Image Denoising and Zooming under the LMMSE Framework

Lei Zhang [a,1], Xin Li [b], and David Zhang [a]

[a] Dept. of Computing, The Hong Kong Polytechnic University, Hong Kong, China
[b] Lane Dept of CSEE, West Virginia University, WV, USA

Abstract – Most of the existing image interpolation schemes assume that the image to be interpolated is noise free. This assumption is invalid in practice because noise will be inevitably introduced in the image acquisition process. Usually the image is denoised first and is then interpolated. The denoising process, however, may destroy the image edge structures and introduce artifacts. Meanwhile, edge preservation is a critical issue in both image denoising and interpolation. To address these problems, in this paper we propose a directional denoising scheme, which naturally endows a subsequent directional interpolator. The problems of denoising and interpolation are modeled as to estimate the noiseless and missing samples under the same framework of optimal estimation. The local statistics is adaptively calculated to guide the estimation process. For each noisy sample, we compute multiple estimates of it along different directions and then fuse those directional estimates for a more accurate output. The estimation parameters calculated in the denoising processing can be readily used to interpolate the missing samples. Compared with the conventional schemes that perform denoising and interpolation in tandem, the proposed noisy image interpolation method can reduce many noise-caused interpolation artifacts and preserve well the image edge structures.

Keywords — Interpolation, denoising, optimal estimation, data fusion

[1] Corresponding author. Email: cslzhang@comp.polyu.edu.hk. This research is supported by the Hong Kong General Research Fund (PolyU 5330/07E) and the National Science Foundation Council of China under Grant no. 60634030.

1. INTRODUCTION

Interpolation is a fundamental problem in image processing to re-sample the image size [1]. Due to the physical limitation of imaging hardware, image interpolation techniques are often employed to reconstruct a higher resolution (HR) image from its low resolution (LR) counterpart. Image interpolation is widely used in digital photographs, medical imaging and remote sensing, etc, and many interpolation algorithms have been proposed, including the simple but fast linear interpolators [2-5] and those more complex non-linear interpolators [6-15].

Most of the existing interpolation schemes assume that the original image is noise free. This assumption, however, is invalid in practice because noise will be inevitably introduced in the image acquisition process. Usually denoising and interpolation are treated as two different problems and they are performed separately. However, this may not be able to yield satisfying result because the denoising process may destroy the edge structure and introduce artifacts, which can be further amplified in the interpolation stage. With the prevalence of inexpensive and relatively low resolution digital imaging devices (e.g. webcam, camera phone), demands for high-quality image denoising and interpolation algorithms are being increased. Hence new interpolation schemes for noisy images need to be developed for better suppressing the noise-caused artifacts and preserving the edge structures.

Instead of performing denoising and interpolation separately, it is possible to perform interpolation and denoising jointly to reduce the artifacts introduced in the denoising process. Actually, both denoising and interpolation can be viewed as an estimation problem. Denoising is to estimate the original pixels from the noisy measurements, while interpolation is to estimate the missing sample from its local neighbors. The idea of joint denoising and interpolation has been exploited in [16-17]. Hirakawa *et al* [16] proposed an algorithm of joint denoising and color demosaicking (which is a special case of interpolation) to reconstruct the full color image from the color filter array image. The total least square technique was used to estimate the noiseless and missing color components. In [17], Zhang *et al* developed a directional estimation and wavelet based denoising scheme for joint denoising and color demosaicking. A well designed scheme of interpolating noisy images can generate fewer

artifacts and preserve better edge structures compared with schemes that perform denoising interpolation separately.

Edge preservation is a crucial issue in both denoising and interpolation. A general principle is to smooth the noise and interpolate the missing samples along the edge direction, instead of across the edge direction. As a directional image decomposition tool, wavelet transform has been widely used in image denoising [18-23]. Another important class of denoising techniques is partial differential equation (PDE) based anisotropic diffusion (AD). The representative anisotropic filters include the models proposed by Perona and Malik [24] and Alvarez *et al* [25]. The total variation minimization method was proposed by Rudin et al [26] to minimize the energy function related to nonlinear anisotropic diffusion. An iterated total variation refinement scheme was developed in [27], and the iterative regularization method was generalized to nonlinear inverse scale space and applied to wavelet based denoising [28-29]. Recently, regularization and PDE-based techniques for solving inverse problems have been attracting much research attention [30-31, 35-36]. They are able to perform both zooming and denoising within the same framework.

The directional information also plays a key role in interpolation. The traditional linear interpolation methods, such as bi-linear and bi-cubic interpolation, are simple and fast but they do not work well in edge preservation due to the ignorance of local directional information. Most of the later developed interpolation techniques aim at maintaining the edge sharpness. Jensen and Anastassiou [6] detected edges and fitted them by some predefined templates to improve the visual perception of enlarged images. The image interpolator by Carrato and Tenze [7] first replicates the pixels and then corrects them by using some preset 3×3 edge patterns and optimizing the parameters in the operator. Muresan [10] detected the edge in diagonal and non-diagonal directions and then recovered the missing samples along the detected direction by using 1-D polynomial interpolation. In [8], Li *et al* estimated the local covariance matrix, which can reflect the local directional information, from the LR image, and then used it to calculate the interpolation parameters. Chang *et al* [11] adaptively estimated the singularity of wavelet coefficients to predict the unknown image edge information for image zooming. The EASE scheme in [14] tries to correct the interpolation error generated by bilinear interpolator to enhance the edge structure. Other edge preserving interpolators can be found in [13, 15].

The linear minimum mean square-error estimation (LMMSE) technique [37] was used in [33-34] to estimate the missing samples. In [12], Zhang *et al* proposed an interpolation algorithm by computing the directional estimates of the missing samples and then fusing them optimally. Directional edge guided interpolation has proved to be very effective in preserving the edges for image enlargement.

Although denoising and interpolation have been studied for many years as two independent problems, they can be modeled under the same framework of signal estimation. In this paper, we propose a directional denoising and interpolation algorithm to estimate both the noiseless and missing samples from the noisy image under the LMMSE framework. For each noisy pixel, a local window centered at it is used to analyze the local statistics. To calculate and preserve the directional information for image interpolation, we first compute multiple estimates of a noisy pixel along different directions. Those directional estimates are then fused for a more accurate output. The data fusion is adaptive to the statistics of the directional estimates to ensure the denoising being along the main direction of the local window. Such a directional denoising process naturally provides a directional interpolator, which consequently enlarges the image to a higher resolution.

This rest of the paper is organized as follows. Section 2 formulates the problem and presents the basic idea of the proposed solution. Section 3 describes in detail the directional estimation based denoising and interpolation scheme under the LMMSE framework. Section 4 presents the experimental results and Section 5 concludes the paper.

2. PROBLEM FORMULATION

In this paper, we consider the problem of image interpolation as to construct a high resolution (HR) image, whose size is $r \cdot N \times r \cdot M$, from its low resolution (LR) counterpart, whose size is $N \times M$, where r is the interpolation factor. As in many previous literature, for the convenience of discussion, we focus on the case of $r = 2$. If $r=2^k$, where k is an integer, the interpolation can be implemented by running k times the proposed algorithm, while each time the output of previous interpolation step is taken as the input for next round interpolation. If r is an arbitrary real number, we write r as $r=2^k \times c$, where 2^k is the closest integer to r and $c=r/2^k$. Then we first interpolate the image by a factor of 2^k

using the proposed method, and subsequently apply bicubic interpolation to the interpolated image with a factor of *c*.

There are two commonly used models to generate an LR image from an HR image. In the first model, the LR image is directly down-sampled from the HR image. In the second model, the HR image is smoothed by a point spread function (PSF), e.g. a Gaussian function, and then down-sampled to the LR image. The estimation of the HR image under the first model is often called image interpolation, while the HR image estimation under the second model can be viewed as an image restoration problem. In this paper, we consider the first model. Figure 1 illustrates the sample placement of HR and LR images, where the black dots represent the available LR pixels and the white dots represent the unknown HR samples to be interpolated. The black and white dots together form the full HR image.

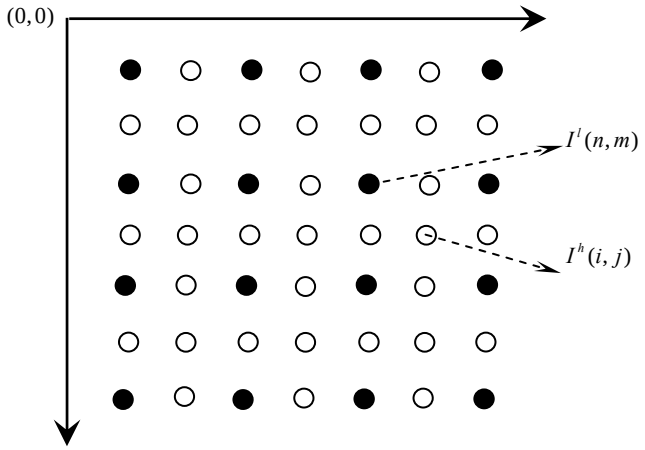


Fig. 1. The low resolution image samples (black dot) and the high resolution image samples (white dot) to be interpolated.

