

Article

Not peer-reviewed version

---

# A Robust Skeletonization Method for High-Density Fringe Patterns in Holographic Interferometry Based on Parametric Modeling and Strip Integration

---

[Sergey Lychev](#)\* and [Alexander Digilov](#)

Posted Date: 16 December 2025

doi: 10.20944/preprints202512.1319.v1

Keywords: holographic interferometry; fringe skeletonization; speckle noise; parametric modeling; strip integration; displacement field reconstruction



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a [Creative Commons CC BY 4.0 license](#), which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

# A Robust Skeletonization Method for High-Density Fringe Patterns in Holographic Interferometry Based on Parametric Modeling and Strip Integration

Sergey Lychev\*  and Alexander Digilov 

Ishlinsky Institute for Problems in Mechanics RAS, 119526 Moscow, Russia

\* Correspondence: lychevsa@mail.ru

## Abstract

Accurate displacement field measurement by holographic interferometry requires robust analysis of high-density fringe patterns, which is hindered by speckle noise inherent in any interferogram, no matter how perfect. Conventional skeletonization methods, such as edge detection algorithms and active contour models, often fail under these conditions, producing fragmented and unreliable fringe contours. This paper presents a novel skeletonization procedure that overcomes these limitations through a threefold approach: (1) representation of the entire fringe family within a physics-informed, finite-dimensional parametric subspace (e.g., a collection of Fourier-based contours), ensuring global smoothness and connectivity of each fringe; (2) introduction of a robust strip-integration functional, which replaces noisy point sampling with a Gaussian-weighted intensity integral across a narrow strip, yielding a smooth objective function, which is convenient to optimize with standard gradient-based techniques; and (3) a recursive quasi-optimization algorithm that takes into account fringe similarity for efficient and stable identification. The method's efficiency is quantitatively validated on synthetic interferograms with controlled noise, demonstrating significantly lower error compared to baseline techniques. Its practical utility is confirmed by successful processing of a real, interferogram of a bent plate containing over 100 fringes, enabling precise reconstruction of the displacement field that closely matches result of independent theoretical modelling. The proposed procedure provides a reliable tool for processing challenging interferograms where traditional methods may fail to obtain satisfactory result.

**Keywords:** holographic interferometry; fringe skeletonization; speckle noise; parametric modeling; strip integration; displacement field reconstruction

## 1. Introduction

Holographic interferometry is a cornerstone optical technique for full-field, non-contact measurement of displacements and deformations in experimental mechanics [1]. Its high sensitivity, enabling resolution of displacements on the order of the light wavelength, makes it invaluable for studying complex deformation phenomena. The accuracy of the method is intrinsically connected with the precision with which the resulting interference fringe patterns can be analyzed [2]. In principle, higher measurement resolution is achieved by analyzing interferograms with a high density of fringes. However, in modern practice, the outcome becomes critically dependent on the chosen digital image processing algorithm, as the manual detection of hundreds of fringes is unreliable [3].

The primary obstacle to automation is the presence of *speckle noise* – an unavoidable granular interference pattern generated by the random scattering of coherent light from surface micro-roughness, which is intrinsic to any physical hologram [4]. While post-processing filters (e.g., median or Gaussian) can reduce speckle visibility in a digital image, they require a delicate approach. Intensive smoothing can introduce systematic biases that shift the fringe locations, a particularly detrimental effect for

high-density patterns where fine details are essential. Consequently, the development of robust, noise-resistant skeletonization algorithms, that can dispense with significant pre-filtering remains an actual problem in optical metrology.

A variety of general-purpose image processing techniques have been applied to fringe identification. For patterns with moderate noise, edge detection algorithms (e.g., Marr — Hildreth [5], Canny [6] or Shen – Castan [7]) can be employed. However, for high-density patterns corrupted by strong speckle noise, these methods tend to produce fragmented, disconnected edge maps that require complex and often unreliable post-processing to assemble into complete contours.

Active contour models (snakes) and their variants (e.g., Gradient Vector Flow snakes [8,9]) offer a more flexible method by formulating the fringe identification process as a solution to an energy minimization problem [10]. This method has been successfully adapted for fringe analysis [11–14]; however, it has an inherent limitation when processing highly noisy, dense patterns, which lies in its high sensitivity to the snake's initial position.

To overcome these limitations, we propose a novel skeletonization procedure that strategically integrates physical prior knowledge with a noise-robust numerical core. Our contribution is threefold:

1. **Physics-informed parametric modeling.** We constrain the search for fringes to a purpose-built, finite-dimensional functional subspace  $\mathcal{A}_\theta$ , which is defined by specific parametric curves (e.g., trigonometric polynomials or splines). This subspace is constructed in such a way as to endow its elements with the required properties, such as smoothness, closure, and specific shape, guaranteeing that the identified fringes are physically plausible, connected curves.
2. **A robust strip-integration functional with local smoothing.** For fringe localization we consider the specific functional in the form of a Gaussian-weighted integral of intensity over a narrow strip surrounding the candidate curve. This formulation allows the use of efficient gradient-based optimization techniques. To compute this functional, a continuously differentiable intensity field  $\tilde{I}(x, y)$  is first obtained by local bicubic interpolation, described in Subsection 3.1. The resulting functional  $\mathcal{J}^s(\theta)$  is smooth with respect to the curve parameters  $\theta$ , ensuring stable convergence.
3. **A recursive quasi-optimization algorithm.** Exploiting the geometric similarity of adjacent fringes, the identification process proceeds recursively outward (or inward). The optimal parameters for an identified fringe seed the initial guess for the next, via a simple scaling transformation (*quasi-optimization*), followed by a full local refinement of all parameters. This strategy dramatically improves computational efficiency and reliability.

The performance of the proposed method is carefully validated. First, using synthetic interferograms with precisely controlled geometry and additive-multiplicative speckle noise, we demonstrate its acceptable accuracy and robustness compared to baseline methods. The error is estimated using two metrics: the Euclidean norm ( $L^2$ -norm) of the difference and its maximal absolute value ( $L^\infty$ -norm). Second, we apply the algorithm to a real, challenging interferogram from a bending experiment on a square plate, containing over 100 fringes. The algorithm successfully extracts the complete skeleton, enabling an accurate reconstruction of the displacement field  $w(x, y)$ . This result is shown to be in good agreement with an independent analytical solution, obtained in the paper [15], confirming the practical utility and accuracy of the entire procedure.

The remainder of this paper is organized as follows. Section 2 details the proposed skeletonization procedure. Section 3 describes key implementation aspects. Section 4 presents the quantitative validation on synthetic data. Section 5 showcases the application to real high-density interferograms. Finally, Section 6 discusses the results and outlines future work.

## 2. The Proposed Skeletonization Method

### 2.1. Physical Origins of Imperfections in Interferograms

To motivate the design choices of the proposed algorithm, it is essential to understand the physical nature of the imperfections present in a raw interferogram. These imperfections, which corrupt the ideal sinusoidal fringe pattern, arise from the optical setup imperfections and the nature of coherent

light scattering. They can be categorized by their spatial scale, necessitating different corrective strategies in the digital processing procedure.

In a typical off-axis holographic setup (e.g., the Leith – Upatnieks scheme [2]), the recorded intensity results from the interference of a reference plane wave  $E_r(\mathbf{r})$  and an object wave  $E_o(\mathbf{r})$  reflected from the specimen surface. In complex notation, these waves are:

$$\begin{aligned} E_r(\mathbf{r}) &= A_r e_r e^{ik_r \cdot \mathbf{r}}, \\ E_o(\mathbf{r}) &= A_o(x, y) J(x, y) e_o e^{ik_o \cdot \mathbf{r}}, \end{aligned}$$

where  $A_r$  and  $A_o(x, y)$  are the scalar amplitudes,  $e_r$  and  $e_o$  are the unit polarization vectors of the reference and initial object waves (directly from the laser source),  $k_r$  and  $k_o$  are the corresponding wave vectors, and  $J(x, y)$  is the spatially varying Jones matrix accounting for polarization changes upon reflection from the deformed surface.

The interference of the wavefronts from the specimen's reference and deformed states yields an intensity pattern of the form [4,16]:

$$I_{\text{obs}}(x, y) \propto I_0(x, y) \left[ 1 + V(x, y) \cos(\Delta\phi(x, y)) \right] R(x, y) + \text{"noise"}. \quad (1)$$

Here,  $\Delta\phi(x, y) = (4\pi/\lambda) w(x, y)$  is the phase difference directly proportional to the deformable object out-of-plane displacement  $w(x, y)$ ,  $V(x, y) \propto |(J_1 e_o) \cdot (J_2 e_o)^*|$  stands for the polarization-induced visibility term and  $R(x, y)$  is the contrast variation, induced by coherent noise. A transition between consecutive dark (or bright) fringes corresponds to a displacement increment of  $\lambda/2$ .

The terms  $I_0(x, y)$ ,  $V(x, y)$  and  $R(x, y)$  describe large- to medium-scale corruptions:

- **Non-uniform background intensity**  $I_0(x, y)$ : Caused by uneven illumination, varying surface reflectivity, and polarization effects ( $J_1 \neq J_2$ ). This leads to slow intensity variations across the image.
- **Visibility term and contrast variation**  $V(x, y)$ ,  $R(x, y)$ : The coherence noisy factor  $R(x, y)$  (dependent on laser temporal/spectral properties and surface roughness) modulates the overall fringe contrast. The visibility term  $V(x, y)$  induced by polarization and speckle decorrelation can further reduce contrast, even to zero.
- **Geometric distortion**: The non-coaxiality of reference and object beams in off-axis schemes, combined with lens imperfections, introduces a projective distortion between the object plane and the image sensor.

These low-frequency imperfections are corrected in the **preprocessing stage** (Section 3) via geometric unwarping and local intensity equalization.

The speckle-dependent terms in (1) present the most significant challenge. It is a fine-grained, high-frequency random pattern resulting from two physically unavoidable phenomena: (i) the finite spectral bandwidth of the laser source, and (ii) the random interference of waves scattered by surface micro-roughness. Unlike Gaussian additive noise, speckle is signal-dependent and exhibits characteristic correlation lengths ( $\rho_{\text{speckle}}$ ). Its spatial frequency content often overlaps with that of the high-density fringes themselves, making linear filtering unsuitable as it erodes fringe edges and introduces systematic localization errors.

This multi-scale analysis dictates our algorithmic strategy: large-scale distortions are removed by explicit preprocessing, while the fine-scale speckle noise is addressed not by filtering, but by the main idea of the proposed method – the **parametric modeling** (Subsection 2.2) and the **strip-integration functional** (Subsection 2.4) – which provide robustness without distorting the signal.

## 2.2. Parametric Modeling and Problem Formulation

The goal of skeletonization is to extract a family of planar curves  $\Gamma = \{\gamma_i\}_{i=1}^N$  that represent the level sets (fringes) of the phase map  $\Delta\phi(x, y)$  from a discretely sampled intensity image  $I$ . A

fundamental premise of our approach is to treat fringes as *continuous geometric entities* (curves) rather than collections of disconnected pixels. Consequently, the search for an optimal curve  $\gamma$  must be performed within a suitable functional space.

