for the data rendered from different keyframes (1,2,5). Compare different keyframe splits, the distribution in depth and non-rigid area ratio in the images are similar and when using larger keyframes, the output optical flow tends to have a larger displacement. When using rendered outputs from larger keyframes, we can simulate the observations from a camera with larger motions.

During training, we empirically find the network generalize the best when using keyframe [1,2,5] from the optimized trajectory from BundleFusion. We use the first seven scenes in BundleFusion as our training set ('apt0', 'apt1', 'apt2', 'copyroom', 'office0', 'office1', 'office2') as our training set with a total of 69218 pairs of frames, and use 'office3' as the validation set with 6390 pairs of frames.

4.3 Visualization

We visualize some examples of our datasets in Figure 4 across different scenes. The invalid regions are visualized as black in the depth image and white in the ground truth optical flow.

5 Evaluation on SINTEL Dataset

Quantitative Evaluation on Entire SINTEL Dataset

We evaluate our method using RTN and refine step on the all frames in entire SINTEL dataset compared to the two baseline methods in the paper as a supplement to the comparison in our test set split. First one is *refinement only*, which we denote as solving the refinement stage without any information acquired from RTN. Secondly, we compare our method to semantic rigidity estimation [9], which assumes that the non-rigid motion can be predicted from its semantic labeling. The semantic network is trained using the DeepLab [10] architecture with weights initialized from the pre-trained MS-COCO model on the same data we used for our model. In the pose refinement stage, we substitute our rigidity from RTN with the semantic rigidity. Both baselines use the same optical flow network with the same weights, and all methods use the same depth

Table 2: The number of rendered images generated in our REFRESH dataset using BundleFusion [6] as 3D scenes.

	apt0	apt1	apt2	copyroom	office0	office1	office2	office3	Total
keyframe 1	8560	8495	3873	4478	6083	5727	3494	3757	44467
keyframe 2	4280	4248	1937	2239	3043	2863	1748	1882	22240
keyframe 5	1712	1700	776	895	1220	1146	700	752	8901
keyframe 10	856	849	338	447	609	572	349	376	4446
keyframe 20	427	424	195	223	304	286	174	189	2222
keyframe 50	171	169	78	89	123	114	69	75	888
Total	16006	15885	7247	8371	11382	10708	6534	5149	83164

