The Visual Computer
https://doi.org/10.1007/s00371-024-03263-z

ORIGINAL ARTICLE

The devil in the details: simple and effective optical flow synthetic data generation

Byung-Ki Kwon[1] · Sung-Bin Kim[2] · Tae-Hyun Oh[1,2,3]

Accepted: 2 January 2024

Abstract
Recent work on dense optical flow has shown significant progress, primarily in a supervised learning manner requiring a large amount of labeled data. Due to the expensiveness of obtaining large-scale real-world data, computer graphics are typically leveraged for constructing datasets. However, there is a common belief that synthetic-to-real domain gaps limit generalization to real scenes. In this paper, we show that the required characteristics in an optical flow dataset are rather simple and present a simpler synthetic data generation method that achieves a certain level of realism with compositions of elementary operations. With 2D motion-based datasets, we systematically analyze the simplest yet critical factors for generating synthetic datasets. Furthermore, we propose a novel method of utilizing occlusion masks in a supervised method and observe that suppressing gradients on occluded regions serves as a powerful initial state in the curriculum learning sense. The RAFT network initially trained on our dataset outperforms the original RAFT on the two most challenging online benchmarks, MPI Sintel and KITTI 2015.

Keywords Curriculum learning · Deep learning · Optical flow · Synthetic data

1 Introduction

Optical flow provides the clues of motion between subsequent frames, which can be utilized for other computer vision tasks such as object tracking, action recognition, 3D reconstruction, and video enhancement, *etc*. Recently, deep neural networks have shown great progress in optical flow estimation [12, 15, 29–31]. The progress has been made primarily in a supervised learning manner requiring a large amount of labeled data. Despite the effectiveness of the learning-based approaches, obtaining labeled real-world data is prohibitively expensive at a large scale. Therefore, synthetic computer graphics data [1, 3, 6, 23] are typically leveraged.

A common belief of using synthetic data is that the data rendered by graphics engines limit generalization to real scenes due to synthetic-to-real domain gaps in quality. Those gaps involve real-world effects such as noise, 3D motion, non-rigidity, motion blur, occlusions, large displacements, and texture diversity. Thus, synthetic datasets [1, 3, 6, 23] for optical flow have been developed by considering these effects to some extent, *i.e.*, mimicking the real-world effects.

In this paradigm, we throw a question, “Which factor of the synthetic dataset is essential for the generalization ability to the real domain?” In this work, we found that the required characteristics for an optical flow dataset are simple; achieving only a certain level of realism is enough for training highly generalizable and accurate optical flow models. We empirically observe that a simple 2D motion-based dataset as training data often shows favorable performance for ordinary purposes or much higher than the former synthetic datasets [1, 22], which are rendered by complex 3D object or motion with rich textures. Furthermore, we found that using occlusion masks to give the network incomplete information is effective for a powerful initial state of curriculum learning.

✉ Tae-Hyun Oh
taehyun@postech.ac.kr

Byung-Ki Kwon
byungki.kwon@postech.ac.kr

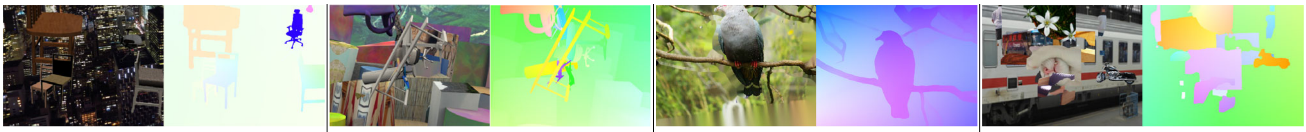
Sung-Bin Kim
sungbin@postech.ac.kr

[1] Graduate School of AI, POSTECH, Pohang, South Korea

[2] Department of Electrical Engineering, POSTECH, Pohang, South Korea

[3] Institute for Convergence Research and Education in Advanced Technology, Yonsei University, Seoul, South Korea

Published online: 17 February 2024



(a) FlyingThings3D (Mayer et al., 2016) (b) FlyingChairs (Dosovitskiy et al., 2015) (c) dCOCO (Aleotti et al., 2021) (d) Ours

Fig. 1 The prior arts of synthetic data and our proposed dataset. Sampled frames and its corresponding flow maps are visualized. While being diverse in motion, **a**,**b** include many thin object parts and unrealistically simple reflectance. **c** includes semantically coherent flow map but the diversity of the motion is limited by a global camera motion. Our method, in contrast, includes both controllable and diverse motion characteristics with semantically coherent object shapes and rich texture

We design easily controllable synthetic dataset generation recipes using a cut-and-paste method with segmented 2D object textures. As shown in Fig. 1, our generated data appears to be far from the real-world one, but training on those shows promising results both on generalization and fine-tuning regimes, outperforming the networks trained on the competing datasets. We also utilize occlusion masks to stop gradients on occluded regions, and the RAFT network initially trained with occlusion masks outperforms the original RAFT on the two most challenging online benchmarks, MPI Sintel [3] and KITTI 2015 [24]. Our key contributions are summarized as follows: (1) We present simple synthetic data generation recipes with compositions of simple elementary operations and show comparable performance against competing methods, (2) we propose a novel method of utilizing occlusion masks in a supervised method and show that suppressing gradients on occluded regions in a supervised optical flow serves as a powerful initial state in the curriculum learning protocol, and (3) we systematically analyze our dataset and the effects according to different factors of motion type, motion distribution, data size, texture diversity, and occlusion masks.

2 Related work

We briefly review our target task, *i.e.*, optical flow estimation, and the training datasets that have been used for training learning-based optical flow estimation methods.

Optical Flow Fundamentally, optical flow estimation for each pixel is an ill-posed problem. Traditional approaches [2, 11, 25, 33] attempted to deal with imposing smoothness priors to regularize the ill-condition in an optimization framework. According to the advance of deep learning, the ill-posedness has been tackled by learning, yielding superior performance. Starting with the success of FlowNet [6, 15], Recent optical flow estimation methods have been developed by coarse-to-fine approaches [10, 29, 32] or iterative refinement approaches [13, 31]. However, these approaches strongly rely on training datasets, where real supervised data of optical flow is extremely difficult to obtain [22].

