A comparative review of optical flow estimation methods for computer-generated holograms

Nabil Madali, Antonin Gilles, Patrick Gioia, Luce Morin

A comparative review of optical flow estimation methods for computer-generated holograms.

Nabil Madali[1,3,*] Antonin Gilles[1] Patrick Gioia[1,2] Luce Morin[1,3]

[1] IRT b<>com Cesson-Sévigné France
[2] Orange Labs Rennes France
[3] INSA Rennes Rennes France

Abstract

Accurately estimating 3D optical flow in computer-generated holography poses a challenge due to the scrambling of 3D scene information during hologram acquisition. Therefore, to estimate the scene motion between consecutive frames, the scene geometry should be recovered first. Recent studies have demonstrated that a 3D RGB-D representation can be extracted from an input hologram with relatively low error under well-chosen numerical reconstruction parameters. However, limited attention has been given to how the produced error can impact the flow estimation algorithms. Therefore, in this study, we evaluate different learning/non-learning methodologies for recovering 3D scene geometry. Next, we analyze the types of distortions produced by these methods and attempt to minimize estimation error using spatial and temporal constraints. Finally, we compare the performance of several state-of-the-art methods to estimate the 3D optical flow vectors on the recovered sequence of RGB-D images.

Keywords : 3D Imaging, Holography, Depth Estimation, Optical Flow.

1 Introduction

In traditional video compression frameworks, optical flow algorithms [1] are used to analyze the movement of objects between consecutive frames to identify areas that remain static through the video sequence. These areas contain redundant information that can be stored once in the encoder part to significantly reduce the amount of data needed to represent the video while maintaining reasonable video quality. The motion vectors predicted by the optical flow algorithms are usually estimated by assuming that if a block of pixels moves from one frame to the next, the intensity values within the block should remain relatively constant. This assumption referred to as the brightness constancy assumption forms the basis of many optical flow estimation techniques, including block matching [2] which is a simple yet effective method for flow field estimation. There are different variants of the block-matching approach, which in essence are based on dividing a reference frame into small blocks and then finding the corresponding blocks in the target frame using a search window and a matching criterion.

Unfortunately, those optical flow methods such as the block-matching algorithm can not be directly applied to holographic video frames due to the violation of the brightness constancy assumption [3, 4]. In fact, during the hologram recording each scene point contributes to all the hologram pixels, thus losing the pixel-to-pixel relationship that provided well-localised scene objects inside the image plane and the brightness constancy assumption in natural images. In addition, as the scene moves in three different directions(namely in the X,Y, and Z axis), a third motion vector along the axial axis has to be estimated. This vector can not be estimated directly from the hologram and requires the knowledge of the used scene geometry.

In the case of a planer scene, i.e. a scene that is relatively flat and located at a distance d from the hologram plane, its geometry can be retrieved using Depth-from-Focus (DFF) approach with relatively high accuracy. The DFF is usually used in Digital Holographic Microscopy (DHM) [5] and consists of numerical propagating the hologram at different axial distances and then selecting the distance where the level of focus is maximum as an estimate of the scene position. The selected distance is commonly referred to as the *optimal focus plane distance* in the literature.

The real challenge of the DFF approach comes when dealing with complex scenes with a rapid surface variation and a large depth extent as is often the case for Computer Generated Hologram (CGH). In this scenario, the estimated scene geometry could be highly degraded and thus potentially affect subsequent tasks

*This work has been achieved within the Research and Technology Institute b<>com, dedicated to digital technologies. It has been funded by the French government through the National Research Agency (ANR) Investment referenced ANR-A0-AIRT-07. Corresponding author: nabil.madali@b-com.com.

such as motion estimation. In the present work, we analyze how we can efficiently retrieve the scene geometry using both learning-based and no-learning-based methods. Then, we assess how a badly retrieved scene geometry affects the 3D motion field estimation.

The remaining of the article is organized as follows: Section 2 discusses the related works. Section 3 details the different components of the proposed method. Then, in Section 4 a series of experiments are conducted to validate the proposed approach. Finally, in Section 5, we discuss the advantages and limitations of the presented method.

2 Related works

The present section discussed the related works for scene geometry retrieval and optical flow estimation.

2.1 Auto Focusing Methods

In holography, auto-focusing is a procedure to automatically retrieve the optimal focus plane distance from the input hologram. In the early days, this procedure was performed by numerical propagating the hologram at different candidate axial distances. Then, in selecting the distance where the amplitude of the numerical reconstruction has the highest focus level using Focus Measure (FM). In the literature, different measures have been proposed from which we can cite: the self-entropy [6], the amplitude modulus [7], the spectral ℓ_1 norm [8], the gray level-variance [9], and the Fresnelet-sparsity criterion [10].

With the emergence of deep neural architecture, traditional focus measure operators have been progressively replaced with learned Convolutional Neural Network (CNN) models [11]. The first works on the subjects initially attempt to predict the relative offset between the input numerical reconstruction and the optimal focus plane distance either using a regression or a classification head [12]. This approach is interesting in two aspects: First, it helps to generate large datasets, by propagating the input hologram at random reconstruction distance. Second, it is well-posed as the CNN has to learn the adequate filters to quantify the level of focus and then regress the addition offset to increase the level of focus. Subsequent works remove the need for numerical reconstruction and directly predict the optimal focus distance from the input hologram [13, 14]. To alleviate the lack of a large dataset, data augmentation [15] including random crop, resizing, and horizontal and vertical flips are used.

In the more recent approaches [16, 17, 18], a CNN model with a U-shape [19] architecture is used to predict different modalities (such as amplitude and phase) of the in-focus numerical reconstruction directly from the hologram.

Those approaches are effective when a single optimal focus distance is required for the hologram. Through training, the CNN can extract local features using a set of convolution layers. Then, merge them into a single global feature vector to regress the correct focus distance. However, when a per-pixel focus distance is required for each scene point, using a single global features vector is not effective. In fact, during the hologram recording each scene points contribute to all the hologram pixels, therefore the extracted local features from the convolution layers are related to all the scene points and cannot be used to perform the pixel-to-pixel estimation.

To effectively retrieve the geometry of a complex scene, Madali et al. proposed to detect the optimal focus plane locally from the 3D hologram reconstruction volume. The author proposed either using the learning-free [20] DFF method or using learned-based models that segment the in-focus area horizontally [21] or vertically [22]. The three cited methods will be used in the present work and will be described in more detail in the next section.

2.2 Optical Flows Methods

A pioneer work in deep optical flow estimation [23] was proposed by Fischer etl. [24], which introduced the FlowNetS and FlowNetC network architectures. FlowNetS is a simple U-Net [19], that takes as input the concatenation of the two video frames recorded at timestamp t, and $t+1$. Then, directly predicts an estimate of the optical flow between the two estimates. In contrast, FlowNetC employs a more sophisticated approach, that extracts local image features through the U-Net contracting path. Then, a correlation layer is applied to the obtained feature maps to evaluate the local similarity between the two images. Finally, the expanding path is used to progressively predict the final optical flow, utilizing both the acquired similarity score and the intermediate feature maps predicted by the contracting path on the image at timestamp t. For both networks, the authors choose to predict a flow field that is 4 times smaller than the input. Then, use an additional variational approach [25] to bring the flow field to the full resolution. The experiment results showed no significant performance difference between the two network architectures.

