UNIVERSITY OF AMSTERDAM

UvA-DARE (Digital Academic Repository)

Automatic generation of dense non-rigid optical flow

Lê, H.-Â.; Nimbhorkar, T.; Mensink, T.; Baslamisli, A.S.; Karaoglu, S.; Gevers, T.

DOI
10.1016/j.cviu.2021.103274

Publication date
2021

Document Version
Final published version

Published in
Computer Vision and Image Understanding

License
CC BY

Link to publication

Citation for published version (APA):
Lê, H. .-Â., Nimbhorkar, T., Mensink, T., Baslamisli, A. S., Karaoglu, S., & Gevers, T. (2021). Automatic generation of dense non-rigid optical flow. *Computer Vision and Image Understanding*, *212*, Article 103274. https://doi.org/10.1016/j.cviu.2021.103274

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: https://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, P.O. Box 19185, 1000 GD Amsterdam, The Netherlands. You will be contacted as soon as possible.

UvA-DARE is a service provided by the library of the University of Amsterdam (https://dare.uva.nl)

Download date:05 Sept 2026

Computer Vision and Image Understanding 212 (2021) 103274

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

Automatic generation of dense non-rigid optical flow

Hoàng-Ân Lê [a,*], Tushar Nimbhorkar [b], Thomas Mensink [a,c], Anil S. Baslamisli [a], Sezer Karaoglu [a,b], Theo Gevers [a,b]

[a] Computer Vision Lab, University of Amsterdam, the Netherlands
[b] 3DUniversum, Amsterdam, the Netherlands
[c] Google Research, Amsterdam, the Netherlands

ARTICLE INFO

Communicated by Nikos Paragios

Keywords:
non-rigi
optical flow
dataset
generation
as-rigid-as-possible

ABSTRACT

There hardly exists any large-scale datasets with dense optical flow of non-rigid motion from real-world imagery as of today. The reason lies mainly in the required setup to derive ground truth optical flows: a series of images with known camera poses along its trajectory, and an accurate 3D model from a textured scene. Human annotation is not only too tedious for large databases, it can simply hardly contribute to accurate optical flow. To circumvent the need for manual annotation, we propose a framework to automatically generate optical flow from real-world videos. The method extracts and matches objects from video frames to compute initial constraints, and applies a deformation over the objects of interest to obtain dense optical flow fields. We propose several ways to augment the optical flow variations. Extensive experimental results show that training on our automatically generated optical flow outperforms methods that are trained on rigid synthetic data using FlowNet-S, LiteFlowNet, PWC-Net, and RAFT. Datasets and implementation of our optical flow generation framework are released at https://github.com/lhoangan/arap_flow.

1. Introduction

Optical flow estimation has gained significant progress with the emergence of convolutional neural networks (CNN). The first end-to-end architecture, FlowNet, is proposed by Dosovitskiy et al. (2015) and extended to FlowNet2 by Ilg et al. (2017), which achieves state-of-the-art results. Notable improvements using domain knowledge and classical principles include LiteFlowNet (Hui et al., 2018), PWC-Net (Sun et al., 2018) and RAFT (Teed and Deng, 2020), with 30, 17, and 60 times fewer parameters than FlowNet2, respectively. Additionally, attempts for unsupervised learning are presented, such as Meister et al. (2018) using occlusion-aware bidirectional flow estimation and Liu et al. (2019b) learning by distilling reliable flow estimations from non-occluded pixels. However, they are limited by the ability to model the problem and contribution of component weights of the loss functions (Meister et al., 2018).

With the design of CNNs for optical flow, there is a growing demand for large scale datasets with corresponding dense optical flow fields. However, large-scale datasets with real world imagery and known ground truths in terms of dense optical flow fields simply do not exist. The reason is that dense flow fields are neither measurable with a sensor nor trivial to annotate by humans. Optical flow annotation requires having a full matching of all points in the latent 3D space for each image pair before projecting into the image space. For example, to construct the KITTI datasets (Geiger et al., 2012; Menze and Geiger, 2015), the point clouds from 10 consecutive frames are extracted and registered together and manually checked so that ambiguous or trivially wrong pairs are removed before being projected back to the image space. Such tasks require *accurate* 3D positions and orientations of *all* points in each image as well as a perfect matching algorithm, otherwise the flow fields will be sparse. As a result, while being the largest optical flow dataset available today containing real world images, the KITTI datasets contain only 200 pairs of frames with sparse flow fields, which is insufficient for supervised training of optical flow prediction CNNs.

To resolve the data demand of CNNs, synthetic (generated) data is often used. A well-known synthetic dataset of optical flow is MPI-Sintel (Butler et al., 2012), which uses images and annotations rendered from a computer-generated movie called Sintel. This dataset contains non-rigid optical flow and serves as a well-established basis for comparing CNNs. However, for fully supervised training of CNNs, the number of frames in MPI-Sintel (around 2K) is still insufficient.

Large-scale synthetic datasets are often generated from CAD models, where objects are deformed by affine transformations (zooming, rotation, and translation) and projected on randomly transformed backgrounds. This process is the basis of the FlyingChairs dataset (Dosovitskiy et al., 2015). Due to the large number of available frames, these datasets are useful for training optical flow CNNs. However,

* Corresponding author.
E-mail address: h.a.le@uva.nl (H.-Â. Lê).

https://doi.org/10.1016/j.cviu.2021.103274
Received 1 August 2020; Received in revised form 31 August 2021; Accepted 4 September 2021
Available online 8 September 2021
1077-3142/© 2021 The Authors. Published by Elsevier Inc. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

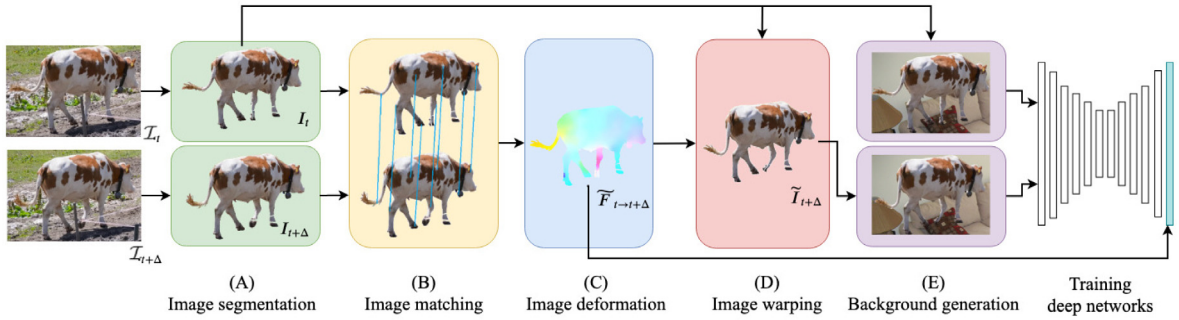


Fig. 1. Overview of the proposed pipeline to generate a dense optical flow field from two video frames. (A) the objects of interest are extracted; (B) motion characteristic is captured by finding correspondences between the objects; (C) object deformation constrained by the correspondences results in a dense flow field; (D) the resulting flow field is used to warp the object; and (E) both the extracted first-frame object and the warped object are pasted on a random background. The resulting pair of frames is used to train a deep neural network with the obtained dense flow field as the ground truth.

deep networks trained on rigid flows with repetitive textures fail to generalize well (Mayer et al., 2018).