Let  $\Omega \subset \mathbb{R}^2$  denote the rectangular image domain. We consider the space of admissible curves  $\mathcal{V}$  to be a set of rectifiable, closed Jordan curves contained in  $\Omega$ . For any such curve  $\gamma \in \mathcal{V}$ , we define a functional as the normalized integral of the image intensity over its support:

$$\mathcal{J}(\gamma) = \frac{1}{\mu(\gamma)} \int_{\gamma} I d\mu,$$

where  $\mu$  is an appropriate measure on  $\gamma$ . In the simplest case, where the support is the one-dimensional curve itself,  $\mu$  is the arc-length measure ( $d\mu = ds$ ), and the normalization factor  $\mu(\gamma)$  is the total curve length  $L(\gamma)$ . This yields the *line-integral functional*:

$$\mathcal{J}^l(\gamma) = \frac{1}{L(\gamma)} \oint_{\gamma} I ds.$$

The choice of measure and its domain will be generalized in Subsection 2.4 to define more robust functionals. The core optimization problem is to find curves within a constrained set that extremize this functional. For an ideal, normalized interferogram where the low-frequency imperfections from (1) are absent ( $I_0 \equiv \text{const.}$ ,  $V \equiv 1$ ,  $R \equiv 1$ ) and speckle noise is removed, the intensity is proportional to  $1 + \cos(\Delta\phi(x, y))$ . The bright fringes (intensity maxima) correspond to curves that maximize  $\mathcal{J}^l$ , while dark fringes (minima) correspond to its minimizers. In a real, noisy image, the functional  $\mathcal{J}^l$  must be extremized despite the corruptions described in Subsection 2.1.

However, operating directly in the infinite-dimensional space  $\mathcal{V}$  is intractable and ignores the crucial physical prior: the expected fringe family possesses a specific global structure. Instead of regularizing the problem within  $\mathcal{V}$  (as active contours do), we *constrain* the solution a priori to a finite-dimensional subspace  $\mathcal{A}_{\theta} \subset \mathcal{V}$ . This subspace is defined by a parametric model:

$$\mathcal{A}_{\theta} = \{\gamma(\theta) \mid \theta \in \Theta \subset \mathbb{R}^m\}, \quad (2)$$

where  $\theta = (\theta_1, \dots, \theta_m)$  is a vector of parameters, and the mapping  $\theta \mapsto \gamma(\theta)$  is continuously differentiable. The choice of the model  $\mathcal{A}_{\theta}$  takes into account our a priori knowledge about the deformation. For instance, for the bending of a centrally loaded, clamped plate, the fringes are expected to be nested, closed, and quasi-concentric curves, which can be effectively modeled by a family of perturbed circles or ellipses (see Subsection 3.4).

By restricting the search to  $\mathcal{A}_{\theta}$ , the infinite-dimensional variational problem of extremizing  $\mathcal{J}^l$  over  $\mathcal{V}$  reduces to a finite-dimensional optimization problem over the parameter space  $\Theta$ :

$$\theta^* = \arg \operatorname{extr}_{\theta \in \Theta} \mathcal{J}^l(\gamma(\theta)).$$

This is the core conceptual shift from other methods: the smoothness, connectivity, and global shape of the solution are *built into* the search space  $\mathcal{A}_{\theta}$ , rather than being enforced via penalty terms during optimization.

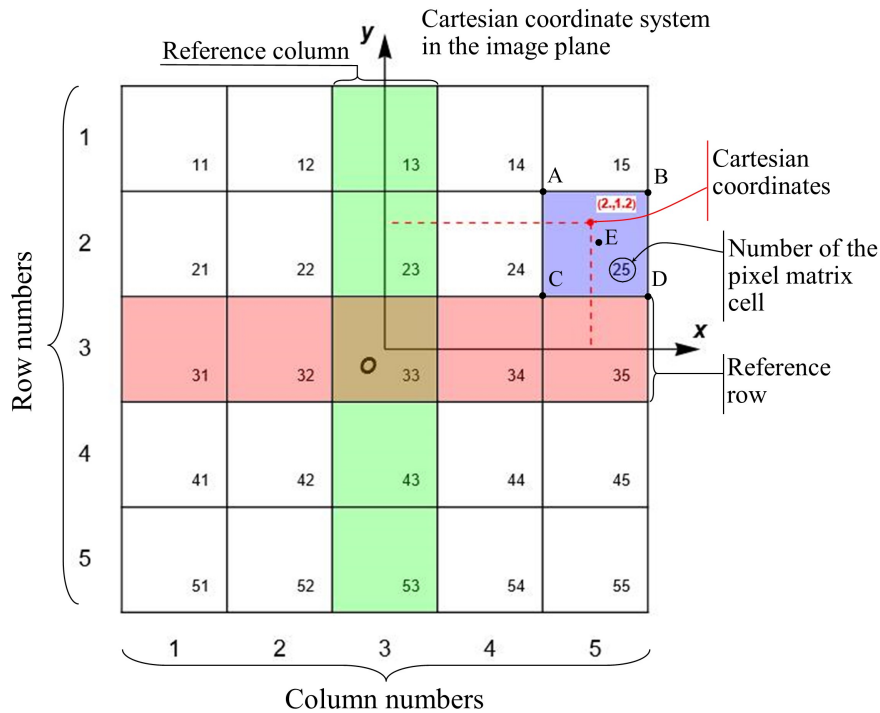
### 2.3. Discretization and Coordinate System

The image data  $I$  is provided as a discrete matrix  $\mathbf{I}[r, c]$  of size  $r_{\max} \times c_{\max}$ , where  $r$  and  $c$  are integer row and column indices. To connect the continuous formulation with the discrete data, we establish a fixed coordinate system. We assume a unit pixel pitch and place the origin of the Cartesian

coordinates  $(x, y)$  at the center of a chosen reference pixel with indices  $(r_0, c_0)$ . The correspondence between a pixel  $P[[r, c]]$  and the coordinates of its center  $(x_p, y_p)$  is then given by:

$$(x, y \in P[[r, c]]) \Leftrightarrow (r = r_0 - \lfloor y + 1/2 \rfloor, c = c_0 + \lfloor x + 1/2 \rfloor). \quad (3)$$

This mapping is illustrated in Figure 1.



**Figure 1.** Relationship between pixel matrix indices  $(r, c)$  and the continuous Cartesian coordinate system  $(x, y)$ . The grid represents the pixel centers. The origin  $(x, y) = (0, 0)$  is fixed at the center of the reference pixel with indices  $(r_0, c_0) = (3, 3)$ . The coordinates of point E (the center of pixel (2, 5)) are  $(x_E, y_E) = (1, 2)$ , following Equation (3). The vertices A, B, C, D denote the corners of a pixel, located at  $\pm 0.5$  offsets from its center.

#### 2.4. Variants of the Functional

The core of the skeletonization method is the optimization of a functional that measures how well a candidate curve aligns with a fringe. We trace its development from a naive discrete form to the final robust, continuous formulation.

##### 2.4.1. Point-Sampling Along the Curve

The most direct approach is to rasterize the curve  $\gamma(\theta)$  and sum the intensities of its pixels:

$$\mathcal{J}_{\text{point}}^l(\theta) = \frac{1}{N(\theta)} \sum_{\mathcal{R}(\gamma(\theta))} \mathbf{I}[r, c],$$

where  $\mathcal{R}$  is the set of pixels that rasterize the curve, and  $N(\theta) = |\mathcal{R}|$  is their number (an integer measure of curve length). This functional is *discontinuous* with respect to  $\theta$ , it means that a small parameter change can alter the set  $\mathcal{R}$ , causing a jump in the sum. This discontinuity precludes the use of gradient-based optimization.

#### 2.4.2. Line-Integration with Interpolation

To obtain a smooth functional, we first construct a  $C^1$ -continuous intensity field  $\tilde{I}(x, y)$  via bicubic interpolation (detailed in Subsection 3.1). Replacing the discrete sum with a normalized line integral yields:

$$\mathcal{J}^l(\boldsymbol{\theta}) = \frac{1}{L(\boldsymbol{\theta})} \oint_{\gamma(\boldsymbol{\theta})} \tilde{I}(x, y) ds, \quad (4)$$

where  $L(\boldsymbol{\theta})$  is the curve length.  $\mathcal{J}^l(\boldsymbol{\theta})$  is now continuously differentiable, enabling gradient-based optimization. However, it remains sensitive to speckle noise, as it samples intensity only along an infinitesimally thin path.

#### 2.4.3. Point-Sampling Over a Strip

To improve robustness, we consider a strip  $\mathcal{S}(\gamma, \delta)$  of width  $2\delta$  around the curve. A discrete, robust functional can be defined by summing pixel intensities within this strip, weighted by their approximate Gaussian distance to the curve:

$$\mathcal{J}_{\text{point}}^s(\boldsymbol{\theta}) = \frac{1}{N(\boldsymbol{\theta})} \sum_{\mathcal{R}_{\mathcal{S}}} G\left(\frac{d_{rc}}{\sigma}\right) \mathbf{I}[r, c],$$

where  $\mathcal{R}_{\mathcal{S}}$  is the set of pixels whose centers lie within the strip  $\mathcal{S}$ ,  $d_{rc}$  is the minimal distance from the pixel center to the curve, and  $G(u) = \exp(-u^2/2)$ . This choice of weight function is motivated by the statistical nature of speckle noise. The functional  $\mathcal{J}_{\text{point}}^s$  averages over many pixels, reducing the impact of speckles. However, it shares the discontinuity flaw of  $\mathcal{J}_{\text{point}}^l$  due to the discrete pixel set  $\mathcal{R}_{\mathcal{S}}$ .

#### 2.4.4. Strip-Integration with Interpolation

The final, optimal form combines the robustness of strip averaging with the smoothness of continuous integration. We define the *strip-integration functional* as the normalized, weighted area integral over the continuous strip:

$$\mathcal{J}^s(\boldsymbol{\theta}) = \frac{\iint_{\mathcal{S}(\gamma(\boldsymbol{\theta}), \delta)} G\left(\frac{d(\mathbf{x}, \gamma)}{\sigma}\right) \tilde{I}(x, y) dA}{\iint_{\mathcal{S}(\gamma(\boldsymbol{\theta}), \delta)} dA}, \quad (5)$$

where,  $d(\mathbf{x}, \gamma)$  is the minimal Euclidean distance between  $\mathbf{x}$  and the curve. The functional,  $\mathcal{J}^s$ , is both robust to noise (due to area averaging) and continuously differentiable in  $\boldsymbol{\theta}$  (due to the smoothness of  $\tilde{I}$ ,  $G$ , and the parametric curve). It is the functional used in our final algorithm.

### 3. Numerical Implementation

The evaluation of  $\mathcal{J}^l$  and  $\mathcal{J}^s$  requires efficient numerical techniques for interpolation and integration.

