



A NEW SVTV-STOKES MODEL WITH BAYESIAN OPTIMIZATION FOR COLOR IMAGE DENOISING

YITAO SHENG ^{1,2} AND ZHIGANG JIA ^{*1,3}

¹School of Mathematics and Statistics,
Jiangsu Normal University, Xuzhou 221116, China

²Academy of Mathematics and Systems Science,
Chinese Academy of Sciences, Beijing 100190, China

³School of Big Data,
Fuzhou University of International Studies and Trade, Fuzhou 350202, China

(Communicated by Yifei Lou)

ABSTRACT. A new model is proposed for color image denoising, with combining tangential field smoothing techniques, image reconstruction techniques, and Bayesian optimization methods. First, the smooth tangential vector field method is used to process the color image, and the “texture” information of the denoised image is obtained by using the anisotropic TV-Stokes model to effectively improve the smoothness of the image. Second, owing to the regularization characteristics of the SVTV model, the image is further refined after preliminary processing, with colors smoothed and details preserved. This process helps eliminate noise to the maximum extent while maintaining the natural appearance of the image. Bayesian optimization methods are then used to optimize the parameters to improve the algorithm’s performance. Numerical experimental results demonstrate that the proposed method can effectively capture details in color images, exhibiting superior denoising effects. The new model proposed in this paper brings innovation and effective solutions to the field of color image denoising, offering important insights and guidance for research and applications in image processing. This model is expected to provide reliable technical support for enhancing image quality and information extraction in practical applications.

1. Introduction. In the field of digital image processing, color image processing is a highly significant and prominent issue from a mathematical academic perspective. Unlike traditional grayscale images, color images possess additional color information and correlations, making traditional grayscale image processing methods challenging to directly apply to color images. In recent years, researchers have been

2020 *Mathematics Subject Classification.* Primary: 68U10, 65K10; Secondary: 65F22.

Key words and phrases. SVTV-Stokes model, color image denoising, inverse problem, machine learning, Bayesian optimization.

This work is supported in part by the National Key Research and Development Program of China under grant 2023YFA1010101; the National Natural Science Foundation of China under grants 12171210, 12090011, 12271467 and 11771188; the “QingLan” Project for Colleges and Universities of Jiangsu Province (Young and middle-aged academic leaders); the Major Projects of Universities in Jiangsu Province under grant 21KJA110001; and the Natural Science Foundation of Fujian Province of China under grant 2022J01378.

*Corresponding author: Zhigang Jia.

exploring more effective processing methods specifically tailored for color images. Due to the processing of color images requiring the processing of multiple channels, such as RGB, CMYK, HSV, etc., while simultaneously considering color information preservation and coupling between color channels, this increases the complexity of the problem.

Assume that for a given image z , the following model holds:

$$z = Ku + \eta,$$

where K is the blur operator (usually a convolution), u is the restored image, and η represents the additive noise. Image processing involves using various techniques to reduce or eliminate undesired pixel value fluctuations in order to improve the quality of the image. On the other hand, image processing involves utilizing mathematical methods, such as deconvolution to maximize the restoration of image clarity and details. This type of problem typically falls within the realm of inverse problems.

Color image processing fundamentally constitutes a class of ill-posed problems. This topic often involves complex large-scale matrix operations, high-dimensional discrete matrix linear equations, the discretization of partial differential equations, and concepts of ill-posedness in solutions. To accurately recover the image u from the noisy image z , we must utilize a series of regularization methods.

In early advancements of color image processing, a significant contribution was made with the introduction of a new total variation (TV) norm specifically designed for vector-valued functions [2]. This novel TV norm is tailored to restore color images effectively by addressing several key aspects: it does not penalize image discontinuities, thereby preserving edges; it maintains rotational invariance in the image space; and it simplifies to the conventional TV norm when applied to scalar-valued functions. These properties make it particularly suitable for handling the complexities of color images. This pioneering work laid the groundwork for subsequent advancements in the field of color image processing.

To achieve fast and efficient algorithms, Bresson and Chan [3] proposed a dual form based on the vector total variation (VTV) norm. The VTV norm is a method used to measure gradient changes in images, particularly effective in preserving image edges and detail information. The core idea of this algorithm is to minimize the VTV norm to achieve regularization of the image. Compared with traditional methods, this algorithm is more precise in minimizing the VTV norm, thus better preserving the structure and details of the image. Additionally, the implementation of this algorithm is relatively simple, and it converges quickly, making it suitable for a wide range of color image processing tasks, including denoising, image inpainting, and image decomposition. Six years later, Ono and Yamada [13] introduced a new regularization model for color images, namely decorrelated vector total variation (D-VTV). Unlike the VTV model, the D-VTV model separately measures the discrete gradients of the luminance component and the chrominance components, significantly reducing unnecessary uneven color effects. Additionally, the authors developed a high-order generalized model of D-VTV called decorrelated vector total generalized Variation (D-VTGV) to avoid the staircase effect observed when using VTV. A notable characteristic of D-VTGV is its ability to efficiently minimize the objective function involving it through the primal-dual splitting method.

In recent years, many powerful models based on geometry methods and partial differential equations (PDE) have been developed. Liu et al. [11] proposed a model based on Euler's elastic energy, which extends the single-channel Euler's

elastic model and TV model to address color image processing and multi-channel data processing. The model aims to minimize the curvature of color images by introducing the Polyakov action model. To solve the minimization problem of the nonlinear geometric model efficiently, they proposed an operator splitting method. This method decouples the nonlinearities by introducing vector-valued and matrix-valued variables, transforming the problem into solving steady states of related initial value problems. The initial value problem is then solved through time splitting in three fractional steps, where each sub-problem has a closed-form solution or can be solved using fast algorithms. Duan et al. [5] introduced a variant of the Beltrami framework, which successfully achieves color image processing by treating color images as manifolds embedded in a five-dimensional space—the color space. However, due to the highly nonlinear nature of the relevant models, optimizing related optimization problems becomes numerically challenging. To address this, they proposed an operator splitting method for the Beltrami regularization model. This method optimizes the optimality conditions associated with minimizing the Beltrami regularization function, and solves the gradient flow problem through an operator splitting scheme involving three steps. Tai et al. [14] proposed a simple yet efficient Newton-Krylov solver for nonlinear partial differential equations in the ROF model and minimal surface model. By introducing block diagonal preconditioners in appropriate Sobolev spaces, this method maintains computational efficiency even when dealing with complex models. Theoretical analysis and experimental results of these methods demonstrate their effectiveness and robustness in the field of color image processing.

In this paper, we focus specifically on image denoising, assuming the blur operator $K = I$. And, we further develop a new color image denoising model that combines the geometry metric, high-order PDEs, and quaternion representation. This is to provide an innovative solution for color image processing to achieve more accurate and comprehensive denoising by combining the two-step TV-Stokes model [8] with the saturation value total variation (SVTV) model [9]. By combining the edge recovery capability of the TV-Stokes model and the color TV regularization characteristic of the SVTV model, we can effectively eliminate noise while preserving the structure and details of color images. Our goal is to further improve the denoising effect and maintain the robustness and the practicality of the model when dealing with color images. This fusion method is designed to fully utilize the strengths of both models to provide a more comprehensive and efficient solution for color image processing tasks, potentially achieving significant performance improvements in practical applications.

The contribution of this paper lies in:

- Introduction of a new SVTV-Stokes model for color image denoising.
- Incorporation of Bayesian optimization for enhancing algorithm performance.
- Efficient parameter decomposition for reducing computational complexity.
- Use of iterative PDE solvers for faster convergence and improved precision.
- Superior denoising performance demonstrated by quantitative and visual results.

This paper is organized as follows: In Section 2, we introduce the primary knowledge, including the TV model and the SVTV model. In Section 3, we derive the details of the two-step SVTV-Stokes denoising model, including the iterative computation methods and algorithms used. In Section 4, we present numerical experiments and performance evaluations, comparing the proposed model with other

denoising models using three metrics. In Section 5, we provide concluding remarks summarizing our findings, highlighting the effectiveness of the proposed model and suggesting future research directions.

2. Primary knowledge. We consider U as a mapping from a region with Lipschitz boundary Ω to u , denoted as $U: \Omega \rightarrow u$. If we choose $(\sigma^1, \sigma^2) \in \Omega$, we have $U(\sigma^1, \sigma^2) = u = (u_1(\sigma^1, \sigma^2), u_2(\sigma^1, \sigma^2), u_3(\sigma^1, \sigma^2))$.

The normal vector of u is defined as

$$\vec{n} = (n_{ij}) = \nabla u = (\nabla u_1, \nabla u_2, \nabla u_3) = (\vec{n}_1, \vec{n}_2, \vec{n}_3) = \begin{pmatrix} u_{11}, u_{21}, u_{31} \\ u_{12}, u_{22}, u_{32} \end{pmatrix},$$

with u_{ij} representing the derivative of the i -th channel of u with respect to the j -th spatial direction where $j = 1$ corresponds to the x -direction ($\frac{\partial u_i}{\partial x}$) and $j = 2$ corresponds to the y -direction ($\frac{\partial u_i}{\partial y}$). The tangential vector is defined as

$$\vec{\tau} = (\tau_{ij}) = \nabla^\perp u = (\nabla^\perp u_1, \nabla^\perp u_2, \nabla^\perp u_3) = (\vec{\tau}_1, \vec{\tau}_2, \vec{\tau}_3) = \begin{pmatrix} u_{12}, u_{22}, u_{32} \\ -u_{11}, -u_{21}, -u_{31} \end{pmatrix},$$

where $\nabla^\perp u_i$ is obtained by rotating the gradient of u_i 90° in the 2D plane. This rotation swaps the x - and y - directions and introduces a sign change in the x -direction. Mathematically, we have $\tau_{i1} = u_{i2} = n_{i2}$ and $\tau_{i2} = -u_{i1} = -n_{i1}$.

One can derive that

$$\nabla \times \vec{n} = \nabla \times (\vec{n}_1, \vec{n}_2, \vec{n}_3) = 0,$$

and

$$\nabla \cdot \vec{\tau} = \nabla \cdot (\vec{\tau}_1, \vec{\tau}_2, \vec{\tau}_3) = 0.$$

This is also referred to as the incompressibility condition on each color channel, denoting the space $\Psi = \left\{ \vec{\tau} \mid \vec{\tau} = (\vec{\tau}_1, \vec{\tau}_2, \vec{\tau}_3) \in (BV(\Omega; \mathbb{R}^2))^3, \nabla \cdot \vec{\tau} = 0 \right\}$.

We introduce two classic denoising models. The TV denoising model is a popular method for reducing noise in gray-scale images while preserving edges. This model, based on minimizing the TV of the image to retain sharp edges and important features, can be formulated as the following optimization problem [6]:

$$\min_u \int_{\Omega} |\nabla u| dx dy + \frac{\mu}{2} \int_{\Omega} (u - z)^2 dx dy.$$

To extend this model to color images, we consider each color channel (e.g., RGB) separately. Thus, the functional becomes [2]

$$\min_u \int_{\Omega} \sum_{i=1}^3 |\nabla u_i| dx dy + \frac{\mu}{2} \int_{\Omega} \sum_{i=1}^3 (u_i - z_i)^2 dx dy,$$

where u_i , and z_i represent the i -th color channel of the denoised and noisy images, respectively. The first term sums the TV over all color channels, promoting smoothness while preserving edges. The second term ensures that the denoised image remains close to the original noisy image in all color channels, controlled by the parameter λ . And, we can easily derive the discrete form of the model proposed above:

$$\min_u \sum_{i=1}^3 \sum_{k=1}^M \sum_{j=1}^N |\nabla u_i|_{kj} + \frac{\lambda}{2} \sum_{i=1}^3 \sum_{k=1}^M \sum_{j=1}^N (u_i - z_i)_{kj}^2,$$

where M and N represent the number of rows and columns of the image, respectively, corresponding to the height and width of the image.