In subsequent work, Ilg et al. [26] enhanced the original FlowNet architectures in several key aspects, and proposed the FlowNet 2.0 architecture. First, they used a longer learning rate scheduler during network training

to significantly improve the overall performance. Second, the authors highlighted the significance of the order in which network training and fine-tuning are executed on the achieved results and proposed an optimal procedure. Third, the authors proposed the use of a stacked FlowNet model, where the initial model predicts an initial flow and subsequent networks refine this initial estimate. To guide these networks to concentrate on areas with substantial errors, the two input images, the wrapped image, and the estimation error of the warping are provided. The optimal configuration involves employing a FlowNetC as a baseline, followed by two other FlowNetS models, resulting in the creation of a comprehensive model known as FlowNet-CSS. Fourth, the author proposed a modified version of the FlowNetS model to handle the small displacement called FlowNet-Sd. The final network called FlowNet 2.0, takes as input the output from the FlowNet-CSS and the FlowNet-Sd networks, then uses an additional fusion network to fuse to results and produce the final optical flow.

Sun et al. [27] highlighted the large parametrization of the FlowNet 2.0 model that can lead to overfitting. In addition to the model large memory footprint that prevents inference on embedded devices. To alleviate those drawbacks, the authors proposed a compact CNN model called PWC-Net which is about 17 times smaller in size, and 2 times faster in inference than FlowNet 2.0. The proposed network takes as input the two video frames and uses a Feature Pyramid Network (FPN) [28] to extract a multiscale feature representation of the inputs. Then, at each scale, the predicted optical flow at the lower scale is up-sampled and is used to warp features of the second image toward the first image. The obtained warped features are correlated with the first image features using a cost volume layer to asses the pixel-wise correlation between the two feature maps. Finally, the output from the cost volume, the features of the first image, and the up-sampled optical flow are used as input to a multi-layer CNN to predict the optical flow at the current scale.

In the same year, Hui et al. [29] proposed LiteFlowNet which is designed as a lightweight network that approaches the performance of FlowNet 2.0. The proposed network has a similar architecture to PWC-Net, with a coarse to a fine framework, a learnable FPN as an encoder, and the use of features map wrapping layers. The main contribution of the paper resides in two points. First the use of a cascaded flow inference, which is coarse to fine flow estimation at each scale of the pyramid. The authors estimate a coarse flow, which is further refined to improve its sub-pixel accuracy. Second, the author introduced a Feature-driven Local Convolution (f-lcon) to smooth the flow field and thus attenuate undesired artifacts along the image boundaries.

Hur et al. [30] proposed to reformulate flow estimation as an Iterative Residual Refinement (IRR) procedure. The main idea is that the networks are no longer trained to directly predict the optical flow but rather to predict a residual, which is subsequently added to an initial flow estimate. The author applied this new formulation to the FlowNet and PWC-Net and simplified their architectures using a single sharable encoder-decoder network that predicts the optical flow residual. In addition to this fundamental change, the authors introduced several enhancements to the models. These include the incorporation of an additional decoder specifically designed for occlusion prediction, and the implementation of bi-directional estimation, where the network is supervised to predict optical flows in both directions (from image 1 to image 2 and vice versa). A learned bilateral filter was introduced to mitigate blurry estimations along boundaries. Finally, to address resolution concerns, an additional up-sampling layer was included to scale up the predicted flow and occlusion, bringing them to the original image resolution instead of retaining them at a quarter of the resolution.

Zhao et al. [31] pointed out that using feature or image-wrapping layers tends to cause a ghosting effect in the occluded areas, which can confuse the network during flow inference. To alleviate this issue, the author proposes a learnable occlusion mask which is applied after feature wrapping to remove unnecessary information. Then, those areas are filled with a learnable bias term that provides extra information at the masked areas. In addition, the author replace the wrapping operation with a deformable convolution [32], which has convolution filters parametrized by the learned optical flow. The proposed network is called MaskFlownet, which is based on the PWC-Net architecture.

Teed et al. [33] proposed a novel network architecture called RAFT to handle large and small displacements in a single lightweight network. The network architecture can be divided into three main stages: First, the two input images are fed to an encoder to extract local feature maps. Second, a 4-D correlation volume is constructed by correlating each pixel from the first image feature map with the pixels in the second image feature map. Third, the current optical flow is used to wrap the pixel from the first image to the second, and only the correlation values in a local neighborhood of the wrapped pixel are used to assess the wrapping error. Finally, the obtained correlation values with additional local context information are used to refine the current optical flow. These operations are repeated for several iterations, to get the final optical flow.

Jiang et al. [34] extended the RAFT architecture by adding a Global Motion Aggregation (GMA) module to handle occlusion. The GMA module is used to capture inter-frame similarity using a self-attention layer with an additional positional encoding. The computed features are used to have a homogeneous motion prediction in areas with similar textures.

More recently Huang et al. [35] proposed FlowFormer, a state-of-the-art optical flow network architecture based on Transformers [36]. The network inner working can be decomposed into four steps. First, the local feature maps are extracted from the input images using a CNN backbone network. Then, a 4D cost volume is computed using pixel-wise correlation between the pixels of the computed feature maps. Second, a patchification

t]

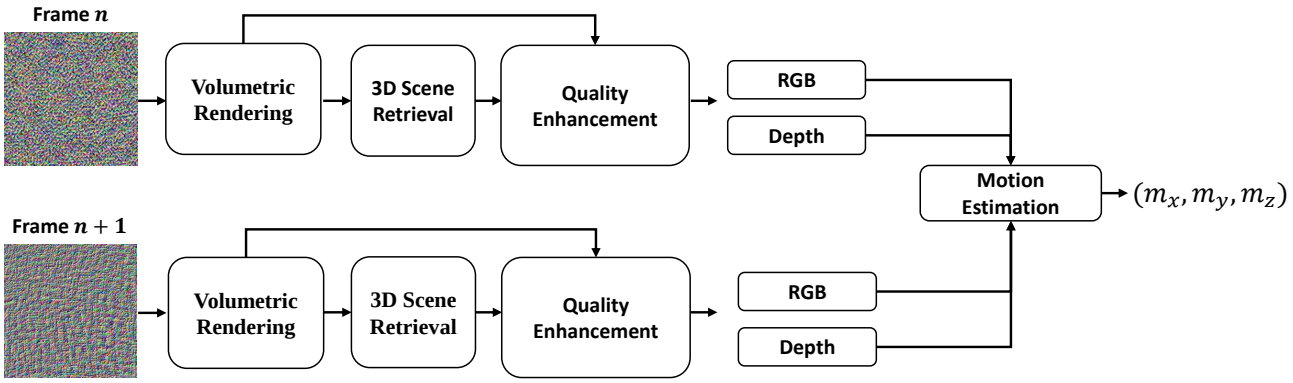


Figure 1: Block diagram representing the different steps of the proposed method.

layer is used on the 2D correlation matrix of each pixel, to group local information and thus to reduce its spatial resolution. Then, a compact representation of the cost patch is extracted by summarizing the obtained patches into k latent vectors. Thid, an Alternate-Group Transformer Layer (AGTL) is used to propagate the information inner and across the k latent vectors of the source pixels. Finally, the optical flow is predicted similarly to the RAFT network using an iterative refinement approach with a recurrent attention decoder layer.