#### 3.1. Bicubic Interpolation Scheme

To obtain the  $C^1$  intensity field  $\tilde{I}(x, y)$  from the pixel matrix  $\mathbf{I}[r, c]$ , we employ a compact bicubic formula. For a point  $(x, y)$  in a grid cell with corners at pixel centers, the interpolation is:

$$\tilde{I}(x, y) = \begin{bmatrix} 1 & \Delta(x) & \Delta(x)^2 & \Delta(x)^3 \end{bmatrix} \mathbf{M}^T \mathbf{D}(i(y), j(x)) \mathbf{M} \begin{bmatrix} 1 \\ \Delta(y) \\ \Delta(y)^2 \\ \Delta(y)^3 \end{bmatrix},$$

where  $\Delta(x) = x + \frac{1}{2} - \lfloor x + \frac{1}{2} \rfloor$ ,  $\Delta(y) = y + \frac{1}{2} - \lfloor y + \frac{1}{2} \rfloor$  are local coordinates within a cell of the interpolation grid. The integer-valued indices  $i(y)$  and  $j(x)$ , which locate the pixel, are functions of the coordinates:

$$i(y) = r_0 - \left\lfloor y + \frac{1}{2} \right\rfloor, \quad j(x) = c_0 + \left\lfloor x + \frac{1}{2} \right\rfloor. \quad (6)$$

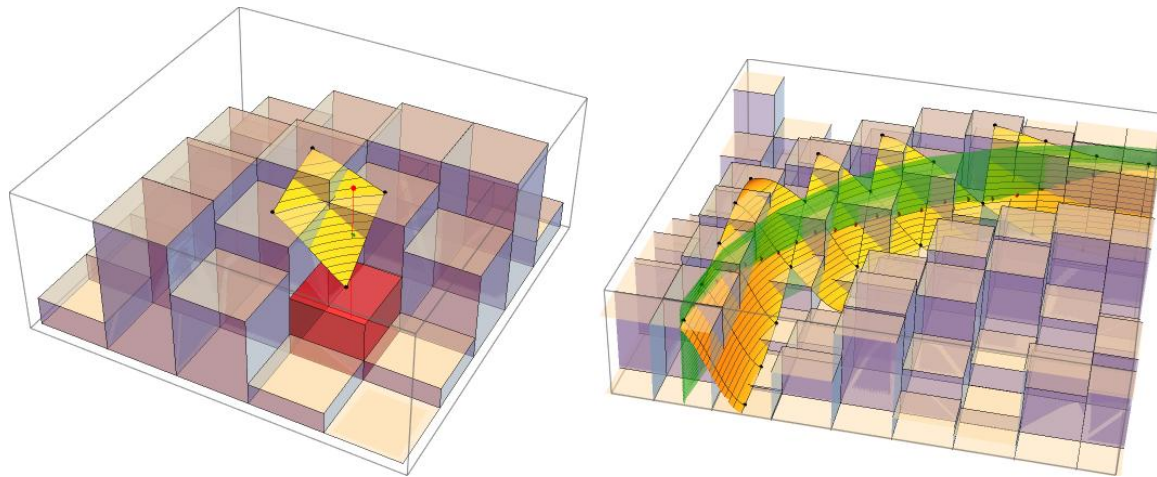
The constant matrix  $M$  and the matrix  $D(i, j)$  (now with  $i, j$  understood as the outputs of (6)) are:

$$M = \begin{pmatrix} 1 & 0 & -3 & 2 \\ 0 & 0 & 3 & -2 \\ 0 & 1 & -2 & 1 \\ 0 & 0 & -1 & 1 \end{pmatrix}, \quad D(i, j) = \begin{pmatrix} f_{ij}^{00} & f_{ij}^{01} & h_{ij}^{00} & h_{ij}^{01} \\ f_{ij}^{10} & f_{ij}^{11} & h_{ij}^{10} & h_{ij}^{11} \\ g_{ij}^{00} & g_{ij}^{01} & q_{ij}^{00} & q_{ij}^{01} \\ g_{ij}^{10} & g_{ij}^{11} & q_{ij}^{10} & q_{ij}^{11} \end{pmatrix}.$$

The sixteen elements of  $D$  are finite-difference approximations of the intensity and its derivatives at the four cell corners, computed from the  $4 \times 4$  neighborhood of the pixel  $(i, j)$ :

$$\begin{aligned} f_{ij}^{pk} &= \mathbf{I}[i - k, j + p], \\ g_{ij}^{pk} &= \frac{1}{2} (\mathbf{I}[i - k, j + p + 1] - \mathbf{I}[i - k, j + p - 1]), \\ h_{ij}^{pk} &= \frac{1}{2} (\mathbf{I}[i - k - 1, j + p] - \mathbf{I}[i - k + 1, j + p]), \\ q_{ij}^{pk} &= \frac{1}{4} (\mathbf{I}[i - k - 1, j + p + 1] - \mathbf{I}[i - k + 1, j + p + 1] \\ &\quad - \mathbf{I}[i - k - 1, j + p - 1] + \mathbf{I}[i - k + 1, j + p - 1]), \end{aligned}$$

for  $p, k \in \{0, 1\}$ . This scheme guarantees that  $\tilde{I}(x, y)$  is a  $C^1$  function across cell boundaries, providing the smoothness required for stable gradient computation. The example of interpolation is shown in Figure 2.



(a) A cell of interpolation grid

(b) A segment of interpolation surface.

Figure 2. Bicubic interpolation

### 3.2. Numerical Quadrature

The line integral (4) is approximated by discretizing the curve into  $n$  segments. For a parametric curve  $(x(\varphi), y(\varphi))$ ,  $\varphi \in [0, 2\pi]$ , we use the trapezoidal rule:

$$\mathcal{J}^l(\theta) \approx \frac{\sum_{k=1}^n \left( \tilde{I}(x_k, y_k) + \tilde{I}(x_{k-1}, y_{k-1}) \right) \sqrt{(x_k - x_{k-1})^2 + (y_k - y_{k-1})^2}}{2 \sum_{k=1}^n \sqrt{(x_k - x_{k-1})^2 + (y_k - y_{k-1})^2}},$$

where  $(x_k, y_k) = (x(\varphi_k) + 1/2, y(\varphi_k) - 1/2)$ . The discretization step  $\Delta\varphi$  is chosen adaptively based on the curve's local curvature to ensure at least one sample per intersected pixel:

$$\Delta = (\dot{x}^2 + \dot{y}^2)^{-1/2}.$$

In case of strip integration the denominator of (5) can be computed more convenient analytically:

$$\iint_{\mathcal{S}(\gamma(\boldsymbol{\theta}), \delta)} dA = \int_{\gamma-\delta}^{\delta} J d\zeta d\varphi,$$

using the expansion for the Jacobian determinant,  $J$ , of the coordinate transformation from  $(\varphi, \zeta)$  (curve parameter and normal offset) to  $(x, y)$ :

$$J = \begin{vmatrix} \dot{x} - \zeta(\ddot{y}\Delta + \dot{y}\dot{\Delta}) & \dot{y} + \zeta(\ddot{x}\Delta + \dot{x}\dot{\Delta}) \\ -\dot{y}\Delta & \dot{x}\Delta \end{vmatrix} = \frac{1}{\Delta} - \zeta(\dot{x}\dot{y} - \ddot{x}\ddot{y})\Delta^2.$$

Substituting this relation into (5) brings it to a form similar to (4):

$$\mathcal{J}^s(\boldsymbol{\theta}) = \frac{2\delta}{L(\boldsymbol{\theta})} \iint_{\mathcal{S}(\gamma(\boldsymbol{\theta}), \delta)} G\left(\frac{d(\mathbf{x}, \gamma)}{\sigma}\right) \tilde{I}(x, y) dA,$$

This integral is approximated by summing contributions from triangular prisms formed by adjacent sample points:

$$\mathcal{J}^s(\boldsymbol{\theta}) \approx \frac{2\delta}{L(\boldsymbol{\theta})} \sum_{k=1}^{n-1} \sum_{p=-P}^{P-1} \left[ V(\mathbf{x}_k^p, \mathbf{x}_k^{p+1}, \mathbf{x}_{k+1}^p) + V(\mathbf{x}_k^{p+1}, \mathbf{x}_{k+1}^{p+1}, \mathbf{x}_{k+1}^p) \right],$$

where  $\mathbf{x}_k^p = \mathbf{x}_k + (p\delta/P) \mathbf{n}_k$  are the coordinates of a point along the normal  $\mathbf{n}_k$  at  $\mathbf{x}_k$ , and  $V(A, B, C)$  is the volume of a triangular prism with base  $ABC$ :

$$V(A, B, C) = \frac{G(d_A, \sigma)\tilde{I}(A) + G(d_B, \sigma)\tilde{I}(B) + G(d_C, \sigma)\tilde{I}(C)}{3} \cdot \mathcal{S}_{ABC}.$$

The first multiplier here is a height of the approximating prism, while the second one is  $ABC$  triangle area, computed through coordinates of its corners:

$$\mathcal{S}(A, B, C) = \frac{1}{2} \begin{vmatrix} A_x & A_y & 1 \\ B_x & B_y & 1 \\ C_x & C_y & 1 \end{vmatrix}.$$

### 3.3. Discrete Differential Operators for Accurate Rasterization

The transition from continuous parametric curves to the discrete pixel grid is a critical step that, if done naively, can introduce systematic errors comparable to the inter-fringe spacing in high-density patterns. To achieve sub-pixel accuracy and topological correctness (4-connectivity), we introduce a unified approach based on integer-valued discrete differential operators. These operators analyze the local geometry of the curve directly in pixel-index space, enabling efficient and exact rasterization of both the curve and the surrounding strip.

### 3.3.1. First-Order Operator: Tangent Direction and Curve Rasterization

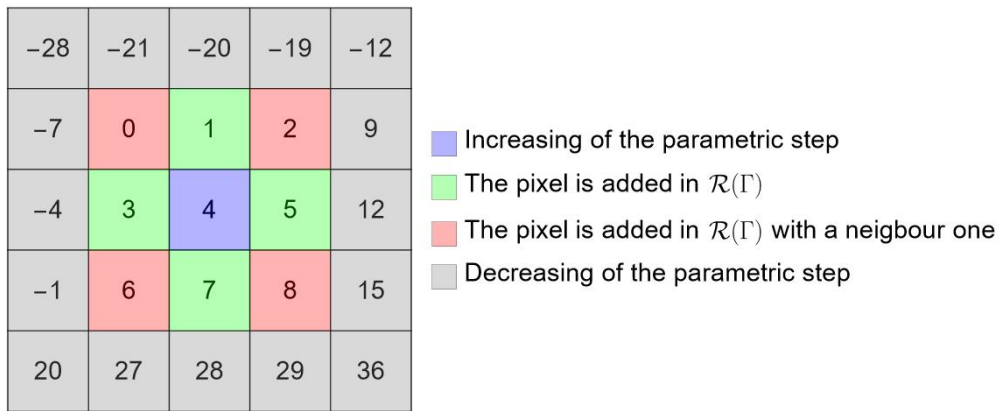
Given two consecutive pixel centers  $P_{i-1} = (r_{i-1}, c_{i-1})$  and  $P_i = (r_i, c_i)$  along a discretized curve, we define the first-order discrete directional operator:

$$D_1(P_{i-1}, P_i) = 4 + 3(\Delta r)^3 + (\Delta c)^3, \quad \text{where } \Delta r = r_i - r_{i-1}, \Delta c = c_i - c_{i-1}.$$

This compact formula serves three purposes simultaneously:

- **Direction defining:** It maps the 9 possible neighbor vectors  $(\Delta r, \Delta c) \in \{-1, 0, 1\}^2$  to unique integers in the range  $[0, 8]$ . This mapping is shown in Figure 3.
- **Neighborhood Filter:** The sum of the cubic terms,  $3(\Delta r)^3 + (\Delta c)^3$  acts as a filter because for  $(\Delta r, \Delta c) \in \{-1, 0, 1\}^2$ , the identity  $3(\Delta r)^3 + (\Delta c)^3 = 3\Delta r + \Delta c$  holds, while if a parameter step is too large and causes a jump to a non-adjacent pixel, the sum makes  $3(\Delta r)^3 + (\Delta c)^3 \geq 8$  (within 2 pixels from  $P_i$ ), flagging an invalid "skip".
- **Adaptive Step Control:** This flag triggers an adaptive adjustment of the parametric step  $\Delta s$ :  $\Delta s \leftarrow \Delta s / k_{\text{down}}$  if a skip is detected;  $\Delta s \leftarrow \Delta s \times k_{\text{up}}$  if  $D_1 = 4$  (meaning the parameter advanced but the pixel did not change). Crucially, we use *asymmetric* coefficients ( $k_{\text{up}} = 1.5, k_{\text{down}} = 1.4$ ) to prevent resonant cycling near stationary points, ensuring robust convergence.

