PSDFH: A Phase-Space-Based Depth from Hologram Extraction Method

Nabil Madali, Antonin Gilles, Patrick Gioia, Luce Morin

applied sciences

MDPI

Article

PSDFH: A Phase-Space-Based Depth from Hologram Extraction Method

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

[1] Institute of Research & Technology b-com, 35510 Cesson-Sévigné, France
[2] University Rennes, INSA Rennes, CNRS, IETR—UMR 6164, 35000 Rennes, France
[3] Orange Labs, 35510 Rennes, France
* Correspondence: nabil.madali@b-com.com

Abstract: Object pre-localization from computer-generated holograms is still an open problem in the current state of the art. In this work, we propose the use of the hologram phase space representation to determine a set of regions of interest where the searched object can be located. The extracted regions can be used to pre-locate the object in 3D space and are further refined to produce a more accurate depth estimate. An iterative refinement method is proposed for 1D holograms and is extended in a parsimonious version for 2D holograms. A series of experiments are conducted to assess the quality of the extracted regions of interest and the sparse depth estimate produced by the iterative refinement method. Experimental results show that it is possible to pre-localize the object in 3D space from the phase space representation and thus to improve the calculation time by reducing the number of operations and numerical reconstructions necessary for the application of s (DFF) methods. Using the proposed methodology, the time for the application of the DFF method is reduced by half, and the accuracy is increased by a factor of three.

Keywords: 3D imaging; holography; depth estimation; phase space; short-term Fourier transform; Fourier optics

Citation: Madali, N.; Gilles, A.; Gioia, P.; Morin, L. PSDFH: A Phase-Space-Based Depth from Hologram Extraction Method. *Appl. Sci.* **2023**, *13*, 2463. https://doi.org/10.3390/app13042463

Academic Editor: Andrés Márquez

Received: 6 January 2023
Revised: 9 February 2023
Accepted: 13 February 2023
Published: 14 February 2023

1. Introduction

Holography [1], introduced by Dennis Gabor in 1948, was originally a two-stage coherent imaging technology for recording and reconstructing the light wave scattered by an illuminated object through interference. While initially restricted to monochromatic still models captured in optics laboratories, its modern evolution into digital holography has recently allowed the processing of completely general content, such as color-animated hologram sequences.

A hologram contains the entire light field of the recorded object, including intensity and phase information. It therefore requires larger storage space, but this additional information can be used for a wider range of applications. In most computer vision applications, only the intensity image and depth map of the scene are used to completely describe the 3D scene. Both of them are contained in the hologram but require a preprocessing step in order to extract them. The simplest and most effective approach for this purpose is to use the *depth-from-focus (DFF)* method [2]. This technique starts from a reconstruction volume, built using several numerical reconstructions performed at different sampled depths. It then estimates a depth map of the recorded scene by detecting the reconstruction depth at which the focus is locally optimal using a *focus measure (FM)* for each pixel. Once the depth map has been estimated, the intensity map is constructed by assigning to each pixel the intensity present at the same position in the numerical reconstruction performed at the estimated depth.

Although the DFF approach gives convincing results, it has many limitations. First, it requires carefully choosing several hyperparameters, including the reconstruction interval, the number of reconstructions to be performed, the focus operator, and how it applies to the

reconstruction volume. Each of these factors can greatly affect the estimation performance. Second, this approach is computationally expensive, since it requires computing a large number of numerical reconstructions and in some particular cases pre-processing the hologram to remove the zero-order, twin image, and speckle noise from the reconstructed images, resulting in additional computational overhead.

In this work, we propose a PS-DFF (phase-space-based DFF), a *region of interest* (ROI) extraction procedure based on the phase space representation of holograms to pre-localize the scene in 3D space. The extracted regions can be used to select an appropriate numerical reconstruction interval, as well as the spatial location where the focus measure should be applied. In addition, an iterative method is proposed to refine the extracted regions of interest and produce a finer depth estimate. A series of experiments are conducted to evaluate the quality of the extracted regions of interest, the gain in computation time with respect to the classical DFF method, and the accuracy of the sparse depth estimate produced by the iterative refinement method.

The main contributions of this paper can be summarized as follows:

- A novel methodology to estimate a complex and deep depth map with multiple values based on the phase space representation of the hologram.
- An iterative voting process to extract regions containing scene objects (called ROIs in the paper) from the hologram phase space transform support.
- An ROI-constrained DFF method to improve speed and accuracy with respect to classical DFF.

The remainder of this article is organized as follows: Section 2 reviews previous works that use phase space to process holographic data. Then, Section 3 presents the proposed method. 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 proposed approach.

2. Related Works

In this section, we review the two main approaches that can be used to estimate the depth map from a digital hologram: depth-from-focus methods and phase-space-based methods.

2.1. Depth-From-Focus Methods

A simple depth estimation method is the use of the DFF method [3–5]. First, a reconstruction volume is built from the input hologram, with a predefined number of holographic reconstructions within a fixed depth interval. Each numerical reconstruction can be computed using the angular spectrum method [6] defined as

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

where H is the hologram, $\mathcal{F}$ is the Fourier transform, λ is the acquisition wavelength, f_x and f_y are the spatial frequencies along the X- and Y-axis, λ is the acquisition wavelength, and z_i is the reconstruction depth, given by

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

where N is the total number of holographic reconstructions, and $z_{\min}, z_{\max}$ are the boundaries of the reconstruction interval. Then, the focus level of each pixel (m, n) in the reconstruction volume is evaluated locally using a focus measure FM applied on a centered patch $R_{m,n}$ around the pixel. Finally, the pixel depth is set to the depth $d_{m,n}$ at which the maximum focus level $FM(R)$ is reached. More formally,

$$d_{m,n} = \underset{i\in[1,N]}{\arg\max}\{FM(R_{m,n,i})\}, \tag{3}$$

where

$$R_{m,n,i}(u,v) = |\mathcal{P}_{z_i}\{H\}|(m+u-s/2, n+v-s/2), \quad (4)$$

and s is the used patch size. Despite its time-consuming nature, the DFF method can produce relevant results under optimal conditions in which the holographic reconstruction parameters, focus measure, and patch sizes are carefully chosen.

In practice, determining the proper reconstruction parameters is difficult. It is simpler to use a very large reconstruction interval $[0, z]$ that spans the entire space and a high sampling rate N to cover all possible depths. Even if this methodology appears to be advantageous, the obtained results will depend on the focus measure FM, which must be sensitive enough to distinguish the correct optimal focus plane.

A detailed benchmark of focus measurement operators can be found in [7]. More recently, [8] studied various types of focus measurements and how different choices of hyperparameters can affect their performance when used in CGH. The results obtained demonstrate that the focusing curves are subject to a change in polarity depending on the region treated. Therefore, the authors propose an automatic switching method to automatically choose the global minimum or maximum depending on the offset from the mean value.

2.2. Phase-Space-Based Methods

One of the pioneer works on the extraction of depth information directly from a hologram without any numerical reconstruction was published in [3]. The proposed method is based on Wigner's analysis of a 1D cross section of 2D in-line holograms. Each particle that was used to record the hologram is represented by a cross-shaped pair of line impulses in Wigner's analysis of the hologram. The authors showed analytically that if one of the two straight lines slopes is known, the distance between the particle and the hologram plane can be obtained. The experiments were conducted using several holograms with a varying number from one to three particles that are spaced far enough. The obtained results confirmed the analytical formula and showed that the proposed approach can be used to obtain an accurate estimate of the particle depth.

Ozgen et al. [9] extended the previous work by replacing the Wigner–Ville distribution by the Cohen bilinear class with fixed kernels given by analytical expressions in order to improve the precision in the case of multiple particles. The proposed method was compared to a scaling of the chirp transformation [10] and showed better results in terms of automation of the algorithm and a better visual interpretation of the obtained results.

In the same spirit, Zhang et al. [11] used the Gabor transform to extract the depth information of small particles from in-line holograms by using the analytical relation between the Gabor angle and the object distance from the CCD camera. The experiments were performed on a 1D cross section of a 2D in-line hologram of three particles located at the same depth. The 1D cross section was analyzed using the Gabor transform, and the Gabor angle was calculated manually in order to obtain a depth estimate. The estimated depth is very close to the ground truth with an approximation error of 3 µm. The authors pointed out that the estimation error may be due to the high signal-to-noise ratio caused by the acquisition conditions.