Refer to Figure 1, we use two coordinate systems to represent the LR and HR images. The pixels in the LR image are denoted as $I^l(n,m)$, n=1,2,…,N; m=1,2,…,M; while the pixels in the HR image are denoted as $I^h(i,j)$, i=1,2,…,2N; j=1,2,…,2M. For both the coordinate systems, the origin is set as the top left corner. At the black dot positions, we have $I^h(i,j) = I^l(n,m)$ with $i = 2n-1, j = 2m-1$. All the other samples at the white dot positions are to be interpolated with the help of their black dot neighbors.

Most of the existing interpolation techniques [2-15] assume a noiseless LR image. In real applications, however, the available LR data are noisy because of the inevitable noise corruption in the image acquisition process. It is accepted that the corrupted noise in charge-coupled device (CCD) and complementary-symmetry/metal-oxide semiconductor (CMOS) sensors is signal-dependent [32]. Foi *et al* [32] pointed out that the noise variance depends on the signal magnitude, while Poisson, film-grain, multiplicative and speckle models can be used to model the noise. In [16], Hirakawa modeled the raw sensor output as $y = x + (k_0 + k_1 x)\upsilon$, where x is the desired noiseless signal, $\upsilon \in N(0,1)$ is unit Gaussian white noise and k_0 and k_1 are sensor dependent parameters. Although this noise model may fit some sensors better, the design of denoising algorithms may be very complex and the computational cost may be very expensive.

One simple and widely used noise model is the signal-independent additive noise model $y = x + \upsilon$. It is a special case of the signal-dependent noise model with $k_1 = 0$ and commonly used to approximate the Poisson noise in CCD/CMOS sensors. Since the additive noise model is simple to use in the design and analysis of denoising algorithms, it has been widely used in the literature [18-23]. The signal-dependent noise characteristic can be compensated by estimating the noise variance adaptively in each local area [33], i.e. we can let the additive noise level vary spatially to approximate the signal-dependent noise characteristic. Therefore, in this paper we adopt the widely used additive noise model, and for the convenience of expression we assume the noise level is a constant in the whole image. There is

$$I_\upsilon^l(n,m) = I^l(n,m) + \upsilon(n,m) \tag{1}$$

where I_υ^l is the noisy LR image and υ is zero mean white noise with variance σ^2.

The conventional way to enlarge the noisy LR image I_υ^l is to perform denoising and interpolation in tandem, i.e. denoising first and interpolation later. However, if the denoising process is designed without consideration of the following interpolation process, the artifacts (such as block effects, blur, stings, etc) generated in the denoising process can be amplified in the interpolation process. Another strategy is to interpolate the noisy image first and then denoise the interpolated image. This strategy

will complicate the noise characteristics and generate many noise-caused interpolation artifacts, which will be hard to remove in the following denoising process. Therefore, there is a high demand to develop new interpolation techniques for noisy images.

Instead of viewing denoising and interpolation as two separate processes, in this paper we aim to develop a unified framework for denoising and interpolation. The basic idea is to model the two problems under the same framework of signal estimation. For the available noisy samples, we estimate their noiseless counterparts; then for the missing samples, we estimate them from the available estimated noiseless neighbors. Since the human visual system is sensitive to image edge information, directional denoising and interpolation techniques will be used.

For an existing noisy pixel in I_{υ}^{l}, the denoising of it can be viewed as how to estimate the true value at this position by using the noisy measurements in the neighborhood of it. For example, to estimate the value at noisy pixel $I_{\upsilon}^{l}(n,m)$, we can take the estimation, denoted by $\hat{I}^{l}(n,m)$, as the linear function of pixel $I_{\upsilon}^{l}(n,m)$ and its nearest neighbors, i.e.

$$\hat{I}^{l}(n,m) = \vec{a}^{T}\vec{x} + b \tag{2}$$

where vector $\vec{x}$ contains the noisy pixels in the neighborhood of position (*n*, *m*), $\vec{a}$ is the weight vector containing the weights assigned to $\vec{x}$, *b* is a constant representing some bias value, and symbol "*T*" is the transpose operator. With $\vec{a}$ and *b*, $\hat{I}^{l}(n,m)$ is estimated as a linear combination of noisy pixels in the neighborhood around position (*n*, *m*), and thus the problem of denoising is transformed into the problem of how to determine the weights $\vec{a}$ and constant *b*. Similarly, for a missing HR sample $I^{h}(i,j)$, we can estimate it as the linear function of its denoised LR neighbors. For instance, for $I^{h}(i,j)$ that has four closest diagonal LR neighbors, it can be estimated as

$$\hat{I}^{h}(i,j) = \vec{\alpha}^{T}\vec{\chi} + \beta \tag{3}$$

where vector $\vec{\chi}$ contains the four diagonal LR neighbors of $I^{h}(i,j)$, $\vec{\alpha}$ is the weight vector assigned to $\vec{\chi}$, and β is a constant representing some bias value. We see that the problem of interpolation is also a problem of how to determine the weight vector $\vec{\alpha}$ and constant β.

Now that both the denoising and interpolation problems are an estimation problem, how to calculate the weights $\vec{a}$ and $\vec{\alpha}$, and constants b and β, is the key issue in the denoising and interpolation process. In next section, we will present a spatially adaptive LMMSE scheme to compute the estimates of $I_{\upsilon}^{l}(n,m)$ and $I^{h}(i,j)$. Particularly, in order to perform a directional interpolation of the LR image, we present a directional LMMSE scheme for the denoising of I_{υ}^{l} so that the weights $\vec{a}$ learned in the denoising process can be consequently used in the interpolation step.

3. DIRECTIONAL DENOISING AND ITERPOLATION

Section 2 shows that the denoising and interpolation of a noisy LR image can be accomplished by calculating the weights $\vec{a}$ and $\vec{\alpha}$, and constants b and β. This section will present a spatially adaptive scheme for this purpose. The whole algorithm can be divided into two stages. In the first stage, the weights $\vec{a}$ are calculated to denoise the noisy LR pixels. In the second stage, the weights $\vec{\alpha}$ will be directly generated from $\vec{a}$, and interpolation can be consequently accomplished.

We employ directional filtering and data fusion techniques to denoise and estimate the noiseless and missing image samples. In sub-sections 3.1 and 3.2, we calculate the multiple directional estimates of the noiseless sample by adaptively determining the linear weights $\vec{a}$ and constant b. Sub-sections 3.3 and 3.4 fuse the multiple directional estimates for a better estimation. In sub-section 3.5, the interpolation weights $\vec{\alpha}$ are directly obtained from $\vec{a}$ and then the missing HR samples $I^{h}(i,j)$ can be interpolated. Next we discuss in detail the proposed techniques.

3.1. Directional Estimates of the Noiseless Samples

Let's first consider how to estimate the noiseless LR pixel $I^{l}(n,m)$, from its noisy measurement $I_{\upsilon}^{l}(n,m)$ and the neighbors of $I_{\upsilon}^{l}(n,m)$, as shown in Eq. (2). Certainly, the weights in $\vec{a}$ can be computed by using some optimal estimation technique such as the LMMSE [37]. For the convenience of expression, we denote by s_0 the desired noiseless pixel $I^{l}(n,m)$, by s_0^{υ} the noisy measurement

$I_{\upsilon}^{l}(n,m)$, and by s_i^{υ}, i=1, 2, …, n, the noisy neighbors of $I^{l}(n,m)$. Let $\vec{s}_{\upsilon} = \begin{bmatrix} s_1^{\upsilon} & s_2^{\upsilon} & \ldots & s_n^{\upsilon} \end{bmatrix}^T$, then the LMMSE of s_0, denoted by $\hat{s}_0$, given the measurement vector $\vec{s}_{\upsilon}$, is

$$\hat{s}_0 = E[s_0] + \frac{Cov(s_0, \vec{s}_{\upsilon})}{Var(\vec{s}_{\upsilon})}(\vec{s}_{\upsilon} - E[\vec{s}_{\upsilon}]) \tag{4}$$

where $Cov(\vec{x}, \vec{y}) = E\left[(\vec{x} - E[\vec{x}])(\vec{y} - E[\vec{y}])^T\right]$ is the co-variance of vectors $\vec{x}$ and $\vec{y}$, and $Var(\vec{x}) = Cov(\vec{x}, \vec{x})$ is the variance of vector $\vec{x}$. Comparing Eq. (2) with Eq. (4), we have

$$\vec{a} = \frac{Cov(s_0, \vec{s}_{\upsilon})}{Var(\vec{s}_{\upsilon})} \text{ and } b = E[s_0] - \vec{a}E[\vec{s}_{\upsilon}] \tag{5}$$

If a 3×3 window around $I^{l}(n,m)$ is used, $\vec{s}_{\upsilon}$ will be a 9×1 variable vector and its variance matrix $Var(\vec{s}_{\upsilon})$ is a 9×9 matrix. The inverse of the 9×9 matrix $Var(\vec{s}_{\upsilon})$ will cost much computation, and as we will see in Section 3.5, in order to accommodate the directional interpolation, we divide the estimation of $I^{l}(n,m)$ into several sub-problems, each of which yields a directional estimate of $I^{l}(n,m)$, and then fuse those directional estimates into a more robust one.