For color image denoising, Jia, Ng, and Wang [9] proposed the SVTV model, which has been proven to be an effective denoising method. The SVTV model is effective in color image denoising due to its innovative use of the hue saturation value (HSV) color space and its ability to leverage the important characteristics of the saturation and value channels. In HSV, the hue channel contains less structural information, while the saturation and value channels capture significant details and edges. By focusing on the saturation and value components, the SVTV model can better preserve edge information and reduce color artifacts. The SVTV norm is defined as follows [9]:

$$\text{SVTV}(u) := \int_{\Omega} \left[\sqrt{\|\nabla_x^s u(x, y)\|_s^2 + \|\nabla_y^s u(x, y)\|_s^2} + \mu \sqrt{\|\nabla_x^v u(x, y)\|_v^2 + \|\nabla_y^v u(x, y)\|_v^2} \right] dx dy,$$

where $\nabla_x^s u(x, y)$ and $\nabla_y^s u(x, y)$ represent the partial derivatives of the saturation channel in the x and y directions, respectively, while $\nabla_x^v u(x, y)$ and $\nabla_y^v u(x, y)$ represent the partial derivatives of the value channel in the x and y directions, respectively. This formulation ensures that both local and global features are considered during denoising, making the model robust against various types of noise. And, we have

$$\begin{aligned} \|\nabla_x^s u(x, y)\|_s &= \frac{1}{3} \|\mathbf{C} \nabla_x u(x, y)^T\|_2, \\ \|\nabla_y^s u(x, y)\|_s &= \frac{1}{3} \|\mathbf{C} \nabla_y u(x, y)^T\|_2, \\ \nabla_x u(x, y) &= [\nabla_x u_1(x, y), \nabla_x u_2(x, y), \nabla_x u_3(x, y)], \\ \nabla_y u(x, y) &= [\nabla_y u_1(x, y), \nabla_y u_2(x, y), \nabla_y u_3(x, y)], \\ \mathbf{C} &= \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}, \\ \|\nabla_x^v u(x, y)\|_v &= \frac{1}{\sqrt{3}} |\nabla_x u_1(x, y) + \nabla_x u_2(x, y) + \nabla_x u_3(x, y)|, \\ \|\nabla_y^v u(x, y)\|_v &= \frac{1}{\sqrt{3}} |\nabla_y u_1(x, y) + \nabla_y u_2(x, y) + \nabla_y u_3(x, y)|. \end{aligned}$$

The discrete form of the SVTV denosing model has been proven as

$$\begin{aligned} \min_q \sum_{k=1}^M \sum_{j=1}^N & \left[\sqrt{(\nabla_x q_1)_{kj}^2 + (\nabla_x q_2)_{kj}^2 + (\nabla_y q_1)_{kj}^2 + (\nabla_y q_2)_{kj}^2} \right. \\ & \left. + \mu \sqrt{(\nabla_x q_3)_{kj}^2 + (\nabla_y q_3)_{kj}^2} \right] + \frac{\lambda}{2} \|q - z\|_2^2, \end{aligned} \quad (1)$$

where

$$q = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} = \mathbf{P} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}, \mathbf{P} = \begin{bmatrix} \frac{1}{\sqrt{2}} \mathbf{I} & \frac{-1}{\sqrt{2}} \mathbf{I} & \mathbf{0} \\ \frac{1}{\sqrt{6}} \mathbf{I} & \frac{1}{\sqrt{6}} \mathbf{I} & -\frac{2}{\sqrt{6}} \mathbf{I} \\ \frac{1}{\sqrt{3}} \mathbf{I} & \frac{1}{\sqrt{3}} \mathbf{I} & \frac{1}{\sqrt{3}} \mathbf{I} \end{bmatrix}.$$

3. Two-step SVTV-Stokes denoising model. In this section, we introduce a combination of the TV-Stokes model and the SVTV model for color image denoising, leveraging the strengths of both approaches to achieve superior results in preserving image details and reducing noise. In fact, this process aligns with painting. We typically start by drawing the texture and edges. This first step outlines the essential structure and details of the image. Once the sketch is complete, we proceed to fill in the colors, ensuring that the shades and tones enhance the initial figure without blurring or distorting the edges.

3.1. Smoothing tangential field. The TV-Stokes model is similar to sketching the texture and edges of a painting. In this model, we smooth the tangential field of each color channel independently, ensuring that critical structural elements, such as edges and textures, are preserved. By focusing on the tangential field, we reduce noise while maintaining the integrity of the edges, just as a precise sketch defines the contours and key details of a painting.

When extending the TV-Stokes model to color images, each color channel is treated separately. This approach ensures that the specific characteristics of each channel are preserved, which is essential for maintaining the integrity of the color information in the image.

By smoothing the tangential field, the TV-Stokes model effectively reduces noise while preserving edges and important structural details. Through this extension, each color channel's edges are treated independently to ensure that the edges remain sharp and well-defined. By solving

$$\arg \min_{\vec{\tau}=(\vec{\tau}_1, \vec{\tau}_2, \vec{\tau}_3) \in \Psi} J(\vec{\tau}) = \int_{\Omega} \sum_{i=1}^3 \sum_{j=1}^2 |\nabla \tau_{ij}| dx dy + \frac{\mu_1}{2} \int_{\Omega} \sum_{i=1}^3 \sum_{j=1}^2 (\tau_{ij} - \tau_{ij}^z)^2 dx dy, \quad (2)$$

one can get the ideal tangential field $\vec{\tau}^0$ of the ideal image, where τ^z is the tangential field of the noised image z . In fact, solving Eq. (2) equals finding $\vec{\tau}^0 \in \Psi$ that satisfies

$$J(\vec{\tau}^0) = \inf_{\vec{\tau} \in \Psi} J(\vec{\tau}).$$

According to Theorem 3.2 in Ref. [10], this problem has a unique solution when $\vec{\tau}^z \in (L^2(\Omega; R^2))^3$. First, by applying the Lagrange multiplier method to Eq. (2), we have

$$\begin{aligned} \arg \min_{\vec{\tau}=(\vec{\tau}_1, \vec{\tau}_2, \vec{\tau}_3) \in \Psi} & \int_{\Omega} \sum_{i=1}^3 \sum_{j=1}^2 |\nabla \tau_{ij}| dx dy \\ & + \frac{\mu_1}{2} \int_{\Omega} \sum_{i=1}^3 \sum_{j=1}^2 (\tau_{ij} - \tau_{ij}^z)^2 dx dy + \int_{\Omega} \sum_{i=1}^3 \lambda_i \nabla \cdot \vec{\tau}_i dx dy, \end{aligned}$$

where $\nabla \cdot \vec{\tau}_i = 0, i = 1, 2, 3$.

Solving the Euler-Lagrange equations for each color channel according to the variational principle, we have

$$-\nabla \cdot \left(\frac{\nabla \tau_{i1}}{|\nabla \tau_{i1}|} \right) + \mu_1 (\tau_{i1} - \tau_{i1}^z) + \nabla_x \lambda_i = 0 \quad \text{in } \Omega, \quad (3)$$

$$-\nabla \cdot \left(\frac{\nabla \tau_{i2}}{|\nabla \tau_{i2}|} \right) + \mu_1 (\tau_{i2} - \tau_{i2}^z) + \nabla_y \lambda_i = 0 \quad \text{in } \Omega, \quad (4)$$

$$\nabla \cdot \tau_i = 0 \quad \text{in } \Omega. \quad (5)$$

And, Eqs. (3), (4), and (5) satisfy the following boundary conditions:

$$\frac{\nabla \tau_{i1}}{|\nabla \tau_{i1}|} \cdot v_i + \lambda_i v_{i1} = 0, \quad \text{on } \partial\Omega, \quad (6)$$

$$\frac{\nabla \tau_{i2}}{|\nabla \tau_{i2}|} \cdot v_i - \lambda_i v_{i2} = 0, \quad \text{on } \partial\Omega, \quad (7)$$

where $\vec{v} = (\vec{v}_1, \vec{v}_2, \vec{v}_3) = \begin{pmatrix} v_{11} & v_{21} & v_{31} \\ v_{12} & v_{22} & v_{32} \end{pmatrix}$ is the outward normal vector at the boundary $\partial\Omega$ for u .

Through iterative methods, we can solve Eqs. (3), (4), (5), (6), and (7) together. When λ^n is given, τ_{i1}^{n+1} and τ_{i2}^{n+1} can be iteratively solved using the following formulas:

$$-\nabla \cdot \left(\frac{\nabla \tau_{i1}^{n+1}}{|\nabla \tau_{i1}^{n+1}|} \right) + \mu_1 \left[\tau_{i1}^{n+1} - \left(\tau_{i1}^z - \frac{1}{\mu_1} \nabla_x \lambda^n \right) \right] = 0 \text{ in } \Omega, \quad (8)$$

$$-\nabla \cdot \left(\frac{\nabla \tau_{i2}^{n+1}}{|\nabla \tau_{i2}^{n+1}|} \right) + \mu_1 \left[\tau_{i2}^{n+1} - \left(\tau_{i2}^z - \frac{1}{\mu_1} \nabla_y \lambda^n \right) \right] = 0 \text{ in } \Omega, \quad (9)$$

$$\frac{\nabla \tau_{i1}^{n+1}}{|\nabla \tau_{i1}^{n+1}|} \cdot v_i + \lambda^n v_{i1} = 0, \quad \frac{\nabla \tau_{i2}^{n+1}}{|\nabla \tau_{i2}^{n+1}|} \cdot v_i - \lambda^n v_{i2} = 0 \text{ on } \partial\Omega, \quad (10)$$

$$\text{s.t } \nabla \cdot \tau_i^{n+1} = 0 \text{ in } \Omega. \quad (11)$$

Eqs. (8), (9), and (10) can be solved using the algorithm proposed by Chambolle [4]:

$$p_{ij}^{k+1} = \frac{p_{ij}^k + \kappa (\nabla (\nabla \cdot p_{ij}^k - \mu_1 (\tau_{ij}^z - \frac{1}{\mu_1} \nabla_j \lambda_i^n)))}{1 + \kappa \left| \nabla (\nabla \cdot p_{ij}^k - \mu_1 (\tau_{ij}^z - \frac{1}{\mu_1} \nabla_j \lambda_i^n)) \right|}, \quad (12)$$

$$\tau_{ij}^{n+1} = \tau_{ij}^z - \frac{1}{\mu_1} \nabla_j \lambda^n - \frac{1}{\mu_1} \lim_{k \rightarrow \infty} \nabla \cdot p_{ij}^k, \quad (13)$$

where the index j takes values of 1 and 2, $\nabla_j = \begin{cases} \nabla_x, j=1, \\ \nabla_y, j=2, \end{cases}$. κ is a constant, chosen to ensure the convergence and faster convergence speed of the algorithm.