Other applications based on the phase space transform are discussed in [12].

2.3. State-of-the-Art Limitations

As shown in this section, while depth-from-focus methods enable estimation of accurate depth maps from digital holograms, they require a set of hyperparameters to be carefully chosen. In addition, since multiple numerical reconstructions need to be computed, their computational complexity increases for scenes with a large depth range. On the other hand, phase-space-based methods do not require computing numerical reconstructions. Nevertheless, state-of-the-art works were limited to scenes composed of a small number of particles, and their application to complex scenes with a large depth range was never studied in the literature.

In this work, we propose using these two approaches together: we use a phase-space-based method to extract regions of interest (ROIs) from the input hologram, which are then exploited to determine the depth range to apply a DFF method to extract the scene depth map from the hologram.

3. Proposed Methodology

The following sections introduce the proposed method illustrated in Figure 1 for phase space depth estimation. The proposed methodology is designed for 1D holograms and is applied to 1D slices of 2D holograms.

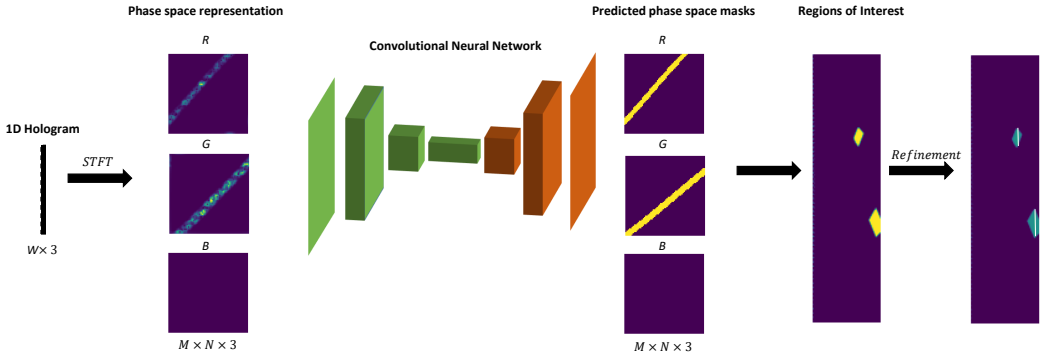

Figure 1. An illustration of the different components of the proposed method. First, the phase space representation of the input hologram is computed. Then, a neural network is used to predict phase space masks, which will be back-projected to extract regions of interest. Finally, the proposed refinement algorithm is used to extract the final depth estimate.

3.1. Overview

Our proposed methodology comprises 4 steps for 1D holograms and an additional step when processing 1D slices of 2D holograms. First, considering a 1D hologram obtained from a 2D scene, the hologram phase space representation is computed using the *short-term Fourier transform* (STFT), producing for each color channel a 2D matrix of size $N \times M$ corresponding to the axis (x, ξ). This matrix represents the decomposition of the hologram as local frequencies ξ around the specified position x. Since frequencies are related to light propagation through the grating equation, each couple (x, ξ) represents a light beam passing through x and oriented according to ξ. The phase space transform is a compact representation of the hologram and contains a sufficient amount of information to fully describe it.

Second, a neural network is used to estimate the support of the hologram phase space representation, resulting in a set of binary masks. The phase space segmentation makes it possible to consider only the shape of the representation created by the scene in phase space without taking into account the values of the transform, which are subject to variation from one hologram to another. Furthermore, by using this segmentation step, a unique thresholding value that is not affected by the color intensity or phase value of the input hologram can be defined.

Third, the predicted masks are used to detect regions of interest containing the scene objects. The probability for each 2D position to belong to the scene is estimated based on the binary masks, and points exceeding a given threshold define the region of interest.

Finally, the initial regions of interest are refined using an iterative method to reduce their depth extent and obtain a set of straight lines that represent the final depth estimate of the scene.

In the case of 1D slices of the 2D hologram, a fixed-size ROI centered on the estimated straight lines is defined and used with the DFF method to further recover the curvature of the extracted lines.

3.2. Step 1: Computing the Phase Space Representation of the Hologram

We assume that a 1D hologram of a 2D scene has been generated. Given the 1D input hologram denoted $\mathcal{H} \in \mathbb{C}^W$, its 2D phase space transform is computed using the STFT based on sub-sampled hologram points x and sampled frequencies ξ defined as

$$\left\{x[n_1] := \frac{W\Delta_x}{2N}(2n_1 - N + 1) : n_1 = 0, 1, \ldots, N-1\right\}, \tag{5}$$

$$\left\{\xi[m_1] := \frac{1}{2\Delta_x M}(2m_1 - M + 1) : m_1 = 0, 1, \ldots, M-1\right\}, \tag{6}$$

where W is the hologram size, Δ_x is the hologram pixel pitch, and N and M are the number of sampled points and frequencies.The 2D phase space representation of the hologram is therefore given by

$$L[n_1, m_1] = \sum_{s=0}^{L-1} w[n_1 - s]H[s]e^{-js\xi[m_1]}, \tag{7}$$

where w is a Hann window of size $M + 1$ defined as

$$w(m) = 0.5(1 - \cos(2\pi\frac{m}{M})), \quad 0 \leq m \leq M + 1. \tag{8}$$

For a colored hologram, the STFT transform is applied to each color channel (R, G, B) of the hologram producing a complex-valued tensor of size $N \times M \times 3$.

3.3. Step 2: Extraction of Phase Space Masks

In this step, a binary mask representing the support of the hologram phase space representation is computed through binarization. To do so, the computed phase space representation L is fed to a *convolution neural network* (CNN) denoted by G based on the U-Net architecture [13] to binarize the phase space representation and estimate phase space masks, with one $M \times N$ binary mask per color channel. The network is supervised using ground truth masks computed from the scene point cloud $\mathcal{P}$ which produces the hologram $\mathcal{H}$ using the analytical formula proposed in [14]. The function Ω mapping the 2D point cloud $\mathcal{P}$ onto phase space can be defined as

$$\Omega(\mathcal{P}) = \bigcup_{n_1=0}^{N-1} \bigcup_{p=(y_1,y_2)\in\mathcal{P}} \Lambda_p(n_1) \tag{9}$$

where

$$\Lambda_p(n_1) = \left\{(x[n_1], f_\xi(n_1)) \mid -\frac{1}{2\Delta x} \leq f_\xi(n_1) \leq \frac{1}{2\Delta x}, \right\} \tag{10}$$

and $f_\xi(n_1)$ is the frequency component related to the sampled point $x[n_1]$, which is given as

$$f_\xi(n_1) := -\frac{(x[n_1] - y_1)}{\lambda\sqrt{(x[n_1] - y_1)^2 + (y_2)^2}}. \tag{11}$$

From there, the ground truth binary mask defining the phase space projection support can be computed as

$$M_b[n_1, m_1] = \mathbb{1}_{(x[n_1],\xi[m_1])\in\Omega(\mathcal{P})}. \tag{12}$$

where $\mathbb{1}$ is the indicator function.

Using the resulting phase space representation as input, the network is trained to output a binary mask $\hat{M}_b$ as close as possible to the ground truth binary mask M_b. The training process can be formalized as follows:

$$\hat{M}_b = \mathcal{G}_\theta\{(\mathbb{R}e\{L\}, \mathbb{I}m\{L\})\}, \quad L \in \mathcal{C}^{N\times M} \tag{13}$$

$$\min_{\theta} \mathcal{L} = BCE(\hat{M}_b, M_b), \quad M_b, \hat{M}_b \in \mathbb{R}^{N \times M} \tag{14}$$

where $(\mathbb{R}e\{L\}, \mathbb{I}m\{L\})$ is the concatenation of the real and imaginary parts of the hologram phase space representation L defined in Equation (7), θ are the parameters of the network, and BCE is the binary cross-entropy loss.

The use of a CNN model was preferred to thresholding the values of the matrix L for two main reasons. First, the threshold value varies from one hologram to another; it is therefore necessary to compute an optimal threshold value for each hologram. Second, the resulting binary mask from thresholding is coarse and requires an additional processing step using filters with sizes adapted to the level of degradation. This is an additional operation that requires significant computation time and often results in a degradation of the initial outline of the object. Using a CNN model ensures fast and accurate phase space segmentation without the need for manual editing or an additional post-processing step.