3 Methodology

Our proposed approach comprises four steps, depicted in Figure 1. First, a 3D volumetric rendering is computed by numerical reconstructing the input hologram H at different axial distances. Second, an initial scene depth map is computed by optimizing the level focus locally using a small neighborhood centered around each pixel of the spatial domain. Third, the initially predicted depth map is refined using a voting agreement method, and an extended focus RGB image is extracted. Finally, 3D flow field vectors are computed for each scene point using a state-of-the-art optical flow method.

3.1 Volumetric Rendering

The first step of our proposed methodology is to construct a 3D volumetric rendering of the targeted scene from the input hologram. Here, we assume that a user-defined scene search interval bounded by a minimal $z_{\min}$ and maximal $z_{\max}$ axial distance is given. Then, a set of numerical reconstructions are computed using the Angular Spectrum Method (ASM) [37] at a uniformly sampled axial distance inside the search interval, as follows:

$$I_{z_i}\{H\}(x,y) = \left|\mathcal{F}^{-1}\left\{\mathcal{F}(H)e^{j2\pi z_i\sqrt{\lambda^{-2}-f_x^2-f_y^2}}\right\}\right|(x,y), \tag{1}$$

where $\mathcal{F}$ denotes the 2D Fourier transform, f_x and f_y are the spatial frequency along the X and Y direction of the hologram plane, λ is the acquisition wavelength, and z_i is the sampled axial distance defined as:

$$z_i = \frac{z_{\max} - z_{\min}}{N} i + z_{\min}. \tag{2}$$

For each computed numerical reconstruction I_{z_i}, only the part of the reconstruction plane that intersects the scene will be in focus. Therefore, recovering the scene geometry is similar to filtering the in-focus from out-of-focus areas on each computed numerical reconstruction.

3.2 3D Scene Retrieval

The second phase of the proposed methodology involves obtaining an initial scene depth map through the optimization of the local focus change. To achieve this, three different approaches proposed by our research group will be evaluated.

3.2.1 Learning Free Focus Measure

The first approach [20] does not require any extensive model pretraining and only requires a predefined mathematical FM operator to quantify the change of the focus level. Here, as our objective is to retrieve a dense scene depth map that associates for each pixel (m,n) of the spatial domain $\mathcal{X}\times\mathcal{Y}$ an estimated axial distance,

we will apply the selected FM locally around each pixel rather than globally as it is often the case in DHM. To achieve this, the spatial domain is first decomposed into a set of overlapped patches centered around each pixel of the spatial domain. Then, the axial axis is scanned to find at which axial distance, the focus level is locally optimal.

The definition of "optimal" will depend on the type of surface texture present in the local neighborhood around the pixel (m, n). For highly textured surfaces, the optimal focus plane is located at the global maximum of the focus curve. Conversely, in regions with uniformly textured surfaces, the optimal focus plane is located at the global minimum of the focus curve. To automatically switch between the global maximum and minimum, we proposed in our previous work [20] to select the axial distance that highly deviates from the mean focus values. Mathematically, this can be expressed as follows:

$$d(m,n) = \arg\max_i \left\{ \left| \bar{d} - FM(R_{i,m,n}) \right| \right\}, \qquad \bar{d} = \frac{1}{N} \sum_{i=1}^{N} FM(R_{i,m,n}). \tag{3}$$

where d is the resulting depth, FM is an $\mathbb{R}^2$ to $\mathbb{R}$ mapping that asses the level of focus, and $\mathcal{N}$ is the local neighborhood of size $s \times s$ around the pixel, defined as:

$$R_{i,m,n}(u,v) = I_{z_i}(m + u - s/2, n + v - s/2). \tag{4}$$

3.2.2 Learning Based Focus Measure

In the second approach [21], the used FM in Eq. 3 is replaced by a learned CNN model called "PatchNet" which is supervised to segment only the in-focus pixels from the input amplitude image. This model can not only predict at which axial distance the local focus is optimal but also where in the image there are the in-focus pixels. In addition to the increase in accuracy, the inference time is also greatly improved as the learned model can be directly applied to the whole numerical reconstruction without requiring a patch decomposition of the spatial domain. Mathematically, the second approach can be formulated as:

$$d(m,n) = \arg\max_i \{\mathcal{U}\{I_{z_i}\}(m,n)\}, \tag{5}$$

where $\mathcal{U}$ is a U-shaped CNN network [19].

3.2.3 Horizontal Volume Segmentation

The third approach [22] is similar to the second one, the only change relies on the fact that the in-focus pixels are segmented horizontally rather than vertically. This new formulation called "H-seg" helps to decrease the GPU consumption and to increase the speed of the inference time. In addition, there is less discontinuity in the final depth map as it could appear in the second approach. Mathematically the third approach can be formulated as

$$d(m,n) = \arg\max_z \{\mathcal{G}(S_m)(z_i,n)\}. \tag{6}$$

where $\mathcal{G}$ is a U-shape CNN, and $S_m(z_i, n)$ are 2D slices extracted along the Y axis, as follows:

$$S_m(z_i,n) = I_{z_i}(m,n). \tag{7}$$

3.3 Quality Enhancement

The third step of our methodology is to refine the initially predicted depth map to reduce the discontinuity and the noise level. The used method is the voting agreement [20] procedure, which assigns for each pixel (m, n) the axial distance with the highest consensus score inside its local neighborhood. The consensus can be seen as the level of certainty about the predicted axial distance and is simply defined as the number of times it has been predicted in the local neighborhood around the pixel. Mathematically the refined depth map can be given as:

$$d_{ref}(m,n) = \arg\max_{z_i} Score(z_i). \tag{8}$$

where $Score(z_i)$ is the consensus score for the candidate axial distance z_i given as,

$$Score(z_i) = \sum_{(x,y)\in\mathcal{N}_{m,n}} \mathbb{1}_{d(x,y)=z_i}. \tag{9}$$

3.3.1 Extended Focus Image

The refined depth map is used to compute an Extended Focus Image (EFI), by assigning for each pixel (m, n) the intensity of the numerical reconstruction at the estimated depth. More formally:

$$EFI(m,n) = I_{d_{ref}(m,n)}(m,n) \tag{10}$$

The refined depth map and EFI represent the final RGB-D scene estimate of the proposed approach.

3.4 Motion Estimation

In the fourth step, the RGB-D scene representations are extracted for two consecutive holographic frames H_t and H_{t+1} using one of the three previously discussed approaches. Then, an optical flow algorithm denoted $\mathcal{A}$ is used to compute the spatial motion vectors of the scene using the computed RGB images as follows:

$$m_x, m_y = \mathcal{A}\left(RGB_t, RGB_{t+1}\right). \tag{11}$$

Finally, the estimated motion vectors are used to compensate for the spatial motion on the computed depth map at timestamp t, and the result is subtracted from the depth map at timestamp $t+1$, as follows:

$$m_z(x,y) = D_{t+1}(x,y) - D_t(x+m_x, y+m_y). \tag{12}$$

The obtained 3D motion vector m_x, m_y, m_y can be used to compensate for the scene motion between the two timestamps.

4 Experimentation

The following section discusses the conducted experiments to validate the proposed approach for holographic scene motion estimation and the obtained results.

