Creative Flow+ Dataset

Maria Shugrina[1,2,3]
www.shumash.com

Ziheng Liang[1,4]
zhliang@cs.ubc.ca

Amlan Kar[1,2]
amlan@cs.toronto.edu

Jiaman Li[1,2]
ljm@cs.toronto.edu

Angad Singh[1,5]
angad.singh@alum.utoronto.ca

Karan Singh[1]
karan@dgp.toronto.edu

Sanja Fidler[1,2,3]
fidler@cs.toronto.edu

[1]University of Toronto [2]Vector Institute [3]NVIDIA
[4] University of British Columbia [5] Evertz Microsystems

Abstract

We present the Creative Flow+ Dataset, the first diverse multi-style artistic video dataset richly labeled with per-pixel optical flow, occlusions, correspondences, segmentation labels, normals, and depth. Our dataset includes 3000 animated sequences rendered using styles randomly selected from 40 textured line styles and 38 shading styles, spanning the range between flat cartoon fill and wildly sketchy shading. Our dataset includes 124K+ train set frames and 10K test set frames rendered at 1500x1500 resolution, far surpassing the largest available optical flow datasets in size. While modern techniques for tasks such as optical flow estimation achieve impressive performance on realistic images and video, today there is no way to gauge their performance on non-photorealistic images. Creative Flow+ poses a new challenge to generalize real-world Computer Vision to messy stylized content. We show that learning-based optical flow methods fail to generalize to this data and struggle to compete with classical approaches, and invite new research in this area. Our dataset and a new optical flow benchmark will be publicly available at: www.cs.toronto.edu/creativeflow/. We further release the complete dataset creation pipeline, allowing the community to generate and stylize their own data on demand.

1. Introduction

For millenia, humans have used drawings, paintings, sketches and diagrams to demonstrate their ideas, plan engineering designs and tell stories. Human vision is impressively robust to abstraction and lack of detail. Without any prior training, a person can easily recognize an object in a

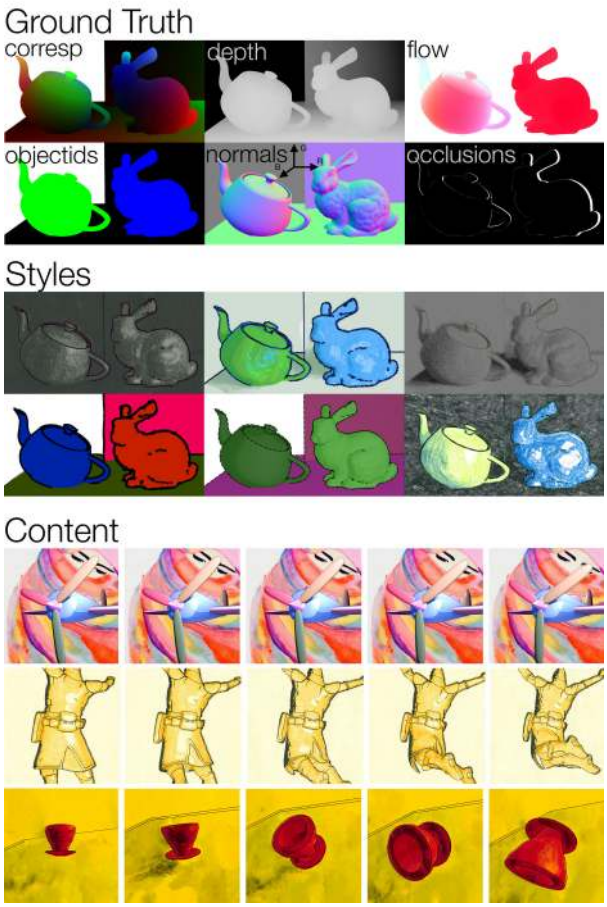


Figure 1: **Creative Flow+ Dataset** contains extensive per-pixel ground truth data for frames rendered in 24 shading styles and 40 line styles and sourced from a variety of 3D animated sequences.[1]

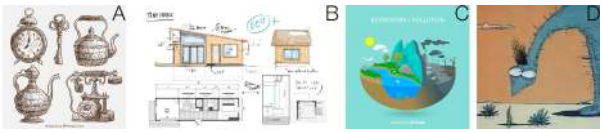


Figure 2: **Beyond Photorealism:** While humans find it easy to visualize 3D shape in sketches (A), find correspondences between different styles/views (B), follow stylized informational videos (C), enjoy hand-drawn cartoons with no temporal coherence (D), such tasks remain challenging for Computer Vision.[2]

rough sketch, visualize its approximate 3D shape, find correspondences between drawings in vastly different styles and views and enjoy motion in hand-drawn cartoons with no temporal coherence (Fig.2). Today's Computer Vision algorithms cannot rival human vision in analysis and understanding of stylized, abstracted content. Yet, their ability to do so could transform the digital creative tools for all domains of design and communication, including education, industrial design, film industry and architecture. For example, correspondences can be used in the animation workflow to autocomplete [47] or interpolate frames [41, 44, 3]. In addition, progress on analysis of stylized content would open up new domains to automatic information retrieval and summarization. The goal of our dataset and benchmark is to enable more research in this area.

In particular, our goal is to enable research on Computer Vision tasks related to motion, correspondences and 3D shape estimation. In the domain of natural photos and video, Computer Vision techniques have made impressive strides in optical flow estimation, segmentation, tracking, correspondence finding and shape estimation from a single image, in part due to availability of large representative benchmarks such as KITTI [22], MPI Sintel [11] and a various RGB-D datasets [17]. However, robustness of these methods to general stylized content is unknown, as none of the existing non-photorealistic datasets include ground truth optical flow or cover a comprehensive set of styles. Likewise, the few correspondence and tracking algorithms specifically tailored to cartoon content [44, 47, 50] have not been evaluated on a breadth of styles and typically make strong assumptions about the input, which makes them brittle in practice. We build a large, diverse dataset to enable research on robust tracking of stylized images.

