

Convolutional Edge Diffusion for Fast Contrast-guided Image Interpolation

Wei Ye, *Student Member, IEEE*, and Kai-Kuang Ma, *Fellow, IEEE*

Abstract—A recently introduced image interpolation method, called the *contrast-guided interpolation* (CGI), has shown superior performance on producing high-quality interpolated image. However, its *iterative edge diffusion* (IED) process for diffusing continuous-valued *directional variation* (DV) fields inevitably incurs high computational complexity due to its iterative optimization process. The key objective of this work lies in how to greatly reduce the computation of this diffusion process while maintaining CGI's superior performance on its interpolated image. The novelty of our work started with a critical observation as follows. Since each diffused DV field needs to be thresholded for generating a binary *contrast-guided decision map* (CDM) in the subsequent step, such binarization operation will definitely destroy the fidelity that was preserved previously through the data term of the IED's energy functional. Therefore, the data term is lifted in our approach to yield a new energy functional. It turns out that the diffusion equation derived from this simplified functional is, in fact, the well-known *heat equation*, from which a highly attractive property of the heat equation can be exploited for conducting diffusion. That is, given a desired amount of diffusion to yield, it can be realized by simply convolving the DV field with a Gaussian kernel *once*, rather than gradually updating the DV field through iterations. Note that the variance of the Gaussian kernel corresponds to the amount of diffusion desired. As a result, the total computation time is significantly reduced. Extensive simulation results have shown that the proposed CED can generate nearly identical CDMs as those produced by the IED, while only requiring about 1/10 of its computation time. By replacing the IED with the proposed CED in the CGI framework, the total run time of our *fast CGI* is only 1/4 of the original CGI's on average.

Index Terms—Contrast-guided image interpolation, edge diffusion, contrast-guided decision map, heat equation, convolution, Gaussian filtering.

I. INTRODUCTION

Conventional image interpolation methods, such as bilinear [1], bicubic spline [2], and bicubic convolution [3], are computationally efficient and easy to implement, but they often suffer from various forms of artifacts, such as blurring, ringing, jaggy edges, etc. To generate high-quality interpolated image, more sophisticated interpolation methods have been proposed; among them, the *edge-guided interpolation* (EGI) (e.g., [?], [4]–[8]) has been considered as a fairly attractive approach. In this approach, edge information will be extracted from the input image first, followed by utilizing this information to guide the interpolation process to conduct *directional* interpolation.

Copyright (c) 2015 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

Wei Ye and Kai-Kuang Ma are with School of Electrical and Electronic Engineering, Nanyang Technological University, 639798, Singapore (E-mails: ye0003ei@e.ntu.edu.sg; ekmma@ntu.edu.sg).

Recently, a novel interpolation method, called the *contrast-guided interpolation* (CGI) [9], has been proposed and demonstrated state-of-the-art performance on the interpolated image quality. The CGI is the first algorithm that innovatively incorporates the local *contrast* information into the interpolation process. The novelty to fulfill this goal lies on the re-evaluation and possible re-classification of some edge-nearby non-edge pixels as 'edge' ones to conduct directional interpolation for their associated to-be-interpolated pixels, respectively. This re-classification is realized by applying an edge diffusion process, called the *iterative edge diffusion* (IED), in the CGI to diffuse the variations on each *directional variation* (DV) field computed from the input image under a specific direction. Each diffused DV field is *continuous*-valued and will be thresholded to generate a *contrast-guided decision map* (CDM), which is *binary*-valued and will be used to conduct the directional interpolation along various directions, respectively. There are four such DV fields that are subject to be diffused and then thresholded in the CGI method [9].

Despite its superior interpolation performance, the CGI's IED process incurs high computational complexity and algorithmic complexity, due to its iterative updating process on generating four diffused DV fields. This inevitably hinders the use of CGI in those applications that require real-time performance and low power consumption. To address this critical issue, a *fast* edge diffusion scheme, called the *convolutional edge diffusion* (CED), is proposed in this letter, which greatly reduces the computation time while maintaining superior performance on image quality. Our work was motivated by a critical observation as follows. Since each diffused DV field (*continuous*-valued) will be thresholded to generate a *binary* CDM in the subsequent step [9], such binarization operation will inevitably destroy the fidelity that was preserved in the previous stage through the data term of the IED's energy functional. Furthermore, due to the presence of this data term, an iterative optimization process is performed in the IED, which is computationally expensive. Therefore, this data term is lifted in our approach to yield a simplified energy functional to begin with, and the derived diffusion equation bears the well-known heat equation form with a very significant property to be utilized on the development of the proposed CED.