To calculate λ^{n+1} at each step, we first compute the differentials of Eqs. (12) and (13) with respect to x and y separately. Then, we add them together, resulting in

$$\begin{aligned} & \frac{\partial}{\partial x} \left[-\nabla \cdot \left(\frac{\nabla \tau_{i1}^{n+1}}{|\nabla \tau_{i1}^{n+1}|} \right) \right] + \frac{\partial}{\partial y} \left[-\nabla \cdot \left(\frac{\nabla \tau_{i2}^{n+1}}{|\nabla \tau_{i2}^{n+1}|} \right) \right] + \frac{\partial^2 \lambda_i^n}{\partial x^2} + \frac{\partial^2 \lambda_i^n}{\partial y^2} \\ & + \mu_1 \left(\frac{\partial \tau_{i1}^{n+1}}{\partial x} + \frac{\partial \tau_{i2}^{n+1}}{\partial y} \right) - \mu_1 \left(\frac{\partial \tau_{i1}^z}{\partial x} + \frac{\partial \tau_{i2}^z}{\partial y} \right) = 0. \end{aligned}$$

Since $\nabla \cdot \vec{\tau}_i^{n+1} = 0$ and $\nabla \cdot \vec{\tau}_i^z = 0$, λ^{n+1} can be computed using the following expression:

$$\Delta \lambda_i^n = \frac{\partial}{\partial x} \left[\nabla \cdot \left(\frac{\nabla \tau_{i1}^{n+1}}{|\nabla \tau_{i1}^{n+1}|} \right) \right] + \frac{\partial}{\partial y} \left[\nabla \cdot \left(\frac{\nabla \tau_{i2}^{n+1}}{|\nabla \tau_{i2}^{n+1}|} \right) \right], \quad (14)$$

$$\frac{\nabla \tau_{i1}^{n+1}}{|\nabla \tau_{i1}^{n+1}|} \cdot v_i + \lambda_i^{n+1} v_{i1} = 0, \text{ in } \Omega, \quad (15)$$

$$\frac{\nabla \tau_{i2}^{n+1}}{|\nabla \tau_{i2}^{n+1}|} \cdot v_i - \lambda_i^{n+1} v_{i2} = 0, \text{ in } \Omega. \quad (16)$$

Eq. (14) is a Poisson equation, which can be solved using the PyAMG library¹ in Python or by employing the discrete cosine transform (DCT) method. In fact, when solving Poisson equations, finite differences are typically used for approximation. When using the PyAMG library to solve the aforementioned Poisson equation, we only need to reshape the $M \times N$ -dimensional color channel matrix into a column vector format. Then, we solve it using sparse matrix operations for each row. In [6], using the DCT method to solve the Poisson equation for square matrix images ($M \times M$ matrices) was introduced. In this paper, we further extend this method to solve $M \times N$ matrix images. First, we need to define the difference matrix,

$$B_{M|N} = \frac{1}{h} \begin{pmatrix} -1 & 1 & & & \\ & -1 & 1 & & \\ & & \ddots & \ddots & \\ & & & -1 & 1 \end{pmatrix}_{M|N},$$

$$-B_{M|N}^T = \frac{1}{h} \begin{pmatrix} 1 & & & & \\ -1 & 1 & & & \\ & \ddots & \ddots & & \\ & & & -1 & 1 \\ & & & & -1 \end{pmatrix}_{M|N}.$$

Therefore, according to the definition, one can conclude that

$$\Delta^h \lambda_i^n = -\lambda_i^n B_N^T B_N - B_M^T B_M \lambda_i^n. \quad (17)$$

Performing singular value decomposition (SVD) on the matrix $B_{M|N}$, we have

$$B_{M|N} = S_{M|N} [0, \Sigma_{M|N}] C_{M|N}, \quad \Sigma_{M|N} = \text{diag}(\sigma_1, \dots, \sigma_{M|N-1}), \quad (18)$$

where $\sigma_k = \frac{2}{h} \sin \frac{\pi k}{2M|N}$, $k = 1, 2, \dots, M|N - 1$. Thus, Eq. (18) can be written as

$$f_i^n = \Delta^h \lambda_i^n = -\lambda_i^n C_N^T \begin{bmatrix} 0 & \\ & \Sigma_N^2 \end{bmatrix} C_N - C_M^T \begin{bmatrix} 0 & \\ & \Sigma_M^2 \end{bmatrix} C_M \lambda_i^n. \quad (19)$$

Denote $\hat{f}_i^n = C_M f_i^n C_N^T$, $\hat{\lambda}_i^n = C_M \lambda_i^n C_N^T$. We left multiply Eq. (19) by matrix C_M , and then right multiply matrix C_N^T . Eq. (19) can be written as

$$\hat{f}_i^n = -\hat{\lambda}_i^n \begin{bmatrix} 0 & \\ & \Sigma_N^2 \end{bmatrix} - \begin{bmatrix} 0 & \\ & \Sigma_M^2 \end{bmatrix} \hat{\lambda}_i^n. \quad (20)$$

Thus, by using Eq. (20), we calculate $\hat{\lambda}_i^n$ as

$$\begin{aligned} \hat{\lambda}_{i,0,0}^n &= 0, \\ \hat{\lambda}_{i,k,0}^n &= -\hat{f}_{i,k,0}^n / \sigma_{k,y}^2, \quad k = 1, 2, \dots, M-1, \\ \hat{\lambda}_{i,0,j}^n &= -\hat{f}_{i,0,j}^n / \sigma_{j,x}^2, \quad j = 1, 2, \dots, N-1, \\ \hat{\lambda}_{i,k,j}^n &= -\hat{f}_{i,k,j}^n / (\sigma_{k,y}^2 + \sigma_{j,x}^2), \quad k = 1, 2, \dots, M-1, j = 1, 2, \dots, N-1. \end{aligned} \quad (21)$$

According to $\hat{\lambda}_i^n = C_M \lambda_i^n C_N^T$, we deduce that

$$\lambda_i^n = C_M^T \hat{\lambda}_i^n C_N. \quad (22)$$

When computing λ_i^n at each step, we substitute it into Eqs. (12) and (13) to calculate τ_{ij}^{n+1} for each iteration. When the stopping condition $\|\tau_{ij}^{n+1} - \tau_{ij}^n\| / \|\tau_{ij}^{n+1}\| \leq \varepsilon_1$ is met at a certain iteration, we denote $\lim_{n \rightarrow \infty} \tau_{ij}^{n+1} = \tau_{ij}^0$. Normalize τ_{ij}^0 and let

¹PyAMG library can be downloaded and used by <https://github.com/pyamg/pyamg>.

$\hat{\tau}_{ij}^0 = \tau_{ij}^0 / \|\tau_{ij}^0\|_2$. Thus, based on the correspondence between normal vectors and tangential vectors, we obtain

$$n_{i,1}^0 = \hat{\tau}_{i2}^0, n_{i,2}^0 = -\hat{\tau}_{i1}^0,$$

where $i = 1, 2, 3$.

Denote

$$\nabla_x g_i = n_{i,1}, \nabla_y g_i = n_{i,2}. \quad (23)$$

Then, we have

$$-\Delta g_i = \nabla_x^T n_{i,1}^0 + \nabla_y^T n_{i,2}^0. \quad (24)$$

The solution for g in Eq. (24) can be obtained by using the method described above for solving Eq. (17).

Algorithm 1 Iterative Computation of τ_{i1}^{n+1} and τ_{i2}^{n+1}

Input: $\mu_1 > 0, h > 0$

Initialization: $\lambda_i^1 = 0$, where $i = 1, 2, 3$.

while $\frac{\|\tau_{i1}^{n+1} - \tau_{i1}^n\|}{\|\tau_{i1}^{n+1}\|} \leq \varepsilon_1$ and $\frac{\|\tau_{i2}^{n+1} - \tau_{i2}^n\|}{\|\tau_{i2}^{n+1}\|} \leq \varepsilon_1$ **do**

Solve Eqs. (8), (9) and (10) for τ_{i1}^{n+1} and τ_{i2}^{n+1} by Eq. (12);

Solve Eqs. (14), (15) and (16) for λ_i^{n+1} by Eqs. (18), (19), (20), (21) and (22).

end while

3.2. Image reconstruction. We first utilize the TV-Stokes model to perform preliminary denoising and edge extraction on the image with lower computational cost. Building on this, we leverage the color TV regularization feature of the SVTV model to further enhance the preliminarily processed image by applying color smoothing and detail preservation. This combined approach effectively eliminates noise while maintaining the natural appearance of the image.

In this part, we focus on adding colors to enhance the initial figure, which is why we employ the SVTV model. This model operates in the HSV color space, concentrating on the saturation and value channels.

The SVTV model specifically focuses on noise reduction in the saturation (S) and value (V) channels, where most structural details and intensity information are preserved. This approach can be likened to adding color to a painting—enhancing the underlying sketch without overwhelming it. The hue (H) channel is excluded because it does not provide sufficient edge or detail information for effective image restoration. As illustrated in Figure 2 of [9], the H channel image lacks significant features, such as edges, which are critical for restoration tasks. In contrast, the S and V channels retain most of the important color and edge information, making them more suitable for TV regularization. By focusing on the S and V channels, the SVTV model achieves improved restoration performance by effectively leveraging the meaningful image details contained in these channels.

We denote the solution obtained in the first step as the tangential vector $\vec{\tau}^0$, which allows us to derive the normal vector \vec{n}^0 based on the corresponding relationship

that $\vec{n}_i^0 = (n_{i1}^0, n_{i2}^0) = (\tau_{i2}^0, -\tau_{i1}^0)$. Here, we denote $W(h)$ as

$$W(h) = \int_{\Omega} \sqrt{\sum_{i=1}^2 [(\nabla_x h_i)^2 + (\nabla_y h_i)^2]} + \mu \sqrt{(\nabla_x h_3)^2 + (\nabla_y h_3)^2} dx dy \\ + \frac{\mu_2}{2} \int_{\Omega} \sum_{i=1}^3 (h_i - z_i + g_i)^2 dx dy,$$

where $h_i = u_i - g_i$, and $g = (g_1, g_2, g_3)$ satisfies $\nabla g_i = (n_{ix}^0, n_{iy}^0)$. The existence of g can be ensured by $\nabla \times \vec{n}^0 = 0$. Therefore, the second step of the SVTV-Stokes model is equivalent to finding $h^0 \in (BV(\Omega))^3$ that satisfies

$$W(h^0) = \inf_{h \in (BV(\Omega))^3} W(h), \quad (25)$$

by which we can obtain the ideal image u .

Theorem 3.1. *Eq. (25) guarantees a unique solution.*

Proof. In fact, to ensure the existence and uniqueness of the solution to Eq. (25), we only need to ensure Eq. (25) is strictly convex. The function $W(h)$ consists of two main parts:

Part 1. The anisotropic TV term,

$$\int_{\Omega} \sqrt{\sum_{i=1}^2 [(\nabla_x h_i)^2 + (\nabla_y h_i)^2]} + \mu \sqrt{(\nabla_x h_3)^2 + (\nabla_y h_3)^2} dx dy,$$

is strictly convex due to the nature of the Euclidean norm and the fact that the square root function is strictly convex and increasing.

Part 2. The quadratic fidelity term,

$$\frac{\mu_2}{2} \int_{\Omega} \sum_{i=1}^3 (h_i - z_i + g_i)^2 dx dy,$$

is strictly convex because it is a sum of squared differences.

Given that both components of $W(h)$ are strictly convex, their sum is also strictly convex. Strict convexity implies that, for any two distinct points $h, h' \in (BV(\Omega))^3$ and $\lambda \in (0, 1)$,

$$W(\lambda h + (1 - \lambda)h') < \lambda W(h) + (1 - \lambda)W(h').$$

This strict inequality ensures that there is a unique minimizer for the functional $W(h)$. The space $(BV(\Omega))^3$ is a Banach space, and it is well-known that any lower semicontinuous, strictly convex functional defined on a reflexive Banach space has a unique minimizer. The functional $W(h)$ is lower semicontinuous due to the properties of the integral and the norms involved.

Additionally, the coercivity of $W(h)$ can be argued based on the quadratic fidelity term, which grows without bound as $\|h\| \rightarrow \infty$, ensuring that the functional does not have minimizers at infinity, and thus must attain its minimum at some finite point in $(BV(\Omega))^3$.

The strict convexity of $W(h)$ directly implies the uniqueness of the minimizer. In a strictly convex functional, the convex combination of two distinct minimizers would lead to a lower value of the functional, contradicting the assumption that both points are minimizers.