Training with non-rigid motion is important for optical flow in real world imagery as many real objects deform in a non-rigid manner. Unfortunately, non-rigid optical flow ground-truth is not available in the current datasets.

In this paper, we present a new approach to automatically generate dense optical flow fields from real-world videos. As illustrated in Fig. 1, our approach collects motion statistics from real-world videos by computing image correspondences between segmented objects-of-interest. Then, the segmented objects are warped to generate complex deformations according to physical principles to generate dense optical flow fields. Our method generates large amounts of optical flow data consisting of natural textures and non-rigid motions, which can be used for training CNNs designed for optical flow estimations.

This paper has the following contributions:

- We introduce the first method to automatically generate dense non-rigid optical flow fields from real-videos;
- We make available a dataset with 55 K frames consisting of natural textures and non-rigid optical flow, created from the DAVIS (Pont-Tuset et al., 2017) video dataset;
- We extensively analyze optical flow network architectures trained using our generated optical flow fields.

The paper is organized as follows. We first review the current datasets being used for training optical flow in Section 2. Then, Section 3 describes our approach to generate a dense flow field from a pair of images. Afterwards, Section 4 explains how the proposed approach is used to generate a dataset and explores the characteristics of the dataset. Finally, Section 5 compares the efficiency of our dataset with alternative approaches for optical flow training.

2. Related work

Most of the benchmark datasets provide optical flow for synthetically generated scenes, including Sintel (Butler et al., 2012), FlyingChairs (Dosovitskiy et al., 2015), and Body Flow (Ranjan et al., 2018). Among them, only the KITTI datasets (Geiger et al., 2012; Menze and Geiger, 2015) provide optical flow for real-world images. However, these datasets are limited to around 200 frames of car-driving scenes and consist mostly of rigid motion patterns.

Dosovitskiy et al. (2015) are the first to generate large-scale optical flow dataset. They warp 2D chair images rendered from CAD models using random affine transformations, hence the name FlyingChairs. The first frame of a pair is created by randomly positioning multiple chair images on an image background. Then, the second frame is generated by warping each object using a flow field generated by the affine model with random parameters. While the parametric model is able to generate many images, the affine transformation yields rigid optical flow fields limiting the type of motion.

SlowFlow (Janai et al., 2017) exploits the linearity of small motions and tracks pixels through densely sampled space–time volumes using high-resolution and high-speed cameras (>1440p resolution and > 200 fps). High spatial resolution provides fine texture details, while high temporal resolution ensures small displacements allowing to integrate strong temporal constraints. However, the requirement of using special recording devices as well as the potential inaccuracies in the *estimated* optical flow limits the applicability of the method.

Data Augmentation. Data augmentation entails a plethora of strategies to create more training data. Widely used techniques for augmenting image data is to perform geometric augmentation (such as translation, rotation, and scaling) and color augmentation (such as changing brightness, contrast, gamma, and color). Data augmentation for optical flow networks is first proposed by Dosovitskiy et al. (2015) and studied in detail by Mayer et al. (2018). The results show that both color and geometry types of augmentation are complementary and improve the performance. Inspired by these data augmentation techniques, we propose methods to increase the diversity of the generated optical flow data by texture augmentations.

In conclusion, large scale datasets with dense optical flow of non-rigid motion from real-world imagery are not available today. This is mainly due to the difficulty of human annotation to generate optical flow ground-truth. Instead, synthetic optical flow datasets with computer-generated imagery are widely preferred. To circumvent human annotation and the use of synthetic imagery data, we propose a framework to automatically generate dense non-rigid optical flow from real-world videos.

3. Generating image pairs for optical flow

In this section, we describe our approach to generate a dense optical flow field from a pair of images, see Fig. 1. The framework is described in the following sections as follows:

3.1 *Image segmentation* receives 2 image frames and extracts the object of interest from them;

3.2 *Image matching* receives the extracted objects and obtains corresponding points between them;

3.3 *Image deformation* computes the flow fields, guided by the correspondences;

3.4 *Warping* of the first object with the flow field to generate a warped object;

3.5 *Random background* on which we paste the first object and the warped object as an input pair for training, with the optical flow field as ground truth.

3.1. Image segmentation

Our aim is to generate flow fields from non-rigid moving objects in videos. From a pair of sequential frames $\mathcal{I}_t$ and $\mathcal{I}_{t+\Delta}$ in a video sequence, where Δ is an offset between the frames ($\Delta = 1$ indicates consecutive frames), the objects of interest I_t and $I_{t+\Delta}$ are localized (see Fig. 1.A).

We compare different ways to localize the objects, including ground truth segments and by using a pre-trained Mask R-CNN (He et al., 2017). Segments can be the entire image frame and do not need to correspond to the objects precisely (see Section 4.3 for more details).

To increase the amount of variations in object motion, different offsets between frames (Δ) are explored (Section 4.1). The localization of objects is also used to replace textures while keeping their shapes (Section 4.2).

3.2. Image matching

The generated flow fields should adhere to the non-rigid motion of the objects in videos. To steer the computation of the flow field, the statistics of the object motion are computed by finding pairs of corresponding points (matches) between the segmented objects I_t, and $I_{t+\Delta}$ (see Fig. 1.B).

As the paper aims to generate datasets for training deep networks, the data-agnostic DeepMatching method by Weinzaepfel et al. (2013) is used to obtain the motion statistics between image frames. The method is inspired by the hierarchical, multi-layer and correlational structure of deep convolutional networks, but it does not require training. The matching is performed in 2 stages: bottom-up correlation pyramid computation and top-down corresponding extraction. Analyses show that DeepMatching is robust to non-rigid deformations and repetitive textures (due to multi-scale correlation). The method is efficient in determining dense correspondences from strong image changes.