Datasets The supervised learning-based methods for optical flow estimation requires exact and pixel-accurate ground truth. While obtaining true real motion is extremely difficult without the support of additional information, several real-world optical flow datasets [9, 16, 20, 24] have been proposed. However, these datasets are relatively small scale and biased to limited scenarios; thus, those are not sufficient for training a deep model but more suitable for benchmark test sets.

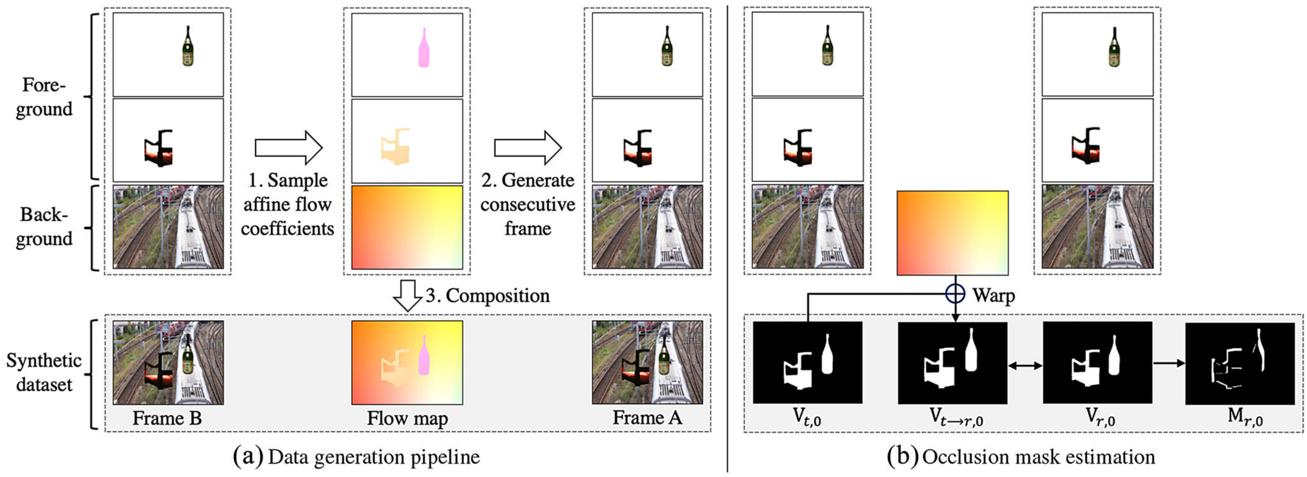
To address persistent data scarcity, studies for generating large-scale synthetic datasets have been attempted. Dosovitskiy et al. [6] propose a synthetic dataset of moving 3D chairs superimposed on the images from Flickr. Similarly, Mayer et al. [23] present datasets where not only chairs but various objects are scattered in the background. Aleotti et al. [1] leverage an off-the-shelf monocular depth network to synthesize a novel view from a single image and compute an accurate flow map.

Mayer et al. [22] present critical factors of the synthetic dataset, *i.e.*, the object shape, motion types and distributions, textures, real-world effects, data augmentation, and learning schedules. Prior work [5, 28] generate a learning-based synthetic dataset for training accurate optical flow networks, but it is still challenging to distinguish the key factors for synthetic data intuitionally. We build upon the observations of Mayer et al. [22] and design easily controllable synthetic dataset generation recipes and identify additional key factors such as *balanced motion distribution, amount of data, texture combination, and learning schedules with occlusion masks.*

3 Data generation pipeline

In this section, we present a simple method to generate an effective optical flow dataset. Unlike the prior arts using 3D motions and objects with computer graphics, our generation scheme remains simple by using 2D image segment datasets and 2D affine motion group. The proposed simple dataset enables analyzing the effect of each factor of the synthetic dataset.

Overall Pipeline The overall data generation pipeline is illustrated in Fig. 2. As shown, we use a simple cut-and-paste



(a) Data generation pipeline

(b) Occlusion mask estimation

Fig. 2 Schematic overview of our data generation pipeline and occlusion mask estimation. **a** Given a background image and foreground objects, we sample affine flow coefficients and generate a consecutive frame. These coefficients can be used to extract exact ground-truth optical flow map. **b** We describe the process of estimating the occlusion mask ($M_{r,i}$) for the first layer ($i = 0$), which is the background. This process is recursively conducted in ascending order until the end of the layers

method where foreground objects are pasted on an arbitrary background image. Inspired by Oh et al.[26], the segmented foreground objects and random background images are obtained from two independent datasets to encourage combinatorial diversity while avoiding texture overlaps. In this work, we use PASCAL VOC [7] and MS COCO [21] as suggested by Oh et al. [26]. The foreground objects are first superimposed randomly, and its consecutive frame is composed of randomly moving both the foreground objects and the background image by simple affine motions. This allows us to express diverse motions, easily control the motion distribution, and compute occlusion masks.

Background Processing We first sample an image from an image dataset for background and resize them to 712 × 584. We regard this frame as the target frame (Frame B in Fig. 2). Then, we generate a flow map using random affine coefficients, including translation, rotation, and scaling (zooming), and inverse-warp the target frame to obtain the reference frame (Frame A in Fig. 2). We sample the translation coefficient of background from the range $[-20, 20]$ pixels for each direction, and with a 30% chance, the translation coefficient is reset to zero. The rotation and scale coefficients are sampled from $[-\frac{\pi}{100}, \frac{\pi}{100}]$ and $[0.85, 1.15]$, respectively. From the sampled affine matrix, we obtain a ground-truth flow map by subtracting the coordinates of two background image pairs as $\mathbf{f} = \mathbf{Ax} - \mathbf{x}$, where $\mathbf{f}$ denotes each flow vector of a pixel at the reference frame, $\mathbf{A}$ the affine transform, and $\mathbf{x}$ a homogeneous coordinate $[x, y, 1]$ of each pixel on the reference frame. We sample 7,849 background images from MS COCO [21].