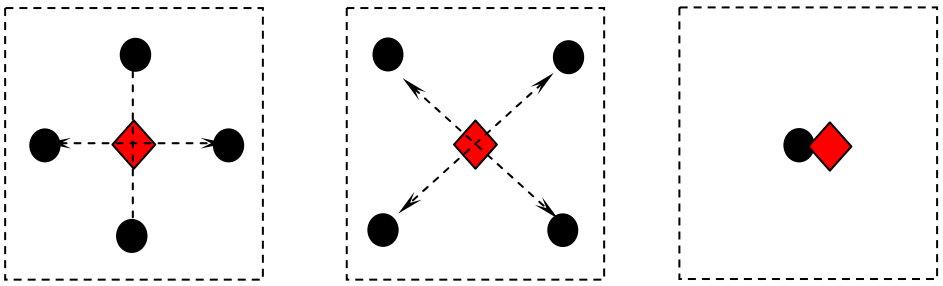


Fig. 2. Partition of the nine measurements into 3 groups to estimate the unknown noiseless sample.

Refer to Figure 2, we partition the nine noisy samples within the 3×3 window centered on (*n*, *m*) into three groups along different directions: horizontal/vertical, diagonal and the noisy sample $I_{\upsilon}^{l}(n,m)$. The red diamond represents the noiseless pixel to be estimated and the black circles represent the available noisy measurements. Each of the first two groups has four elements and the last group has one member only. With the three groups, we are able to calculate three directional estimates

of $I^l(n,m)$. The three estimates can then be adaptively fused to obtain a more robust and accurate estimation of $I^l(n,m)$ (refer to sub-section 3.3 please).

3.2. Adaptive Parameter Estimation

Let's discuss how to estimate $I^l(n,m)$ using the four horizontal/vertical neighboring measurements. The other two directional estimates can be obtained similarly. Let s_1^υ, …, s_4^υ be the four horizontal/vertical noisy neighbors of $I^l(n,m)$ and let $\vec{s}_\upsilon = \begin{bmatrix} s_1^\upsilon & s_2^\upsilon & s_3^\upsilon & s_4^\upsilon \end{bmatrix}^T$, then the LMMSE of s_0 can be computed by using Eq. (4). To solve the LMMSE $\hat{s}_0$ in Eq. (4), however, we need to estimate the 1st order and 2nd order statistics of s_0 and $\vec{s}_\upsilon$, including the mean values $\mu_0 = E[s_0]$, $\vec{\mu}_s = E[\vec{s}_\upsilon]$, the variance matrix $\mathbf{R} = E\left[(\vec{s}_\upsilon - \vec{\mu}_s)(\vec{s}_\upsilon - \vec{\mu}_s)^T\right]$ and the covariance vector $\vec{r} = E\left[(s_0 - \mu_0)(\vec{s}_\upsilon - \mu_s)^T\right]$.

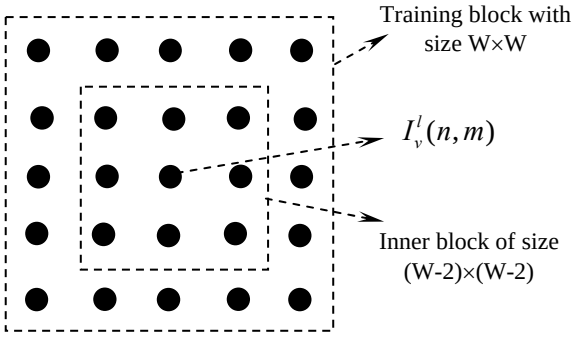


Fig. 3. Illustration of the training block centered on $I_\upsilon^l(n,m)$.

The parameters μ_0, $\vec{\mu}_s$, $\mathbf{R}$ and $\vec{r}$ can be estimated adaptively by using a window centered on s_0^υ, i.e. $I_\upsilon^l(n,m)$. Referring to Figure 3, suppose the size of the training block is W×W. For each pixel within the horizontal inner block of size (W-2)×(W-2), there are two horizontal and two vertical neighbors of it. Denote by $\vec{S}_0^\upsilon$ the column vector containing all the noisy measurements inside the

inner block and by $\vec{S}_0$ its unknown noiseless counterpart. Then we can form four sample vectors, denoted by $\vec{S}_1^{\upsilon}$, …, $\vec{S}_4^{\upsilon}$, containing the four sets of horizontal/vertical neighbors of $\vec{S}_0^{\upsilon}$, respectively. The parameters μ_0, $\vec{\mu}_s$, $\mathbf{R}$ and $\vec{r}$ can be estimated by using the sample vectors $\vec{S}_0^{\upsilon}$, $\vec{S}_1^{\upsilon}$, …, $\vec{S}_4^{\upsilon}$. However, sample structures may change within the training block. Involving all the samples in $\vec{S}_0^{\upsilon}$ for statistics calculation may lead to much estimation bias. Intuitively, one solution to this problem is to select from $\vec{S}_0^{\upsilon}$ the similar samples to the underlying central sample $I_{\upsilon}^{l}(n,m)$ and use them only for parameter estimation. Such a training sample selection procedure can make the estimation of parameters μ_0, $\vec{\mu}_s$, $\mathbf{R}$ and $\vec{r}$ more accurate.

We perform the training sample selection by using the simple block matching technique. Denote by $\mathbf{x}$ the 5×5 block [2] centered on $I_{\upsilon}^{l}(n,m)$ and by $\vec{x}_0$ the vectors containing the pixels in $\mathbf{x}$. Then for each pixel in $\vec{S}_0^{\upsilon}$, we have a vector, denoted by $\vec{x}_k$, containing the pixels in the 5×5 block centered on it. The L_1–norm distance between $\vec{x}_0$ and $\vec{x}_k$ can be computed as

$$d_k = \sum_{i=1}^{25} \left| \vec{x}_k(i) - \vec{x}_0(i) \right| \tag{6}$$

Obviously, the smaller the distance d_k is, the more similar $\vec{x}_k$ is to $\vec{x}_0$. Therefore, we select the best K training samples from $\vec{S}_0^{\upsilon}$ based on their associated distances d_k, i.e. we select the first K samples with the smallest distances d_k. In practice, K should be large enough to guarantee a reasonable estimation of the parameters μ_0, $\vec{\mu}_s$, $\mathbf{R}$ and $\vec{r}$.

For the convenience of expression, in the following development we still use $\vec{S}_0^{\upsilon}$ to represent the vector of the K selected samples, and use $\vec{S}_1^{\upsilon}$, …, $\vec{S}_4^{\upsilon}$ to represent its associated four neighboring sample vectors. Because the additive noise is zero mean, the mean value of s_0, i.e. μ_0, can be approximately estimated as

[2] The size of the block should be small to represent the local structural information. However, a too small block size will be sensitive to noise. By our experience, a block size of 5×5 is appropriate.

$$\mu_0 \approx \frac{1}{K}\sum_{k=1}^{K}\vec{S}_0^{\upsilon}(k) \tag{7}$$

Similarly, the mean values of $\vec{S}_1^{\upsilon}$, …, $\vec{S}_4^{\upsilon}$ can be estimated as $\mu_i \approx \frac{1}{K}\sum_{k=1}^{K}\vec{S}_i^{\upsilon}(k)$, i=1,2,3,4, approximately. Thus the mean vector of $\vec{s}_{\upsilon}$ is

$$\vec{\mu}_s = \begin{bmatrix}\mu_1 & \mu_2 & \mu_3 & \mu_4\end{bmatrix}^T \tag{8}$$

Subtracting the mean values from the associated variable and data vector, we obtain the centralized variables and data vectors as $\overline{s}_0 = s_0 - \mu_0$, $\overline{s}_i^{\upsilon} = s_i^{\upsilon} - \mu_i$, $\bar{\vec{S}}_0 = \vec{S}_0 - \mu_0$ and $\bar{\vec{S}}_i^{\upsilon} = \vec{S}_i^{\upsilon} - \mu_i$, i=0,1,2,3,4. Let $\overline{\mathbf{S}} = \begin{bmatrix}\bar{\vec{S}}_1^{\upsilon} & \bar{\vec{S}}_2^{\upsilon} & \bar{\vec{S}}_3^{\upsilon} & \bar{\vec{S}}_4^{\upsilon}\end{bmatrix}$ be the centralized sample matrix of $\vec{s}_{\upsilon}$. The covariance matrix $\mathbf{R}$ can then be estimated by

$$\mathbf{R} \approx \frac{1}{K}\overline{\mathbf{S}}^T\overline{\mathbf{S}} \tag{9}$$

The element of $\vec{r}$, $\vec{r}(i) = E\left[\overline{s}_0\overline{s}_i^{\upsilon}\right]$, i=1,2,3,4, can be estimated as $\vec{r}(i) \approx \frac{1}{K}\bar{\vec{S}}_0^T\bar{\vec{S}}_i^{\upsilon}$. However, $\bar{\vec{S}}_0$ is not available in practice. Fortunately, since the noise υ is white additive and uncorrelated with signal I^l , we have $\vec{r}(i) = E\left[\overline{s}_0\overline{s}_i^{\upsilon}\right] = E\left[(\overline{s}_0 + v_0)\overline{s}_i^{\upsilon}\right] = E\left[\overline{s}_0^{\upsilon}\overline{s}_i^{\upsilon}\right]$. Thus $\vec{r}$ can be estimated approximately by

$$\vec{r} \approx \frac{1}{K}(\bar{\vec{S}}_0^{\upsilon})^T\overline{\mathbf{S}} \tag{10}$$

