Unwarping Screen Content Images via Structure-texture Enhancement Network and Transformation Self-estimation

Zhenzhen Xiao ,Heng Liu,Bingwen Hu

School of Computer Science and Technology, Anhui University of Technology
Ma'anshan 243032, China
Institute of Artificial Intelligence, Hefei Comprehensive National Science Center
Hefei 230088, China
hengliu@ahut.edu.cn

Abstract

While existing implicit neural network-based image unwarping methods perform well on natural images, they struggle to handle screen content images (SCIs), which often contain large geometric distortions, text, symbols, and sharp edges. To address this, we propose a structure-texture enhancement network (STEN) with transformation self-estimation for SCI warping. STEN integrates a B-spline implicit neural representation module and a transformation error estimation and self-correction algorithm. It comprises two branches: the structure estimation branch (SEB), which enhances local aggregation and global dependency modeling, and the texture estimation branch (TEB), which improves texture detail synthesis using B-spline implicit neural representation. Additionally, the transformation self-estimation module autonomously estimates the transformation error and corrects the coordinate transformation matrix, effectively handling real-world image distortions. Extensive experiments on public SCI datasets demonstrate that our approach significantly outperforms state-of-the-art methods. Comparisons on well-known natural image datasets also show the potential of our approach for natural image distortion.

Keywords: Image warping, B-spline neural representation, Structure-texture, Transformation Self-estimation.

1. Introduction

Image warping [33, 19] is a vita image processing technique that alters the spatial positions of pixels to achieve specific visual effects or changes in image morphology. This process involves adjusting the coordinates of each pixel in the image and mapping them to the new positions, thereby changing the shape or perspective of the image. In practice, image warping is often combined with other image processing techniques, such as image correction [9, 8],repositioning [30, 38], and lens distortion correction [3, 29, 23], to achieve comprehensive visual enhancement. The technique relies on mathematical models and algorithms to compute the necessary pixel mappings, making it a versatile and crucial tool in visual effects processing and real-world image correction.

Figure 1: Real-world screen content image correction employs the proposed STEN method. The image is captured by a camera, and both SRWarp[33] and LTEW[19], along with our proposed STEN, utilize the predicted transformation matrix to achieve super-resolution (SR).

Image warping aims to deform images defined on a rectangular grid into continuous shapes. By using the precise transformation or mapping function, one can effectively reconstruct the spatial structure of an image with arbitrary shape and scale to achieve the desired image transformation effect. SRwarp [33] first redefines the image warping problem as a spatial variation of generalized super-resolution and effectively addresses the reconstruction of high-frequency detail images by utilizing a deep single-image super-resolution (SISR) [43, 12, 25]architecture as the backbone.

In recent years, inspired by implicit neural functions

[32, 20, 27], implicit neural representations have shown significant potential and effectiveness in handling image distortions. LIIF [7] parameterizes the implicit neural representation using multi-layer perceptrons (MLP), mapping the input coordinates and corresponding latent variables to RGB values, thereby achieving super-resolution reconstruction at arbitrary scales. However, due to lacking spatial variation considerations, LIIF [7] has limitations in addressing image warping. Although the above methods can achieve rather good effects in dealing with natural image distortions, they often lose their effectiveness when facing Screen Content Images(SCIs) with huge deformations, including geometric shapes, text, symbols, and sharp edges. Given this, LTEW [19] utilizes the Fourier feature representations and the Jacobian matrices of coordinate transformation to continuously transform the input SCI into various shaped images. However, the signal representation obtained from LTEW [19] only includes the finite sinusoidal components, which can lead to the reconstructed values appear the Gibbs phenomenon[15, 16]. Ensuring the excellent reconstruction of the specific image structures and textures at any scale and shape is crucial for SCIs unwarping.

In this work, we propose a structure-texture enhancement model with transformation self-estimation for SCI warping. Our model employs an implicit neural representation of SCI's B-spline texture coefficient estimator [27] to mitigate the risk of undershoots and overshoots during reconstruction. Our model consists of two main branches: the structure estimation branch and the texture estimation branch. The structure estimation branch employs the Transformer technology[24, 45, 4], introducing global attention and local attention modules that operate in parallel to enhance structure features, thus facilitating effective global modelling. The texture estimation branch utilizes the B-spline features estimated from the input images and the Jacobian matrix of coordinate transformations. In geometry, the determinant of the Jacobian matrix [34, 6]represents local magnification ratios, thereby enhancing the network's ability to learn texture information. Inspired by LTEW [19], the Jacobian matrix of spatial changes is multiplied by the B-spline features of each pixel to compute pixel shapes described by directional curvature, further enhancing the model's ability to learn texture information. Since structural features better represent the overall shape and structure of SCI images, while texture features capture details and texture characteristics, we propose a structure-texture fusion module to enhance the representation capability of image features, making them more comprehensive and diverse.

In addition, we design a transformation self-estimation algorithm based on training a transformation error estimation network to obtain the correct transformation matrix in real-world uncalibrated scenarios. As shown in Fig. 1, based on the warped input image, our method can automatically and correctly estimate the transformation matrix. Even with the same transformation matrix, our approach can generate clearer, unwarped results than existing methods.

In summary, our contributions can be outlined as follows:

- We propose a dual-branch structure-texture enhancement model for SCIs warping. Our approach enhances image texture details through feature-based structure and texture enhancements.
- We also introduce a transformation self-estimation algorithm based on training a transformation error estimation network that can predict the correct coordinate transformation matrix for the corresponding warped image, thereby enhancing the robustness of our model in real-world scenarios.
- Comprehensive comparative experiments and ablation studies demonstrate the superior performance and effectiveness of our unwarping approach across three SRC datasets and five natural image datasets, achieving state-of-the-art results for various scaling factors.

2. Related Works

2.1. Image Warping

Image warping refers to the process of changing or reshaping the shape or visual appearance of an image by remapping or transforming pixels within it. This technique is widely used in computer vision[39], computer graphics[1], and image processing[37] for various purposes, including image correction, enhancement, and special effects. One common approach is grid transformation [46, 26], which involves defining a grid within the image and locally deforming pixels within this grid to achieve more precise shape adjustments and deformation effects. Through warping, the shape and size of objects in an image can be modified to enhance its visual quality and appearance. SRWarp [33] interprets the image warping task as a spatial transformation problem within the super-resolution framework, proposing a method for handling arbitrary image transformations. However, SRWarp's generalization capability is limited when dealing with unseen transformations, such as homography transformations with significant magnification factors. LTEW [19] builds on the advantages of Fourier features and the spatial variation Jacobian matrix of coordinate transformations, introducing a continuous neural representation for image distortion. Despite its advancements, LTEW's use of Fourier representation, with its finite sinusoidal components, can lead to signal undershoots or overshoots at discontinuities, a phenomenon known as the Gibbs phenomenon [15, 16].