Foreground Processing For synthesizing foreground objects' motion, we use segmented objects from a semantic image segmentation dataset. For the target frame, we first sample the number of foreground objects to be composited in $\{7, 8, \cdots, 14, 15\}$. Then, we randomly place these objects on the target one and apply inverse-warping to obtain the warped objects on the reference frame using optical flow maps obtained from random affine transformations. The sampling ranges of rotation and scale coefficients are the same as those of the background case. The distribution of the translation coefficient is designed to follow the exponential distribution as $\frac{1}{Z}\exp(-f/T)$, where the temperature T is empirically set to 20, and Z the normalization term. The distribution is inspired by natural statistics of optical flow [27], where the statistics of motions tend to follow Laplacian distribution. We limit the distribution range $[0, 150]$ by resampling if the magnitude is over 150 pixels. The translation direction of foregrounds is sampled at uniformly random. We use 2913 images from PASCAL VOC [7], and from the set, we extract 5543 preprocessed segments as foreground objects.

Composition We sequentially paste foregrounds on the background to generate a single pair of consecutive frames. The flow maps of each foreground are pasted only when the alpha channel value is at least the threshold c. Following the implementation details of [28], we set c to 0.4 and empirically found the performance is not sensitive to the setting of the threshold.

After composition, we conduct the center crop to the composited images to obtain outputs of size 512 × 384 which is the same as FlyingChairs [6]. Our data generation speed is faster than AutoFlow [28], which generates a learning-based dataset for given target data, and ours about 500 times faster than dCOCO [1] as shown in Table 1. Our fast data generation

Table 1 Data generation speed

	Dataset	Number of foregrounds	100 pairs generation time
(A)	AutoFlow [28]	–	336 days
(B)	dCOCO [1]	–	5593.2 s
(C)	Ours	2	6.86 s
(D)	Ours	7	9.49 s
(E)	Ours	15	12.98 s

We evaluate the speed for generating 100 pairs of synthetic data with a single NVIDIA Titan RTX GPU: (A) dCOCO, and (B, C, D) ours with the different number of foregrounds. The number of the foregrounds is sampled between 7 to 15

speed is beneficial for analyzing the required characteristics to train accurate optical flow networks.

Occlusion Mask Similar to the prior arts [3, 9, 23, 24], our data generation method exports occlusion masks as well. Predicting motions of regions being occluded is an intractable problem and requires uncertain forecasting, which can act as detrimental outliers during training. Thus, prior arts [14, 17] estimate occlusion masks as well to encourage reliable optical flow estimation. Unlike prior arts, we utilize occlusion masks in a supervised method by suppressing gradients on occluded regions in a supervised optical flow. The gradient suppression with occlusion masks serves as a powerful initial state in the curriculum learning protocol, which will be discussed in the experimental section. To obtain occlusion masks, given the alpha maps of each layer including foregrounds ($i \geq 1$) and background ($i = 0$) in order, we binarize the alpha map by thresholding with 0.4, denoting $\alpha_{\{r,t\},i}$ for the i-th object layer in the reference and target frames, respectively. The non-visible regions $\mathrm{V}_{\{r,t\},i}$ of the i-th layer in each frame are computed by $\mathrm{V}_{\{r,t\},i} = \alpha_{\{r,t\},i} \cap (\cup_{k=i+1}^{L} \alpha_{\{r,t\},k})$. Using the i-th layer flow map $\mathbf{f}_i$, we inverse-warp the $\mathrm{V}_{t,i}$ to the reference frame as $\mathrm{V}_{t\rightarrow r,i} = \mathbf{f}_i \circ \mathrm{V}_{t,i}$ and binarize it by 0.4 again, where $\circ$ denotes the warping operation. Then, because the occluded regions are only visible in the reference frame, we can find such an occlusion mask of each layer by $\mathrm{M}_{r,i} = \max(\mathrm{V}_{t\rightarrow r,i} - \mathrm{V}_{r,i}, 0)$. The compromised occlusion mask M_r is obtained by $\mathrm{M}_r = \cup_{i=0}^{L} \mathrm{M}_{r,i}$.

4 Experiments

In this section, we compare the performance of respective optical flow networks by training on our datasets with/without the occlusion mask and competing datasets. Utilizing the simple data generation recipe, we also analyze the effects of characteristics in optical flow datasets.

Optical Flow Network We use RAFT [31] as a reference model to evaluate the benefits of our synthetic dataset in generalization and fine-tuning setups. RAFT is a representative supervised model that is widely used to estimate the effectiveness of optical flow datasets [1, 28]. We follow the same hyper-parameters suggested by the implementation of [31], and the experiment setup by Aleotti et al. [1] that shows one-/multi-stage training results. For our synthetic datasets, in the initial training stage, we train RAFT for 100k iterations with the batch size[1] of 10, image crops of size 496 × 368, the learning rate 4×10^{-4}, and the weight decay of 1×10^{-4}.

For multi-stage training with FlyingThings3D [23], from the RAFT networks pre-trained on our datasets, we further train with the `frames_cleanpass` split of FlyingThings3D that includes 40k consecutive frame pairs. We train the model for 100k iterations with a batch size of 6, image crops of size 720 × 400, the learning rate of 1.25×10^{-4}, and the weight decay of 1×10^{-4}. These hyper-parameters are the same with the *Things training stage* reported in [31].

Competing Datasets for Training We choose FlyingChairs (Ch) [6] and dCOCO [1] as the competing datasets, and leverage the RAFT networks pre-trained on each dataset provided by the authors and dCOCO. For multi-stage training models, from the networks pre-trained on ours, we further train with FlyingThings3D (Th) [23] in sequence to compare with the RAFT model trained with FlyingChairs followed by FlyingThings3D (Ch→Th).

Test Datasets We evaluate on Sintel [3] and KITTI 2015 [24]. These datasets contain crucial real-world effects, such as occlusions, illumination changes, motion blur, and camera noise, making them challenging and widely used standard benchmarks for evaluating optical flow models. We report the performance of the model trained with the base datasets without fine-tuning on Sintel or KITTI, called *generalization* and that of the model fine-tuned on the training set of Sintel or KITTI, called *fine-tuning*.

