JOURNAL OF VISUAL COMMUNICATION AND IMAGE REPRESENTATION
Vol. 8, No. 4, December, pp. 338–355, 1997
ARTICLE NO. VC970364

Edge Preserving Interpolation of Digital Images Using Fuzzy Inference

Hou-Chun Ting and Hsueh-Ming Hang

Department of Electronics Engineering, National Chiao Tung University, Hsinchu, Taiwan 30050, Republic of China
E-mail: u8111830@cc.nctu.edu.tw, hmhang@cc.nctu.edu.tw

Received September 5, 1996; revised July 14, 1997

This paper presents a novel edge preserving interpolation method for digital images. This new method reduces drastically the blurring and jaggy artifacts at the high-contrast edges, which are generally found in the interpolated images using conventional methods. This high performance is achieved by two proposed operations: a fuzzy-inference based edge preserving interpolator and a highly oblique edge compensation scheme developed based on an edge orientation detector. The former synthesizes the interpolated pixels to match the image local characteristics. Hence, edge sharpness can be retained. However, due to the small footage of the fuzzy interpolation method, it cannot avoid edge jaggedness along the highly oblique edges that have very sharp angles against one of the coordinates. Therefore, a segment matching technique is developed to identify precisely the orientation of the highly oblique edges. Combining these two techniques, we improve significantly the visual quality of the interpolated images, particularly at the high-contrast edges. Both the synthesized images (such as letters) and the natural scenes (captured by camera) have been tested and the results are very promising. © 1997 Academic Press

1. INTRODUCTION

Interpolation is an important technique in multi-rate image signal processing such as pyramid coding [1], multiresolution television broadcasting [2, 3], and image zooming for viewing comfort. Linear interpolation methods are commonly used [4–7]. According to sampling theory, an ideal lowpass filter is needed to remove the replicated copies of the down-sampled (low resolution) signal spectra located at high frequencies. Therefore, the conventional linear spatial invariant interpolator is designed to be a lowpass filter. An extension of the above approach is patching pieces of continuous (spline) functions that match the given (known) pixels, and then, the pixels in-between can be synthesized from the continuous functions [5].

Blurring and jaggedness are the two most obvious artifacts in the interpolated images using the preceding interpolation methods. These artifacts are produced by the lowpass filter used to remove the unwanted highpass replica of the interpolated images in the frequency domain. Because it is not possible to implement the ideal low pass filter in practice, nonideal filters such as the zero-order hold (nearest neighbor) and the first-order hold (bilinear) operators are often employed to filter out the high frequency replica. These nonideal lowpass filters suppress low frequency components and bring in high frequency component aliasing. The former, low frequency suppression, reduces the spatial resolution of the interpolated images (blurring) and the latter, undesired high frequency aliasing, produces broken edges (jaggedness).

Recently, several adaptive nonlinear methods have been suggested to tackle the aforementioned problems. As it has been pointed out to various levels of extent [8–13] that in order to achieve a better visual quality, the lost high frequency components need to be reconstructed based on certain assumptions (models) of image characteristics. The basic concept is as follows. In the down-sampling process, the high frequency components in the original high resolution pictures are removed by lowpass filtering before resolution reduction. Now, the interpolated image has a higher spatial resolution and thus our eyes expect to see more *details* in the picture. For most typical pictures, the information contained in the subsampled (down-sampled) image usually can provide clues about the lost components. The most obvious example is the high-contrast edges, which are modeled as step signals. Thus, if a segment of image is recognized as an edge, proper high frequency components can be added into the interpolated (lowpass) signals to enhance the edge shape. Therefore, non-linear and/or spatially variant interpolators designed based on this concept could potentially produce images with better perceptual quality than the conventional filtering methods. Our approach in this paper is another attempt to regenerate the *original* high resolution images based on our prior

This work was supported in part by National Science Council of R.O.C. under Grant NSC86-2213-E-009-064.

1047-3203/97 $25.00
Copyright © 1997 by Academic Press
All rights of reproduction in any form reserved.

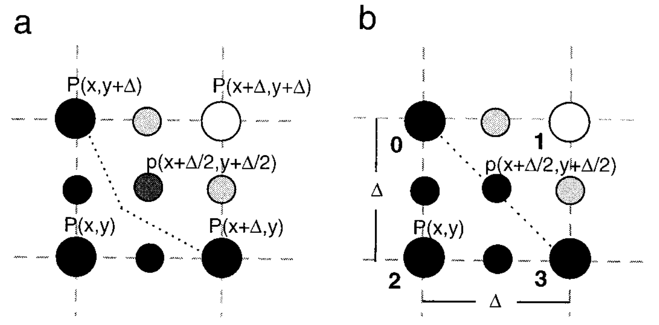


FIG. 1. Image interpolation results of (a) linear separable operations and (b) nonlinear operation using diagonal correlation on a quadrant set.

knowledge of *typical* images. Thus, similar to many other model-based approaches [8–12] our method contains essentially two elements [13]: (i) determine the image local characteristics (flat regions, edges, etc.) and (ii) generate the interpolated pixels by properly weighted averaging. However, there are many different ways of utilizing the image local characteristics for interpolation purpose. Also, the exact procedure used in synthesizing the interpolated pixels has a strong impact on the final performance. Our method described below seems to be able to produce very promising pictures particularly on the *difficult* parts of an image such as the *highly oblique edges*.[1]

Furthermore, many existing interpolation methods are separable operators along two axes. Although separable operators are simple in implementation, they often result in jaggedness, particularly, along the diagonal edges. Figure 1 illustrates the potential advantage of using diagonal correlations. Figure 1a shows the interpolation result generated by a separable bilinear interpolation filter. It is clear that the interpolated pixel intensity $p(x + \Delta/2, y + \Delta/2) = \frac{1}{4}(P(x, y) + P(x + \Delta, y) + P(x, y + \Delta) + P(x + \Delta, y + \Delta))$ is not the most desirable value, where $P(x, y)$, $P(x + \Delta, y)$, $P(x, y + \Delta)$, and $P(x + \Delta, y + \Delta)$ are the four given pixel intensities. If we consider the diagonal correlation, since three out of four given pixels have close intensity values, it seems more natural to set the interpolated center pixel the same as the three dominated pixels as shown in Fig. 1b. Again, this more *desirable* result is judged based on our *prior* knowledge of typical pictures that high-contrast edges are often contiguous.

Unlike the ideal case shown in Fig. 1, the captured natural images usually contain camera noise. Also, the surfaces of flat objects do not produce exactly the same pixel values when captured by a camera. How to compute the directional correlations of the original pixels from the sampled pixels and use them properly to generate the interpolated pixels is one of the key issues in this paper. Six directional classes are defined on the *quadrant* (*pixel*) *set* which is composed of the four neighboring pixels on a square. According to its local gradient magnitudes, each quadrant set is classified into one of the six classes. Furthermore, a fuzzy inference method is devised to synthesize the interpolated pixels with reduced blurring. According to its class, either the bilinear or the fuzzy method is applied to a quadrant set. The switch between the fuzzy and the bilinear interpolation can preserve the edge contrast while at the same time keeping the computational complexity at a reasonably low level because the bilinear interpolation costs much less in computation but is frequently used in a typical image with large smooth regions. This is our first contribution.[2]

Because edges of an image convey the most essential perceptible information to the human eyes,[3] it is important to preserve the edge integrity and to enhance edge sharpness which is blurred due to the lost high frequency components in the subsampling process. In the previous studies [10, 14, 15], local edge detectors are often used to extract the edges. However, it is observed [13] that the highly oblique edges in natural images often cannot be correctly identified by a local edge detector. Because the fuzzy based interpolation method described in the last paragraph uses only the local information around four pixels, it cannot trace the highly oblique edges. In order to identify the orientations of these edges, a segment matching method is proposed in this paper. This method detects the edge orientation by shifting a segment of edge profile and matching it against the candidate segment in its neighborhood. After the edge orientation is accurately identified, pixels parallel to the edge orientation can be interpolated with little distortion. Jaggy artifacts of the interpolated images can thus be significantly reduced. This is our second contribution.[2]

This paper is organized as follows. An edge-preserving interpolation method based on fuzzy inference is described in Section 2. In order to efficiently apply this edge-preserving method to images, a classification and interpolation scheme is proposed in Section 3. Because it is difficult to interpolate the highly oblique edges with local edge detector only, a segment-matching method is suggested in Section 4. With the help of screening criteria, this method locates the highly oblique edges quite successfully. Simulation results of the proposed methods and some other approaches are shown in Section 5, and Section 6 concludes this paper.

2. FUZZY-INFERENCE BASED INTERPOLATION

As described earlier that the second key element in the modern model-based interpolation is the method of

[1] The edge orientation has a very sharp angle against either the horizontal or the vertical axes.

[2] An early version of our method is summarized in [13].

[3] It is reported [16] that the very first step in human perceptual system is to convert images into nearly edge-only signals.

synthesizing the interpolated pixels. Typically, a weighted neighboring pixel interpolator with spatially varying weighting factors is used [8, 9, 17, 18]. In these approaches, the weighting factors are either preselected sets of values or ordinary arithmetic functions of image local features. From time to time, their results do not match the human subjective expectation. Hence, we propose an interpolator based on fuzzy inference. This method preserves edges and it is almost scale invariant. Another advantage is that it is easy to extend this method to the multidimensional cases without the drawback of the separable operators.

