Weakly-Supervised Optical Flow Estimation for Time-of-Flight

Michael Schelling, Pedro Hermosilla, Timo Ropinski
Ulm University, Germany

https://github.com/schellmi42/WFlowToF

Abstract

Indirect Time-of-Flight (iToF) cameras are a widespread type of 3D sensor, which perform multiple captures to obtain depth values of the captured scene. While recent approaches to correct iToF depths achieve high performance when removing multi-path-interference and sensor noise, little research has been done to tackle motion artifacts. In this work we propose a training algorithm, which allows to supervise Optical Flow (OF) networks directly on the reconstructed depth, without the need of having ground truth flows. We demonstrate that this approach enables the training of OF networks to align raw iToF measurements and compensate motion artifacts in the iToF depth images. The approach is evaluated for both single- and multi-frequency sensors as well as multi-tap sensors, and is able to outperform other motion compensation techniques.

1. Introduction

Time-of-Flight (ToF) cameras are sensors that aim to capture depth images by measuring the time the light needs to travel from a light source on the camera to an object and back to the camera sensor. Apart from direct ToF cameras, such as LiDAR, which register the time of incoming reflections of a light pulse at a high temporal resolution, another common and cost-efficient approach are indirect ToF (iToF) cameras, which do not require as precise measuring devices. One realization of iToF devices are Amplitude Modulated Continuous Wave (AMCW) ToF sensors, as for example used in the Kinect system. These sensors continuously illuminate the scene with a periodically modulated light signal and aim to retrieve the phase offset between the emitted and the retrieved signal, which gives information about the travel time of the signal [11]. In order to retrieve the phase offset it is necessary to perform multiple captures, which makes this approach sensible to movements of both, the camera and the objects in the illuminated scene. As the measurements are taken with differing sensor settings, so called multi modality, standard Optical Flow (OF) algorithms achieve only low performance, and hence require adaptation. While there are works that investigate the compensation of motion using OF, they are only applicable to specific sensor types [18, 12] or require carefully designed datasets [9] to train OF networks. Hence, it is still a common approach to merely detect motion artifacts and mask the affected pixels in the final depth image, as is for example realized by the LF2-algorithm [27] for the Kinect sensor.

In this work, we propose a training algorithm for OF networks which allows to supervise the flow prediction using the ToF depth image, without the need to directly supervise the predicted flow, see Fig. 1. To this end, we analyze the ToF depth computation to provide reliable and stable gradients during training. Further, we introduce a set of regulatory losses, which guide the network towards predicting flows, that are consistent with the underlying images.

2. Technical Background

In this section, we briefly describe iToF cameras.

ToF Working Principle. An AMCW iToF camera emits a modulated light signal $s(t)$, which is correlated at the sensor with a phase shifted version of the emitted signal $s(t+\theta)$ during the exposure time. The resulting measurement m is repeated sequentially for different phase shifts θ, from which the distance d is retrieved indirectly by estimating the phase shift $\Delta\varphi$ of the signal s when arriving at the sensor. In the common case of four measurements $m_0, \dots, m_3$ at $\theta \in \{0, \pi/2, \pi, 3\pi/2\}$, the distance d is retrieved as

$$\Delta\varphi = \arctan\left(\frac{m_3 - m_1}{m_0 - m_2}\right), \tag{1}$$

$$d_{ToF} = \frac{c \cdot \Delta\varphi}{4\pi f}, \tag{2}$$

where c is the speed of light, and f is the modulation frequency of the signal s [11]. Due to the periodic nature of Eq. (1), the reconstructed d_{ToF} is only unambiguous up to a maximum distance of

$$d_{max} = c/(2f), \tag{3}$$

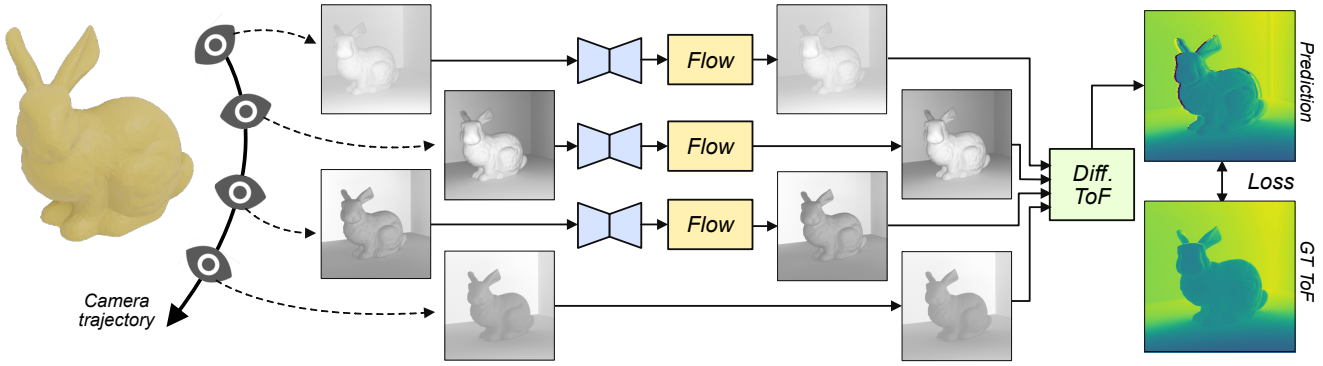

Figure 1. Illustration of the flow estimation. Given iToF measurements at subsequent time steps, a network is used to predict optical flows, in order to align the images to the reference image (bottom row). From the warped measurements a ToF depth image can be reconstructed. We propose to supervise the training directly on this ToF depth, and propagate gradients through the ToF depth computation. This figure shows the single frequency, single-tap case with four measurements. Note the modality change in the input due to different phase shifts θ.

specifically, $d_{ToF} = d \mod d_{max}$, where the distance d is referred to as *depth*, as is common practice in the area of ToF imaging. The so called phase wrapping of d_{ToF} is typically resolved by using additional measurements at different frequencies f [11].