Evaluation Following the convention, we report the average End-Point Error (EPE) and the errors that exceed 3 pixels and 5% of its true value (Fl). We further evaluate the percentage of pixels with an absolute error smaller or equal to 1 ($\leq$1). The **bold** will be used to highlight the best one among the methods.

4.1 Comparison with other synthetic datasets

We compare the generalization and fine-tuning performance of the networks trained on our dataset and other competing datasets [1, 6, 23]. For fair comparisons, we train the network on our dataset (denoted as Ours) with 20k image pairs that include translation, rotation, and zooming. We also evaluate our dataset with occlusion masks $\langle \mathrm{O} \rangle$ (denoted as Ours+O).

Generalization The left part of Table 2 summarizes the generalization test. Among the models trained on a single dataset,

[1] The authors of [1, 31] use the batch size of 12 and 6 for training FlyingChairs and dCOCO, respectively.

Table 2 Comparison with other datasets.

	Dataset	Motions	Generalization test								Fine-tuning test							
			Sintel C.		Sintel F.		KITTI12		KITTI15		Sintel C.		Sintel F.		KITTI12		KITTI15	
			EPE	≤1	EPE	≤1	EPE	Fl	EPE	Fl	EPE	≤1	EPE	≤1	EPE	Fl	EPE	Fl
(A)	Ch	2D	2.28	0.79	4.51	0.72	4.66	30.54	9.85	37.56	0.89	0.93	1.49	0.89	1.39	4.69	2.36	8.43
(B†)	dCOCO	3D	–	–	–	–	–	–	–	–	–	–	–	–	1.37	4.70	2.76	9.15
(B)	dCOCO	3D	2.62	0.45	3.90	0.39	**1.82**	**6.62**	**3.81**	**12.43**	1.08	0.92	1.84	0.88	1.37	4.76	2.57	8.81
(C)	Ours	2D	**1.98**	**0.86**	3.85	**0.82**	3.63	20.00	7.17	29.24	**0.85**	**0.94**	1.40	**0.89**	**1.33**	4.37	2.20	8.19
(D)	Ours+O	2D	2.02	**0.86**	**3.67**	**0.82**	3.66	19.37	7.88	28.41	0.89	0.93	**1.39**	**0.89**	1.35	**4.36**	**2.15**	**7.60**
(E)	Ch → Th	2D+3D	1.47	0.90	**2.79**	0.85	2.15	9.30	5.00	17.44	0.84	0.93	1.31	0.89	1.31	4.25	2.28	7.96
(F)	Ours→Th	2D+3D	**1.29**	**0.91**	2.81	0.85	2.04	9.02	**4.77**	16.72	**0.83**	**0.94**	**1.29**	**0.90**	1.32	4.24	2.10	7.52
(G)	Ours+O→Th	2D+3D	**1.29**	**0.91**	2.86	**0.86**	**2.03**	**8.64**	4.84	**16.38**	0.86	**0.94**	1.31	**0.90**	**1.28**	**4.11**	**2.02**	**7.34**

We evaluate the generalization and fine-tuning test of the RAFT networks trained on training datasets: (A) FlyingChairs, (B) dCOCO, (C) ours, (D) ours with occlusion mask, (E) FlyingChairs and FlyingThings3D, (F) ours and FlyingThings3D, and (G) ours with occlusion mask and FlyingThings3D. (B†) is obtained from the original paper of [1]

Table 3 Generalization results on other benchmarks

	Dataset	HD1K (real)	Virtual KITTI (synthetic)
(A)	Ch	1.70	6.52
(B)	dCOCO	1.44	**3.92**
(C)	Ours	**1.06**	6.38

We evaluate the generalization test of the RAFT networks on HD1K [20] and Virtual KITTI [8]

our datasets (C, D) show the best performance on Sintel. However, dCOCO (B) shows better performance on KITTIs. We further evaluate the performance on two other benchmarks as shown in Table 3, and observe that dCOCO achieves better performance on Virtual KITTI [8], which is a synthetic dataset. On the other hand, ours achieves more accurate optical flow estimation in a real dataset, *i.e.*, HD1K [20]. From these results, we assume that dCOCO, which uses depth-aware data generation approach with real images, is effective in autonomous driving scenarios and the similar motion distribution and texture between the synthetic and target dataset are key factors of generalization. We also pre-train the network on 2D motion datasets, such as FlyingChairs [6] and our datasets, and sequentially train on FlyingThings3D [23]. Compared to (E) which uses FlyingChairs at the initial stage, (F, G) show better generalization performance in the KITTIs and Sintel Clean pass. These show that the choice of the initial training stage significantly affects the final performance.

Fine-tuning We fine-tune the networks of the left part of Table 2 on Sintel or KITTIs, and the results are reported in the right part of the table. Overall, our datasets show favorable performance. Compared to (E) first pre-trained on FlyingChairs, (F, G) show better performance. (G) especially achieves the lowest Fl and noticeable performance improvement in KITTI 2015. These results suggest that utilizing occlusion masks as a gradient suppression tool is effective in fine-tuning real-world datasets, *i.e.*, KITTI 2012 and KITTI 2015. We observe a consistent tendency with the online benchmark results as follows.

Online Benchmarks We follow the training procedure described in RAFT [31] to fine-tune the model pre-trained by our dataset and test on the public benchmarks of Sintel and KITTI 2015. As summarized in Table 4, using our dataset for the initial curriculum outperforms the original RAFT on both public benchmarks. On the KITTI 2015 test set, the network pre-trained on our synthetic dataset with occlusion masks shows better performance compared to RAFT. In the Sintel test dataset, we observe that the performance improvement in Sintel Clean and Final passes with our dataset. With and without the *warm-start* initialization, the network trained with our training schedule also achieves better results in both passes. From these results, we assume that learning the simplest characteristics for estimating optical flow at the initial learning schedule without occlusion estimation helps the network perform better.