The operator  $D_1$  is the core of the curve rasterization algorithm. Its scheme is shown in the Appendix A (Algorithm 1). The algorithm marches along the continuous curve  $\mathbf{c}(s)$ , adaptively adjusting  $\Delta s$  to produce an ordered, 4-connected list of pixels  $\mathcal{R}$ . A key subtlety is the handling of diagonal moves: when the curve passes through a pixel corner, there are two equally valid 4-connected paths. In such cases, the algorithm use a special variable, which memorizes the choice made at the previous step, ensuring a smooth, consistent rasterization without jagged artifacts.



**Figure 3.** Schematic of the curve rasterization process using a first-order operator  $D_1$ .

### 3.3.2. Second-Order Operator: Local Curvature and Strip Rasterization

Rasterizing a strip of width  $2\delta$  around a curve requires understanding the local curvature to avoid gaps or overlaps in the coverage. Given three consecutive pixel centers  $P_{i-1}, P_i, P_{i+1}$  of the already rasterized curve, we define a second-order operator that classifies the local shape. Let  $\Delta \mathbf{p}_1 = P_i - P_{i-1}$  and  $\Delta \mathbf{p}_2 = P_{i+1} - P_i$ . We define the signs of their components as integers:  $s_x = \text{sgn}(\Delta c_1) + 2 \cdot \text{sgn}(\Delta c_2)$  and  $s_y = \text{sgn}(\Delta r_1) + 2 \cdot \text{sgn}(\Delta r_2)$ . The configuration is then defined as:

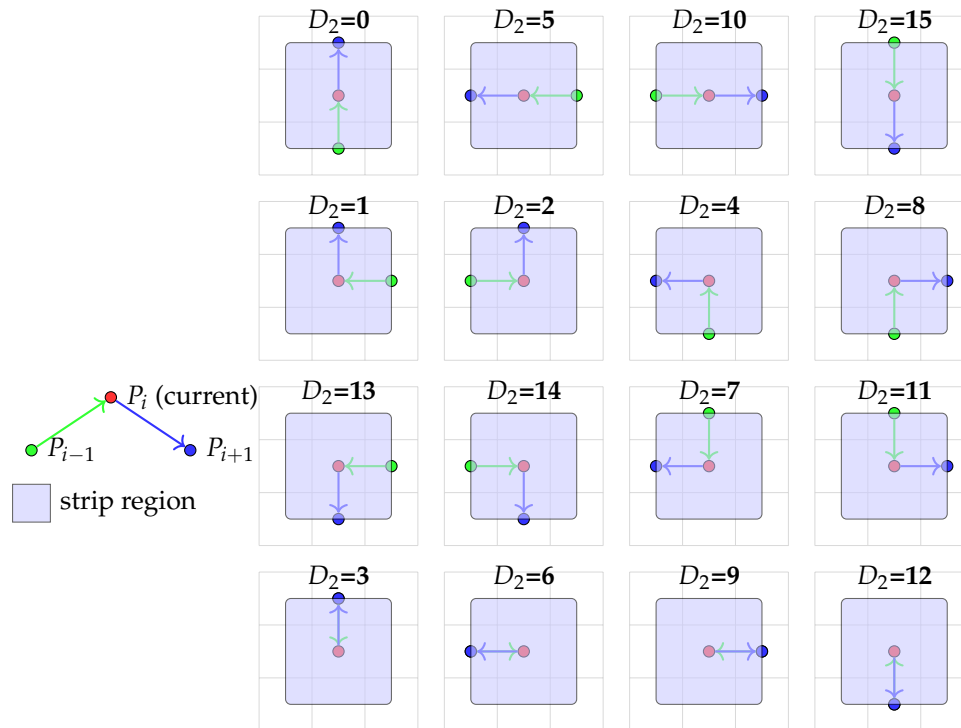
$$D_2 = 4s_y + s_x \in \{-15, -14, \dots, 14, 15\}.$$

After shifting and clamping to the valid range of physically realizable 4-connected triples, we obtain the final formula used in the implementation:

$$D_2(P_{i-1}, P_i, P_{i+1}) = \left\lfloor \frac{15 - c_{i-1} - 3c_i + 4c_{i+1} - 3r_{i-1} - 9r_i + 12r_{i+1}}{2} \right\rfloor \in [0, 15]. \quad (7)$$

The operator  $D_2$  evaluates to one of 16 integer values, each corresponding to a distinct local configuration of the three points (see Figure 4). Geometrically, it approximates the discrete curvature. For example:

- $D_2 \in \{0, 5, 10, 15\}$  indicates nearly linear motion (up, left, right, down), warranting a simple **rectangular strip fill**.
- $D_2 \in \{1, 2, 4, 7, 8, 11, 13, 14\}$  indicates a smooth turn, requiring a **circular sector fill** to cover the curved corner of the strip without gaps.
- $D_2 \in \{3, 6, 9, 12\}$  corresponds to inflection or "return" points, necessitating a **combined fill** (e.g., a half-circle plus strips).



**Figure 4.** The 16 discrete configurations classified by the second-order operator  $D_2$  (Equation 7). For each configuration, the three consecutive curve pixels  $P_{i-1}$  (green),  $P_i$  (red), and  $P_{i+1}$  (blue) are shown with arrows indicating the local direction. The shaded blue area illustrates the corresponding region of the strip that needs to be filled. Configurations are grouped: linear motion ( $D_2 \in \{0, 5, 10, 15\}$ ), smooth turns ( $D_2 \in \{1, 2, 4, 7, 8, 11, 13, 14\}$ ), and return points ( $D_2 \in \{3, 6, 9, 12\}$ ).

Based on the value of  $D_2$ , the strip rasterization algorithm (Algorithm (2) in Appendix A) selects an optimal geometric primitive (line strip or circular arc) and fills the corresponding pixels within distance  $\delta$  from the central pixel  $P_i$ . A distance mask ensures that if a pixel is covered by multiple primitives, the smallest distance to the central curve is retained, and the corresponding Gaussian weight  $\exp(-d^2/(2\sigma^2))$  is assigned. This approach guarantees a skip-free, accurate, and weight-aware rasterization of the continuous strip  $\mathcal{S}(\gamma(\theta), \delta)$ .

### 3.4. Parametric Models for Fringe Contours

The choice of the parametric subspace  $\mathcal{A}_\theta$  is dictated by the expected physics of the deformation. For a broad class of problems involving axisymmetric or nearly axisymmetric bending of plates and membranes, the fringe contours are expected to be closed, nested, quasi-concentric curves. We introduce two related parametric families that efficiently described such shapes while offering sufficient flexibility to capture deviations from ideal geometry.

### 3.4.1. Perturbed Circle Model

The most natural model for quasi-circular fringes is a *perturbed circle*, where the constant radius of a circle is replaced by a periodic function of the polar angle  $\varphi$ . In Cartesian coordinates with an origin at  $(p_x, p_y)$ , this is expressed as:

$$\begin{aligned} x(\varphi) &= p_x + R(\varphi) \cos \varphi, \\ y(\varphi) &= p_y + R(\varphi) \sin \varphi, \end{aligned} \quad (8)$$

where the radius function  $R(\varphi)$  is given by a trigonometric polynomial:

$$R(\varphi) = r \left[ 1 + \sum_{k=1}^n (a_k \cos((k+1)\varphi) + b_k \sin((k+1)\varphi)) \right].$$

Here,  $r > 0$  is a base radius,  $(p_x, p_y)$  is the center, and  $\{a_k, b_k\}_{k=1}^n$  are the perturbation coefficients. Setting all  $a_k = b_k = 0$  recovers a perfect circle of radius  $r$  centered at  $(p_x, p_y)$ . The shift in harmonic indices (using  $(k+1)\varphi$  instead of  $k\varphi$ ) ensures that the lowest-order perturbation already affects the shape, leaving the coefficients  $a_0$  and  $b_0$  implicitly absorbed into  $r$  and the center coordinates.

An equivalent, sometimes computationally convenient, representation expands (8) directly into a Fourier series:

$$\begin{aligned} x(\varphi) &= A_0 + \sum_{k=1}^{n+1} (A_k \cos(k\varphi) + B_k \sin(k\varphi)), \\ y(\varphi) &= C_0 + \sum_{k=1}^{n+1} (C_k \cos(k\varphi) + D_k \sin(k\varphi)), \end{aligned}$$

where the coefficients  $\{A_k, B_k, C_k, D_k\}$  are linear combinations of  $\{r, p_x, p_y, a_k, b_k\}$ . It is simple to show that  $A_0, C_0$ , and the first harmonics are primarily governed by the “coarse” parameters  $r, p_x, p_y$ , while higher harmonics depend on the perturbation coefficients. This structure naturally suggests a two-stage optimization strategy:

1. **Quasi-optimization:** Vary only the coarse parameters  $(r, p_x, p_y)$  to quickly locate the approximate position and scale of the fringe.
2. **Full optimization:** Refine all parameters  $(r, p_x, p_y, \{a_k, b_k\})$  to capture fine shape details.

The total number of parameters is  $m = 3 + 2n$ . In our experiments with plate bending,  $n = 15$  (giving 33 parameters) proved sufficient to accurately represent fringes even under significant nonlinear deformation.

### 3.4.2. Perturbed Ellipse Model

For specimens with pronounced anisotropy or non-axisymmetric boundary conditions, the fringe contours may exhibit a preferred elliptical orientation. The model can be generalized to a *perturbed ellipse* by introducing semi-axes  $a$  and  $b$  ( $a \geq b > 0$ ) and an orientation angle  $\psi$ :

$$\begin{aligned} x(\varphi) &= p_x + \rho(\varphi; a, b, \psi) \left[ 1 + \sum_{k=1}^n (a_k \cos((k+1)(\varphi - \psi)) + b_k \sin((k+1)(\varphi - \psi)) \right] \cos \varphi, \\ y(\varphi) &= p_y + \rho(\varphi; a, b, \psi) \left[ 1 + \sum_{k=1}^n (a_k \cos((k+1)(\varphi - \psi)) + b_k \sin((k+1)(\varphi - \psi)) \right] \sin \varphi, \end{aligned}$$

where

$$\rho(\varphi; a, b, \psi) = \frac{ab}{\sqrt{(b^2 - a^2) \cos^2(\varphi - \psi) + a^2}}.$$

When  $a = b = r$ , this relation reduces to  $\rho = r$ , and the model reverts to the perturbed circle (8). The perturbation is now applied relative to the elliptical base shape and is oriented with it (via the phase shift  $\varphi - \psi$ ). The coarse parameter set for this model is  $(a, b, \psi, p_x, p_y)$ , totaling  $5 + 2n$  parameters.