Now that all the required parameters μ_0 , $\vec{\mu}_s$, $\mathbf{R}$ and $\vec{r}$ have been estimated, the LMMSE estimate $\hat{s}_0$ can then be calculated using Eq. (4). Denote by $e_0 = \hat{s}_0 - s_0$ the estimation error of $\hat{s}_0$. The variance of estimation error e_0 can be calculated as

$$\sigma_0^2 = E\left[e_0^2\right] = E\left[\overline{s}_0^2\right] - \vec{r}^T\mathbf{R}^{-1}\vec{r} \tag{11}$$

where $E\left[\overline{s}_0^2\right]$ can be estimated as:

$$E\left[s_0^2\right] \approx \frac{1}{K}(\bar{\vec{S}}_0^{\upsilon})^T\bar{\vec{S}}_0^{\upsilon} - \sigma^2 \tag{12}$$

Similar to what described above, each one of the three groups of measurements in Figure 2 will yield an estimate of s_0. Therefore we have three estimates of s_0 from different directions. Those directional estimates may have different estimation errors, depending on the edge structure at the current pixel s_0. Intuitively, the three estimates can be fused for a more accurate estimate of s_0.

3.3. Fusion of Directional Estimates

In sections 3.1 and 3.2, we have calculated three estimates of s_0. Denote by $\hat{s}_0^i$, i=1,2,3, the three directional estimates, each of which can be written as

$$\hat{s}_0^i = s_0 + e_0^i \tag{13}$$

where e_0^i is the directional estimation error and its variance, denoted by $\sigma_i^2 = E\left[\left(e_0^i\right)^2\right]$, can be calculated via Eq. (11). The optimal fusion of $\hat{s}_0^i$ (e.g. in the sense of minimum mean square error) requires to know the joint statistics of the e_0^i and s_0. However, this is hard to know *a prior* in practice. In this section, we present a simple but efficient way to fuse the three estimates.

We take the fusion of $\hat{s}_0^i$ as the weighted average of them:

$$\hat{s}_0^f = \sum_{i=1}^{3} w_i \hat{s}_0^i \tag{14}$$

where the weights $w_1 + w_2 + w_3 = 1$ and $\hat{s}_0^f$ is the fusion result of $\hat{s}_0^i$. Intuitively, if a directional estimate $\hat{s}_0^i$ has greater variance of estimation error, it should contribute less to the final fusion result. Therefore, we use the variances σ_i^2 to control the weights. Let

$$\vec{\delta} = \begin{bmatrix} 1/\sigma_1^2 & 1/\sigma_2^2 & 1/\sigma_3^2 \end{bmatrix} \tag{15}$$

We then normalize $\vec{\delta}$ as

$$\bar{\vec{\delta}} = \vec{\delta} / \Sigma_\delta \tag{16}$$

where $\Sigma_\delta = \sum_{i=1}^{3} \frac{1}{\sigma_i^2}$. Finally, we set the weight vector $\vec{w} = \begin{bmatrix} w_1 & w_2 & w_3 \end{bmatrix}$ as

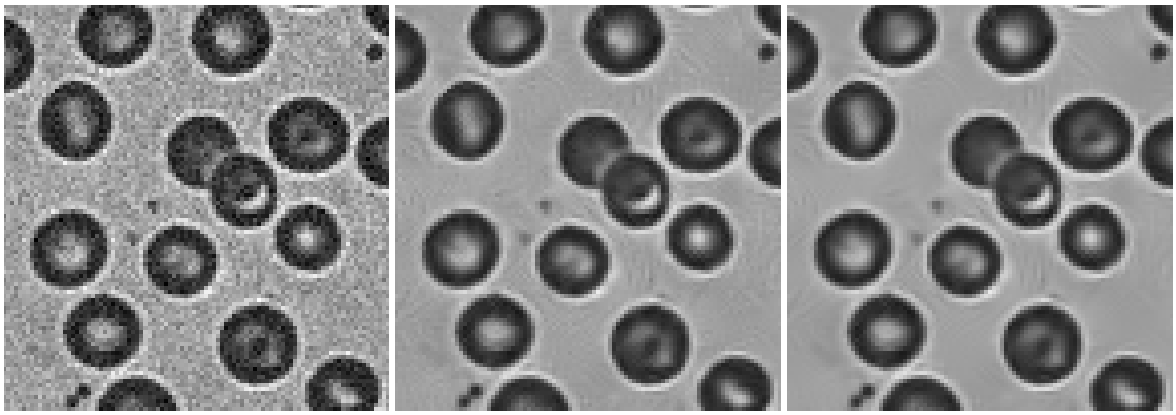
$$\vec{w} = \vec{\bar{\delta}} \tag{17}$$

By assuming that the directional estimation errors e_0^i and e_0^j are uncorrelated for $i \neq j$, we can calculate the variance of the estimation error of $\hat{s}_0^f$, denoted by σ_f^2, as

$$\sigma_f^2 = E\left[\left(\hat{s}_0^f - s_0\right)^2\right] = E\left[\left(\sum_{i=1}^{3} w_i \cdot e_0^i\right)^2\right] = \sum_{i=1}^{3} w_i^2 \cdot \sigma_i^2 \tag{18}$$

3.4. Denoising Refinement

Sub-sections 3.1~ 3.3 present a directional denoising scheme by calculating the image local directional statistics adaptively. However, because the image is noise corrupted, there will be inevitably estimation errors of the key denoising parameters μ_0, $\vec{\mu}_s$, $\mathbf{R}$ and $\vec{r}$. Such errors will not only degrade the denoising performance but also affect the sequent directional interpolation process because $\mathbf{R}$ and $\vec{r}$ will be directly used for interpolation. Figure 4(a) shows a noisy *Blood Cell* image (128×128, PSNR=24.51dB) and Figure 4(b) is the denoising output by the proposed method in sub-sections 3.1~3.3 (PSNR=29.56dB). We can see some residual noise in the denoised image. Since the noise has been significantly removed in Figure 4(b), one intuitive solution to refine the denoising result is to apply the directional denoising procedure one more time to the denoising output so that the denoising parameters μ_0, $\vec{\mu}_s$, $\mathbf{R}$ and $\vec{r}$ can be more accurately estimated.



(a) (b) (c)

Fig. 4. (a) A noisy *Blood Cell* image; (b) the directional denoising result; and (c) the final denoising result after one-time iteration of the directional denoising procedure.

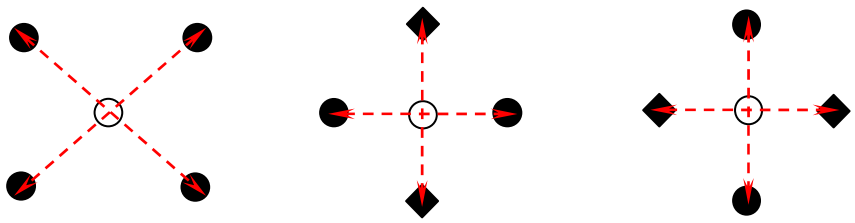
The only parameter to be re-configured in the second round directional denoising is the noise level because most of the noise energy has been removed in the first round. Refer to Eq. (18), for each noisy pixel, after directional estimation and optimal fusion we can compute the variance of its estimation error, i.e. σ_f^2. Denote by $\bar{\sigma}_f^2$ the average variance over the whole image. The average variance $\bar{\sigma}_f^2$ can reflect the magnitude of estimation error, which mainly comes from two sources: the residual noise and the estimation error from the neighboring pixels. Therefore, we can set $\sigma_r = c \cdot \bar{\sigma}_f$ as the noise level in the second round directional filtering, where *c* is a constant. By our experience, let *c* be around 0.4 can achieve satisfying output. Figure 4(c) shows the second round directional denoising result of the noisy *Blood Cell* image (PSNR=29.81dB). We see that although the PSNR values are almost the same, much residual noise in Figure 4(b) is reduced and the image is more visually pleasing.

3.5. Directional Interpolation

By re-writing the directional LMMSE of the LR noisy pixel in Eq. (4) as the form in Eq. (2), we have

$$\vec{a} = \vec{r} \cdot \mathbf{R}^{-1} \text{ and } b = \mu_0 - \vec{a}\vec{\mu}_s \tag{19}$$

and hence $\hat{s}_0 = \vec{a} \cdot \vec{s}_\upsilon + b$. We can clearly see that the directional LMMSE of a pixel is actually the weighted average of its neighbors plus a constant, and the weights are represented as the vector $\vec{a}$. Using the directional estimation technique described in sub-sections 3.1 ~ 3.4, for each pixel in the LR image, we have three sets of weights along different directions. As mentioned in Section 2, the problem of interpolating the missing samples can also be considered as the weighted average of its available neighbors plus a constant (referring to Eq. (3) please). In this section, we will see that the weights estimated in the directional denoising process can be simultaneously used to estimate the missing HR samples.