The rest of this letter is organized as follows. Section II provides the essential background of the CGI's IED scheme. Section III describes the proposed CED process. By replacing the CGI's IED with our developed CED, the performance of the resulted *fast CGI* algorithm is then evaluated against that of the original CGI and several other state-of-the-art methods as presented in Section IV. Conclusions are drawn in Section V.

II. CONTRAST-GUIDED IMAGE INTERPOLATION

Given a low-resolution image I_l , four DV fields U_θ need to be generated, one for each direction θ under consideration. In [9], the four considered directions are 45° , 135° , 0° , and 90° . Each U_θ is computed by $U_\theta = |I_l * d_\theta|$, where $*$ denotes the convolution and d_θ is a differentiation filter for extracting the edges of I_l along the direction that is *perpendicular* to θ . The generated four *continuous*-valued DV fields U_θ are then diffused individually by minimizing the following energy functional:

$$E(u_\theta) = \iint \left\{ U_\theta^2 (u_\theta - U_\theta)^2 + \lambda \left[(u_\theta)_x^2 + (u_\theta)_y^2 \right] \right\} dx dy, \quad (1)$$

where u_θ denotes the *diffused* continuous-valued DV field. (The argument (x, y) of $U_\theta(x, y)$ and $u_\theta(x, y)$ is omitted for the ease of presentation.) Symbols $(u_\theta)_x$ and $(u_\theta)_y$ represent the partial derivatives of u_θ along the horizontal (i.e., x -axis) and the vertical (i.e., y -axis) directions, respectively.

The first term of the integrand in (1) is called the *data term*, which is the fidelity constraint imposed on the diffusion process, reflecting how close of the diffused u_θ when compared it with the original or undiffused U_θ . The second term is called the *smoothness term*, which measures the degree of *unsmoothness* (or *variation*) incurred in u_θ . The constant λ provides a trade-off between these two terms, and $\lambda = 0.2$ was empirically determined in [9]. Based on the calculus of variations [10], the task of finding the optimal u_θ that minimizes the functional $E(u_\theta)$ in (1) is boiled down to search a *steady-state solution* for the following derived diffusion equation [9]:

$$\partial u_\theta(t) / \partial t = \lambda \nabla^2 u_\theta(t) - U_\theta^2 (u_\theta(t) - U_\theta), \quad (2)$$

where t is a time variable and ∇^2 denotes the Laplacian operator. Based on (2), u_θ can be iteratively updated through

$$u_\theta(t + \Delta t) = u_\theta(t) + \partial u_\theta(t) / \partial t, \quad (3)$$

starting from $u_\theta(0) = U_\theta$. Note that the second term of (3) on the right-hand side can be found in (2). The steady-state u_θ is considered reached when there is no further noticeable change incurred between $u_\theta(t + \Delta t)$ and $u_\theta(t)$ in (3), or when (2) is sufficiently close to zero.

After diffusion, the obtained four *continuous*-valued diffused DV fields will be individually binarized via a simple thresholding process to generate four *binary* CDMs, respectively. These CDMs will be used to guide the follow-up interpolation process for determining whether a pixel under interpolation should be interpolated via directional interpolation or not, and if so, it will be conducted along the direction as indicated by the current CDM. (Refer to [9] for more details.)

III. PROPOSED CONVOLUTIONAL EDGE DIFFUSION (CED)

Despite the superior interpolation performance achieved by the CGI, its iterative edge diffusion process (IED) is computationally expensive as it requires an iterative updating process to generate four diffused DV fields. To reduce the computational cost, a *fast* edge diffusion approach, called *convolutional edge diffusion* (CED), is developed in this letter.