### 3.4.3. Parameter Constraints and Initialization

To ensure physical plausibility and improve optimization convergence, constraints are applied:  $r, a, b > 0$ ; perturbation coefficients are bounded ( $|a_k|, |b_k| < 0.5$  to prevent self-intersection); and for the ellipse,  $a \geq b$ . The initial guess for the first (innermost) fringe is obtained via a coarse search: the image center is estimated from intensity function cross-sections ( $\varphi = \text{const}$ ), then  $r$  (and  $a, b$  later) is varied until a local extremum of  $\mathcal{J}^s$  is found. For subsequent fringes, the parameters of the previous fringe are scaled outward by a factor slightly greater than one to provide a good starting point for the quasi-optimization step.

These parametric models, combined with the strip-integration functional and the rasterization operators, constitute the proposed skeletonization framework. However, before proceeding to the experimental validation of its performance in Section 4, it is necessary to prepare the intensity matrix to reduce the influence of speckle noise.

### 3.5. Local Intensity Equalization Algorithm

Raw interferograms often exhibit slow spatial variations in background intensity and contrast due to uneven illumination and polarization effects (see Subsection 2.1). To normalize these variations without distorting the fringe positions, we employ a local intensity equalization procedure based on quantile statistics within sliding windows.

#### 3.5.1. Local Quantile Computation

Before processing the intensity matrix it is necessary to remove areas of the image without fringes, that contain no fringes, retaining only the useful region. Furthermore geometric distortion correction must be applied to compensate for the misalignment between the camera lens axis and the object beam. The example of this procedure, which we will call the geometric correction, is shown in Figure 5.

Let  $\mathbf{I}[r, c]$  denote the geometrically corrected intensity matrix of size  $r_{\max} \times c_{\max}$ . For each pixel  $(r_q, c_q)$  considered as a potential window center, we define a local block  $M_{r_q, c_q, q}$  of size  $q \times q$ :

$$M_{r_q, c_q, q}[i, j] = \mathbf{I}[R(r_q + i - \lfloor q/2 \rfloor), C(c_q + j - \lfloor q/2 \rfloor)], \quad i, j = 1, \dots, q,$$

where  $R(k)$  and  $C(k)$  are reflection-padding functions that handle boundary pixels:

$$R(k) = \begin{cases} k, & 1 \leq k \leq r_{\max} \\ 1 - k, & k < 1 \\ 2r_{\max} - k, & k > r_{\max} \end{cases}, \quad C(k) = \begin{cases} k, & 1 \leq k \leq c_{\max} \\ 1 - k, & k < 1 \\ 2c_{\max} - k, & k > c_{\max} \end{cases}.$$

This padding ensures that windows near image boundaries remain fully populated without introducing artificial discontinuities.

For each block  $M_{r_q, c_q, q}$ , we compute two statistics: the lower quantile  $a[r_q, c_q]$  and the upper quantile  $b[r_q, c_q]$  at probability levels  $\nu$  and  $1 - \nu$ , respectively (typically  $\nu = 0.05$ ). These quantiles estimate the local minimum and maximum intensity while being resistant to outlier pixels caused by speckle noise:

$$a[r_q, c_q] = Q_{\nu}(M_{r_q, c_q, q}), \quad b[r_q, c_q] = Q_{1-\nu}(M_{r_q, c_q, q}).$$

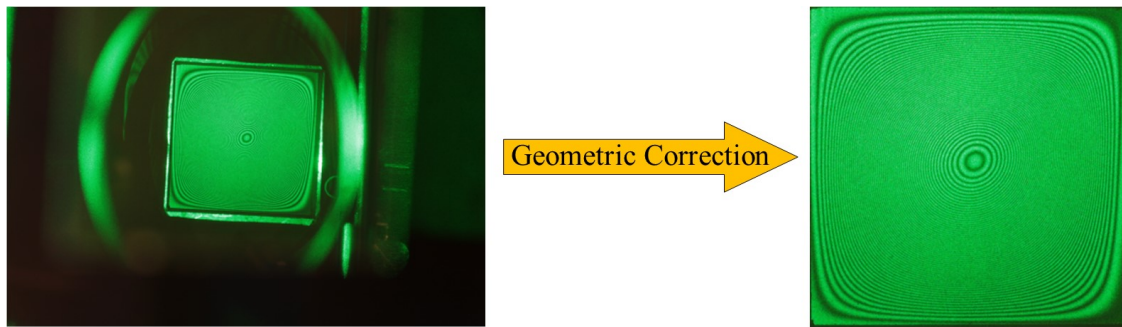


Figure 5. Geometric correction example.

### 3.5.2. Interpolation and Normalization

The quantiles  $a[r, c]$  and  $b[r, c]$  are computed only on a coarse grid with spacing  $d = \lfloor q/2 \rfloor$  to reduce computational cost. To obtain values at every pixel  $(i, j)$ , we use linear interpolation:

$$\hat{a}(i, j) = \text{Int} \begin{pmatrix} a[1, 1] & a[1, d+1] & \dots & a[1, c_{max}] \\ a[d+1, 1] & a[d+1, d+1] & \dots & a[d+1, c_{max}] \\ \dots & \dots & \dots & \dots \\ a[r_{max}, 1] & a[r_{max}, d+1] & \dots & a[r_{max}, c_{max}] \end{pmatrix},$$

$$\hat{b}(i, j) = \text{Int} \begin{pmatrix} b[1, 1] & b[1, d+1] & \dots & b[1, c_{max}] \\ b[d+1, 1] & b[d+1, d+1] & \dots & b[d+1, c_{max}] \\ \dots & \dots & \dots & \dots \\ b[r_{max}, 1] & b[r_{max}, d+1] & \dots & b[r_{max}, c_{max}] \end{pmatrix}.$$

The normalized intensity at each pixel is then obtained by linear rescaling:

$$\tilde{\mathbf{I}}[i, j] = \frac{\mathbf{I}[i, j] - \hat{a}(i, j)}{\hat{b}(i, j) - \hat{a}(i, j)}.$$

This operation maps the local intensity range  $[\hat{a}(i, j), \hat{b}(i, j)]$  approximately to  $[0, 1]$ , effectively compensating for low-frequency inhomogeneities while preserving the high-frequency fringe structure.

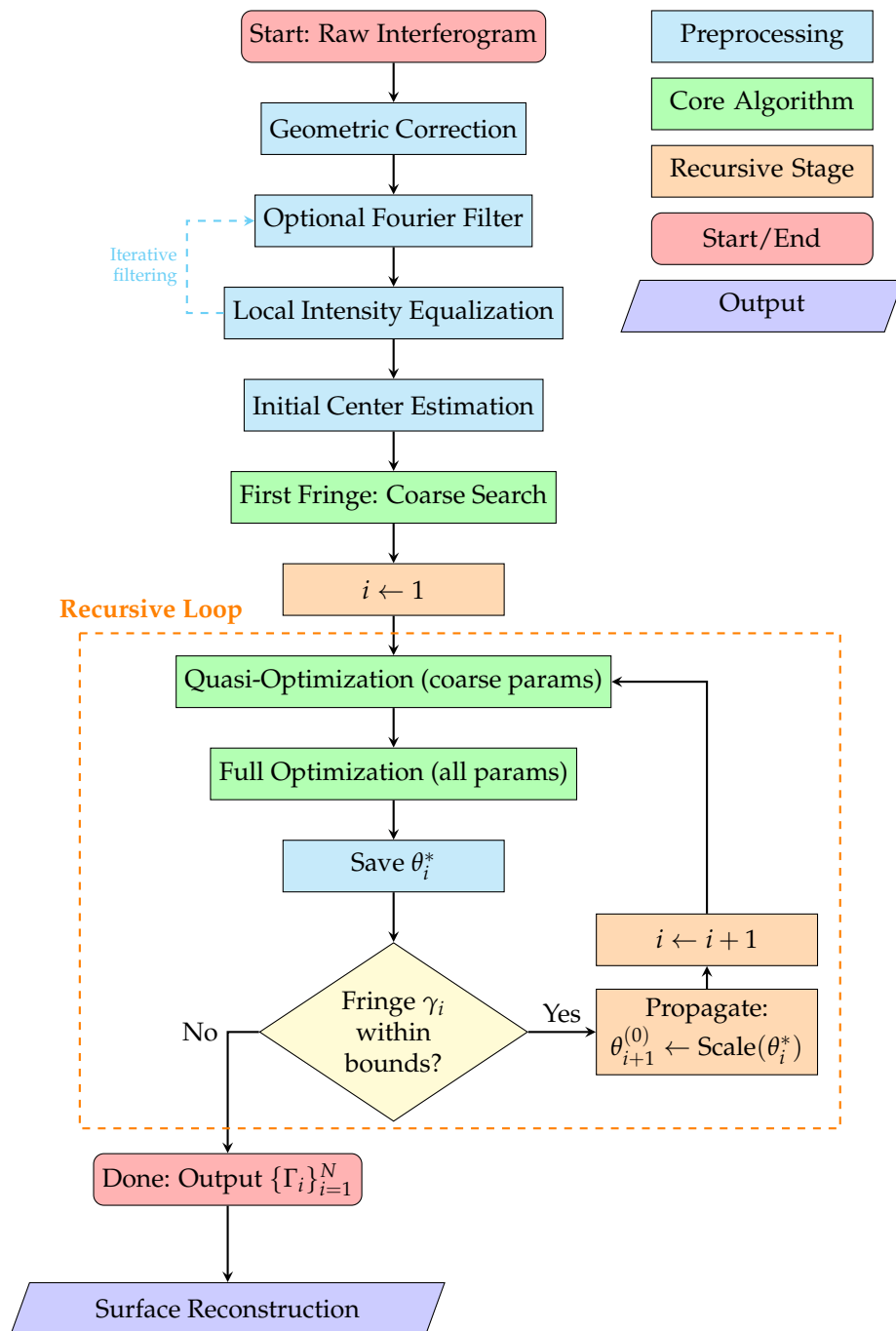
### 3.5.3. Parameter Selection

The window size  $q$  is chosen to be several times larger than the expected inter-fringe distance but smaller than characteristic scales of illumination non-uniformity. In our experiments with square plates,  $q = 31$  pixels (covering about 3–4 fringes) worked well. The quantile level  $\nu = 0.05$  provides a good compromise between rejecting speckle outliers and retaining true fringe extrema. The equalization is applied once after geometric correction and, if needed, repeated after optional Fourier filtering to compensate for global intensity shifts introduced by the filter.

### 3.6. Complete Skeletonization Algorithm

Integrating all components described in the previous subsections, we present the complete procedure for automated fringe skeletonization. The algorithm proceeds recursively from the innermost to the outermost fringe, taking into account the quasi-similarity property. The main steps are illustrated in Figure 6. The algorithm iteratively identifies fringes until one of the three termination conditions is met:

- The last computed intensity integral is equal to zero;
- The last identified fringe lies outside the image boundaries;
- The required (or preset) number of fringes has been identified;



**Figure 6.** Flowchart of the complete skeletonization algorithm. Blue boxes indicate preprocessing steps, green boxes represent core algorithmic stages, and orange boxes show the recursive optimization loop. The dashed blue arrow indicates optional iterative filtering-equalization cycles for severe noise cases.

### 3.7. Synthetic Interferogram Generation

To quantitatively validate the proposed algorithm under controlled conditions, we developed a virtual interferogram generator that produces images with known pattern geometry and physically realistic noise characteristics. The generator taking into account two key aspects: (1) the typical fringe pattern of a bent square plate, and (2) the corruption mechanisms described in Subsection 2.1.