(a) (b) (c)

Fig. 5. Directional interpolation of the missing HR samples (white circle). (a) Diagonal, (b) horizontal and (c) vertical interpolation. The black circles represent the denoised LR pixels, and the black diamonds represent the interpolated HR samples in (a).

As shown in Figure 5, the missing HR samples to be interpolated can be divided into three groups: the diagonal samples, the horizontal samples and the vertical samples. For a missing diagonal HR sample, denoted by I_d^h, it has four diagonal LR neighbors (refer to Figure 5 (a)). Suppose we have estimated I_d^h with the help of its neighbors, then for a missing horizontal (vertical) HR sample, denoted by I_h^h (I_v^h), it has two horizontal (vertical) LR neighbors and two interpolated vertical (horizontal) HR neighbors (refer to Figures 5 (b) and 5 (c) please). Next we discuss the interpolation of I_d^h. The interpolation of horizontal (vertical) HR samples I_h^h (I_v^h) can be conducted similarly.

Denote by I_k^l, k=1,2,3,4, the four diagonal denoised LR neighbors of a missing HR sample I_d^h. We interpolate I_d^h as

$$\hat{I}_d^h = \bar{\alpha}_d \bar{\chi}_d + \beta_d \tag{20}$$

where $\bar{\chi}_d$ is the vector containing the four LR pixels I_k^l, $\bar{\alpha}_d$ is the weight vector and β_d is a constant. Recall that in sub-section 3.2, we have calculated a set of weights along diagonal direction, denoted as $\bar{a}_k^d$, for each I_k^l. Since I_k^l and I_d^h are within the same local area, they will have similar local structures. The directional weights that yield a good directional estimate of I_k^l will also tend to yield a good directional estimate of I_d^h. Therefore, we utilize the weights associated with I_k^l to interpolate I_d^h by setting $\bar{\alpha}_d$ as the average of $\bar{a}_k^d$

$$\bar{\alpha}_d = \frac{1}{4}\sum_{k=1}^{4} \bar{a}_k^d \tag{21}$$

Once $\bar{\alpha}_d$ is fixed, the other parameter, i.e. β_d, in Eq. (20) can be calculated as

$$\beta_d = \mu_0 - \bar{\alpha}_d \bar{\mu}_d \tag{22}$$

where μ_0 is the mean value of I_d^h and $\bar{\mu}_d = \begin{bmatrix} \mu_1 & \mu_2 & \mu_3 & \mu_4 \end{bmatrix}^T$ is the mean vector of $\bar{\chi}_d$. In implementation, μ_0 is estimated as the average of the *K* nearest LR neighbors of I_d^h. For each one of μ_k, *k*=1,2,3,4, it is estimated as the average of I_k^l and the *K* nearest LR neighbors of I_k^l. Finally, $\hat{I}_d^h$ can be interpolated using Eq. (20).

After interpolating the diagonal HR samples I_d^h, the horizontal (vertical) HR samples I_h^h (I_v^h) can then be interpolated. Referring to Figures 5 (b) and 5 (c), a horizontal (vertical) HR sample has two LR neighbors and two interpolated diagonal HR neighbors (black diamond). Similar to the interpolation of diagonal HR samples, the horizontal (vertical) HR samples are estimated by weighting the four neighbors. Different from the interpolation of diagonal HR samples, for which the directional interpolation weights are calculated as the average of the weights of the four diagonal neighbors $\bar{\chi}_d$, the directional interpolation weights for a horizontal (vertical) HR sample are calculated as the average of the weights of its two horizontal (vertical) LR neighbors. Now, all the missing HR samples can be estimated and the directional denoising and interpolation is accomplished.

3.6. Summary of the Algorithm

The proposed noisy image denoising and zooming algorithm can be summarized as follows.

Input: noisy LR image.
Output: denoised and zoomed HR image.

---

1. Denoising stage

Input: noisy LR image.

For each LR pixel s_0 with noisy measurement s_0^v

(i) Calculate three directional estimates of it by using its four diagonal neighbors, four horizontal and vertical neighbors and its central measurement s_0^v. Denote by $\hat{s}_0^i$, *i*=1,2,3, the three estimates.

(ii) By using Eq. (19), the three estimates can be written as

$$\hat{s}_0^i = \bar{a}_i \cdot \bar{s}_v^i + b_i$$

---

where $\vec{a}_i$ is weight vector and $\vec{s}_\upsilon^{\,i}$ is the vector containing the neighbors of s_0^υ.

(iii) Fuse the three estimates by using Eq. (14).

End

Update the noise level and repeat the above denoising procedure as described in Section 3.4. Then we have the refined denoising results and the updated weight vectors $\vec{a}_i$.

Output: denoised LR image and weight vectors $\vec{a}_i$.

2. Zooming stage

Input: denoised LR image and weight vectors $\vec{a}_i$.

By using Eqs. (21) and (22) and weights $\vec{a}_i$, compute $\vec{\alpha}$ and β.

By using Eq. (20), first interpolate the missing diagonal samples, and then interpolate the missing horizontal and vertical samples.

Output: zoomed HR image.

4. EXPERIMENTAL RESULTS

This section performs experiments to verify the proposed directional denoising and interpolation algorithm. For comparison, we employ the sophisticated wavelet based denoising schemes [21, 23] and the anisotropic diffusion denoising scheme [27] to denoise the LR image and then interpolate the denoised images using the state-of-the-art directional interpolation schemes [8, 12] respectively. Five test images, *Blood Cells*, *Lena*, *Butterfly*, *House* and *Peppers* are used in the experiments. The size of all the original images is 256×256. In the experiments, the original images are first downsampled to 128×128 and then added Gaussian white noise. Two noise levels with standard deviation σ=15 and σ=25 are tested respectively. Different denoising and interpolation schemes are then applied to the noisy LR image to compute the HR images. In the proposed directional denoising and interpolation scheme, the size of the local training window, i.e. W, should be preset. In general, a larger window size could improve the denoising result. However, this requires higher computational cost, and a higher window size may degrade the interpolation result because the estimation of image local structural statistics can be affected. Therefore, we set the size of local window as W=13 by experience. The parameter *K* is set about 70 in the implementation.

Table I. The PSNR (dB) results for different denoising and interpolation schemes (σ=15).

Methods	[21]+[8]	[21]+[12]	[23]+[8]	[23]+[12]	[27]+[8]	[27]+[12]	proposed
Blood Cells	29.31	29.47	28.88	29.11	28.01	28.05	29.43
Lena	26.01	26.05	25.87	25.97	25.60	25.65	25.87
Butterfly	24.92	25.24	24.74	25.18	24.58	24.85	24.65
House	29.02	29.26	28.52	28.84	28.33	28.52	28.71
Peppers	27.43	27.47	27.27	27.31	26.99	27.02	27.13

Table II. The PSNR (dB) results for different denoising and interpolation schemes (σ=25).

Methods	[21]+[8]	[21]+[12]	[23]+[8]	[23]+[12]	[27]+[8]	[27]+[12]	proposed
Blood Cells	26.80	26.82	26.03	26.08	25.74	25.66	26.53
Lena	24.54	24.58	24.21	24.32	24.21	24.17	24.21
Butterfly	23.21	23.39	22.78	23.01	22.99	23.20	22.61
House	27.42	27.54	26.79	26.96	26.07	26.08	26.72
Peppers	25.90	25.92	25.50	25.54	25.12	25.01	25.42

Table III. The MSSIM results for different denoising and interpolation schemes (σ=15).

Methods	[21]+[8]	[21]+[12]	[23]+[8]	[23]+[12]	[27]+[8]	[27]+[12]	proposed
Blood Cells	0.9670	0.9669	0.9611	0.9611	0.9557	0.9554	0.9667
Lena	0.9474	0.9478	0.9452	0.9457	0.9374	0.9379	0.9428
Butterfly	0.9728	0.9735	0.9688	0.9697	0.9678	0.9686	0.9709
House	0.9465	0.9474	0.9364	0.9367	0.9370	0.9382	0.9441
Peppers	0.9537	0.9535	0.9490	0.9488	0.9473	0.9469	0.9492

Table IV. The MSSIM results for different denoising and interpolation schemes (σ=25).

Methods	[21]+[8]	[21]+[12]	[23]+[8]	[23]+[12]	[27]+[8]	[27]+[12]	proposed
Blood Cells	0.9465	0.9460	0.9319	0.9313	0.9211	0.9184	0.9412
Lena	0.9136	0.9144	0.9084	0.9091	0.9019	0.9005	0.9087
Butterfly	0.9465	0.9460	0.9319	0.9313	0.9211	0.9184	0.9422
House	0.9245	0.9247	0.9049	0.9041	0.8615	0.8583	0.9116
Peppers	0.9225	0.9218	0.9114	0.9108	0.8874	0.8822	0.9140

The peak signal to noise ratio (PSNR) is a commonly used metric to evaluate the reconstructed image quality. However, it is well known that PSNR may not be able to faithfully reflect the image perceptual quality. How to better assess the image quality is still an open problem. In [38], Wang et al proposed the structural similarity (SSIM) index and multi-scale SSIM (MSSIM) for image quality assessment (IQA). SSIM and MSSIM have been used as new IQA measures in many image processing

applications. Here we also employ MSSIM to evaluate the image denoising and interpolation results in the experiments. The PSNR and MSSIM values of the interpolated images by various schemes are listed in Tables I ~ IV.