Other Backbone Networks To evaluate the effectiveness of our dataset other than RAFT, we selected two more optical flow models: FlowNet [6] and PWC-Net [29]. We use the re-implementation of FlowNet [2] and PWC-Net .[3] Table 5 shows the result of each network trained on our dataset outperforming the one trained on FlyingChairs [6]. We also contain the previous experiment with RAFT in (C) as a reference. These results prove that the simple properties of our dataset are effective for not only the RAFT [31], but also general optical flow networks.

[2] https://github.com/ClementPinard/FlowNetPytorch.

[3] https://github.com/visinf/irr.

Table 4 Test results on Sintel and KITTI 2015

	Training methods	w/*warm-start*		wo/*warm-start*		-
		Sintel C	Sintel F	Sintel C	Sintel F	KITTI15
		EPE	EPE	EPE	EPE	Fl
(A)	RAFT	1.61	2.86	1.94	3.18	5.1
(B)	RAFT-Ours+O	**1.59**	**2.83**	**1.81**	**3.10**	**4.91**

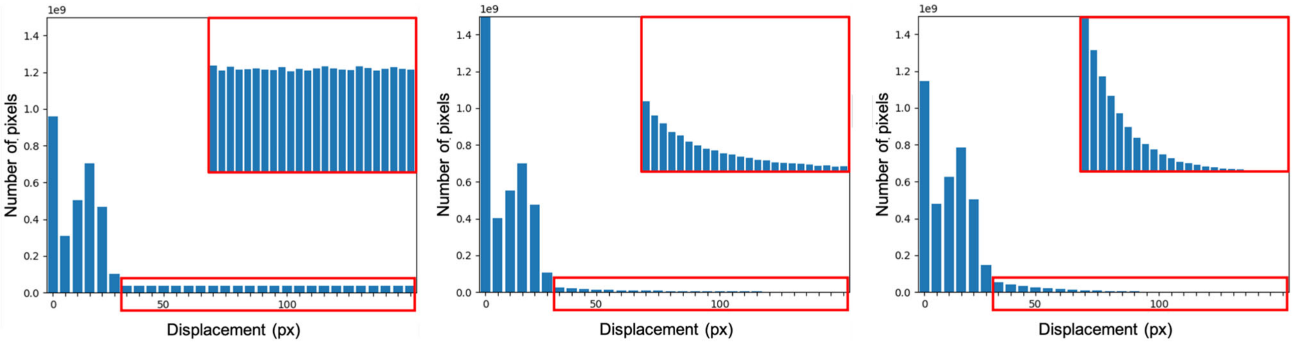
We evaluate the test performance of RAFT and RAFT-ours. Using our synthetic dataset with occlusion masks as an initial learning schedule achieves the higher performance in Sintel and KITTI 2015 test set

Table 5 Generalization results on other backbone networks

	Model	Dataset	Sintel C.	Sintel F.	KITTI12		KITTI15	
			EPE	EPE	EPE	F1	EPE	F1
(A)	FlowNetC	Ch	5.17	6.43	11.82	57.67	20.65	62.91
(B)	FlowNetC	Ours	**4.48**	**6.07**	**10.64**	**52.72**	**18.53**	**55.15**
(C)	PWC-Net	Ch	3.25	4.36	6.27	**27.18**	14.22	40.38
(D)	PWC-Net	Ours	**2.94**	**4.29**	**5.26**	27.28	**10.61**	**38.63**
(E)	RAFT	Ours	1.98	3.85	3.63	20.00	7.17	29.24

We evaluate the generalization performance of the FlowNetC and PWC-Net trained on different datasets: (A, C) FlyingChair, and (B, D) our dataset. (B, D) achieves better performance compared to (A, C). (E) is RAFT trained on our dataset as a reference

	Sintel C.		Sintel F.		KITTI12		KITTI15	
	EPE	≤1	EPE	≤1	EPE	Fl	EPE	Fl
(A)	3.39	0.72	5.68	0.66	8.56	44.05	14.11	48.20
(B)	2.63	0.73	4.33	0.69	6.95	38.21	12.57	43.46
(C)	**2.55**	**0.75**	**4.16**	**0.71**	**5.74**	**35.0**	**10.31**	**41.86**



(A) (B) (C)

Fig. 3 Generalization results and histograms of datasets depending on foreground translation distribution. From left to right, **A** uniform, **B** Gaussian, and **C** exponential distribution. **A** is sampled from a uniform distribution of the interval [0,150]. **B** is the suggested distribution by FlyingChairs [6] given as $\max(\min(\text{sign}(\gamma)\cdot|\gamma|^3, 150), -150)$, where $\gamma \sim \mathcal{N}(0, 2.3^2)$. **C** is the proposed distribution that follows natural statistics [4]. Note that we sample foreground translation magnitude from the three distributions while the background distribution is fixed

4.2 Ablation study

By virtue of the fast generation speed from the simple recipes and the controllability of our dataset, we can conduct a series of ablation studies to determine the critical factors of our dataset which affect the network performance the most.

Foreground Translation Distributions We evaluate the effect of the translational motion distribution of foregrounds with 20K image pairs. We use three different distributions to sample magnitudes of translation. Figure 3 shows the histograms of each dataset distribution and summarizes the generalization results achieved by the RAFT network. (A) is uniform distribution and (B) is Gaussian distribution suggested by FlowNet [6]. (C) is the proposed distribution that follows natural statistics [4].