Given the strict convexity and lower semicontinuity of $W(h)$, we conclude that the minimization problem

$$W(h^0) = \inf_{h \in (BV(\Omega))^3} W(h)$$

has a unique solution. The functional form ensures both the existence and the uniqueness of the solution due to the combination of the anisotropic TV term and the quadratic fidelity term, both contributing to the overall strict convexity of the functional. \square

By utilizing (1), we can derive the discrete form of (25) as

$$\begin{aligned} \min_q \sum_{k=1}^M \sum_{j=1}^N & \left[\sqrt{(\nabla_x q_1 - \tilde{n}_{1,x})_{kj}^2 + (\nabla_x q_2 - \tilde{n}_{2,x})_{kj}^2 + (\nabla_y q_1 - \tilde{n}_{1,y})_{kj}^2 + (\nabla_y q_2 - \tilde{n}_{2,y})_{kj}^2} \right. \\ & \left. + \mu \sqrt{(\nabla_x q_3 - \tilde{n}_{3,x})_{kj}^2 + (\nabla_y q_3 - \tilde{n}_{3,y})_{kj}^2} \right] + \frac{\lambda}{2} \|q - z\|_2^2, \end{aligned}$$

where

$$\tilde{n} = \begin{bmatrix} \tilde{n}_{1,x} & \tilde{n}_{1,y} \\ \tilde{n}_{2,x} & \tilde{n}_{2,y} \\ \tilde{n}_{3,x} & \tilde{n}_{3,y} \end{bmatrix} = \mathbf{P} \begin{bmatrix} n_{1,x}^0 & n_{1,y}^0 \\ n_{2,x}^0 & n_{2,y}^0 \\ n_{3,x}^0 & n_{3,y}^0 \end{bmatrix}.$$

According to Eq. (23), the above equation can be rewritten as

$$\begin{aligned} \min_q \sum_{k=1}^M \sum_{j=1}^N & \left[\sqrt{(\nabla_x (q_1 - \tilde{g}_1))_{kj}^2 + (\nabla_x (q_2 - \tilde{g}_2))_{kj}^2 + (\nabla_y (q_1 - \tilde{g}_1))_{kj}^2 + (\nabla_y (q_2 - \tilde{g}_2))_{kj}^2} \right. \\ & \left. + \mu \sqrt{(\nabla_x (q_3 - \tilde{g}_3))_{kj}^2 + (\nabla_y (q_3 - \tilde{g}_3))_{kj}^2} \right] + \frac{\lambda}{2} \|q - z\|_2^2, \end{aligned} \quad (26)$$

where

$$\tilde{g} = \mathbf{P}g. \quad (27)$$

If we denote $q - \tilde{g}$ as h , Eq. (26) can be rewritten as

$$\begin{aligned} \min_h \sum_{k=1}^M \sum_{j=1}^N & \left[\sqrt{(\nabla_x h_1)_{kj}^2 + (\nabla_x h_2)_{kj}^2 + (\nabla_y h_1)_{kj}^2 + (\nabla_y h_2)_{kj}^2} \right. \\ & \left. + \mu \sqrt{(\nabla_x h_3)_{kj}^2 + (\nabla_y h_3)_{kj}^2} \right] + \frac{\lambda}{2} \|q - z\|_2^2. \end{aligned}$$

Letting $w_1^x = \nabla_x h_1$, $w_1^y = \nabla_y h_1$, $w_2^x = \nabla_x h_2$, $w_2^y = \nabla_y h_2$, $w_3^x = \nabla_x h_3$, and $w_3^y = \nabla_y h_3$, we denote $w = \begin{pmatrix} w_1^x & w_1^y \\ w_2^x & w_2^y \\ w_3^x & w_3^y \end{pmatrix}$. Therefore, by using the Laplacian

multiplier method, the above equation can be written as

$$\begin{aligned}
\min_{w,h} \sum_{k=1}^M \sum_{j=1}^N & \left[\sqrt{|(w_1^x)_{kj}|^2 + |(w_2^x)_{kj}|^2 + |(w_1^y)_{kj}|^2 + |(w_2^y)_{kj}|^2} \right. \\
& + \mu \sqrt{|(w_3^x)_{kj}|^2 + |(w_3^y)_{kj}|^2} \Big] + \frac{\lambda}{2} \|\tilde{g} - z + h\|^2 + (\xi_1^x, w_1^x - \nabla_x h_1) + (\xi_2^x, w_2^x - \nabla_x h_2) \\
& + (\xi_3^x, w_3^x - \nabla_x h_3) + (\xi_1^y, w_1^y - \nabla_y h_1) + (\xi_2^y, w_2^y - \nabla_y h_2) + (\xi_3^y, w_3^y - \nabla_y h_3) \\
& + \frac{\beta}{2} \left[\|w_1^x - \nabla_x h_1\|^2 + \|w_2^x - \nabla_x h_2\|^2 + \|w_3^x - \nabla_x h_3\|^2 \right. \\
& \left. + \|w_1^y - \nabla_y h_1\|^2 + \|w_2^y - \nabla_y h_2\|^2 + \|w_3^y - \nabla_y h_3\|^2 \right].
\end{aligned} \tag{28}$$

We need to solve the minimization problem (28), which is generally very challenging to find the exact minimum values for w and h in practical computations. This difficulty arises because w and h are coupled together. Typically, we can separate and solve for the variables w and h using Eqs. (29) and (30) and then use an alternating minimization process to solve Eq. (28). There are

$$\begin{aligned}
h^{n+1} = \operatorname{argmin}_h & \frac{\lambda}{2} \|\tilde{g} - z + h\|_2^2 - (\xi_1^{n,x}, \nabla_x h_1) - (\xi_2^{n,x}, \nabla_x h_2) - (\xi_3^{n,x}, \nabla_x h_3) \\
& - (\xi_1^{n,y}, \nabla_y h_1) - (\xi_2^{n,y}, \nabla_y h_2) - (\xi_3^{n,y}, \nabla_y h_3) \\
& + \frac{\beta}{2} \left[\|w_1^{n,x} - \nabla_x h_1\|^2 + \|w_2^{n,x} - \nabla_x h_2\|^2 + \|w_3^{n,x} - \nabla_x h_3\|^2 \right. \\
& \left. + \|w_1^{n,y} - \nabla_y h_1\|^2 + \|w_2^{n,y} - \nabla_y h_2\|^2 + \|w_3^{n,y} - \nabla_y h_3\|^2 \right],
\end{aligned} \tag{29}$$

$$\begin{aligned}
w^{n+1} = \operatorname{argmin}_w & \sum_{k=1}^M \sum_{j=1}^N \left[\sqrt{|(w_1^x)_{kj}|^2 + |(w_2^x)_{kj}|^2 + |(w_1^y)_{kj}|^2 + |(w_2^y)_{kj}|^2} \right. \\
& + \mu \sqrt{|(w_3^x)_{kj}|^2 + |(w_3^y)_{kj}|^2} \Big] + (\xi_1^{n,x}, w_1^x) + (\xi_2^{n,x}, w_2^x) \\
& + (\xi_3^{n,x}, w_3^x) + (\xi_1^{n,y}, w_1^y) + (\xi_2^{n,y}, w_2^y) + (\xi_3^{n,y}, w_3^y) \\
& + \frac{\beta}{2} \left[\|w_1^{n,x} - \nabla_x h_1^{n+1}\|^2 + \|w_2^{n,x} - \nabla_x h_2^{n+1}\|^2 + \|w_3^{n,x} - \nabla_x h_3^{n+1}\|^2 \right. \\
& \left. + \|w_1^{n,y} - \nabla_y h_1^{n+1}\|^2 + \|w_2^{n,y} - \nabla_y h_2^{n+1}\|^2 + \|w_3^{n,y} - \nabla_y h_3^{n+1}\|^2 \right], \\
& \xi^{n+1,x|y} = \xi^{n,x|y} + \beta \left(w^{n,x|y} - \nabla_{x|y} h^n \right).
\end{aligned} \tag{30}$$

Here, we select $\xi^0 = 0$, $h^0 = 0$, and $w^0 = 0$.

In this way, it is possible to approximate the minimum value of Eq. (28) in practice. The convergence of this computational approach was proven by Wu and Tai [15].

Eq. (29) can be solved using the fast fourier transform (FFT). In fact, for convenience, Eq. (29) can be written in the following form:

$$h^{n+1} = \arg \min_h \frac{\lambda}{2} \|\tilde{g} - z + h\|_2^2 - (\xi^n, \nabla h) + \frac{\beta}{2} \|w^n - \nabla h\|_2^2. \quad (31)$$

By solving the Euler-Lagrange equation of the right-hand side, we obtain

$$\lambda(g - z + h) + \nabla \cdot \xi^n + \beta \nabla \cdot w^n - \beta \Delta h = 0.$$

Thus, we have

$$(\lambda - \beta \Delta) h + \lambda(g - z) + \nabla \cdot \xi + \beta \nabla \cdot w^n = 0. \quad (32)$$

Performing FFT processing on Eq. (32), we have

$$h^{n+1} = F^{-1} \left(\frac{\lambda(z - g) - F(\nabla_x) F(\xi^{n,x} + \beta w^{n,x}) - F(\nabla_y) F(\xi^{n,y} + \beta w^{n,y})}{\lambda - \beta F(\Delta)} \right). \quad (33)$$

For operators ∇_x, ∇_y , and Δ , their Fourier transforms are considered as the transforms of their corresponding convolution kernels.

The solution to Eq. (30) can be divided into two sub-problems:

$$\begin{aligned} w_1^{n+1}, w_2^{n+1} = \arg \min_{w_1, w_2} \sum_{k=1}^M \sum_{j=1}^N & \left(\sqrt{|(w_1^x)_{kj}|^2 + |(w_2^x)_{kj}|^2 + |(w_1^y)_{kj}|^2 + |(w_2^y)_{kj}|^2} \right) \\ & + (\xi_1^{n,x}, w_1^x) + (\xi_2^{n,x}, w_2^x) + (\xi_1^{n,y}, w_1^y) + (\xi_2^{n,y}, w_2^y) \\ & + \frac{\beta}{2} \left[\|w_1^x - \nabla_x h_1^{n+1}\|^2 + \|w_2^x - \nabla_x h_2^{n+1}\|^2 \right. \\ & \left. + \|w_1^y - \nabla_y h_1^{n+1}\|^2 + \|w_2^y - \nabla_y h_2^{n+1}\|^2 \right], \end{aligned} \quad (34)$$

$$\begin{aligned} w_3^{n+1} = \arg \min_{w_3} \mu \sum_{k=1}^M \sum_{j=1}^N & \left(\sqrt{|(w_3^x)_{kj}|^2 + |(w_3^y)_{kj}|^2} \right) + (\xi_3^{n,x}, w_3^x) \\ & + (\xi_3^{n,y}, w_3^y) + \frac{\beta}{2} \left(\|w_3^x - \nabla_x h_3^{n+1}\|^2 + \|w_3^y - \nabla_y h_3^{n+1}\|^2 \right). \end{aligned} \quad (35)$$

For Eq. (34), by [15], one can derive that

$$w_1^{n+1} = \max(0, 1 - \frac{\alpha}{\beta |s_1^{n+1}|}) s_1^{n+1}, \quad (36)$$

$$w_2^{n+1} = \max(0, 1 - \frac{\alpha}{\beta |s_2^{n+1}|}) s_2^{n+1}, \quad (37)$$

where

$$s_1^{n+1} = (s_{1,1}^{n+1}, s_{1,2}^{n+1}), \quad (38)$$

$$s_{1,1}^{n+1} = \nabla h_1^{n+1} - \xi_1^n / \beta, \quad (39)$$

$$s_{1,2}^{n+1} = \nabla h_2^{n+1} - \xi_2^n / \beta, \quad (40)$$

$$|s_1^{n+1}| = \sqrt{(s_{1,1}^{n+1})^2 + (s_{1,2}^{n+1})^2}. \quad (41)$$

Similarly, we can solve Eq. (35), resulting in

$$w_3^{n+1} = \max(0, 1 - \frac{\alpha \mu}{\beta |s_2^{n+1}|}) s_2^{n+1}, \quad (42)$$

where

$$s_2^{n+1} = \nabla h_3^{n+1} - \xi_3^n / \beta. \quad (43)$$

When the stopping criterion $\|h^{n+1} - h^n\| / \|h^{n+1}\| \leq \varepsilon_2$ is met at a certain iteration, utilizing $u = \mathbf{P}^{-1} \left(\lim_{n \rightarrow \infty} h^n + \tilde{g} \right)$, we can yield the denoised ideal image u .

Algorithm 2 Iterative Computation of h^{n+1}

Input: $\alpha > 0, \lambda > 0, \beta > 0, \mu > 0$

Initialization: $\xi^0 = 0, h^0 = 0, w^0 = 0$

while $\frac{\|h^{n+1} - h^n\|}{\|h^{n+1}\|} \leq \varepsilon_2$ **do**

 Solve Eq. (31) by Eq. (33) for h^{n+1} ;

 Solve Eq. (34) by Eqs. (36), (37), (38), (39), (40), and (41) for w_1^{n+1} and w_2^{n+1} ;

 Solve Eq. (35) by Eqs. (42) and (43) for w_3^{n+1} ;

end while

3.3. Bayesian optimization. Notice that there are many parameters to smooth the tangential vectors in the first step and reconstruct the image in the second step (totaling $h, \mu_1, \alpha, \beta, \lambda, \mu$), and different images have different requirements for parameter selection during denoising. In practical operations, after repeated trials, we found that setting λ to 1 is relatively appropriate. However, to better optimize this denoising model, it is feasible to use optimization algorithms to change the values of other parameters as well. For simplicity, we will only consider optimizing the five parameters h, μ_1, α, β , and μ .