Figure 6 shows the interpolation results on the noisy image *Blood Cells* (σ=15). Figure 6 (a) is the original HR *Blood Cells* image; (b) is the denoised and interpolated result by the proposed directional denoising and interpolation scheme; (c) and (d) are the interpolation results by the method in [8] in combination with the denoising schemes [23, 21]; (e) and (f) are the interpolation outputs by the method in [12] in combination with the denoising schemes; and (g) and (h) are the interpolation outputs by the method in [27] in combination with the denoising schemes. Figures 7 ~ 10 show the interpolation results for the images *Lena*, *Butterfly*, *House* and *Peppers*, respectively.

From Tables I~IV, we can see that in average the proposed joint denoising and interpolation method have similar PSNR and MSSIM results to the "wavelet based denoising + directional interpolation" schemes. However, from Figures 6~10 we see that the proposed method leads to less block effects and ring effects. Since the edge directional information is adaptively estimated and employed in the denoising process and consequently in the interpolation process, the proposed directional joint denoising and interpolation algorithm can better preserve the edge structures, for example, blood cell's boundaries, Lena's hat, butterfly's wing, etc. Although the advanced interpolation schemes in [8, 12] are also designed for edge preservation, they fail to preserve some edge patterns because the previous denoising process may destroy such structures. In addition, some block effects, ring effects and other noise caused artifacts introduced in the wavelet denoising or anisotropic diffusion process can be amplified in the interpolation stage. Overall the presented joint denoising and interpolation scheme yields encouraging results.

5. CONCLUSION

This paper presented a directional interpolation scheme for noisy images. Unlike the conventional schemes that perform denoising first and interpolation later, the proposed method treats both denoising and interpolation as an estimation problem and implements them under a unified framework of linear

minimum mean square-error estimation (LMMSE). To avoid destroying much the edge directional information in the denoising stage and in order for a subsequent directional interpolation scheme, we computed multiple directional estimates of the noisy samples and then fused them for a more robust estimation. The LMMSE parameters computed in the directional denoising stage provide automatically a directional LMMSE interpolator for the low resolution image. The experimental results validate that the proposed denoising and interpolation scheme results in visually pleasing enlarged images by preserving well the edge structures and suppressing many noise-caused artifacts.

6. REFERENCES

[1] William K. Pratt, *Digital Image Processing*, John Wiley & Sons, 2nd edition, April 1991.

[2] H. S. Hou et al, "Cubic splines for image interpolation and digital filtering," *IEEE Trans. Acoustic, Speech and Signal Processing*, vol. ASSP-26, pp. 508-517, 1978.

[3] M. Unser, "Splines: a perfect fit for signal and image processing," *IEEE Signal Processing Magazine*, pp. 22-38, Nov. 1999.

[4] J. Han and S. Baek, "Parametric cubic convolution scaler for enlargement and reduction of image," *IEEE Trans. on Consumer Electronics*, vol. 46, no. 2, pp. 247-256, May 2000.

[5] P. Thévenaz, T. Blu,and M. Unser, "Interpolation revisited," *IEEE Trans. Medical Imaging*, vol. 19, pp. 739-758, July 2000.

[6] K. Jensen and D. Anastassiou, "Subpixel edge localization and the interpolation of still images," *IEEE Trans. Image Processing*, vol. 4, pp. 285-295, Mar. 1995.

[7] S. Carrato and L. Tenze, "A high quality 2×image interpolator," *IEEE Signal Processing Letters*, vol. 7, pp. 132-135, June 2000.

[8] Xin Li and M. T. Orchard, "New edge-directed interpolation," *IEEE Trans. Image Processing*, vol. 10, pp. 1521-1527, Oct. 2001.

[9] F. Malgouyres and F. Guichard, "Edge direction preserving image zooming: a mathematical and numerical analysis," *SIAM J. Numer. Anal.*, vol. 39, pp. 1-37, 2001.

[10] D. D. Muresan, "Fast Edge Directed Polynomial Interpolation," *Proceedings of International Conference on Image Processing*, vol. 2, pp. 990-993, 2005.

[11] S. G. Chang, Z. Cvetkovic´ and M. Vetterli, "Locally adaptive wavelet-based image interpolation," *IEEE Trans. Image Processing*, vol. 15, pp. 1471-1485, June 2006.

[12] Lei Zhang and X. Wu, "An edge guided image interpolation algorithm via directional filtering and data fusion," *IEEE Trans. on Image Processing,* vol. 15, pp. 2226-2238, Aug. 2006.

[13] Q. Wang and R. K. Ward, "A new orientation-adaptive interpolation method," *IEEE Trans. on Image Processing*, vol. 16, pp. 889-900, April 2007.

[14] Y. Cha and S. Kim, "The error-amended sharp edge (EASE) scheme for image zooming," *IEEE Trans. Image Processing*, vol. 16, pp. 1496-1505, June 2007.

[15] S. Yang, Y. Kim and J. Jeong, "Fine edge-preserving techniques for display devices," *IEEE Trans. on Consumer Electronics*, vol. 54, no. 4, pp. 1761-1769, Nov. 2008.

[16] K. Hirakawa and T. W. Parks, "Joint demosaicking and denoising," *IEEE Trans. on Image Processing*, vol. 15, no. 8, pp. 2146 – 2157, Aug. 2006.

[17] Lei Zhang, X. Wu and D. Zhang, "Color reproduction from noisy CFA data of single sensor digital cameras," *IEEE Trans. Image Processing*, vol. 16, no. 9, pp. 2184-2197, Sept. 2007.

[18] D. L. Donoho and I. M. Johnstone, "Ideal spatial adaptation via wavelet shrinkage," *Biometrika*, vol. 81, pp. 425–455, 1994.

[19] R. R. Coifman and D. L. Donoho, "Translation-invariant de-noising," in *Wavelet and Statistics*, A. Antoniadis and G. Oppenheim, Eds. Berlin, Germany: Springer-Verlag, 1995.

[20] S. G. Chang, B. Yu and M. Vetterli, "Spatially adaptive wavelet thresholding with context modeling for image denoising," *IEEE Trans. on Image Processing*, vol. 9, pp. 1522-1531, Sep. 2000.

[21] J. Portilla, V. Strela, M. J. Wainwright and E. P. Simoncelli, "Image denoising using scale mixtures of Gaussians in the wavelet domain," *IEEE Trans. on Image Processing*, vol. 12, pp. 1338 – 1351, Nov. 2003.
[22] Lei Zhang, P. Bao and X. Wu, "Multiscle LMMSE-based image denoising with optimal wavelet selection," *IEEE Trans. on Circuits and Systems for Video Technology,* vol. 15, pp. 469-481, April 2005.
[23] A. Pizurica and W. Philips, "Estimating the probability of the presence of a signal of interest in multiresolution single- and multiband image denoising," *IEEE Trans. on Image Processing*, vol. 15, no. 3, pp. 654-665, March 2006.
[24] P. Perona and J. Malik, "Scale space and edge detection using anisotropic diffusion," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 12, pp. 629-639, 1990.
[25] L. Alvarez, P.-L. Lions, and J.-M. Morel, "Image selective smoothing and edge detection by nonlinear diffusion," *SIAM J. Numer. Anal.*, vol. 29, pp. 845-866, 1992.
[26] L. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Phys. D*, vol. 60, pp. 259–268, 1992.
[27] S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin, "An iterative regularization method for total variation-based image restoration," *Multiscale Model. Simul.*, vol. 4, pp. 460-489, 2005.
[28] M. Burger, G. Gilboa, S. Osher, J. Xu, "Nonlinear inverse scale space methods," *Comm. Math. Sci.*, vol. 4, no. 1, pp. 175-208, 2006.
[29] J. Xu, S. Osher, "Iterative regularization and nonlinear inverse scale space applied to wavelet based denoising," *IEEE Trans. Image Processing*, vol. 16, no. 2, pp.534-544, 2007.
[30] A. Roussos and P. Maragos, "Reversible Interpolation of Vectorial Images by an Anisotropic Diffusion-Projection PDE," *International Journal of Computer Vision*, (2009) 84: 130–145.
[31] G. Peyre, S. Bougleux and L. D. Cohen, "Non-local Regularization of Inverse Problems," *ECCV 2008*, Part III, LNCS 5304, pp. 57–68, 2008.
[32] A. Foi, S. Alenius, V. Katkovnik, and K. Egiazarian, "Noise measurement for raw-data of digital imaging sensors by automatic segmentation of nonuniform targets," *IEEE Sensors Journal*, vol. 7, pp. 1456–1461, 2007.
[33] H. J. Trussell, R. E. Hartwig, "Mathematics for demosaicking," *IEEE Trans. on Image Processing*, vol.11, issue 4, pp. 485-492, April 2002.
[34] D. Taubman, "Generalized Wiener reconstruction of images from colour sensor data using a scale invariant prior," *IEEE International Conference on Image Processing 2010*, vol. 3, pp. 801–804, 2000.
[35] David Tschumperlé and Rachid Deriche, "Anisotropic Diffusion Partial Differential Equations in Multi-Channel Image Processing: Framework and Applications," *Advances in Imaging and Electron Physics (AIEP)*, Academic Press (Ed.), pp. 145 – 209, 2007.
[36] P. Getreuer, "Contour stencils for edge-adaptive image interpolation," *Proc of the SPIE*, vol. 7257, 2009.
[37] E. W. Karmen and J. K. Su, *Introduction to Optimal Estimation*, Springer-Verlag London Limited, 1999.
[38] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Transactions on Image Processing,* vol. 13, no. 4, Apr. 2004.