2.1. Spatially Adaptive Interpolation Based on Local Gradients

To evaluate the performance of different interpolation methods, we assume that the original signal $f(x)$ is continuous. Let $f(x + n\Delta)$ be the given samples, where Δ is the sampling interval. Our goal here is to find the interpolated value of $f(x + \Delta/2)$ based on the information at locations x and $(x + \Delta)$. The Taylor series expansion at $(x + \Delta/2)$ gives

$$f\left(x+\frac{\Delta}{2}\right) = f(x) + f'(x)\frac{\Delta}{2} + f''(x)\frac{\Delta^2}{8} + R_1, \tag{1}$$

where R_1 represents all the remaining higher order terms (similar definition for R_2 in the following equation). Similarly, we could express $f(x + \Delta/2)$ using Taylor series expansion at $(x + \Delta)$; that is,

$$f\left(x+\frac{\Delta}{2}\right) = f(x+\Delta) - f'(x+\Delta)\frac{\Delta}{2} + f''(x+\Delta)\frac{\Delta^2}{8} + R_2. \tag{2}$$

Averaging (1) and (2), we obtain

$$f\left(x+\frac{\Delta}{2}\right) = \frac{1}{2}\left[f(x) + f(x+\Delta) + \frac{f'(x) - f'(x+\Delta)}{2}\Delta + \frac{f''(x) + f''(x+\Delta)}{8}\Delta^2 + R_1 + R_2\right]. \tag{3}$$

It is ready to see that the ordinary bilinear interpolation can be deduced from (3) by assuming that the first-order differential of $f(x)$ is constant and the second and higher order terms are negligible for all x. Though only the middle point is expressed in (3), expressions for all the points between the given samples can be derived similarly. If the second-order differential becomes significant and is assumed to be constant, the cubic spline interpolation with parameter $a = -0.5$ is then derived as described in Parker *et al.* [5]. In the cubic spline interpolation, the first-order differential is approximated by $(f(x + \Delta) - f(x))/\Delta$ and the second-order differential by $(f'(x + \Delta) - f'(x))/\Delta$. As we can expect heuristically the interpolation error is reduced if the second-order differential, i.e., the slope variation, is included. From the signal processing viewpoint, we are designing a higher order filter to approximate the ideal lowpass filter.

It seems that including additional higher order terms into our interpolator would improve the interpolation performance. However, this direction has its limit because we are given only the samples on the discrete grids. Approximating differentials using differences at the known samples has a limited accuracy and this approximation error increases as the differentiation order becomes higher. Another problem is that in the high contrast areas high order filters may not produce a better subjective image quality. Hence, instead of exploring the higher order differences further, we propose a nonlinear interpolation method based on the first order differences.

The basic concept of our method is as follows. We know that the interpolated sample at the middle point can be approximated by (3). Assume that the differentials with orders higher than three are insignificant and the second-order differential is a fixed constant. First, we try to find the relationship between the to-be-interpolated sample and the local gradients at the given samples. If $|f'(x)| > |f'(x + \Delta)|$, we have either

$$f'(x) > |f'(x+\Delta)| \geq 0$$

or

$$f'(x) < -|f'(x+\Delta)| \leq 0. \tag{4}$$

In the first case, we have $f'(x) - f'(x + \Delta) > 0$. Since $f''(x)$ equals to $f''(x + \Delta)$ (our assumption) and we assume $f''(x)$ can be approximated by $(f'(x + \Delta) - f'(x))/\Delta$, we obtain

$$f\left(x+\frac{\Delta}{2}\right) = \frac{1}{2}\left[f(x) + f(x+\Delta) + \frac{f'(x) - f'(x+\Delta)}{4}\Delta\right]. \tag{5}$$

Because $f'(x) - f'(x + \Delta) > 0$, we conclude from (5) that

$$f\left(x+\frac{\Delta}{2}\right) > \frac{1}{2}[f(x) + f(x+\Delta)]. \tag{6}$$

Note that $f(x) < f(x + \Delta)$ because $f'(x) > 0$. Equation (6) shows that the middle interpolated sample should have a value closer to $f(x + \Delta)$, the sample with a smaller gradient. A similar observation can be derived for the second case in (4).

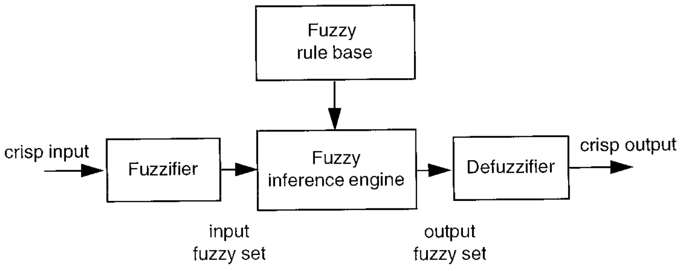


FIG. 2. Block diagram of a fuzzy logic system.

2.2. *Edge Preserving Interpolation Based on Fuzzy Inference*

We adopt the fuzzy inference [19, 20] in our interpolation process. The interpolated sample at $x + \Delta/2$ is produced by

$$f\left(x + \frac{\Delta}{2}\right) = w(|f'(x)|) \cdot f(x) + w(|f'(x+\Delta)|) \cdot f(x+\Delta), \tag{7}$$

in which

$$w(|f'(x)|) + w(|f'(x+\Delta)|) = 1 \tag{8}$$

with $w(|f'(x)|), w(|f'(x+\Delta)|) \geq 0$. Based on the reasoning in the Section 2.1, $w(\cdot)$ should be a nonincreasing function; that is, $w(a)$ decreases as a increases. Here we adopt the fuzzy logic to implement (7) instead of defining an arithmetic weighting function $w(\cdot)$ explicitly. The fuzzy method is chosen due to two major reasons. First, it can convey the concept in (6) in a way closer to human inference and thus produce better visual quality pictures (e.g., sharper edges) than the conventional methods. Second, by properly selecting the membership functions, the 1D fuzzy method can be extended directly to the 2D case for image applications.

In its general form, a fuzzy logic system consists of four basic components: fuzzy rule base, fuzzifier, fuzzy inference engine, and defuzzifier. Figure 2 shows the block diagram of a typical fuzzy logic system. In our application, a fuzzy rule base is a collection of IF–THEN rules that have the following structure:

$$\textit{Rule}(k) : \textit{IF } r_1 \textit{ is } F_{k1} \textit{ and } \cdots \textit{ and } r_n \textit{ is } F_{kn} \textit{ THEN } z \textit{ is } G_k, \tag{9}$$

where k is the rule index; F_{ki} and G_k are fuzzy sets defined on their universes of discourse $U_i \subset R$ (e.g., domain of pixel location) and $V \subset R$ (e.g., domain of pixel intensity), respectively. In this application, R is the real axis, and $\boldsymbol{r} = (r_1, \ldots, r_n)^T \in U_1 \times \cdots \times U_n$, where r_i denotes the ith input linguistic variable (e.g., coordinate of the to-be-interpolated pixel) and $z \in V$ is the output linguistic variable (e.g., distribution of possible interpolated pixel value). To convert (7) to a fuzzy logic system, a simple fuzzy rule base with two rules is constructed as follows:

$$\begin{aligned} &\textit{Rule}(1) : \textit{IF } r_1 \textit{ is proximate to } x \\ &\qquad\qquad \textit{THEN } f(r_1) \textit{ is } f(x) \\ &\textit{Rule}(2) : \textit{IF } r_1 \textit{ is proximate to } x + \Delta \\ &\qquad\qquad \textit{THEN } f(r_1) \textit{ is } f(x+\Delta). \end{aligned} \tag{10}$$

The meaning of "*proximate to*" and its implication in this inference is the key to the success of our scheme. In our usage, "r_1 is *proximate to x*" means that the pixel at location r_1 should have a value close to the pixel at x based on the image local characteristics around r_1 and x. The degree of "r_1 is *proximate to x*" depends on both the shape of the membership function defined around x and the distance between r_1 and x. The discussion at the end of Section 2.1 suggests that if $r_1 = x + \Delta/2$ (r_1 is equally distant from x and $x + \Delta$) and the gradient of $f(x)$ is smaller than that of $f(x + \Delta)$, then the degree of "r_1 is *proximate to x*" will be higher than that of "r_1 is *proximate to x* + Δ." This is the principle behind our design of the gradient dependent membership functions. Next, we like to choose the shape of fuzzy membership functions that would properly convey this *proximity* concept; this choice is often subjective and application dependent. Because, in our simulations, the Gaussian membership function demonstrates a better performance than the triangular or the trapezoidal functions in preserving edge contrast, it is chosen as the antecedent part in (10),

$$\mu_{F_{ki}}(r_i) = \exp\left[-\left(\frac{r_i - \bar{r}_{ki}}{\sigma_{ki}}\right)^2\right], \quad k = 1, 2, \tag{11}$$

where k is the rule index and $\bar{r}_{ki}$ and σ_{ki} are adjustable parameters. Note that $i = 1$ for 1D input signals. In this application, $\bar{r}_{11} = x$ and $\bar{r}_{21} = x + \Delta$. According to the observations at the end of Section 2.1, we make σ_{k1} a function of the absolute value of the "gradient" (first-order difference). We assume a simple relation between σ_{k1} and the gradient $D(\cdot)$,

$$\sigma_{k1} = \alpha \cdot D(\bar{r}_{k1}) + \beta, \tag{12}$$

where α and β are fixed parameters. $D(\bar{r}_{k1})$ is the normalized first-order difference,

$$D(\bar{r}_{k1}) \equiv \frac{|f'(\bar{r}_{k1})|}{f'_{max}}, \tag{13}$$

where $f'_{\max}$ is the absolute value of the maximum possible first-order difference. It is determined by the allowable range of $f(\cdot)$; for example, $f'_{\max}$ is 255 for 8-bit gray level images. Note that the α and β values are chosen such that σ_{k1} is small when $D(\bar{r}_{k1})$ is large. Therefore, α is negative and $\beta > |\alpha|$. We will show, by examples, our choice of their values and their impact on interpolation.