3.4. Step 3: Regions of Interest First Estimate

The goal of this step is to extract an area of the 2D scene, further called ROI (region of interest), the 2D points belonging to the scene. To do so, each point $p = (x, y)$ belonging to the 2D scene is mapped into phase space using the mapping Ω defined in Equation (9). The obtained phase space representation of the 2D candidate point is intersected with the binary mask estimated $\hat{M}_b$ in Step 2. The relative intersection provides the candidate point score. More formally:

$$score(p, \hat{M}_b) = \frac{\left|\hat{\Omega}_{\hat{M}_b}(\mathcal{P}) \cap \Omega(p)\right|}{|\Omega(p)|}, \quad p = (x, z) \in \mathbb{R}^2 \tag{15}$$

where

$$\hat{\Omega}_{\hat{M}_b}(\mathcal{P}) = \{(x[n_1], \xi[m_1]) \mid \mathbb{1}_{\hat{M}_b[n_1, m_1]=1}\}. \tag{16}$$

The score for a candidate point (x, z) defined in Equation (15) quantifies how much active frequencies that define this point in phase space are also active in the support $\hat{M}_b$ of the input hologram phase space representation. For a candidate point to truly belong to the scene, all those frequencies should be active. Therefore, a thresholding step is performed in order to keep only the points (x, z) with a score equal to 1, i.e., all points fulfill the condition of active frequencies, as follows:

$$ROI(\hat{M}_b) = \hat{\mathcal{P}} = \{(x, z) \mid \mathbb{1}_{score(p, \hat{M}_b)=1} \quad \forall p = (x, z) \in \mathbb{R}^2\}. \tag{17}$$

A score less than 1 can also be encountered with a point that really belongs to the scene but with a degraded mask $\hat{M}_b$.

Figure 2 gives two mapping examples with the extracted ROI colored in blue. In the first case, the extracted ROI consists of perfect diamonds that encapsulate the scene points colored in red. In this specific case, the geometry of the scene can be estimated using the set of lines that connect the two extremities of each diamond along the X-axis. In the second case, the extracted ROI does not have a perfect diamond shape due to the proximity of the different scene points.

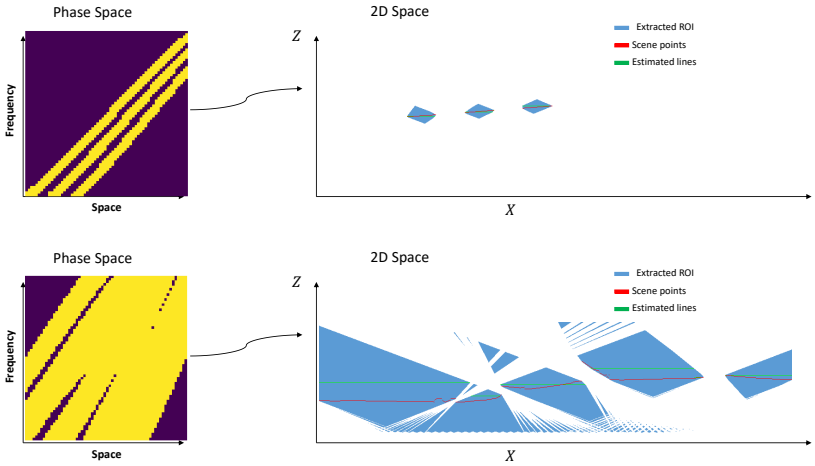

Figure 2. Illustration of the two different cases that can be observed during the ROI extraction. The first case, shown at the top of the figure with a scene composed of 3 segments, is a perfect mapping that produces diamond shapes that encapsulate the scene points. The second case, shown at the bottom of the figure, is a more common mapping case of larger scene segments where diamonds are hidden due to lack of space between scene points. In both of these cases, the iterative method produces relevant results.

The closer the points are to each other, the larger the extracted ROI and the bigger its depth extent. In contrast, the further apart the points are, the finer the ROI and the smaller the depth extent. Therefore, if the scene points are very close to each other and extend along the entire X-axis, then the extracted ROI will be very large with a high depth extent.

3.5. Step 4: Regions of Interest Iterative Refinement

To better control the depth extent of the extracted regions of interest, a proposed iterative refinement algorithm is introduced in the next section and is applied to the extracted ROI in order to calculate the optimal representation line. Then, a new ROI of fixed size centered on the predicted line is defined and used as the final ROI.

The initially extracted ROI is a subset of the minimal set of points that produce the same binarized phase space representation as $\hat{M}_b$. As a result, the ROI can be further refined to produce the smallest set of points that when mapped into phase space produce the same representation as $\hat{M}_b$. To reduce combinatorial complexity, primitives larger than the points, namely lines, are chosen as the set of candidates to be combined in order to produce the same representation as $\hat{M}_b$, as shown in Figure 3.

The decomposition into a minimal set of lines is described in Algorithm 1. First, the set of candidate points $\hat{P}$ is decomposed into a set of disjoint regions $ROIs(\hat{P})$ using the **findContours** from the Opencv Library [15]. Then, all lines that are completely embedded in each subregion are extracted. Each line is mapped into phase space, and its contribution to the estimated phase space representation $\hat{M}_b$ is evaluated. The line that contributes the most is selected. Finally, all candidate points that are in the same interval along the X-axis as the selected line are removed.

As shown in Figure 4, the contribution of the selected line is not removed from the matrix $\hat{M}_b$ because other candidate lines may share the same points as the selected line in phase space. Therefore, removing the line contribution would remove candidate lines that may actually belong to the scene. To prevent this from happening, a buffer M_{pred} is used and is incremented with the phase space representation of the selected line at each iteration. The iteration stops when the contribution of the newly selected line is less than 5% overlap of the input phase space representation. An example of phase space decomposition is given in Figure 5.

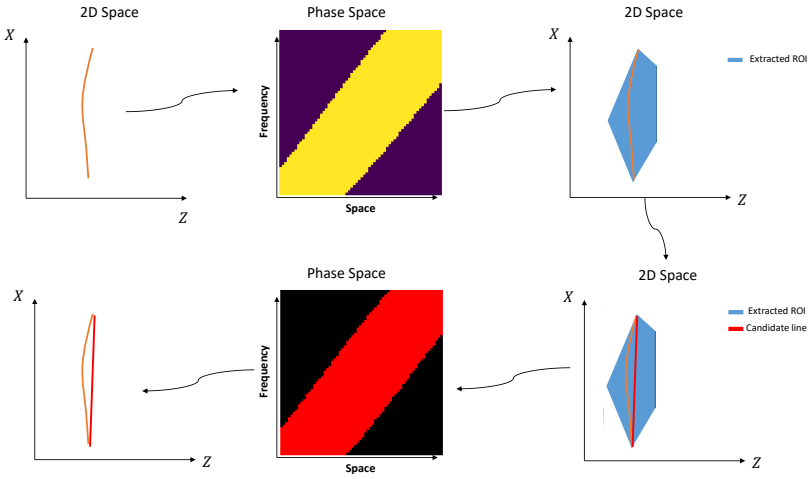

Figure 3. Illustration of the ROI refinement method. The initial ground truth 2 d scene is composed of a curved line, and its phase space representation is a band with a very large thickness and a slight slope. The region of interest colored in blue completely encompasses the scene points; however, it does not represent the minimal set of points that generate the same representation as the scene points. From all possible lines that are fully embedded inside the ROI, the red line is selected as the optimal candidate line that produces the same phase space representation as the scene points and will represent the final depth estimate of the scene in 1D.

Algorithm 1 Iterative decomposition into a set of lines

Require: Set of candidate points $\hat{P}$, and the estimated phase space representation $\hat{M}_b$.

$M_{pred} \leftarrow zeros(size(\hat{M}_b))$;
$S_{last} \leftarrow 0$;
$\hat{\Omega}_{\hat{M}_b}(\mathcal{P}) = \{(x[n_1], \xi[m_1]) \mid \mathbb{1}_{\hat{M}_b[n_1,m_1]=1}\}$
repeat
  $ROIs(\hat{P}) = findContours(\hat{P})$;
  $lines_{scores} \leftarrow []$;
  for $R \in ROIs(\hat{P})$ **do**
    Extract the set $\Theta(R)$ of horizontal lines that are completely included in R;
    for $l \in \Theta(R)$ **do**
      $\gamma = \frac{|\Omega(l) \cap \hat{\Omega}_{\hat{M}_b}(\mathcal{P})|}{|\Omega(l)|}$
      $lines_{scores}.append(\gamma)$
    end for
  end for
  Select the optimal line $l = \arg\max lines_{scores}$
  $S_{current} \leftarrow \max lines_{scores}$
  if $(S_{current} - S_{last}) \geq 0.05$ **then**
    $S_{last} \leftarrow S_{current}$
  else
    Break
  end if
  $M_{pred} \leftarrow (\Omega(l) \cup M_{pred})$
  Remove contribution of line l in set $\hat{P}$,

$$\hat{P} \leftarrow \{(x,z) : x \notin [min_x(l), max_x(l)] \quad \forall (x,z) \in \hat{P}\}$$

until $(S_{current} - S_{last}) < 0.05$.