Table 6 Impact of motion complexity and occlusion masks

	Motion types	Generalization test								Fine-tuning test							
		Sintel C.		Sintel F.		KITTI12		KITTI15		Sintel C.		Sintel F.		KITTI12		KITTI15	
		EPE	≤1	EPE	≤1	EPE	Fl	EPE	Fl	EPE	≤1	EPE	≤1	EPE	Fl	EPE	Fl
(A)	T	2.55	0.75	4.16	0.71	5.74	35.69	10.31	41.86	0.93	0.93	1.45	0.89	1.46	5.09	2.49	8.71
(B)	R	6.84	0.64	7.16	0.59	16.16	66.92	26.07	64.22	1.19	0.93	1.82	0.88	1.91	5.13	2.55	8.45
(C)	Z	3.82	0.78	5.12	0.74	5.66	27.69	13.82	36.91	1.16	0.93	1.69	0.88	1.48	4.72	2.31	8.08
(D)	T+R	2.33	0.81	4.01	0.77	6.09	30.71	12.09	38.82	0.90	0.93	1.40	0.89	1.44	4.82	2.53	8.84
(E)	T+R+Z	**1.98**	**0.86**	3.85	**0.82**	**3.63**	20.00	**7.17**	29.24	**0.85**	**0.94**	1.40	**0.89**	**1.33**	4.37	2.20	8.19
(F)	T+R+Z+O	2.02	**0.86**	**3.67**	**0.82**	3.66	**19.37**	7.88	**28.41**	0.89	0.93	**1.39**	**0.89**	1.35	**4.36**	**2.15**	**7.60**
(G)	Ch	2.28	0.79	4.51	0.72	4.66	30.54	9.85	37.56	0.89	0.93	1.49	**0.89**	1.39	4.69	2.36	8.43

We evaluate the generalization and fine-tuning performance of the RAFT networks trained on our datasets with different motion types and occlusion masks: (A) translation, (B) rotation, (C) zooming, (D∼ E) combinatorial motion type, and (F) applying occlusion masks. We provide (G) the performance of the network trained on Flyingchairs [6] for the comparison

As shown in the histograms, peaks are near zero (in a factor of 10^9) due to the background translation. Thus, we focus on the tails of the distributions, which typically occur by foregrounds. (A) includes excessively large motions, which are unrealistic in real-world scenarios and eventually degrade the performances. Comparing with (B), (C) outperforms on overall metrics of benchmarks. The main difference between these two is the density of the focused region in the histogram, where (C) decays faster than (B). From this, we observe that slight differences in tails of translation distributions affect the performance of the model significantly; thus, we take special care of a balanced motion distribution design. We choose (C) as the distribution of translation for the following experiments.

Motion Complexity We assess the effect of each motion type in training. We start by evaluating the dataset having each of translation ⟨T⟩, rotation ⟨R⟩, and zooming ⟨Z⟩, respectively. Then, we sequentially apply rotation ⟨R⟩ and zooming ⟨Z⟩ to the dataset with the translation ⟨T⟩ only. As shown in Table 6, the network trained on translation motion (A) demonstrates comparable performance to a network trained on FlyingChairs (G). In contrast, with only rotation (B), the generalization performance significantly drops in both benchmarks. When applying zooming alone (C), the performance is sub-par in Sintel, but in KITTI, it exhibits a favorable EPE and surpasses the performance of (A) in the F1 score. This result is likely attributed to KITTI's characteristics, which predominantly feature driving scenes with frequent forward-backward ego motions, which can be mimicked by zooming. We also constitute the dataset (D) by adding rotation transformation to (A) and measure the performance of the trained network. Compared to (A), the network trained on (D) achieves an improved EPE score on Sintel because the cinematic scene of Sintel frequently has rotation motion. In KITTI, the model trained on (D) achieves lower EPE scores. This implies that adding rotation might confuse the network on the test dataset that contains few rotation motions, *i.e.*, the driving scenes of KITTI. Interestingly, both (A) and (D) show comparable performance to the network trained on FlyingChairs (G), which contains three motion types, T+R+Z. We believe that different translation distributions and abundant textures lead to these results. Finally, by adding rotation and zooming (E), the generalization performance outperforms (A∼D), and (G) in all cases. We observe that zooming mimics the backward and forward object or ego motions, which frequently happens in both benchmarks. In summary, translation motion is the most fundamental factor influencing the generalization ability of both benchmarks, while the effects of rotation and zooming vary depending on the characteristics of the test dataset. Although the effects of each type of motion may differ, the combination of translation, rotation, and zooming demonstrates the highest generalization performance on both benchmarks.

The networks trained on our datasets have not seen any 3D motion during training; thus, we can further fine-tune on another dataset, including 3D motions in practice. To figure out the ability of our datasets as pre-training datasets, we further fine-tune the aforementioned networks to the benchmarks, KITTI 2015 or Sintel. We follow the same fine-tuning protocol suggested by Aleotti *et al.* [1] on the KITTI datasets. The same fine-tuning protocol is applied to the Sintel dataset as well, with the initial 80% of the data used for fine-tuning and the remaining portion used for validation. The fine-tuning results in the right part of Table 6 show a consistent tendency with the above generalization study. While the improvement is marginal due to the high accuracy regime, in the KITTI datasets, the best performance is achieved when the pre-trained network has been exposed to diverse types of motion (E), *i.e.*, translation, rotation, and zooming. In Sintel, we observe that the best performance is nearly achieved when exposed to both translation and rotation in the pre-training stage (D). These results suggest that the required motion char-

Table 7 Impact of abundant texture

	Dataset	Number of foregrounds	Blur	Sintel C.		Sintel F.		KITTI12		KITTI15	
				EPE	≤1	EPE	≤1	EPE	Fl	EPE	Fl
(A)	Ours	4	✗	2.29	0.85	3.88	**0.82**	3.95	21.04	8.84	30.96
(B)	Ours	8	✗	**2.17**	**0.86**	**3.69**	**0.82**	**3.57**	**18.86**	**7.34**	**28.09**
(C)	Ours	8	✓	2.25	**0.86**	3.70	**0.82**	3.91	21.87	8.96	31.05
(D)	Ch	–	–	2.28	0.79	4.51	0.72	4.66	30.54	9.85	37.5

We train RAFT with a different number of foregrounds with/without applying gaussian blur. We provide the performance of the network trained on FlyingChairs [6] (D) for the comparison

acteristics of the pre-trained dataset vary depending on the test dataset.