When reconstructing signals with discontinuities [13] (such as step changes), the Gibbs phenomenon is prone to occur. Obviously, for SCIs, due to the presence of text, shapes, and symbols with large jumps or sharp edges, it is vital to overcome the Gibbs phenomenon and achieve high-definition reconstruction when performing unwarping transformations on them.

2.2. Implicit neural representation (INR)

Implicit Neural Representations (INR) [7, 20, 32, 27] utilize neural networks to provide implicit and continuous representations of signals, demonstrating significant advantages in handling spatial resolution and arbitrary coordinate transformations. The core idea of INR is to learn an implicit representation of a signal through neural networks, which is defined in a continuous space, thus overcoming the limitations of traditional grid-based representation methods. Specifically, implicit neural representations can finely describe signals, effectively handling and reconstructing them even under high resolutions and complex transformations. The LTEW [19] method combines local texture estimators with feature maps from deep neural network encoders and relative coordinates (or local grids), using local implicit neural representations (INR) to enhance the spatial resolution of input signals. This method performs exceptionally well during training and generalizes effectively to unseen tasks, showcasing robustness and versatility in complex environments and transformations. This paper proposes a new approach that combines B-spline basis functions [27, 28] with relative coordinates or local grids to more effectively handle high discontinuities in signals, such as Screen Content Images (SCI). By incorporating B-spline basis functions, this method improves accuracy at discontinuities, thereby further extending the potential and effectiveness of implicit neural representations in practical applications.

2.3. B-spline representation

B-spline is a mathematical tool widely used in computer graphics[14, 22], image processing[21, 10], and signal processing[31, 35, 36]. Its representation consists of control points, basis functions, and a knot vector. Control points define the shape of the curve, while basis functions are used to combine these control points, and the knot vector determines the support intervals of the basis functions. Unlike traditional methods that rely on signal resolution, B-spline representation achieves efficient memory management through linear scaling of model parameters. In recent years, it has gained significant attention in the field of signal processing[35], particularly for handling highly discontinuous images, such as SCI, where B-spline methods excel in accurately reconstructing local features and image shapes.

In current research, implicit B-spline representation is widely utilized for distortion removal and reconstruction. Researchers have increasingly combined trainable B-spline basis functions with Non-Uniform Rational B-Splines (NURBS) layers[28, 27], achieving notable success in geometric modeling and complex signal reconstruction. These methods can accurately estimate coefficient information and extract structural and distortion information from local regions and advanced features of input images. Building on these research achievements, we integrate deep neural network backbones with B-spline basis functions to represent deformed images under arbitrary coordinate transformations, further enhancing processing effectiveness.

3. Method

In this section, our objective is to reconstruct a high-resolution warped image $I^{WARP} \in R^{3\times H\times W}$ from a low-resolution RGB image $I_{LR} \in R^{3\times h\times w}$ using a differentiable and invertible coordinate transformation $f: X \rightarrow Y$. Here, X represents the input coordinate space, which is a set of points on a 2D plane, and Y denotes the output coordinate space after transformation. To prepare the input image I_{LR}, we first apply the inverse coordinate transformation:

$$I_{LR} = I_{HR}[f^{-1}(Y)] \tag{1}$$

In practice, warping modifies the image resolution while preserving pixel density, transforming it from ($h \times w \rightarrow H \times W$).

In traditional mathematical methods, addressing image warping issues often relies on point-matching techniques, which are widely used in multi-view stereo reconstruction. By precisely matching corresponding points from different views, the geometric transformations of images can be derived. However, this point-matching method exhibits significant limitations when handling large-scale or complex deformation scenarios. In such cases, extracting and matching feature points becomes challenging, leading to poor performance in large-range warping tasks. In recent years, end-to-end deep learning models with implicit Fourier neural representation [33, 19, 41] have been introduced to tackle unknown image transformations. However, these methods still face challenges when dealing with complex and sharp geometric transformations. This is because Fourier representation typically relies on a finite set of sine waves for local signal analysis and reconstruction. While it performs well in processing smooth frequency variations, it is prone to Gibbs phenomena when dealing with SCIs that exhibit discontinuous texture features, causing oscillations at the edges of images.

In contrast, B-spline representation utilizes piecewise polynomials for local interpolation, which has significant advantages in capturing local texture details and high-frequency components of images. In addition to fine texture