A. Proposed CED Algorithm

As discussed previously in Section II, the data term of (1), $U_\theta^2 (u_\theta - U_\theta)^2$, essentially imposes a fidelity constraint for the iterative optimization process to minimize $E(u_\theta)$. This term has the effect of enforcing each diffused DV field u_θ to be as close to its corresponding initial DV field U_θ as possible. As a result, this data term allows the IED scheme to preserve the prominent edge information contained in the original DV field U_θ . However, we have observed that each diffused DV field will be binarized in the subsequent step to generate its corresponding binary CDM in the CGI framework. This could make such carefully designed, and computationally expensive, IED scheme with the above-described fidelity constraint becoming ‘over-sophisticated’, since the binarization operation is a very crude process. Based on this observation, a new energy functional is proposed in our approach by simply dropping this data term as

$$E_s(u_\theta) = \iint \left[(u_\theta)_x^2 + (u_\theta)_y^2 \right] dx dy. \quad (4)$$

By following the same derivations as presented in [9], a new diffusion equation can be easily arrived at

$$\partial u_\theta(t) / \partial t = \nabla^2 u_\theta(t). \quad (5)$$

Following the same development and arguments as that of the IED, the steady-state solution of (5) will minimize the new energy functional in (4), and this can be obtained by iteratively updating $u_\theta(t)$ based on (3) until it converges, where the derivative $\partial u_\theta(t) / \partial t$ in (3) can be found in (5).

Even without performing the above-described iterative optimization, in fact one can easily judge from the objective functional in (4) and see that $u_\theta = c$, where c is a constant and corresponds to a completely uniform field, is the optimal solution on minimizing $E_s(u_\theta)$. This is clearly undesirable for our image interpolation task because this means that all the edge information, which is originally presented on U_θ , will be completely ‘flatten’ or destroyed. (This means the generated binary CDM will be all zeros, leading to no guidance to the follow-up interpolation process.) To avoid such issue, a stopping criterion is imposed in our approach to terminate the iteration process before it converges. Obviously, such early termination (before converging) will lead to a non-optimal $u_\theta(t)$ on minimizing (4). However, this sacrifice on optimality causes no concern on the generation of CDMs for our image interpolation task because the edge information on the original DV fields, which is crucial for guiding the follow-up interpolation process, is still largely preserved in the diffused DV fields.

Interestingly, it turns out that the derived (5) bears the form of the well-known *heat equation*, which was originally developed to model the conduction of heat in an isolated and uniform material. Furthermore, there is a very significant and useful property of the heat equation that can be exploited to greatly benefit our DV-field diffusion task, particularly on the aspect of computational efficiency. That is, given a desired amount of diffusion to yield, it can be realized by simply convolving the *initial* DV field U_θ with a Gaussian kernel

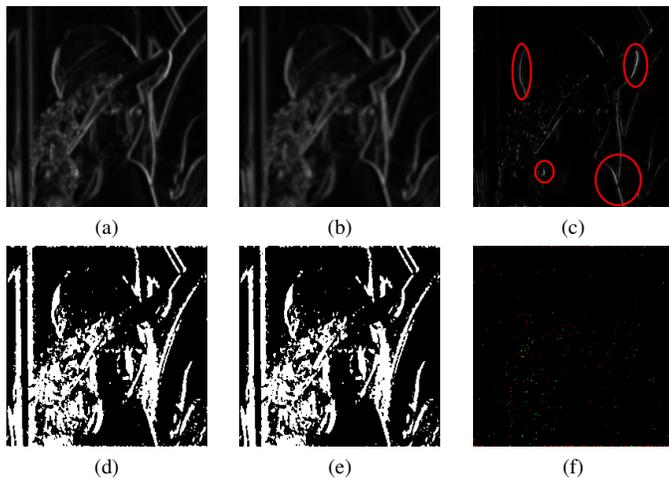


Fig. 1. A comparison of the diffused DV fields (*continuous-valued*) and the final CDMs (*binary-valued*), generated by the CGI's IED and our proposed CED, respectively, using the test image "Lena": (a) u_0 by IED; (b) u_0 by CED; (c) differences between (a) and (b); (d) CDM_{90} by IED; (e) CDM_{90} by CED; and (f) differences between (d) and (e).

once, rather than gradually updating the DV field through iterations [11]. In other words, the resultant $u_\theta(t)$ computed at each time t of the above-described iterative updating process can be directly obtained by convolving the initial DV field U_θ with a continuous 2-D isotropic Gaussian kernel $G(\mu, \sigma^2)$, where $\mu = 0$ and $\sigma = \sqrt{2t}$; that is,

$$u_\theta(t) = G_{(0,2t)} * U_\theta. \quad (6)$$

Therefore, one can determine at what time t the diffusion should be stopped by setting the value of the parameter σ of the Gaussian kernel accordingly, and this proposed fast diffusion method is called *convolutional edge diffusion* (CED).