4.1 Experimental Setup

Dataset: The proposed methodology is evaluated on the IRT B-com open access dataset [38], which has been proposed for codec analysis and machine learning applications. The dataset includes six holographic video sequences, each one containing 300 frames recorded along a specific path. For each holographic frame, the dataset provided the used scene geometry in RGB-D format, the recorded hologram, and the scene rigid motion between the current and the next timestamp.

Segmentation Models Pre-training: To train the PatchNet and the H-Seg networks, the dataset is decomposed into a training and test set. The training set contains the first 200 frames from the scenes: *Piano*, *Table*, *Woods*, and *Cars*. The test contains the remaining frames from the training scenes, in addition to the 300 frames of the unseen scenes namely the *Dices*, and the *Woods* scenes. In total 4×200 frames are used for training the models and $4 \times 100 + 2 \times 300$ frames are used to evaluate the models.

Focus Measures Operators: The performance of the pre-trained neural networks is compared to the learning free FM operators in terms of accuracy and computational times. Here, three FM operators that have performed best in our previous extensive study [20] have been chosen. The selected operators are the Image contrast (CONT), the Normalized Graylevel variance (GLVN), and the Thresholded absolute gradient (GRAT).

Optical Flow Models: In the present work, we utilize a FlowFormer model pre-trained on the Sintel final dataset to forecast spatial motion vectors. The model architecture and weights employed are sourced directly from the authors project GitHub repository. A comparative analysis is conducted between the pre-trained model and a simple block matching algorithm employing a block size of 16×16 and Mean Absolute Difference (MAD) as the matching criteria.

4.2 Evaluation Procedure

The evaluation of the proposed methodology is decomposed into two parts.

4.2.1 Qualitative Evaluation of The Retrieved Scene

The first part will be dedicated to the selection of the best method for scene geometry retrieval. The three proposed methods will be evaluated in terms of accuracy using the Mean Absolute Error (MAE) metric and in terms of inference time. The MAE metric is only calculated on the predicted depth maps as follows:

$$MAE = \frac{1}{|\mathcal{X} \times \mathcal{Y}|} \sum_{x,y \in \mathcal{X} \times \mathcal{Y}} |D(x,y) - \hat{D}(x,y)| \tag{13}$$

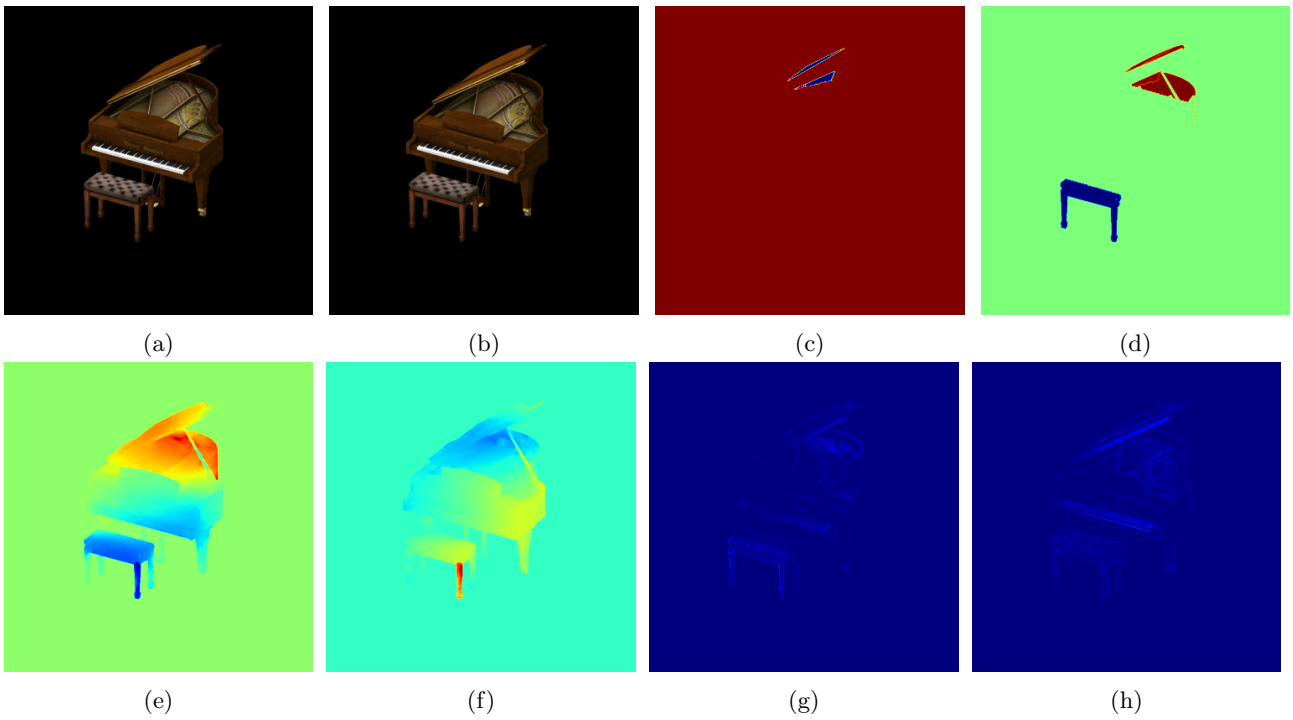


Figure 2: Illustration of the aperture problem when estimating optical flow. In (a), and (b) the ground truth amplitude maps at time frames t and $t+1$. In (b), and (c) the U and V components of the computed optical flow using the rigid body motion. In (d), and (e) the U and V components of the computed optical flow using a pre-trainned FlowFormer model. In (f), and (g) the obtained residual after the motion compensation using the computed and the predicted optical flow respectively.

where D and $\hat{D}$ are the ground truth and the predicted depth maps respectively. The RGB part is only used for visual quality assessment.

4.2.2 Evaluation of The Predicted Flow Field

In the second part, the selected method for scene geometry retrieval is applied to each frame of the test set to extract their RGB-D representations. Then, 3D flow field vectors will be extracted using SOTA optical flow methods and the result will be compared to the one extracted using the ground truth RGB-D representation using the Average Endpoint Error (AEE), which is defined as follows:

$$AEE(m_x, m_y) = \frac{1}{|\mathcal{X} \times \mathcal{Y}|} \sum_{(i,j) \in \mathcal{X} \times \mathcal{Y}} \sqrt{(m_x(i,j) - \hat{m}_x(i,j))^2 + (m_y(i,j) - \hat{m}_y(i,j))^2}, \tag{14}$$

where m_x, m_y and $\hat{m}_x, \hat{m}_y$ are the ground truth and the predicted motion vectors along the X and Y direction respectively. Here the ground truth motion vectors are predicted using the ground truth RGB-D format provided in the dataset. The rigid body transform which is associated with each frame, can not be used to estimate ground truth motion vectors due to the aperture problem [39]. Figure 2 is provided to illustrate this problem, in which the reader can observe that the motion vectors which has been estimated from the rigid body motion highly deviate from the one predicted using the pre-trained optical flow model. Therefore, the computed AEE will be high even though both result in a perfectly motion-compensated frame.