In this paper we introduce Creative Flow+ Dataset, the largest (124K frames) high-resolution (1500×1500) synthetic optical flow dataset to date, featuring challenging motions, extensive per-pixel ground truth annotations and a diverse set of artistic styles (Fig.1). We use a held out 10K frame test set to show that existing optical flow methods do not generalize well to this challenging content, and will publish a public benchmark posing this new challenge. We give an overview of the data in §3, and detail styles in §4. Comparison with other datasets is provided in §5, followed by optical flow method evaluation in §6.

[1]Image credit: row 5 background is a cropped image by karen sanchez alvarado, sourced from BAM! [45] and licensed under cc by-nc.

[2]Image credits: A,C by Freepik.com, B by Rawpixel.com at Freepik.com, D - frame from "Wings, legs and tails" by Studio Ekran.

2. Related Work

2.1. Existing Motion and Shape Datasets

The core tasks of tracking, optical flow and shape estimation have many established datasets and benchmarks [17, 11, 26]. Optical flow has smaller real world benchmarks such as Middlebury [4] and KITTI [22], as well as larger synthetic datasets that can be used for training, including MPI Sintel [11] and much larger Flying Chairs [15] and Flying Things 3D [31] datasets. The use of synthetic datasets to train Deep Learning (DL) models for performance on the real world has been studied in some detail [30]. However, the question of how well modern Computer Vision methods generalize to understanding of stylized, non-photorealistic content remains unanswered. The human visual system has no trouble adapting from the real world to abstract renditions such as cartoons, but an in-depth investigation of this topic for automatic algorithms has been impossible due to lack of data. Much like existing large optical flow datasets, our dataset is constructed synthetically, but with the opposite goal in mind. Rather than striving for data that leads to better performance on the *real world*, our new optical flow benchmark is designed to systematically test algorithms on stylized, unrealistic content. Further, our train and test sets make it possible to develop approaches that generalize across visual styles.

2.2. Stylized Datasets

Although non-photorealistic content is prevalent in the wild, annotated datasets are limited. The BAM dataset includes 2.5 million images in diverse artistic styles [45], but contains only limited image-level annotations. Photo-Art-50 contains manual labels for 50 classes for a much smaller collection of art and photography [46]. Other datasets, especially those with labels that are richer than image-level categories, are typically confined to a specific drawing style or content domain. For example, several datasets of portrait drawings have been collected [43, 29], some including various levels of abstraction [6], modeling artist's memory of the person [34] or providing multiple caricatures of the same public figure [33]. There are a number of labeled free-hand sketch datasets, including TU-Berlin 20,000 [16], the Sketch Database [36] with photo-sketch pairs, and the fine-grained sketch-based image retrieval dataset of shoes and chairs [49]. Sketches in each of these datasets have a single specific stroke style, limiting their applicability to general sketch understanding in the wild.

The fragility of Deep Neural Networks trained on a specific domain is well known, for example Simo-Sierra et al. observe it for the task of sketch simplification, and instead propose an unsupervised approach [37]. However, unsupervised methods may not be well suited to all tasks. An alternative direction is supervised or unsupervised domain adaptation. Li et al. combine existing stylized datasets into PACS, a domain adaptation benchmark containing 7 categories and 4 domains [28]. Similar datasets with more fine-grained annotations, such as cross-style correspondence pairs or motion information, do not exist.

2.2.1 Synthetic Stylization

Obtaining ground truth annotations can be difficult, and many supervised approaches rely on synthetic data instead. For example, synthetic line drawings have been used to train networks for sketch-based modeling of 3D objects [14, 27] and faces [23]. This approach works well if at test time the trained model responds to sketches drawn in a particular user interface using the same medium, but breaks down if input sketches come from an unconstrained outside domain.

Like most other datasets containing optical flow ground truth, our dataset is created synthetically by rendering 3D scenes. Unlike existing corpora containing both drawings and photos or 3D information [36, 49, 14, 27], we make a specific effort to diversify our dataset across many drawing styles. There are many techniques for non-photorealistic rendering of 3D models, and we refer to Bénard and Hertzmann [5] and related work in [18] for a survey of line drawing, stylization and style transfer techniques. We use Freestyle engine [9] integrated into Blender for outlines, and rely on both Blender and Stylit illumination-guided style transfer by Fišer et al. for artistic shading [18].

3. Creative Flow+ Dataset Overview

To our knowledge, we present the first richly annotated multi-style non-photorealistic *video* dataset, which includes ground truth optical flow and spatial correspondences. In addition, our dataset is the only multi-style artistic *image* dataset that contains per-pixel ground truth labels for normals, depth and object segmentation.

In order to obtain per-pixel ground truth labels (§3.2), we construct our dataset synthetically (§3.4) by configuring animated 3D scenes (§3.1) with a number of stylized rendering styles (§3.3). Our dataset is split into a train and test set, with ground truth from the test set held private for benchmarking (§3.5). Separate sections detail our choice of styles (§4) and dataset statistics (§5).