Generally, we adjust the parameter α based on the standard deviation of Gaussian noise σ , i.e. $\alpha = \zeta \times \sigma \times \sqrt{M \times N \times Chs}$, where M, N , and Chs represent the number of rows (height), columns (width), and channels of the image, respectively. Considering that M, N , and Chs are only related to the image itself, we can optimize the parameter α by only considering ζ .

Considering the above, we only need to consider optimizing h, μ_1, ζ, β , and μ . Here, we denote $\mathbf{x} = (h, \mu_1, \zeta, \beta, \mu)$. Next, we will consider introducing Bayesian optimization from machine learning to optimize the above parameters. In fact, calling Bayesian optimization is simple, as it only requires using the scikit-optimize library² in Python. Next, we briefly introduce the principle of Bayesian optimization.

Bayesian optimization is an iterative method used to optimize objective functions. Its core idea is to construct a posterior probability model of the objective function in the parameter space and combine intelligent sampling strategies to find the optimal solution [7, 1]. This is also known as the Gaussian process, which estimates the mean and variance of the objective function using known initial observation data. As iterations progress, the model is updated based on new observation data, providing a more accurate estimate of the objective function. This method is particularly suitable for expensive objective function evaluations because it can find better solutions in relatively few iterations. The process of Bayesian optimization mainly consists of the following four steps.

²The scikit-optimize library can be downloaded and used by <https://scikit-learn.org/stable/>.

Step 1. Define the objective function. The objective function describes the performance metric of the denoising model. Here, our objective function is peak signal-to-noise ratio (PSNR) [16], defined as

$$\text{PSNR}(u(\mathbf{x}), z_0) = 10 \log_{10} \frac{255^2 \times M \times N}{\sum_{(x,y) \in \Omega} |u(\mathbf{x}) - z_0|^2},$$

where z_0 is the initial figure without adding Gaussian noise. In practical operations, we often consider minimization problems. Therefore, to maximize PSNR, we take its negative value as the objective function. Here, we denote the objective function as $f(\mathbf{x}) = -\text{PSNR}(u(\mathbf{x}), z_0)$.

Step 2. Parameter space definition. Next, we need to define the search space for parameters. Here, the parameter space includes parameters h, μ_1, ζ, β , and μ , each with its range of values. We set the ranges as $h \in [1, 13]$, $\mu_1 \in [0.0001, 0.1]$, $\zeta \in [0.0001, 0.1]$, $\mu \in [0.01, 1]$, and $\beta \in [0.01, 1]$. These ranges are based on the actual scenario of the problem and prior knowledge.

Step 3. Invoke Bayesian optimization method. We use the Bayesian optimization algorithm to search for the optimal parameter combination. This algorithm uses a Gaussian process as a surrogate model for the objective function. It iteratively estimates the mean and variance of the objective function using existing observation data and selects the next parameter combination to evaluate. The Gaussian process confidence interval strategy plays a balancing role between exploration and exploitation.

The Gaussian process regression model for the objective function can be expressed as

$$f(\mathbf{x}) \sim \mathcal{GP}(\mu(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')),$$

where $\mu(\mathbf{x})$ denotes the mean function, and $k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{|\mathbf{x} - \mathbf{x}'|^2}{2l^2}\right)$ represents the Gaussian kernel function, with l as the length scale parameter that controls the smoothness of the resulting Gaussian process. The Gaussian kernel is preferred because it provides a smooth and flexible model that can capture the variations in the objective function values effectively.

The acquisition function guides the selection of the next point to evaluate. And, in this paper, we select the expected improvement (EI) acquisition function because the EI function not only considers the predicted mean, but also the uncertainty, making it suitable for finding the optimal parameter set in complex function spaces, which is defined as

$$\alpha_{\text{EI}}(\mathbf{x}) = \mathbb{E} [\max(f(\mathbf{x}) - f(\mathbf{x}^+), 0)],$$

where $f(\mathbf{x}^+)$ is the current best observation.

The next point to observe can be obtained by

$$\mathbf{x}_{\text{next}} = \arg \max_{\mathbf{x}} \alpha(\mathbf{x}|\mathcal{D}),$$

where \mathcal{D} represents the current set of observed data.

In our implementation, we use the “gp_minimize” function in Python to perform Bayesian optimization. This function iteratively constructs the posterior probability model of the objective function and selects the next most promising parameter combination for evaluation until reaching the specified number of iterations or convergence condition.

Step 4. Output of results. After the optimization process, we can extract the optimized optimal parameter combination and the corresponding optimal PSNR value. These results can help us further analyze the performance of the algorithm and the impact of parameters on the results. Then, we obtain

$$\mathbf{x}^* = \arg \min_{\mathbf{x}} f(\mathbf{x}).$$

4. Numerical experiments. In this section, we compare CTV [3], DTV [13], SVTV [9], and SVTV-Stokes denoising models by adding Gaussian noise to images and evaluating their performance using PSNR, SSIM [17], and NIQE (natural image quality evaluator) [12] metrics. Here, we do not include the TV-Stokes model in the comparison because it is primarily designed for grayscale image denoising and processes each color channel independently. This approach does not consider the correlations between color channels that are essential for preserving both structural and chromatic consistency in color images. Consequently, the TV-Stokes model is less effective for color image denoising compared to vector-valued models such as the introduced models. The proposed SVTV-Stokes model demonstrated superior denoising capabilities numerically and visually, indicating higher image quality. Furthermore, by introducing new solutions to CTV and DTV model, it showed improved efficiency and better detail preservation compared to the traditional the CTV and DTV model. These findings highlight the effectiveness of the new models in practical denoising scenarios.

4.1. Quantitative comparison. We added Gaussian noise with standard deviations of 0.05, 0.1, and 0.2 to six original images in Figure 1 and compared them with the original images. We select $\varepsilon_1 = \varepsilon_2 = 10^{-4}$ to ensure the convergence of every model during the optimization process. This threshold was set during our initial simulation experiment using the Figure 1a (House). This choice of tolerance values is crucial for achieving consistent and reliable results across different scenarios. As illustrated in Table 2, we recorded the PSNR and relative error to demonstrate that the model’s performance metrics reach the desired levels of accuracy and stability.

4.1.1. Potential drawbacks of PSNR & SSIM metrics. Metrics like PSNR and SSIM are commonly used to determine the effectiveness of image reconstruction tasks in different domains. However, in the field of super-resolution reconstruction tasks, the introduction of generative adversarial networks (GANs) in recent years has revealed a more nuanced understanding. It has been observed that high PSNR or SSIM values do not always correspond to better reconstruction quality. This phenomenon stems from the fact that images with high PSNR or SSIM values may not accurately capture the texture details that are perceptible to human vision.

Furthermore, considering that in practical scenarios we often only have access to noisy images and it is challenging to obtain the original clean images, the aforementioned PSNR and SSIM evaluation metrics are not applicable. Therefore, to better assess the denoising performance of each model, a new evaluation metric needs to be considered. As a result, researchers have been striving to find more effective metrics for measuring image reconstruction quality.

4.1.2. NIQE metric: its dual nature and practical considerations. The NIQE metric demonstrates a dual nature in its application. On one hand, it is a powerful and objective tool for no-reference image quality assessment, providing valuable insights into the quality of processed images without requiring ground truth references.

This makes it particularly suitable for our study, where we aim to showcase the performance of our method. On the other hand, like any metric, its effectiveness may vary depending on the specific characteristics of the images being evaluated. While it is based on natural image statistics, it may not fully capture certain subjective aspects of human visual perception. Nevertheless, NIQE remains a robust and widely recognized metric that effectively reflects practical scenarios in real-world applications.

To better compare color images, we made certain modifications to the NIQE value from [12]. We took the arithmetic mean of the three color channels, namely

$$NIQE(u) = \frac{\sum_{ch=1}^3 NIQE(u_{ch})}{3},$$

where u_{ch} is the ch -th color channel of image u , and $ch = 1, 2, 3$. The value of $NIQE(u_{ch})$ can be obtained following the process provided in [12].

In this paper, we also use the NIQE metric to compare the performance of denoising models. The NIQE score reflects the naturalness of an image and its closeness to real-world characteristics. A lower NIQE score indicates that the image is closer to a natural, high-quality image, while a higher NIQE score suggests the presence of more noise, distortions, or unnatural features, leading to lower perceived quality.

It is worth noting that in practical applications, we observed certain inconsistencies in NIQE-based evaluations. For instance, the NIQE value of an original image may sometimes be higher than that of a noisy image, and in some cases, adding Gaussian noise with a higher standard deviation can result in a lower NIQE score after denoising. This behavior arises because original images may contain complex textures, fine details, or intricate patterns, which can increase the NIQE value. Adding Gaussian noise may simplify the image by blurring these details, thereby reducing the NIQE score.

Moreover, the NIQE algorithm may not fully capture all the features of an image. It is more sensitive to certain attributes, such as smoothness and texture, while being less responsive to others. Factors like image content, noise type, and structural complexity can all influence NIQE evaluations.

In practical denoising scenarios, it is often infeasible to use PSNR and SSIM metrics, as the original images are typically unavailable. However, models that perform poorly in terms of PSNR and SSIM metrics tend to have higher (worse) NIQE values, which highlights the consistency of these metrics in evaluating denoising quality.

To ensure meaningful comparisons, we believe that NIQE values are most informative when applied to images denoised by relatively mature models—models that have demonstrated reasonable denoising performance either through visual comparisons or through other metrics, such as PSNR and SSIM, when reference images are available. Especially in the current era of semi-supervised learning, where ground truth data may be limited, combining visual assessments, traditional metrics, and no-reference metrics like NIQE provides a well-rounded evaluation approach. This methodology allows us to effectively assess the strengths and limitations of denoising models in both controlled and practical scenarios.

4.1.3. Comparison of denoising performance numerically and computational efficiency. In this case, we calculated the PSNR and SSIM values for each reconstructed image using different denoising models and recorded the runtime. Based



FIGURE 1. The original images intended to use in this paper.

on the Bayesian optimization, we optimized the parameters of the new denoising model proposed in this paper and obtained the optimal PSNR and SSIM values for the denoising of the SVTV-Stokes model, as shown in Table 1. According to the results in Table 1, we can compare the performance of different denoising methods in terms of PSNR, SSIM, and CPU time.

Note that, in this work, we utilized the relative error threshold condition as the stopping criterion for all models. This criterion is widely used in optimization problems due to its simplicity and effectiveness in ensuring computational efficiency.

The CTV and DTV models are enhanced by utilizing the algorithms presented in Section 3.2 to solve the minimization problem. To differentiate, we denote them as CTV^2 and DTV^2 , respectively. The main difference between $DTV^2(u)$ and $SVTV(u)$ compared in this paper lies in the generation of the orthogonal matrices for the transformations used in the optimization procedure. The benefits of these improvements will be discussed in Section 4.3.

Figure 2 compares the performance of the four denoising models (CTV^2 , DTV^2 , SVTV, and SVTV-Stokes) in terms of PSNR (A), SSIM (B), and relative errors (C) over iterations.

In Figure 2a, the PSNR values highlight the advantages of the SVTV-Stokes model, which not only achieves the highest peak values early in the iterations, but also remains relatively stable as the iterations progress. The DTV^2 and SVTV models also show significant improvement in PSNR during the initial iterations, eventually stabilizing at slightly lower levels than SVTV-Stokes. In contrast, the CTV^2 model exhibits a downward trend in PSNR over iterations, reflecting its limited ability to enhance image quality.

Figure 2b further supports this observation through the SSIM values, which measure structural similarity. The SVTV-Stokes model achieves consistently higher SSIM values, indicating its effectiveness in preserving structural details. Both DTV^2 and SVTV maintain relatively strong structural similarity, though slightly below SVTV-Stokes. Conversely, the CTV^2 model reaches the lowest SSIM values, highlighting its poorer performance in retaining structural information when dealing with strong noise.