A fuzzy inference engine is an operator that maps the input fuzzy sets in $U = U_1 \times \cdots \times U_n$ to the output fuzzy sets in V according to the IF–THEN rules in the rule base. Let A be the input fuzzy set in U and B_k be the output fuzzy set produced by *Rule*(k) in (9). The output membership function is described by

$$\mu_{B_k}(z) = \sup_{\boldsymbol{r}\in U}[\mu_{F_{k1}\times\cdots\times F_{kn}\to G_k}(\boldsymbol{r}, z) * \mu_A(\boldsymbol{r})]. \tag{14}$$

The fuzzy reasoning $F_{k1} \times \cdots \times F_{kn} \to G_k$ can be defined in several ways. Here we adopt the product-operation rule [21, 22] for it fulfills our requirement in (7). That is,

$$\mu_{F_{k1}\times\cdots\times F_{kn}\to G_k}(\boldsymbol{r}, z) = \prod_{i=1}^{n} \mu_{F_{ki}}(\boldsymbol{r}) \cdot \mu_{G_k}(z). \tag{15}$$

And $*$ is chosen to be the algebraic product operator which is one type of the t-norm intersection operations [22].

As described earlier, the input to a fuzzy logic system should be a fuzzy set. In the case of interpolation, the input is a sample location, a crisp point. We need to map this crisp input to a fuzzy set. The singleton fuzzifier is thus used. It maps a crisp point a into a fuzzy set A with the property: $\mu_A(x) = 1$ for $x = a$ and $\mu_A(x) = 0$ for all other x belonging to U, where U is the universe of discourse. At the end of this inference process, a defuzzifier maps the output fuzzy set back to a crisp point, which is an interpolated pixel in our case. In this paper, we choose a center averaging defuzzifier defined by

$$z = \frac{\sum_{k=1}^{N} \bar{z}_k \cdot \mu_{B_k}(\bar{z}_k)}{\sum_{k=1}^{N} \cdot \mu_{B_k}(\bar{z}_k)}, \tag{16}$$

where N is the number of rules and $\bar{z}_k$ represents the point at which the fuzzy membership function μ_{B_k} achieves its maximum value. Note that the structure of (16) matches the requirements of (7) and (8). Figure 3 shows an example of the inference process for the one-dimensional case. The fuzzy sets in the consequent part are set to be singleton fuzzy sets in this case. In this example, the gradient magnitude $|f'(x + \Delta)|$ is smaller than $|f'(x)|$ because the width of the membership function $\mu_{F_{21}}$ is larger than that of $\mu_{F_{11}}$. The inference process is indicated by the dashed lines in this figure. We can conclude from the output fuzzy set μ_B that the interpolated sample at $x + \Delta/2$ should have a value closer $f(x + \Delta)$, the one with a smaller gradient magnitude.

We compare the step response of this method with those of the conventional interpolation methods in Fig. 4, an example of factor-of-8 interpolation. We apply the fuzzy interpolation process three times to produce the final interpolated signal; the input signal at each step is zoomed only by a factor of 2. Different from the shift invariant linear interpolator, the fuzzy interpolator is nonlinear and local signal dependent. Our experiments indicate that the iterative interpolation process described in the above produces better results than inserting seven interpolated points between two given samples directly. It can be seen from this figure that the high-contrast intensity jump is preserved by the fuzzy inference method. The parameters α and β in (12) control the shape of the Gaussian membership function and thus have an impact on the final results. When properly selected, different (α, β) sets produce similar results as shown in this example. For a zooming factor of 2, if α is set to zero, the interpolation results of the fuzzy method would be the same as those of the bilinear method. Alternatively, if the shape of the membership function is chosen to be triangular and the width is intentionally set to $2 \cdot \Delta$, the fuzzy method would also degenerate into the bilinear method.

3. DIGITAL IMAGE INTERPOLATION BASED ON FUZZY INFERENCE

We now extend the one-dimensional (1D) interpolation method described in the previous section to the two-dimensional (2D) case. Applying the fuzzy inference method to every to-be-interpolated sample point is costly in computation, because the fuzzy inference process requires many more calculations than the bilinear interpolator. Instead, we analyze the image local characteristics and apply the fuzzy interpolator only to the points that cannot be handled well by the bilinear interpolator. For example, in the smooth image regions, nearly no difference can be found between the interpolated samples generated by the bilinear and the fuzzy methods. Using the bilinear interpolation method in these regions can cut the number of computations drastically because typical images contain large smooth regions.

Our fuzzy image interpolation method consists of two steps: (1) classify a quadrant set based on its gradient magnitudes; (2) apply either the fuzzy or the bilinear interpolation to a quadrant set according to its assigned class.

3.1. *Quadrant Set Classification*

As mentioned earlier, a quadrant (pixel) set P is a set of four nearby pixels located on a square. We first calculate four directional gradients using the horizontal (H), vertical (V), diagonal (D), and anti-diagonal (A) 3×3 compass operators [23]:

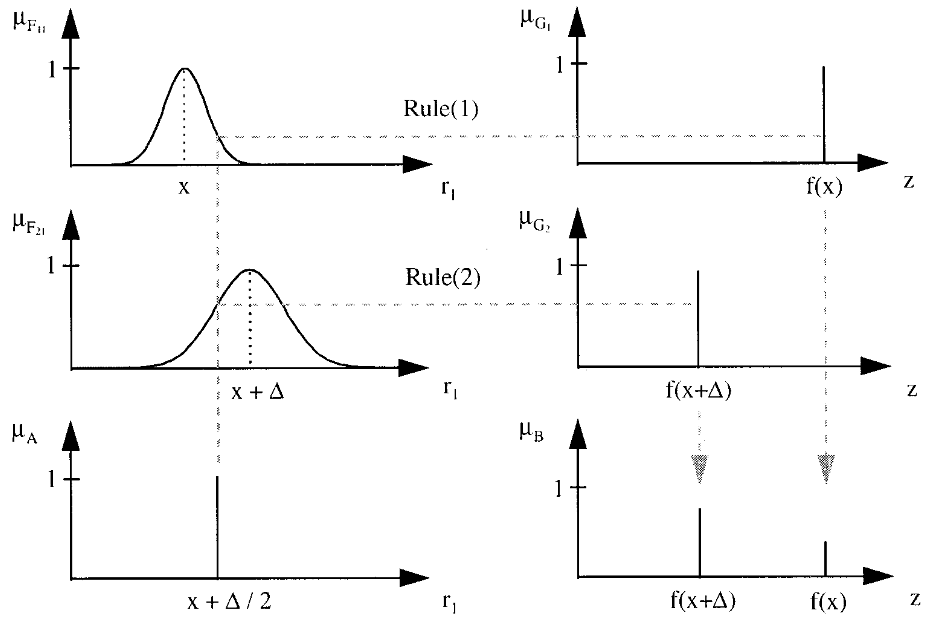


FIG. 3. An example of the fuzzy inference used for interpolation.

$$(H)\begin{bmatrix}-1 & 0 & 1\\ -1 & 0 & 1\\ -1 & 0 & 1\end{bmatrix}, \qquad (V)\begin{bmatrix}-1 & -1 & -1\\ 0 & 0 & 0\\ 1 & 1 & 1\end{bmatrix}$$

$$(D)\begin{bmatrix}-1 & -1 & 0\\ -1 & 0 & 1\\ 0 & 1 & 1\end{bmatrix}, \qquad (A)\begin{bmatrix}0 & -1 & -1\\ 1 & 0 & -1\\ 1 & 1 & 0\end{bmatrix}.$$

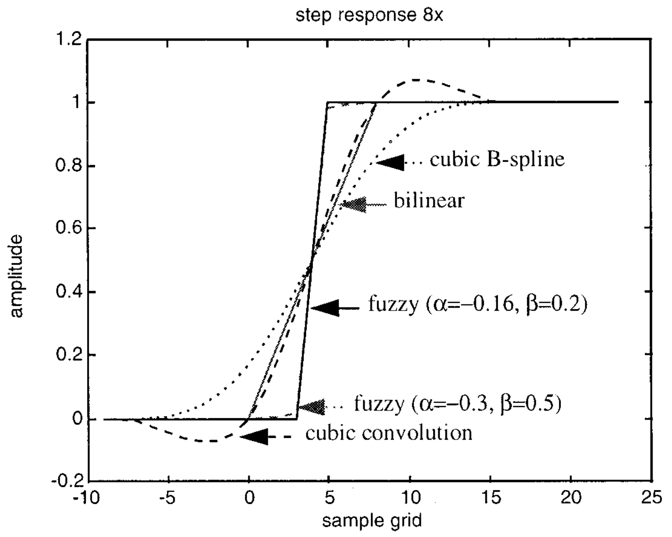


FIG. 4. Step response comparison of different interpolation methods.