The algorithm obtains a set of point-to-point matches. We denote with $\mathcal{M}\left(\mathbf{x}_t^k\right) = \mathbf{x}_{t+\Delta}^k$ the map of the pixel coordinates of the kth pixel in I_t to the corresponding pixel coordinate $\mathbf{x}_{t+\Delta}^k$ in $I_{t+\Delta}$. The obtained correspondences are quasi-dense and robust to non-rigid deformations and repetitive textures.

3.3. Image deformation

To generate a dense flow-field, we deform the segmented object I_t to match with $I_{t+\Delta}$, using the obtained image matches $\mathcal{M}$ to guide the deformation process (see Fig. 1.C). The *as-rigid-as-possible* (ARAP) (Alexa et al., 2000; Wang et al., 2008; Dvorožňák, 2014; DeVito et al., 2017) principle is used to deform the objects. ARAP allows for large non-rigid deformations but also conforming to physical feasibility by minimizing scaling and shearing factors of the local image regions.

The deformation method is illustrated in Fig. 2. We define a rectangular grid tightly bounding the object I_t, where each vertex corresponds to a pixel. This grid deformation (see Fig. 2(b) to (c)) is steered by image matches $\mathcal{M}$ and regularized by local deformations. The dense flow field $\tilde{F}_{t\to t+\Delta}$ is obtained by interpolating the grid vertices before and after the deformation process.

Mathematically, the image deformation process is formulated as an energy optimization problem over the grid structure. We minimize per image the energy of a data fitting term weighed with a regularizer:

$$E(\mathbf{d}, \mathbf{R}, \mathbf{x}_t, \mathcal{M}) = \sum_k w_{\text{fit}}\ E_{\text{fit}}(\mathbf{d}^k, \mathbf{x}_t^k, \mathcal{M}) + w_{\text{reg}}\ E_{\text{reg}}(\mathbf{d}^k, \mathbf{x}_t^k, \mathbf{R}^k), \tag{1}$$

where $\mathbf{d}$ denotes the deformed grid fitted to object $I_{t+\Delta}$. Following Dvorožňák (2014), we set $w_{\text{fit}} = 10$ and $w_{\text{reg}} = 0.1$. The data fit term is guided by the matches $\mathcal{M}$:

$$E_{\text{fit}}(\mathbf{d}^k, \mathbf{x}_t^k, \mathcal{M}) = \left\| \mathbf{d}^k - \mathcal{M}(\mathbf{x}_t^k) \right\|^2. \tag{2}$$

For pixel coordinates without an image match, the relationship $\mathcal{M}(\mathbf{x}_t^k) = \mathbf{x}_t^k$ is used instead.

As regularizer, the relative rigid rotation between neighboring pixels with an image wide rotation matrix $\mathbf{R}^{kj} \in \mathbb{R}^{2\times 2}$ is used. This has been shown to enforce rigid rotation and translation (Wang et al., 2008). This yields:

$$E_{\text{reg}}(\mathbf{d}^k, \mathbf{x}_t^k, \mathbf{R}^k) = \frac{1}{4} \sum_{j=1}^{4} \left\| \mathbf{R}^{kj} \left(\mathbf{x}_t^{kj} - \mathbf{x}_t^k \right) - \left(\mathbf{d}^{kj} - \mathbf{d}^k \right) \right\|^2, \tag{3}$$

where $\mathbf{x}_t^{kj}$ denotes the jth neighbor from pixel k and $\mathbf{d}^{kj}$ the coordinates after deformation of $\mathbf{x}_t^{kj}$. The four connected pixels to each pixel are used as neighbors.

Eq. (1) is minimized with respect to $\mathbf{d}$ and $\mathbf{R}$, resulting in a non-linear least square problem, which is solved by the iterative Gauss–Newton method (DeVito et al., 2017).

3.4. Image warping

The dense optical flow field $\tilde{F}_{t\to t+\Delta}$ is obtained by bilinear interpolating $\mathbf{x}_t$ and $\mathbf{d}$. Due to possible errors introduced by the matching and deformation algorithm, it is only an approximation of the true field $F_{t\to t+\Delta}$, and hence it does not necessarily transform the object I_t to the exact shape of $I_{t+\Delta}$.

To generate correct triples, image warping $\tilde{I}_{t+\Delta} = \mathcal{W}(I_t, \tilde{F}_{t\to t+\Delta})$ is used, see Fig. 1.D. This results in a correctly generated triple $(I_t, \tilde{I}_{t+\Delta}, \tilde{F}_{t\to t+\Delta})$[1]. In Fig. 2, segmented objects I_t and $I_{t+\Delta}$ are illustrated with the obtained optical flow $\tilde{F}_{t\to t+\Delta}$ and the warped object $\tilde{I}_{t+\Delta}$, including some of the matching errors discovered by the warping process.

3.5. Background generation

To obtain a full frame image pair, the object I_t and the warped object $\tilde{I}_{t+\Delta}$ are projected on a (static) background image (Fig. 1.E). This background image is randomly sampled from a set of 8 K Public Domain images from Flickr of general scenery, similar to the approach used for the creation of the FlyingChairs dataset (Dosovitskiy et al., 2015). However, in contrast to the FlyingChairs dataset, static backgrounds are used instead of affine transformed backgrounds.

4. Generating the DAVIS-mask-OpticalFlow dataset

In this section, datasets generated from video frames using the proposed method are explored. See the pseudo-code in Algorithm 1. The algorithm takes a video dataset $\mathcal{V}$ as input, together with an integer number Δ for the frame distance, a segmentation method S, and a texture replacing method T. CNNs learn a better model when the training set consists of samples with a large variety of textures, motion patterns, and displacements (Mayer et al., 2018). Hence, different choices for Δ, S, and T to create optical flow datasets are investigated.

The DAVIS (Pont-Tuset et al., 2017) video dataset is used to generate optical flow datasets. DAVIS contains 6 K frames of real imagery with provided segmentation masks. The generated optical flow datasets are used to train a FlowNet-S (FNS) model (Dosovitskiy et al., 2015). Evaluation is performed on a subset of 410 image pairs from the training set of MPI-Sintel (Butler et al., 2012), coined Sintel-val. Results are reported using the average *end-point-error* metric (AEPE, lower is better). The obtained results are compared to a FlowNet-S model trained on the FlyingChairs (Dosovitskiy et al., 2015) dataset as baseline. This dataset has 22 K image pairs of chairs projected on different backgrounds with corresponding optical flow ground truths. FlowNet-S is chosen since it is fast to train, thus suitable for extensive experimentation. In Section 5, experiments are conducted with our final dataset using more recent architectures, and a more diverse range of datasets.