3.1. Animation Sources

Scenes in open source movies, such as Sintel [8], contain complex custom rendering effects, which require manual handling to ensure proper rendering of ground truth, as detailed by Butler et al. [11], who manually curated the 35 animated scenes in the MPI Sintel dataset. The need to automatically stylize content further complicates the process. Instead, we largely automatically process a much greater number of 2,968 simpler animated sequences:

- 51 animations from [42, 19, 7, 38]
- 1647 character motion sequences, each retargeted to one of 53 characters from Mixamo [2]
- 1270 sequences of unique ShapeNet [12] objects under randomized rigid body simulation

Motion retargeting of Mixamo scenes, ShapeNet rigid body simulation set up and camera set up for both was done automatically. For ShapeNet sequences, a unique object was launched from a random position and hit the randomly tilted floor at a randomized point with varying physical parameters. In 50% of sequences, objects were allowed to break, resulting in complex motion of multiple parts. 50% of ShapeNet sequences include camera tracking and 20% of Mixamo sequences include camera motion. We made a significant effort to ensure that the ground truth rendering is correct for a range of input blends, and the final sequences in our dataset have been filtered to contain reasonable ranges of motion. See §5 for details.

3.2. Ground Truth Information

Each consecutive frame pair (f_0, f_1) in each animated sequence is labeled with the following pixel-level information at a 1500 x 1500 resolution (Fig. 1, ground truth):

- forward and back optical flow
- occlusion map
- object ids
- surface normals
- alpha mask
- depth
- correspondences

Optical flow fields contain per-pixel (u, v) speed vectors of pixels in f_0, and the occlusion map includes pixels in f_0 that are occluded in f_1. Surface normals are rendered as RGB channels relative to the camera, with G corresponding to up in the image plane, R to the right, and B toward the camera; a value of 0.5 corresponds to a zero normal for that component. Object ids are rendered as unique RGB colors with antialiasing turned off, with each color assigned to a unique 3D object in the input animation. In the case of animated characters (§3.1), object ids can also correspond to unique vertex groups, such as shoes or hands. We provide no formal categorization of these object ids, but do include dictionary files with color to object/vertex group mappings. Finally, each object or vertex group is embedded into a bounding box, assigning a unique RGB color to each position on the object using its XYZ position within the bounding box. These colors are rendered into correspondence images. To-

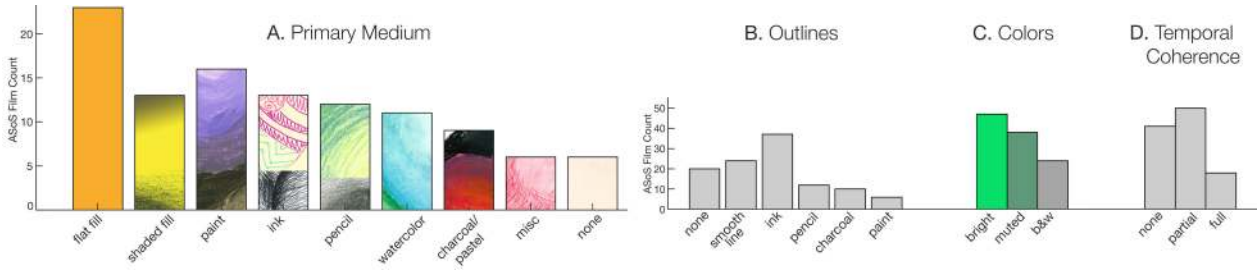


Figure 3: **Styles in the Wild:** breakdown of animated film styles in the Animation Show of Shows (ASoS) series[1].

gether with the object id map, these correspondence images provide a way to find closest corresponding points for any pair of frames (f_i, f_{i+k}) in a sequence. This makes it possible to create training sets for a sparse correspondence task spanning many frames and very large motion without tracking flows across scenes. In addition, it is possible to find the closest corresponding point across frames, even in the presence of occlusions, which would invalidate optical flow tracking.

Undefined areas: Unlike the real world, where every pixel by definition originates from a physical location, stylized imagery may include areas of undefined information. For example, a scene of a character dancing over a flat background provides no information to determine the background flow. The objectid masks in our dataset split objects into three categories: transparent, black for floor/background objects, and color labels for foreground objects. Optical flow and other ground truth is only well-defined for these foreground objects.

3.3. Stylized Frames

Each stylized rendering for a frame f_i includes:

- composited frame, some with background and license
- shading image
- shading alpha channel
- outline image
- outline alpha channel

The final composited frame includes shading, outline and background. In the case of Blender shading styles, where the background is left transparent, we select random images from the BAM dataset [45], which have suitable licensing terms[3]. All of such images require license information to be propagated; and we therefore include license files with all stylized sequences that contain an image background. In addition to the full composited frame we include separate images of shading and outlines, and the alpha components of each. This enables the creation of custom composited datasets. For example, one could use our dataset to create a diverse collection of outlines, including different backgrounds, line colors and textures by using the outline alpha channel. See §4 for style details.

[3] Authors of BAM kindly shared licensing information with us.

3.4. Dataset Construction Pipeline

Our dataset construction pipeline is implemented using Blender 2.79 python API, a variety of command line utilities, and an implementation of Stylit 3D rendering stylization algorithm [18]. The pipeline automatically processes animations in the blend file format. In addition to multiple ground truth passes, each blend is automatically processed to be rendered in one or several stylizations (§4). With the exception of the Stylit [18] implementation, which was kindly provided by the authors, our dataset construction pipeline will be open-sourced upon publication to enable construction of custom datasets.

3.5. Benchmarks

10K frames in our dataset are reserved for testing, with ground truth withheld. We will release a public optical flow benchmark on this test set. In the future, we plan to release other challenges using the sequences in our test set. See §5 for test/train split details.

4. Styles

4.1. Styles in the Wild

Our objective is to make this benchmark applicable to a wide range of visual domains, but the choice of visual styles is not an obvious one. To our knowledge, there is no comprehensive taxonomy of human-generated image styles used in animated content. As a proxy, we have categorized 162 short animated films in the 54 DVDs published by The Animation Show of Shows [1] since 1998. We excluded 37 films using standard 3D rendering (already covered in [11]) and 16 mixed-style films. The remaining 109 films were categorized along 4 axes: A) primary visual medium used, B) type of outlines, C) overall color scheme, and D) frame-to-frame temporal coherence of textures and outlines (Fig.3). Out of 109 films, 70 corresponded to a unique combination of these 4 characteristics. While there is no reason to strive for the specific distribution of styles in [1], we aim to cover a similar diversity.