These four directional gradients for each sample grid point in the quadrant set P are denoted by $g_H(i)$, $g_V(i)$, $g_D(i)$, and $g_A(i)$, where $i = 0, 1, 2, 3$ represents the four pixel locations shown in Fig. 1b. Hereafter, we use these directional gradients to classify the quadrant set. The classification algorithm is described below:

```
If $|g_H(0)| < T_s$ and $|g_V(0)| < T_s$ and $|g_D(0)| < T_s$
        and $|g_A(0)| < T_s$
        then $P$ is *smooth*
else if $|g_V(0)| > \lambda \cdot |g_H(0)|$ and $|g_V(1)| > \lambda \cdot |g_H(1)|$
        then $P$ is *horizontal*
else if $|g_H(0)| > \lambda \cdot |g_V(0)|$ and $|g_H(2)| > \lambda \cdot |g_V(2)|$
        then $P$ is *vertical*
else if $|g_A(0)| > \lambda \cdot |g_D(0)|$ and $|g_A(3)| > \lambda \cdot |g_D(3)|$
        then $P$ is *diagonal*
else if $|g_D(1)| > \lambda \cdot |g_A(1)|$ and $|g_D(2)| > \lambda \cdot |g_A(2)|$
        then $P$ is *anti-diagonal*
else $P$ is *rugged.*
```

The threshold value T_s affects the percentage of the quadrant sets (in an image) that are classified into the smooth class. The multiplicative parameter λ has an influence on the decision of the quadrant set orientation. The quadrant sets that do not satisfy the smooth and the directional orientation criteria are assigned to be the rugged class. To see the effects of T_s and λ, we apply this algorithm with different parameter values to three 256×256 images: Lena, Baboon, and Pepper. The results are shown in Table 1. Obviously, increasing the threshold value T_s increases

TABLE 1
Percentages of Classes of Three Test Images with (1) $T_s = 40$ and $\lambda = 2$, (2) $T_s = 80$ and $\lambda = 2$, (3) $T_s = 40$ and $\lambda = 1$

	Lena			Baboon			Pepper		
Image	(1)	(2)	(3)	(1)	(2)	(3)	(1)	(2)	(3)
Smooth	63.7	79.4	63.7	28.2	56.1	28.2	65.6	81.1	65.6
Horizontal	3.0	1.7	7.0	13.0	9.9	25.9	6.6	4.0	11.7
Vertical	15.4	9.3	22.5	11.5	7.4	23.4	10.5	6.4	17.1
Diagonal	6.8	4.1	3.2	6.6	4.1	6.4	5.1	3.2	2.3
Anti-diagonal	2.5	1.5	1.4	5.8	3.4	5.3	3.5	1.6	1.5
Rugged	8.6	4.0	2.2	34.9	19.0	10.8	8.6	3.7	1.8

the percentage of the smooth set. Decreasing the multiplicative parameter λ increases the percentage of the directional sets. Proper values of these parameters should be chosen to keep a balance between reducing computations and maintaining good interpolation results. For these test images, we found that $T_s = 40$ and $\lambda = 2$ can produce satisfactory results.

3.2. *Quadrant Set Interpolation*

According to its assigned class a quadrant set is interpolated in different manner as defined by Fig. 5. The lines indicate which original pixels are used to generate the interpolated pixel in-between. The interpolated pixel on the dashed lines is produced after all its reference neighbors have been computed. The triangle in Fig. 5 represents the pixel generated by the bilinear interpolator and the square denotes the pixel generated by the fuzzy interpolator. In general, the simple bilinear interpolation method is employed when the to-be-interpolated pixel is located parallel to the detected orientation; otherwise, the fuzzy method is applied to preserve the intensity contrast. The details of the interpolation scheme in each case (for a factor-of-2 in each axis) are described below.

Smooth Class. As shown in Fig. 5a, the three to-be-interpolated pixels are all interpolated by bilinear interpolation. In the smooth regions, using either the bilinear or the fuzzy methods does not make a noticeable difference; hence, the bilinear method is used to save computation.

Horizontal and Vertical Classes. The intensity variation along the detected edge orientation is small for the directional quadrant sets. The to-be-interpolated pixel located on the detected orientation line can thus be well interpolated by the bilinear method using the neighboring pixels along the same direction. As shown in Fig. 5b, the upper triangle pixel is produced by bilinear interpolation using its two horizontal neighboring pixels. For the three to-be-interpolated pixels located on the center line, no

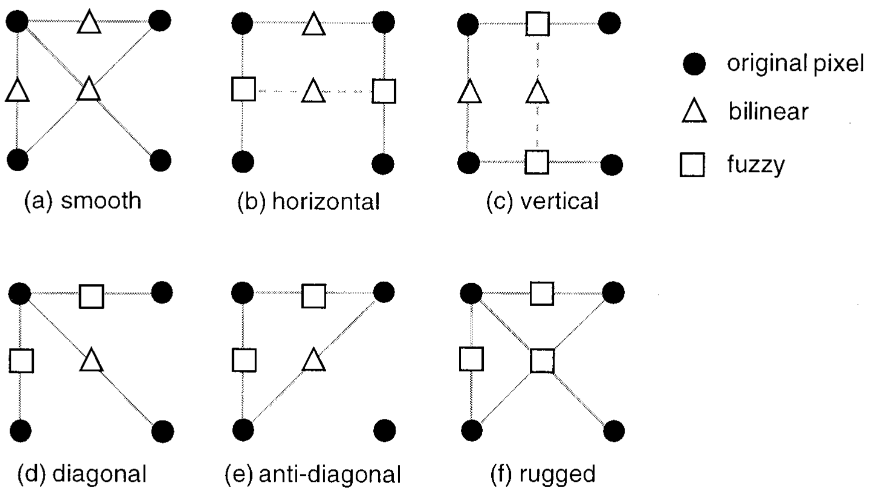


FIG. 5. Quadrant set interpolation schemes for (a) smooth, (b) horizontal, (c) vertical, (d) diagonal, (e) anti-diagonal, and (f) rugged quadrant set classes.

original pixel is available for interpolation along the detected direction. To preserve the intensity contrast, we first apply the 1D fuzzy inference method to generate the two square pixels in Fig. 5b. Then the bilinear interpolation is applied to generate the center pixel. The horizontal edge characteristics is thus well preserved. Similarly, the vertical class quadrant sets are interpolated as indicated in Fig. 5c.

Diagonal and Anti-diagonal Classes. Based on the same reasoning given in the preceding paragraph, the center pixel in Fig. 5d is produced by the bilinear interpolator using the two diagonal original pixels. On the other hand, there is no original pixel available along the diagonal direction to produce the square pixels. Thus, the 1D fuzzy method is applied to preserve the intensity contrast. The anti-diagonal case is similarly derived as shown in Fig. 5e.

Rugged Class. For those quadrant sets that are identified as the rugged class, all the to-be-interpolated pixels are generated by the fuzzy inference method. The two interpolated pixels located on the upper line and the left line are generated by the 1D fuzzy method as before. Because of the low correlation among the four original pixels, the center pixel is generated using all the four pixels located on the two-dimensional grid as shown in Fig. 5f. The 1D fuzzy interpolation method now needs to be extended to 2D space. The rule base of the 2D fuzzy interpolation method consists of the following four rules:

$$\begin{aligned}
&\textit{Rule}(1): \textit{IF } (r_1, r_2) \textit{ is proximate to } (x, y) \\
&\qquad\qquad \textit{THEN } p(r_1, r_2) \textit{ is } P(x, y); \\
&\textit{Rule}(2): \textit{IF } (r_1, r_2) \textit{ is proximate to } (x+\Delta, y) \\
&\qquad\qquad \textit{THEN } p(r_1, r_2) \textit{ is } P(x+\Delta, y); \\
&\textit{Rule}(3): \textit{IF } (r_1, r_2) \textit{ is proximate to } (x, y+\Delta) \\
&\qquad\qquad \textit{THEN } p(r_1, r_2) \textit{ is } P(x, y+\Delta); \\
&\textit{Rule}(4): \textit{IF } (r_1, r_2) \textit{ is proximate to } (x+\Delta, y+\Delta) \\
&\qquad\qquad \textit{THEN } p(r_1, r_2) \textit{ is } P(x+\Delta, y+\Delta).
\end{aligned} \tag{17}$$

A fuzzy inference engine similar to that in Fig. 3 can thus be constructed. In determining the σ_{ki} of the membership functions for the two antecedent parts, instead of setting them separately, we now have

$$\sigma_k = \alpha \cdot D(\bar{r}_{k1}, \bar{r}_{k2}) + \beta, \tag{18}$$

where k is the rule index and $D(\bar{r}_{k1}, \bar{r}_{k2})$ is the *average* of the two directional gradient magnitudes at $P(\bar{r}_{k1}, \bar{r}_{k2})$ along the horizontal and the vertical axes. Similar to the 1D case, the product-operation rule is adopted in the fuzzy

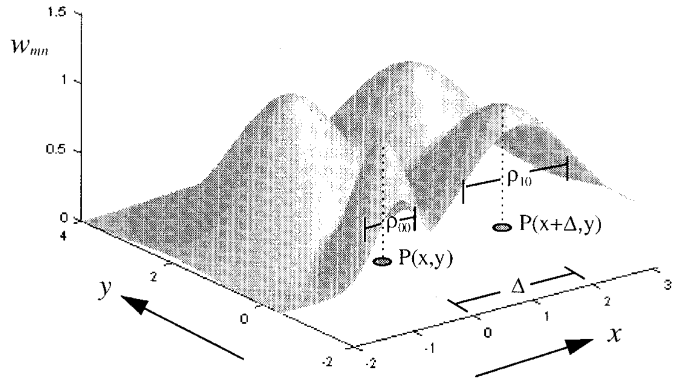


FIG. 6. Two-dimensional Guassian shape membership functions.

inference. A graphical display of the 2D fuzzy membership functions is shown in Fig. 6.