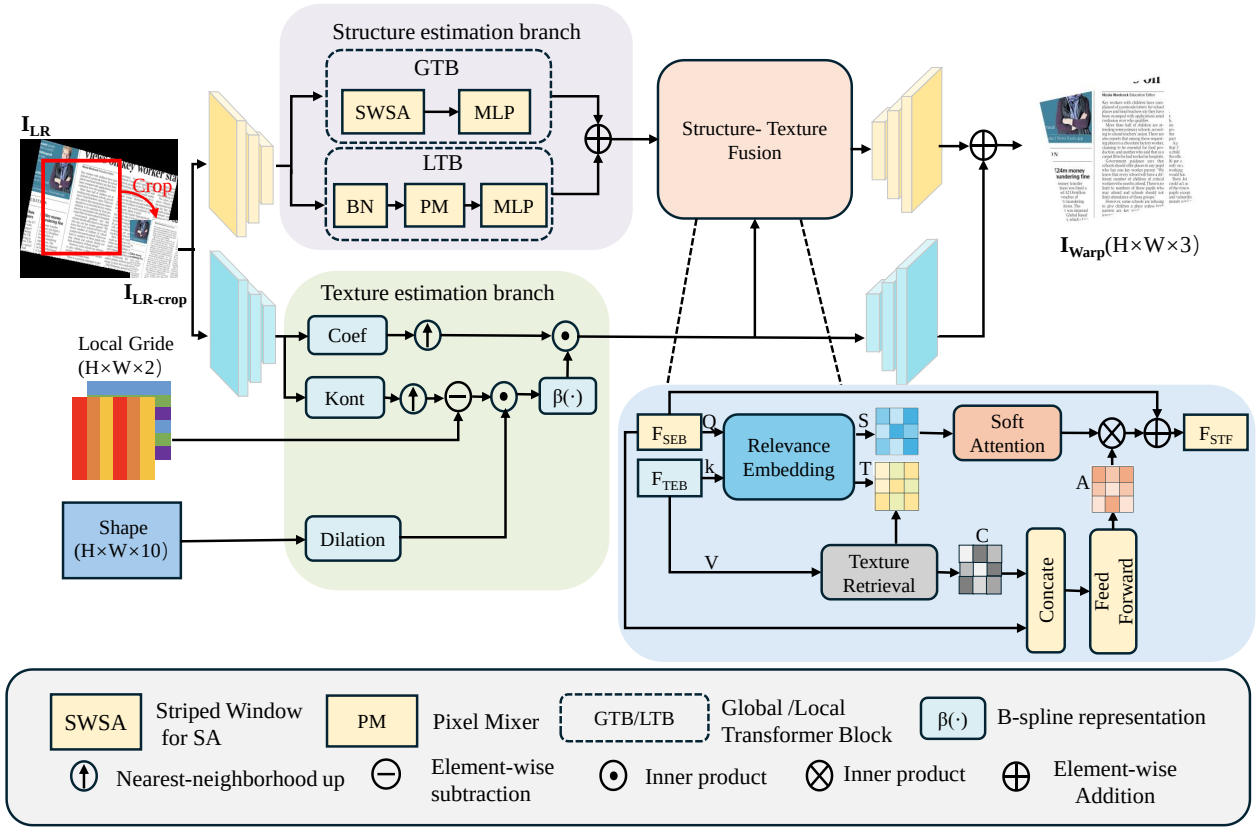

Figure 2: The overall structure of the proposed STEN model is as follows: First, the low-resolution samples I_{LR} are subjected to an inverse transformation using the inverse transformation matrix M^{-1}, followed by cropping to obtain $I_{LR-\text{Crop}}$ images. The entire model consists of two main branches: the Structure branch and the texture branch. The structure estimation branch allows us to obtain richer structural features, while the texture estimation branch enhances the network's ability to estimate texture information and local details more effectively, thus extracting texture features more efficiently. These features from both branches are then mapped into the Structure-Texture Fusion (STF) module for feature enhancement and subsequently decoded. By combining the Structure and texture branches, our STEN model can better handle deformation issues in screen content images.

reconstruction, it is also necessary to consider the structural integrity of SCIs when unwarping. Due to the powerful contextual information retention capability of the Transformer, both global and local Transformers are used to maintain SCIs' structural integrity. Thus, in this work, we propose a structure-texture enhancement network (STEN) for SCI warping, which combines implicit B-spline neural representation for texture estimation and global-local transformers for structure estimation.

3.1. Overview

The overall architecture of our proposed STEN is illustrated in Fig. 2. Our model consists of two main branches, *i.e.,* the structure estimation branch (SEB) and the texture estimation branch (TEB). Each incorporates an encoder and a decoder. Initially, we extract the features F_{LR} from the low-resolution input image using the encoder and then feed them into the Structure estimation branch and the texture estimation branch of the model.

In the Structure estimation branch, we enhance the aggregation of local knowledge through channel shift and mix operations (called pixel mixer, PM) and utilize image anisotropy for effective global dependency modelling, thereby obtaining richer structural features. Meanwhile, in the B-spline texture estimation branch, we improve the network's capability to estimate texture information and local details, thereby enhancing texture features. Given that SCIs often have discontinuous tones and many similar text regions, we design a structure-texture fusion module to globally retrieve texture features and integrate the most relevant

information into the structural features for feature-based texture enhancement. This module combines the image's structural information with texture features, effectively enhancing image details and visual effects while preserving the structural integrity of the image, resulting in clearer and more vivid images. Finally, the fused features and texture features are reconstructed through decoders separately. The decoders are constructed with multi-layer perceptrons (MLPs) and predict the RGB values at queried coordinates. The predicted values are then added to produce the final reconstructed high-resolution un-warped image.

3.2. Structure Estimation Branch

As shown in Fig. 2, the Structure estimation branch consists of two types of transformer modules: the Local Transformer Block (LTB) and the Global Transformer Block (GTB), which process the features F_{SLR} from the encoder.

In the Local Transformer Block (LTB), we first use a Batch Normalization (BM) layer to stabilize and accelerate training, and then use a Pixel Mixer (PM) layer to enhance the aggregation of local information. Specifically, PM divides the feature channels into four equally sized groups, and each group undergoes a specific shift sequence (left, right, top, bottom) to introduce locality and spatial correlations. This approach allows us to quickly capture and integrate knowledge from surrounding areas while improving the effective information exchange mechanisms between feature channels. The entire process of the Local Transformer Block (LTB) can be expressed as:

$$F_L = MLP(PM(BM(F_{LR}^S))), \tag{2}$$

where PM denotes the Pixel Mixer, BM represents Batch Normalization, and MLP refers to the multi-layer perceptron used for further feature transformation, F_{LR}^S being the input encoded features of the structure estimation branch.

In the Global Transformer Block (GTB), we construct an effective global dependency model based on self-attention (SA). We employ stripe window self-attention (SWSA) to effectively capture multi-scale symmetries and similarities present in the image. Firstly, self-attention is computed for each stripe window:

$$F_{out}^n = \text{Softmax}\left(\frac{Q^n \cdot (Q^n)^T}{\text{scale}}\right) \cdot V^n, \tag{3}$$

where n denotes the n-th stripe window, with $Q^n = F_{LR}^S \cdot W_Q$ and $V^n = F_{LR}^S \cdot W_V$ being linear transformations of F_{LR}^S using weight matrices W_Q and W_V. Subsequently, the outputs of all stripe windows are concatenated along the channel dimension:

$$F_{out} = [F_{out}^1; F_{out}^2; \ldots; F_{out}^n], \tag{4}$$

Next, F_{out} undergoes further feature transformation through a multi-layer perceptron (MLP):

$$F_G = MLP(F_{out}) \tag{5}$$

Finally, the features obtained from these two modules are concatenated:

$$F_{SEB} = F_L + F_G. \tag{6}$$

3.3. Texture Estimation Branch

Due to the spatial variability of image deformation and the discontinuity of SCIs, we propose the texture estimation branch (TEB) to predict the features of image deformation effectively. We employ non-uniform B-splines for feature embedding and multiply the estimated local texture by the local Jacobian matrix of the coordinate transformation to predict the features of the distorted image. Specifically, the implicit neural representation of non-uniform B-splines is:

$$I_{\text{warp}}[y; \theta, \psi] = \sum_{t \in N} w_t f_\theta(g_\psi(z_t, y - f(x_t), s)), \tag{7}$$

where $z = E_\phi(I_{\text{LR}})$, and N is a set defined as $N = \{N \mid N = [f^{-1}(y) + [\frac{m}{w}, \frac{n}{h}]] \mid [m, n] \in [-1, 1]\}$.In this context, w_t represents the weight of the local set, $z_t \in \mathbb{R}^D$ is a latent variable for index t, $x_t \in X \subset \mathbb{R}^2$ is a coordinate of z_t, and s is the cell value represented by a superscript factor. The local grid $\delta_y = y - f(x_t)$ represents the query point $y = f(x) \in \mathbb{R}^2$. The function $g_\psi(z_t, \delta_y, s)$ represents the B-spline texture coefficients estimator (BTC), which includes three estimators: the coefficient estimator $g_c : \mathbb{R}^D \to \mathbb{R}^C$, the structure estimator $g_k : \mathbb{R}^D \to \mathbb{R}^{2C}$, and the dilation estimator $g_d : \mathbb{R}^{10} \to \mathbb{R}^C$. The encoding function $g_\psi : (\mathbb{R}^D, \mathbb{R}^2, \mathbb{R}^{10}) \to \mathbb{R}^C$ is defined as :