Figure 4: **Stylit Stylization:** top row - example 2-color style collected from volunteers, bottom row - randomized versions of the style applied to a new rendering (inset) using [18].

4.2. Styles in the Creative Flow+ Dataset

At rendering time, we randomly select a shading and a line style for the composited frame. All styles are split into test and train sets, summarized in Fig.5.

Shading: We have configured our Blender pipeline to allow rendering in flat and cartoon (toon) shading, as well as textured shading which mimics a static paper texture that remains fixed even as the objects move (an effect observed during our analysis). This covers "flat fill" and some (smooth) kinds of "shaded fill" in Fig.3A, but it is clear that a benchmark aimed at general animated content must also cover a range of hand-drawn styles and textures. While there exist Deep Learning stylization techniques pioneered by [21], we were concerned that they may introduce a strong bias into the textures of our stylized data. Instead we opt to use Stylit [18], a more classical illumination-guided style transfer technique that borrows textures directly from provided style example. We have organized a style collection drive, where 11 volunteers used various physical media to create 24 style examples. Each example required drawing a sphere exactly aligned to a 3D rendering (Fig.4, first row), and each style was drawn in either one or two colors. Further, in order to avoid pasting textures from the same image for every frame, every color of every style was drawn twice. Given a new rendering, annotated with normals and object ids, Stylit applies the style exemplar to the new rendering (Fig.4, second row). Applying Stylit to every frame eliminates temporal coherence, much like in real hand-drawn sequences (Fig.3D). We automatically configure Blender to render a red material lit exactly as the sphere for every blend. All styles examples were extensively processed in Photoshop and tested to minimize rendering artifacts.

Outlines: We manually collected a variety of textures, such as ink and pencil, and configured line styles using Blender Freestyle engine [9], covering Fig.3B. While it is difficult for automatic stylization to emulate a range of *expressive* outline styles, such as overdrawing and imprecise strokes, we made an effort to introduce some line modulation and variety of textures to increase diversity.

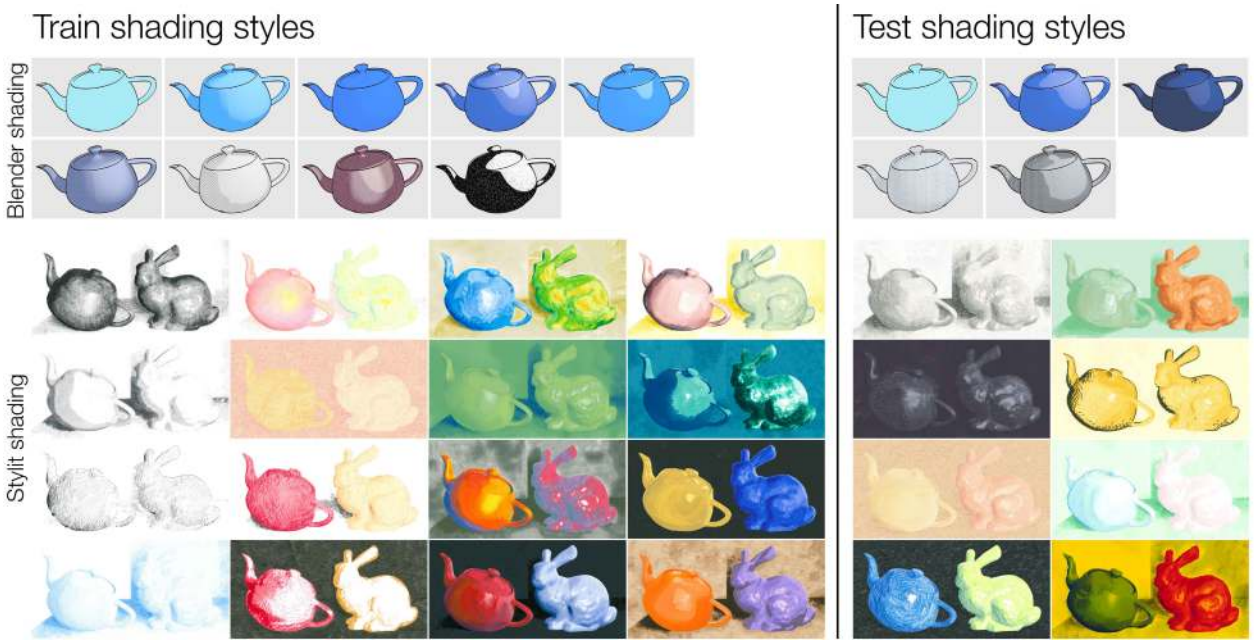
Colors: For flat and toon shading, color of objects is randomized. We found that truly random colors are not representative of stylized content and may make tracking easier by providing more contrast. Therefore, for 20% of the sequences colors are randomly picked from 3570 train or 1500 test sets of discrete color themes collected from [13]. To increase diversity of Stylit styles, we modulate hue, value and saturation of applied styles with a 0.60 probability, and also use this to increase the number of colors (each style example has at most 2 colors). The ranges of allowed modulation were manually determined for each style (e.g. a faint style may become white if its value is set too high). The color of the lines is randomized ensuring that it remains dark with the probability of 0.8, following an observation that most lines in the wild are dark and not random-colored.