Based on the choices we have made earlier, the interpolated pixel $p(x+\delta_x, y+\delta_y)$ is computed by the defuzzification process [22],

$$p(x+\delta_x, y+\delta_y) = \frac{\sum_{m=0}^{1}\sum_{n=0}^{1} P(x+m\Delta, y+n\Delta)\cdot w_{mn}(\delta_x, \delta_y)}{\sum_{m=0}^{1}\sum_{n=0}^{1} w_{mn}(\delta_x, \delta_y)}, \quad 0 \le \delta_x, \delta_y \le \Delta, \tag{19}$$

where $P(x+m\Delta, y+n\Delta)$ is the given pixel intensity value on the sampling grid before interpolation. For the center pixel in the quadrant set, $\delta_x = \frac{1}{2}\Delta$ and $\delta_y = \frac{1}{2}\Delta$. The function $w_{mn}(\delta_x, \delta_y)$ is a location-dependent 2D Gaussian membership function

$$w_{mn}(\delta_x, \delta_y) = \exp\left(-\frac{(\delta_x - m\Delta)^2 + (\delta_y - n\Delta)^2}{\rho_{mn}^2}\right), \quad m, n = 0, 1, \tag{20}$$

where $\rho_{00} = \sigma_1$, $\rho_{10} = \sigma_2$, $\rho_{01} = \sigma_3$, and $\rho_{11} = \sigma_4$; σ_k is derived from (18).

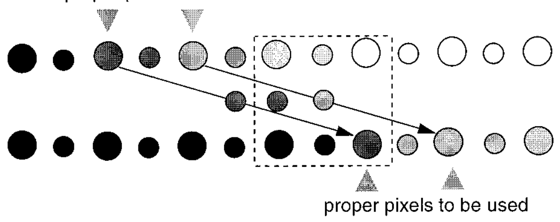


FIG. 7. Limitation of local operations for interpolating on highly oblique edges.

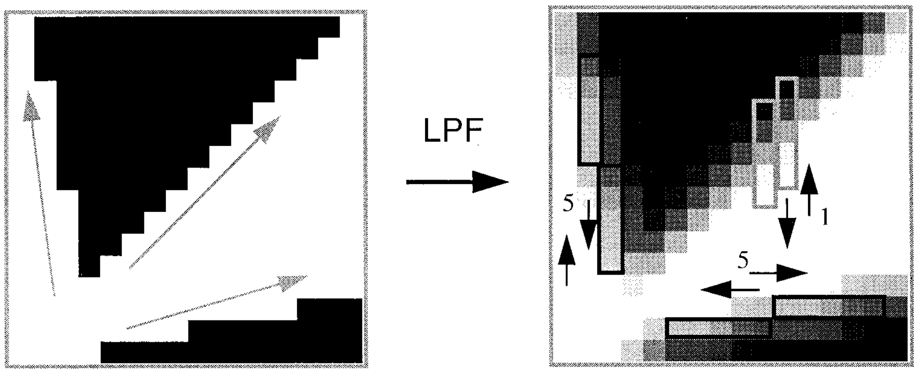


FIG. 8. Edge shadowing effect due to low pass filtering.

4. SEGMENT MATCHING METHOD FOR IDENTIFYING EDGE ORIENTATION

Although the proposed interpolation method based on fuzzy inference preserves reasonably well the edges in the horizontal, vertical, and diagonal directions, it cannot retain the edge integrity of the highly oblique edges, edges with sharp angles. Because only four neighboring pixels are used in computing the interpolated pixels (as shown in (19)) and a square region of 4×4 pixels are involved in computing the four gradients (when the 3×3 gradient operators are in use), we do not have sufficient information to determine the precise edge orientation; therefore, highly oblique edges are often approximated by the horizontal or vertical edges. For the same reason, the ordinary small size edge detectors such as the 3×3 Sobel operator fail to identify the correct edge orientations of the highly oblique edges. An example is illustrated by Fig. 7, where the large circles are the given original pixels and the small circles are the interpolated pixels. Ideally, the middle line pixels enclosed by the dashed-line box should be interpolated using the four pixels marked by the triangles. However, some of the marked pixels are outside the quadrant set. Hence, it is difficult, if not impossible, to synthesize those middle line pixels using only the original pixels inside the quadrant set. The challenge thus becomes how to detect the correct orientation of the highly oblique edges.

Examining the characteristics of the highly oblique edges in images, we develop a spatial domain-matching algorithm to identify the edge orientation. After the edge orientation is identified, the interpolated pixels can then be synthesized along its detected orientation. Combining this technique together with the fuzzy-inference based interpolator, highly oblique edges can be recreated without noticeable artifacts.

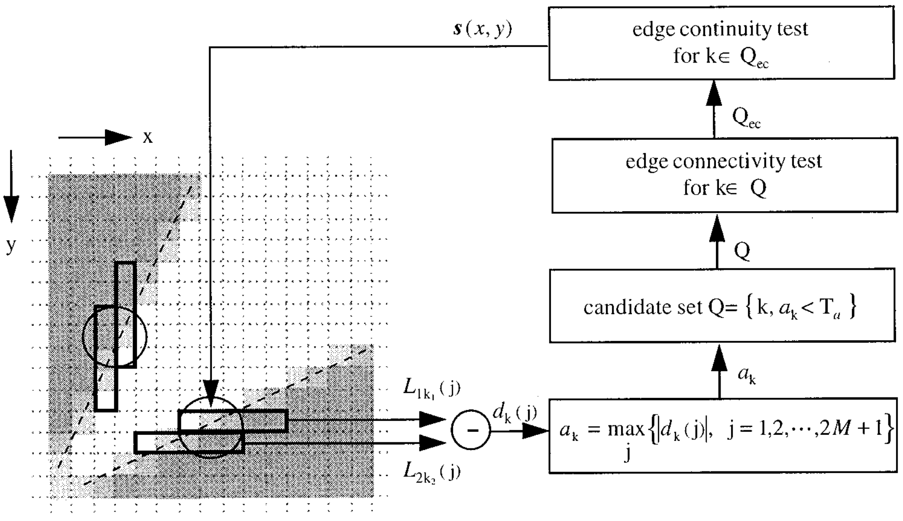
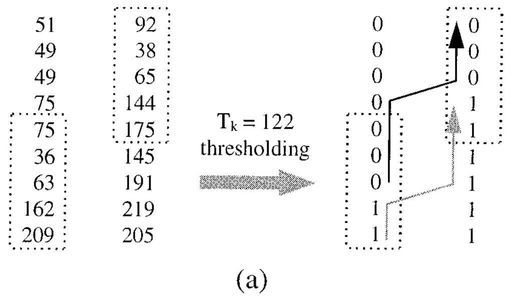

$s(x,y)$
edge continuity test for $k \in Q_{ec}$
Q_{ec}
edge connectivity test for $k \in Q$
Q
candidate set $Q = \{k, a_k < T_a\}$
a_k
$a_k = \max_j \{|d_k(j)|, \; j = 1,2,\cdots,2M+1\}$
$d_k(j)$
$L_{1k_1}(j)$
$L_{2k_2}(j)$
x
y

FIG. 9. Segment matching method.



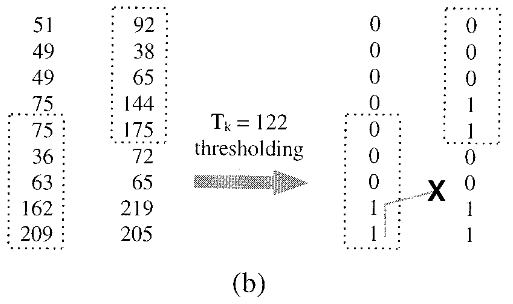


FIG. 10. Edge connectivity examples for (a) connected and (b) nonconnected matched pair of line segments.

4.1. *Edge Shadowing Effect due to Lowpass Filtering*

The sampling process is essential to generate digital images. An anti-aliasing lowpass filter [24] is usually applied to analog images before sampling to avoid aliasing. In addition, typical image sensing devices such as cameras also behave like lowpass filters. As a result, the original high-contrast edges (with intensity discontinuities) are blurred due to lowpass filtering.

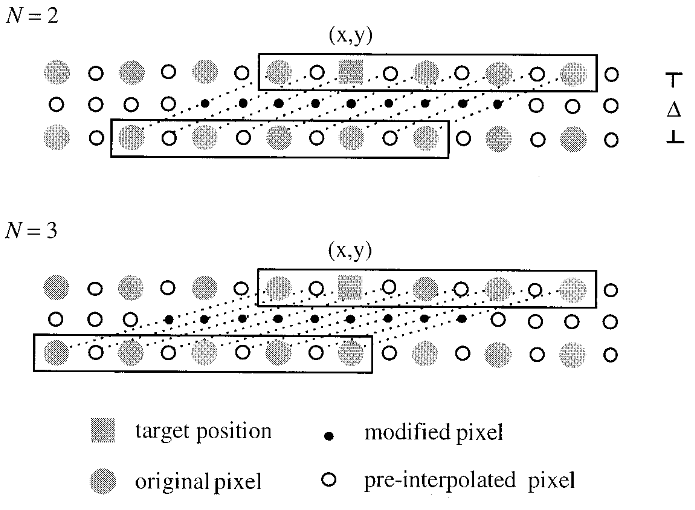


FIG. 11. Highly oblique edge interpolation.