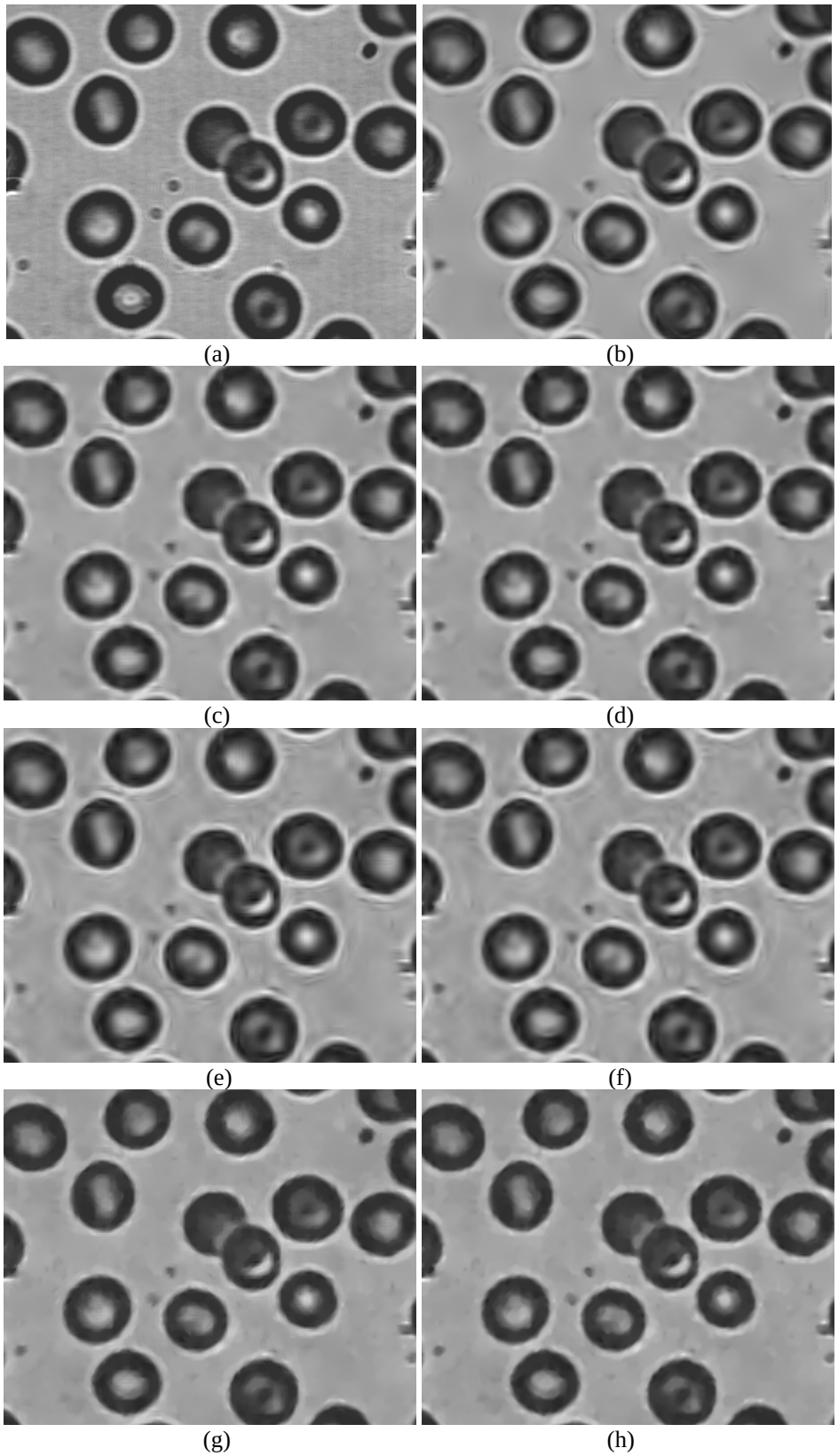


Fig. 6. Interpolation results on image *Blood Cells.* (a) Original image; (b) interpolated image by the proposed scheme; (c) and (d) are the interpolated images by the schemes [8] and [12] on the denoised image by [21]; (e) and (f) are the interpolated images by the schemes [8] and [12] on the denoised image by [23]; (g) and (h) are the interpolated images by the schemes [8] and [12] on the denoised image by [27].

Fig. 7. Interpolation results on image *Lena.* (a) Original image; (b) interpolated image by the proposed scheme; (c) and (d) are the interpolated images by the schemes [8] and [12] on the denoised image by [21]; (e) and (f) are the interpolated images by the schemes [8] and [12] on the denoised image by [23]; (g) and (h) are the interpolated images by the schemes [8] and [12] on the denoised image by [27].

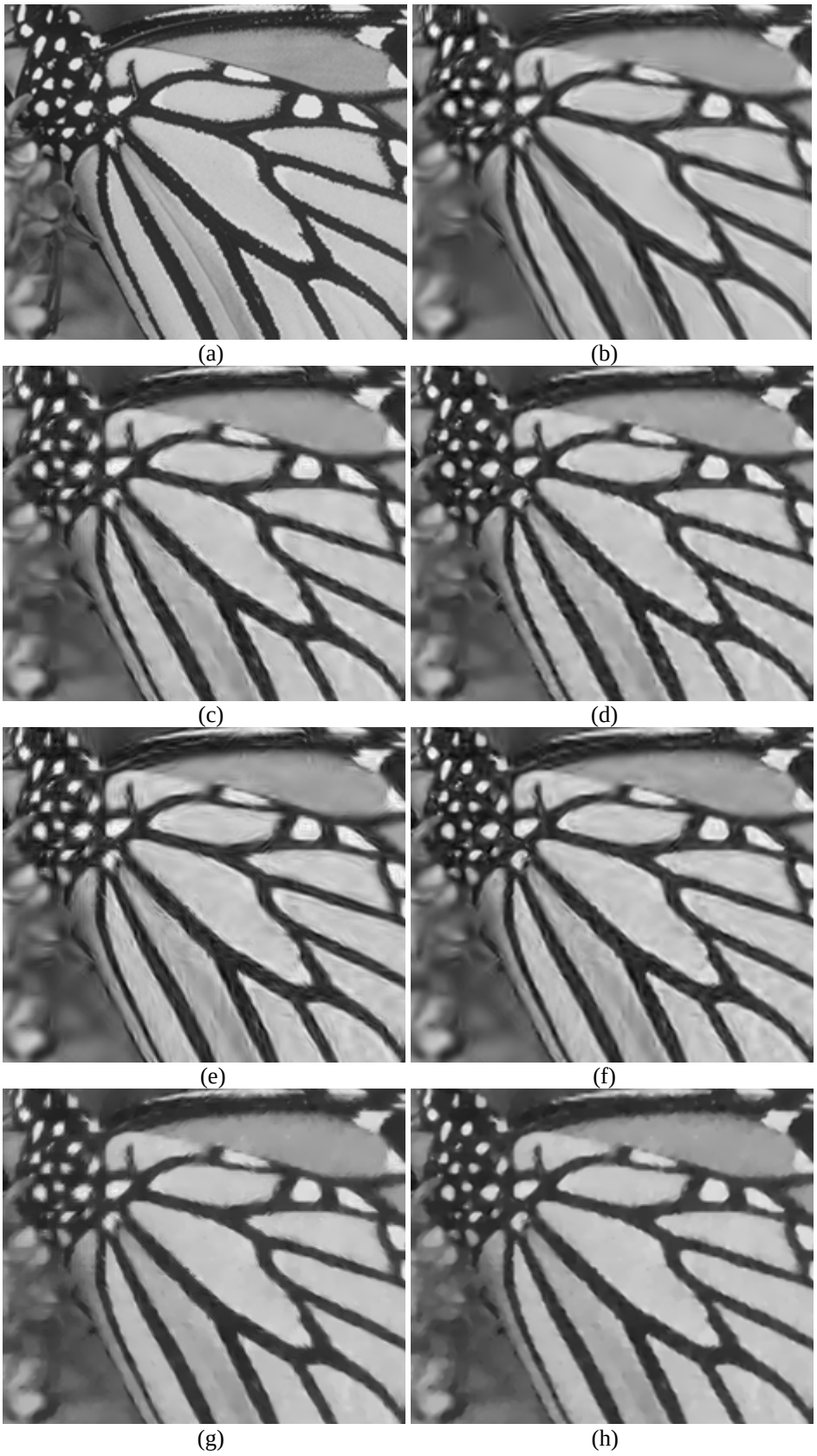


Fig. 8. Interpolation results on image *Butterfly*. (a) Original image; (b) interpolated image by the proposed scheme; (c) and (d) are the interpolated images by the schemes [8] and [12] on the denoised image by [21]; (e) and (f) are the interpolated images by the schemes [8] and [12] on the denoised image by [23]; (g) and (h) are the interpolated images by the schemes [8] and [12] on the denoised image by [27].