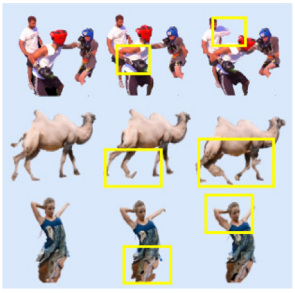
[1] Image warping might introduce artifacts due to interpolation used, it is however commonly used, *e.g.*in the FlyingChairs dataset (Dosovitskiy et al., 2015)

Fig. 2. Illustration of ARAP image deformation. Left: deformation process includes constructing a control lattice (a), deforming the control lattice steered by the image matches (b,c), obtaining the flow field by interpolating the deformed lattice (d). Right: examples of image segments I_t and $I_{t+\Delta}$ annotated with point matches, the computed flow $\tilde{F}_{t\to t+\Delta}$ and the warped images $\tilde{I}_{t+\Delta}$. Note the significant differences between $I_{t+\Delta}$ and $\tilde{I}_{t+\Delta}$ (bottom row) due to the errors in the point matches. However $(I_t, \tilde{I}_{t+\Delta}, \tilde{F}_{t\to t+\Delta})$ form a correct triplet.

Algorithm 1 Generate optical flow from a video dataset $\mathcal{V}$.

Input: a video $\mathcal{V}$, frame-distance Δ, segmentation method S, texture method T
Output: optical flow dataset $\mathcal{F}$

for $t \in [0, \text{length}(\mathcal{V}) - \Delta]$ **do**
  $\mathcal{I}_t, \mathcal{I}_{t+\Delta} \leftarrow$ sampled from $\mathcal{V}$ # Section 4.1
  $I_t, I_{t+\Delta} \leftarrow$ segmenting $\mathcal{I}_t$ and $\mathcal{I}_{t+\Delta}$ using S # Section 4.3
  $\mathcal{M} \leftarrow$ image_matching$(I_t, I_{t+\Delta})$
  if $\mathcal{M}$ **is** $\varnothing$: **skip frame**
  $\tilde{F}_{t\to t+\Delta} \leftarrow$ ARAP deformation$(\mathcal{M}, I_t)$
  $I_t \leftarrow$ replace texture of object I_t using T # Section 4.2
  $\tilde{I}_{t+\Delta} \leftarrow$ image warping $\mathcal{W}(I_t, \tilde{F}_{t\to t+\Delta})$
  $\mathcal{I}_t, \tilde{\mathcal{I}}_{t+\Delta} \leftarrow$ pasting I_t and $\tilde{I}_{t+\Delta}$ on a random background
  $\mathcal{F} \leftarrow \mathcal{F} + \{(\mathcal{I}_t, \tilde{\mathcal{I}}_{t+\Delta}, \tilde{F}_{t\to t+\Delta})\}$



(a) Warped images

	Sintel-val
FC	5.75
Δ1	5.50
Δ1-2	5.37
Δ1-3	5.17
Δ1-4	5.10
Δ1-5	5.06
Δ1-5,8	**5.03**
Δ1-5,8,12	5.06

(b) AEPE

Fig. 3. Influence of the frame distance (Δ): Increasing the frame distance increases the motion magnitude (a), and introduces artifacts in the warped objects (b), yet increasing the dataset by adding frame distances up to $\Delta = 5$ is beneficial for performance(c). Larger frame distances have neglectable influence.

4.1. Displacement variation

To increase the variation in object motion, different frame distances Δ in the video sequence are used. Larger Δ reflects motion further in time, resulting in more variation in the non-rigid motion statistics of the objects, as shown in Fig. 3(a). Note, however, that larger frame distances also introduce artifacts, likely due to matching errors.

In order to study the influence of creating a dataset with larger frame distances, datasets generated with $\Delta = \{1, 2, \ldots, 12\}$ are combined into a single dataset. Despite the increase of the training data size, the images' appearances basically stay the same as they are extracted from the same set of 6 K videos. Hence, the performance gain is attributed to the increased displacement.

(a) Appearance variation

	T	Δ5 SynR	Δ5 ReaR	Δ5 SINv	Sintel val
FC		7.29	6.92	6.28	5.10
Δ1	O	4.82	4.62	4.20	5.50
Δ1-4	O	4.75	4.42	4.02	5.10
Δ1-4	R	4.61	4.32	3.99	**4.96**
Δ1-4	C	**4.60**	**4.24**	**3.86**	4.98

(b) Performance

Fig. 4. Re-textured objects: (a) examples of different textures, with left the original, and (b) performance analysis for different textures. We conclude that training with re-textured objects (both R and C) ensures better performance.

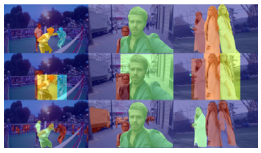
The performance is presented in the table of Fig. 3(b). From these results, it can be derived that increasing the frame distance is, in general, beneficial for AEPE error on Sintel-val. We observe some diminishing gains, especially for $\Delta > 5$. This is attributed to the introduced artifacts in the warped images. More importantly, the results show that there is *no* need to strictly match the distributions of the training and testing sets to achieve the best performance (Mayer et al., 2018). Although Δ1-5,8 has slightly better performance, given that the large frame-distance version also produces more artifacts, this version is discouraged. The smaller set is also more favorable because of time efficiency. Thus, for the remaining experiments Δ1-5 is used to generate optical flow, unless stated otherwise.

4.2. Texture variation

Object re-texturing allows for increasing the variation in the datasets appearances, see Fig. 4(a). Moreover, re-textured objects enforce the network to disentangle semantic (class specific) information from optical flow information. This is likely beneficial for a generic (class agnostic) optical flow prediction model.

To re-texture objects, denoted by T in Algorithm 1, the following procedure is followed. After obtaining the flow field, the original texture (O) of I_t is replaced by a new random texture (R), using the segmentation mask. The random texture is taken from the set of general natural images utilized as background scenes. Then, the corresponding $\tilde{I}_{t+\Delta}$ is obtained by warping the re-textured I_t, using $\tilde{F}_{t\to t+\Delta}$. We explore using a re-textured dataset, denoted by R, and using a combination of original textures with random re-textures, denoted by C.

FlowNet-S is trained on the newly re-textured data (Δ1-4) and its robustness for unseen textures and displacements is analyzed. The models are evaluated on Δ5 and Sintel-val. The former has been re-textured with 3 texture types: *SynR*, synthetic images with repetitive patterns; *ReaR*, real images with repetitive patterns; and *SINv*, images from the Sintel-val set, see examples of the re-textured images in Fig. 4(a).



