

# SOFT ADAPTIVE GRADIENT ANGLE INTERPOLATION OF GRAYSCALE IMAGES

*Christine M. Zwart<sup>1</sup> and David H. Frakes<sup>1,2</sup>*

Arizona State University,

<sup>1</sup>School of Biological and Health Systems Engineering,

<sup>2</sup>School of Electrical, Computer, and Energy Engineering,  
Tempe, Arizona 85287, USA

## ABSTRACT

We introduce a new edge-directed interpolator based on locally defined, straight line approximations to image isophotes. The first spatial derivatives of image intensity are used to describe the behavior of pixel-intersecting isophotes in terms of their slopes. Slopes are determined by inverting a tridiagonal matrix and are forced to vary linearly from pixel to pixel. Image resizing is performed using standard, 1D interpolators along the approximated isophotes. The proposed method can accommodate arbitrary scaling factors, provides state-of-the-art results in terms of PSNR as well as other quantitative quality metrics, and has the advantage of computational complexity that is directly proportional to the number of pixels.

*Index Terms*— interpolation, least squares methods

## 1. INTRODUCTION

Image resolution limits the extent to which zooming enhances clarity, restricts the quality of digital photograph enlargements, and, in the context of medical images, can prevent a correct diagnosis. Single image interpolation (zooming, upsampling, or resizing) can artificially increase image resolution for viewing or printing, but is generally limited in terms of enhancing image clarity or revealing higher frequency content. A wide variety of non-linear interpolation methods have been designed to improve upon linear interpolation results by avoiding or correcting for common artifacts that stem from aliasing and artificial grid-alignment.

Jagged or blurred edges significantly detract from image appearance and many interpolation algorithms have been developed to avoid such artifacts. Adaptive-kernel methods adjust the contributions of pixels in the region according to some definition of local structure. These adaptations are designed to interpolate primarily along the edges minimizing the contributions of cross-edge neighbors. [1] uses radial basis functions and selects from a collection of stencils based on enclosed curvature. Stencils or blocks of wavelet coefficients are also used in [2] to identify interpolation directions with the greatest regularity.

---

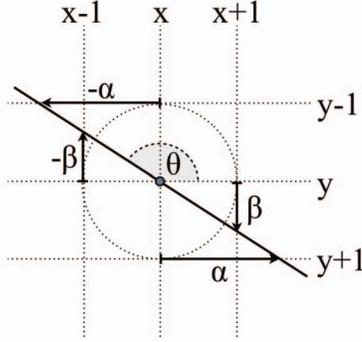
C.Zwart is funded by the NSF Graduate Research Fellowship Program.

In [3], the low-resolution covariance is used to estimate high-resolution covariances and define the best weighting scheme based on Wiener filtering theory. [4] extends this approach to use soft-decisions to define interpolation parameters for groups of pixels simultaneously. First-order derivatives have also been used to identify edge direction explicitly ([5, 6]). In [7], first order derivatives are used as a preliminary thresholding value and covariances are used to determine more closely matched pairs of neighboring pixels.

The edge-adaptive-kernel interpolators proposed in previous work are not always equipped to handle fine detail, as some require rather large support regions. Furthermore, the computational burden of these methods can be extensive. As a result, potential orientations are commonly quantized and sufficiently smooth regions are often parsed out and addressed with simpler methods. Additionally, corners and regions of high curvature are poorly accommodated or must be addressed as special cases. The same is true of ridges or very thin edges. Many previous methods are also limited to scaling factors of two.

We present a new method, soft-adaptive gradient angle (SAGA) interpolation, that uses first-order derivatives to approximate isophotes. This new approach adapts to interpolate along vectors that represent the local isophotes (lines of constant intensity). The vectors are determined based on the isophote slope while ensuring that the direction of interpolation varies linearly from pixel to pixel. The SAGA method addresses smooth, high-curvature, and ridge regions, can be implemented for arbitrary (non-integer) scaling factors, and is separable into row-wise and column-wise interpolations.

Based on quantitative image quality metrics, SAGA interpolation performs comparably to state-of-the-art methods such as Soft-Decision Adaptive Interpolation (SAI) [4] and Sparse Mixing Estimators (SME) interpolation [2], and performs better than common benchmarking algorithms including bicubic interpolation, improved New Edge Directed Interpolation (iNEDI) [3], and Iterative Curvature-Based Interpolation (ICBI) [7]. The primary advantage of SAGA interpolation is uniquely low computational complexity (it scales directly with the number of pixels).