$$g_\psi(z_t, \delta_y, s) = c_t \odot \text{vec}\left[\beta_n((\delta_y - k_t) \odot d)\right], \tag{8}$$

where $c_t = g_c(z_t)$, $k_t = g_k(z_t)$, and $d = g_d(s)$. To enable the model to represent distorted images, we linearize the given coordinate transformation into an affine transformation: $\delta_y = J_f(x_j)\delta_x$. Here,$J_f(x_j) \in \mathbb{R}^{2\times 2}$ represents the Jacobian matrix of the coordinate transformation f at x_j.

Our model can then extract B-spline information from distorted images by utilizing the local grid in the input coordinate space δ_x rather than in δ_y. In arbitrary scale super-resolution (SR) tasks, the pixels in the upsampled image are square and spatially invariant. However, when the image undergoes warping, the pixels in the resampling image can have arbitrary shapes and spatial transformations. To effectively represent and compute pixel shapes, we represent the pixel shape $s(y) \in \mathbb{R}^{12}$ (where $\mathbb{R}^4$ denotes pixel direction and $\mathbb{R}^4$ denotes pixel curvature) with the gradient of the coordinate transformation at point y as follows:

$$s(y) = [J_f^{-1}(y), H_f^{-1}(y)], \tag{9}$$

where $J_f^{-1}(y) \in \mathbb{R}^{2\times2}$ and $H_f^{-1}(y) \in \mathbb{R}^{2\times2\times2}$ represent the numerical Jacobian matrix in the pixel direction and the numerical Hessian tensor in the pixel curvature, which are used to describe edge positions and pixel shapes, respectively.

Therefore, we redefine the implicit neural representation of non-uniform B-splines in Eq. 8 as follows:

$$F_{TEB} = c_t \odot \text{vec}\left[\beta_n((J_f(x_j)\delta_x - k_t) \odot g_d(s(y)))\right] \quad (10)$$

3.4. Structure-Texture Fusion Module

We design a structure-texture fusion module that integrates the most relevant texture features into structural features to enhance the quality and consistency of texture synthesis. As illustrated in Fig. 2, we use nearest-neighbor interpolated features sampled from F_{SEB} as queries (Q), and utilize F_{TEB} as keys (K) and values (V). To retrieve texture features most relevant to pixel features F_{TEB}, we compute the similarity matrix R between queries Q and keys K, where each element $r_{i,j}$ is calculated according to the following formula:

$$r_{i,j} = \frac{q_i \cdot k_j}{\sqrt{q_i \cdot q_i}\sqrt{k_j \cdot k_j}} \quad (11)$$

Based on the similarity matrix R, we obtain a position index matrix T that identifies the positions of the texture features k_j most similar to each query q_i. For each query q_i, we find the position with the highest relevance, as represented by the following formula:

$$t_i = \arg max_j(r_{i,j}) \quad (12)$$

where t_i denotes the position index of the texture feature most similar to the query q_i, with values ranging from 1 to $\frac{h}{2} \times \frac{w}{2}$. Once the most relevant position t_i for each query is determined, we can use this index to extract the corresponding data from the texture branch feature map V, obtaining the retrieved texture features C as follows: $c_i = v_{t_i}$.where c_i is an element of the retrieved texture features C, and v_{t_i} represents the value at the t_i-th position in V. To fuse the retrieved texture features with the structural features F_{SEB}, we first concatenate C with F_{SEB} and process them through a feed-forward network to obtain the aggregated features A. Finally, we compute the soft attention map S, where the elements s_i in S represent the confidence of each element a_i in the retrieved texture features A, calculated as follows:

$$s_i = max_j(r_{i,j}) \quad (13)$$

The final structure-texture fusion features F_{STF} can be obtained using the following formula:

$$F_{STF} = F_{SEB} + (A \otimes S) \quad (14)$$

3.5. The Output of STEN Model

As illustrated in Fig. 2, two decoders, each implemented by an MLP, are individually used to predict the corresponding outputs based on the fused features and texture features mentioned above. Specifically, we input the texture features F_{TEB} into the MLP f_θ, allowing them to be directly decoded into RGB values. Simultaneously, the structure-texture fusion features F_{STF} are also processed through the MLP f_θ to produce additional RGB values. Finally, we sum the RGB values obtained from both decoding processes to generate the final output image $I_{P_{red}}$. The relationship can be expressed mathematically as follows:

$$I_{P_{red}} = f_\theta(F_{STF}) + f_\theta(F_{TEB}) \quad (15)$$

3.6. Transformation Self-estimation Module

The proposed method requires two inputs: the input warped image I and the transformation matrix. In practical applications, the transformation matrix is often unknown and needs to be estimated. Motivated by blur kernel estimation for blind super-resolution [11], we first construct and train a convolutional neural network to predict the transformation error for the estimated transformation matrix based on the warped input image, then we update the estimation of the transformation matrix based on gradient-descend algorithm. This optimization procedure is executed iteratively till the termination condition is satisfied. During the training of the transformation error estimation CNN, we treat the difference between the image generated by the estimated

Algorithm 1: Transformation Self-estimation

I_w: The input warped image;
$m_1, m_2, \ldots, m_n$: The initialized transformations;
E: The well-trained transformation error estimator;
n, T: The samples' number and iterations' number;
$E^*_{\text{min}} \leftarrow \infty$;
for $i \in 1..n$ **do**
  $I^* \leftarrow \text{STEN}(I_w, m_i)$;
  $E^* \leftarrow E(I^*)$;
  if $E^* < E^*_{min}$ **then**
    $E^*_{\text{min}} \leftarrow E^*$;
    $m \leftarrow m_i$;
  end
end
for $j \in 1..T$ **do**
  $I^* \leftarrow \text{STEN}(I_w, m)$;
  $e \leftarrow E(I^*)$;
  $loss \leftarrow e + \alpha \cdot |m|$;
  $loss$.backward(m);
end
Output: m;