TABLE 2
Parameters of our Scheme in Simulations

Classification		Fuzzy		Segment matching			
T_s	λ	α	β	M	K	T_g	T_a
40	2	−0.16	0.2	2	10	20	50

Figure 8 shows the smoothing effect due to low pass filtering on the high-contrast edges with different orientations. The right-hand figure also demonstrates an orientation identification method for the highly oblique edges. Edges are divided into two classes according to their orientations: the horizontally dominant and the vertically dominant classes. For example, in the left-hand figure of Fig. 8, the angle of the lower right edge is less than 45° and thus this edge is classified as the horizontally dominant edge. It is shown that two horizontal line segments 5 pixels apart have almost the same intensity profiles (enclosed by rectangular boxes). In our algorithm, we only examine the horizontal or the vertical profile of an edge. Another example is the upper left edge, whose angle is greater than 45° and thus it is a vertically dominant edge. The third edge in the middle is a diagonal edge with a nearly 45° angle and thus it is not a highly oblique edge. It can be processed by the method described in Section 3.

4.2. *Segment Shift Matching*

Most edge preserving interpolation methods consist of two steps: (i) edge detection and tracing and (ii) pixel interpolation along the orthogonal direction to the estimated edge with a contrast enhancement scheme [8, 9, 12, 25]. In the first step, highly oblique edges are usually difficult to detect using small-size edge detectors. Although an operator with a larger window size can be used [26, 27], the number of pixels involved increases significantly and thus the detection algorithm becomes very complicated. Therefore, we propose a simple yet effective edge orientation identification method for, particularly, the highly oblique edges.

First, we determine whether an edge exists or not and then identify the dominant orientation if an edge indeed exists. Local operators such as the smoothed gradient (Prewitt) or the Sobel gradient operators can be used to detect edges. However, for the goal of detecting highly oblique edges, we found the following two small-size local gradient operators are sufficient.

$$(h)\begin{bmatrix}0 & 0 & 0\\ 0 & -1 & 1\\ 0 & -1 & 1\end{bmatrix}, \qquad (v)\begin{bmatrix}0 & 0 & 0\\ 0 & -1 & -1\\ 0 & 1 & 1\end{bmatrix}.$$

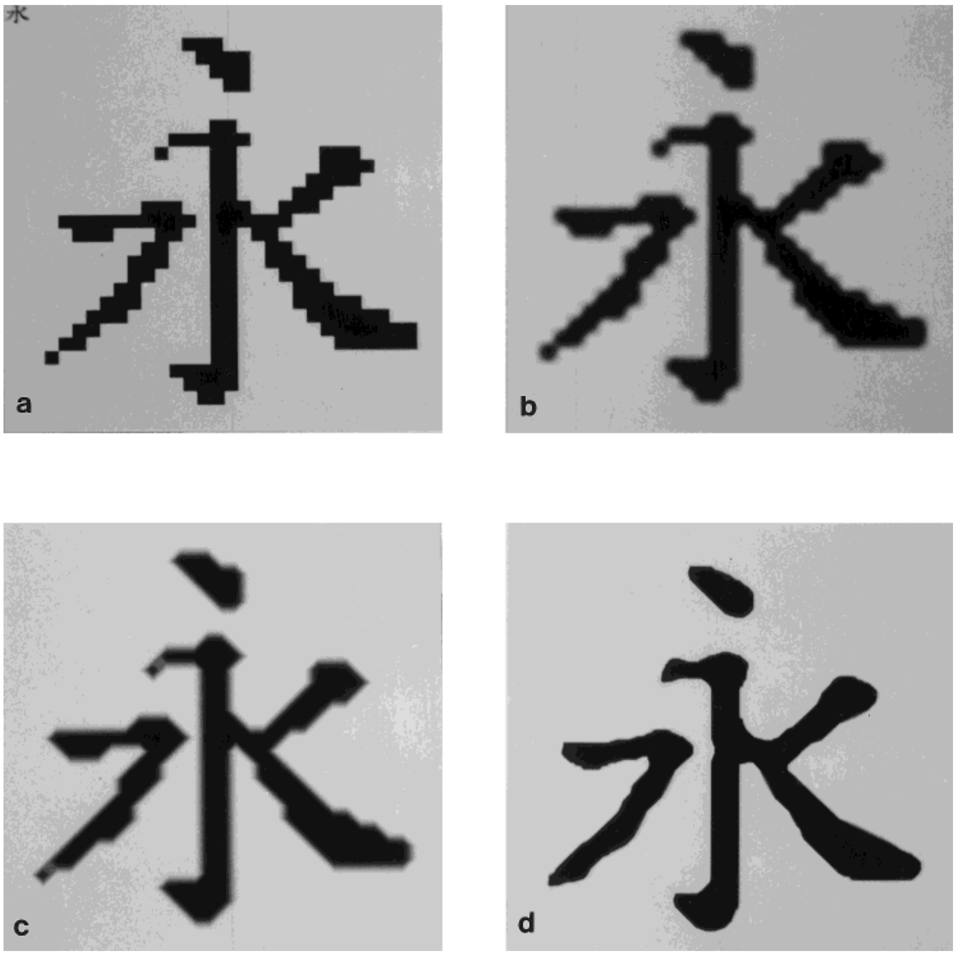


FIG. 12. "Chinese character," $z = 16$: (a) original image (32×32); (b) cubic spline; (c) median; and (d) our method.

We apply the above gradient operators to an image and calculate the gradient vector, $\boldsymbol{g}(x, y) = (g_h(x, y), g_v(x, y))$, $x, y \in Z$. An edge is detected if the gradient magnitude $|g_h(x, y)| > T_g$ or $|g_v(x, y)| > T_g$, where T_g is a preselected threshold value. We then determine the dominant edge orientation. If $|g_h(x, y)| > |g_v(x, y)|$, the edge is vertically dominant; otherwise, it is horizontally dominant.

In the second step, we try to identify the accurate edge orientation by shifting two line segments around the detected edge and compare their profiles. If an edge is horizontally dominant, we create two line segments: $L_{1k_1} = \{P(x + (k_1 - M)\Delta, y), P(x + (k_1 - M + 1)\Delta, y), \ldots, P(x + (k_1 + M)\Delta, y)\}$, and $L_{2k_2} = \{P(x + (-k_2 - M)\Delta, y + \Delta), P(x + (-k_2 - M + 1)\Delta, y + \Delta), \ldots, P(x + (-k_2 + M)\Delta, y + \Delta)\}$ in which k_1 and k_2 define the relative locations of these two segments and $P(x, y)$ denotes the intensity value of the pixel at location (x, y). The initial values of k_1 and k_2 are K_1 and K_2, respectively. These two parameters decide the search range K $(=K_1 + K_2)$ and M is a constant that determines the segment size, $2M + 1$. Similarly, vertical line segments can be created for vertically dominant edges. Let $L_{ik}(j)$ represent the jth component in the line segment L_{ik}. We first compute the line segment pointwise difference $d_{k=(k_1+k_2)}(j) = L_{1k_1}(j) - L_{2k_2}(j)$, and its maximum absolute value

$$a_k = \max_j \{|d_k(j)|, j = 1, 2, \ldots, 2M + 1\}. \qquad (21)$$

Although the sum of the squared difference is another possible measure for matching, absolute difference is much easier for computing and is less vulnerable to intensity offset (dc shift) which may occur on the rim of an edge. If $a_k < T_a$, we add the index $k = (k_1 + k_2)$ to the (matching) candidate set Q. Threshold T_a decides a match and its value is empirically selected. Repeat the above procedure by shifting the line segment one pixel inward from one side of an edge alternately; that is, one of k_1 or k_2 is decreased by one at each iteration.