5. Dataset Statistics

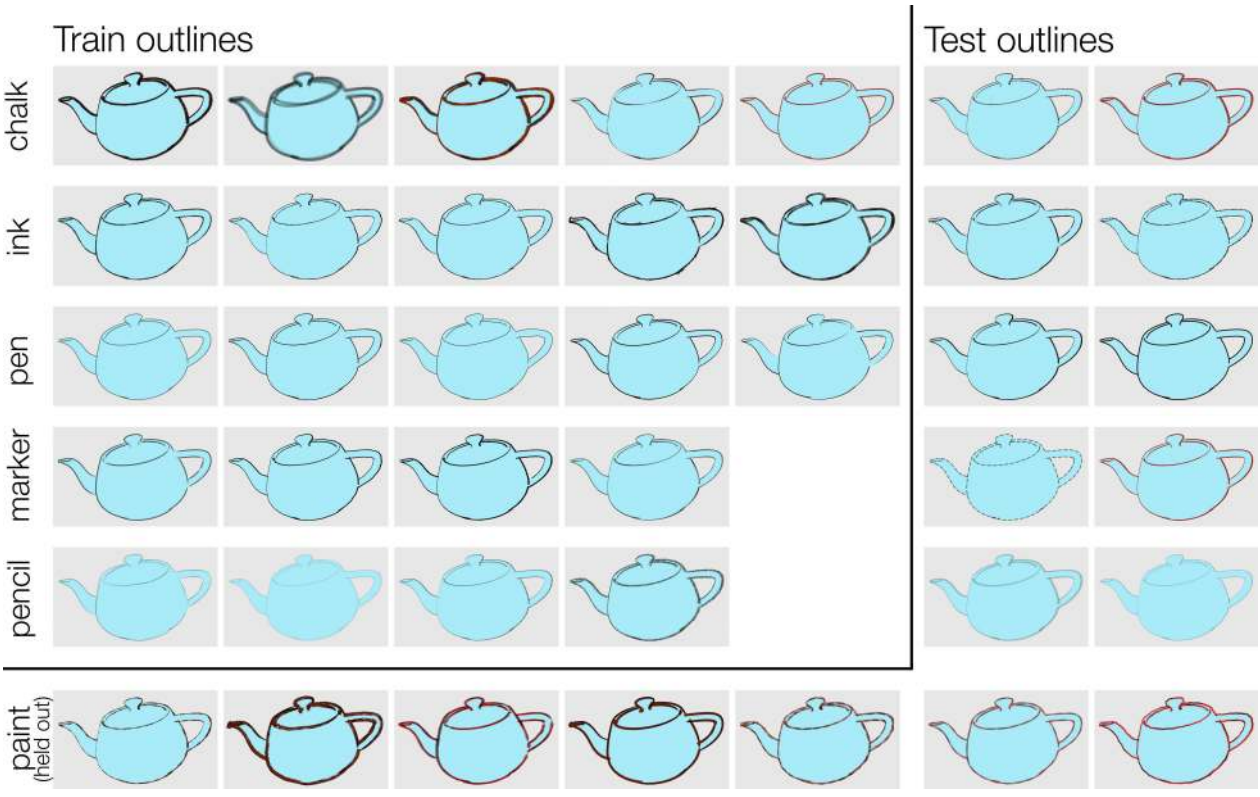
Our **train set** has 2,559 sequences (124,390 frames), consisting of 1,379 Mixamo sequences (82,913 frames), 1,146 ShapeNet sequences (35,570 frames), and 34 Web sequences each at 2 camera angles (5,907 frames). Mixamo sequences were built by retargeting 1,379 unique motions to 38 characters, with each character appearing on average in 36 sequences. ShapeNet sequences were generated from 43 out of 55 ShapeNet classes. Each train sequence is rendered using 2 shading and 2 line styles from train styles in Fig.5, and composited into 2 stylized animated sequences.

Our **test set** has 409 sequences (10,031 frames), consisting of 268 Mixamo sequences (6,559 frames), 124 ShapeNet sequences (2,732 frames), and 17 Web sequences each at 2 camera angles (740 frames). Mixamo sequences were built by retargeting each of 268 held out motions to one of 15 held out characters. ShapeNet sequences were generated using unique objects from 12 held out ShapeNet classes. Each train sequence is rendered using only one shading and one line style from test styles in Fig.5, and composited into a single animated sequence. Backgrounds, when applied, come from a held out subset of BAM [45].

Comparsions: We compare our our dataset to other widely used general purpose optical flow datasets in Fig. 6, omitting datasets tailored to specific real-world scenarios, like Virtual Kitty [20] for driving and SceneNet RGB-D [32] for indoor scene navigation. Refer to [30] for a more inclusive comparison. With the exception of the SceneNet RGB-D dataset, which provides optical flow for 5 million realistic indoor frames at a much lower resolution of 320x240, our dataset far exceeds other existing optical flow datasets in size. We also provide higher resolution images. Both MPI Sintel [11] and Monkaa [30] are based on renderings of 3D movies, similarly to our synthetically rendered data, but do not provide diversity of visual styles. MPI Sintel does include images for 3 different rendering passes, albedo, clean and final, but these cover only a very limited range of stylized imagery. The flow magnitude distribution in our dataset, computed only over the well-defined

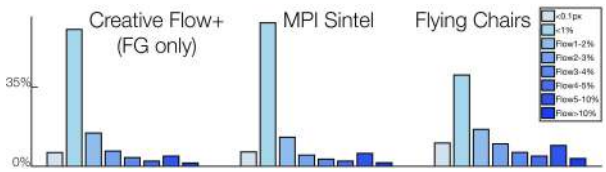


(a) Test and train shading styles using Blender and Stylit rendering.



(b) Test and train outline styles rendered with Blender Freestyle.