Train set: SCI1K (n=800)		In-training-scale			Out-of-training-scale			
Test set	Method	×2	×3	×4	×5	×6	×7	×8
SCI1K (n = 200)	Bicubic	28.81	25.15	23.18	22.02	21.23	20.72	20.26
	RDN [44]	38.45	33.59	29.81	-	-	-	-
	MetaSR [17]	38.57	33.67	30.12	27.52	26.13	23.91	23.19
	LIIF [7]	38.65	33.97	30.55	27.77	26.07	23.99	23.24
	LTE [20]	39.14	34.50	30.93	28.22	26.19	24.28	23.17
	BTC[27]	39.17	34.58	31.10	28.33	26.31	**24.47**	23.38
	STEN(Ours)	**39.22**	**34.87**	**31.33**	**28.34**	**26.38**	24.37	**23.40**
SCID (n = 40)	Bicubic	25.22	22.78	21.60	20.90	20.42	20.04	19.77
	RDN [44]	34.00	28.34	25.74	-	-	-	-
	MetaSR [17]	33.84	29.08	25.76	23.62	22.38	21.59	21.07
	LIIF [7]	34.24	29.10	25.89	23.77	22.53	21.73	21.21
	LTE [20]	34.49	29.60	26.34	24.06	22.67	21.81	21.28
	BTC[27]	34.48	29.56	26.30	24.09	**22.69**	21.84	**21.29**
	STEN(Ours)	**34.59**	**29.81**	**26.60**	**24.15**	22.65	**21.85**	21.28
SCIAQ (n = 22)	Bicubic	22.89	20.66	19.70	19.18	18.79	18.46	18.20
	RDN [44]	33.53	26.89	23.38	-	-	-	-
	MetaSR [17]	34.12	28.40	23.55	21.18	20.18	19.63	19.25
	LIIF [7]	34.31	28.27	23.44	21.16	20.25	19.70	19.36
	LTE [20]	35.07	29.33	24.21	21.52	20.39	19.78	19.43
	BTC [27]	34.91	29.36	24.25	21.57	**20.43**	19.82	19.45
	STEN(Ours)	**35.20**	**29.71**	**24.60**	**21.63**	20.40	**19.85**	**19.48**

Table 1: Quantitative comparison of arbitrary scale super-resolution methods on SCI1K, SCID, and SIQAD datasets within in-scale (PSNR (dB)). The best results are highlighted in bold, while the second-best results are indicated with underlined.

transformation and the warped input image as the loss function. This approach ensures that the input image can be restored to its original state after undergoing cycle-consistent translations [47], maintaining feature consistency between the input and output.

Once the transformation error estimator (error estimation CNN) is well-trained, we can estimate the unknown transformation in real-world scenarios. Concretely, given a warped image I, the transformation error estimator can obtain the transformation difference. Based on the difference, we can update the estimated transformation matrix iteratively and finally get a good approximation of the genuine transformation. Assuming M is the estimated transformation matrix, the transformation estimation optimization can be described as:

$$M^* = \arg\min_M \left(E(S, M, I) + \alpha |M|\right) \tag{16}$$

where E represents the transformation error estmatior, S is our proposed unwarping model (STEN), $|M|$ denotes the size of the matrix M, and $\alpha \geq 0$ is a tunable parameter. The description of the transformation self-estimation algorithm is detailed in Algorithm. 1.

4. Experiments

4.1. Implementation Details

Dataset We use the existing SCI1K, SCID, and SCIAQ dataset [42] to construct our warped dataset - SCI1KW, SCIDW, and SCIAQW to train and test our STEN model. Specifically, we first randomly generate a warping matrix, which includes transformations such as random scaling, cropping, rotation, and projection. Then, we distort each image in SCI1K with the warping matrix to obtain a distorted version and finally form the SCI1KW datasets. During training prepossessing, we crop the maximum effective region of the warped result as the input image according to the transformation effect. In the testing phase, we centre-crop the 384×384 size patch from each test image in the test set and assign a transformation on the patch. This method allows us to effectively evaluate the model's performance under different warping conditions. **Settings** For optimization, we employ L1 loss [25] and the Adam optimizer [18] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. Our STEN model is trained for 1000 epochs with a batch size 16, starting with a learning rate of 1×10^{-4}, halving it every 200 epochs. In the transformation self-estimation algorithm, we set the adjustable parameter $\alpha = 0.05$. All compared methods used in this work adopt the same settings as our STEN model.

Figure 3: Qualitatively compare to other arbitrary scale super-resolution (SR) methods, *i.e.,* MetaSR [17], LIIF[7], LTE[20], and BTC[27], at scales $\times 2$ and $\times 3.5$ within in scale.

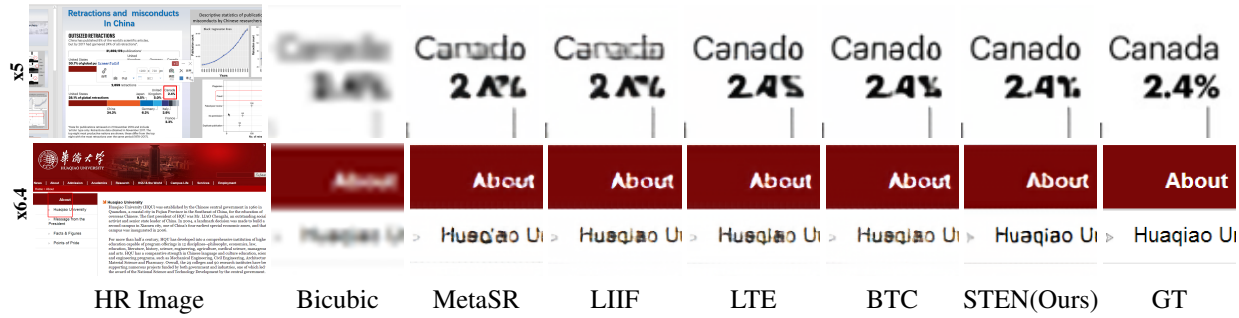

Figure 4: Qualitatively compare to other arbitrary scale super-resolution (SR) methods, *i.e.,* MetaSR [17], LIIF[7], LTE[20], and BTC[27], at scales $\times 2$ and $\times 3.5$ within in scale. at scales x5 and x6.4 out of scale.

4.2. Comparisions for Arbitrary Scale Super-Resolution

We compare our proposed STEN model with BTC[27], LTE[20], LIIF[7], and MetaSR[17] for arbitrary-scale image super-resolution in both in-scale and out-of-scale scenarios. To ensure a fair comparison, all methods use RDN[44] as the encoder for feature extraction and are trained on the SCI1K dataset. As shown in Table 1,our method outperforms existing arbitrary-scale super-resolution techniques across most scale factors and datasets.

Additionally, Fig. 3 and Fig. 4 show a qualitative comparison between our method and the others, demonstrating reconstruction performance at different scales. Whether in-scale or out-of-scale, our method consistently surpasses others in recovering text and graphical details, highlighting its clear advantages in detail preservation and visual quality.