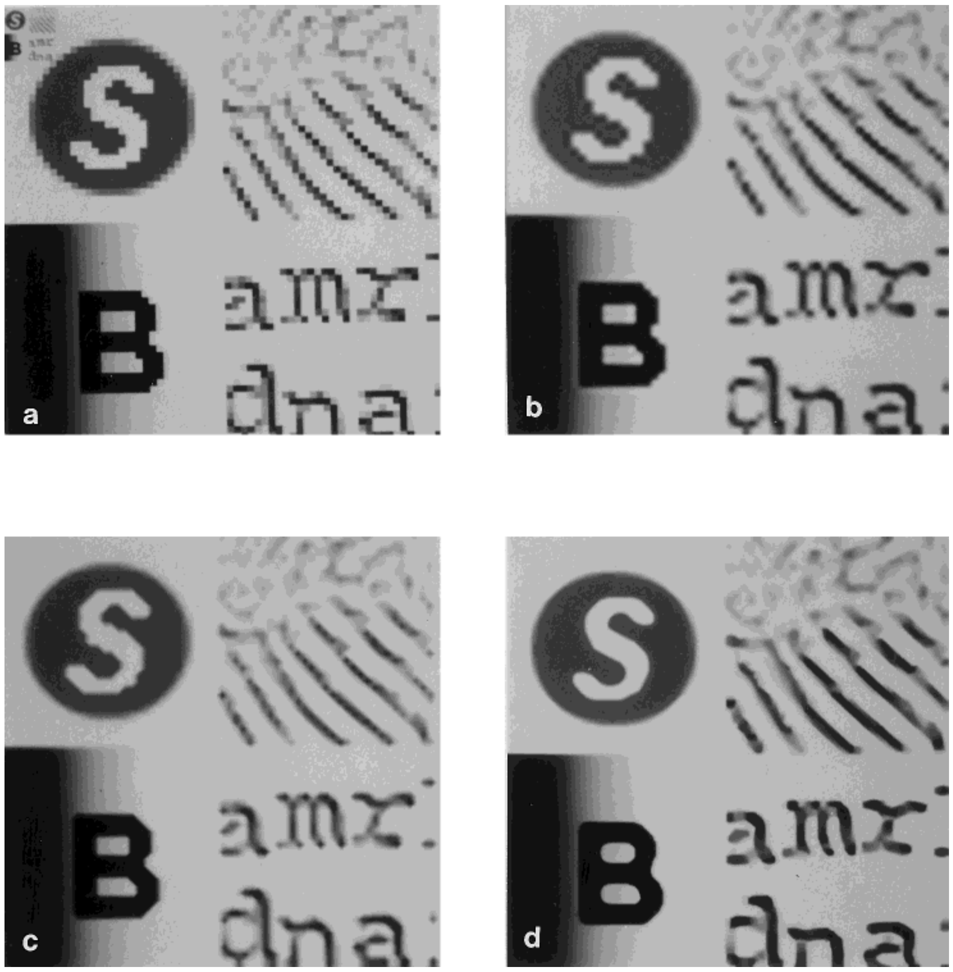


FIG. 13. "Shapes and text," $z = 8$: (a) original image (64×64); (b) cubic spline; (c) median; and (d) our method.

As shown in Fig. 9, after the candidate set Q is produced, two tests are designed to single out the best matching index in Q. First, based on our knowledge of image processing, if two neighboring line segments locate on the rim of the same edge, their pixel intensities are often contiguous without abrupt intensity change. The so-called edge *connectivity* test is thus devised to eliminate some fake matching line segments in Q. The surviving (passed the test) indices are collected in the set Q_{ec}. Second, an edge is usually longer than a few pixels; that is, the edge orientation should be continuous for several pixels. An edge *continuity* test is thus devised to further eliminate some more disqualified candidates in the set Q_{ec}. If the final Q_{ec} contains more than one element, the one with the smallest shift is then chosen to be the detected orientation vector, $\boldsymbol{s}(x, y)$. This vector is then used for interpolating the detected highly oblique edge. If no candidate can pass through both tests, we assume that this is not a highly oblique edge. The details of these two tests are described in the following subsections.

4.2.1. Edge Connectivity Test. To determine the correct edge orientation, we examine all the candidates in set Q. If Q is an empty set, it implies that no strong edge orientation presents. Otherwise, for every horizontally dominant edge candidate in Q, we create two arrays which contain all the pixels within the extends of the two horizontally matched segments. Similarly, two vertical arrays are formed if the edge is vertically dominant. Figure 10 is an example used to illustrate the edge connectivity test.

In Fig. 10a, two vertical pixel arrays are displayed on the left-hand side. The matched line segments are enclosed by dashed line boxes. To examine the connectivity, we first threshold the pixel values of these two arrays to extract their profile patterns. This threshold value is determined by the image data in these two line segments,

$$T_{k=(k_1+k_2)} = \tfrac{1}{2} \cdot \left(\max_{i,j}\{L_{ik_1}(j), L_{ik_2}(j)\} + \min_{i,j}\{L_{ik_1}(j), L_{ik_2}(j)\}\right), \tag{22}$$

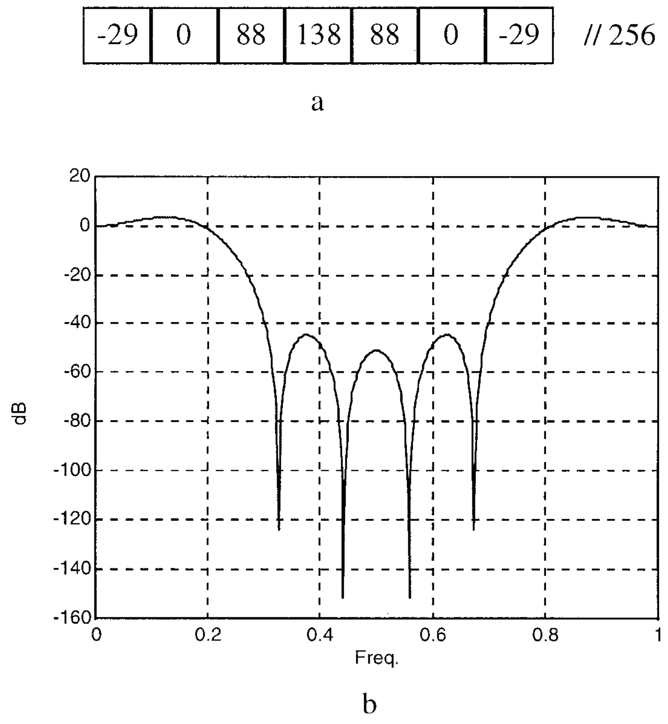


FIG. 14. Anti-aliasing lowpass FIR filter: (a) tap coefficients; (b) frequency response.

where $i = 1, 2$ and $j = 1, 2, \ldots, 2M + 1$. The extracted profile patterns are displayed on the right-hand side of Fig. 10a. Here, "1" indicates its pixel intensity is greater than T_k, and "0," otherwise. To determine whether the two matched line segments are connected, a simple graph tracing algorithm [28] is applied. The basic concept here is that the distribution of large and small pixels in both line segments should show similar patterns and one is roughly the shifted version of the other. In Fig. 10a, all the thresholded pixels in the line segments can be connected by the traces indicated by arrowed lines. Consequently, this edge candidate is included in the survival candidate set Q_{ec}, the collection of the edges passing through the connectivity test. Fig. 10b is an example that fails because there exists an intensity gap in these two pixel arrays, and thus it is removed from Q_{ec}.

TABLE 3
Percentages of Detected Highly Oblique Edges

Image (256 × 256)	Lena	Baboon	Pepper
Horizontally dominant	2.7% (1773)	5.6% (3671)	2.4% (1571)
Vertically dominant	5.5% (3576)	5.3% (3498)	3.2% (2073)

TABLE 4
Mean Squared Errors of the Interpolated Images

Image	Lena	Baboon	Pepper
Pixel replication	104.3	472.1	136.2
Bilinear	26.6	289.8	46.3
Cubic spline	25.5	277.0	45.3
Median	26.0	291.2	44.9
Our method	34.2	291.1	56.2

4.2.2. Edge Continuity Test. Typically, edges are composed of samples of high contrast and these samples have similar edge orientation vectors; that is, the orientation vectors should not change their directions abruptly along the edges [29]. Therefore, the survival indices in Q_{ec} that passed the connectivity test are further examined. If the two matched line segments corresponding to a specific survival index are separated by N pixels, the orientation vector $\boldsymbol{s}(x, y)$ is $(N, 1)$ for the horizontally dominant edges and $(1, N)$ for the vertically dominant edges. Assuming the current interpolation position is $\boldsymbol{l} = (x, y)$, we examine the cases at locations $(\boldsymbol{l} + \boldsymbol{s})$ and $(\boldsymbol{l} - \boldsymbol{s})$. If an edge having an orientation vector close to $\boldsymbol{s}$ is detected at one of the two locations, the orientation vector $\boldsymbol{s}(x, y)$ is assumed to be valid and its corresponding index is retained. Only one-direction edge continuity is sufficient because edges (such as corners) may not have contiguous orientations in both directions.

4.3. *Highly Oblique Edge Interpolation*

After determining the orientation vector of a highly oblique edge, we then modify the interpolated pixels generated by the fuzzy-inference method along the orientation vector. We simply use the bilinear interpolation to produce the new interpolated pixels around a given pixel $P(x, y)$. Because the gradient along the highly oblique edge orientation is small, the bilinear interpolation does not introduce noticeable blurring. Let the coordinate of the target sample be (x, y). The orientation vector is either $\mathbf{s} = (N, 1)$ or $\mathbf{s} = (1, N)$ depending on the detected edge is horizontally or vertically dominant. For a horizontally dominant edge, the interpolated pixels are replaced by

$$p\left(x + \frac{c}{2}\Delta, y + \frac{1}{2}\Delta\right) = \frac{1}{2}\left[p\left(x + \frac{c+N}{2}\Delta, y\right) + p\left(x + \frac{c-N}{2}\Delta, y + \Delta\right)\right], \tag{23}$$

where c is the (relative) location of pixels to be modified. The value of c ranges from $-2M$ to $2M$ if N is even, and from $-2M - 1$ to $2M - 1$ if N is odd. Note that $2M + 1$