#### 3.7.1. Ideal Fringe Geometry via Morphing

The underlying displacement field is modeled by a family of closed curves that morph smoothly from a circle at the center to a square at the boundary, parameterized by a morphing parameter

$\alpha \in [0, 1]$ . In the first octant ( $\varphi \in [0, \pi/4]$ ), the curve consists of a straight segment and a curved segment that ensures  $C^1$  continuity:

$$\begin{aligned} r_s(\varphi; \alpha) &= \alpha \sec \varphi, & \varphi \in (0, \alpha\pi/4], \\ r_c(\varphi; \alpha) &= \alpha \left[ \frac{2}{\pi(1-\alpha)} \left( \frac{\alpha\pi}{4} - \varphi \right) \left( \frac{\alpha\pi}{4} + \varphi - \frac{\pi}{2} \right) \tan \frac{\alpha\pi}{4} + 1 \right] \sec \frac{\alpha\pi}{4}, & \varphi \in (\alpha\pi/4, \pi/4]. \end{aligned}$$

The full curve over all angles is obtained by symmetric replication. Equivalently, it can be expressed compactly as:

$$r(\varphi; \alpha) = \begin{cases} \alpha r_1(\varphi \bmod \frac{\pi}{2}, \frac{\pi\alpha}{4}), & 0 < \varphi \bmod \frac{\pi}{2} \leq \frac{\pi}{4}, \\ \alpha r_2(\varphi \bmod \frac{\pi}{2}, \frac{\pi}{4}(2-\alpha)), & \frac{\pi}{4} < \varphi \bmod \frac{\pi}{2} \leq \frac{\pi}{2}, \end{cases}$$

where

$$\begin{aligned} r_1(\varphi, \alpha) &= \begin{cases} \left[ \frac{\alpha-\varphi}{\pi-4\alpha} (2\alpha + 2\varphi - \pi) \tan \alpha + 1 \right] \sec \alpha, & \varphi > \alpha \\ \sec \varphi & \varphi \leq \alpha \end{cases}, \\ r_2(\varphi, \alpha) &= \begin{cases} \csc \varphi, & \varphi > \alpha \\ \left[ \frac{\alpha-\varphi}{\pi-4\alpha} (\pi - 2\alpha - 2\varphi) \cot \alpha + 1 \right] \csc \alpha, & \varphi \leq \alpha \end{cases}. \end{aligned}$$

The ideal, noise-free intensity pattern is then defined as a periodic function of  $\alpha$ :

$$I_{\text{ideal}}(x, y) = \sin^2(k\pi\alpha(x, y)),$$

where  $k$  controls the fringe density. The mapping  $(x, y) \mapsto \alpha$  is obtained by numerically solving the transcendental equation:

$$r(\varphi, \alpha) = \sqrt{x^2 + y^2},$$

for  $\alpha$  at each grid point.

### 3.7.2. Physically Motivated Noise Model

The ideal intensity is corrupted by two noise sources derived from the physical model in Subsection 2.1, which presents the general formula for corrupted intensity (1); however the terms describes speckle noisy, were not specified. Accurately defining of these terms is a complicated problem, which was considered in a huge amount of works (e. g. [16–18]). For this paper, it is enough to mark that speckle noise consists of three parts, which related with reflected surface factor, depended on its roughness, spatial and temporal coherence:

$$N_{\text{temp}} = 1 + (2\Delta L\Delta\nu/c)^2, \quad N_{\text{spat}} = 1 + \left( D/\rho_{\text{speckle}} \right)^2,$$

where  $c$  is the speed of light in vacuum,  $\Delta\nu$  is the laser spectral linewidth,  $\Delta L$  the path length difference,  $D$  is the diameter of the illuminated object area and  $\rho_{\text{speckle}}$  is the characteristic speckle size:

$$\rho_{\text{speckle}} \approx \max(\lambda z/D, L_c z/D).$$

Here,  $\lambda$  is the laser wavelength,  $z$  is the distance from object to observation plane and  $L_c$  is the reflected surface roughness parameter. The resulting speckle noise can be modeled as the product of different coherence terms and the surface factor, which is obtained empirically.

In practice, for rapid commutation and parameter studies, it is convenient to employ a simplified combined noise model, constructed as stochastic perturbations of phase difference and amplitude:

$$\tilde{I}(x, y) = I_0(x, y) + kv(x, y) + 2\sqrt{kv(x, y)I_0(x, y)} \cos(\delta\pi),$$

where  $k$  controls the overall noise level and  $v(x, y)$ ,  $\delta(x, y)$  are independent stochastic processes with correlation:

$$\langle f(\mathbf{x})f(\mathbf{x}') \rangle = \left[ 2J_1\left(\pi|\mathbf{x} - \mathbf{x}'|/\rho_{\text{speckle}}\right) / \left(\pi|\mathbf{x} - \mathbf{x}'|/\rho_{\text{speckle}}\right) \right]^2,$$

where  $J_1(\cdot)$  is the Bessel function. This model approximates the statistical properties of speckle while remaining computationally efficient.

### 3.7.3. Error Metrics

For each synthetic image, the exact curves  $\{\gamma_i^{\text{true}}\}$  are known analytically. This allows us to define rigorous error metrics for any skeletonization result  $\{\gamma_i^{\text{res}}\}$ :

$$\begin{aligned} \text{Euclidean metric: } \quad \varepsilon_{\text{Euc}} &= \sqrt{\frac{1}{L^{\text{true}}} \int_0^{2\pi} \left[ (x^{\text{true}}(\varphi) - x^{\text{res}}(\varphi))^2 + (y^{\text{true}}(\varphi) - y^{\text{res}}(\varphi))^2 \right] d\varphi}, \\ \text{Max error: } \quad \varepsilon_{\text{max}} &= \frac{\max_{\varphi \in [0, 2\pi)} \sqrt{(x^{\text{true}}(\varphi) - x^{\text{res}}(\varphi))^2 + (y^{\text{true}}(\varphi) - y^{\text{res}}(\varphi))^2}}{\max_{\varphi \in [0, 2\pi)} \sqrt{(x^{\text{true}}(\varphi))^2 + (y^{\text{true}}(\varphi))^2}}. \end{aligned}$$

These metrics are used in Section 4 to quantitatively compare the performance of different methods.

### 3.8. Implementation Details.

The described skeletonization procedure has been implemented in a custom C++ program. This program is capable of performing all steps of the proposed algorithm (see Figure 6):

1. Geometric correction;
2. Filtering in the frequency domain with different kernels;
3. Intensity equalization, as described in Subsection 3.5;
4. Localization of the image center;
5. Recursive identification of fringes, which includes:
  - (a) Initializing of parametric curve (Subsections 3.4.1 and 3.4.2);
  - (b) Computing of the intensity integral (Subsection 2.4) using numerical quadratures (Subsection 3.2);
  - (c) Defining of the approximation curve's parameters by one of the optimization methods:
    - **Advanced Coordinate Descent with Nesterov's Acceleration** for fastest convergence (typically 20-30 iterations per fringe) (e. g. [19]).
    - **Nelder – Mead simplex method** when derivative information is not readily available (robust but slightly slower).
    - **Conjugate Gradient method** for highest precision in final refinement (used only when the highest accuracy is required, despite longer computation time).

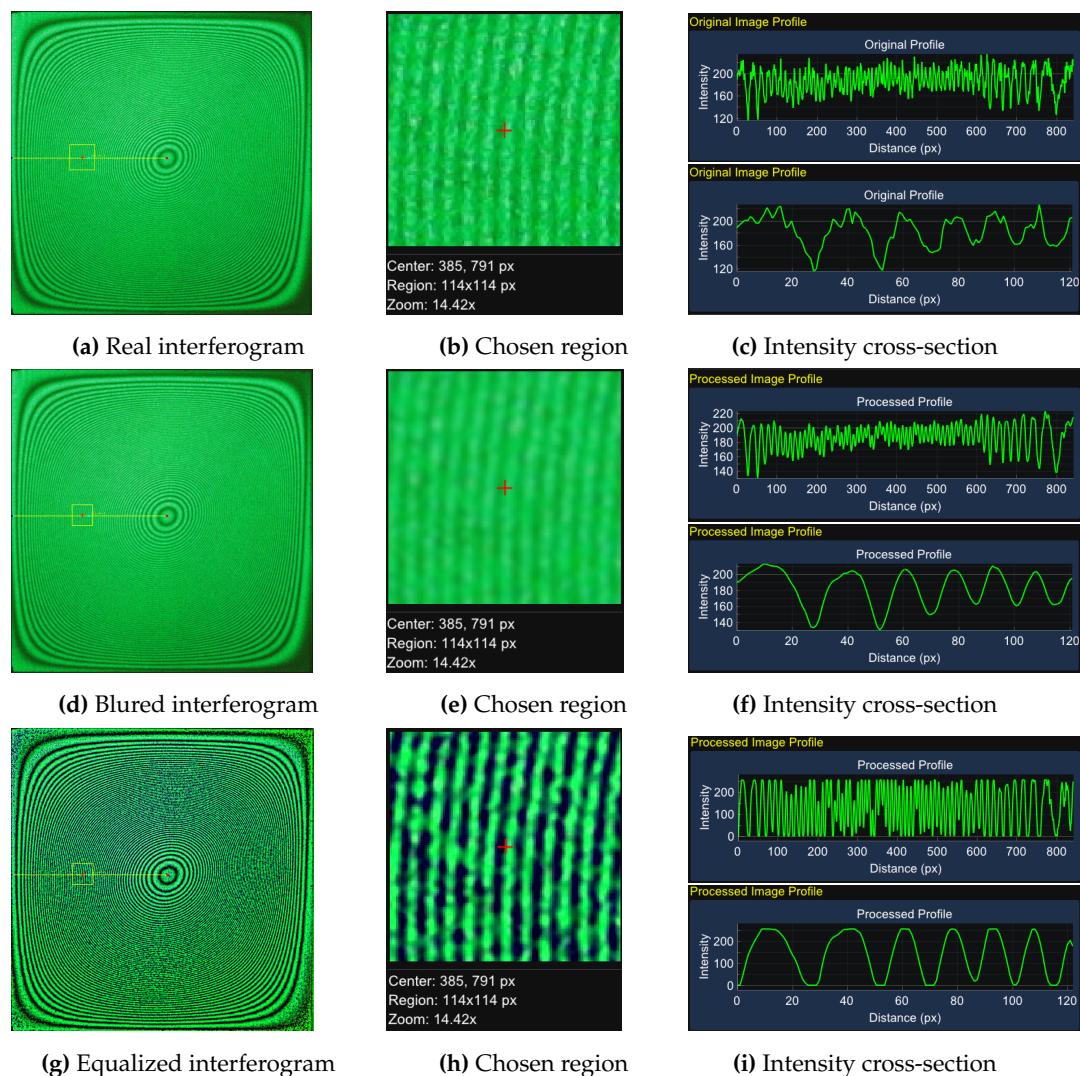
An example of the preprocessing of a real interferogram, performed using the developed program, is shown in Figure 7.

Apart from that, the program contains a set of standard tools for working with images and includes a custom generator of synthetic fringe patterns, a described in Subsection 3.7. The generator accepts the following control parameters:

- Fringe density  $k$  (number of fringes across the field);
- Speckle noise, dependent on surface roughness parameter and the laser spectral linewidth;
- Noise models the influence of non-uniform illumination;

For each generated image, the exact curves  $\{\gamma_i^{\text{true}}(\alpha = i/k)\}$  are known analytically. This allows for the quantification of the algorithm errors.