However Eq. (2) is based on the assumptions that, (a) only the direct reflection $s(t + \Delta\varphi)$ is captured and (b) the scene is static between the different captures. While (a) has been dealt with to a large extent in recent work on correcting iToF depths [1, 20, 23], only little research has been done to reduce motion artifacts stemming from (b).

Multi-Tap Sensors. A realization of iToF sensors are so called *multi-tap* sensors, which are able to capture multiple measurements of m_θ in parallel. The most widespread approach are two-tap sensors, which allow the capture of $m_{A,i} = m_i$ and $m_{B,i} = m_{i+2}$ at the same time, by sorting the electrons generated by incoming photons into two quantum wells using a modulated electric field [24]. Internally, these two measurements are used to compensate for hardware inaccuracies and reduce noise [12] by computing:

$$m_i = m_{A,i} - m_{B,i}. \tag{4}$$

In order to make direct use of m_A, m_B in Eq. (1), it is necessary to calibrate the differences in the photo responses [24]

$$m_{A,i} = r_\theta(m_{B,i+2}), \tag{5}$$

which doubles the effective frame rate, and reduce, but not eliminate, motion-artifacts. Recently also prototypes for four-tap sensors have been developed [5, 15], which in the future might eliminate motion artifacts in single-frequency captures, but not in multi-frequency sensors.

3. Related Work

This section briefly summarizes previous work on related fields.

ToF Motion Artifact Correction. Early methods on motion compensation used detect-and-repair approaches [24, 11], *e.g.* by performing bilateral filtering [19]. One of the first methods to resolve movement artifacts using optical flow was introduced by Lindner *et al.* [18] who aim to tackle the cross modality through a correction scheme to compute intensity images from two-tap captures, which can be used as input to a standard OF algorithm. Based on this method, Hoegg *et al.* [12] derived optimizations for the OF prediction algorithm by incorporating motion detection and refining the spatial consistency to achieve real-time performance. The performance of these approaches was further improved with the calibration of Gottfried *et al.* [8]. In contrast we integrate the entire computational flow, from raw iToF measurement to depth reconstruction into our optimization pipeline.

The first learned approach was presented by Guo *et al.* [9], who provide methods to correct errors for the Kinect2 sensor, including an encoder-decoder network for OF prediction. To enable the supervised learning of motion compensation, a specific dataset is generated, which allows for simulating linear movements in the image domain, while separating the motion of foreground and background. Contrarily, we propose a weakly supervised training, which does not require flow labels, and instead uses ToF depths for supervision, which are available in existing iToF datasets.

Optical Flow. Recent works on OF regression rely on neural networks, which have proven to outperform traditional approaches [26]. The typical design, using shared image encoders and a latent cost volume, was first introduced Dosovitskiy *et al.* [7] in their FlowNetC architecture, alongside the FlowNetS network, which uses a encoder decoder architecture. Subsequent, a large literature on various applications [29, 17] and formulations [2, 13] in the field of motion estimation emerged. In order to reduce the com-

putational costs, Sun *et al*. [26] introduced a hierarchical architecture with coarse-to-fine warping in their Pyramid-Warping-Cost-volume (PWC) network. This design was further refined by Kong *et al*. [16] in their FastFlowNet (FFN) architecture, which reduced the computational complexity and achieves fast inference times.

To overcome the need of generating ground truth flows for a supervised training, unsupervised approaches [14, 22, 28, 13] optimize the photometric consistency between images and apply regularizations to refine the flow prediction.

ToF Correction. The occurrence of Multi-Path-Interference (MPI) is the main source of errors in iToF depth reconstructions. Consequently, existing works on correcting iToF data focus on removing MPI artifacts. As with OF prediction, 2D neural networks have proven to achieve high noise removal performance [1, 20, 25, 9, 6]. However, also other learned approaches have been investigated recently, such as reconstructing the transient response [4, 10] or using 3D point networks [23].

4. Method

In this work we propose a weak supervision of an OF network using the ToF-depth d_{ToF} as label, without providing ground truth flow vector fields. In order to enable training using depth labels, the phase wrapping discontinuities in Eq. (1) of the $\arctan$ function require consideration, and regularizations on the flow prediction need to be established to predict consistent flows without direct supervision.

We consider an OF network $g : (\{m_i\}_{i=0}^{N-1}, m_N) \rightarrow \{V_i\}_{i=0}^{N-1}$, which predicts a set of optical flows V_i for a set of measurements m_i, in order to align them to a measurement m_N taken at the reference time step. The standard photometric loss in this setting would be given as

$$\hat{m}_i = warp(m_i, V_i) \tag{6}$$

$$\mathcal{L}_{photo} = \sum_i \|\hat{m}_i - m_i^{GT}\|_1, \tag{7}$$

where m_i^{GT} is taken at the same time step as m_N.

Instead, we propose to supervise the network g indirectly on the reconstructed depth using the ToF depth d_{ToF} without motion as target. To increase the numerical stability we formulate the reconstructed depth $\hat{d}$ as

$$s = \text{sign}(\hat{m}_0 - \hat{m}_2) \tag{8}$$

$$\hat{d} = \frac{c}{4\pi f} \arctan\left(\frac{\hat{m}_3 - \hat{m}_1}{\hat{m}_0 - \hat{m}_2 + s \cdot \epsilon}\right), \tag{9}$$

$$\mathcal{L}_{ToF} = \|\hat{d} - d_{ToF}\|_1, \tag{10}$$

which avoids singularities as the denominator in Eq. (9) is strictly positive for $\epsilon > 0$. The implementation of Eq. 10 on commonly used learning packages with auto-differentiable features, such as Pytorch [21] or JAX [3], allows to train the flow network g in a weakly-supervised fashion.