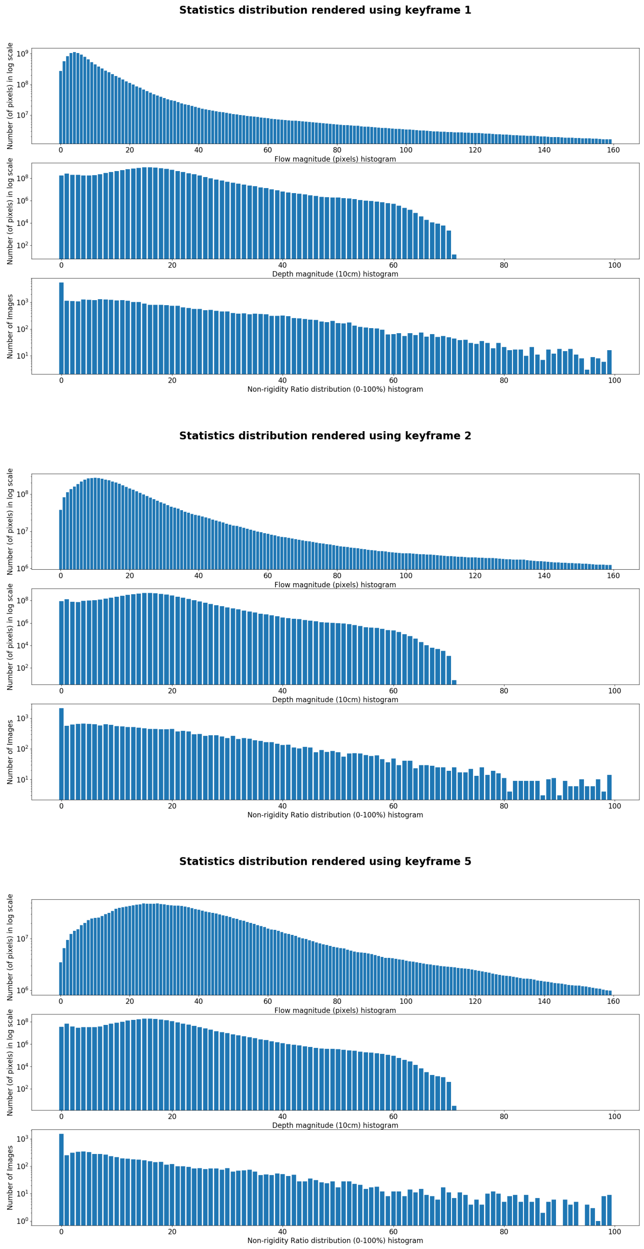
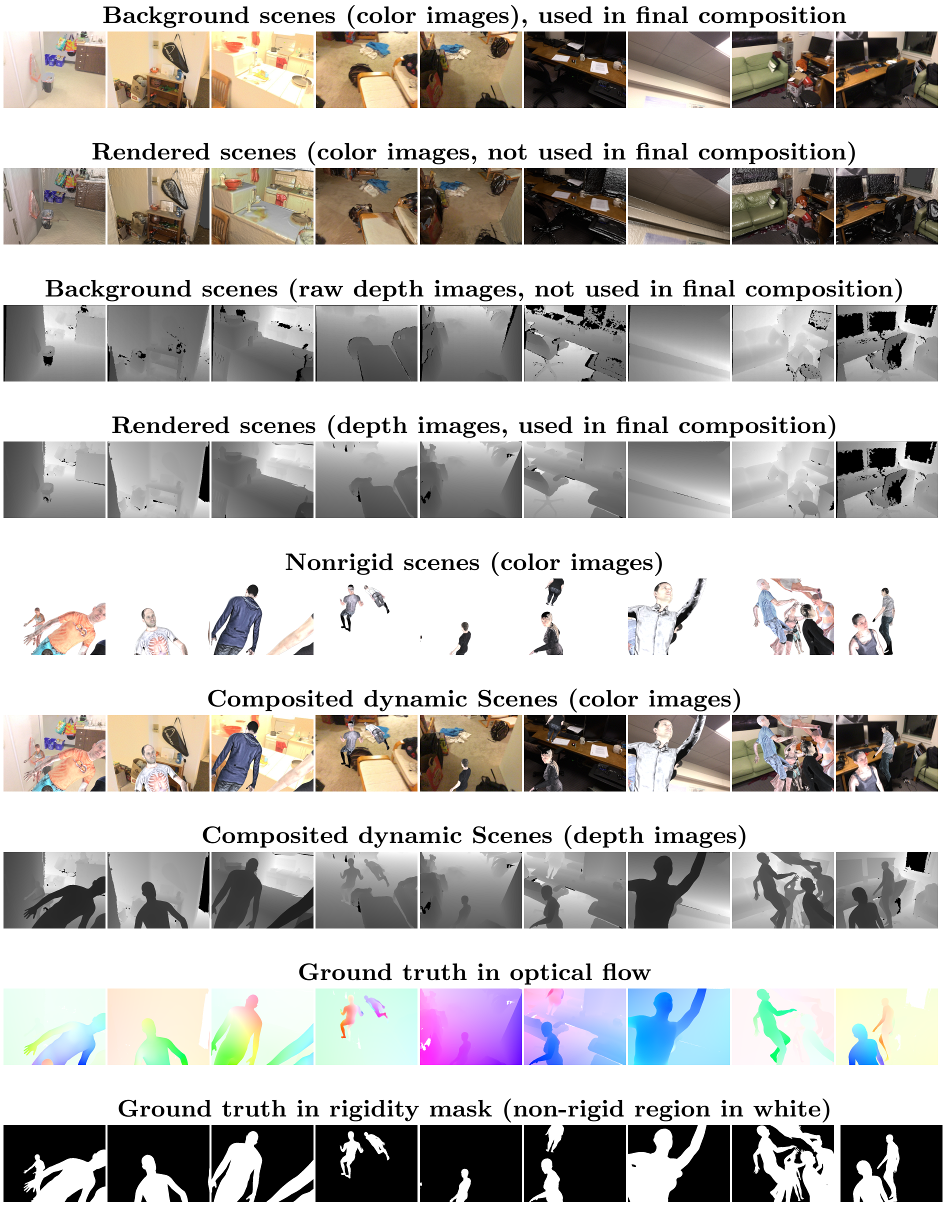