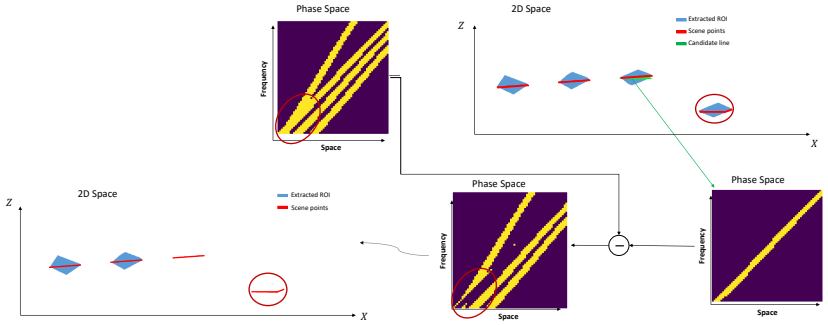

Figure 4. An illustrative example that motivates the use of a buffer in the proposed algorithm. Given the input phase space representation, the ROIs colored blue are extracted. The extracted regions of interest encapsulate all the points of the scene colored in red and are completely disjoint in 2D space. However, their representation in the phase space is not disjoint. A first candidate line colored in green which is fully embedded in one of the regions of interest and contributes the most to the input mask is selected. Since no buffer is used, the contribution of the selected line is directly removed from the inputted phase space representation, and the resulting representation is mapped again in the 2D space. The resulting ROI encapsulates only two objects, and no ROI is associated with the third object which shares similar points in the phase space as the candidate line.

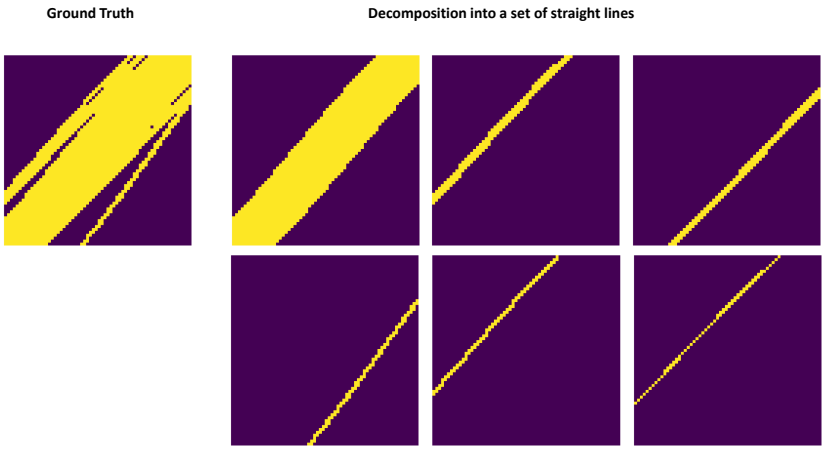

Figure 5. Result of the application of the iterative method for the decomposition of phase space into sets of straight lines. The proposed algorithm provides a minimal set of lines that when mapped in the phase space produce the same representation as the input phase space representation. Each 2D line is represented in phase space by a band with a slope and a thickness which are given by the depth and the length of the 2D line segment. Even though 2D lines are spatially disjoint in 2D space, their representation in phase space can intersect.

3.6. Step 5: Curvature Estimation Using a DFF Method

The three steps described in the previous sections implicitly assume that the 2D scene is composed of piece-wise planar objects and are thus not suited for curved objects. In this section, we propose to refine the set of linear segments into a set of curved segments when processing 1D slices of a 2D hologram by using the DFF method presented in the following.

It is not possible to extract 1D holograms corresponding to each of the 2D space slices directly from the 2D hologram. However, it is possible to extract a 2D phase space (x, ξ) corresponding to one slice (x, z) from 4D phase space (x, y, ξ, η) computed from a 2D hologram by selecting only the 2D phase space representation along a specific frequency. More formally, given a 2D hologram, the 4D representation in phase space is calculated using STFT producing a four-dimensional complex-valued matrix of size $N \times N \times M \times M$ which represents the axes x, y, ξ, η, given as:

$$\left\{x[n_1] := \frac{W\Delta_x}{2N}(2n_1 - N + 1) : n_1 = 0, 1, \ldots, N-1\right\}, \tag{18}$$

$$\left\{y[n_2] := \frac{H\Delta_y}{2N}(2n_2 - N + 1) : n_2 = 0, 1, \ldots, N-1\right\}, \tag{19}$$

$$\left\{\xi[m_1] := \frac{1}{2\Delta_x M}(2m_1 - M + 1) : m_1 = 0, 1, \ldots, M-1\right\}, \tag{20}$$

$$\left\{\eta[m_2] := \frac{1}{2\Delta_y M}(2m_2 - M + 1) : m_2 = 0, 1, \ldots, M-1\right\}, \tag{21}$$

where H, W are the height and width of the hologram, and Δ_x, Δ_y are the pixel along the X and Y directions.

By rearranging the axes of the matrix to $y, \eta, (x, \xi)$, the phase space representation can be seen as a set of 2D phase space representation (x, ξ) of oriented 2D planes with an orientation and a position that depend on η and y, respectively, as shown in Figure 6. The choice of frequency η corresponds to a certain orientation of the 2D plane, and the central frequency $\eta = 0$ produces a set of 2D planes (x, z) which cut the scene horizontally at y without containing occlusion, i.e., two points cannot share the same x position at different depths.

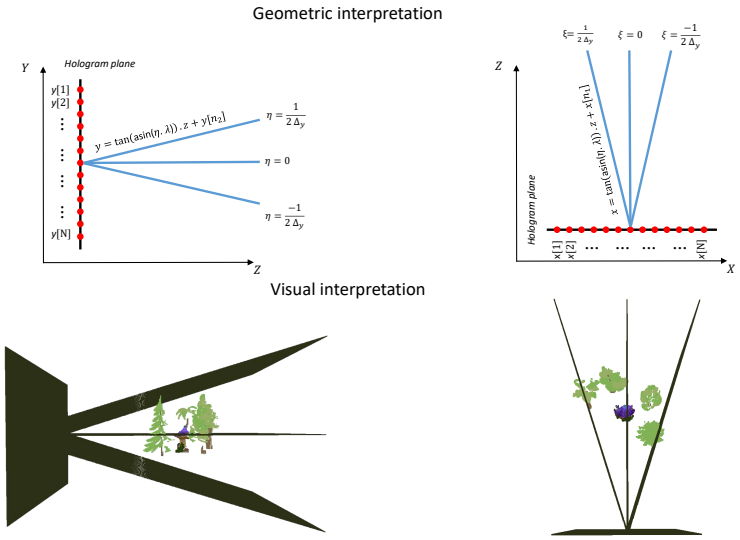

Figure 6. Illustration of the geometric and visual interpretation of the 4D phase space transform (x, y, ξ, η) of a 2D hologram. The 4D phase space transform represents propagation value along rays that have an orientation determined by ξ and η and a position determined by x and y. By fixing the values for a position–frequency pair (x, ξ), the 2D matrix (y, η) can be interpreted as the phase space transform of the 1D hologram produced by the intersection points between the oriented 2D plane (whose orientation and position are determined by the selected values of ξ and x, respectively) and the 3D scene.

In the present work, only a sub-part of the phase space representation defined from a central frequency $\eta = 0$ is kept, thus producing a tensor denoted L of size $N \times N \times M$ corresponding to a 2D phase space representation (x, ξ) of a 2D horizontal plane (x, z) with a height equal to y. The choice of the frequency $f_\eta = 0$ is motivated by two reasons. First, it produces horizontal slices which do not contain an occlusion. Second, it makes it easier to extract 2D patches for the application of the DFF method, which does not need to be oriented in the same direction as the plane defined by the frequency f_eta.