TABLE 1. Numerical results for denoising using different models.

Figure	Std	Metric	Noisy	Model			
				CTV^2	DTV^2	SVTV	SVTV-Stokes
1 256 × 256	0.05	PSNR	26.0318	30.1228	<u>32.5668</u>	31.9694	32.7968
		SSIM	0.7109	0.9108	0.9209	<u>0.9296</u>	0.9343
		Time(s)	—	8.54	18.71	23.99	<u>9.23</u>
	0.1	PSNR	20.1027	27.1935	<u>28.7861</u>	28.3648	29.8283
		SSIM	0.4266	0.8390	0.8461	<u>0.8496</u>	0.8949
		Time(s)	—	<u>10.90</u>	19.64	23.19	8.26

Continued on next page

Figure	Std	Metric	Noisy	Model			
				CTV ²	DTV ²	SVTV	SVTV-Stokes
2 256 × 256	0.2	PSNR	14.5705	23.2684	<u>25.2004</u>	24.9061	26.3971
		SSIM	0.2093	0.7076	<u>0.7767</u>	0.7434	0.8414
		Time(s)	—	43.97	<u>20.21</u>	23.02	9.84
	0.05	PSNR	26.2127	25.6603	30.8947	<u>31.4728</u>	31.6897
		SSIM	0.8322	0.7929	0.9274	<u>0.9389</u>	0.9459
		Time(s)	—	<u>14.28</u>	18.55	15.51	9.14
	0.1	PSNR	20.3521	23.8179	26.9913	<u>27.3373</u>	27.6021
		SSIM	0.5818	0.7112	0.8312	<u>0.8551</u>	0.8653
		Time(s)	—	<u>20.20</u>	30.24	21.48	14.91
	0.2	PSNR	15.0276	21.8177	<u>23.5843</u>	23.2084	23.7629
		SSIM	0.3052	0.6118	<u>0.7184</u>	0.7009	0.7312
		Time(s)	—	<u>25.25</u>	179.94	31.94	12.67
3 256 × 256	0.05	PSNR	26.3954	27.0658	30.9542	<u>30.9651</u>	31.4035
		SSIM	0.8101	0.8633	0.9356	<u>0.9442</u>	0.9496
		Time(s)	—	8.03	19.55	22.76	<u>8.31</u>
	0.1	PSNR	20.5079	24.7660	26.7011	<u>26.9503</u>	27.5194
		SSIM	0.5567	0.7827	0.8463	<u>0.8746</u>	0.8791
		Time(s)	—	<u>11.58</u>	23.56	30.87	7.95
	0.2	PSNR	14.9121	22.1419	<u>23.2897</u>	23.2810	23.5717
		SSIM	0.2877	0.6731	<u>0.7558</u>	<u>0.7531</u>	0.7631
		Time(s)	—	<u>14.09</u>	127.63	38.54	7.11
	0.05	PSNR	26.3953	26.1511	31.0096	<u>31.0419</u>	31.5385
		SSIM	0.8174	0.9069	0.9311	<u>0.9410</u>	0.9496
		Time(s)	—	<u>10.22</u>	19.24	19.52	7.05
4 256 × 256	0.1	PSNR	20.7232	23.5336	26.0638	<u>27.1254</u>	27.4524
		SSIM	0.6003	0.8333	0.8952	<u>0.9100</u>	0.9137
		Time(s)	—	<u>14.51</u>	26.71	28.53	7.88
	0.2	PSNR	15.2204	21.2311	22.9162	<u>22.9844</u>	23.3504
		SSIM	0.3516	0.7214	0.7938	<u>0.7982</u>	0.8123
		Time(s)	—	<u>16.07</u>	64.22	36.84	8.73
	0.05	PSNR	26.0292	32.3236	<u>37.2384</u>	37.1404	37.4622
		SSIM	0.6484	0.9197	0.9732	<u>0.9757</u>	0.9793
		Time(s)	—	7.85	100.97	181.32	<u>79.58</u>
	0.1	PSNR	20.2456	28.6257	<u>32.7229</u>	32484	33.6027
		SSIM	0.3535	0.8182	<u>0.9352</u>	0.9278	0.9588
		Time(s)	—	10.01	116.53	179.47	<u>53.12</u>
5 513 × 512	0.2	PSNR	15.1890	24.8940	<u>27.8408</u>	26.3744	28.8149
		SSIM	0.1653	0.6895	<u>0.8572</u>	0.7989	0.9295
		Time(s)	—	14.40	821.13	181.68	<u>56.31</u>
	0.05	PSNR	26.0292	29.4154	<u>33.7182</u>	31.4976	33.9660
		SSIM	0.6484	0.8395	<u>0.9397</u>	0.9166	0.9437
		Time(s)	—	311.17	5431.17	<u>2502.97</u>	4921.12
	0.1	PSNR	20.2456	26.3911	29.3676	<u>29.4874</u>	30.1244
		SSIM	0.3535	0.7316	0.8520	<u>0.8726</u>	0.8812
		Time(s)	—	678.81	5582.01	<u>2398.03</u>	4427.15
	0.2	PSNR	15.1890	22.8440	25.0899	<u>25.1266</u>	25.3703
		SSIM	0.1653	0.5817	0.7308	<u>0.7498</u>	0.7580
		Time(s)	—	1082.87	5837.74	<u>2356.12</u>	3988.88

The trend of relative errors in Figure 2c aligns with the observations from Figures 2a and 2b. All models experience rapid error reduction in the early iterations, with SVTV-Stokes converging to the lowest relative error, followed by SVTV and DTV². The CTV² model consistently exhibits the highest relative error throughout the iterations. It is important to note that to avoid distortion caused by the extremely high errors in the initial iterations, especially for the CTV² model, the 11th iteration is chosen as the starting point for visualization and is labeled as the 1st iteration in Figure 2c.

From Table 1, we can find that the improvement of our algorithm under strong noise conditions (e.g., Gaussian noise with a standard deviation of 0.2) is limited, which can be explained by the inherent challenge of balancing noise removal and

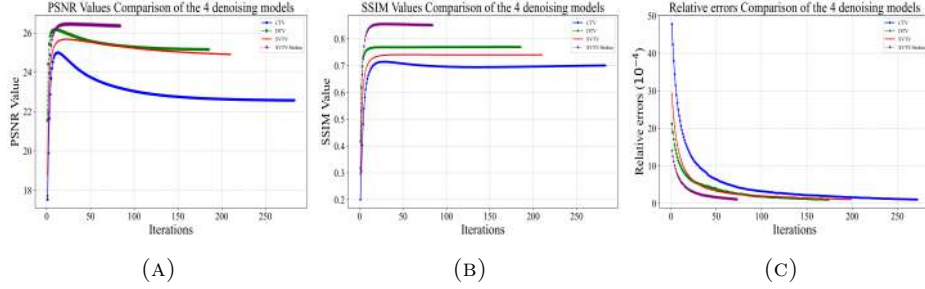


FIGURE 2. PSNR (A), SSIM (B) and ε_r (C) of CTV², DTV², SVTV, SVTV-Stokes denoising models on Figure 1a (House).

detail preservation. Our two-step approach is designed to reduce noise while preserving edges and structural details. However, in the presence of strong noise, the tangential vector field smoothing step faces a trade-off: excessive smoothing may blur critical details, while insufficient smoothing can leave residual noise, making it challenging to achieve optimal performance.

Moreover, the processing time in Figure 1f is slower than that of the original SVTV model, as shown in Table 1. This could be attributed to the slow convergence of the tangential vector field smoothing step. Additionally, in our current implementation, the three channels are processed individually. In the future, we plan to improve computational efficiency by leveraging advanced techniques such as parallel computing. Moreover, we may explore the possibility of coupling the tangential vector fields of the three channels to further optimize performance.

One can notice that the relative error threshold condition leads to premature convergence in certain cases, such as with the CTV² model. Specifically, the CTV² model reaches the stopping threshold more quickly, resulting in fewer iterations and a shorter runtime. However, this also limits the model’s ability to further refine the image, which is reflected in the lower PSNR and SSIM values compared to the proposed SVTV model especially for Figure 1e and 1f. It is worth noting that alternative stopping criteria, such as $\|u^{n+1} - z\|_2 / \|z\|_2$, could be explored in future work to better balance computational efficiency and denoising performance. This norm-based condition might allow the model to achieve greater accuracy at the cost of slightly increased runtime, potentially mitigating the limitations caused by the relative error threshold.

We focus on the house image (Figure 1a) with Gaussian noise added with a standard deviation of 0.2, denoted as Figure 3. The PSNR value of the noise image is 14.5705, and the SSIM value is 0.2093, which serve as the baseline for denoising. When compared to the DTV² and SVTV models, both of these models achieve PSNR values of 25.2004 and 24.9061, and SSIM values of 0.7767 and 0.7434, respectively. This clearly demonstrates that denoising methods that consider color channel coupling, such as DTV² and SVTV, can more effectively handle denoising tasks for color images.

The SVTV-Stokes model proposed in this paper extracts the smooth part of the image from the smoothed tangent field and uses the anisotropic TV-Stokes model to obtain the “texture” part of the denoised image. This allows for better capture of image features and noise distribution. After optimizing the parameters using Bayesian optimization, the performance of the algorithm is improved, and

the denoising effect for this image is optimal. Its PSNR value reaches 26.3971, and SSIM value reaches 0.8414, with a shorter runtime compared to the CTV², DTV², and SVTV models.

Thus, we chose to demonstrate with a standard deviation of 0.2 in the paper based on several considerations. Firstly, under the setting of a standard deviation of 0.2, the quality evaluation metrics (PSNR and SSIM) of the image are relatively high, reaching 26.3971 and 0.8414, respectively. This indicates good image reconstruction quality, preserving a significant amount of detail and structural information. Secondly, compared to other scenarios, the execution time at a standard deviation of 0.2 is relatively short, only 9.84 seconds. This shows that the algorithm has a high processing speed, suitable for real-time or near real-time applications. Furthermore, selecting a standard deviation of 0.2 for demonstration purposes is also due to its balanced performance in terms of PSNR, SSIM, and execution time, making it highly representative.

Considering that CTV² model performed poorly in color image denoising as shown in Table 1, which may also result in them having lower NIQE values, we only compare the original image, DTV² model, SVTV model, and the SVTV-Stokes model proposed in this paper in terms of NIQE values for handling Gaussian noise with a standard deviation of 0.05. The results are shown in Table 2:

TABLE 2. NIQE values for the 6 images in Figure 1: Original figure, DTV² model, SVTV model, and SVTV-Stokes model with 0.05 standard deviation Gaussian noise.

Figure	Original Figure	DTV ²	SVTV	SVTV-Stokes
1	21.5801	<u>23.5557</u>	24.3421	22.5578
2	21.5330	25.6952	<u>22.7802</u>	22.4496
3	21.8178	24.8210	<u>22.8489</u>	22.5021
4	21.6821	23.4951	<u>22.8235</u>	22.2453
5	21.0213	24.3662	<u>22.9397</u>	21.8193
6	19.7544	21.0252	<u>20.8011</u>	20.7763

The results from Table 2 show that the SVTV-Stokes model proposed in this paper achieves lower NIQE values compared to the DTV² model and SVTV model when the Gaussian noise standard deviation is 0.05. These observations not only deepen our understanding of the SVTV-Stokes model but also provide valuable insights for research in the field of image processing. Overall, the SVTV-Stokes model proposed in this paper has achieved satisfactory results in image denoising research, and the introduction of the NIQE metric provides beneficial insights for further exploration of image processing algorithms in practical usage scenarios.