**Fig. 1.** For a straight line of isointense pixels forming an angle  $\theta$  with the horizontal pixel grid, points on the line where it intersects with the lattice can be written in terms of  $\alpha$  and  $\beta$ .

## 2. VECTOR APPROXIMATIONS OF ISOPHOTES

Image isophotes (lines of constant intensity) are essential to human visual perception. Excessive curvature or breaks in isophotes can make a digital image look unnatural and unappealing. Interpolating along rather than across isophotes can minimize the introduction of such artifacts. Defining what constitutes ‘along’ is the primary aim of edge-directed interpolation algorithms, including SAGA.

The straight isophote shown in Figure 1 that crosses through the image pixel at location  $(x, y)$  has an intensity that can be expressed as  $I(x, y)$ . Using parameters  $\alpha$  and  $\beta$ , points along the isophote have the same intensity:

$$I(x - \alpha, y - 1) = I(x, y) = I(x + \alpha, y + 1), \quad (1)$$

$$I(x - 1, y - \beta) = I(x, y) = I(x + 1, y + \beta). \quad (2)$$

The Taylor series expansion can be used to separate the parameters  $\alpha$  and  $\beta$  from the intensity terms using the  $x$  and  $y$  partial derivatives. For simplicity, we will focus only on the determination of the parameter  $\alpha$ . Equivalent expansions and expressions for defining  $\beta$  are a straight-forward extension. Expressions derived from the expansion of Equation 1 are:

$$0 \approx I(x, y) - I(x, y - 1) + \alpha \frac{\partial I(x, y - 1)}{\partial x}, \quad (3)$$

$$0 \approx \alpha \frac{\partial I(x, y)}{\partial x} + \frac{\partial I(x, y)}{\partial y}, \text{ and} \quad (4)$$

$$0 \approx I(x, y + 1) - I(x, y) + \alpha \frac{\partial I(x, y + 1)}{\partial x}. \quad (5)$$

Based on these expansions we define the following consolidated expression for  $\alpha$  moving forward:

$$0 = \alpha \bar{I}_x(x, y) + I_y(x, y). \quad (6)$$

Here,  $I_y$  is the partial  $y$  derivative computed with a centered, finite difference and  $\bar{I}_x$  is the  $x$  partial derivatives computed with a  $3 \times 3$  derivative kernel (we use the modified Sobel formulation). Use of the  $3 \times 3$  kernels is motivated

by the three row span of Equations 3-5. It is noteworthy that an  $\alpha$  that satisfies Equations 6 and a similarly defined  $\beta$  do not necessarily satisfy Equations 1 and 2. Furthermore,  $\beta$  is not necessarily equal to  $\alpha^{-1}$ . Rather, Equation 6 provides a separate expression for estimating vectors  $[\pm\alpha, \pm 1]$  that approximate the local isophote based on the gradient defined slope. Our method comprises determining and interpolating along these vectors and those estimated by the parameter  $\beta$ .

## 3. SOFT ADAPTIVE GRADIENT ANGLE INTERPOLATION

Effective interpolation based on the gradient angle or other first-order derivative metrics has been demonstrated previously [5, 7]; however, a number of scenarios present problems for such techniques. For example, in areas of high curvature, the neglected second order terms can become significant and contribute artifacts. Furthermore, in smooth or ridge regions where the gradient magnitude is small, computations are sensitive to noise and are likely to be misestimated. As a result, many approaches use alternative methods to interpolate pixels that violate thresholds for curvature or gradient magnitude.

To avoid these pitfalls, we ‘soften’ Equations 6 by introducing a regional dependency. The  $\alpha$  describing the isophote at position  $(x, y)$  is allowed to influence the  $\alpha$  assigned to  $(x - 1, y)$  and  $(x + 1, y)$  and visa versa. Details of the softened optimization framework as well as the specifics of our algorithm implementation follow.

For a given line of  $N$  pixels, we assert that there exists a vector of displacement parameters ( $\alpha$  or  $\beta$ ) that describes the offset in the indexing direction ( $x$  for rows and  $y$  for columns) to the best intensity-matched pixels in the adjacent lines. We introduce a stiffness parameter  $k$  such that elements in  $\alpha$  or  $\beta$  are linearly related over segments of  $k$  pixels.