4.1. Phase Unwrapping

The phase wrapping in the above formulation can be tackled by generating multiple candidate depths $\hat{d}_k = \hat{d} + k \cdot d_{max}$ and using the one closest to the label as prediction

$$\hat{d}_k = \hat{d} + k \cdot d_{max} \tag{11}$$

$$\mathcal{L}_{ToF,PU} = \min\{\|\hat{d}_k - d_{ToF}\|_1 \mid k \in \mathbb{Z}\}. \tag{12}$$

As both $\hat{d}$ and d_{ToF} are in the range of $[0, d_{max})$, the candidate space is reduced to $k \in \{-1, 0, 1\}$ and the minimization in Eq. (12) can be realized by a simple lookup table

$$\hat{d} - d_{ToF} \in (-d_{max}, d_{max}/2] : \qquad k = -1, \tag{13}$$

$$\hat{d} - d_{ToF} \in (-d_{max}/2, d_{max}/2] : \qquad k = 0, \tag{14}$$

$$\hat{d} - d_{ToF} \in (d_{max}/2, d_{max}) : \qquad k = 1. \tag{15}$$

However, during training only the gradients of $\mathcal{L}_{ToF,PU}$ are relevant, which can be derived from the lookup table as

$$\nabla\mathcal{L}_{ToF,PU} = \begin{cases} \nabla\mathcal{L}_{ToF}, & 0 \leq \mathcal{L}_{ToF} < d_{max}/2, \\ -\nabla\mathcal{L}_{ToF}, & \mathcal{L}_{ToF} \geq d_{max}/2, \end{cases} \tag{16}$$

and can thus be directly computed from Eq. (10). This allows a computational cheap and elegant implementation of the phase unwrapping, by only adjusting the gradients of $\mathcal{L}_{ToF}$, Eq. (10), based on the conditions in Eq. (16) in the backpropagation step during the training of g.

4.2. Regularization

By regularizing the predictions, additional constraints for the predicted flows V_i are established, which enables the network to produce coherent predictions without using flow labels. We use two additional regularization losses, a smoothing loss $\mathcal{L}_{smooth}$ and an edge-aware loss $\mathcal{L}_{edge}$.

For smoothing we adapt the formulation of Jonschkowski *et al*. [14] to our setting

$$\mathcal{L}_{smooth} = \sum_{i,j} \exp\left(-\lambda \left|\frac{\partial m_i}{\partial x_j}\right|\right) \cdot \left|\frac{\partial V_i}{\partial x_j}\right|, \tag{17}$$

where λ is an edge weighting factor and x_0, x_1 are the two image dimensions. This loss penalizes high gradients on V_i in homogeneous regions of m_i, *i.e*. regions where m_i has small gradients. The intuition of $\mathcal{L}_{smooth}$ is that homogeneous regions are expected to move in the same direction.

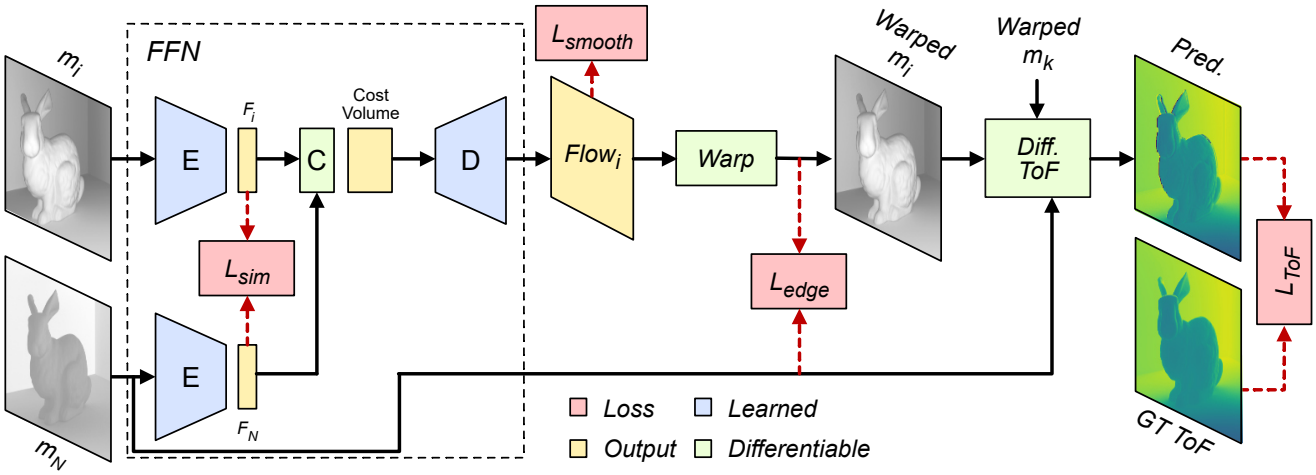

Figure 2. Overview over the loss functions used in this work. Our main loss is the ToF-loss $\mathcal{L}_{ToF}$ (right), which is computed on the reconstructed ToF depth using a differentiable operation, and is adapted to provide phase unwrapped gradients. To constrain the flow prediction the loss $\mathcal{L}_{smooth}$ (top) is used to regularize the flow, and an additional regularization on the warped image m_i is given through the loss $\mathcal{L}_{edge}$ (center). Finally, the loss $\mathcal{L}_{sim}$ aims to create consistency between the latent representations inside the network. Note: The losses $\mathcal{L}_{ToF}$ and $\mathcal{L}_{sim}$ are computed over all i. This figure shows the single-tap case, where only one measurement is taken per time step.

To further regularize the network to predict correctly aligned object boundaries, we introduce an edge-aware loss

$$\mathcal{L}_{edge} = \sum_{i,j} \exp\left(\frac{-1}{\epsilon + \left|\frac{\partial m_N}{\partial x_j}\right|}\right) \cdot \frac{1}{\left|\frac{\partial \hat{m}_i}{\partial x_j}\right| + s}, \tag{18}$$

where ϵ is a small constant for numerical stability and the shift s is used to provide an upper bound on the gradients of $\mathcal{L}_{edge}$. This loss penalizes small gradients in the warped measurements $\hat{m}_i$ in regions where m_N has large gradients, *i.e.* regions where m_N has edges. The intuition of $\mathcal{L}_{edge}$ is that boundaries of objects can be expected to create edges in the measurements independent of the modality.

