Postprint: Image Interpolation Algorithm Preserving Texture Details and Edge Structures

Authors: Fu Pengbin, Tie Huijie, Yang Huirong

Date: 2020-09-28T00:00:00+00:00

Abstract

To address the issues of edge structure degradation and texture detail loss during image reconstruction, we propose an image interpolation algorithm that preserves texture details and edge structures. First, image regions are segmented using eight-directional edge detection with adaptive thresholds. Second, a bivariate rational function model is constructed, which can transition between rational and polynomial models. Finally, we propose a method that adjusts the spatial distance of interpolation points based on the local asymmetry and gradient characteristics of edge data; coordinates of interpolation points in edge regions are adjusted and input into the rational model for interpolation, while non-edge regions utilize polynomial model interpolation. Experimental results indicate that the proposed algorithm achieves an average improvement of 0.48dB-2.17dB in Peak Signal-to-Noise Ratio (PSNR) and 0.004-0.028 in Structural Similarity Index (SSIM), yielding superior objective evaluation metrics. By modifying conventional fixed spatial distance interpolation to variable spatial distance interpolation, the algorithm effectively preserves image edge structures and texture details, resulting in reconstructed images with enhanced visual quality.

Full Text

Texture Detail and Edge Structure Preserving Image Interpolation Algorithm

Fu Pengbin, Tie Huijie, Yang Huirong
(Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China)

Abstract

To address the problems of edge structure degradation and texture detail loss during image reconstruction, this paper proposes an image interpolation algorithm that preserves texture details and edge structures. First, an eight-directional edge detection method with adaptive thresholding is employed to partition the image into distinct regions. Second, a bivariate rational function model is constructed, which can switch between rational and polynomial formulations. Finally, a method is proposed to adjust the spatial distance of interpolation points based on the local asymmetry of edge data and gradient features. This approach modifies the coordinates of points to be interpolated in edge regions and substitutes them into the rational model for interpolation, while non-edge regions utilize polynomial model interpolation. Experimental results demonstrate that the proposed algorithm achieves average improvements of 0.48–2.17 dB in Peak Signal-to-Noise Ratio (PSNR) and 0.004–0.028 in Structural Similarity Index (SSIM), yielding superior objective evaluation metrics. By transforming conventional spatially-invariant interpolation into spatially-varying interpolation, the algorithm effectively preserves edge structures and texture details, producing reconstructed results with enhanced visual quality.

Keywords: image interpolation; texture detail and edge structure preservation; adaptive gradient; rational function; spatial distance correction

Image interpolation technology refers to the image processing technique of transforming low-resolution images into high-resolution images through interpolation. It finds extensive applications in remote sensing imaging, medical imaging, public security, computer vision, digital entertainment, and multimedia communications.

Conventional interpolation algorithms can be categorized into two classes based on their approach: discrete interpolation algorithms and continuous interpolation algorithms. Discrete methods, such as nearest-neighbor and bilinear interpolation [**?**], establish transformations based on intrinsic relationships between known pixel values to determine unknown pixel values, but often produce blurred and distorted results. Continuous methods, including bicubic [**?**] and cubic spline [**?**] algorithms, construct interpolation surfaces from known pixel values to calculate unknown pixels, yet they also introduce jaggies and blurring. To address these issues, Li et al. [**?**] proposed an edge-directed interpolation algorithm with improved edge reconstruction but susceptible to aliasing and noise in texture details. Liu et al. [**?**] introduced a rational image interpolation algorithm that preserves texture details and enables arbitrary-scale magnification through parameter optimization. Building upon this rational interpolation function, subsequent works [**?**, **?**, **?**] proposed improvements that effectively maintain texture details but exhibit unsatisfactory edge structure preservation.