(a) Segment variations

Segment type	Sintel-val
Full frame (F)	5.69
Ground truth (G)	5.06
Box (B)	5.16
M-RCNN (M)	**4.91**
SipMask (S)	5.09

(b) Performance

Fig. 5. Object Segmentation. (a) From top to bottom: examples of ground-truth segments (G), bounding boxes (B) and Mask R-CNN predictions (M); and (b) Sintel-val performance. The approximately correct segments M yield best performance. (Δ1-5-O).

The results are presented in the table of Fig. 4(b). We conclude that training with re-textured data (R or C) is beneficial for good performance on both Δ5 and Sintel-val. The performance differences between the different re-textured datasets used in Δ5 show the dependency of the performance on the test images' texture. This confirms the hypothesis that the network needs to be trained with a wide variety of texture types. Hence, for the subsequent experiments, a combination of original-texture and re-textured images (C) is used.

4.3. Object segmentation

In this section, different methods for selecting the object of interest are compared. So far, the ground truth segmentation masks have been utilized. Now, the following alternatives are considered: (1) selecting the entire frame as the object of interest; (2) using tight bounding boxes enclosing the ground-truth segments; (3) using the ground-truth segments; and (4) using segments from Mask R-CNN (He et al., 2017) and SipMask (Cao et al., 2020), pre-trained off-the-shelf segmentation networks. See Fig. 5(a) for examples.

Entire-frame deformations include constraints from both backgrounds and foreground objects, which might limit the flexibility and variation in the generated deformation. The bounding boxes increase the segment sizes by including background parts while keeping the objects of interest in focus. For Mask R-CNN (He et al., 2017) and SipMask (Cao et al., 2020), the available pre-trained models are used, which is trained on the class labels of the MS-COCO (Lin et al., 2014) datasets. Due to uncertainties in inference, the Mask R-CNN segments may generate larger regions rather than strictly focusing on the centered objects like the ground-truth segments. This might result in creating a large variation in terms of object shapes and sizes. SipMask model is more accurate and can generate segments closer to the ground truth, thus reducing the variability of background segments which are shown to be useful and might reduce the network performance. The performance on Sintel-val is indeed close to that of using GT segments.

The results of training FlowNet-S on the data generated using the original textures (E1-5-O), comparing different segmentation methods, are provided in the table of Fig. 5(b). It can be concluded that focusing on objects is beneficial with the performance being $F < B < G$. Surprisingly, the network trained with the dataset using Mask R-CNN segments outperforms the ones using ground truth segments and SipMask ($M > G > S$). This is because Mask R-CNN segments, in general, have more variation and cover more object types in a scene compared against the ground truth and SipMask segments: not only the objects of interest, but also those in the background. Hence, it provides the network with a larger range of patterns, which appears to be useful for training. This indicates that it is possible to use any real-world in-the-wild videos with Mask R-CNN segments for training optical flow deep networks. In the subsequent experiments, datasets using Mask R-CNN (M) are utilized.

4.4. Non-rigid motion analysis

The Sintel movie is created using mostly static scenes (backgrounds) and moving characters (objects). In this section, the performance of the FNS models on non-rigid movements and occluded regions is evaluated. Evaluation is performed over (a) foreground objects using the segments provided by (Butler et al., 2012) and (b) occluded regions, defined as those regions which appear in one of the two input frames only.

Table 1
Non-Rigid Motion Analysis: Performance on subsets of Sintel-val using the full image (F), non-rigid motion (N-R) or occluded regions (Occ), for three different architectures: FlowNet-S (FNS), PWC-Net (PWC), and LiteFlowNet (LFN). Deep networks trained on our non-rigid motion datasets outperform those trained on the FlyingChairs dataset.

	FNS			LFN			PWC		
	Full	N-R	Occ	Full	N-R	Occ	Full	N-R	Occ
FC	5.09	14.56	12.32	4.32	14.70	12.86	4.01	13.52	11.27
Δ1-5-O-M	4.96	13.88	12.64	4.34	**13.59**	12.88	3.88	12.59	11.21
Δ1-5-C-M	**4.54**	**13.28**	**11.82**	**4.23**	13.83	**12.79**	**3.62**	**11.90**	**10.57**

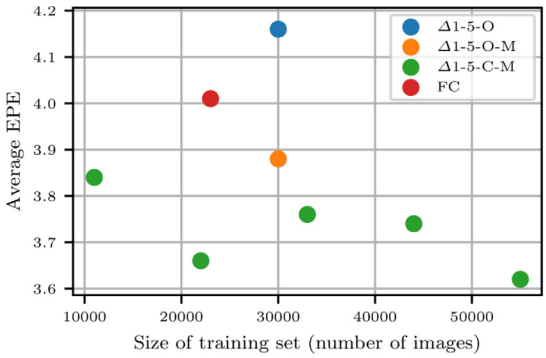


Fig. 6. Size of Training Set: Performance on Sintel-val of PWC-Net (Sun et al., 2018) trained on differently sized datasets. Results of sub-sampled Δ1-5-C-M are indicated in green. PWC-Net trained on Δ1-5-C-M outperform the others, regardless of training size.

Different training sets are compared (FC, Δ1-5-O-M, and Δ1-5-R-M) using FlowNet-S (Dosovitskiy et al., 2015), PWC-Net (Sun et al., 2018), and LiteFlowNet (Hui et al., 2018). The results are provided in Table 1. We conclude that training on our non-rigid motion datasets outperforms training on the rigid transformations from the FlyingChairs dataset. This holds especially for non-rigid and occluded regions in the images. All in all, non-rigid motion is a necessity to train robust CNNs for optical flow prediction.

4.5. Training dataset size

In this section, the performance is evaluated as a function of number of images in the training dataset. The aim is to distinguish improvements based on the sheer number of annotated training examples from the improvements based on non-rigid motion and texture variations. Since the FlyingChairs dataset contains ~22 K images, the 55 K images from our Δ1-5-C-M dataset are sub-sampled to generate training sets with 11 K, 22 K, …, 55 K examples.

PWC-Net is trained on these different datasets and evaluated on Sintel-val. The results are provided in Fig. 6. It can be observed that the training set size does have an influence on the performance. The results show that Δ1-5-C-M gradually improves as the size of datasets increases (with 22 K being the outlier likely due to sub-sampling effects). However, the results also show that regardless of the size of this dataset, training on Δ1-5-C-M performs the best.