Note that $\mathcal{L}_{smooth}$ acts on the flows V_i whereas $\mathcal{L}_{edge}$ is computed on the warped measurements $\hat{m}_i$, see Fig. 2.

4.3. Cross Modality

To guide the network towards learning latent representations F_i, see Fig. 2, that are robust to the input modality, we make use of a latent similarity loss on the column vectors $F_i(k,l)$ of the latent representation in $g(m_i)$, inspired by the formulation of contrastive learning

$$\mathcal{L}_{sim} = \sum_{i \neq j} \sum_{k,l} L\Big(F_i(k,l), F_i(k,l)\Big), \tag{19}$$

where L is a similarity loss, *e.g.* L_1, L_2, the cosine-similarity or a cost function.

During training we optimize the similarity loss on static scenes, without motion. An overview of all losses and their integration in the computational flow are shown in Fig. 2.

4.4. Network Architecture

As OF backbone we investigate two networks with different architectures, the Motion Module (MOM), which was introduced by Guo *et al.* [9] for ToF motion correction, and the FFN of Kong *et al.* [16] which is a lightweight network with on-par performance to State-of-the-Art OF networks. The MOM network is an encoder-decoder network based on FlowNetS [7], while the FFN integrates a latent cost volume and is based of the PWC network. Both networks allow for fast evaluation times and low memory consumption which enables us to predict multiple flows.

While the flow prediction of the MOM network is rather straightforward, *i.e.* it takes the set $\{m_i\}_{i=0}^N$ as input and predicts all flows $\{V_i\}_{i=0}^N$ at once, we will briefly describe how we execute the FFN in the following. Please note, that the computations of FFN are realized on a hierarchical feature pyramid, but for compact notation we neglect the hierarchy levels in the following description.

The FFN consists of the common building blocks, an image encoder E, a cost volume computation C and a flow prediction decoder D. Given the measurements $\{m_i\}_{i=0}^N$, we encode each measurement m_i into a latent vector $F_i = E(m_i)$. The latent vectors are then used to compute cost volumes for each pairing with the last measurement m_N, *i.e.* $c_i = C(F_i, F_N)$ for $i = 1, \ldots, N-1$. The decoder then predicts the flows using pairs of cost volumes and latent vectors as input $V_i = D(F_i, c_i)$, the process for a single image pair is also shown on the left of Fig. 2. After warping the measurement m_i, parts of the image might remain empty, as no pixels were warped to this region, these regions are referred to as masked.

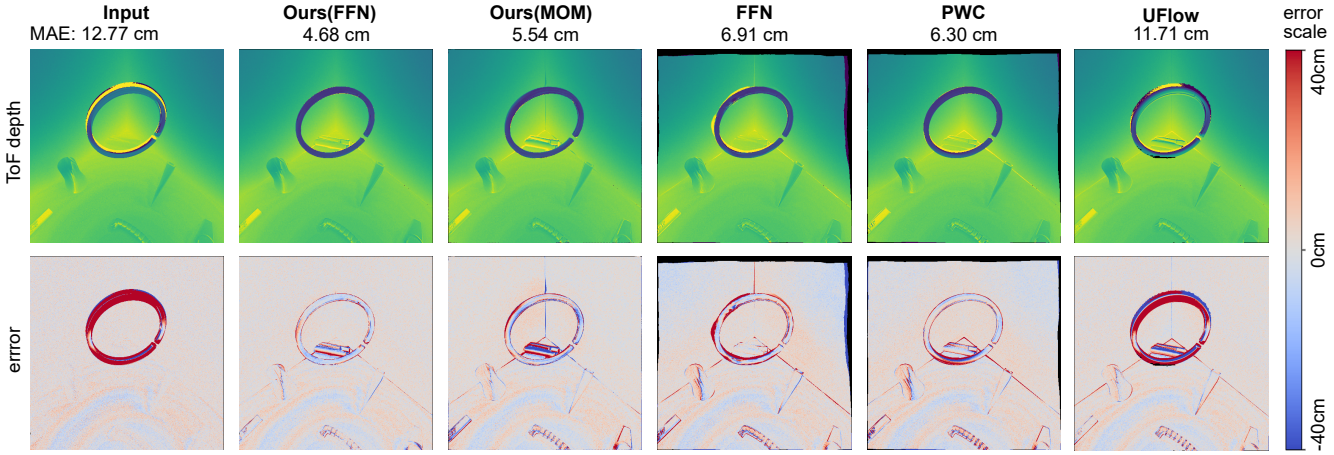

Figure 3. Motion compensation results in the single frequency single tap case. Both pre-trained networks and our method resolve the motion artifacts, however our method improves performance over the pre-trained networks. Moreover, the UFlow method is not able to correct the motion artifacts. However, while the camera is static and only the center object is moving, all methods have some tendency to move the background, which introduces additional artifacts. (Empty regions after warping are shown in black.)

In this formulation the network only considers the two measurements m_i, m_N to compute V_i. Although the other measurements contain additional information about the movement, the above formulation allows to share the encoder and decoder networks for all measurements and does not increase the number of parameters.

We further apply an instance normalization to the input of the network, as also used in the ToF error correction approach of Su *et al*. [25], which does not affect the depth reconstruction in Eq. (2), as it is invariant to uniform scaling and translation of the measurements.

In case of multi-tap sensors we change the input dimension of the encoder E such that it receives all measurements captured at the same time step as input.