To achieve smooth edge transitions and texture detail fidelity, this paper draws inspiration from bivariate rational interpolation models [**?**, **?**] to design a texture detail and edge structure preserving image interpolation algorithm. Based on the bivariate rational interpolation model, the algorithm first adjusts the spatial distance of points to be interpolated in edge regions according to the local asymmetry of edge data, establishing preliminary coordinates. It then obtains the horizontal and vertical gradient angles at interpolation points to compute gradient weights of known pixels along these directions. Using these weights, an adaptive gradient-based spatial distance correction is performed to obtain refined coordinates for edge region interpolation points. The proposed algorithm retains the high texture fidelity of the original interpolation model while upgrading from spatially-invariant to spatially-varying interpolation, thereby enhancing smoothness in edge regions. Experimental results show that our algorithm achieves the highest average PSNR and SSIM values, demonstrating superior objective metrics while producing clear textures and smooth edges with excellent visual quality.

1. Bivariate Rational Interpolation Function

Given an image of size $m \times n$, the original image region is $[1, m; 1, n]$. Suppose the image is enlarged horizontally to width u and vertically to height v, resulting in an enlarged image region of $[1, u; 1, v]$. Let (x_i, y_j) denote the coordinates of original pixels and $f_{i,j}$ their corresponding pixel values. For any interpolation point (x^*, y^*) in the interpolation region W, where $x \in [x_i, x_{i+1}]$ and $y \in [y_j, y_{j+1}]$, we construct directional interpolation curves using partial derivatives $d_{i,j} = \frac{\partial f}{\partial x}(x_i, y_j)$ and $e_{i,j} = \frac{\partial f}{\partial y}(x_i, y_j)$.

The x-direction interpolation function is defined as:

$$t_{i,j}(x) = P_{i,j}(x) + \frac{W_{i,j}(x)}{V_{i,j}(x)} \cdot (x - x_i)(x - x_{i+1})$$

where $P_{i,j}(x)$, $V_{i,j}(x)$, and $W_{i,j}(x)$ are polynomial functions. Similarly, the y-direction function $q_{i,j}(y)$ is constructed. The bivariate rational interpolation function is then defined as:

$$t_{i,j}(x, y) = P_{i,j}(x, y) + \frac{W_{i,j}(x, y)}{V_{i,j}(x, y)} \cdot (x - x_i)(x - x_{i+1})(y - y_j)(y - y_{j+1})$$

This interpolation function is determined jointly by interpolation data and shape parameters. When the shape parameters $\alpha_{i,j}$ and $\beta_{i,j}$ take different values, the function form changes accordingly. If $\alpha_{i,j} = \beta_{i,j} = 1$, the function reduces to a polynomial interpolation.

2. Texture Detail and Edge Structure Preserving Image Interpolation Algorithm

The bivariate rational interpolation-based algorithm effectively preserves texture details but suffers from directional asymmetry, causing jaggies in edge regions while performing well in non-edge areas. Natural image edges exhibit sigmoid curve distributions due to low-pass filtering during acquisition, displaying local asymmetry characteristics [?]. Image gradients reflect intensity variations and serve as crucial cues for estimating unknown information. Therefore, this algorithm adjusts pixel spatial distances using local asymmetry and gradient information to make the interpolation surface closer to the ideal shape.

The algorithm flowchart is shown in [Figure 1: see original paper]. The process involves: (1) extracting eight-directional gradient features from the low-resolution image and using simple statistical thresholding to partition the image into edge and non-edge regions; (2) adjusting spatial distances of interpolation points in edge regions based on local asymmetry to obtain initial coordinates, then computing horizontal and vertical gradient angles to calculate gradient weights; (3) performing adaptive gradient-based spatial distance correction to obtain refined coordinates; and (4) applying polynomial interpolation to non-edge regions and rational function interpolation with corrected coordinates to edge regions.

2.1 Image Region Segmentation

Edge and non-edge regions exhibit distinct structural characteristics. Edge regions are structurally complex with large pixel value variations, while non-edge regions are simple with minimal variations. To improve interpolation accuracy, different models are applied to different regions.

This paper employs an eight-directional Scharr operator-based edge detection algorithm for region segmentation. First, eight-directional Scharr templates convolve with the image to extract gradient information. Then, an adaptive threshold for each pixel is computed using simple statistics:

$$T_{i,j} = \frac{1}{8}\sum_{k=1}^{8} |G_k(i,j)|$$

where $G_k(i,j)$ represents gradient values in eight directions. Finally, each pixel is classified as edge or non-edge by comparing its gradient magnitude with the threshold.