4.6. Discussion

From these extensive yet initial analyses of various design choices, we derive that the generated datasets with non-rigid optical flow fields are well suited for training CNNs for optical flow prediction. In the next section, the Δ1-5-C-M dataset, coined Deepmatching-Maskrcnn-OpticalFlow (**DMO**), generated using Mask R-CNN segments using original and re-textured objects is utilized.

Table 2
Performance of RAFT being trained on datasets generated with different matching and segmentation algorithms. The best results of the entire column are shown in boldface while those among ARAP-based methods are underlined. The dataset generated by DeepMatching and Mask-RCNN, i.e. DMO, shows balancing performance between final and clean pass.

Training dataset	Sintel-val	Sintel-test			
		final	clean	final	clean
FlyingChairs		4.25	**2.91**	7.69	**4.48**
DM	M-RCNN	3.98	3.12	**6.15**	4.86
DM	SipMask	3.91	3.27	6.20	4.88
NCNet	M-RCNN	3.92	3.30	6.44	5.48
NCNet	SipMask	**3.81**	3.45	6.57	6.55

5. Comparison using state-of-the-art architectures

In this section, experiments are conducted to compare models trained on the proposed DMO dataset to various state-of-the-art (SOTA) baselines and benchmarks.

5.1. Experimental setup

Datasets For most of the experiments, models trained on the FlyingChairs dataset (Dosovitskiy et al., 2015) are used to provide baseline comparisons. This allows to analyze the effect of the training set on the performance of different network architectures.

Evaluation is performed on the test-set of MPI-Sintel (Butler et al., 2012) containing large displacements of non-rigid optical flow. Additional evaluations are performed on the validation split of the HumanFlow (Ranjan et al., 2018), which contains 530 image pairs of non-rigid motion of human bodies, and on a subset of 50 randomly selected images from the KITTI 2012 (Geiger et al., 2012) and KITTI 2015 (Menze and Geiger, 2015) training set containing real-textures from a self driving car. Both MPI-Sintel and HumanFlow contain CGI-rendered images, and KITTI contains mostly rigid motion. The quantitative results on real-world non-rigid images can be found in the supplementary materials. Additionally, the QUVA repetition dataset (Runia et al., 2018) is used to qualitatively evaluate on real videos with non-rigid motion. It consists of video sequences of repetitive activities recorded in real life with minor camera motion, mostly consisting of non-rigid object motions.

Network Architectures Different CNN architectures are compared, namely: (i) FlowNet-S (Dosovitskiy et al., 2015) which is also used in Section 4, (ii) LiteFlowNet (Hui et al., 2018), PWC-Net (Sun et al., 2018), and RAFT (Teed and Deng, 2020) as recent supervised models, (iii) three unsupervised architectures, specifically MFOF (Janai et al., 2018), DDFlow (Liu et al., 2019a), and SelFlow (Liu et al., 2019b). For each model, the standard training settings are used, including data augmentation and learning schemes, as provided by the model's authors.

5.2. Performance with SOTA matching and segmentation

In this experiment, the datasets generated using different matching and segmentation methods are compared by training the state-of-the-art flow prediction network RAFT (Teed and Deng, 2020). In particular, we compare the use of DeepMatching (DM) (Weinzaepfel et al., 2013) in the previous experiments with NCNet (Rocco et al., 2018) for matching and Mask-RCNN (He et al., 2017) with SipMask (Cao et al., 2020) for segmentation. The results are presented in Table 2, where the combination (DM, M-RCNN) corresponds to the DMO dataset. The performance evaluations from training with the FlyingChairs dataset are also included.

RAFT being trained on the FlyingChairs dataset achieves superior performance on the clean pass of the Sintel dataset (both Sintel validation and test set), while being inferior by a large margin on the final pass. This is somewhat surprising as the final pass is considered more difficult because of the severe environmental and motion artifacts (Butler et al., 2012). From the training data viewpoint, it could be explained that the networks trained with ARAP-based datasets are more robust to artifacts as the training data contain artifacts from the matching and deformation processes (Fig. 2), while the FlyingChairs images are all clean-cut (due to well-defined rigid transformation).

Among those using the same matching method, datasets using Mask-RCNN help the network produce slightly higher performance. This is consistent with the results of the table in Fig. 5(b) which shows that the high accuracy of SipMask segmentation focusing mostly on image central objects (similar to ground truth segmentation) are undesired for optical flow datasets. On the other hand, among those using the same segmentation, the networks trained with datasets using DeepMatching perform better. This is because NCNet is prone to repetitive patterns, especially when there are large changes in scale and locally geometrically consistent groups of incorrect matches (Rocco et al., 2018), while DeepMatching is designed to be robust to repetitive patterns (Weinzaepfel et al., 2013). Considering the performance balance in both final and clean pass, we confirm the usage of the DMO dataset, i.e. generated by DeepMatching and Mask-RCNN, in the following experiments, unless stated otherwise.

Table 3
Comparison of different models (FNS, PWC, LFN) trained on FC and DMO, evaluated on Sintel benchmark, Human Flow, and KITTI. The constant zero flow indicates the displacement statistics of the test sets. The results of DeepFlow is reported by Revaud et al. (2015) and DeepFlow* by Weinzaepfel et al. (2013). For all networks and all evaluations hold that training on non-rigid motion data (DMO) outperforms training on rigid/affine motion (FC).

		Sintel-test		Sintel-test occ		Human flow	KITTI val	
		final	clean	final	clean		2012	2015
Zero flow		–	–	–	–	0.73	28.23	24.03
DeepFlow		6.93	–	38.17	–	–	–	–
DeepFlow*		7.21	–	38.78	–	–	–	–
FNS	FC	8.16	7.17	35.88	34.02	0.63	4.63	7.71
	DMO	**7.64**	**6.61**	**34.98**	**33.17**	**0.36**	**3.53**	**5.30**
LFN	FC	7.89	6.77	38.79	37.28	0.30	2.75	7.61
	DMO	**7.73**	**6.50**	**38.68**	**36.30**	**0.26**	**2.73**	**6.27**
PWC	FC	6.97	5.61	33.58	30.61	0.30	2.22	5.36
	DMO	**6.62**	**5.52**	**31.56**	**30.00**	**0.26**	**1.72**	**3.18**
RAFT	FC	7.69	**4.48**	35.11	**24.99**	**0.21**	3.55	6.17
	DMO	**6.15**	4.86	**32.75**	28.39	0.25	**1.45**	**2.58**

5.3. Performance with SOTA flow prediction networks

We compare different state-of-the-art algorithms for optical flow, namely LiteFlowNet (LFN) (Hui et al., 2018), PWC-Net (Sun et al., 2018), and RAFT (Teed and Deng, 2020). For each network, we compare the performance between the models trained with DMO dataset against the same model trained on FlyingChairs. The results are evaluated on the MPI-Sintel benchmark server (Sintel-test), the HumanFlow and KITTI 2012, 2015 datasets in Table 3.