For  $\theta_1$  and  $\theta_2$  linear basis functions:

$$\theta_1(i) = \frac{k - i}{k} \quad \text{and} \quad \theta_2(i) = \frac{i}{k}, \quad (7)$$

the full displacement vectors can be computed from every  $k^{\text{th}}$  displacement. For example:

$$\alpha(x + i, y) = [\theta_1(i) \quad \theta_2(i)] \begin{bmatrix} \alpha(x, y) \\ \alpha(x + k, y) \end{bmatrix}, \quad (8)$$

where pixel locations  $(x, y)$  and  $(x + k, y)$  are considered ‘nodes’ and the inter-nodal pixel displacements are interpolated.

For simplicity, we’ll continue by describing the algorithm in 1D (indexing only in  $x$ ) as it is used to determine the vector of horizontal displacements  $\alpha$  for a single row. The determination of  $\alpha$  for each row is an independent and identical process that can be run in parallel. The algorithm for determining the vector  $\beta$  for a column is a straightforward change of variables.

### 3.1. Determination of Optimal Displacements

Using Equation 6 and the softened framework, we define the matching error for the horizontal displacements associated with pixels in a given row as:

$$E(\alpha_L) = \|\text{diag}(\bar{I}_x)\Theta\alpha_L + I_y\|^2, \quad (9)$$

where  $\Theta$  applies the bilinear interpolation basis functions such that,

$$\alpha = \Theta\alpha_L, \quad (10)$$

$\alpha$  is the vector of horizontal displacements,  $L$  is the set of nodes  $[1, 1+k, 1+2k, \dots, N]$ , and  $\alpha_L = \alpha(x \in L)$  is the set of nodal displacements.  $\bar{I}_x$  and  $I_y$  are column vectors containing the derivatives for each pixel in the row and  $\text{diag}(\bar{I}_x)$  is a matrix with the elements of  $\bar{I}_x$  along the diagonal.

### 3.2. Selection of Nodes and Matrix Condition

The matching error associated with the least squares minimization of Equation 9 is determined by the density ( $1/k$ ) and the placement of the nodes. Considering the case where  $k = 1$  and  $\Theta$  reduces to the identity matrix:

$$E(\alpha_L) = \|\text{diag}(\bar{I}_x)\alpha_L + I_y\|^2, \quad (11)$$

the solution reduces to a strict, gradient angle-based assessment of the image isophotes ( $\alpha(x, y) = -I_y(x, y)/I_x(x, y)$ ) and edges are treated as perpendicular to  $\angle \nabla I(x, y) = I_y(x, y)/I_x(x, y)$ . For images that are piecewise stationary, the matrix  $\text{diag}(\bar{I}_x)$  will likely be singular or poorly conditioned (the singular values are such that  $\sigma_1 = \max(|\bar{I}_x|)$  and  $\sigma_N = \min(|\bar{I}_x|)$ ). Ideally, the nodes in  $L$  should serve to partition the line of pixels into stationary groupings. For computational efficiency, the precomputed derivatives ( $\bar{I}_x$ ) can be used as a simple indicator of change within a finite-size search window.

### 3.3. Solving for Displacements and Complexity

We define the combined coefficient matrix:  $J = \text{diag}(\bar{I}_x)\Theta$ . The normal equation describing the least-squares solution to Equation 9 can be written as:

$$J^T J \alpha_L = J^T (-I_y). \quad (12)$$

$J$  has  $2N - L$  nonzero entries where  $N$  is the number of pixels in the line and  $L$  is the number of nodes (nominally equal to  $N/k$ ). Computing the products  $J^T J$  takes  $(2(2N - L) - L)$  multiplications. Defining  $J^T (-I_y)$  requires  $(2N - L)$  multiplications. Setting up Equation 12 has complexity  $O(N)$ . In solving the normal equation, we observe that  $J^T J$  is a tridiagonal,  $L \times L$  matrix. The solution can be directly obtained using the tridiagonal matrix algorithm ( $O(L)$ ).