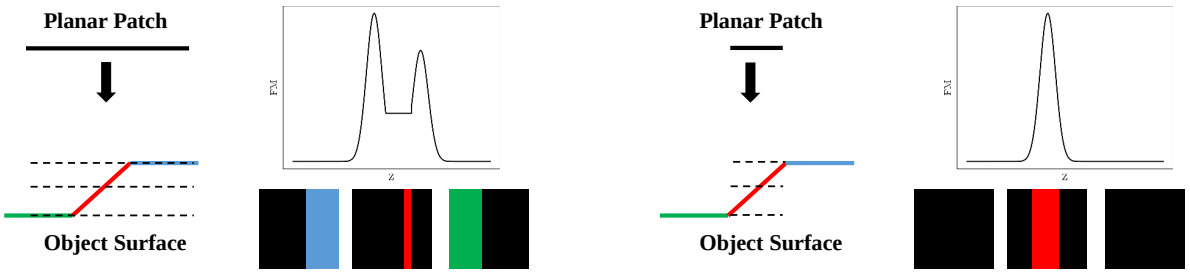
In addition, to the error of the spatial motion vectors, the axial motion vectors are also evaluated as follows:

$$AEE(m_z) = \frac{1}{|\mathcal{X} \times \mathcal{Y}|} \sum_{(x,y)\mathcal{X} \times \mathcal{Y}} |m_z(x,y) - \hat{m}_z(x,y)| \tag{15}$$

where m_z, $\hat{m}_z$ are the ground truth and the predicted motion vectors. Here, the evaluation is performed separately to better have an insight into the error propagation in both the spatial and the axial domain.

4.3 Results

Table 1 report the obtained MAE metrics and the computational time for the scene geometry retrieval part using the three different approaches on the test set. The readers can notice from the table that the learning-based



(a) Depth Map (b) RGB Map

Figure 3: Illustration of the geometrical setup for the DFF using different patch sizes.

	5×5	9×9	13×13	17×17	21×21
Learning Free Methods					
CONT	18.98	13.85	13.06	13.3	13.88
GLVN	33.4	14.66	11.21	10.58	10.75
GRAT	25.27	14.57	12.99	13.55	13.65
Learning Based Methods					
PatchNet			2.74		
H-seg			4.56		

(a)

	5×5	9×9	13×13	17×17	21×21
Learning Free Methods					
CONT	1049.81	1058.76	1077.49	1103.4	1136.24
GLVN	22.64	35.26	58.19	88.22	126.29
GRAT	30.24	55.69	101.38	160.35	235.39
Learning Based Methods					
PatchNet			26.05		
H-seg			14.30		

(b)

Table 1: The obtained MAE metric in Table 1a and the computation time in Table 1b for the initial depth estimate.

methods produce more accurate depth maps at a lower computation cost than the learning-free DFF approach. This performance gap can be explained by the geometrical setup of the problem formulation illustrated in Figure 3. In the DFF method, a planar patch with a fixed spatial position is moved along the axial axis to locate the distance where the focus level is maximum. The computed level of focus will depend on the mathematical formulation of FM and the size of the in-focus area inside the planar patch. The larger it is the greater the focus values. For a single planar patch, there could be multiple focus areas that will appear each time the planar patch intersects the surface of the object. Therefore, the higher the local object surface variation, the noisier the focus curve and the weaker the obtained performance. To smooth the focus curve a smaller patch can be used, that will intersect the surface of the object on a smaller area but will contain less texture to correctly asses the level of texture.

The learning-based methods have better performances due to their ability to automatically segment the in-focus pixels present on numerical reconstruction and then merge them into an independent area without requiring any adaptive patch size. A larger predefined patch size can be used to have enough texture to correctly asses the focus level and to automatically segments the in-focus pixel. The obtained results are smother and better approximates the local surface variation with high accuracy. Contrary to the DFF method which can only approximate a local mean surface height. Figure 4 illustrates an example of the obtained scene geometry using PatchNet and the GLVN methods. The readers can observe from the RGB part of the retrieved RGB-D scene geometry that for the GLVN method the obtained results are relatively noisy and lack details. This is because the method only approximates a mean surface height and can not approximate complex surface curvature. On the contrary, the PatchNet method produces a finer result with a low level of noise and clear details, which indicates that the surface geometry is retrieved correctly.

The performance difference between the H-seg and the PatchNet method regarding the performance reported in the original paper can be explained by the used train and test split. In the original paper, the test split is sampled inside the 300 frames, whereas in the following work, the test split is the last 100 frames of the training frames. This difference in split can cause a lack of generalization ability for the learned network due to unseen horizontal patterns during the training stage.

Table 2 report the obtained MAE metrics and the computational time after refining the initial estimate using different approaches. The readers can observe from the table that first the voting process gives better results than other refinement methods that tend to degrade the initial performance, with a degradation that increases with the used patch size. Second, we can observe that when the initial depth map is well estimated a lower patch size suffices to refine the depth map. On the contrary, when the initial depth map is highly degraded, a large patch size is required to correctly refine the initial estimate. This behaviour can be explained, by the fact that the smoothing operation tends to flatten the object surface. Therefore, if the predicted surface is highly curved and close to the ground truth surface as is the case for the PatchNet results, using a large smoothing patch size will significantly degrade the obtained MAE. On the contrary, the retrieved surface from the DFF

	5×5	9×9	13×13	17×17	21×21
Voting Agreement					
PatchNet	1.28	1.07	1.24	1.5	1.79
GLVN	11.74	10.72	10.01	9.43	8.93
Median Filter					
Patch Net	2.78	3.02	3.49	4.14	5.05
GLVN	12.07	11.82	11.92	12.26	12.8
Mean Filter					
Patch Net	6.19	9.23	12.13	14.9	17.55
GLVN	14.27	16.14	18.17	20.28	22.43
Gaussian Filter					
Patch Net	4.92	6.53	7.99	9.4	10.77
GLVN	13.4	14.27	15.14	16.02	16.97

(a)

	5×5	9×9	13×13	17×17	21×21
Voting Agreement	0.0178	0.0564	0.1238	0.2181	0.3369
Median Filter	0.0042	0.0268	0.0088	0.0091	0.009
Mean Filter	0.001	0.0008	0.0007	0.0009	0.0006
Gaussian Filter	0.0004	0.0004	0.0005	0.0005	0.0007

(b)

Table 2: The obtained MAE metric in Table 2a and the computation time in Table 2b for the refinement procedure.

	Piano	Table	Woods	Cars	Dices	Animals	Avg
Patch Net							
AEE	0.29	0.6	0.44	0.23	0.84	0.32	0.45
$\lvert m_z - \hat{m}_z \rvert$	5.6	7.0	3.62	1.63	5.44	3.22	4.42
GLVN							
AEE	0.78	1.98	0.65	0.48	0.74	0.73	0.89
$\lvert m_z - \hat{m}_z \rvert$	26.25	28.62	8.44	6.94	10.11	9.44	14.97

(a)

	Piano	Table	Woods	Cars	Dices	Animals	Avg
Patch Net							
$AEE(m_x, m_y)$	0.38	0.69	2.97	2.12	1.6	1.05	1.47
$AEE(m_z)$	4.27	4.44	2.86	1.26	3.55	1.33	2.95
GLVN							
$AEE(m_x, m_y)$	2.0	1.75	3.86	3.83	3.0	3.16	2.93
$AEE(m_z)$	23.75	24.79	7.59	6.36	8.66	6.52	12.94

(b)

Table 3: The obtained optical flow error using the FlowFormer model in Table 3b and using the Block-matching algorithm in Table 3a

method is relatively flat, and smoothing it using a large or small patch size will not significantly change its surface curvature and thus do not degrade the obtained MAE metric.