Except for RAFT, the networks trained with our dataset outperform those trained with FlyingChairs on most of the cases. The results on Sintel occluded regions show that the proposed dataset improves models' robustness even in the challenging occlusion conditions. The results on HumanFlow show that non-rigid optical flow estimation (e.g. human body movement) benefits from DMO. In particular, although FNS is well-known for poor performance on small-displacement data (Ranjan et al., 2018), the performance on HumanFlow trained with DMO is close to that of the other powerful algorithms, whereas FNS trained on FlyingChairs is close to zero-flow, which is consistent with the conclusion of Ranjan et al. (2018) as FlyingChairs was used in their experiment. This suggests that the weakness of the methods can be improved using our proposed dataset. The results are mixed for RAFT and consistent to the results in table of Fig. 5(b) where FlyingChairs more superior on Sintel clean pass and HumanFlow as they are both

Table 4
Comparison to unsupervised (U), transferred (T) and finetuned methods (Ft). PWCNet trained on DMO outperforms all the transferred methods and compares favorably to finetuned methods after pre-training on FC. The gap between unsupervised/transferred methods and finetuned methods indicate the necessity of annotated optical flow.

	Training dataset		Sintel-test		Sintel-test occ	
			final	clean	final	clean
MFOF	RoamingImages	T	8.81	7.23	39.70	36.78
DDFlow	FC	T	7.40	6.18	39.94	38.05
SelFlow	SintelM	U	6.57	6.56	34.72	38.30
SelFlow	SintelM → KITTI → Sintel	Ft	**4.26**	**3.74**	**22.37**	**22.50**
PWC	FC	T	6.97	5.61	33.58	30.61
PWC	FC → Sintel	Ft	6.22	5.38	30.46	29.69
PWC	DMO	U/T	6.62	5.52	31.56	30.00
PWC	DMO → Sintel	Ft	**5.86**	**5.26**	**29.09**	**29.75**

clean-cut and contain no artifacts. On the other hand, RAFT trained with the artifacts of DMO are more robust to the artifacts in the Sintel final pass. This is an interesting benefit of artifacts that could be further exploited to improve optical flow datasets.

It should also be noted that both Sintel and HumanFlow data are computer generated (synthetic), while DMO contains only real-world textured images. Moreover, the results on KITTI val set indicate how the networks perform on real-world textured images. The network trained with our dataset outperforms those trained with FlyingChairs. The performance gaps are larger with more sophisticated networks, such as PWC and RAFT. This shows the importance to have real-world textured datasets for real-world scenarios. Therefore, we conclude that optical flow prediction networks benefit from training on natural textures and non-rigid movements, which are generated by our dense optical flow method.

5.4. Performance on unsupervised and finetuned methods

In this section, we measure the ability of our dataset DMO to transfer to different domains. To this end, we compare PWC trained on our dataset with unsupervised methods such as MFOF (Janai et al., 2018), DDFlow (Liu et al., 2019a), and SelFlow (Liu et al., 2019b). We train the network on a dataset generated in an unsupervised way, without using any ground truth optical flow from the test domain, i.e. the MPI-Sintel dataset. We also show the results of finetuning on the Sintel train set with results reported by finetuned state-of-the-art methods. The results are provided in Table 4.

For the transferred methods, PWC trained on our dataset outperforms the others, showing the transferrability of our method. For the finetuning methods, SelFlow performs the best. Note the improvement of the supervised results (Ft) over the unsupervised (U) and how it is trained on Sintel-related data (pretrained on SintelMovie, finetuned with MPI-Sintel flow ground truth). This shows the necessity of having annotated data on target domains.

5.5. Performance on real-world images

As there are currently no optical flow benchmarks with real-texture and non-rigid motion, we qualitatively show the results of PWC-Net and LiteFlowNet on real world images. Fig. 7 presents the optical flow prediction by LiteFlowNet (top) and PWC-Net (bottom) trained with FlyingChairs and our DMO on the QUVA repetition dataset (Runia et al., 2018). The models trained with our non-rigid flow set capture better delineation and details of the objects, especially for non-rigid movements of human body parts (indicated by the changes of colors).

Fig. 7. Qualitative results on QUVA dataset (Runia et al., 2018) of PWCNet trained on FC (midlle) and DMO (bottom). The networks trained using our pipeline capture the non-rigid motion of objects in the scenes with higher detail and delineation.

Table 5
PWC-Nets finetuned on ARAP-generated flow perform closely to one finetuned on the ground truth flow of Sintel and outperfom the baseline, showing the usefulness and realism of the generated flow.

Training dataset	Sintel-test		Sintel-test occ	
	final	clean	final	clean
FlyingChairs	6.97	5.61	33.58	30.61
ft Sintel-GT	**6.22**	**5.38**	**30.46**	**29.69**
ft ARAP-Sintel	6.63	5.47	33.02	31.02
ft ARAP-SintelM	6.48	5.50	30.08	32.67

5.6. Realism test

In this experiment, the goal is to assess the realism of the optical flow generated by the ARAP deformation model. The baseline is PWC-Net pre-trained on the FlyingChairs dataset. Various settings are considered, including finetune the same network architecture using (a) ground truth Sintel flow, (b) ARAP-generated flows from Sintel-train images, and (c) ARAP-generated flows from Sintel movie. The results are presented in Table 5. The performance of the network finetuned on ground truth flows are considered the theoretical maximum as GT flow is not always available in reality, while the distribution is more aligned with the test set distribution. The results show that we improve our baseline and achieve results that are closer to the theoretical upper bound. Therefore, we conclude that ARAP-generated flow could capture the data distribution.

6. Conclusion and discussion

In this paper, we introduced a pipeline to generate densely annotated optical flow datasets from videos to train supervised deep networks for optical flow. The method employs a matching process to capture the motion characteristics in videos while varying objects' textures to increase appearance variations. An as-rigid-as-possible (ARAP) image deformation is performed and constrained on the pixel matches to obtain an optical flow field. The flow field is used to warp the first-frame input to create the second frame of the pair for which it is the ground truth flow field, and thus guarantee its correctness. Extensive experiments are done on the framework to pick the best performance configuration, *i.e.*using DeepMatching and off-the-shelf Mask-RCNN. The generated dataset tested on several state-of-the-art optical flow prediction architectures show favorable results over the commonly used FlyingChairs for pre-training purposes.