The main computational cost of the algorithms associated with the strip-integrating procedure. Each evaluation of  $\mathcal{J}^s(\theta)$  requires  $O(M)$  operations, where  $M$  is the number of quadrature points in the strip discretization. With  $N$  fringes and an average of  $K$  optimization iterations per fringe, the total cost is  $O(KNM)$ . In practice, for a  $1000 \times 1000$  image with 100 fringes, the algorithm completes in 2 – 3 minutes on a standard desktop CPU, making it suitable for practical laboratory use.



**Figure 7.** The preprocessing of a real interferogram. Rows (top to bottom): (a, b, c) original image; (d, e, f) image after Gaussian filtering (blur); (g, h, i) image after intensity equalization. Columns (left to right): (a, d, g) full image; (b, e, h) magnified chosen region; (c, f, i) intensity cross-section, showing the original function (top) and the function after removal of false extrema (bottom).

#### 4. Validation on Synthetic Interferograms

We compared the efficiency of fringe identification with different variants of the intensity integral, using the metrics defined in Subsection 3.7.3. Figure 8 demonstrates that slightly noisy images, both methods have similar performance, whereas for highly noisy patterns the stripe-integration method has a significant advantage over the line-integration. Moreover, as it is shown in Figure 9, the stripe-integration method is capable of identifying fringes in images with noise levels as high as 95%, where it is hardly possible to differ the fringes even visually.

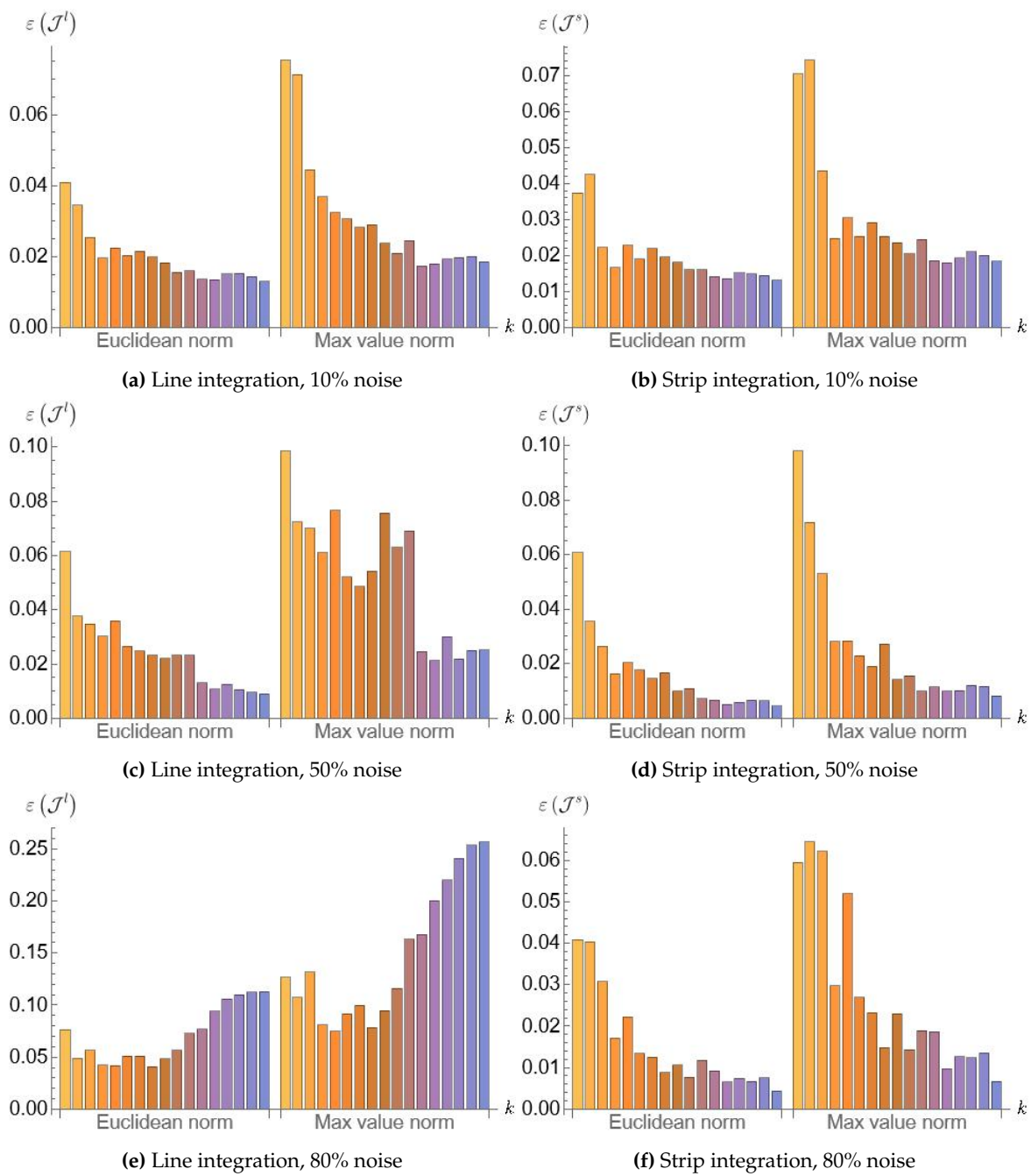


Figure 8. A Comparison of identification efficiency for different integration methods.

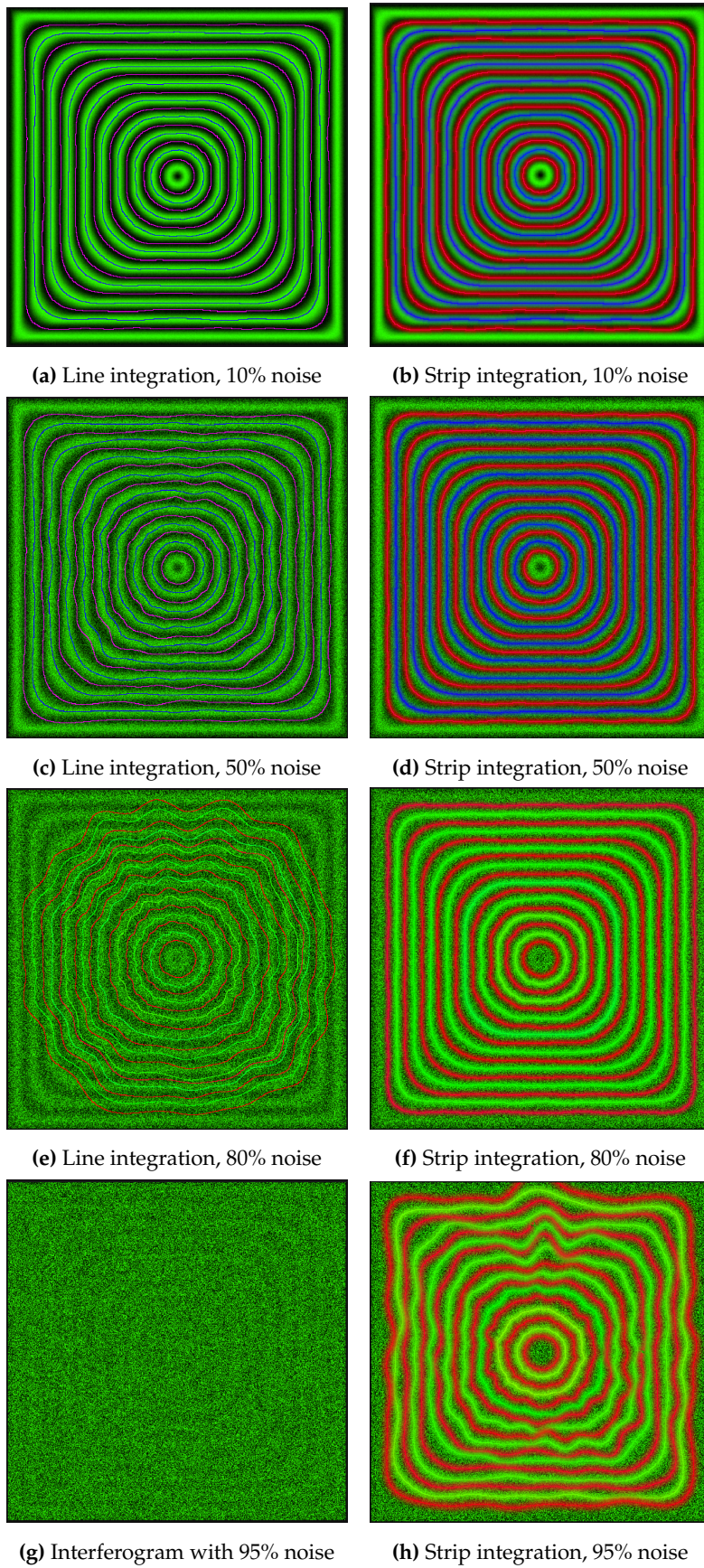
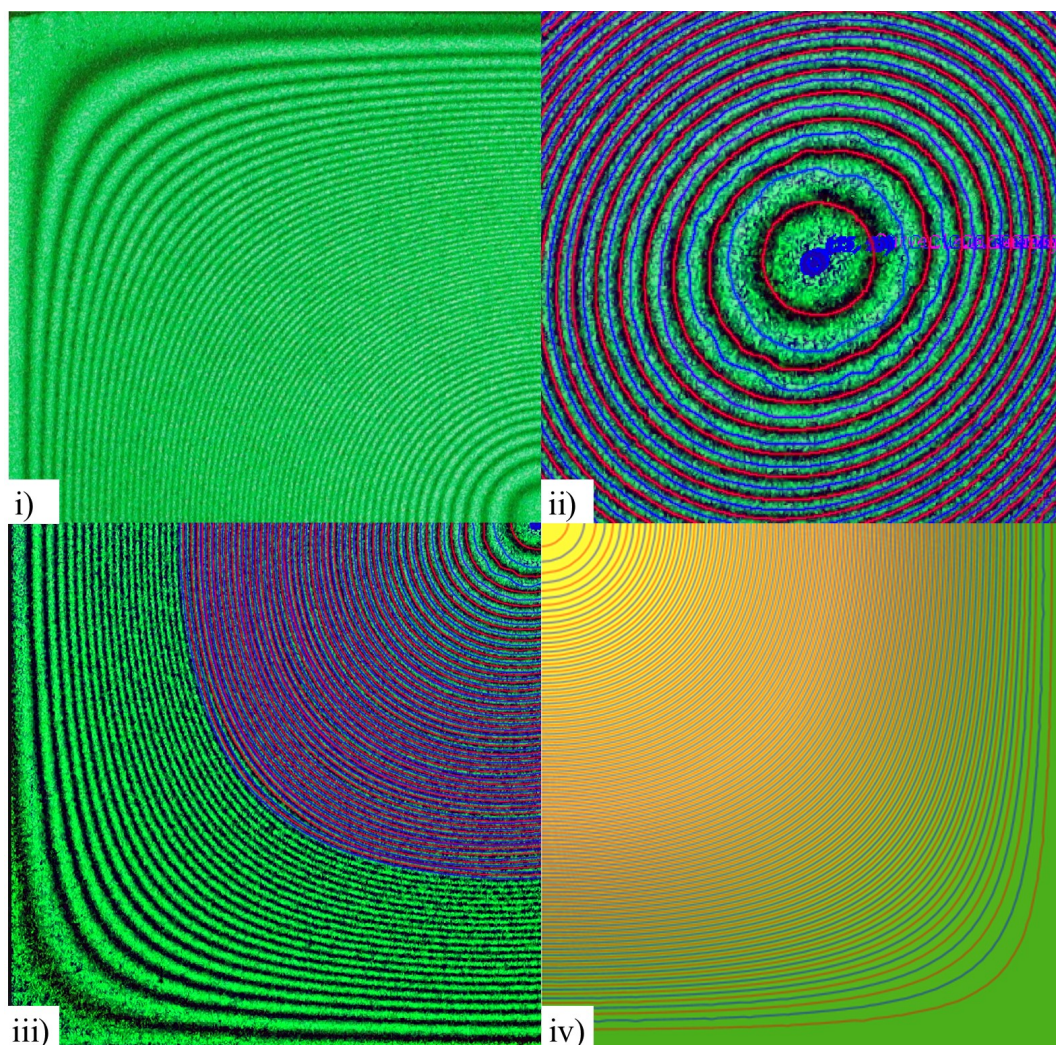


Figure 9. Fringe identification from synthetic interferogram.