Effects of Occlusion Mask The prior works [18, 19, 34] show the effectiveness of occlusion masks ⟨O⟩. Unlike these prior arts, we propose an intuitive and effective method utilizing the easily obtainable occlusion masks by suppressing the gradients at the regions to be occluded in a supervised manner. In the left part of Table 6, generalization results with occlusion mask (F) show comparable EPE to (E) on the benchmarks but lower Fl on the KITTI datasets. To further evaluate, we fine-tune the network (F) from the left part of Table 6 on the benchmarks and show its results in the right part of the table. The results also show lower Fl on the KITTI dataset. Besides, (F) outperforms (E) on both metrics in fine-tuning on KITTI 2015, which contains the most complicated real-world scenes. In Sintel, applying occlusion masks to pre-training shows a marginal difference in performance compared to the results observed in KITTI 2015. We hypothesize the reason for the marginal performance gap in Sintel as the occlusion patterns between our and Sintel datasets are similar, which are both synthetic datasets. Also, those are easier than the occlusion pattern of KITTI 2015. Thus, the EPE errors in Sintel are much lower. On the other hand, occlusion patterns between our synthetic data and KITTI 2015 data have a clear discrepancy. Thus, learning occlusion during pre-training can produce bias toward our synthetic data set, which has different characteristics with KITTI 2015. This may hint that applying the occlusion mask in pre-training allows to focus on learning the correspondences in the early stage and the specific occlusion patterns of benchmark datasets later in fine-tuning. This phenomenon can be regarded as curriculum learning, where learning more concepts gradually to complex one helps the network perform better. Applying the occlusion mask is an intuitive method for curriculum learning, and we demonstrate the effectiveness of occlusion masks in improving the final performance, particularly when the occlusion pattern varies between the pre-training and fine-tuning datasets.

Abundant Textures We analyze the effect of the abundant textures of foregrounds in training. Considering that the average number of foregrounds in the FlyingChairs [6] is 5, we compared the case when the number of foregrounds is 4 and 8. We also apply a Gaussian filter whose kernel size is 5 to the foregrounds for simulating the lack of high-frequency textures of chairs used in FlyingChairs. Table 7 shows that more foregrounds with high-frequency textures lead to overall improvement. These results hint that abundant textures are another important factor in generating synthetic data.

5 Discussion and limitation

We propose an easily controllable synthetic dataset recipe by cut-and-paste, which enables conducting comprehensive studies. Through the experiments, we reveal the simple yet crucial factors for generating synthetic datasets and learning curriculums. We introduce a supervised occlusion mask method, which stops the gradient at the regions to be occluded. Combining these findings, we observe that the networks trained on our datasets achieve favorable generalization performance, and our datasets with occlusion masks serve as a powerful initial curriculum, which achieves superior performance in fine-tuning and online benchmarks.

Limitation In this work, using the proposed controllable synthetic dataset, we analyzed the effect of key factors in the optical flow training dataset, such as the balanced motion distribution, amount of data, texture combination, and learning schedules with occlusion masks. Although we examined the impact of these fundamental and simple factors through extensive analysis, the impact of various real-world effects, such as motion blur and fog, has not been addressed in this paper, which would also have an impact. Those real-world effects require certain levels of physics simulation or introduce notable complexity in data generation, of which the direction is not aligned with the direction of this work, *i.e.* simplicity. Nonetheless, motion estimation in extreme cases, including weather artifacts or degraded photos, is indeed an important problem and the next challenge. It is an interesting research question, which factors are important to deal with such artifacts, and the complicated and realistic simulation is necessary, which we leave for future research.

Acknowledgements This work was supported by the Korea Research Institute for defense Technology planning & advancement (KRIT) grant funded by the Defense Acquisition Program Administration (DAPA) (No. KRIT-CT-22-037, Hyper-connected space-time information fusion artificial intelligence technologies for signs detection and analysis)

Data Availability The datasets generated during and/or analyzed during the current study are available from the first and corresponding authors on reasonable request. The data generation code will be published through a separate project web-page if accepted.

Declarations

Conflict of interest of potential conflicts of interest: Not applicable.

Compliance with Ethical Standards The authors ensure objectivity and transparency in research and ensure that accepted principles of ethical and professional conduct have been followed.

Research involving Human Participants and/or Animals: Not applicable.

Informed consent: Not applicable.