5. Experiments

In our experiments we train instances of both FFN and MOM using the loss functions described in Sec. 4. In the case of the MOM network we do not use the similarity loss $\mathcal{L}_{sim}$, as the network does not produce latent vectors F_i due to its different architectural design. We compare against using pre-trained instances on RGB data of FFN and and also the larger PWC [26], which needs ≈8 times the compute [16]. In the case of multi-tap sensors we additionally compare against the Lindner method [18] in combination with the pre-trained instances of FFN and PWC. Further, we compare against the UFlow method [14], which is a method to train OF networks in an unsupervised fashion, and uses the PWC as backbone. We train the UFlow method on the same dataset as our method.

Dataset. We conduct the experiments on the CB-dataset of Schelling *et al*. [23], as it contains raw measurements m_i for three different frequencies. It consists of 143 scenes each rendered from 50 viewpoints along a camera trajectory, which allows to simulate real movements that change the point of view. As the CB-Dataset only incorporates static scene geometries we generated 14 additional scenes with moving objects using the same data simulation pipeline, to increase the variation of movements in the dataset. We divide the dataset using the original training,

	Method	$\mathcal{L}_{photo}$	$\mathcal{L}_{ToF}$	mask
SF 1Tap	Input	50.09	16.87	-
	FFN	54.21	14.63	12.40%
	PWC	49.16	13.70	4.12%
	UFlow	58.71	12.76	3.24%
	Ours(MOM)	34.64	7.64	0.97%
	Ours(FFN)	**23.27**	**5.81**	1.60%
SF 2Tap	Input	34.45	5.93	-
	FFN	29.83	5.44	6.18%
	PWC	19.77	4.03	3.55%
	UFlow	38.22	4.90	2.07%
	Lindner (FFN)	21.01	4.22	2.35%
	Lindner (PWC)	18.11	3.85	2.12%
	Ours(MOM)	24.67	**3.25**	0.73%
	Ours(FFN)	**17.22**	3.66	0.56%

Table 1. Results for single frequency single-tap (SF 1Tap) and two tap (SF 2Tap). The pre-trained networks, FFN and PWC, and the unsupervised UFlow method achieve only low correction rates in most cases. The Lindner method reduces the error notably, especially when using the larger PWC as backbone, still it is outperformed by our proposed method on smaller backbones.

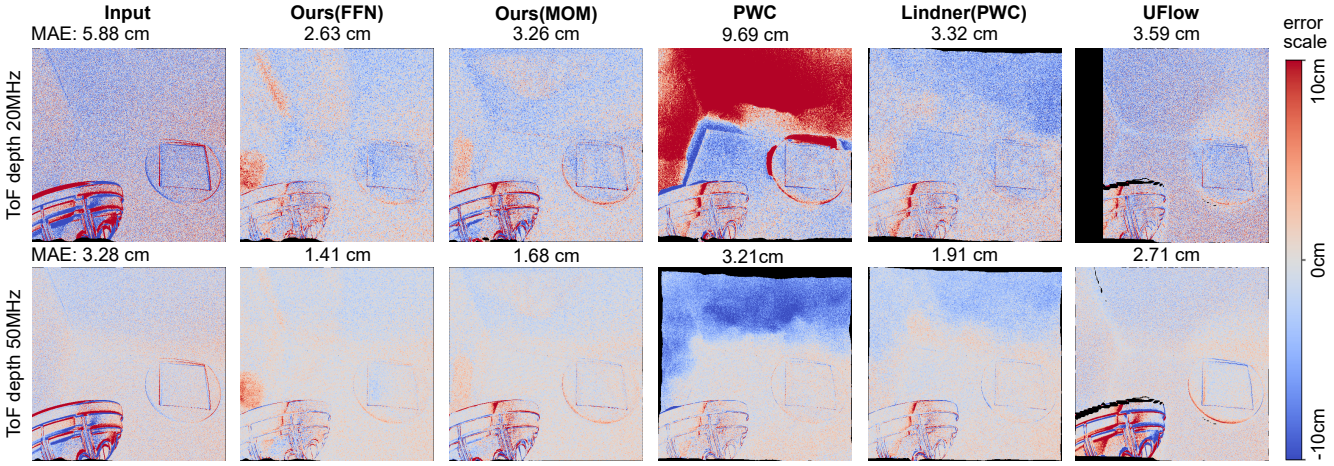

Figure 4. Motion compensation results in the multi frequency four-tap case, for a scene with moving camera. Our method achieves the best motion compensation, followed by Lindner's method on the more powerful backbone PWC, although Lindner's method introduces more additional errors. Both the pre-trained PWC and the UFlow method fail in this case. (Empty regions after warping are shown in black.)

validation and testing split, and further divide the additional scenes into 10 training scenes, and 2 each for testing and validation, whereby we use the 20MHz measurements.

5.1. Single Frequency Motion Compensation

For the single frequency experiment we also use the 20MHz measurements of the datasets. In the case of single-tap we take the four measurements from four subsequent time steps, in the case two-tap we take the pairs (m_0, m_2) and (m_1, m_3) from two times steps. We measure $\mathcal{L}_{ToF}$, the photometric loss $\mathcal{L}_{photo}$ and the percentage of masked pixels after warping, and report results on the test set in Tab. 1. We find that the networks trained with our method achieve better results than the pre-trained OF networks and the UFlow method. Results for the single tap case can be seen in Fig. 3. The results of Lindner's method come close to our method, but only when using the larger backbone network PWC. On the same backbone FFN the gap in performance is larger. Additionally, in the simple setting of two-taps, and thus also two time steps, the simple MOM backbone results in better performance than the more complex FFN backbone, both trained with our method.

Further, we observe that the UFlow method increases the photometric loss, which we attribute to the fact that the method aims to minimize the photometric loss between the images of different modalities. Additionally, UFlow has a tendency to mask out areas affected by motion, as is shown in Fig. 4, which leads to a reduced ToF loss, without correcting the errors.