The adaptive-threshold eight-directional Scharr edge detection effectively identifies line segments at various angles, producing continuous and complete edge contours. [Figure 2: see original paper] shows edge detection results for the Lena image, demonstrating accurate detection of valid edges.

2.2 Adaptive Gradient Spatial Distance Adjustment

Based on the importance of local asymmetry and gradient features for edge structure preservation, this section proposes a combined spatial distance correction method.

2.2.1 Local Asymmetry-Based Spatial Distance Correction Since edge data follows a sigmoid distribution due to low-pass filtering, we model edges using the sigmoid curve shown in [Figure 3: see original paper]. For known points P_i and P_{i+1} with pixel values f_i and f_{i+1}, conventional interpolation directly substitutes the interpolation point P' coordinate into the interpolation function, considering only the distance s to known points while ignoring other factors. This creates a deviation between the actual position P'_{real} and ideal position P'_{ideal}, yielding inaccurate pixel values.

To obtain more accurate edge interpolation, we adjust the distance between interpolation and known points. As illustrated in [Figure 3: see original paper], when the distance between P' and P_i is s, the interpolated value is f'_{real}, but adjusting the distance to s' yields the ideal value f'_{ideal}.

Based on local data asymmetry [**?**], we adjust the spatial distance in one dimension. Let f' be the pixel value at interpolation point P' with coordinate x', and adjacent points P_{i-1} and P_{i+1} have coordinates x_{i-1} and x_{i+1}. The distance between P' and P_i is defined as $s = x' - x_i$. The corrected distance s' is computed using:

$$s' = s + k \cdot A \cdot L$$

where A represents geometric similarity in the interpolation point' s neighborhood, defined as:

$$A = \frac{|f_{i+1} - f'| - |f' - f_{i-1}|}{|f_{i+1} - f_{i-1}|}$$

If $A = 0$, the neighborhood is symmetric; if $A > 0$, the point is closer to the right pixel; if $A < 0$, it' s closer to the left. L is the maximum gray level (256 for 8-bit images) to ensure $A \in [-1, 1]$, and k is a correction factor (typically 1 or 2).

This one-dimensional adjustment extends to two dimensions as shown in [Figure 4: see original paper]. Horizontal and vertical corrections s'_h and s'_v are computed separately using pixel values along each direction. These one-dimensional corrections are then integrated into two-dimensional space using the geometric relationships illustrated in [Figure 5: see original paper], yielding corrected horizontal and vertical distances s'_x and s'_y through:

$$s'_x = \frac{s'_{h1} \cdot s'_{v2} - s'_{h2} \cdot s'_{v1}}{s'_{v2} - s'_{v1}}, \quad s'_y = \frac{s'_{h1} \cdot s'_{v2} - s'_{h2} \cdot s'_{v1}}{s'_{h1} - s'_{h2}}$$

2.2.2 Gradient Angle Computation Given a low-resolution image, we first obtain an initial high-resolution version through bicubic interpolation. The Sobel operator then computes horizontal and vertical gradients G_x and G_y, from which gradient angles are derived. For any pixel (i, j) in the interpolated image, we consider a 5×5 neighborhood $N_{i,j}$. For each neighbor (x, y), the angle $\beta_{x,y}$ between its gradient direction and the central pixel' s horizontal gradient direction is computed. Neighbors with $\beta_{x,y} < 45°$ are selected, and their horizontal gradient mean $G'_{i,j}$ replaces the original horizontal gradient. The vertical gradient $G'_{i,j}$ is similarly computed.

2.2.3 Gradient-Adaptive Spatial Distance Adjustment While local asymmetry-based adjustment considers edge data distribution, it doesn' t fully exploit gradient information, which is crucial for edge structure preservation. Since pixel values vary most significantly along gradient directions, the projection distance of neighboring pixels onto the interpolation point' s gradient direction indicates spatial correlation strength.