4.3. Comparisions for Homography Warping

We compare our STEN model with SRWarp[33] and LTEW[19] on the generated benchmark datasets (SCI1KW, SCIDW, and SIQADW) with homography warping transformation, including in-distribution and out-of-distribution scenarios. In the in-scale scenario, we consider the scale factors present in the training dataset, while in the out-of-scale scenario, the scale factors were not included in the training dataset. To ensure a fair comparison, all methods use RDN[44] as the encoder for feature extraction and are trained on the SCI1K dataset. Additionally, we include results from bilinear interpolation and the RDN model. For

Method	SCI1KW		SCIDW		SIQADW	
	isc	osc	isc	osc	isc	osc
Bicubic	24.93	23.04	22.43	21.02	20.36	19.21
RDN[44]	33.81	25.08	28.54	22.39	27.24	21.04
SRWarp-RDN[33]	35.93	27.65	31.65	24.38	31.24	21.89
LTEW-RDN[19]	<u>36.18</u>	<u>28.55</u>	<u>32.17</u>	<u>25.91</u>	<u>31.67</u>	<u>22.31</u>
STEN-RDN(est)	25.84	22.78	22.94	20.68	20.43	18.62
STEN-RDN(Ours)	**36.97**	**29.29**	**32.49**	**26.17**	**32.15**	**22.59**

Table 2: Quantitative comparison of homography transform methods on SCI1KW, SCIDW, and SIQADW datasets within in-scale (isc) and out-of-scale (osc) (PSNR (dB)). The best results are highlighted in bold, while the second-best results are indicated with underlined.

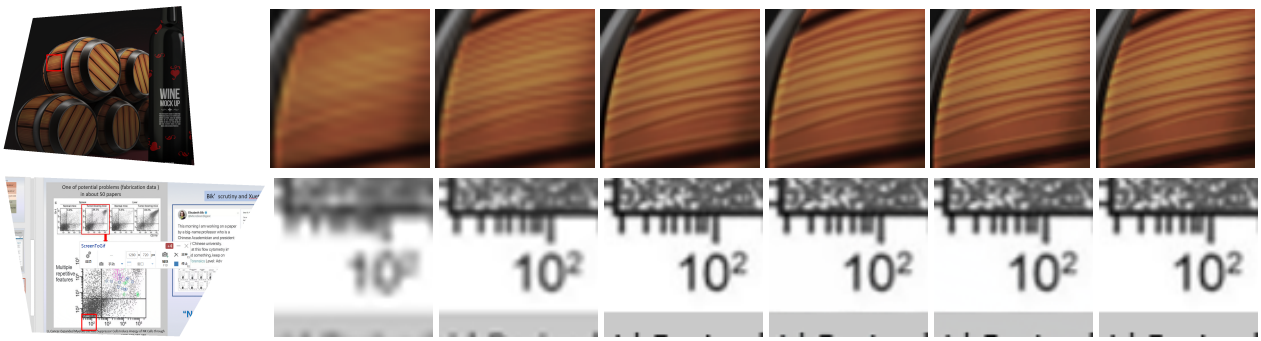

Figure 5: Qualitative comparison to other homography transform methods, *i.e.,*RDN[44],SRWarp[33],LTEW [19] within in-scale.

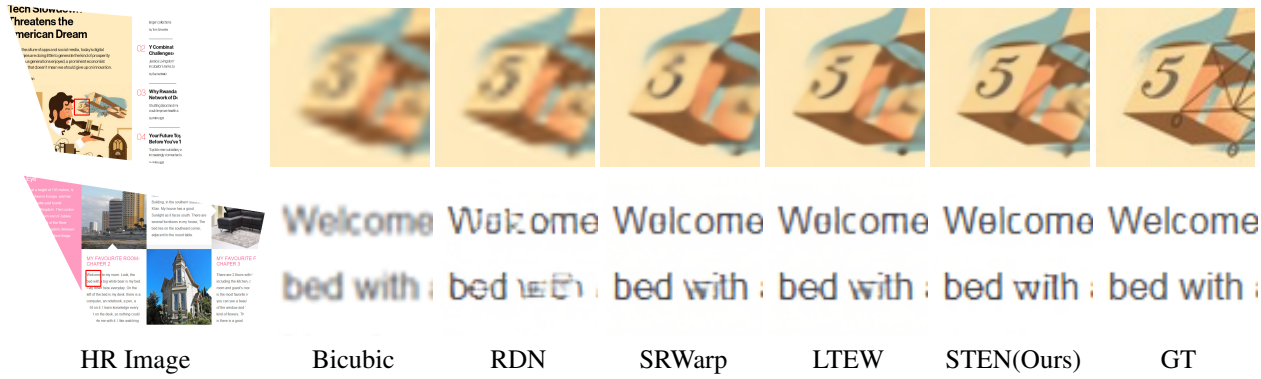

Figure 6: Qualitative comparison to other homography transform methods, *i.e.,*RDN[44],SRWarp[33],LTEW [19] within out-scale.

RDN , we upsample the input images and use the WarpPerspective function[5] with a bicubic kernel to synthesize the warped images. For SRWarp and LTEW, we implement them using their publicly available source code.

Table 2 presents the average mPSNR results of our method compared to other methods on benchmark datasets under both in-scale and out-of-scale conditions. It can be observed that our method outperforms LTEW[19] in both cases. Additionally, we validate the effectiveness of the transformation self-estimation model, where STEN(est) is the result obtained using the estimated transformation matrix M_i. From Table 2, it is evident that although the performance metrics of STEN(est) are lower, it can still effectively predict the transformation matrix. Furthermore, Fig. 5 and Fig. 6 provide qualitative comparisons of in-scale and out-of-scale methods, showing that our approach generates text and texture details more clearly. In summary, our method outperforms existing homography transformation techniques in terms of both mean Peak Signal-to-Noise Ratio (mPSNR) and visual quality.

TEB	SEB	STF	B-splines	Fourier	SCI1KW isc	SCI1KW osc
✓	✗	✗	✓	✗	36.65	29.01
✓	✗	✓	✓	✗	36.80	29.14
✓	✓	✗	✓	✗	36.79	29.10
✓	✓	✓	✗	✓	36.83	29.13
✓	✓	✓	✓	✗	**36.97**	**29.29**

Table 3: For structure estimation branch (SEB) and Structure-Texture Fusion Module (STF),as well as B-spline representation and Fourier representation, quantitative ablation experiments (mPSNR (dB)) are conducted for homography transformations within in-scale (isc) and out-of-scale (osc) scenarios on the SCI1K dataset.