5.2. Multi Frequency Motion Compensation

For the multi frequency experiment we use the three frequencies 20MHz, 50MHz and 70MHz of the datasets. In the case of single-tap, we take the twelve measurements from twelve subsequent time steps. In the two-tap case, we take pairs (m_0, m_2) and (m_1, m_3) from six time steps. Lastly,

	Method	$\mathcal{L}_{photo}$	$\mathcal{L}_{ToF}$	mask
MF 1Tap	Input	113.73	19.68	-
	FFN	124.88	25.06	10.76%
	PWC	83.15	16.01	8.91%
	UFlow	136.55	13.86	7.76%
	Ours(MOM)	**65.91**	**11.92**	1.43%
	Ours(FFN)	80.43	13.77	0.34%
MF 2Tap	Input	69.06	8.17	-
	FFN	78.33	9.71	5.90%
	PWC	49.23	7.51	4.02%
	UFlow	81.45	5.95	4.82%
	Lindner (FFN)	40.26	5.60	2.55%
	Lindner (PWC)	35.24	5.16	1.80%
	Ours(MOM)	44.68	4.98	0.64%
	Ours(FFN)	**30.71**	**4.43**	0.32%
MF 4Tap	Input	40.42	5.26	-
	FFN	57.54	6.93	0.06%
	PWC	31.09	5.41	0.06%
	UFlow	51.10	4.17	1.96%
	Lindner (FFN)	27.52	3.94	0.06%
	Lindner (PWC)	**22.17**	3.49	0.06%
	Ours(MOM)	29.64	3.11	0.48%
	Ours(FFN)	27.14	**3.03**	0.08%

Table 2. Results for multi frequency single-tap (MF 1Tap), two-tap (MF 2Tap) and four-tap (MF 4Tap). In this setting with stronger modality changes pre-trained networks fail in most cases. Our method is again closely followed by Lindner on the larger PWC.

	Method	MAE	Rel. Error
SF 1Tap	Input	39.49	100.00%
	CFN	19.39	49.10%
	CFN + Ours(FFN)	**11.47**	**29.05%**
	DeepToF	16.65	42.17%
	DeepToF + Ours(FFN)	15.11	38.26%
MF 2Tap	Input	10.65	100.00%
	CFN	6.71	63.01%
	CFN + Ours(FFN)	**5.54**	**52.02%**
	E2E	10.44	98.03%
	E2E + Ours(FFN)	8.27	77.65%
	RADU	11.21	105.26%
	RADU + Ours(FFN)	8.00	75.12%

Table 3. Results of motion, multi-path-interference and sensor noise compensation, for the single frequency single tap (SF 1Tap) and the multi frequency two-tap (MF 2Tap) case. All methods benefit from the motion correction using our method.

in the case of four-tap, we use three time steps, one per frequency. The results on the test set for both $\mathcal{L}_{ToF}$ and the photometric loss $\mathcal{L}_{photo}$ are reported in Tab. 2, and are shown for the four-tap case in Fig. 4.

The findings from the single frequency experiment can also be observed in this setting, with our approach achieving the best performance followed by Lindner's method. Further, the FFN trained with our method, while still outperforming the other methods, achieves rather low performance in the single tap setting, which is arguably the hardest case with the highest number of time steps, and thus the largest motion, and additionally the lowest input dimensionality of only one tap, which might make it harder for the encoder E to extract modality invariant features.

Additionally, the pre-trained OF networks have a tendency to fail in these settings, especially the FFN, which might come from the larger modality gap of measurements taken at different frequencies, as can also be seen in Fig. 4.

5.3. Motion Compensation and Error Correction

To measure the influence on downstream error compensation techniques, we train instances of ToF correction networks on the output of our model. For this experiment, the single frequency single tap case and the multi frequency two-tap case are considered. We use the single frequency approaches DeepToF [20] and an adapted CFN [1] in the single frequency case, and the multi-frequency approaches CFN, E2E [25] and RADU [23] in the multi frequency case. For comparison we also train instances of the networks without performing motion compensation, and report results on the test set in Tab. 3

We observe, that all methods benefit from motion compensation in their input. We further observe that the 2D

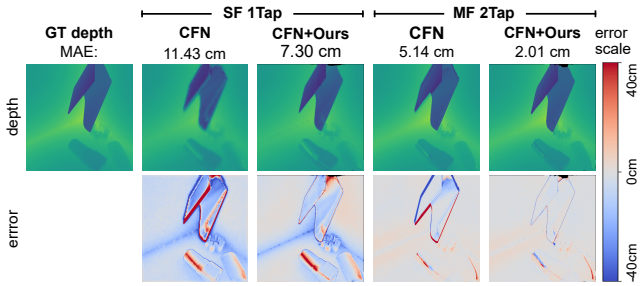

Figure 5. Results of combined motion and MPI correction using the CFN network. Without additional motion compensation, the motion artifacts are only partially corrected. In combination with method they are restricted to the object boundaries.

networks that frame the task as denoising handle motion artifacts quite well, see Fig. 5, whereas the more complex approaches E2E, which formulates a generative image translation task, and RADU, which operates on 3D point clouds, struggle in this setting. It is to be remarked, that none of the approaches were designed to correct motion artifacts.

5.4. Ablations

This section provides ablations on the loss components.

5.4.1 Component Ablation

To investigate the influence of each loss component separately, we train instances of the FFN network while disabling individual components. Further, we replace the ToF loss $\mathcal{L}_{ToF}$ with the photometric loss $\mathcal{L}_{photo}$ and additionally train an instance using only the ToF loss as baselines. The results on the validation set are reported in Tab. 4

From the results it can be seen that the combination of all losses achieves the best performance, and that each component reduces the loss. Out of the regulatory losses the smoothing loss $\mathcal{L}_{smooth}$ has the highest impact, followed by the edge-aware loss $\mathcal{L}_{edge}$ and finally the latent similarity loss $\mathcal{L}_{sim}$. Further, the ToF loss yields a large performance

Method	$\mathcal{L}_{photo}$	$\mathcal{L}_{ToF}$
Input	70.39	23.71
$\mathcal{L}_{photo} + \mathcal{L}_{smooth} + \mathcal{L}_{edge} + \mathcal{L}_{sim}$	38.65	12.43
$\mathcal{L}_{ToF}$	38.42	10.17
$\mathcal{L}_{ToF} + \mathcal{L}_{edge} + \mathcal{L}_{sim}$	35.94	9.67
$\mathcal{L}_{ToF} + \mathcal{L}_{smooth} + \mathcal{L}_{sim}$	34.65	8.54
$\mathcal{L}_{ToF} + \mathcal{L}_{smooth} + \mathcal{L}_{edge}$	32.57	7.87
$\mathcal{L}_{ToF} + \mathcal{L}_{smooth} + \mathcal{L}_{edge} + \mathcal{L}_{sim}$	**28.76**	**7.21**

Table 4. Ablation on the loss components in the single frequency single-tap case, using the FFN as OF backbone.

gain compared to the photometric loss, and even without regularizations achieves a better performance.