For horizontal gradient direction adjustment, we construct a plane l at the interpolation point' s vertical coordinate y_j and project neighboring pixel values onto it. Using the initial pixel value $f(i,j)$ obtained via bicubic interpolation and the horizontal gradient angle $\varphi(i)$ from Section 2.2.2, we create a line function on plane l:

$$f(x) = k \cdot x + b, \quad k = \tan(\varphi(i)), \quad b = f(i,j) - k \cdot i$$

The projection distance d_x of neighbor (x, y) onto the horizontal gradient direction is:

$$d_x = \sqrt{d_1^2 - d_2^2}$$

where d_1 is the Euclidean distance to the line and d_2 is the distance along the line.

The weight function based on projection distance is:

$$w_x = \frac{\exp(-\alpha \cdot |d_x|)}{\sum_{m,n \in N_{i,j}} \exp(-\alpha \cdot |d_{m,n}|)}$$

where α controls exponential decay (typically 0.2).

Using these weights, the horizontal offset distance s'_h is computed as:

$$s'_h = \sum_{(x,y) \in N_{i,j}} w_x \cdot (x - i)$$

Similarly, the vertical offset s'_v is obtained. These one-dimensional corrections are fused into two-dimensional space using the same transformation as in Section 2.2.1, yielding the final gradient-adaptive corrected coordinates.

2.3 Rational Function-Based Image Interpolation

The bivariate rational interpolation model switches between rational and polynomial forms via parameter adjustment. Rational models enable extreme values in small regions, suitable for large pixel value fluctuations, while polynomial models constrain variations, appropriate for smooth regions.

Considering structural differences, edge regions use the rational model and non-edge regions use the polynomial model. As shown in [Figure 7: see original paper], for a region bounded by four known pixels (black points) with four interpolation points (red points), the interpolation surface over $W_{i,j}$ is constructed using 12 surrounding pixels. Any point in $W_{i,j}$ is evaluated using the surface function. For edge regions, corrected coordinates are substituted into the rational function; for non-edge regions, setting $\alpha_{i,j} = \beta_{i,j} = 1$ converts the model to polynomial form for direct evaluation.

3. Experimental Results Analysis

Eight test images shown in [Figure 8: see original paper] were downsampled using row/column decimation. The proposed algorithm was compared against bicubic [?], NEDI [?], DFDF [?], RSAI [?], NARM [?], Lee' s [?], GORI [?], and NSCTRAI [?] algorithms in terms of objective metrics, subjective quality, and time complexity.

presents PSNR and SSIM comparisons. PSNR measures pixel-wise differences, while SSIM [?] evaluates structural, luminance, and contrast similarities. The proposed algorithm achieves the highest average PSNR and SSIM, demonstrating superior objective performance.

[Figure 9: see original paper] and [Figure 10: see original paper] compare texture detail reconstruction. For Barbara' s scarf with multi-directional textures, NEDI, RASI, NARM, and Lee' s exhibit texture breakage; bicubic, DFDF, and NSCTRAI produce blurred textures with visible jaggies in bicubic; GORI performs well on textures but blurs some edges. Our algorithm preserves textures without breakage or blurring. For Fence reconstruction, bicubic, DFDF, and NARM show boundary blurring; NEDI, RSAI, Lee' s, and NSCTRAI produce discontinuous stripes with boundary distortion; GORI preserves clear textures but blurs distant regions and boundaries. Our algorithm effectively retains texture features while maintaining clean boundaries without distortion or jaggies.

[Figure 11: see original paper] compares edge reconstruction. For Barbara' s facial contours, bicubic, NARM, and FDI show obvious jaggies; GORI exhibits twisted edge textures; NEDI and our algorithm produce smoother edges.

Time complexity is crucial for practical applications. shows runtime comparisons. Bicubic is fastest, followed by our algorithm and GORI, while NARM is slowest.

In summary, bicubic is simple and fast but yields poor reconstruction with blurred edges and textures. NEDI and Lee' s preserve edges well but distort textures. RASI maintains textures but produces edge jaggies. DFDF generates blurred results. GORI preserves textures but suffers from edge blurring and jaggies. Our algorithm produces smooth edges and clear textures with superior subjective quality, improving PSNR by 0.48–2.17 dB and SSIM by 0.004–0.028 on average, while maintaining low computational cost.