Fig. 3: Histogram distributions of optical flow, depth, and rigidity from our rendered REFRESH dataset in the training set. We calculate the distribution from three splits using keyframes 1, 2, 5 independently. In each of the split, we show the flow magnitude distribution (top) in pixels, depth distribution (medium) in centimeters, and nonrigid ratio (belows) in the number of different images.



Background scenes (color images), used in final composition

Rendered scenes (color images, not used in final composition)

Background scenes (raw depth images, not used in final composition)

Rendered scenes (depth images, used in final composition)

Nonrigid scenes (color images)

Composited dynamic Scenes (color images)

Composited dynamic Scenes (depth images)

Ground truth in optical flow

Ground truth in rigidity mask (non-rigid region in white)

Fig. 4: **Qualitative visualization of Frames in REFRESH Datasets.**

Table 3: Quantitative Evaluation on SINTEL dataset using all frames. All models in this evaluation are *not finetuned* and trained on REFRESH dataset. We report the EPE in egomotion flow (EF) and projected scene flow (PSF). The number in *failures* indicate the number of frames that has an EPE over 100, which is excluded in the EPE calculation. For all the baseline methods, we use the same optical flow network trained as our method. The lowest residual under the same setting (e.g. clean set) is highlighted as **bold**.

	Final Pass All			Clean Pass All		
	EF	PSF	failures	EF	PSF	failures
Refine (from flow only)	2.71	6.81	19	2.61	6.67	9
Semantic rigidity [9] + refine	6.19	9.35	25	4.57	7.68	12
RTN + Refine	**1.78**	**5.81**	**17**	**1.75**	**5.72**	**6**

from SINTEL ground truth. We use the EPE in egomotion flow and projected scene flow as a metric. To exclude the effects of some catastrophic failures in some particular frames, we exclude those frames that have over 100 EPE values and separately count them as failure cases. None of the models are finetuned on SINTEL dataset. Table 3 shows the quantitative evaluation. It is worth to note that predicting rigidity based on semantics cannot generalize well across different domains, which can lead to bad rigidity localization that significantly harm the correspondence association. This evaluation also shows our method outperforms the two baseline methods.

References

1. Butler, D.J., Wulff, J., Stanley, G.B., Black, M.J.: A naturalistic open source movie for optical flow evaluation. In: European Conf. on Computer Vision (ECCV), The Royal Society (2012) 611–625
2. Sun, D., Yang, X., Liu, M.Y., Kautz, J.: PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). (2018)
3. Dosovitskiy, A., Fischer, P., Ilg, E., Häusser, P., Hazırbaş, C., Golkov, V., v.d. Smagt, P., Cremers, D., Brox, T.: Flownet: Learning optical flow with convolutional networks. In: Intl. Conf. on Computer Vision (ICCV). (2015) 2758–2766
4. Mayer, N., Ilg, E., Hausser, P., Fischer, P., Cremers, D., Dosovitskiy, A., Brox, T.: A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). (2016) 4040–4048
5. Chang, J.R., Chen, Y.S.: Pyramid stereo matching network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2018) 5410–5418
6. Dai, A., Nießner, M., Zollöfer, M., Izadi, S., Theobalt, C.: BundleFusion: real-time globally consistent 3D reconstruction using on-the-fly surface re-integration. ACM Transactions on Graphics 2017 (TOG) (2017)
7. Varol, G., Romero, J., Martin, X., Mahmood, N., Black, M.J., Laptev, I., Schmid, C.: Learning from Synthetic Humans. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). (2017)
8. Green, R.: Spherical Harmonic Lighting: The Gritty Details. Archives of the Game Developers Conference (March 2003)
9. Wulff, J., Sevilla-Lara, L., Black, M.J.: Optical flow in mostly rigid scenes. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). (July 2017)
10. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. arXiv:1606.00915 (2016)