The ROIs are extracted for each of these 2D phase space representations (x, ξ) as shown in Figure 7, and then the minimal representation lines are extracted from the extracted ROIs using the iterative algorithm. More formally:

$$\hat{M}_b(y[n_2]) = \mathcal{G}_\theta\{(\mathbb{R}e\{L[n_2]\}, \mathbb{I}m\{L[n_2]\})\}, \tag{22}$$

$$\hat{\mathcal{P}}(y[n_2]) = \{(x,z) \mid \mathbb{1}_{score(p,\hat{M}_b(y[n_2]))=1} \quad \forall p = (x,z) \in \mathbb{R}^2\}, \tag{23}$$

$$\hat{D}(y) = Refinement(\hat{P}(y[n_2])). \tag{24}$$

Finally, a new fixed-size ROI centered on the predicted lines is defined and used for the DFF method using the auto-switch proposed in [8] as follows:

$$d(x,y) = \underset{i \in [\hat{D}(y)[x]-[N_z/2], \hat{D}(y)[x]+[N_z/2]]}{\arg\max} \mid FM(R_{x,y,i}) - \mu \mid, \tag{25}$$

$$\mu = \frac{1}{N_z} \sum_{\hat{D}(y)[x]-[N_z/2]}^{\hat{D}(y)[x]+[N_z/2]} FM(R_{x,y,i}) \tag{26}$$

$$R_{x,y,i}(u,v) = I_i(x+u-s/2, y+v-s/2), \tag{27}$$

where d is the final depth map, N_z is the interval size, FM is the focus measure function, I_i is the intensity of the holographic reconstruction performed at the i-th distance, and s is the patch size fixed at 17 which showed the best performance in the experiments through hyperparameter tuning.

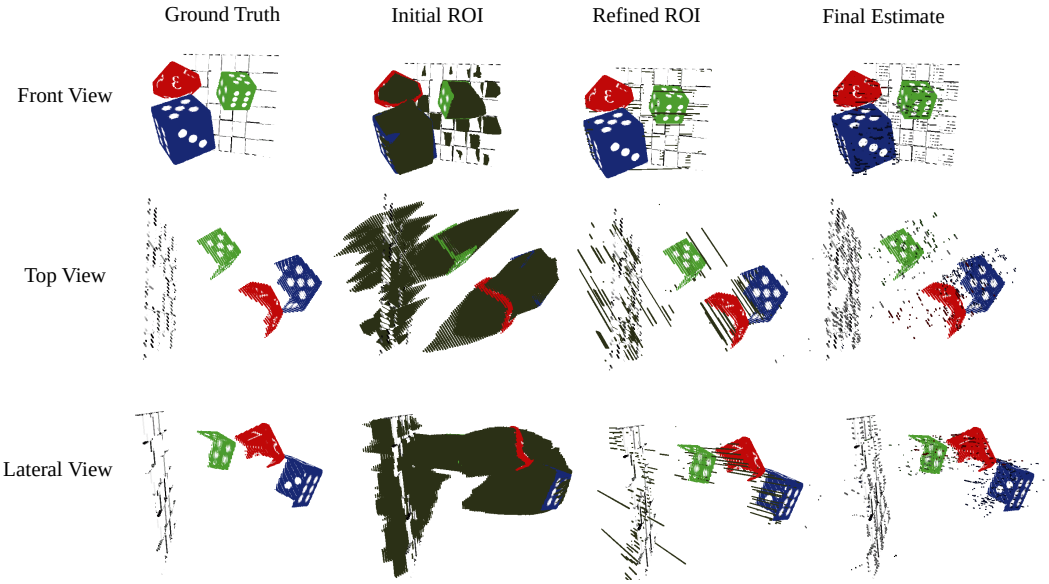

Figure 7. An illustrative example that shows the initially extracted ROI, the refined ROI, and the final estimate produced by the DFF method using a fixed-size ROI of size 31 centered on the refined ROIs at different viewing angles. The extracted ROI can be used to pre-localize the scene in the 3D space. They thus define an appropriate reconstruction interval and contain the pixels on which the focus measurement applies in the reconstruction volume.

In this final stage, the depth of each point (x, y) in the initial line segment $\hat{D}(y)[x]$ is refined by the DFF process, which assesses the depth i for which the focus of the patch $R_{x,y,i}$ is optimal inside a reconstruction interval of size $N_z + 1$ centered around the initial depth estimate $\hat{D}(y)[x]$. The obtained results will depend on the choice of the focus measure, the size of the chosen interval, and the obtained result from the iterative method.

4. Experimental Results

The following section details the experiments that were conducted to validate the proposed approach. Section 4.1 presents the holographic dataset used for the experiments. Section 4.2 analyzes the phase space masks extracted in the second step of our method, and Section 4.3 provides the depth estimation results obtained with our method. Finally, the calculation time of our method is analyzed in Section 4.5.

4.1. Holographic Dataset

For training the $\mathcal{G}$ network, a large-scale dataset consisting of 4200 RGB holograms was obtained from three different scenes (piano, table, wood). The holograms were acquired using a layer-based method [16] with a resolution of 1024×1024, a pixel pitch of 1 µm, and 1400 acquisition angles. The evaluation phase was performed on two sets: a test set consisting of 300 holograms evenly distributed over the training classes and a validation set consisting of 200 holograms from two additional scenes (cars, dice). Table 1 summarizes the depth in centimeters of each object that makes up each scene.

Table 1. Scene object sizes along the X/Y/Z, in centimeters.

	Object 1	**Object 2**	**Object 3**	**Object 4**	**Object 5**
Piano	0.044/0.054/0.052	0.023/0.017/0.01			
Table	0.039/0.026/0.038	0.018/0.037/0.016	0.016/0.037/0.018	0.018/0.037/0.016	0.016/0.037/0.018
Woods	0.03/0.045/0.03	0.021/0.046/0.023	0.034/0.032/0.028	0.025/0.042/0.025	0.031/0.042/0.022
Dice	0.02/0.017/0.021	0.028/0.029/0.034	0.023/0.025/0.028	0.0512/0.0512	
Cars	0.085/0.014/0.052	0.082/0.016/0.052	0.049/0.025/0.082	0.046/0.03/0.073	

For each hologram of the dataset, its 4D phase space representation L of size of $N \times N \times M \times M$ and the associated phase space masks M_b of size $N \times N \times M \times M$ are computed using 3D formulas proposed in [14]. Then, the two tensors L and M are reorganized as a 3D tensor of size $(N \times N) \times M \times M$ and used as the initial batch for the network training. Finally, for each iteration, a sub-sample of size 1024 is randomly sampled from the initial batch to constitute the final network input/output data.

The network was trained for a total of 100 epochs, using the stochastic gradient descent and an exponential learning rate decay.

4.2. Phase Space Segmentation

In the following, we analyze the binary phase masks extracted in the second step of our method. The Dice and Jaccard metrics [17] are used to evaluate the quality of the overlap between the predicted binary mask and the ground truth. Table 2 gives the results obtained for the test and validation set.

Table 2. Segmentation metric for the test (piano, table, woods) and validation set (cars, dice).

	Piano	**Table**	**Woods**	**Cars**	**Dice**
Dice	0.97	0.97	0.98	0.94	0.94
Jaccard	0.94	0.94	0.96	0.89	0.88

From the obtained results, we can see that for the test part, the network has an overall accuracy of 95%. This performance drops by 5% on the validation set, but the general performance remains high enough to hope for good results in the next steps of the algorithm. The low difference in segmentation performance between the validation and test set suggests that the network is not subject to overfitting, which can occur when the network is over-parameterized and the amount of data is not sufficient.