4. Conclusion

This paper proposes a texture detail and edge structure preserving image interpolation algorithm based on the bivariate rational function model. The algorithm first extracts eight-directional gradient features and uses simple statistical thresholding to segment images into edge and non-edge regions. It then adjusts spatial distances of edge region interpolation points based on local asymmetry and gradient features. Finally, polynomial interpolation is applied to non-edge regions while rational function interpolation with corrected coordinates is used for edge regions. Leveraging both the high texture fidelity of rational functions and edge information from local asymmetry and gradients, the algorithm achieves excellent visual quality and superior objective evaluation metrics.

References

[1] Gonzalez R C, Woods R E. Digital Image Processing [J]. Prentice Hall International, 2008, 28 (4): 484-486.

[2] Keys R G. Cubic convolution interpolation for digital image processing [J]. IEEE Trans on Acoustics Speech & Signal Processing, 1981, 37 (6): 1153-1160.

[3] Hou H S, Andrews H C. Cubic splines for image interpolation and digital filtering [J]. IEEE Trans on Acoustics Speech and Signal Processing, 1979, 26 (6): 508-517.

[4] Li Xin, Orchard M T. New edge-directed interpolation [J]. IEEE Trans on Image Processing, 2001, 10 (10): 1521-1527.

[5] Liu Yifang, Zhang Yunfeng, Guo Qiang, et al. Image interpolation based on weighted and blended rational function [C]// Proc of the 12th Asian Conference on Computer Vision. Berlin: Springer International Publishing, 2014: 78-88.

[6] Liu Yifang. Research on image interpolation theory and technology based on parametric rational function [D]. Shandong: Shandong University of Finance and Economics, 2015.

[7] Liu Yifang, Zhang Yunfeng, Guo Qiang, et al. Image interpolation of rational function guided by human visual perception model [J]. Journal of Xidian University, 2016, 43 (01): 151-156.

[8] Du Hongwei, Zhang Yunfeng, Bao Fangxun, et al. Rational function image interpolation for gradient optimization [J]. Journal of Image and Graphics, 2018, 23 (05): 766-777.

[9] Zhang Yunfeng, Bao Fangxun, Zhang Caiming, et al. Local Shape Control of a Bivariate Rational Interpolating Surface with Mixing Conditions [C]// Proc of the 2011 8th International Symposium on Voronoi Diagrams in Science and Engineering, ISVD 2011. Piscataway, NJ: IEEE Press, 2011: 200-205.

[10] Sun Qinghua, Bao Fangxun, Zhang Yunfeng, et al. A bivariate rational interpolation based on scattered data on parallel lines [J]. Journal of Visual Communication and Image Representation, 2013, 24 (1): 75-80.

[11] Ramponi G. Warped distance for space-variant linear image interpolation [J]. IEEE Trans on Image Processing, 1999, 8 (5): 629-639.

[12] Zhang Lei, Wu Xiaolin. An edge-guided image interpolation algorithm via

directional filtering and data fusion [J]. IEEE Trans on Image Processing, 2006, 15 (8): 2226-2238.
[13] Hung K W, Siu W C. Robust soft-decision interpolation using weighted least squares [J]. IEEE Trans on Image Processing A Publication of the IEEE Signal Processing Society, 2012, 21 (3): 1061-1069.
[14] Dong Weisheng, Zhang Lei, Lukac R, et al. Sparse representation based image interpolation with nonlocal autoregressive modeling [J]. IEEE Trans on Image Processing, 2013, 22 (4): 1382-1394.
[15] Lee S J, Kang M C, Uhm K H, et al. An edge-guided image interpolation method using taylor series approximation [J]. IEEE Trans on Consumer Electronics, 2016, 62 (2): 159-165.
[16] Fan Qinglan, Zhang Yunfeng, Bao Fangxun, et al. An region adaptive image interpolation algorithm based on the NSCT [J]. Journal of Computer Research and Development, 2018, 55 (03): 629-642.
[17] Zhu Jiong, Yan Lijun, Ma Yan. Development and application of improved SSIM image evaluation system [J]. Computer Engineering and Applications, 2019, 55 (12): 215-219.