4.4. Ablation Study

To validate the importance of each module in our proposed method, particularly the structure estimation branch

Method	DIV2K		Set5W		Set14W		B100W		Urban100W	
	isc	osc	isc	osc	isc	osc	isc	osc	isc	osc
Bicubic	27.85	25.03	35.00	28.75	28.79	24.57	28.67	25.02	24.84	21.89
RRDB[40]	30.76	26.84	37.40	30.34	31.56	25.95	30.29	26.32	28.83	23.94
SRWarp-RRDB[33]	31.04	26.75	37.93	29.90	32.11	25.35	30.48	26.10	29.45	24.04
LTEW-RRDB[19]	31.10	26.92	38.20	31.07	32.15	26.02	30.56	26.41	29.50	24.25
MFR-RRDB[41]	31.18	27.12	**38.23**	**31.19**	32.26	26.26	30.62	**26.53**	**29.68**	**24.51**
STEN-RRDB(Ours)	**31.96**	**27.75**	38.12	31.10	**32.30**	**26.27**	**30.88**	26.45	29.50	24.26

Table 4: Quantitative comparison of homography transform methods on natural image benchmarks: DIV2KW, Set5W, Set14W, B100W, and Urban100W (PSNR (dB)). The best results are highlighted in bold, while the second-best results are indicated with underlined.

(SEB) and the structure-texture fusion module (STF), we design four different network architectures on the SCI1KW dataset and conduct ablation experiments in both in-scale and out-of-scale scenarios. In the experiments, we first explore the importance of the SEB module. To do this, we remove the SEB module from the model framework and perform feature fusion using direct upsampling of features. As shown in Table 3, the use of the SEB module significantly improves performance metrics in both in-scale and out-of-scale cases, indicating that the SEB plays a crucial role in feature aggregation and information extraction. Next, we assess the effectiveness of the STF module. In this experiment, we directly combine the features from the Structure Estimation Branch and the Texture Estimation Branch before decoding. The results demonstrate that removing the STF module leads to a substantial decline in all performance metrics, highlighting its significance in detail reconstruction and visual quality enhancement.

When both the SEB and STF modules are absent, the model performs the worst, further validating the synergistic and essential roles of these two modules in our framework. These experimental results clearly indicate that our proposed modules not only effectively enhance the overall performance of the model but also better preserve details and improve visual quality when handling complex images.Meanwhile, to validate the effectiveness of B-splines, we conducted another ablation experiment, where we used Fourier representation in addition to B-splines. The results show that using Fourier representation leads to a significant decline in all performance metrics, which validates that B-splines are more effective than Fourier representation for handling screen content images.

4.5. Comparisons for Natural Images

To further validate the performance of our model on real-world natural images, we retrain it on the widely-used DIV2K dataset[2]. In these experiments, we evaluate STEN on several benchmark datasets, including DIV2KW, Set5W, Set14W, B100W, and Urban100W[33]. The results are presented in Table. 4, demonstrating that STEN performs well on natural image benchmarks, despite a slight decline in performance in certain cases. These results further highlight the robustness of STEN across a wide range of datasets, including more diverse natural images, making it applicable to practical scenarios.

Figure 7: Quantitative comparison on real-world images.

4.6. Comparisons for the Real-world Images

We conducted experiments on real-world images to validate the effectiveness of our proposed error estimation model. In the distorted images captured using a camera, the transformation matrix parameters were unknown at the time of capture. Subsequently, we applied the trained error estimation model to estimate these unknown parameters. Due to the lack of ground truth images (GT), we were unable to perform quantitative analysis; therefore, we opted for a qualitative comparison between STEN and SRWarp[33] as well as LTEW[19]. As shown in Fig. 7, we made predictions for both screen content images and natural images. The results indicate that our error estimation model can ef-

fectively predict the parameters of the transformation matrix, thereby validating the model's effectiveness.

5. Conclusion

In this work, we propose an innovative structure-texture enhancement dual-branch model- STEN, to achieve arbitrary scale super-resolution (SR) and homography unwarping. Our STEN model includes a structure estimation branch and a texture estimation branch. The texture estimation branch uses the implicit B-spline neural representation of transformed images to handle coordinate transformation and improve the synthesized texture details. At the same time, the structure estimation branch enhances structural features by using global transformation blocks and local transformation blocks, facilitating effective global modeling and thereby better representing the overall shape and structure of the image. Additionally, our STEN model also introduces the transformation self-estimation module to process unknown transformation cases in real-world scenarios. Experiments demonstrate that our proposed STEN method significantly outperforms existing techniques in homography un-warping tasks across three publicly available SCI datasets, five natural image datasets, and the real-world images.

Acknowledgement

This work is supported in part by the National Natural Science Foundation of China under Grant No. 61971004, the Natural Science Foundation of Anhui Province, China (Grant No. 2008085MF190), the University Synergy Innovation Program of Anhui Province, China (NO. GXXT-2022-044), and the Natural Science Research Project of Anhui Educational Committee under Grant No. 2024AH050161.

References

[1] M. K. Agoston and M. K. Agoston. *Computer graphics and geometric modeling*, volume 1. Springer, 2005. 2

[2] E. Agustsson and R. Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 126–135, 2017. 10

[3] M. Alemán-Flores, L. Alvarez, L. Gomez, and D. Santana-Cedrés. Automatic lens distortion correction using one-parameter division models. *Image Processing On Line*, 4:327–343, 2014. 1

[4] V. Ashish. Attention is all you need. *Advances in neural information processing systems*, 30:I, 2017. 2

[5] G. Bradski. The opencv library. *Dr. Dobb's Journal of Software Tools*, 2000. 9

[6] D. Chen, Y. Zhang, and S. Li. Tracking control of robot manipulators with unknown models: A jacobian-matrix-adaption method. *IEEE Transactions on Industrial Informatics*, 14(7):3044–3053, 2017. 2

[7] Y. Chen, S. Liu, and X. Wang. Learning continuous image representation with local implicit image function. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8628–8638, 2021. 2, 3, 7, 8

[8] M.-C. Chiang and T. E. Boult. Efficient image warping and super-resolution. In *Proceedings Third IEEE Workshop on Applications of Computer Vision. WACV'96*, pages 56–61. IEEE, 1996. 1

[9] M.-C. Chiang and T. E. Boult. Efficient super-resolution via image warping. *Image and Vision Computing*, 18(10):761–771, 2000. 1

[10] K. Chitra and C. Vennila. Retracted article: A novel patch selection technique in ann b-spline bayesian hyperprior interpolation vlsi architecture using fuzzy logic for highspeed satellite image processing. *Journal of Ambient Intelligence and Humanized Computing*, 12(6):6491–6504, 2021. 3

[11] V. Cornillere, A. Djelouah, W. Yifan, O. Sorkine-Hornung, and C. Schroers. Blind image super-resolution with spatially variant degradations. *ACM Transactions on Graphics (TOG)*, 38(6):1–13, 2019. 6