4.2. Visual comparison. In this part, by comprehensively analyzing the the enlarged demonstration image blocks, we can gain a deeper understanding of the differences among different denoising models in terms of image preservation, denoising effectiveness, and algorithm efficiency, thereby providing a more comprehensive basis for the final model selection. Therefore, we further extract the red-boxed part from row 3 (A) in Figures 3, 4, 5, 6, 7, and 8 for detailed display, as shown in row 4 in Figures 3, 4, 5, 6, 7, and 8, which helps us gain deeper insights into the excellent performance of the SVTV-Stokes model proposed in this paper for image denoising

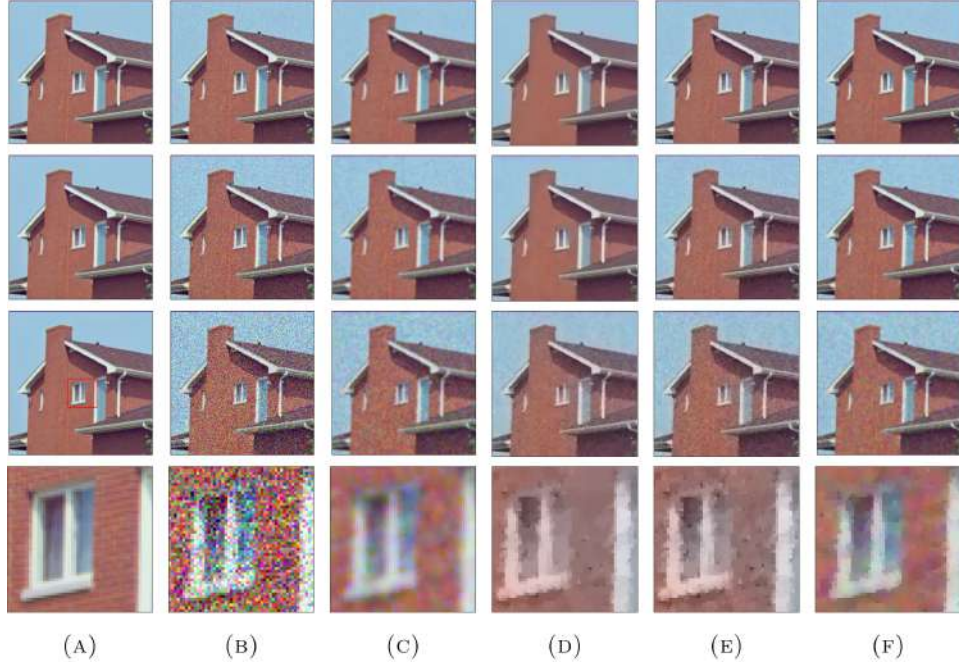


FIGURE 3. The denoising effects of different models on Figure 1a (House) after adding Gaussian noise with standard deviations of 0.05, 0.1, and 0.2 are shown in rows 1, 2, and 3, respectively. Row 4 is the enlarged demonstration section of the red area when adding 0.2 Gaussian noise with the (A) Original image, (B) Noisy image, (C) CTV² denoising model, (D) DTV² denoising model, (E) SVTV denoising model, and (F) SVTV-Stokes model proposed in this paper.

tasks. The SVTV-Stokes model exhibits good preservation of image details and structures.

We can easily observe that both the CTV² and DTV² models exhibit suboptimal performance in visual assessments. The CTV² model produces overly blurred images, while the DTV² model retains a significant amount of noise, detracting from the quality of the restored images.

In contrast, when examining the SVTV model and the newly proposed SVTV-Stokes model, the improvements become immediately apparent. As shown in the enlarged view in Figure 3, the new SVTV-Stokes model demonstrates superior color recovery and more faithful texture restoration compared to the other models. In Figure 4, the SVTV-Stokes model significantly enhances the clarity of the leaves, effectively reducing noise and improving overall detail.

For Figure 5, the SVTV-Stokes model preserves finer details of the paddle, which is notably clearer than in the SVTV model. In Figure 6, the colors restored by the SVTV-Stokes model are much closer to the original image, while the SVTV model results in a more washed-out, grayish appearance. Similarly, in Figure 7, the SVTV-Stokes model retains the fine lines on the nose with high accuracy, whereas

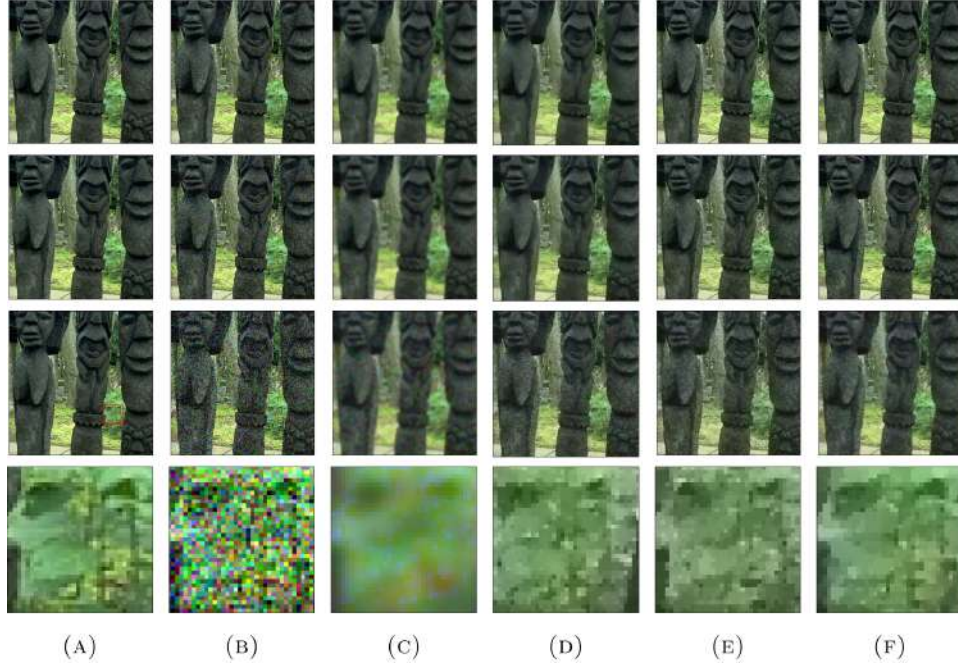


FIGURE 4. The denoising effects of different models on Figure 1b (Stone) after adding Gaussian noise with standard deviations of 0.05, 0.1, and 0.2 are shown in rows 1, 2, and 3, respectively. Row 4 is the enlarged demonstration section of the red area when adding 0.2 Gaussian noise with the (A) Original image, (B) Noisy image, (C) CTV² denoising model, (D) DTV² denoising model, (E) SVTV denoising model, and (F) SVTV-Stokes model proposed in this paper.

the SVTV model produces blurry lines that lack sharpness. In Figure 8, the SVTV-Stokes model delivers a much clearer image than the SVTV model, offering greater detail and overall definition.

Furthermore, upon closely examining the magnified image blocks, it is evident that the SVTV-Stokes model effectively reduces noise, leading to a clearer, more natural visual outcome. When it comes to preserving intricate image details, the SVTV-Stokes model outperforms the alternatives, solidifying its exceptional performance in image restoration tasks.

4.3. Algorithm analysis. In this section, we delve into a comparison between the solving methods of DTV and the enhanced DTV², focusing particularly on their efficiency and scientific rigor. As outlined in Algorithms 1 and 2 provided in [13], DTV utilizes traditional approaches to solve optimization problems, while DTV² significantly improves upon these methods.

DTV² surpasses DTV in efficiency by decomposing the complex optimization problem into simpler subproblems, which can be solved iteratively and more rapidly. This decomposition allows for more manageable computations and reduces the overall time required for the optimization process. Moreover, DTV² leverages iterative

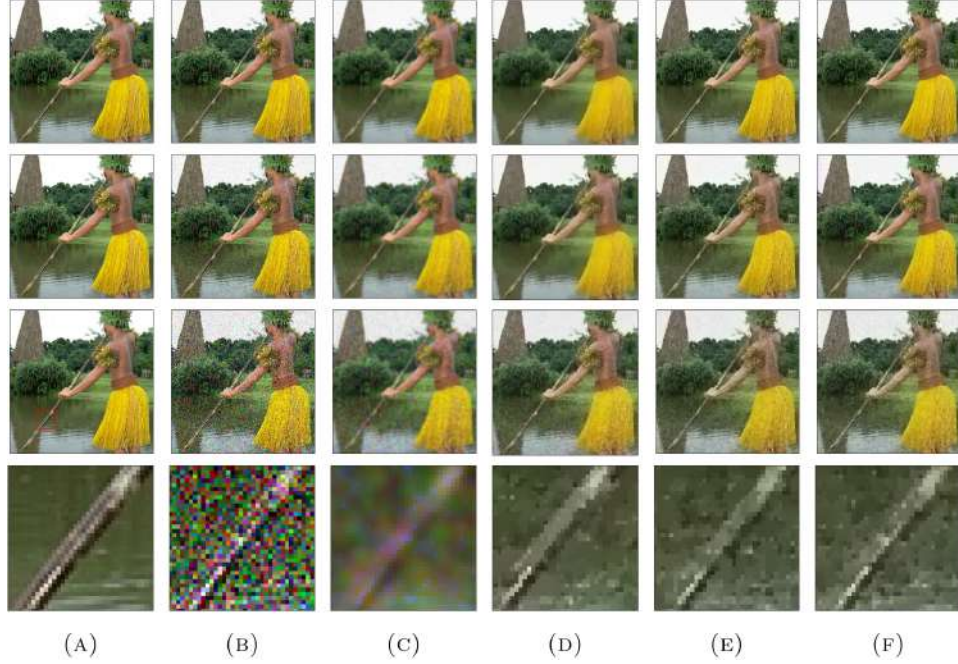


FIGURE 5. The denoising effects of different models on Figure 1c (Aborigine) after adding Gaussian noise with standard deviations of 0.05, 0.1, and 0.2 are shown in rows 1, 2, and 3, respectively. Row 4 is the enlarged demonstration section of the red area when adding 0.2 Gaussian noise with the (A) Original image, (B) Noisy image, (C) CTV^2 denoising model, (D) DTV^2 denoising model, (E) SVTV denoising model, and (F) SVTV-Stokes model proposed in this paper.

PDE solvers to address the coupled partial differential equations derived from the Euler-Lagrange equations. These iterative solvers offer precise control over the optimization process and can be finely tuned to achieve faster convergence, thus enhancing the model's overall efficiency.

The iterative nature of the PDE solvers in DTV^2 is a crucial advancement, allowing for dynamic adjustments and refinements during the optimization process. This iterative approach ensures that the model converges more quickly to an optimal solution, reducing computational overhead and improving the speed of processing.

The numerical results are presented in the following sections. To illustrate the performance of the denoising algorithms, we focus on the results for the image shown in Figure 1a, where Gaussian noise with a standard deviation of 0.2 was added. After denoising, the PSNR and CPU time for the DTV model were 24.8454 and 30.61 seconds, respectively. In contrast, the DTV^2 model achieved a PSNR of 25.2004 and required only 20.21 seconds of CPU time. The denoised images are displayed in Figure 9. Upon examining these figures, it becomes evident that the denoising result of the DTV model appears excessively smooth, which indicates an over-smoothing effect that can blur important details and edges in the image. Conversely, the DTV^2 model not only improves the PSNR, but also preserves the image details more

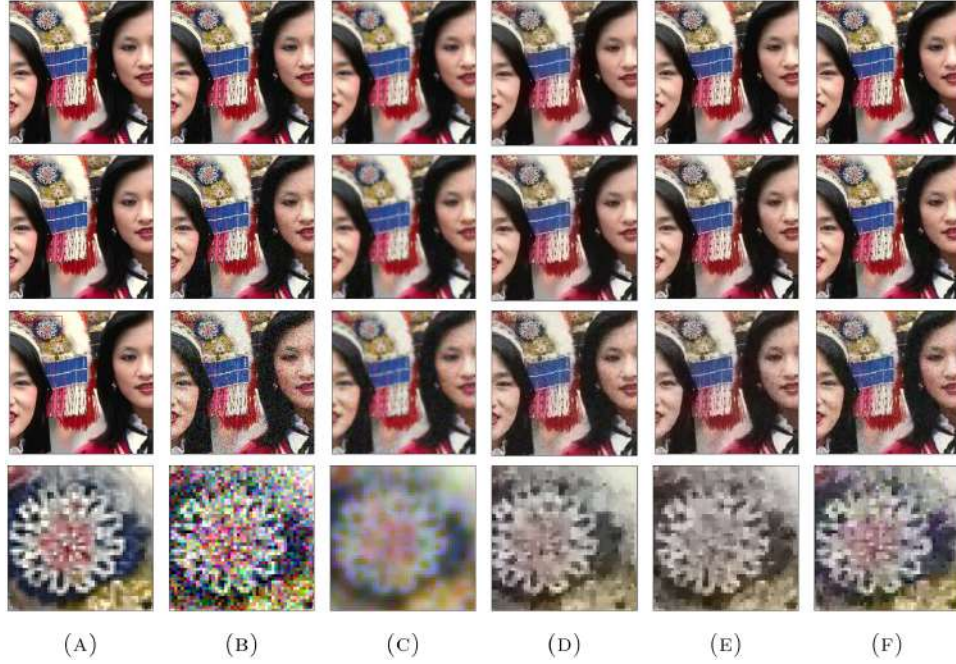


FIGURE 6. The denoising effects of different models on Figure 1d (Girls) after adding Gaussian noise with standard deviations of 0.05, 0.1, and 0.2 are shown in rows 1, 2, and 3, respectively. Row 4 is the enlarged demonstration section of the red area when adding 0.2 Gaussian noise with the (A) Original image, (B) Noisy image, (C) CTV^2 denoising model, (D) DTV^2 denoising model, (E) SVTV denoising model, and (F) SVTV-Stokes model proposed in this paper.

effectively while significantly reducing the computation time. This improvement demonstrates the efficiency and effectiveness of the DTV^2 approach in handling noise while maintaining the structural integrity of the image.