5.4.2 Similarity Loss Function

As the definition of the latent similarity loss $\mathcal{L}_{sim}$ in Eq. (19) was kept general, it allows for the usage of different similarity measures L. We investigate the standard L_1 and L_2 distances, the cost function that is used in the cost volume computation and the cosine similarity

$$L_p: \quad \|F_i(k,l) - F_j(k,l)\|_p, \qquad p = 1,2 \tag{20}$$

$$\text{Cost:} \quad -F_i(k,l) \cdot F_j(k,l), \tag{21}$$

$$\text{Cosine}: \quad \frac{-F_i(k,l) \cdot F_j(k,l)}{\|F_i(k,l)\|_2 \|F_j(k,l)\|_2}, \tag{22}$$

where $\cdot$ denotes the scalar product. We consider the single frequency single tap and the multi frequency two-tap case in this ablation, and train instances of the FFN using the above similarity measures, together with all other loss components. Further, we train an instance using no similarity loss as a baseline, and, in the case of two taps, compare to using Lindner's features as input instead of a similarity measure. From the results, which can be seen in Tab. 5, we find that the cosine similarity achieves the best performance in both cases. Additionally, in the multi frequency two-tap case, the cosine similarity is the only measure that improves over not using a similarity loss at all, including Lindner's method. Consequently, both the use and the choice of the similarity measure needs careful consideration.

6. Limitations

Although, both backbone OF networks achieve good results, we experience cases that escape our regularization losses. For example, the smoothing loss $\mathcal{L}_{smooth}$ ensures a continuous flow for an object, however objects are detected based on their homogeneous appearance, which can fail on high frequency details. While the edge loss $\mathcal{L}_{edge}$ can resolve most of the cases, still sometimes wrong parts of the images are matched, especially when nearby image patches have a similar appearance, see Fig. 6. We attribute this to

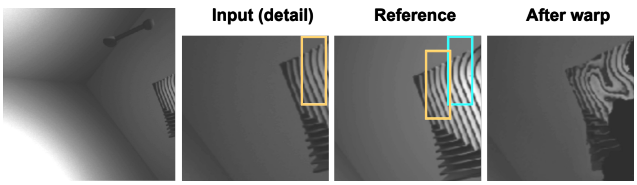

Figure 6. Example of an object, where our regularizations fail. The high frequency pattern prevents $\mathcal{L}_{smooth}$ from enforcing a consistent flow for the object. Due to the repetitive pattern, the network matches the yellow region in the input image with the cyan region in the reference image, and the object gets distorted.

Method	SF 1Tap $\mathcal{L}_{photo}$	SF 1Tap $\mathcal{L}_{ToF}$	MF 2Tap $\mathcal{L}_{photo}$	MF 2Tap $\mathcal{L}_{ToF}$
Input	70.39	23.71	93.17	11.98
Lindner	-	-	45.59	7.48
None	32.57	7.87	48.13	7.36
L_1	32.15	7.97	54.27	7.88
L_2	34.37	7.61	54.32	7.78
Cost	41.88	10.73	53.98	7.87
Cosine	**28.76**	**7.21**	**45.49**	**6.67**

Table 5. Ablation on different loss function for $\mathcal{L}_{sim}$, using FFN as backbone. On validation set.

the fact that without access to ground truth flows, such cases present a local minima during training.

Moreover, while we demonstrated our method on the largest available iToF dataset [23], this work is restricted to a synthetic setting as no real world data set containing raw iToF measurements is currently available. Lastly, the choice of the backbone network impacts the performance in different settings, *i.e*. MOM clearly outperforms the FFN backbone in the multi frequency single tap setting. Additionally, as our contribution is a training algorithm, the execution time is given by the execution time of the underlying OF network, while it is almost constant in the different settings for the MOM network, it grows linearly with the number of predicted flows for the FFN. As a consequence it would be desirable to have a OF network for iToF motion correction with a constant high performance in this multi-modality multi-frame flow prediction problem.

7. Conclusion

In this work, we presented a training method for OF networks to align iToF measurements in order to reduce the motion artifacts in the reconstructed depth images. To this end we enable the weakly supervised training on the ToF loss $\mathcal{L}_{ToF}$ using a phase unwrapping scheme for gradient correction. In combination with the regularizing losses $\mathcal{L}_{smooth}$ and $\mathcal{L}_{edge}$ which regulate the flow predictions, and the similarity loss $\mathcal{L}_{sim}$ to resolve the multi-modality, our method enables training without the need of ground truth flow labels. The experiments indicate that our method is able to compensate motion artifacts for both single and multi frequency settings as well as single and multi tap sensors. Further, our training method was demonstrated for two backbone OF networks, with different architectures, and was able to outperform existing methods.

8. Acknowledgements

This project was financed by the Baden-Württemberg Stiftung gGmbH.

References

[1] Gianluca Agresti and Pietro Zanuttigh. Deep learning for multi-path error removal in ToF sensors. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018.

[2] Filippo Aleotti, Matteo Poggi, and Stefano Mattoccia. Learning optical flow from still images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15201–15211, 2021.

[3] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018.