Lastly, in order to conduct diffusion on the 2-D image lattice, an $N \times N$ discrete Gaussian kernel is used in our approach to approximate the continuous Gaussian kernel. In this case, the kernel width (i.e., filter length) N also becomes a crucial parameter that needs to be properly determined, besides σ . In Section IV, the selection of both N and σ will be discussed in detail.

B. Comparison of the proposed CED and the CGI's IED

Our proposed CED scheme is clearly advantageous on both computational efficiency and algorithmic simplicity, while it is able to generate nearly identical CDMs as those yielded by the computationally expensive IED method. To justify this claim, a set of diffused DV fields (*continuous-valued*) and their resulted CDMs (*binary-valued*), experimented on the test image "Lena", are demonstrated in Fig. 1. Note that since our proposed CED scheme does not have any fidelity constraint being imposed on the diffusion process, it is expected that less edge information will be preserved on its diffused DV field. Indeed, this is the case and can be observed in Fig. 1 (b). To compare it with the diffused DV field yielded by the IED as shown in Fig. 1 (a), their difference image is presented in Fig. 1 (c). One can see that there are clear differences at

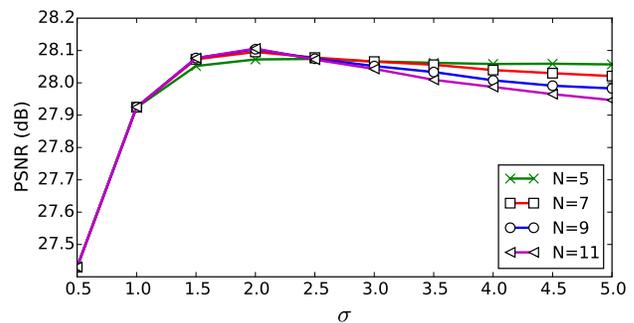


Fig. 2. Some PSNR results of our proposed *fast* CGI algorithm averaged on 8 commonly-used test images (refer to Table I) using different settings of the filter length N and the standard deviation σ of the Gaussian kernel.

those edges as being circled. As expected, such differences are, greatly diminished in the final *binary* CDMs due to the thresholding process that binarizes each continuous-valued diffused DV field into a binary CDM. This can be observed from Fig. 1 (f), which is the difference map yielded between the two binary CDMs as shown in Figs. 1 (d) and (e). In this difference map, the green pixels are those pixels that were classified as 'edge' pixels using our proposed CED but being treated as 'non-edge' pixels using the IED; for the red pixels, the interpretation is reversed. One can see that all the green and red pixels together only occupy a very small portion of the total number of pixels, and they are distributed quite randomly. This means that these two CDMs are nearly identical, practically speaking.

Such conclusion can be drawn for all 500 test images from the BSDS500 dataset [12] that we have experimented, and the total discrepancy (i.e., the total proportion of such red and green pixels) is only 0.82%. Since the generation of CDMs is the core of the CGI as they will be used to guide the follow-up interpolation process, our approach is able to deliver nearly the same interpolation results as that of the original CGI as expected. To be more specific, the average PSNRs yielded by the original CGI (with IED) and our fast CGI (with CED) over the above-mentioned 500 test images are 29.49 dB and 29.51 dB, respectively, which are extremely close to each other.

IV. EXPERIMENTAL RESULTS

Extensive simulation experiments have been conducted to evaluate the interpolation performance of the proposed *fast* CGI algorithm with incorporation of our developed fast edge diffusion scheme, CED. The obtained simulation results are compared with that of the bicubic interpolation and five state-of-the-art methods. Experiments are performed on 8 commonly-used test images (refer to Table I) with spatial resolutions ranging from 256×256 to 512×768 .

To determine the kernel width (i.e., filter length) N and the standard deviation σ of the Gaussian kernel, a study of the interpolation performance, resulted by these two parameters and measured in peak signal-to-noise ratio (PSNR) in dB, has been investigated and documented in Fig. 2. In this work, $N = 7$ and $\sigma = 2.0$ have been empirically determined and used in all our experiments.

TABLE I

AVERAGE PSNR (IN DB) AND SSIM RESULTS. THE HIGHEST PSNR AND SSIM RESULTS FOR EACH TEST IMAGE ARE HIGHLIGHTED IN BOLDFACE.