5. Conclusions. This paper proposes a new method for color image denoising, which utilizes the SVTV model based on a two-step approach, combined with Bayesian optimization from machine learning for parameter tuning. The method first employs tangential field smoothing techniques for initial processing of color images, which effectively improves image smoothness by selecting data fidelity terms and TV regularization terms. Subsequently, leveraging the characteristics of the SVTV model, the image after preliminary processing undergoes further color smoothing and detail preservation. This is done to maximize noise elimination while maintaining the natural appearance of the image. And, Bayesian optimization is applied to optimize parameters and enhance algorithm performance. We establish the solvability properties of the two-step SVTV-Stokes method and propose corresponding fast algorithms. Numerical experimental results demonstrate that our method can effectively capture hidden details in color images and exhibit high denoising performance.

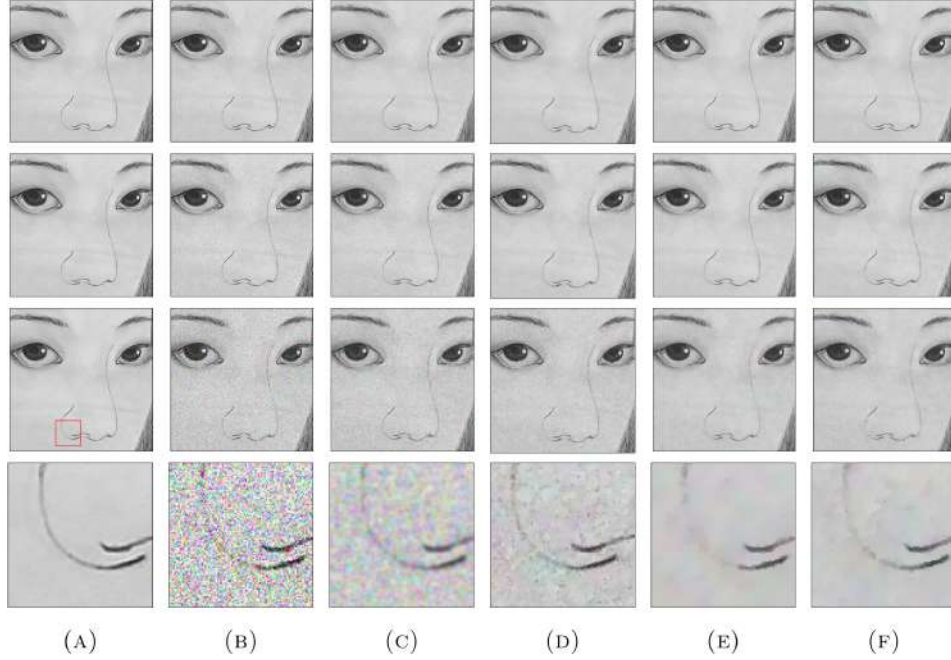


FIGURE 7. The denoising effects of different models on Figure 1e (Menggu) after adding Gaussian noise with standard deviations of 0.05, 0.1, and 0.2 are shown in rows 1, 2, and 3, respectively. Row 4 is the enlarged demonstration section of the red area when adding 0.2 Gaussian noise with the (A) Original image, (B) Noisy image, (C) CTV² denoising model, (D) DTV² denoising model, (E) SVTV denoising model, and (F) SVTV-Stokes model proposed in this paper.

Specifically, for Figure 1a, after adding Gaussian noise with a standard deviation of 0.2, the PSNR value is 14.5705 and the SSIM value is 0.2093. After denoising with the DTV and SVTV models, the PSNR and SSIM values increase to 25.2004 and 0.7767, and 24.9061 and 0.7434, respectively. This clearly demonstrates that denoising methods considering color channel coupling can more effectively handle denoising tasks for color images. It also emphasizes the importance of considering the correlation between color channels in color image denoising.

The new proposed SVTV-Stokes model in this paper achieves optimal denoising results for this image, with a PSNR value of 26.3971 and an SSIM value of 0.8414, while also having a shorter runtime compared to the DTV² and SVTV models. Additionally, we introduce NIQE for comparing denoising models, and the new model achieves an NIQE value of 22.4496. When dealing with images with Gaussian noise of different standard deviations, the new model performs best in terms of PSNR, SSIM, and NIQE, with shorter runtime, further demonstrating its advantages in image quality and algorithm efficiency.

The new model is expected to provide innovative and effective solutions for the future of color image denoising, offering valuable insights and guidance for research and applications in the field of image processing. In the future, the effectiveness



FIGURE 8. The denoising effects of different models on Figure 1f (Xiyingpan) after adding Gaussian noise with standard deviations of 0.05, 0.1, and 0.2 are shown in rows 1, 2, and 3, respectively. Row 4 is the enlarged demonstration section of the red area when adding 0.2 Gaussian noise with the (A) Original image, (B) Noisy image, (C) CTV^2 denoising model, (D) DTV^2 denoising model, (E) SVTV denoising model, and (F) SVTV-Stokes model proposed in this paper.

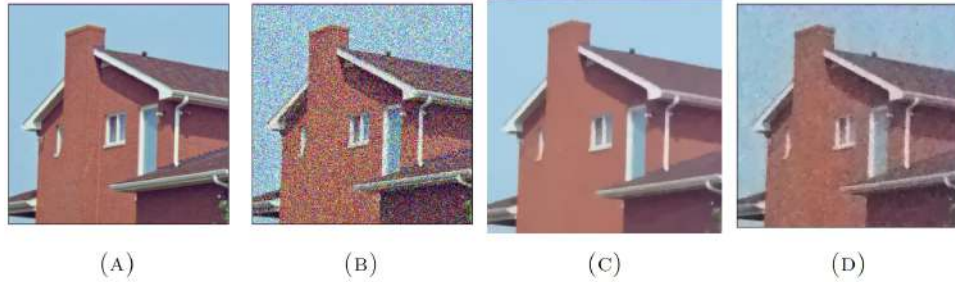


FIGURE 9. The denoising effects of DTV and DTV^2 models on Figure 1a (House) after adding Gaussian noise with standard deviations 0.2. The details are as follows: (A) Original image, (B) Noisy image, (C) DTV denoising model, (D) DTV^2 denoising model.

of the color image denoising model can be further improved by integrating deep learning. By applying convolutional neural networks (CNNs) to this model and training on a large dataset of color images, a more efficient denoising method can be developed, enhancing the model's performance and generalization capabilities.

Moreover, this model can be extended to other image processing tasks, such as dehazing and super-resolution. For dehazing, deep learning models can be used to learn the physical model of the scene and the distribution characteristics of the haze,

thus better restoring image details and color information. For super-resolution, deep learning models can learn high-frequency information of images, thus achieving the goal of converting low-resolution images into high-resolution images.

By combining machine learning, deep learning, and traditional image processing methods, the performance and effectiveness of image processing models can be further enhanced, providing more innovative and effective solutions for future research and applications in the field of image processing.

6. Acknowledgment. We sincerely thank the editor and the anonymous reviewers for their valuable comments and suggestions that have greatly improved the quality of this paper. This work was presented at *the Symposium on Theory and Applications of Mathematical Medical Imaging* (September 29 - October 5, 2024, TRMC in Kunming), and we are grateful for the constructive feedback received during the conference.

REFERENCES

- [1] R. Astudillo and P. I. Frazier, [Thinking inside the box: A tutorial on grey-box bayesian optimization](#), *2021 Winter Simulation Conference (WSC)*, (2021), 1-15, [arXiv:2201.00272](#).
- [2] P. Blomgren and T. F. Chan, [Color TV: Total variation methods for restoration of vector-valued images](#), *IEEE Transactions on Image Processing*, **7** (1998), 304-309.
- [3] X. Bresson and T. F. Chan, [Fast dual minimization of the vectorial total variation norm and applications to color image processing](#), *Inverse Problems and Imaging*, **2** (2008), 455-484.
- [4] A. Chambolle, [An algorithm for total variation minimization and applications](#), *Journal of Mathematical Imaging and Vision*, **20** (2004), 89-97.
- [5] Y. Duan, Q. Zhong, X.-C. Tai and R. Glowinski, [A fast operator-splitting method for Beltrami color image denoising](#), *Journal of Scientific Computing*, **92** (2022), Paper No. 89, 28 pp.
- [6] C. A. Elo, A. Malyshev and T. Rahman, [A dual formulation of the TV-Stokes algorithm for image denoising](#), *Scale Space and Variational Methods in Computer Vision*, 307-318.
- [7] P. Frazier, A tutorial on bayesian optimization, [arXiv:abs/1807.02811](#), <https://api.semanticscholar.org/CorpusID:49656213>.
- [8] Z.-G. Jia and M. Wei, [A new TV-Stokes model for image deblurring and denoising with fast algorithms](#), *Journal of Scientific Computing*, **72** (2017), 522-541.
- [9] Z. Jia, M. K. Ng and W. Wang, [Color image restoration by saturation-value total variation](#), *SIAM Journal on Imaging Sciences*, **12** (2019), 972-1000.
- [10] W. G. Litvinov, T. Rahman and X.-C. Tai, [A modified TV-Stokes model for image processing](#), *SIAM Journal on Scientific Computing*, **33** (2011), 1574-1597.
- [11] H. Liu, X.-C. Tai, R. Kimmel and R. Glowinski, [A color elastica model for vector-valued image regularization](#), *SIAM Journal on Imaging Sciences*, **14** (2021), 717-748.
- [12] A. Mittal, A. K. Moorthy and A. C. Bovik, [No-reference image quality assessment in the spatial domain](#), *IEEE Transactions on Image Processing*, **21** (2012), 4695-4708.
- [13] S. Ono and I. Yamada, [Decorrelated vectorial total variation](#), *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 4090-4097.
- [14] X.-C. Tai, R. Winther, X. Zhang and W. Zheng, [A uniform preconditioner for a Newton algorithm for total variation minimization and minimum-surface problems](#), *SIAM Journal on Numerical Analysis*, **61** (2023), 2062-2083.
- [15] C. Wu and X.-C. Tai, [Augmented Lagrangian method, dual methods, and split Bregman iteration for ROF, vectorial TV, and high order models](#), *SIAM Journal on Imaging Sciences*, **3** (2010), 300-339.
- [16] W. Yang, Z. Huang and W. Zhu, [A first-order Rician denoising and deblurring model](#), *Inverse Problems and Imaging*, **17** (2023), 1139-1164.
- [17] W. Zhou, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli, [Image quality assessment: From error visibility to structural similarity](#), *IEEE Transactions on Image Processing*, **13** (2004), 600-612.

Received August 2024; revised December 2024; early access February 2025.