4.3.1 Optical Flow Prediction

In the second step of the evaluation, the refined RGB-D scene representations using both the PatchNet and the GLVN method have been used to evaluate the 3D motion field vectors using different deep optical flow models, and the results are reported in Table 3.

The reader can observe from the table that regardless of the optical flow method employed, the spatial motion vectors error nearly doubles when transitioning from the GLVN operator to the PatchNet model. This significant performance gap can be attributed to both the density and the random distribution of noise across different frames within the video sequence. As previously mentioned, the accuracy of scene geometry retrieval is heavily influenced by the changing orientation of the scene throughout the video sequence. At any given timestamp t, a segment of the scene might be relatively flat and directly facing the hologram plane, facilitating the scene geometry retrieval. However, at timestamp $t+1$, the same segment could be oriented at an angle θ from the hologram plane, making retrieval more challenging. Therefore the same scene part could be retrieved correctly at timestamp t without any noise in the corresponding texture and could be degraded at timestamp $t+1$ with a higher level of noise in the corresponding texture. This is problematic because the extracted features vector output from the convolution layer will be different for two retrieved textures due to noise contamination and thus can not be matched correctly.

Figure 5 provides an example of the retrieved RGB scene images using the PatchNet and the GLVN operator, and their respective optical flow estimation error. The readers can observe that the obtained RGB images from the GLVN operator are noisier than those provided by the PatchNet model. In addition, we can observe that the area where there is a significant noise level causes a higher optical flow error in both the U and V components.

The same analysis can be observed for the non-learning-based optical flow methods, where we can also notice an increase in the optical flow estimation error between the PatchNet and the GLVN model. In addition, we can notice a larger estimation error than the one obtained using the FlowFormer model, which can be explained by the size of the search area which could be larger than the magnitude of the optical flow vectors in the Sintel dataset.

The primary distinction lies in the matching criteria of optical flow models. In the FlowFormer, matching criteria involve comparing local feature vectors extracted after a set of convolution layers. Conversely, the block matching algorithm directly compares image intensity values using the MAD matching criteria. Employing local feature vectors can mitigate the randomness inherent in noise distribution. Indeed, feature vectors extracted

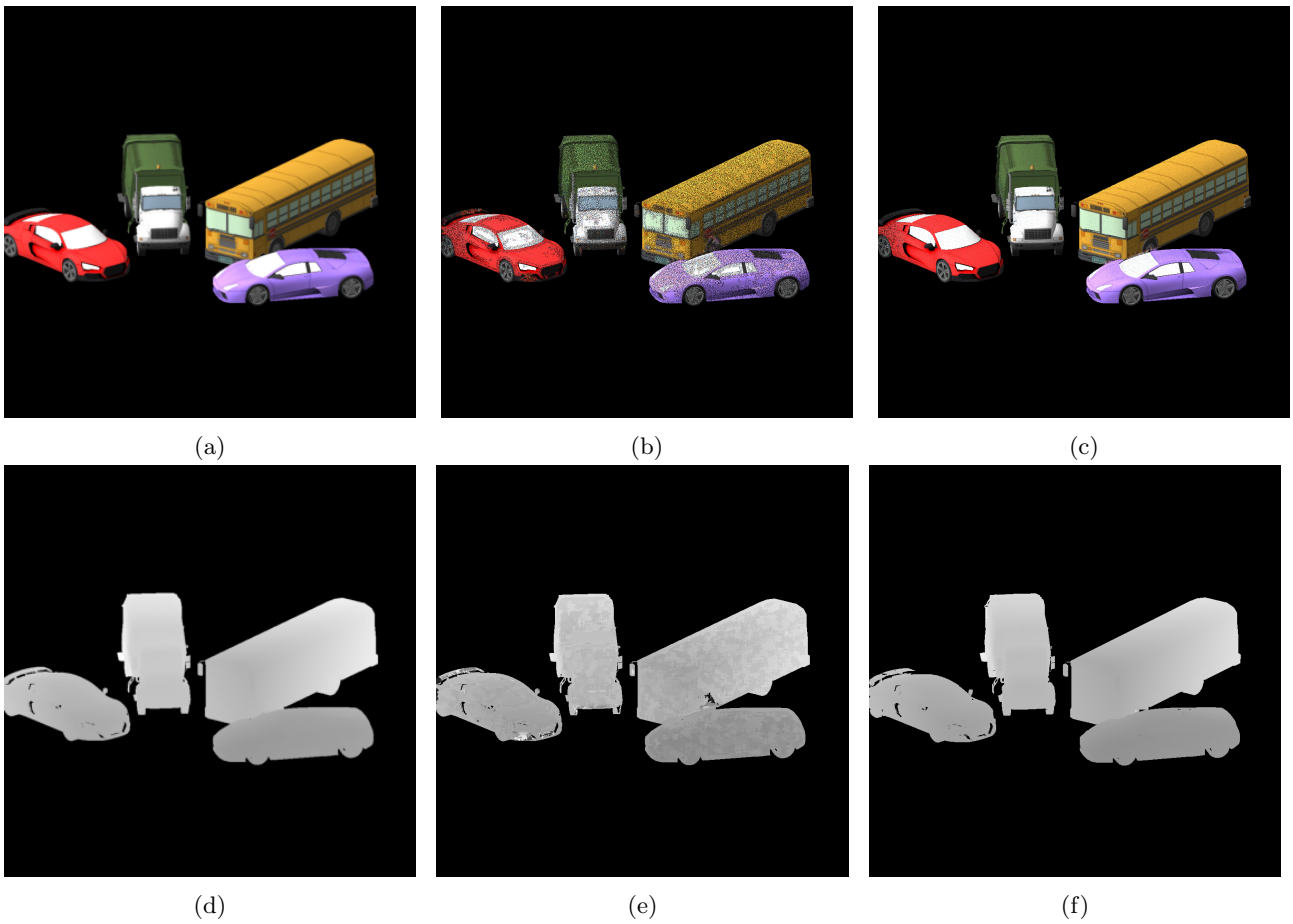


Figure 4: Illustration of the obtained RGB-D representation obtained using the PatchNet method in (b),(e) and the GLVN method in (c), (f) in comparison to the ground truth one in (a), (d).

from noise-free and noisy textures may be relatively close if noise levels are sufficiently low. However, the disparity between feature vectors amplifies with increased noise distribution.

5 Conclusion

In the present work, we have evaluated different models for retrieving the scene geometry from computer-generated holograms. Then, we evaluated how the obtained results affect the 3D flow field estimation.