Images	Bicubic		LMMSE [7]		SAI [8]		ScSR [13]		NARM [14]		CGI [9]		Proposed	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
<i>Station</i>	24.65	0.8928	25.07	0.9028	25.94	0.9160	25.90	0.9160	26.14	0.9215	26.25	0.9213	26.37	0.9218
<i>Cameraman</i>	25.26	0.8649	25.55	0.8692	25.77	0.8732	25.92	0.8751	25.77	0.8777	25.76	0.8720	25.88	0.8736
<i>Wheel</i>	19.59	0.7723	19.64	0.7686	21.53	0.8415	21.48	0.8396	20.75	0.8308	22.37	0.8619	22.37	0.8601
<i>Plane</i>	32.05	0.9218	32.48	0.9243	32.98	0.9274	33.06	0.9288	33.21	0.9314	32.67	0.9265	32.68	0.9268
<i>Lena</i>	34.24	0.9020	34.27	0.9019	34.91	0.9070	34.70	0.9065	35.27	0.9145	34.67	0.9042	34.66	0.9037
<i>PPT</i>	26.77	0.9499	26.79	0.9540	26.26	0.9531	27.46	0.9600	27.68	0.9659	26.77	0.9542	27.05	0.9559
<i>Butterfly</i>	26.17	0.9507	26.44	0.9531	27.40	0.9621	27.75	0.9631	27.31	0.9634	27.65	0.9635	27.68	0.9635
<i>Motobike</i>	27.27	0.8674	27.38	0.8660	28.33	0.8844	28.18	0.8823	28.39	0.8865	28.05	0.8810	28.06	0.8815
<i>Average</i>	27.00	0.8902	27.20	0.8925	27.89	0.9081	28.06	0.9089	28.07	0.9115	28.02	0.9106	28.09	0.9109

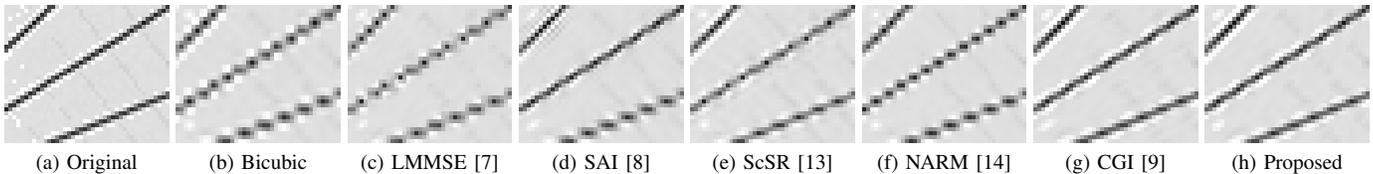


Fig. 3. Visual comparison of the image interpolation results obtained from the test image “Wheel”.

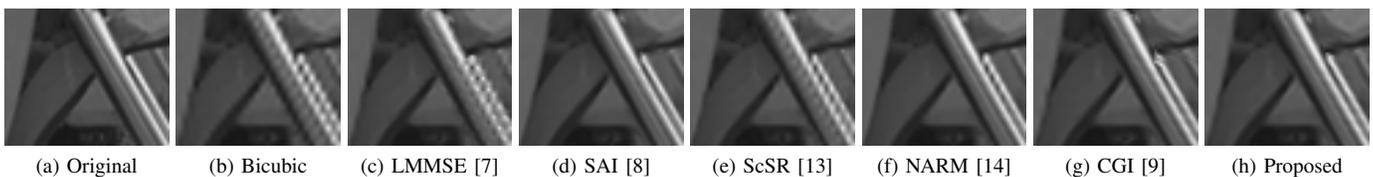


Fig. 4. Visual comparison of the image interpolation results obtained from the test image “Motobike”.

For each test image, its ‘low-resolution’ version can be obtained by down-sampling the original test image by a factor of 2×2 without performing anti-aliasing low-pass filtering. The obtained low-resolution image is then interpolated with an up-scaling factor of 2×2 using the previously-mentioned methods. The PSNR and the *structural similarity* (SSIM) index [15] are measured against the original image, which is viewed as the ground-truth of the interpolated image. The quality of the interpolated images can be thus quantitatively evaluated, and these results are documented in Table I.

For conducting subjective performance evaluation, a small portion of the interpolated images obtained from two test images “Wheel” and “Motobike”, are shown in Fig. 3 and Fig. 4, respectively. It can be observed that although the bicubic interpolation is simple and easy to implement, it generates fairly distinct artifacts. Even for state-of-the-art methods, such as LMMSE, SAI, ScSR, and NARM, they still produce some noticeable artifacts, as can be observed in Figs. 3 (c)-(f) and Figs. 4 (c), (e), (f). In contrast, the original CGI [9] and the proposed fast CGI clearly outperform other methods on the interpolated image quality.