For each line of pixels, the full displacement vector  $\alpha$  is calculated from the nodal displacements in  $\alpha_L$  using the interpolation matrix  $\Theta$  (as in Equation 10) and requires  $(2N -$

$L)$  multiplications. The overall order of complexity for computing the displacement vector  $\alpha$  for a line of pixels of length  $N$  is  $O(N)$  with the matrix multiplication  $J^T J$  being the most intense step. For a full image, two displacements are computed for each pixel and the complexity scales directly with the number of pixels ( $O(MN)$  for an  $M \times N$  matrix).

### 3.4. Interpolation Based on Displacements

The complete set of displacements (one horizontal and one vertical for each pixel) can be used to define four ‘matched’ locations in the original resolution image. For the pixel at location  $(x, y)$  these locations are  $(x \pm \alpha(x, y), y \pm 1)$  and  $(x \pm 1, y \pm \beta(x, y))$ . We construct four, separately interpolated images by inserting data along the vectors connecting the original pixels to the matched locations as described in [6]. The interpolated images are then combined with weighting at each pixel inversely proportional to their distance to the mean estimate.

## 4. RESULTS

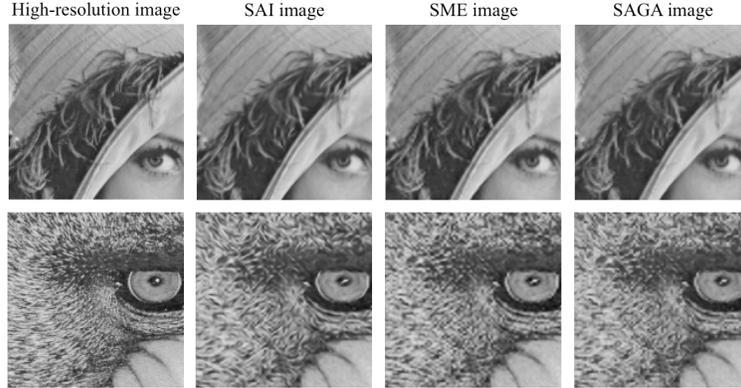
The results of the algorithm will be compared with alternative techniques for two-times expansion of a variety of standard test images. The complexity of similarly accurate methods will be compared to that of SAGA.

### 4.1. Comparison to Other Methods: Accuracy

The results of a two-times interpolation of several standard test images were compared. In addition to the proposed SAGA method, bicubic interpolation, iNEDI [3], ICBI [7], SAI [4], and SME interpolation [2] were used. All results were evaluated in terms of peak signal to noise ratio (PSNR) as well as visual signal to noise ratio (VSNR) (both in decibels) and the Universal Quality Index (UQI). All evaluations were performed using Image and Video Quality Evaluation Software (<http://ivulab.asu.edu/Quality/IVQUEST>) [8]. The values for each metric are reported in Table I with the best performing result in bold. Average improvements relative to bicubic interpolation are also reported. Overall, SAGA interpolation results are as accurate as those obtained with state-of-the-art methods SAI and SME and better than bicubic interpolation, iNEDI, and ICBI. Figure 2 provides visual comparisons of the methods for detailed regions of the Lena and Mandrill images.

### 4.2. Comparison to Other Methods: Computation

In terms of accuracy, SAGA is comparable to SME and SAI. One advantage of SAGA is its low complexity. For  $Z = MN$ , the number of pixels, the overall order of the SAGA approach in computing the isophote-slope-characterizing parameters is  $O(Z)$ . In contrast, SME uses a more complex  $O(Z \log(Z))$  approach to describe edge orientations through



**Fig. 2.** Comparison of interpolation results for the tested methods. Regions shown are from the Lena and Mandrill test images. The “High-resolution” image was two-times down-sampled and enlarged using SAI, SME, and SAGA.