## 5. Application to Real High-Density Interferograms

To validate the proposed algorithms, it was tested on a real hologram, obtained from a bending experiment on a square plate. The plate (copper, side length 6 cm, thickness 184  $\mu\text{m}$ ) was subjected to a uniform transverse load in a holographic interferometry setup (Leith – Upatnieks off-axis scheme with a solid-state laser,  $\lambda = 532 \text{ nm}$ ). Interferograms were recorded for several load increments. The most challenging case, with a deflection generating over 100 fringes (shown in Figure 10i), was selected for analysis.

The set of identified fringes (red and blue lines in figures 10ii) and 10iii)) was compared with the results (Figure 10iv)), obtained from theoretical modeling, presented in [15]. As shown in Figure 10, the identified fringes are in good agreement with isolines of deformed surface, demonstrating the acceptable accuracy of the proposed procedure.



**Figure 10.** Validation of the proposed procedure on a real hologram. i) Region of the original hologram. ii) Magnified central region of the processed image. iii) Full processed image, displaying both the identified fringes (red and blue lines) and the intensity-equalized fringe pattern. iv) Deformed surface of the plate obtained from independent theoretical modeling for comparison.

## 6. Conclusion

This paper presents a novel, robust method for skeletonizing fringe patterns in holographic interferometry. By constraining the solution to a physics-informed parametric subspace and employing a strip-integration functional, the method achieves high accuracy and stability in the presence of strong speckle noise, where conventional techniques may fail. Quantitative validation on synthetic data demonstrated a significant reduction in error compared to baseline methods. The practical utility was

confirmed by successfully processing a real interferogram with over 100 fringes, the results of which closely matched those of an independent theoretical model.

The main limitation of the algorithm is the need to select an appropriate parametric model  $\mathcal{A}_\theta$  beforehand. Thus, the method is most effective when the fringe family topology is known to be a set of nested, simply connected curves. It may be less effective for patterns featuring fringe bifurcations, interruptions, or multiple disconnected families without significant modifications. In the future, the concept of quasi-similarity and recursive search may be modified to a more powerful paradigm for handling complex topologies.

## Appendix A: Rasterization Algorithms Pseudocode

---

### Algorithm 1: Adaptive Curve Rasterization using First-Order Operator $D_1$

---

**Input:** Parametric functions  $x(s), y(s)$ ; domain  $[s_{\text{start}}, s_{\text{end}}]$ ; number of steps  $N_0$   
**Output:** Ordered list of pixels  $\mathcal{R}$  representing the 4-connected curve

```

 $\Delta s \leftarrow (s_{\text{end}} - s_{\text{start}}) / N_0;$ 
 $s \leftarrow s_{\text{start}};$ 
 $(r_{\text{prev}}, c_{\text{prev}}) \leftarrow \text{pixel containing } (x(s), y(s));$ 
 $\mathcal{R}.\text{append}((r_{\text{prev}}, c_{\text{prev}}));$ 
 $k_{\text{up}} \leftarrow 1.5, k_{\text{down}} \leftarrow 1.4$  // Asymmetric coefficients
legacy  $\leftarrow$  true;
while  $s \leq s_{\text{end}}$  do
     $s_{\text{next}} \leftarrow \min(s + \Delta s, s_{\text{end}});$ 
     $(r, c) \leftarrow \text{pixel containing } (x(s_{\text{next}}), y(s_{\text{next}}));$ 
     $\Delta r \leftarrow r - r_{\text{prev}}, \Delta c \leftarrow c - c_{\text{prev}};$ 
     $d \leftarrow 4 + 3(\Delta r)^3 + (\Delta c)^3;$  // Compute  $D_1$ 
    if  $d \in \{0, 1, 2, 3, 5, 6, 7, 8\};$  // Valid move to a neighboring pixel
    then
        if  $d$  corresponds to a diagonal move (0,2,6,8) then
            // Handle corner ambiguity using legacy strategy
            if legacy then
                | Add appropriate corner pixel to  $\mathcal{R}$ ;
            else
                | Add alternative corner pixel to  $\mathcal{R}$ ;
            Add  $(r, c)$  to  $\mathcal{R}$ ;
            Update  $(r_{\text{prev}}, c_{\text{prev}}) \leftarrow (r, c)$ ;
             $s \leftarrow s_{\text{next}};$ 
            Update legacy based on chosen corner;
        else if  $d = 4$  then
            |  $\Delta s \leftarrow \Delta s \times k_{\text{up}};$  // Pixel didn't change, increase step
        else
            |  $\Delta s \leftarrow \Delta s / k_{\text{down}};$  // Skip detected, decrease step
        if  $\Delta s$  below threshold then
            | break;
    return  $\mathcal{R}$ ;
```

---

**Algorithm 2:** Strip Rasterization using Second-Order Operator  $D_2$ 


---

**Input:** Ordered pixel list  $\mathcal{R}$  of the curve; strip half-width  $w$ ; Gaussian scale  $\sigma$   
**Output:** Set of pixels  $\mathcal{S}$  in the strip with weights  $G(d)$

$\mathcal{S} \leftarrow \emptyset;$   
Initialize distance matrix  $\mathbf{D}_{\min} \leftarrow \infty;$   
 $n \leftarrow \text{length}(\mathcal{R});$   
**for**  $i \leftarrow 0$  **to**  $n - 1$  **do**

$P_{\text{prev}} \leftarrow \mathcal{R}[(i - 1) \bmod n];$   
 $P_{\text{curr}} \leftarrow \mathcal{R}[i];$   
 $P_{\text{next}} \leftarrow \mathcal{R}[(i + 1) \bmod n];$   
 $d \leftarrow \lfloor (15 - c_{\text{prev}} - 3c_{\text{curr}} + 4c_{\text{next}} - 3r_{\text{prev}} - 9r_{\text{curr}} + 12r_{\text{next}}) / 2 \rfloor;$  // Compute  $D_2$

**switch**  $d$  **do**

**case**  $0,5,10,15$  // Linear motion  
**do**  
└ Add rectangular strip of width  $w$  perpendicular to the segment  $(P_{\text{prev}}, P_{\text{curr}})$  to  $\mathcal{S};$

**case**  $1,2,4,7,8,11,13,14$  // Smooth turn  
**do**  
└ Add circular sector of radius  $w$  centered at  $P_{\text{curr}}$  covering the turn angle to  $\mathcal{S};$

**case**  $3,6,9,12$  // Return point  
**do**  
└ Add combination of half-circle and rectangular strips to  $\mathcal{S};$

**foreach** pixel  $Q$  added in this step **do**  
└  $d_{\text{curve}} \leftarrow$  distance from  $Q$  to the curve segment;  
└ **if**  $d_{\text{curve}} < \mathbf{D}_{\min}[Q]$  **then**  
└└  $\mathbf{D}_{\min}[Q] \leftarrow d_{\text{curve}};$   
└└  $\text{weight}[Q] \leftarrow \exp(-d_{\text{curve}}^2 / (2\sigma^2));$   
└└ Add/update  $Q$  in  $\mathcal{S}$  with computed weight;

**return**  $\mathcal{S};$

---

**Author Contributions:** Conceptualization, methodology, software, validation, writing – original draft preparation, S.L.; resources, investigation, A.D.; writing – review and editing, S.L. and A.D. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was supported by the Russian Science Foundation (grant No 23-19-00866).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Data are contained within the article.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Kobayashi, A. *Handbook on experimental mechanics*; Prentice-Hall, 1987.
2. Vest, C. *Holographic Interferometry*; Wiley, 1979.
3. Malacara, D.; Servin, M.; Malacara, Z. *Interferogram analysis for optical testing*; Taylor & Francis, 2005.
4. Kreis, T. *Handbook of holographic interferometry: optical and digital methods*; John Wiley & Sons, 2006.
5. Marr, D.; Hildreth, E. Theory of edge detection. *Proceedings of the Royal Society of London. Series B. Biological Sciences* **1980**, *207*, 187–217.
6. Canny, J. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence* **2009**, pp. 679–698.

7. Shen, J.; Castan, S. An optimal linear operator for step edge detection. *CVGIP: Graphical models and image processing* **1992**, *54*, 112–133.
8. Xu, C.; Prince, J.L. Gradient vector flow: A new external force for snakes. In Proceedings of the Proceedings of IEEE computer society conference on computer vision and pattern recognition. IEEE, 1997, pp. 66–71.
9. Xu, C.; Prince, J.L. Snakes, shapes, and gradient vector flow. *IEEE Transactions on image processing* **1998**, *7*, 359–369.
10. Kass, M.; Witkin, A.; Terzopoulos, D. Snakes: Active contour models. *International journal of computer vision* **1988**, *1*, 321–331.
11. Li, B.; Acton, S.T. Active contour external force using vector field convolution for image segmentation. *IEEE transactions on image processing* **2007**, *16*, 2096–2106.
12. Tang, C.; Lu, W.; Cai, Y.; Han, L.; Wang, G. Nearly preprocessing-free method for skeletonization of gray-scale electronic speckle pattern interferometry fringe patterns via partial differential equations. *Optics letters* **2008**, *33*, 183–185.
13. Tang, C.; Ren, H.; Wang, L.; Wang, Z.; Han, L.; Gao, T. Oriented couple gradient vector fields for skeletonization of gray-scale optical fringe patterns with high density. *Applied optics* **2010**, *49*, 2979–2984.
14. Li, Y.H.; Chen, X.J.; Qu, S.L.; Luo, Z.Y. Algorithm for skeletonization of gray-scale optical fringe patterns with high density. *Optical Engineering* **2011**, *50*, 087003–087003.
15. Lychev, S.; Digilov, A.; Djuzhev, N. Galerkin-Type Solution of the Föppl–von Kármán Equations for Square Plates. *Symmetry* **2024**, *17*, 32.
16. Eichhorn, N.; Osten, W. An algorithm for the fast derivation of line structures from interferograms. *Journal of Modern Optics* **1988**, *35*, 1717–1725.
17. Goodman, J.W. *Speckle phenomena in optics: theory and applications*; Roberts and Company Publishers, 2007.
18. Dainty, J.C. *Laser speckle and related phenomena*; Vol. 9, Springer science & business Media, 2013.
19. Wang, Q.; Li, W.; Bao, W.; Zhang, F. Accelerated randomized coordinate descent for solving linear systems. *Mathematics* **2022**, *10*, 4379.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.