References

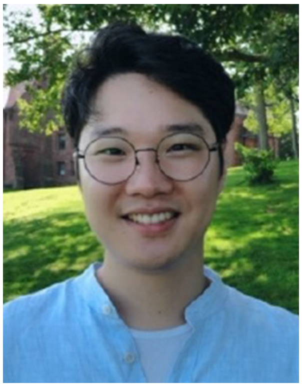
1. Aleotti, F., Poggi, M., Mattoccia, S.: 2021. Learning optical flow from still images, in: IEEE Conference on Computer Vision and Pattern Recognition. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)
2. Black, M.J., Anandan, P.: 1993. A framework for the robust estimation of optical flow, in: 1993 (4th) International Conference on Computer Vision, IEEE. pp. 231–236
3. Butler, D.J., Wulff, J., Stanley, G.B., Black, M.J.: 2012a. A naturalistic open source movie for optical flow evaluation, in: European Conference on Computer Vision (ECCV), Springer. pp. 611–625
4. Butler, D.J., Wulff, J., Stanley, G.B., Black, M.J.: 2012b. A naturalistic open source movie for optical flow evaluation, in: European Conference on Computer Vision (ECCV), Springer. pp. 611–625
5. Byung-Ki, K., Hyeon-Woo, N., Kim, J.Y., Oh, T.H.: Dflow: Learning to synthesize better optical flow datasets via a differentiable pipeline. Presented at the (2022)
6. Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., Van Der Smagt, P., Cremers, D., Brox, T.: 2015. Flownet: Learning optical flow with convolutional networks, in: IEEE International Conference on Computer Vision (ICCV), pp. 2758–2766
7. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (VOC) challenge. Int. J. Comput. Vis. (IJCV) **88**, 303–338 (2010)
8. Gaidon, A., Wang, Q., Cabon, Y., Vig, E.: 2016. Virtual worlds as proxy for multi-object tracking analysis, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR)
9. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: the kitti dataset. Int. J. Robot. Res. (IJRR) **32**, 1231–1237 (2013)
10. Hofinger, M., Bulo, S.R., Porzi, L., Knapitsch, A., Pock, T., Kontschieder, P.: 2020. Improving optical flow on a pyramid level, in: European Conference on Computer Vision, Springer. pp. 770–786
11. Horn, B.K., Schunck, B.G.: Determining optical flow. Artif. intell. **17**, 185–203 (1981)
12. Hui, T.W., Tang, X., Loy, C.C.: 2018. Liteflownet: A lightweight convolutional neural network for optical flow estimation, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 8981–8989
13. Hur, J., Roth, S.: 2019a. Iterative residual refinement for joint optical flow and occlusion estimation, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5754–5763
14. Hur, J., Roth, S.: 2019b. Iterative residual refinement for joint optical flow and occlusion estimation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5754–5763
15. Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., Brox, T.: 2017. Flownet 2.0: Evolution of optical flow estimation with deep networks, in: IEEE International Conference on Computer Vision (ICCV), pp. 1647–1655
16. Janai, J., Guney, F., Wulff, J., Black, M.J., Geiger, A.: 2017. Slow flow: Exploiting high-speed cameras for accurate and diverse optical flow reference data, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3597–3607
17. Jeong, J., Lin, J.M., Porikli, F., Kwak, N.: 2022. Imposing consistency for optical flow estimation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3181–3191
18. Jiang, H., Sun, D., Jampani, V., Yang, M.H., Learned-Miller, E., Kautz, J.: 2018. Super slomo: High quality estimation of multiple intermediate frames for video interpolation, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 9000–9008
19. Jonschkowski, R., Stone, A., Barron, J.T., Gordon, A., Konolige, K., Angelova, A.: 2020. What matters in unsupervised optical flow, in: European Conference on Computer Vision (ECCV)
20. Kondermann, D., Nair, R., Honauer, K., Krispin, K., Andrulis, J., Brock, A., Gussefeld, B., Rahimimoghaddam, M., Hofmann, S., Brenner, C., et al.: 2016. The hci benchmark suite: Stereo and flow ground truth with uncertainties for urban autonomous driving, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 19–28
21. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: 2014. Microsoft coco: Common objects in context, in: European Conference on Computer Vision (ECCV), Springer. pp. 740–755
22. Mayer, N., Ilg, E., Fischer, P., Hazirbas, C., Cremers, D., Dosovitskiy, A., Brox, T.: What makes good synthetic training data for learning disparity and optical flow estimation? Int. J. Comput. Vis. (IJCV) **126**, 942–960 (2018)
23. Mayer, N., Ilg, E., Hausser, P., Fischer, P., Cremers, D., Dosovitskiy, A., Brox, T.: 2016. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4040–4048
24. Menze, M., Geiger, A.: 2015. Object scene flow for autonomous vehicles, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3061–3070
25. Menze, M., Heipke, C., Geiger, A.: 2015. Discrete optimization for optical flow, in: German Conference on Pattern Recognition, Springer. pp. 16–28
26. Oh, T.H., Jaroensri, R., Kim, C., Elgharib, M., Durand, F., Freeman, W.T., Matusik, W.: 2018. Learning-based video motion magnification, in: European Conference on Computer Vision (ECCV), pp. 633–648
27. Roth, S., Black, M.J.: On the spatial statistics of optical flow. Int. J. Comput. Vision (IJCV) **74**, 33–50 (2007)
28. Sun, D., Vlasic, D., Herrmann, C., Jampani, V., Krainin, M., Chang, H., Zabih, R., Freeman, W.T., Liu, C.: 2021. Autoflow: Learning a better training set for optical flow, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 10093–10102
29. Sun, D., Yang, X., Liu, M.Y., Kautz, J.: 2018. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume, in: IEEE

Conference on Computer Vision and Pattern Recognition (CVPR), pp. 8934–8943
30. Sun, D., Yang, X., Liu, M.Y., Kautz, J.: Models matter, so does training: an empirical study of CNNs for optical flow estimation. IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI) **42**, 1408–1423 (2019)
31. Teed, Z., Deng, J.: 2020. Raft: Recurrent all-pairs field transforms for optical flow, in: European Conference on Computer Vision (ECCV), Springer
32. Yang, G., Ramanan, D.: Volumetric correspondence networks for optical flow. Adv. Neural Inform. Process. Syst. (NeurIPS) **5**, 12 (2019)
33. Zach, C., Pock, T., Bischof, H.: 2007. A duality based approach for realtime tv-l 1 optical flow, in: Joint Pattern Recognition Symposium, Springer
34. Zhao, S., Sheng, Y., Dong, Y., Chang, E.I., Xu, Y., et al.: 2020. Maskflownet: Asymmetric feature matching with learnable occlusion mask, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Byung-Ki Kwon received the master's degree in Electrical Engineering from POSTECH, South Korea, in 2023. He is currently working toward his PhD degree at POSTECH. His research interest includes motion estimation, synthetic data generation, and meta-learning.

Sung-Bin Kim received the master's degree in Electrical Engineering from POSTECH, South Korea, in 2023. He is currently working toward his PhD degree at POSTECH. His research interest includes cross-modal generation and multi-modal learning.



Tae-Hyun Oh is an Associate Professor with Electrical Engineering (adjunct with Graduate School of AI and Dept. of Convergence IT Engineering) at POSTECH, South Korea. He received the B.E. degree (First class honors) in Computer Engineering from Kwang-Woon University, South Korea in 2010, and the M.S. and Ph.D. degrees in Electrical Engineering from KAIST, South Korea in 2012 and 2017, respectively. Before joining POST ECH, he was a postdoctoral associate at MIT CSAIL, Cambridge, MA, US, and was with Facebook AI Research, Cambridge, MA, US. He was jointly affiliated with OpenLab, POSCORIST, South Korea, as a research director from 2021 to 2023. He was a research intern at Microsoft Research in 2014 and 2016. He serves as an associate editor for the Visual Computer journal. He was a recipient of Microsoft Research Asia fellowship, Samsung HumanTech thesis gold award, Qualcomm Innovation awards, top research achievement awards from KAIST, and outstanding reviewer awards from CVPR'20 and ICLR'22.