**Table 1.** Quantitative evaluation of interpolators.

	Metric	Bicubic	INEDI	ICBI	SAI	SME	SAGA
Lena	PSNR	34.029	33.973	34.081	<b>34.773</b>	34.628	34.460
	VSNR	26.315	27.082	26.591	<b>27.734</b>	27.408	27.267
	UOI	0.780	0.778	0.773	0.783	0.784	<b>0.785</b>
Monarch	PSNR	31.970	33.004	32.619	<b>33.130</b>	32.739	32.686
	VSNR	24.101	<b>26.685</b>	25.559	26.644	25.588	25.727
	UOI	0.801	0.807	0.796	0.807	0.804	<b>0.810</b>
Peppers	PSNR	32.810	33.191	32.925	<b>33.533</b>	33.432	33.472
	VSNR	26.126	27.077	26.297	<b>27.502</b>	27.387	27.428
	UOI	0.704	0.708	0.696	0.711	0.708	<b>0.713</b>
Houses	PSNR	23.131	22.787	22.438	23.339	23.430	<b>23.439</b>
	VSNR	17.390	17.108	16.520	<b>17.907</b>	17.819	17.903
	UOI	0.754	0.743	0.736	0.761	<b>0.767</b>	0.763
Goldhill	PSNR	31.636	30.971	31.191	31.688	31.701	<b>31.836</b>
	VSNR	26.548	26.125	25.836	26.700	26.638	<b>26.904</b>
	UOI	0.785	0.778	0.774	0.781	0.780	<b>0.789</b>
Mandrill	PSNR	22.909	22.558	22.121	23.167	23.105	<b>23.227</b>
	VSNR	12.523	12.100	11.185	12.821	12.724	<b>13.123</b>
	UOI	0.710	0.707	0.692	<b>0.719</b>	<b>0.719</b>	0.714
Boats	PSNR	32.168	31.448	31.493	32.191	<b>32.595</b>	32.442
	VSNR	21.699	21.135	21.069	22.127	22.244	<b>22.322</b>
	UOI	0.713	0.713	0.705	<b>0.721</b>	0.720	0.720
Mosaic	PSNR	32.291	<b>33.412</b>	32.783	32.690	33.283	33.259
	VSNR	28.301	<b>30.018</b>	29.217	29.387	29.855	29.863
	UOI	0.902	<b>0.967</b>	0.882	0.857	0.786	0.888
Average Improvement over Bicubic	PSNR	*	0.050	-0.161	0.446	0.496	0.485
	VSNR	*	0.541	-0.091	0.977	0.833	0.942
	UOI	*	<b>0.007</b>	-0.012	-0.001	-0.010	0.004

a block matching approach [2]. SAI solves three least-squares problems to directly define an image with two-times the original resolution. The approach is iterative and only pixels in regions with high local variances are directionally interpolated. Based on  $O(Z)$  complexity of SAGA, the  $O(Z \log(Z))$  complexity of SME, and the iterative nature of SAI, SAGA is, generally speaking, the least complex of the three techniques.

## 5. CONCLUSIONS

We have introduced a new, edge-directed interpolator called Soft Adaptive Gradient Angle (SAGA) interpolation. Based on quantitative image quality metrics, SAGA performs comparably to other state-of-the-art methods, and is better in many cases. Furthermore, the algorithm operates with uniquely low computational cost. In addition, the SAGA

algorithm is well suited for parallelization as image lines can be processed independently.

## 6. REFERENCES

- [1] Y. J. Lee and J. Yoon, “Nonlinear Image Upsampling Method Based on Radial Basis Function Interpolation,” *IEEE Trans. on Image Process.*, vol. 19, no. 10, pp. 2682–2692, Oct 2010.
- [2] S. Mallat and G. Yu, “Super-resolution with sparse mixing estimators,” *IEEE Trans. on Image Process.*, vol. 19, no. 11, pp. 2889–2900, 2010.
- [3] N. Asuni and A. Giachetti, “Accuracy improvements and artifacts removal in edge based image interpolation,” in *Proc. 3rd Int. Conf. Computer Vision Theory and Applications*, 2008.
- [4] X. Zhang and X. Wu, “Image interpolation by adaptive 2-d autoregressive modeling and soft-decision estimation,” *IEEE Trans. Image Process.*, vol. 17, no. 6, pp. 887–896, June 2008.
- [5] Q. Wang and R. K. Ward, “A new orientation-adaptive interpolation method,” *IEEE Trans. on Image Process.*, vol. 16, no. 4, pp. 889–900, 2007.
- [6] C. M. Zwart and D. H. Frakes, “Biaxial control grid interpolation: Reducing isophote preservation to optical flow,” in *Proc. IEEE Digital Signal Process. Workshop*, 2011.
- [7] A. Giachetti and N. Asuni, “Fast artifacts-free image interpolation,” in *Proc. of the British Machine Vision Conf.*, Leeds, September 2008, pp. 123–132.
- [8] A. Murthy and L. Karam, “A matlab-based framework for image and video quality evaluation,” in *Quality of Multimedia Experience (QoMEX)*, June 2010, pp. 242–247.