A bad segmentation performance leads to a bad estimation of the phase space representation. Therefore, fewer candidate points can be extracted during the back projection

operation, and some scene points may not be included in the extracted ROIs as shown in Figure 8. By using multiple color channels those areas that are missing on one color channel can be filled in on another channel. It is therefore important but not mandatory when processing RGB holograms to merge the extracted ROIs. The merged ROIs can be projected back into a selected color channel phase space representation and used as a target for the iterative algorithm. More formally,

$$R_b = \mathcal{G}_\theta\{(\mathbb{R}e\{L_R\}, \mathbb{I}m\{L_R\})\}, \tag{28}$$

$$G_b = \mathcal{G}_\theta\{(\mathbb{R}e\{L_G\}, \mathbb{I}m\{L_G\})\}, \tag{29}$$

$$B_b = \mathcal{G}_\theta\{(\mathbb{R}e\{L_B\}, \mathbb{I}m\{L_B\})\}, \tag{30}$$

$$\hat{\mathcal{P}} = \bigcup_{M \in \{R_b, G_b, B_b\}} ROI(M), \tag{31}$$

$$\hat{M}_b = \Omega(\hat{\mathcal{P}}), \tag{32}$$

where R_b, G_b, and B_b are the estimated phase space representation for each color channel. Their mappings in the 2D space are unified by selecting all the points that fall in at least one of the three mappings and projected back into phase space using a single color channel. The set of candidate points $\hat{\mathcal{P}}$ and its projection $\hat{M}_b$ are used as input to the iterative algorithm.

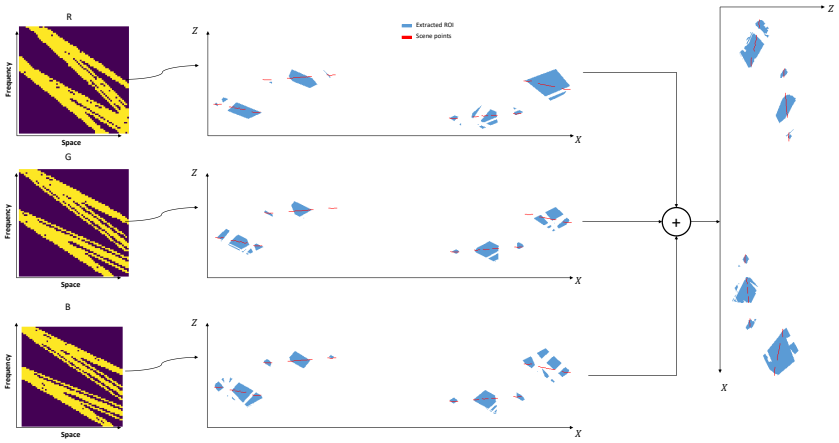

Figure 8. An illustrative example of the importance of color blending. The segmentation masks made for each color channel contain small defects, which lead to a bad extraction of the regions of interest because the points that are defined on the segmentation defects will not be selected. By unifying the results produced by the different color channels, imperfections related to segmentation performance can be bridged, and a greater number of points are included in the extracted regions of interest.

4.3. Depth Estimation Results

To evaluate the depth estimation performance of our proposed method, three experiments illustrated in Figure 9 are conducted:

- Without ROI: in this experiment, the depth extent in Equation (25) spans the interval $[0, z]$, where z is the maximum possible depth value set to 0.20 cm in the experiment.
- With ROI: in this second experiment, the depth extent in Equation (25) is reduced to the interval $[z_{\min}, z_{\max}]$ where $z_{\min}, z_{\max}$ are the maximum and minimum depth values present in the extracted ROI.
- With refined ROI: in this third experiment, the depth extent in Equation (25) is reduced to a fixed-size interval of size $N_z + 1$ centered on the minimal representation lines extracted from the initial ROI using the iterative method.

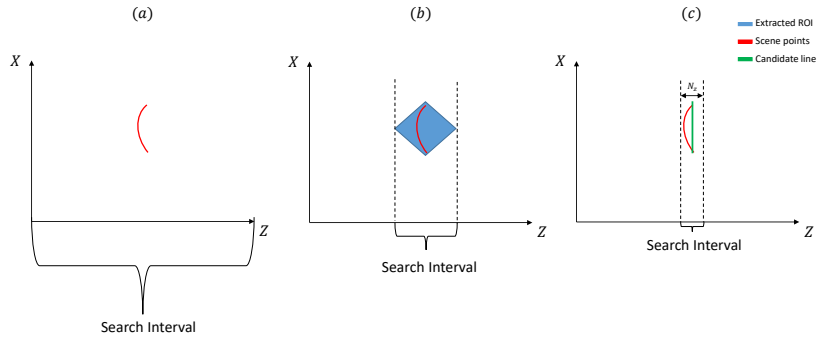

Figure 9. An illustrative figure of the three experimental configurations: (**a**) the depth value is searched over the whole space; (**b**) regions of interest are used to delimit the search interval; (**c**) the regions of interest are refined, and a fixed size interval of size $N_z + 1$ centered on the predicted optimal representation line is used as the search interval.

Those experiments have two objectives: (1) to assess the quality of the extracted regions of interest (ROIs), as any degradation of the ROI can negatively impact the DFF method, especially if a few scene points fall within it, and (2) to evaluate the efficiency of the iterative method by aiming to produce an estimate that is as close as possible to the ground truth. This is crucial in order to minimize the number of reconstruction planes required for the DFF, specified by N_z.

In our experiments, the DFF method was performed using the six most performing focus measures in [8] and detailed in Table 3. The operators used are highly parallelizable and offer a good compromise between performance and speed.We evaluated the performance of each focus measure using the ℓ_1 norm between the predicted and ground truth depth maps used by the layer-based method.

Table 3. Abbreviations of focus measure operators used in the experiments.

Focus Operator	Abbr.	Focus Operator	Abbr.
Variance of Laplacian [18]	LAPV	Image contrast [19]	CONT
Variance of wavelet coefficients [20]	WAVV	Ratio of the wavelet coefficients [20]	WAVR
Gray level variance [21]	GLVA	Normalized Gray level variance [22]	GLVN

Table 4 gives the obtained results with and without the proposed ROI approach, Figure 10 gives the obtained results with an additional ROI refinement step using the iterative method with different N_z values, and Figure 11 gives some visual results.

First, we can observe that the WAVR, WAVV, and GLVA operators are the best focus measure operators for the without ROI case. The FM performance varies depending on the processed scene, with a mean ℓ_1 norm of roughly 15 reconstruction planes. Performance may be lower if the scene contains many flat areas with no texture variation. The focus operators generally work best on regions with high texture variances such as edges and corners.

Second, using extracted regions of interest results in improved performance, with gray-level variance-based operators being the most optimal. This is because the depth extent used in Equation (25) is reduced, providing a more accurate estimate of the depth of each pixel. The performance variance between the first and second cases may be due to the quality of the extracted ROI or to multiple extremum values present in the focus curve.

Table 4. The obtained ℓ_1 norm for the test (piano, table, woods) and validation set (cars, dice).

	Piano	**Table**	**Woods**	**Cars**	**Dice**
			Without ROI		
LAPV	28.80	42.61	71.57	85.65	68.34
WAVV	18.27	43.09	12.28	34.87	17.72
WAVR	14.99	32.84	15.89	31.96	30.03
GLVN	14.87	38.54	15.97	34.82	26.65
GLVA	14.30	39.26	12.09	33.06	18.53
CONT	22.67	54.47	42.88	59.94	74.01
			With ROI		
LAPV	26.22	38.67	29.50	83.80	15.82
WAVV	11.36	23.78	25.37	31.97	23.61
WAVR	8.63	23.68	25.04	46.17	14.73
GLVN	6.84	20.86	13.55	32.42	9.72
GLVA	11.26	23.00	28.62	41.23	22.24
CONT	24.86	35.46	36.28	65.55	29.56
			ROI Refinement		
11	9.28	11.04	10.42	25.58	5.31
21	7.44	10.22	8.65	24.84	4.05
31	6.33	10.03	7.53	24.28	3.8
41	5.65	10.32	6.97	24.18	3.93
51	4.94	10.79	6.90	24.03	4.37
61	4.86	11.08	6.76	24.13	4.98
71	4.73	11.48	6.68	22.87	5.83
81	4.81	12.10	6.57	21.72	6.68
91	4.88	13.00	6.38	20.73	7.54