Figure 5: **Styles in the Creative Flow+ dataset.**



(a) Flow Magnitude Distributions

Dataset		Styles	Frames	Scenes	Res.
MPI Sintel [8]	test	3	564	12	1024×436
	train	1	1064	23	
FlyingChrs.[15]	train	1	22,872	22,872	960×540
FlyingThgs.[31]	test	1	4,248		960×540
	train	1	21,818	2,247	
Monkaa[31]	train	1	8,591	8	960×540
Creative Flow+	test	13+	10,031	409	1500×1500
	train	25+	124,390	2,559	

(b) Sizes

Figure 6: **Optical Flow Datasets**: comparison of large general-purpose optical flow datasets and Creative Flow+.

foreground areas (See §3.2), is comparable to other datasets (Fig.6a). In part, the large number of frames in our dataset is motivated by the requirement to represent each of the many styles sufficiently to make learning feasible.

Practical matters: Even smaller datasets can present technical challenge due to data size. For example, it may take several days to download the 311GB of optical flow for Flying Things 3D [31]. Because our dataset is even larger, we employ various compression strategies for different types of data. Most image-based sequences (renders, normals, etc) are encoded as videos, for which we will provide decompression utilities. Expensive components will be split into separate downloads. The optical flow for our training set has been compressed to 570GB, and will be provided in split downloads.

6. Evaluation of Flow Methods

We use our 10K test set to gauge the performance of several optical flow methods on stylized content. Our analysis includes classical methods of Horn-Schunck [24] implemented in [39], Classic+NLfast method from [39], and Brox et al. large-displacement optical flow [10]. We also evaluate Epic Flow which combines classical techniques with deep matching [35], and several pre-trained Deep Learning networks, including DC Flow [48], PWC-Net [39] trained on FlyingChairs [15] and on MPI Sintel [11], and LiteFlow Net [25] tuned for Sintel. Refer to Fig.7.

Our motivation for evaluating pre-trained networks on Creative Flow+ is to establish how well these learning methods generalize to new, unseen styles. We found that while average end point error of two out of three classical methods is even lower on our data than on Sintel, all methods that involve a pre-trained network exhibit very high average errors and moderately high median error rates. As this could be due to ill-defined optical flow in background regions (See 3.2), we further break down the errors by evaluating only in the foreground regions (Fig.7a, black regions in the inset are not included in the FG error computation). While the average endpoint error for foreground regions is significantly lower than the overall error rate, it is still far above acceptable levels for modern optical flow methods (e.g. compare to performance on Sintel).

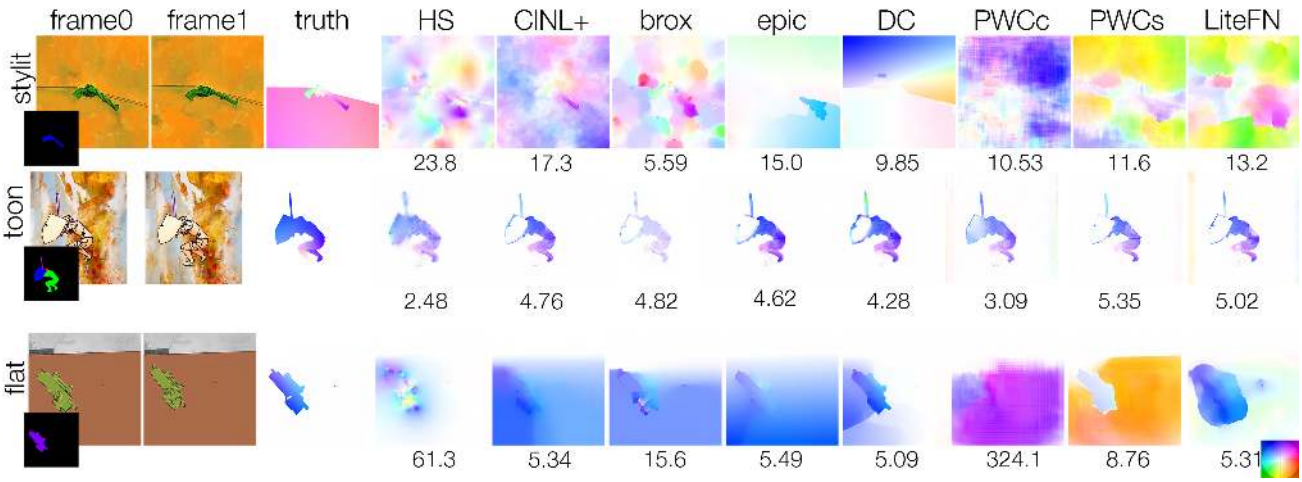
To better understand network generalization across styles, we break down foreground errors by flat, toon, textured and Stylit styles. Across the board, learning approaches perform significantly better on toon shading (Fig.7a, row2), perhaps because it is the closest style to the MPI Sintel dataset, used in training of many of these models. As expected, sketchiest and least coherent Stylit styles (Fig.7a, row1) prove the most difficult. Textured styles, where static texture may detract from the motion of the objects, also challenge both classical and learning-based methods. Apart from general trends, it is apparent that different networks favor different styles even if trained on the same data. For example, PWC-Net performs very poorly on flat shading, whether trained on Sintel or Flying Chairs, but LiteFlowNet, trained on the same data, performs well (See Fig.7a, row3 for flat shading). Performance of these networks on Sintel gives no indication of their generalization to other styles.

We omit background errors from most of our analysis, but the inability of existing methods to deal with noisy backgrounds cannot be ignored. Even examples with relatively low FG errors (Fig.7a, row3) exhibit wild predictions in the background, both for classical and learning-based methods. This would preclude them from being useful in a practice, when foreground annotations are not available. From visual analysis of results, PWC-Net appears to produce the wildest guesses for noisy backgrounds, with the network trained on Flying Chairs favoring upper right direction (purple), and the network trained on Sintel favoring lower right (orange). Further, Stylit backgrounds composed from pasted texture patches can confuse matching algorithms and cause bad Epic Flow and DCFlow predictions (Fig.7a, row1).