The method is propose to generate optical flow data, yet it is also beneficial for analysis purposes by isolating various factors that could affect optical flow learning and prediction. On the one hand, it could be seen that optical flow of objects' complicated 3D motion could be learnt with high effectiveness from its simplified 2D deformation. On the other hand, it provides a new insight that the intermediate components

do not need to be free of errors to obtain a dataset that improves performance. The artifacts due to mismatches and uncertainties of background segmentation show favorable for training a network robust to imagery artifacts. This could open a new direction for exploring optical flow datasets and our method can be used as a controlled way to generate optical flow with high variability.

There exist many robust and accurate non CNN-based dense optical flow methods, e.g. (Trinh and Daul, 2019). Despite working for large and small displacements, scenes with varying textures and under strong illumination changes, the goal of such methods and ours is different. The method predicts optical flow from descriptors modeling biological homologous image regions and assumes small neighboring changes in complex scenery. In contrast, our paper aims to generate a dataset guided by motion statistics captured by the matching process with the purpose to train optical flow networks. On the other hand, the theoretical basis for illumination robustness described in the given paper could be considered in a follow-up research to produce more robust matching with reduced artifacts.

In the paper, the image deformation as-rigid-as-possible is employed in conjunction with deep matching to capture non-rigid motion characteristics. However, any method that generates optical flow fields can be used as long as the second frame of the pair is created by warping the first-input frame with the generated flow. Exploring this direction will lead to better understandings of optical flow datasets.

CRediT authorship contribution statement

Hoàng-Ân Lê: Methodology, Software, Visualization, Writing – original draft, Formal analysis, Investigation, Data curation, Validation. **Tushar Nimbhorkar:** Investigation. **Thomas Mensink:** Validation, Writing – review & editing, Supervision. **Anil S. Baslamisli:** Validation. **Sezer Karaoglu:** Validation, Writing – review & editing. **Theo Gevers:** Conceptualization, Project administration, Validation, Supervision, Writing – review & editing, Resources, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work is performed within the TrimBot2020 project funded by the EU Horizon 2020 program No. 688007.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.cviu.2021.103274.

References

Alexa, M., Cohen-Or, D., Levin, D., 2000. As-rigid-as-possible Shape Interpolation. In: Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '00.

Butler, D., Wulff, J., Stanley, G., Black, M., 2012. A naturalistic open source movie for optical flow evaluation. In: ECCV.

Cao, J., Anwer, R.M., Cholakkal, H., Khan, F.S., Pang, Y., Shao, L., 2020. SipMask: Spatial Information Preservation for Fast Image and Video Instance Segmentation.

DeVito, Z., Mara, M., Zollöfer, M., Bernstein, G., Theobalt, C., Hanrahan, P., Fisher, M., Nieß ner, M., 2017. Opt: A domain specific language for non-linear least squares optimization in graphics and imaging. ACM Trans. Graph..

Dosovitskiy, A., Fischer, P., Ilg, E., Häusser, P., Hazirbacs, C., Golkov, V., v.d. Smagt, P., Cremers, D., Brox, T., 2015. FlowNet: Learning optical flow with convolutional networks. In: ICCV.

Dvorožňák, M., 2014. Interactive as-rigid-as-possible image deformation and registration. In: Central European Seminal on Computer Graphics.

Geiger, A., Lenz, P., Urtasun, R., 2012. Are we ready for autonomous driving? The KITTI vision benchmark suite. In: CVPR.

He, K., Gkioxari, G., Dollar, P., Girshick, R., 2017. Mask R-CNN. In: ICCV.

Hui, T.-W., Tang, X., Loy, C.C., 2018. LiteFlowNet: A lightweight convolutional neural network for optical flow estimation. In: CVPR.

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: CVPR.

Janai, J., Güney, F., Ranjan, A., Black, M.J., Geiger, A., 2018. Unsupervised learning of multi-frame optical flow with occlusions. In: ECCV.

Janai, J., Güney, F., Wulff, J., Black, M., Geiger, A., 2017. Slow flow: Exploiting high-speed cameras for accurate and diverse optical flow reference data. In: CVPR.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollar, P., Zitnick, C.L., 2014. Microsoft COCO: Common objects in context. In: ECCV.

Liu, P., King, I., Lyu, M.R., Xu, J., 2019a. Ddflow: Learning optical flow with unlabeled data distillation. In: AAAI.

Liu, P., Lyu, M.R., King, I., Xu, J., 2019b. Selflow: Self-supervised learning of optical flow. In: CVPR.

Mayer, N., Ilg, E., Fischer, P., Hazirbas, C., Cremers, D., Dosovitskiy, A., Brox, T., 2018. What makes good synthetic training data for learning disparity and optical flow estimation? IJCV.

Meister, S., Hur, J., Roth, S., 2018. UnFlow: Unsupervised learning of optical flow with a bidirectional census loss. In: AAAI.

Menze, M., Geiger, A., 2015. Object scene flow for autonomous vehicles. In: CVPR.

Pont-Tuset, J., Perazzi, F., Caelles, S., Arbeláez, P., Sorkine-Hornung, A., Van Gool, L., 2017. The DAVIS challenge on video object segmentation. arXiv:1704.00675.

Ranjan, A., Romero, J., Black, M.J., 2018. Learning human optical flow. In: BMVC.

Revaud, J., Weinzaepfel, P., Harchaoui, Z., Schmid, C., 2015. DeepMatching: Hierarchical deformable dense matching. In: IJCV.

Rocco, I., Cimpoi, M., Arandjelović, R., Torii, A., Pajdla, T., Sivic, J., 2018. Neighbourhood consensus networks. In: NeurIPS.

Runia, T.F., Snoek, C.G., Smeulders, A.W., 2018. Real-world repetition estimation by div, grad and curl. In: CVPR.

Sun, D., Yang, X., Liu, M.-Y., Kautz, J., Nvidia, J.K., 2018. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In: CVPR.

Teed, Z., Deng, J., 2020. RAFT: Recurrent all pairs field transforms for optical flow. In: ECCV.

Trinh, D.-H., Daul, C., 2019. On illumination-invariant variational optical flow for weakly textured scenes. Comput. Vis. Image Underst. 179, 1–18.

Wang, Y., Xu, K., Xiong, Y., Cheng, Z.-Q., 2008. 2D shape deformation based on rigid square matching. Comput. Animat. Virtual Worlds.

Weinzaepfel, P., Revaud, J., Harchaoui, Z., Schmid, C., 2013. DeepFlow: Large displacement optical flow with deep matching. In: ICCV.