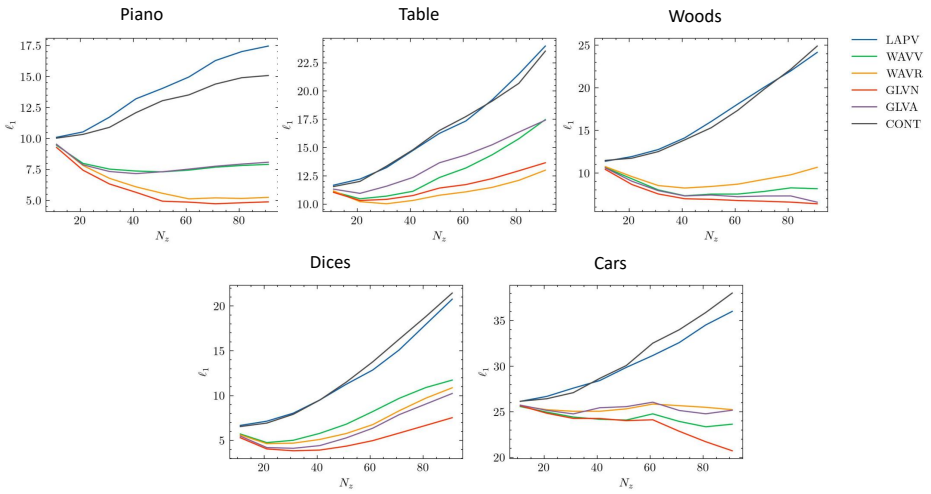

Figure 10. Norm ℓ_1 with respect to the selected size N_z of the new ROI region around the refined ROI used in the DFF method for the five different scenes. The optimal representation lines computed from the extracted regions of interest using the iterative method are relatively close to the scene points but lack curvature which limits the overall performance of the algorithm. By using the DFF method on a fixed size interval centered on computed lines, the curvature can be recovered, resulting in a finer depth estimate closer to the scene points and therefore a low ℓ_1 norm.

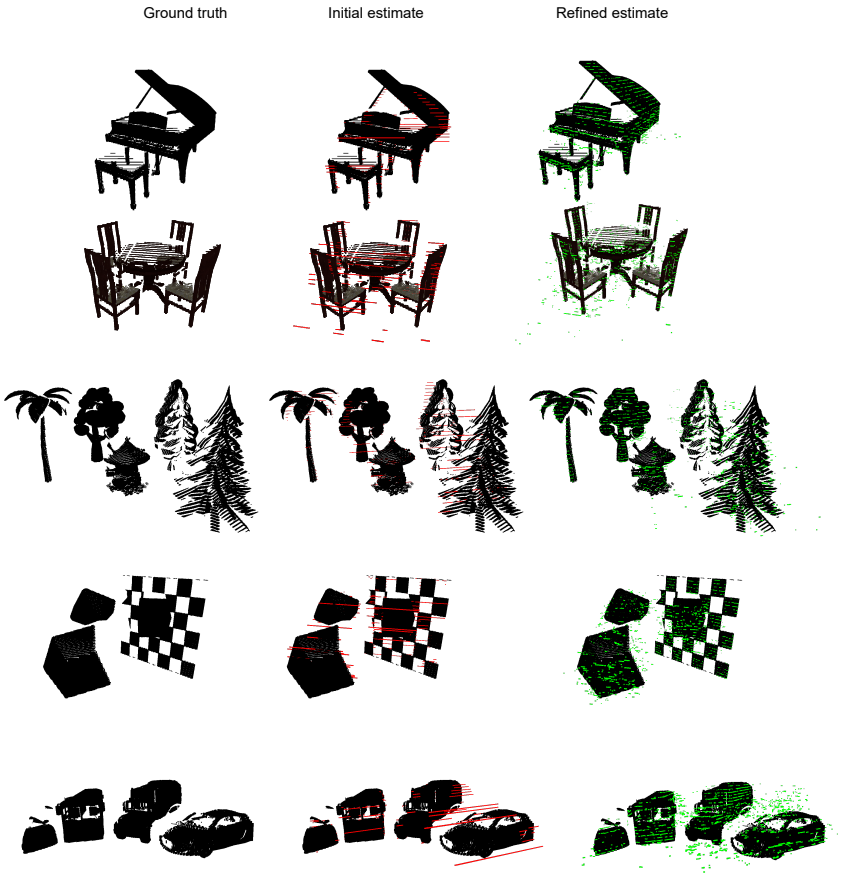

Figure 11. Result of the application of the iterative method on 2D holograms.

Finally, the refinement of the initial ROI and the use of a new fixed-size interval allows an additional gain in performance. From Figure 10, we can observe that optimal performance (except LAPV and CONT operators) can be controlled by changing the parameter N_z in Equation (25), and a minimum value can be reached around $N_z = 41$. Then, the performance degrades with a larger choice of N_z. For the LAPV and CONT operators, the overall performance degrades as N_z increases which indicates that the operators are not sensitive enough to the slight focus change between different images of the reconstruction volume. For the Cars scene, where the objects that compose the scene are very large, the optimal performances are achieved for the largest value of N_z regardless of the choice of the FM function.

4.4. Hyperparameter Tuning

When each object in the scene spans the entire 3D space, the value N_z must be large enough in order to correctly refine the depth of each object. However, as shown in Figure 12, there is a trade-off between choosing the right interval size and the used focus measure. The larger the interval size, the greater the risk of having a large estimation error due to the lack of sensitivity of the focus measures. Therefore, when increasing the size of the interval, some points may converge to the correct value that will be included in the chosen interval, and others may diverge because the used focus measure is not robust enough to detect the optimal focus plane given a very large number of images, which may lead to a bad final result.

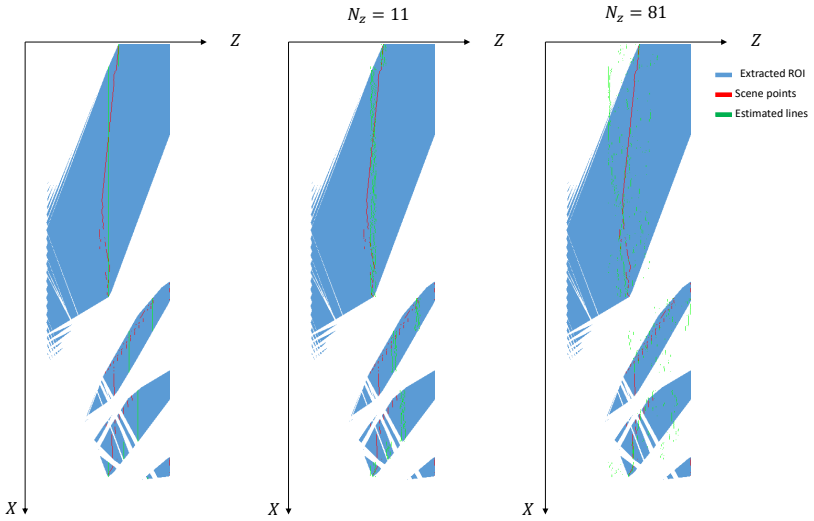

Figure 12. An illustrative figure of the trade-off between the choice of the best interval size N_z and the accuracy of the focus measure on a 1D slice of the Cars scene. The initial estimate of the optimal representation lines is relatively close to the scene points, with exception of some regions located at the bottom of the scene where the distance between the estimated points and the ground truth is quite large. With a very small value of N_z, the distance of the badly estimated points will not be strongly reduced; only the points close to the initial estimate can be well-refined. With a large value of N_z, the badly estimated points start to be close to the ground truth value. However, since the focus measure is not sensitive enough to the focus change, the initial correctly estimated points diverge strongly from the initial estimate, producing a bad final result.

A visual example of the different steps involved in our proposed methodology is given in Figure 7, where the initially extracted ROI perfectly encapsulate the scene points, and their refinement falls either close to the objects or inside them. The curvature is estimated in the last step of the algorithm using the DFF method on a new fixed-size ROI of size 31 around the refined ROI. It can be observed that the final estimate is relatively close to the ground truth value; however, some regions are poorly estimated by the GLVN operator, which results in a dispersion of the point cloud clearly visible in the figure.

In summary, depth estimation without ROI is a tedious task since the possible depth values for each pixel span a large interval. The performance can be improved using the ROI which reduces the possible depth interval for each pixel and therefore reduces uncertainty during the final estimation. However, if the processed scene has complex geometry with points that are close to each other and span over the entire X-axis, then the extracted ROI may have a large depth extent; therefore the gain when using the ROI is reduced.

The ROI refinement and the use of fixed-size ROI can manage the large depth extent and produce better interval bounding. The use of refined ROI may not be the best choice when processing scenes with objects spanning a large depth range. In that case, the best choice for N_z may be large, and the produced depth interval can be approximately $[0, z]$.

The N_z values in the DFF method can be selected adaptively by starting from the ends of the representation lines with a small N_z value, and gradually increasing it as the process moves towards the center of the line, with the maximum value determined by prior knowledge about the object's curvature.