To sum up, robustness of existing optical flow methods to stylized content is severely lacking, calling for new research. While further investigation is beyond the scope of our paper, Creative Flow+ dataset opens doors to this research direction.

7. Conclusion

We presented Creative Flow+ dataset, the largest high-resolution optical flow dataset and the first multi-style non-photorealistic dataset richly annotated with ground truth optical flow, depth, normals and more. We showed the need to improve the generalizability of existing optical flow ap-



(a) Qualitative examples, FG errors included.

		Sintel	Creative Flow+									
				median		Styles				Speeds		
			All	All	FG	FG:flat	FG:toon	FG:tex.	FG:stylit	FG:1%	FG:1-3%	FG:3%
Horn-Schunck	[24]	9.64	8.09	3.39	11.93	11.78	10.75	13.67	11.86	3.51	17.19	60.07
Classic+NLfast	[39]	10.12	13.12	6.74	9.11	9.19	6.84	11.23	9.41	5.58	11.06	29.97
Brox2011	[10]	9.15	8.77	3.03	8.17	7.41	6.13	11.50	8.16	4.03	11.19	30.76
EpicFlow	[35]	6.29	63.50	10.00	14.42	9.44	6.66	11.34	22.94	10.82	15.91	36.98
DC Flow	[48]	5.12	40.68	3.15	10.93	7.68	9.02	12.42	12.96	3.93	17.78	44.50
PWC(chrs.)	[40]	-	66.44	40.41	21.98	39.82	10.43	15.74	22.89	22.04	17.74	32.71
PWC(snt.)	[40]	4.60	74.20	33.00	17.57	24.08	6.85	17.07	20.86	16.65	15.08	30.90
LiteFlowNet	[25]	5.06	35.06	12.69	10.94	6.88	6.27	13.52	14.46	8.15	12.55	27.27

(b) Quantitative results.

Figure 7: **Optical Flow Algorithm Performance**: Evaluation on our 10K set. All numbers, except column marked median, are average endpoint errors. Performance on Creative Flow+ broken down into All (full frame) and FG (foreground), as well as by style and speed (1% - ground truth less then 1% of the frame size or 15 pixels, 1-3% between 15 and 45 pixels, 3% over 45 pixels. Style and speed breakdowns are computed only for the foreground regions.

proaches to the stylized domain, and hope that our data will enable much new research in Computer Vision for non-photorealistic content.

Acknowledgments

We gratefully acknowledge support from Vector Institute, and NVIDIA for donation of GPUs. Part of this work was supported by cloud computing resources generously donated by Amazon Web Services (AWS) to the Vector Institute. Sanja Fidler acknowledges the Canada CIFAR AI Chair award at Vector Institute, and Maria Shugrina acknowledges CGS-D NSERC award. Part of this work was also supported by NSERC Discovery grant. We are grateful to the authors of Stylit [18] for sharing an executable, and thank 11 volunteers for donating their time to paint style exemplars. We also thank Kefan Chen and Daiqing Li for their contributions.

References

[1] ACME Filmworks. The animation show of shows. https://www.animationshowofshows.com/, 2018. 4

[2] Adobe Systems. mixamo. https://www.mixamo.com, 2018. 3

[3] Y. Bai, D. M. Kaufman, C. K. Liu, and J. Popovic. Artist-directed dynamics for 2D animation. *ACM Transactions on Graphics*, 35(4):1–10, July 2016. 2

[4] S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. J. Black, and R. Szeliski. A Database and Evaluation Methodology for Optical Flow. *International Journal of Computer Vision*, 92(1):1–31, 2011. 2

[5] P. Bénard and A. Hertzmann. Line drawings from 3d models. *arXiv preprint arXiv:1810.01175*, 2018. 3

[6] I. Berger, A. Shamir, M. Mahler, E. Carter, and J. Hodgins. Style and abstraction in portrait sketching. *ACM Transactions on Graphics (TOG)*, 32(4):55, 2013. 2

[7] Blend Swap LLC. Blend swap. https://www.blendswap.com, 2018. 3

[8] Blender Foundation. Sintel. https://durian.blender.org, 2010. 3, 7

[9] Blender Foundation. FreeStyle rendering engine. https://docs.blender.org/manual/en/latest/render/freestyle/index.html, 2018. 3, 5

[10] T. Brox and J. Malik. Large displacement optical flow: descriptor matching in variational motion estimation. *IEEE transactions on pattern analysis and machine intelligence*, 33(3):500–513, 2011. 7, 8

[11] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In *European Conf. on Computer Vision (ECCV)*, pages 611–625, Oct. 2012. 2, 3, 4, 5, 7

[12] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015. 3

[13] COLOURlovers. COLOURlovers CC. http://www.colourlovers.com, 2018. 5

[14] J. Delanoy, M. Aubry, P. Isola, A. Efros, and A. Bousseau. 3d sketching using multi-view deep volumetric prediction. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 1. 3

[15] A. Dosovitskiy, P. Fischer, E. Ilg, P. Häusser, C. Hazırbaş, V. Golkov, P. v.d. Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. In *IEEE International Conference on Computer Vision (ICCV)*, 2015. 2, 7

[16] M. Eitz, J. Hays, and M. Alexa. How do humans sketch objects? *ACM Trans. Graph. (Proc. SIGGRAPH)*, 31(4):44:1–44:10, 2012. 2

[17] M. Firman. Rgbd datasets: Past, present and future. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 19–31, 2016. 2