[4] Enrico Buratto, Adriano Simonetto, Gianluca Agresti, Henrik Schäfer, and Pietro Zanuttigh. Deep learning for transient image reconstruction from ToF data. *Sensors*, 21(6):1962, 2021.

[5] Faquan Chen, Rendong Ying, Jianwei Xue, Fei Wen, and Peilin Liu. A configurable and real-time multi-frequency 3D image signal processor for indirect time-of-flight sensors. *IEEE Sensors Journal*, 22(8):7834–7845, 2022.

[6] Guanting Dong, Yueyi Zhang, and Zhiwei Xiong. Spatial hierarchy aware residual pyramid network for time-of-flight depth denoising. In *European Conference on Computer Vision*, pages 35–50. Springer, 2020.

[7] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2758–2766, 2015.

[8] Jens-Malte Gottfried, Rahul Nair, Stephan Meister, Christoph S Garbe, and Daniel Kondermann. Time of flight motion compensation revisited. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 5861–5865. IEEE, 2014.

[9] Qi Guo, Iuri Frosio, Orazio Gallo, Todd Zickler, and Jan Kautz. Tackling 3D ToF artifacts through learning and the FLAT dataset. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 368–383, 2018.

[10] Felipe Gutierrez-Barragan, Huaijin Chen, Mohit Gupta, Andreas Velten, and Jinwei Gu. iToF2dToF: A robust and flexible representation for data-driven time-of-flight imaging. *IEEE Transactions on Computational Imaging*, 7:1205–1214, 2021.

[11] Miles Hansard, Seungkyu Lee, Ouk Choi, and Radu Patrice Horaud. *Time-of-flight cameras: principles, methods and applications*. Springer Science & Business Media, 2012.

[12] Thomas Hoegg, Damien Lefloch, and Andreas Kolb. Real-time motion artifact compensation for PMD-ToF images. In *Time-of-flight and depth imaging. Sensors, algorithms, and applications*, pages 273–288. Springer, 2013.

[13] Joel Janai, Fatma Guney, Anurag Ranjan, Michael Black, and Andreas Geiger. Unsupervised learning of multi-frame optical flow with occlusions. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 690–706, 2018.

[14] Rico Jonschkowski, Austin Stone, Jonathan T Barron, Ariel Gordon, Kurt Konolige, and Anelia Angelova. What matters in unsupervised optical flow. *arXiv preprint arXiv:2006.04902*, 2020.

[15] Min-Sun Keel, Young-Gu Jin, Youngchan Kim, Daeyun Kim, Yeomyung Kim, Myunghan Bae, Bumsik Chung, Sooho Son, Hogyun Kim, Taemin An, et al. A VGA indirect time-of-flight CMOS image sensor with 4-tap 7-μ m global-shutter pixel and fixed-pattern phase noise self-compensation. *IEEE Journal of Solid-State Circuits*, 55(4):889–897, 2019.

[16] Lingtong Kong, Chunhua Shen, and Jie Yang. FastFlowNet: A lightweight network for fast optical flow estimation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10310–10316. IEEE, 2021.

[17] Ruoteng Li, Robby T Tan, Loong-Fah Cheong, Angelica I Aviles-Rivero, Qingnan Fan, and Carola-Bibiane Schonlieb. Rainflow: Optical flow under rain streaks and rain veiling effect. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7304–7313, 2019.

[18] Marvin Lindner and Andreas Kolb. Compensation of motion artifacts for time-of-flight cameras. In *Workshop on Dynamic 3D Imaging*, pages 16–27. Springer, 2009.

[19] Oliver Lottner, Arnd Sluiter, Klaus Hartmann, and Wolfgang Weihs. Movement artefacts in range images of time-of-flight cameras. In *2007 International Symposium on Signals, Circuits and Systems*, volume 1, pages 1–4. IEEE, 2007.

[20] Julio Marco, Quercus Hernandez, Adolfo Munoz, Yue Dong, Adrian Jarabo, Min H Kim, Xin Tong, and Diego Gutierrez. DeepToF: off-the-shelf real-time correction of multipath interference in time-of-flight imaging. *ACM Transactions on Graphics (ToG)*, 36(6):1–12, 2017.

[21] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[22] Zhe Ren, Junchi Yan, Bingbing Ni, Bin Liu, Xiaokang Yang, and Hongyuan Zha. Unsupervised deep learning for optical flow estimation. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[23] Michael Schelling, Pedro Hermosilla, and Timo Ropinski. RADU: Ray-aligned depth update convolutions for ToF data denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 671–680, 2022.

[24] Mirko Schmidt. *Analysis, modeling and dynamic optimization of 3D time-of-flight imaging systems*. PhD thesis, 2011.

[25] Shuochen Su, Felix Heide, Gordon Wetzstein, and Wolfgang Heidrich. Deep end-to-end time-of-flight imaging. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6383–6392, 2018.

[26] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan 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.

[27] Lingzhu Xiang, Florian Echtler, Christian Kerl, Thiemo Wiedemeyer, Lars, hanyazou, Ryan Gordon, Francisco Facioni, laborer2008, Rich Wareham, Matthias Goldhoorn, alberth, gaborpapp, Steffen Fuchs, jmtatsch, Joshua Blake, Federico, Henning Jungkurth, Yuan Mingze, vinouz, Dave Coleman, Brendan Burns, Rahul Rawat, Serguei Mokhov, Paul Reynolds, P.E. Viau, Matthieu Fraissinet-Tachet, Ludique, James Billingham, and Alistair. libfreenect2: Release 0.2, Apr. 2016.

[28] Jason J Yu, Adam W Harley, and Konstantinos G Derpanis. Back to basics: Unsupervised learning of optical flow via brightness constancy and motion smoothness. In *European Conference on Computer Vision*, pages 3–10. Springer, 2016.

[29] Yinqiang Zheng, Mingfang Zhang, and Feng Lu. Optical flow in the dark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6749–6757, 2020.