Besides the superior performance, the most appealing merits of the proposed method are its high computational efficiency and algorithm simplicity. The proposed method was implemented in Matlab and compared with other five state-of-the-art methods that have available Matlab source codes. These codes were run on the same machine with a 2.50 GHz CPU and 16 GB memory. The total run time for interpolating all 8 test images by each method is documented in Table II.

TABLE II
TOTAL RUN TIME (IN SECONDS) FOR INTERPOLATING 8 TEST IMAGES

Bicubic	LMMSE [7]	ScSR [13]	NARM [14]	CGI [9]	Proposed
0.25	118.40	2727.04	4317.99	3.10	0.87

One can see that our proposed fast CGI only takes 0.87 seconds, which is drastically faster than the other methods. Comparing with the original CGI method, the total run time of our proposed fast CGI is only about 1/4 of the original CGI’s on average. By considering the total run time of the edge diffusion process alone, our proposed CED only takes 0.23 seconds for processing these 8 test images, which is about 1/10 of the time consumed by the IED (in 2.48 seconds).

V. CONCLUSIONS

In this letter, a fast *convolutional edge diffusion* (CED) algorithm is developed for replacing the CGI’s *iterative edge diffusion* (IED) so that a much more efficient version of the CGI algorithm can be realized with ease. Compared with the IED, the CED only requires about 1/10 of its computation time, since only *one* convolution operation is needed to conduct diffusion in our proposed method. The resulted *fast* CGI algorithm is able to deliver almost the same interpolated image quality as that of the CGI, while only taking about 1/4 of the CGI’s total running time. This makes our *fast* CGI algorithm highly attractive in those applications that demand real-time performance and low power consumption.

REFERENCES

- [1] A. K. Jain, *Fundamentals of digital image processing*. Englewood Cliffs: Prentice-Hall, 1989.
- [2] H. S. Hou and H. Andrews, "Cubic splines for image interpolation and digital filtering," *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. 26, no. 6, pp. 508–517, 1978.
- [3] R. G. Keys, "Cubic convolution interpolation for digital image processing," *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. 29, no. 6, pp. 1153–1160, 1981.
- [4] K. Jensen and D. Anastassiou, "Subpixel edge localization and the interpolation of still images," *IEEE Trans. on Image Processing*, vol. 4, no. 3, pp. 285–295, Mar 1995.
- [5] J. Allebach and P. W. Wong, "Edge-directed interpolation," in *IEEE Int. Conf. on Image Processing*, vol. 3, Sep 1996, pp. 707–710.
- [6] X. Li and M. T. Orchard, "New edge-directed interpolation," *IEEE Trans. on Image Processing*, vol. 10, no. 10, pp. 1521–1527, Oct 2001.
- [7] L. Zhang and X. Wu, "An edge-guided image interpolation algorithm via directional filtering and data fusion," *IEEE Trans. on Image Processing*, vol. 15, no. 8, pp. 2226–2238, Aug 2006.
- [8] X. Zhang and X. Wu, "Image interpolation by adaptive 2-d autoregressive modeling and soft-decision estimation," *IEEE Trans. on Image Processing*, vol. 17, no. 6, pp. 887–896, Jun 2008.
- [9] Z. Wei and K.-K. Ma, "Contrast-guided image interpolation," *IEEE Trans. on Image Processing*, vol. 22, no. 11, pp. 4271–4285, Nov 2013.
- [10] R. Courant and D. Hilbert, *Methods of mathematical physics*. CUP Archive, 1966.
- [11] J. Weickert, *Anisotropic diffusion in image processing*. Teubner Stuttgart, 1998.
- [12] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. 8th Int'l Conf. Computer Vision*, vol. 2, July 2001, pp. 416–423.
- [13] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE Trans. on Image Processing*, vol. 19, no. 11, pp. 2861–2873, Nov 2010.
- [14] W. Dong, L. Zhang, R. Lukac, and G. Shi, "Sparse representation based image interpolation with nonlocal autoregressive modeling," *IEEE Trans. on Image Processing*, vol. 22, no. 4, pp. 1382–1394, Apr 2013.
- [15] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. on Image Processing*, vol. 13, no. 4, pp. 600–612, Apr 2004.