[12] T. Dai, J. Cai, Y. Zhang, S.-T. Xia, and L. Zhang. Second-order attention network for single image super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11065–11074, 2019. 1

[13] R. J. Duffin and A. C. Schaeffer. A class of nonharmonic fourier series. *Transactions of the American Mathematical Society*, 72(2):341–366, 1952. 3

[14] J. D. Foley. *Computer graphics: principles and practice*, volume 12110. Addison-Wesley Professional, 1996. 3

[15] D. Gottlieb and C.-W. Shu. On the gibbs phenomenon and its resolution. *SIAM review*, 39(4):644–668, 1997. 2

[16] S. Gottlieb, J.-H. Jung, and S. Kim. A review of david gottlieb's work on the resolution of the gibbs phenomenon. *Communications in Computational Physics*, 9(3):497–519, 2011. 2

[17] X. Hu, H. Mu, X. Zhang, Z. Wang, T. Tan, and J. Sun. Meta-sr: A magnification-arbitrary network for super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1575–1584, 2019. 7, 8

[18] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 7

[19] J. Lee, K. P. Choi, and K. H. Jin. Learning local implicit fourier representation for image warping. In *European Conference on Computer Vision*, pages 182–200. Springer, 2022. 1, 2, 3, 8, 9, 10

[20] J. Lee and K. H. Jin. Local texture estimator for implicit representation function. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1929–1938, 2022. 2, 3, 7, 8

[21] T. M. Lehmann, C. Gonner, and K. Spitzer. Addendum: B-spline interpolation in medical image processing. *IEEE transactions on medical imaging*, 20(7):660–665, 2001. 3

[22] L. Li. Application of cubic b-spline curve in computer-aided animation design. *Computer-Aided Design and Applications*, 18(S1):43–52, 2020. 3

[23] L. Li, W. Liu, and W. Xing. Robust radial distortion correction from a single image. In *2017 IEEE*

15th Intl Conf on Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence and Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech), pages 766–772. IEEE, 2017. 1

[24] J. Liang, J. Cao, G. Sun, K. Zhang, L. Van Gool, and R. Timofte. Swinir: Image restoration using swin transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1833–1844, 2021. 2

[25] B. Lim, S. Son, H. Kim, S. Nah, and K. Mu Lee. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 136–144, 2017. 1, 7

[26] T. C. Mok and A. C. Chung. Large deformation diffeomorphic image registration with laplacian pyramid networks. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part III 23*, pages 211–221. Springer, 2020. 2

[27] B. Pak, J. Lee, and K. H. Jin. B-spline texture coefficients estimator for screen content image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10062–10071, 2023. 2, 3, 7, 8

[28] A. D. Prasad, A. Balu, H. Shah, S. Sarkar, C. Hegde, and A. Krishnamurthy. Nurbs-diff: A differentiable programming module for nurbs. *Computer-Aided Design*, 146:103199, 2022. 3

[29] D. Santana-Cedrés, L. Gomez, M. Alemán-Flores, A. Salgado, J. Esclarín, L. Mazorra, and L. Alvarez. An iterative optimization algorithm for lens distortion correction using two-parameter models. *Image Processing On Line*, 6:326–364, 2016. 1

[30] P.-E. Sarlin, A. Unagar, M. Larsson, H. Germain, C. Toft, V. Larsson, M. Pollefeys, V. Lepetit, L. Hammarstrand, F. Kahl, et al. Back to the feature: Learning robust camera localization from pixels to pose. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3247–3257, 2021. 1

[31] I. J. Schoenberg. Contributions to the problem of approximation of equidistant data by analytic functions. part b. on the problem of osculatory interpolation. a second class of analytic approximation formulae. *Quarterly of Applied Mathematics*, 4(2):112–141, 1946. 3

[32] V. Sitzmann, J. Martel, A. Bergman, D. Lindell, and G. Wetzstein. Implicit neural representations with periodic activation functions. *Advances in neural information processing systems*, 33:7462–7473, 2020. 2, 3

[33] S. Son and K. M. Lee. Srwarp: Generalized image super-resolution under arbitrary transformation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7782–7791, 2021. 1, 2, 3, 8, 9, 10

[34] Y. Umetani, K. Yoshida, et al. Resolved motion rate control of space manipulators with generalized jacobian matrix. *IEEE Transactions on robotics and automation*, 5(3):303–314, 1989. 2

[35] M. Unser, A. Aldroubi, and M. Eden. B-spline signal processing. ii. efficiency design and applications. *IEEE transactions on signal processing*, 41(2):834–848, 1993. 3

[36] M. Unser, A. Aldroubi, and M. Eden. B-spline signal processing. ii. efficiency design and applications. *IEEE transactions on signal processing*, 41(2):834–848, 1993. 3

[37] S. Van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, and T. Yu. scikit-image: image processing in python. *PeerJ*, 2:e453, 2014. 2

[38] L. Von Stumberg, P. Wenzel, N. Yang, and D. Cremers. Lm-reloc: Levenberg-marquardt based direct visual relocalization. In *2020 International Conference on 3D Vision (3DV)*, pages 968–977. IEEE, 2020. 1

[39] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis. Deep learning for computer vision: A brief review. *Computational intelligence and neuroscience*, 2018(1):7068349, 2018. 2

[40] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, and C. Change Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In *Proceedings of the European conference on computer vision (ECCV) workshops*, pages 0–0, 2018. 10

[41] J. Xiao, Z. Lyu, C. Zhang, Y. Ju, C. Shui, and K.-M. Lam. Towards progressive multi-frequency representation for image warping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2995–3004, 2024. 3, 10

[42] J. Yang, S. Shen, H. Yue, and K. Li. Implicit transformer network for screen content image continuous super-resolution. *Advances in Neural Information Processing Systems*, 34:13304–13315, 2021. 7

[43] W. Yang, X. Zhang, Y. Tian, W. Wang, J.-H. Xue, and Q. Liao. Deep learning for single image super-resolution: A brief review. *IEEE Transactions on Multimedia*, 21(12):3106–3121, 2019. 1

[44] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu. Residual dense network for image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2472–2481, 2018. 7, 8, 9

[45] L. Zheng, J. Zhu, J. Shi, and S. Weng. Efficient mixed transformer for single image super-resolution. *Engineering Applications of Artificial Intelligence*, 133:108035, 2024. 2

[46] K. Zhou, J. Huang, J. Snyder, X. Liu, H. Bao, B. Guo, and H.-Y. Shum. Large mesh deformation using the volumetric graph laplacian. *ACM Transactions on Graphics*, 24(3):496–503, 2005. 2

[47] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017. 7