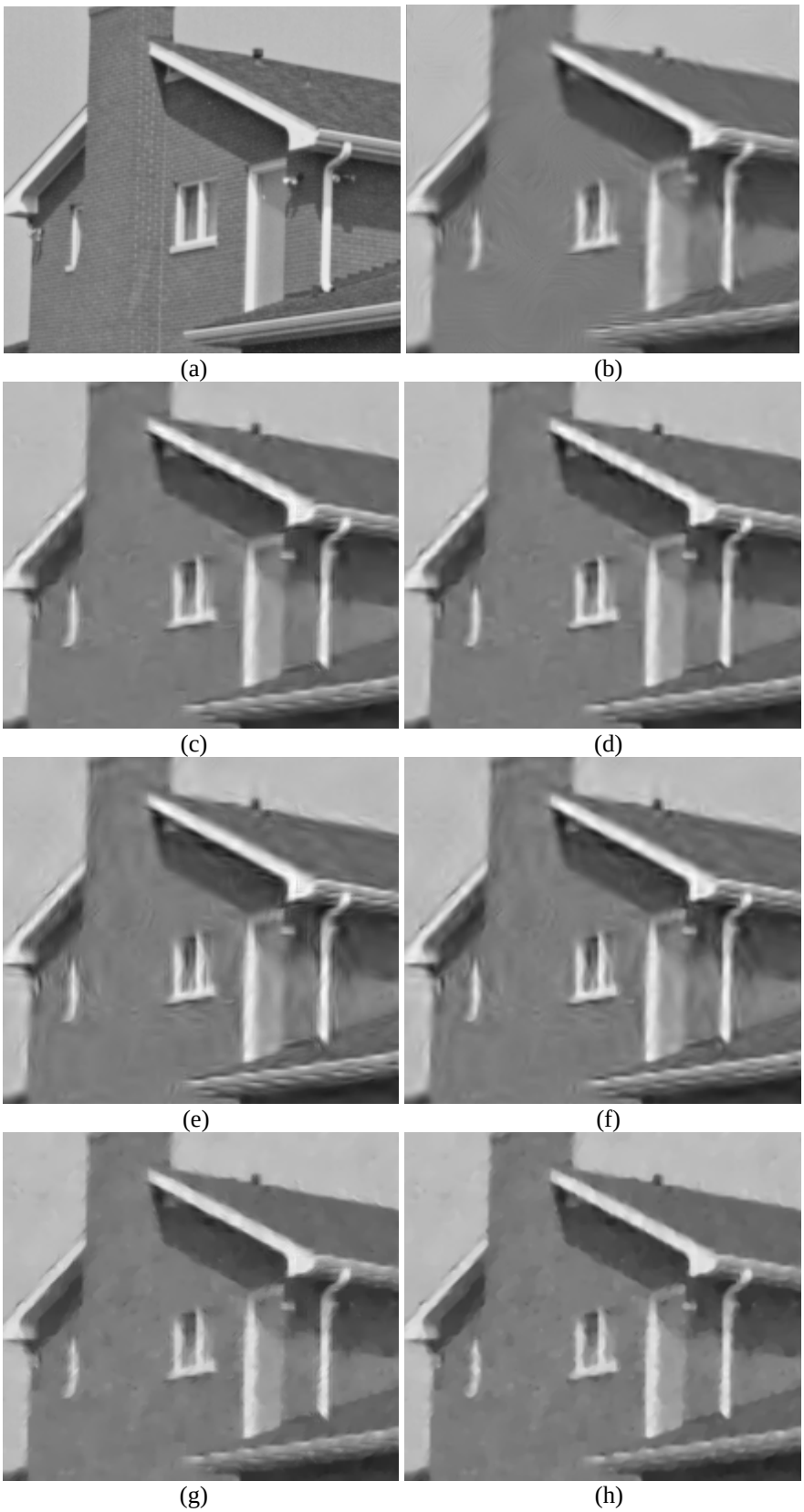


Fig. 9. Interpolation results on image *House.* (a) Original image; (b) interpolated image by the proposed scheme; (c) and (d) are the interpolated images by the schemes [8] and [12] on the denoised image by [21]; (e) and (f) are the interpolated images by the schemes [8] and [12] on the denoised image by [23]; (g) and (h) are the interpolated images by the schemes [8] and [12] on the denoised image by [27].

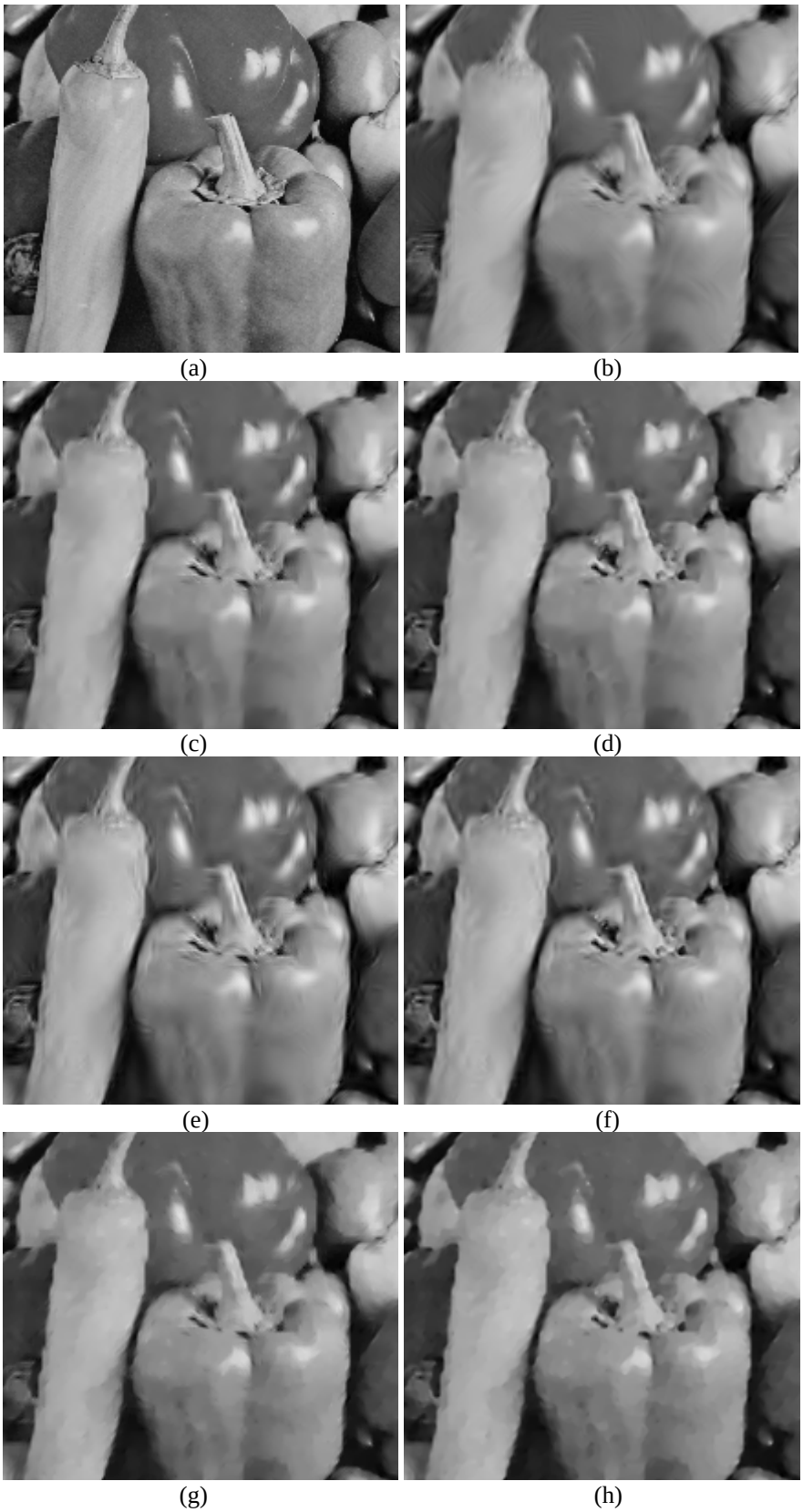


Fig. 10. Interpolation results on image *Peppers.* (a) Original image; (b) interpolated image by the proposed scheme; (c) and (d) are the interpolated images by the schemes [8] and [12] on the denoised image by [21]; (e) and (f) are the interpolated images by the schemes [8] and [12] on the denoised image by [23] ; (g) and (h) are the interpolated images by the schemes [8] and [12] on the denoised image by [27].