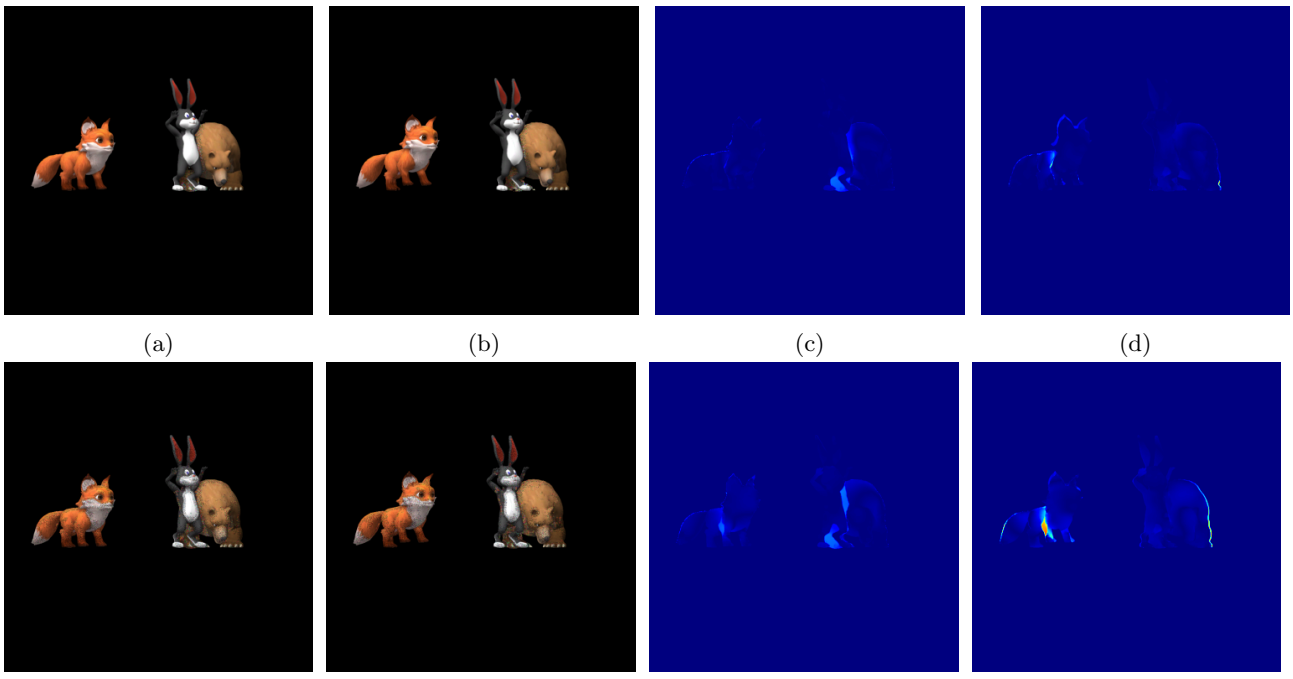
The conducted experiments allow us to conclude that the scene geometry estimation can be executed accurately. However, the retrieved scene across different timestamps may be susceptible to random degradation which appears as speckle noise in the retrieved RGB images. The randomness of the noise distribution can significantly impact the matching phase of the optical flow algorithm and potentially lead to erroneous in the estimated optical flow. To effectively extract consistent optical flow vectors, a denoising stage has to be implemented to only eliminates the noise present in the RGB images and thus ensure similar texture across different timestamps.

Future work will be dedicated to designing more directed approaches using the hologram phase space representation, to reduce not only the computation time but also the estimation uncertainty.

References

[1] M. Zhai, X. Xiang, N. Lv, and X. Kong, "Optical flow and scene flow estimation: A survey," *Pattern Recognition*, vol. 114, p. 107861, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320321000480

[2] M. H. Jakubowski and G. Pastuszak, "Block-based motion estimation algorithms — a survey," *Opto-Electronics Review*, vol. 21, pp. 86–102, 2013. [Online]. Available: https://api.semanticscholar.org/CorpusID:52251298



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 5: The first row of the figure provides the retrieved scene geometry at timestamp t and $t+1$ using the PatchNET model and the absolute optical flow error in the U and V direction. The second row provided the scene information but using the GLVN operator.

[3] T. Dong, K.-J. Oh, J. Park, and E. S. Jang, "Compression performance analysis of experimental holographic data coding systems," *Sensors*, vol. 23, no. 18, 2023. [Online]. Available: https://www.mdpi.com/1424-8220/23/18/7684

[4] P. Schelkens, A. Ahar, A. Gilles, R. K. Muhamad, T. J. Naughton, C. Perra, A. M. G. Pinheiro, P. Stepien, and M. Kujawińska, "Compression strategies for digital holograms in biomedical and multimedia applications," *Light: Advanced Manufacturing*, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:250288301

[5] J. T. Sheridan, R. K. Kostuk, A. F. Gil, Y. Wang, W. Lu, H. Zhong, Y. Tomita, C. Neipp, J. Francés, S. Gallego, I. Pascual, V. Marinova, S.-H. Lin, K.-Y. Hsu, F. Bruder, S. Hansen, C. Manecke, R. Meisenheimer, C. Rewitz, T. Rölle, S. Odinokov, O. Matoba, M. Kumar, X. Quan, Y. Awatsuji, P. W. Wachulak, A. V. Gorelaya, A. A. Sevryugin, E. V. Shalymov, V. Y. Venediktov, R. Chmelik, M. A. Ferrara, G. Coppola, A. Márquez, A. Beléndez, W. Yang, R. Yuste, A. Bianco, A. Zanutta, C. Falldorf, J. J. Healy, X. Fan, B. M. Hennelly, I. Zhurminsky, M. Schnieper, R. Ferrini, S. Fricke, G. Situ, H. Wang, A. S. Abdurashitov, V. V. Tuchin, N. V. Petrov, T. Nomura, D. R. Morim, and K. Saravanamuttu, "Roadmap on holography," *Journal of Optics*, vol. 22, no. 12, p. 123002, 2020, publisher: IOP Publishing. [Online]. Available: https://doi.org/10.1088/2040-8986/abb3a4

[6] J. Gillespie and R. King, "The use of self-entropy as a focus measure in digital holography," *Pattern Recognition Letters*, vol. 9, no. 1, pp. 19–25, 1989. [Online]. Available: https://www.sciencedirect.com/science/article/pii/016786558990024X

[7] F. Dubois, C. Schockaert, N. Callens, and C. Yourassowsky, "Focus plane detection criteria in digital holography microscopy by amplitude analysis," *Opt. Express*, vol. 14, no. 13, pp. 5895–5908, Jun 2006. [Online]. Available: https://opg.optica.org/oe/abstract.cfm?URI=oe-14-13-5895

[8] W. Li, N. C. Loomis, Q. Hu, and C. S. Davis, "Focus detection from digital in-line holograms based on spectral l1 norms," *J. Opt. Soc. Am. A*, vol. 24, no. 10, pp. 3054–3062, Oct 2007. [Online]. Available: https://opg.optica.org/josaa/abstract.cfm?URI=josaa-24-10-3054

[9] L. Ma, H. Wang, Y. Li, and H. Jin, "Numerical reconstruction of digital holograms for three-dimensional shape measurement," *Journal of Optics A: Pure and Applied Optics*, vol. 6, no. 4, p. 396, mar 2004. [Online]. Available: https://dx.doi.org/10.1088/1464-4258/6/4/016

[10] M. Liebling and M. Unser, "Autofocus for digital fresnel holograms by use of a fresnelet-sparsity criterion," *J. Opt. Soc. Am. A*, vol. 21, no. 12, pp. 2424–2430, Dec 2004. [Online]. Available: https://opg.optica.org/josaa/abstract.cfm?URI=josaa-21-12-2424