4.5. Calculation Time

Table 5 gives the number of operations needed for each approach. The table does not include the calculation of pre-operators, such as Laplacian for LAVP which are applicable to the whole image, and the performed holographic reconstructions which are not a good criterion. As shown in the bottom of Figure 2, if the scene points on one of the 1D hologram slices are spatially very close to each other and extend across the entire X-axis, the extracted

regions of interest will have a large depth extent, which can extend in the interval $[0, z]$ and thus require the same number of holographic reconstructions as without ROI.

Table 5. Calculation time in seconds to calculate the final depth map with and without ROI.

	Piano	**Table**	**Woods**	**Cars**	**Dice**
				Without ROI	
LAPV	3.89	4.10	4.92	5.91	4.97
WAVV	11.80	13.00	14.29	17.84	14.35
WAVR	6.87	6.88	7.78	9.20	7.28
GLVN	4.65	4.82	5.31	6.42	5.35
GLVA	3.64	4.19	4.70	5.75	4.50
CONT	2.90	3.16	3.46	4.30	3.51
				With ROI	
LAPV	1.18	1.17	1.46	2.98	1.23
WAVV	2.58	2.38	3.39	8.64	2.54
WAVR	1.37	1.29	1.82	3.85	1.35
GLVN	1.26	1.31	1.62	3.09	1.43
GLVA	1.12	1.10	1.44	2.71	1.19
CONT	0.87	0.87	1.09	2.05	0.92

From the table, we observe that the computation time is significantly reduced by using the extracted regions of interest. The calculation time is variable depending on whether the operator needs to calculate the mean, the variance, or a ratio.

For the same number of holographic reconstructions performed, the post-processing of the reconstruction volume can be performed more quickly, and better results can be expected with the extracted regions of interest.

5. Conclusions

In this work, we proposed a regions of interest extraction procedure from the hologram phase space representation. The extracted regions are used to bound the numerical reconstruction interval and the spatial delimitation when applying the focus measure during the DFF approach. Moreover, an iterative refinement method is proposed to refine the extracted regions of interest and produce a more accurate depth estimate.

Experimental results have shown that the use of extracted regions of interest with the DFF method can reduce the number of calls to focus measure operators while improving the initial performance. In addition, the refinement of the extracted regions of interest and the use of a fixed-size interval when applying the DFF method allows better control of the depth extent of the provided regions of interest and an improvement in the final performance. The proposed methodology halved the computation time while increasing the accuracy by a factor of three.

The iterative method can be an alternative to the depth-from-focus method if the following two limitations are solved. First, the curvature must be directly recovered from the phase space representation without requiring any numerical reconstruction. Second, an appropriate fusing method to merge the estimations produce using different angle values η must be designed.

Future works will be dedicated to the design of an efficient end-to-end learning-based-method for fast and accurate depth estimation from a hologram phase space representation without requiring any additional post- or pre-processing steps, such as numerical reconstruction.

Author Contributions: Conceptualization, N.M.; Methodology, N.M.; Writing—original draft, N.M.; Supervision, A.G., P.G. and L.M. All authors have read and agreed to the published version of the manuscript.

Funding: Agence Nationale de la Recherche (ANR-A0-AIRT-07).

Data Availability Statement: No data were generated or analyzed in the presented research.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Gabor, D. A New Microscopic Principle. *Nature* **1948**, *161*, 777–778. [CrossRef] [PubMed]
2. Grossmann, P. Depth from Focus. *Pattern Recognit. Lett.* **1987**, *5*, 63–69. [CrossRef]
3. Onural, L.; Özgen, M.T. Extraction of three-dimensional object-location information directly from in-line holograms using Wigner analysis. *J. Opt. Soc. Am. A* **1992**, *9*, 252–260. [CrossRef]
4. Tachiki, M.L.; Itoh, M.; Yatagai, T. Simultaneous depth determination of multiple objects by focus analysis in digital holography. *Appl. Opt.* **2008**, *47*, D144–D153. [CrossRef] [PubMed]
5. Sheridan, J.T.; Kostuk, R.K.; Gil, A.F.; Wang, Y.; Lu, W.; Zhong, H.; Tomita, Y.; Neipp, C.; Francés, J.; Gallego, S.; et al. Roadmap on holography. *J. Opt.* **2020**, *22*, 123002. [CrossRef]
6. Goodman, J.W. Introduction to Fourier optics. In *Introduction to Fourier Optics*, 3rd ed.; Roberts and Company Publishers: Englewood, CO, USA, 2005; Volume 1.
7. Pertuz, S.; Puig, D.; Garcia, M.A. Analysis of focus measure operators for shape-from-focus. *Pattern Recognit.* **2013**, *46*, 1415–1432. [CrossRef]
8. Madali, N.; Gilles, A.; Gioia, P.; Morin, L. Automatic depth map retrieval from digital holograms using a depth-from-focus approach. *Appl. Opt.* **2023**, *62*, D77–D89. [CrossRef]
9. Özgen, M.T.; Demirbaş, K. Cohen's bilinear class of shift-invariant space/spatial-frequency signal representations for particle-location analysis of in-line Fresnel holograms. *J. Opt. Soc. Am. A* **1998**, *15*, 2117–2137. [CrossRef]
10. Onural, L.; Kocatepe, M. Family of scaling chirp functions, diffraction, and holography. *IEEE Trans. Signal Process.* **1995**, *43*, 1568–1578. [CrossRef]
11. Zhang, Y.; Zheng, D.X.; Shen, J.L.; Zhang, C.L. 3D locations of the object directly from in-line holograms using the Gabor transform. In Proceedings of the Holography, Diffractive Optics, and Applications II, Beijing, China, 7 February 2005; Sheng, Y., Hsu, D., Yu, C., Lee, B., Eds.; International Society for Optics and Photonics: Bellingham, WA, USA, 2005; Volume 5636, pp. 116–120. [CrossRef]
12. Birnbaum, T.; Kozacki, T.; Schelkens, P. Providing a Visual Understanding of Holography Through Phase Space Representations. *Appl. Sci.* **2020**, *10*, 4766. [CrossRef]
13. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. *arXiv* **2015**, arXiv:1505.04597.
14. Birnbaum, T.; Blinder, D.; Muhamad, R.K.; Schretter, C.; Symeonidou, A.; Schelkens, P. Object-based digital hologram segmentation and motion compensation. *Opt. Express* **2020**, *28*, 11861–11882. [CrossRef]
15. Bradski, G. The OpenCV Library. *Dr. Dobb'S J. Softw. Tools* **2000**, *25*, 120–123.
16. Gilles, A.; Gioia, P.; Cozot, R.; Morin, L. Hybrid approach for fast occlusion processing in computer-generated hologram calculation. *Appl. Opt.* **2016**, *55*, 5459–5470. [CrossRef] [PubMed]
17. Müller, D.; Rey, I.S.; Kramer, F. Towards a guideline for evaluation metrics in medical image segmentation. *BMC Res. Notes* **2022**, *15*, 210. [CrossRef] [PubMed]
18. Pech-Pacheco, J.; Cristobal, G.; Chamorro-Martinez, J.; Fernandez-Valdivia, J. Diatom autofocusing in brightfield microscopy: A comparative study. In Proceedings of the 15th International Conference on Pattern Recognition, Barcelona, Spain, 3–7 September 2000; Volume 3, pp. 314–317. [CrossRef]
19. Nanda, H.; Cutler, R. Practical calibrations for a real-time digital omnidirectional camera. *CVPR Tech. Sketch* **2001**, *20*, 1–4.
20. Yang, G.; Nelson, B. Wavelet-based autofocusing and unsupervised segmentation of microscopic images. In Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453), Las Vegas, NV, USA, 27–31 October 2003; Volume 3, pp. 2143–2148. [CrossRef]
21. Krotkov, E.; Martin, J.P. Range from focus. In Proceedings of the 1986 IEEE International Conference on Robotics and Automation, San Francisco, CA, USA, 7–10 April 1986; Volume 3, pp. 1093–1098.
22. Santos, A.; Ortiz-de Solorzano, C.; Vaquero, J.J.; Peña, J.; Malpica, N.; Del Pozo Guerrero, F. Evaluation of autofocus functions in molecular cytogenetic analysis. *J. Microsc.* **1998**, *188*, 264–272. [CrossRef] [PubMed]