[18] J. Fišer, O. Jamriška, M. Lukáč, E. Shechtman, P. Asente, J. Lu, and D. Sỳkora. Stylit: illumination-guided example-based stylization of 3d renderings. *ACM Transactions on Graphics (TOG)*, 35(4):92, 2016. 3, 4, 5, 8

[19] Free3D. Free 3D. https://free3d.com, 2018. 3

[20] A. Gaidon, Q. Wang, Y. Cabon, and E. Vig. Virtual worlds as proxy for multi-object tracking analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4340–4349, 2016. 5

[21] L. A. Gatys, A. S. Ecker, and M. Bethge. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*, 2015. 5

[22] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 2

[23] X. Han, C. Gao, and Y. Yu. Deepsketch2face: a deep learning based sketching system for 3d face and caricature modeling. *ACM Transactions on Graphics (TOG)*, 36(4):126, 2017. 3

[24] B. K. Horn and B. G. Schunck. Determining optical flow. *Artificial intelligence*, 17(1-3):185–203, 1981. 7, 8

[25] T.-W. Hui, X. Tang, and C. C. Loy. Liteflownet: A lightweight convolutional neural network for optical flow estimation. In *CVPR*, pages 8981–8989, 2018. 7, 8

[26] L. Leal-Taixé, A. Milan, I. Reid, S. Roth, and K. Schindler. MOTChallenge 2015: Towards a benchmark for multi-target tracking. *arXiv:1504.01942 [cs]*, Apr. 2015. arXiv: 1504.01942. 2

[27] C. Li, H. PAN, Y. Liu, X. Tong, A. Sheffer, and W. Wang. Robust flow-guided neural prediction for sketch-based freeformsurface modeling. *ACM Transactions on Graphics (TOG)*, 37(5). 3

[28] D. Li, Y. Yang, Y.-Z. Song, and T. Hospedales. Deeper, broader and artier domain generalization. In *International Conference on Computer Vision*, 2017. 3

[29] A. Limpaecher, N. Feltman, A. Treuille, and M. Cohen. Real-time drawing assistance through crowdsourcing. *ACM Transactions on Graphics (TOG)*, 32(4):54, 2013. 2

[30] N. Mayer, E. Ilg, P. Fischer, C. Hazirbas, D. Cremers, A. Dosovitskiy, and T. Brox. What makes good synthetic training data for learning disparity and optical flow estimation? *International Journal of Computer Vision*, pages 1–19, 2018. 2, 5

[31] N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 2, 7

[32] J. McCormac, A. Handa, S. Leutenegger, and A. J.Davison. Scenenet rgb-d: Can 5m synthetic images beat generic imagenet pre-training on indoor segmentation? 2017. 5

[33] A. Mishra, S. N. Rai, A. Mishra, and C. Jawahar. Iiit-cfw: A benchmark database of cartoon faces in the wild. In *European Conference on Computer Vision*, pages 35–47. Springer, 2016. 2

[34] S. Ouyang, T. M. Hospedales, Y.-Z. Song, and X. Li. Forgetmenot: memory-aware forensic facial sketch matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5571–5579, 2016. 2

[35] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid. Epicflow: Edge-preserving interpolation of correspondences for optical flow. In *CVPR*, pages 1164–1172, 2015. 7, 8

[36] P. Sangkloy, N. Burnell, C. Ham, and J. Hays. The sketchy database: Learning to retrieve badly drawn bunnies. *ACM Transactions on Graphics (proceedings of SIGGRAPH)*, 2016. 2, 3

[37] E. Simo-Serra, S. Iizuka, and H. Ishikawa. Mastering sketching: adversarial augmentation for structured prediction. *ACM Transactions on Graphics (TOG)*, 37(1):11, 2018. 3

[38] Sketchfab. Sketchfab. https://sketchfab.com, 2018. 3

[39] D. Sun, S. Roth, and M. J. Black. Secrets of optical flow estimation and their principles. In *CVPR*, pages 2432–2439. IEEE, 2010. 7, 8

[40] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8934–8943, 2018. 8

[41] D. Sýkora, J. Dingliana, and S. Collins. *As-rigid-as-possible image registration for hand-drawn cartoon animations*. ACM, New York, New York, USA, Aug. 2009. 2

[42] Turbo Squid. Turbo squid. https://www.turbosquid.com, 2018. 3

[43] X. Wang and X. Tang. Face photo-sketch synthesis and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(11):1955–1967, 2009. 2

[44] B. Whited, G. Noris, M. Simmons, R. W. Sumner, M. H. Gross, and J. Rossignac. BetweenIT: An Interactive Tool for Tight Inbetweening. *Comput. Graph. Forum ()*, 29(2):605–614, 2010. 2

[45] M. J. Wilber, C. Fang, H. Jin, A. Hertzmann, J. Collomosse, and S. Belongie. Bam! the behance artistic media dataset for recognition beyond photography. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 2, 4, 5

[46] Q. Wu, H. Cai, and P. Hall. Learning graphs to model visual objects across different depictive styles. In *European Conference on Computer Vision*, pages 313–328. Springer, 2014. 2

[47] J. Xing, L.-Y. Wei, T. Shiratori, and K. Yatani. Autocomplete hand-drawn animations. *ACM Trans. Graph.*, 34(6):1–11, 2015. 2

[48] J. Xu, R. Ranftl, and V. Koltun. Accurate Optical Flow via Direct Cost Volume Processing. In *CVPR*, 2017. 7, 8

[49] Q. Yu, F. Liu, Y.-Z. SonG, T. Xiang, T. Hospedales, and C. C. Loy. Sketch me that shoe. In *Computer Vision and Pattern Recognition*, 2016. 2, 3

[50] H. Zhu, X. Liu, T.-T. Wong, and P.-A. Heng. Globally optimal toon tracking. *ACM Transactions on Graphics (TOG)*, 35(4):75, 2016. 2