[11] T. Zeng, Y. Zhu, and E. Y. Lam, "Deep learning for digital holography: a review," *Opt. Express*, vol. 29, no. 24, pp. 40 572–40 593, Nov 2021. [Online]. Available: https://opg.optica.org/oe/abstract.cfm?URI=oe-29-24-40572

[12] T. Pitkäaho, A. Manninen, and T. J. Naughton, "Performance of Autofocus Capability of Deep Convolutional Neural Networks in Digital Holographic Microscopy," in *Digital Holography and Three-Dimensional Imaging*. JeJu Island: OSA, 2016, p. W2A.5. [Online]. Available: https://opg.optica.org/abstract.cfm?URI=DH-2017-W2A.5

[13] S. Cuenat, L. Andréoli, A. N. André, P. Sandoz, G. J. Laurent, R. Couturier, and M. Jacquot, "Fast autofocusing using tiny transformer networks for digital holographic microscopy," *Opt. Express*, vol. 30, no. 14, pp. 24 730–24 746, Jul 2022. [Online]. Available: https://opg.optica.org/oe/abstract.cfm?URI=oe-30-14-24730

[14] Z. Ren, Z. Xu, and E. Y. Lam, "Autofocusing in digital holography using deep learning," in *Three-Dimensional and Multidimensional Microscopy: Image Acquisition and Processing XXV*, T. G. Brown, C. J. Cogswell, and T. Wilson, Eds., vol. 10499, International Society for Optics and Photonics. SPIE, 2018, p. 104991V. [Online]. Available: https://doi.org/10.1117/12.2289282

[15] S. Yang, W. Xiao, M. Zhang, S. Guo, J. Zhao, and F. Shen, "Image data augmentation for deep learning: A survey," 2023.

[16] K. Wang, J. Dou, Q. Kemao, J. Di, and J. Zhao, "Y-net: a one-to-two deep learning framework for digital holographic reconstruction," *Opt. Lett.*, vol. 44, no. 19, pp. 4765–4768, Oct 2019. [Online]. Available: https://opg.optica.org/ol/abstract.cfm?URI=ol-44-19-4765

[17] Y. Rivenson, Y. Zhang, H. Gunaydin, D. Teng, and A. Ozcan, "Phase recovery and holographic image reconstruction using deep learning in neural networks," *CoRR*, vol. abs/1705.04286, 2017. [Online]. Available: http://arxiv.org/abs/1705.04286

[18] Z. Ren, Z. Xu, and E. Y. M. Lam, "End-to-end deep learning framework for digital holographic reconstruction," *Advanced Photonics*, vol. 1, no. 1, p. 016004, 2019. [Online]. Available: https://doi.org/10.1117/1.AP.1.1.016004

[19] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *CoRR*, vol. abs/1505.04597, 2015. [Online]. Available: http://arxiv.org/abs/1505.04597

[20] N. Madali, A. Gilles, P. Gioia, and L. Morin, "Automatic depth map retrieval from digital holograms using a depth-from-focus approach," *Applied Optics*, vol. 62, no. 10, p. D77, Apr. 2023. [Online]. Available: https://opg.optica.org/abstract.cfm?URI=ao-62-10-D77

[21] ——, "Automatic depth map retrieval from digital holograms using a deep learning approach," *Opt. Express*, vol. 31, no. 3, pp. 4199–4215, Jan 2023. [Online]. Available: https://opg.optica.org/oe/abstract.cfm?URI=oe-31-3-4199

[22] ——, "H-seg: a horizontal reconstruction volume segmentation method for accurate depth estimation in a computer-generated hologram," *Opt. Lett.*, vol. 48, no. 12, pp. 3195–3198, Jun 2023. [Online]. Available: https://opg.optica.org/ol/abstract.cfm?URI=ol-48-12-3195

[23] J. Hur and S. Roth, "Optical flow estimation in the deep learning age," *CoRR*, vol. abs/2004.02853, 2020. [Online]. Available: https://arxiv.org/abs/2004.02853

[24] P. Fischer, A. Dosovitskiy, E. Ilg, P. Häusser, C. Hazirbas, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox, "Flownet: Learning optical flow with convolutional networks," *CoRR*, vol. abs/1504.06852, 2015. [Online]. Available: http://arxiv.org/abs/1504.06852

[25] T. Brox and J. Malik, "Large displacement optical flow: Descriptor matching in variational motion estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 3, pp. 500–513, 2011.

[26] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "Flownet 2.0: Evolution of optical flow estimation with deep networks," *CoRR*, vol. abs/1612.01925, 2016. [Online]. Available: http://arxiv.org/abs/1612.01925

[27] D. Sun, X. Yang, M. Liu, and J. Kautz, "Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume," *CoRR*, vol. abs/1709.02371, 2017. [Online]. Available: http://arxiv.org/abs/1709.02371

[28] T. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, "Feature pyramid networks for object detection," *CoRR*, vol. abs/1612.03144, 2016. [Online]. Available: http://arxiv.org/abs/1612.03144

[29] T. Hui, X. Tang, and C. C. Loy, "Liteflownet: A lightweight convolutional neural network for optical flow estimation," *CoRR*, vol. abs/1805.07036, 2018. [Online]. Available: http://arxiv.org/abs/1805.07036

[30] J. Hur and S. Roth, "Iterative residual refinement for joint optical flow and occlusion estimation," *CoRR*, vol. abs/1904.05290, 2019. [Online]. Available: http://arxiv.org/abs/1904.05290

[31] S. Zhao, Y. Sheng, Y. Dong, E. I. Chang, and Y. Xu, "Maskflownet: Asymmetric feature matching with learnable occlusion mask," *CoRR*, vol. abs/2003.10955, 2020. [Online]. Available: https://arxiv.org/abs/2003.10955

[32] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," *CoRR*, vol. abs/1703.06211, 2017. [Online]. Available: http://arxiv.org/abs/1703.06211

[33] Z. Teed and J. Deng, "RAFT: recurrent all-pairs field transforms for optical flow," *CoRR*, vol. abs/2003.12039, 2020. [Online]. Available: https://arxiv.org/abs/2003.12039

[34] S. Jiang, D. Campbell, Y. Lu, H. Li, and R. I. Hartley, "Learning to estimate hidden motions with global motion aggregation," *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9752–9761, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:233033368

[35] Z. Huang, X. Shi, C. Zhang, Q. Wang, K. C. Cheung, H. Qin, J. Dai, and H. Li, "Flowformer: A transformer architecture for optical flow," *ArXiv*, vol. abs/2203.16194, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:247792986

[36] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *CoRR*, vol. abs/1706.03762, 2017. [Online]. Available: http://arxiv.org/abs/1706.03762

[37] J. W. Goodman, "Introduction to fourier optics," *Introduction to Fourier optics, 3rd ed., by JW Goodman. Englewood, CO: Roberts & Co. Publishers, 2005*, vol. 1, 2005.

[38] A. Gilles, P. Gioia, N. Madali, A. E. Rhammad, and L. Morin, "Open access dataset of holographic videos for codec analysis and machine learning applications," in *2023 15th International Conference on Quality of Multimedia Experience (QoMEX)*, 2023, pp. 258–263, ISSN: 2472-7814.

[39] T. Xue, H. Mobahi, F. Durand, and W. T. Freeman, "Refraction wiggles for measuring fluid depth and velocity from video," *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.