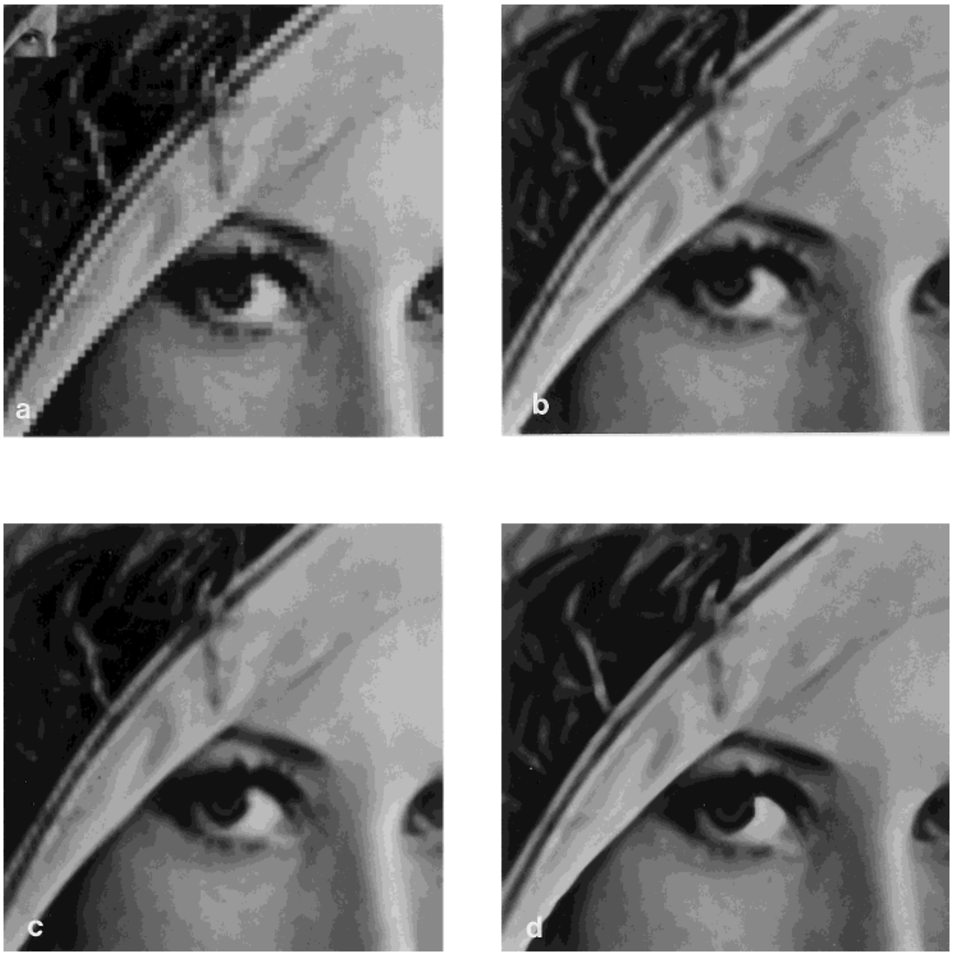


FIG. 15. "Lena," $z = 8$: (a) clipped image (64×64); (b) cubic spline; (c) median; and (d) our method.

is the line segment size. Also note that $p(x, y)$ on the right-hand side of (23) represents the interpolated pixel intensity values using the fuzzy inference interpolation. Figure 11 illustrates the pixel locations used in (23) for a zooming factor of 2. The upper portion is an example that N is even, and the lower portion is odd. Similar process is used for a vertically dominant edge. This interpolation process can be extended to higher zooming factors by modifying (23) slightly.

5. EXPERIMENTAL RESULTS

Two types of images have been tested using the proposed spatially adaptive fuzzy interpolator combined with the segment matching technique. They are classified based on the characteristics of high contrast edges: (i) graphics images with intensity discontinuity and (ii) natural images with/without anti-aliasing filtering. In computer synthesized images, edge intensities may be discontinuous and free from edge shadowing effect. On the other hand, edges in most natural images have shadowing effect (low-pass filtered). Furthermore, we also interpolate images that are subsampled without prefiltering to show the robustness of our method. The following simulation results are all tested using the same set of parameters summarized in Table 2.

The choice of the parameters T_s and λ has already been discussed in Section 3.1. Other than increasing the computational burden, lowering T_s or increasing λ does not affect the results significantly because the increased rugged sets can be well interpolated by the two-dimensional fuzzy method. The parameters α and β control the shape of the Gaussian membership functions whose widths are determined by the local gradients. The sharpness of the high contrast edges would be reduced if the magnitudes of α and β are increased. The values we use seem to be adequate for typical images. The line segment size is empirically chosen to be 5 pixels (i.e., $M = 2$) in our experiments. The edge profile cannot be reliably extracted if the segment size is too small. On the other hand, increasing its size not only increases the risk of covering two nearby edges but

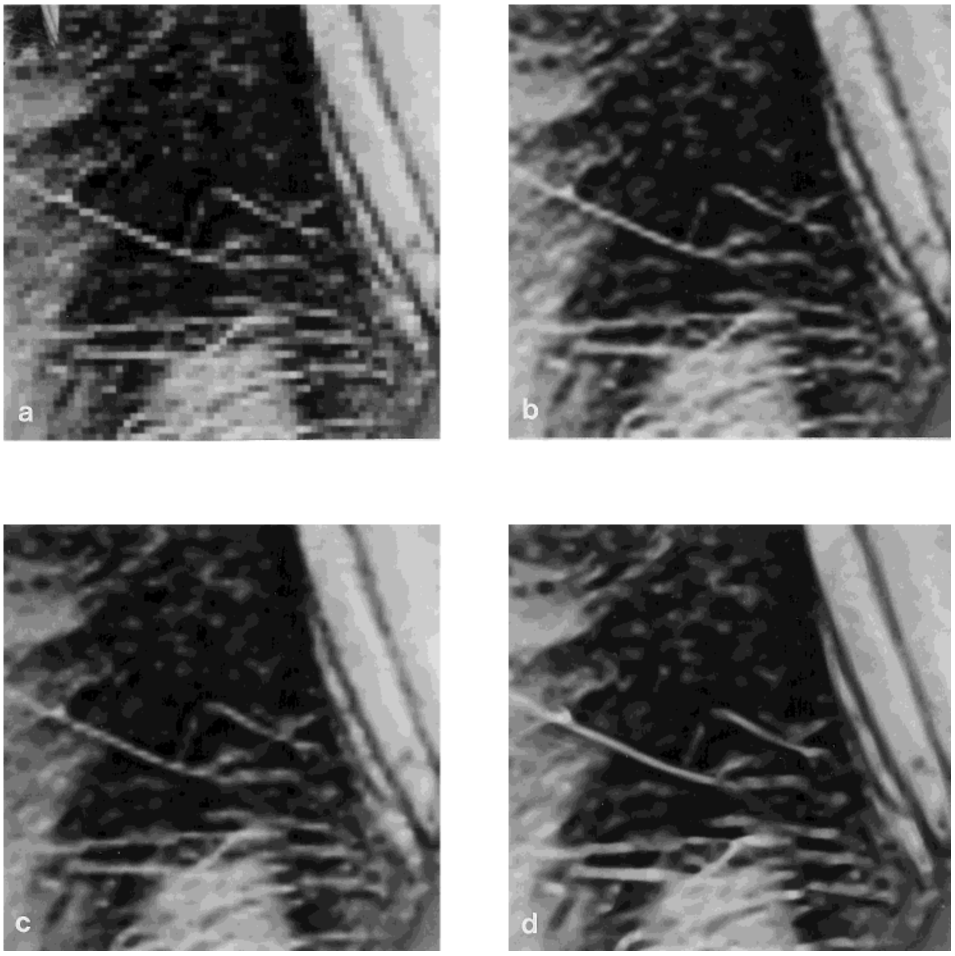


FIG. 16. "Baboon," $z = 8$: (a) clipped image (64×64); (b) cubic spline; (c) median; and (d) our method.

also increases calculations. The search range K determines the maximum detectable orientation angle. A large K ($K > 10$) may slightly improve the performance but the computation complexity is increased (proportional to K). Parameter T_g is used to detect the existence of an edge. It can vary from 0 to 50 without affecting the interpolated image quality for all the test images. If T_g is greater than 50 the probability of missing an edge increases and the jaggy artifact becomes noticeable. A T_g around 20 is found to be a good compromise between the computational burden and the probability of missing an edge. Similarly, T_a determines the matching criterion of two line segments. The T_a value ranges from 30 to 60 is found to be satisfactory for all the test images. Thus, $T_a = 50$ is chosen in our experiments. Lowering T_a (smaller than 30) increases the probability of missing an edge and then the jaggy artifact becomes visible.

Three interpolation methods are compared. They are (a) cubic spline interpolation [5] with $a = -0.5$, (b) median filter based method (INTER1 scheme in [30]), and (c) our proposed adaptive fuzzy method combined with segment matching technique.

5.1. *Graphics Images with Intensity Discontinuity*

To investigate the effect of our proposed interpolation method on the synthesized images, we generate two test images using computer graphics tools. The first synthesized image is a "Chinese character." This image is originally bi-level (0 and 255). It contains many highly oblique edges. The original image size is 32×32 pixels. The zooming factor is $z = 16$ for each direction. The results using three different methods are shown in Fig. 12. The original image is displayed in both the original small size and the enlarged version by pixel replication. The blurring and the jaggy artifacts can be found in the images interpolated by methods (a) and (b). In contrast, the edge sharpness and integrity are well preserved using our proposed method.

The second synthesized test image is the "shapes and text." This image contains texts (with and without shading), geometric shapes (sphere and right-angle corners), and

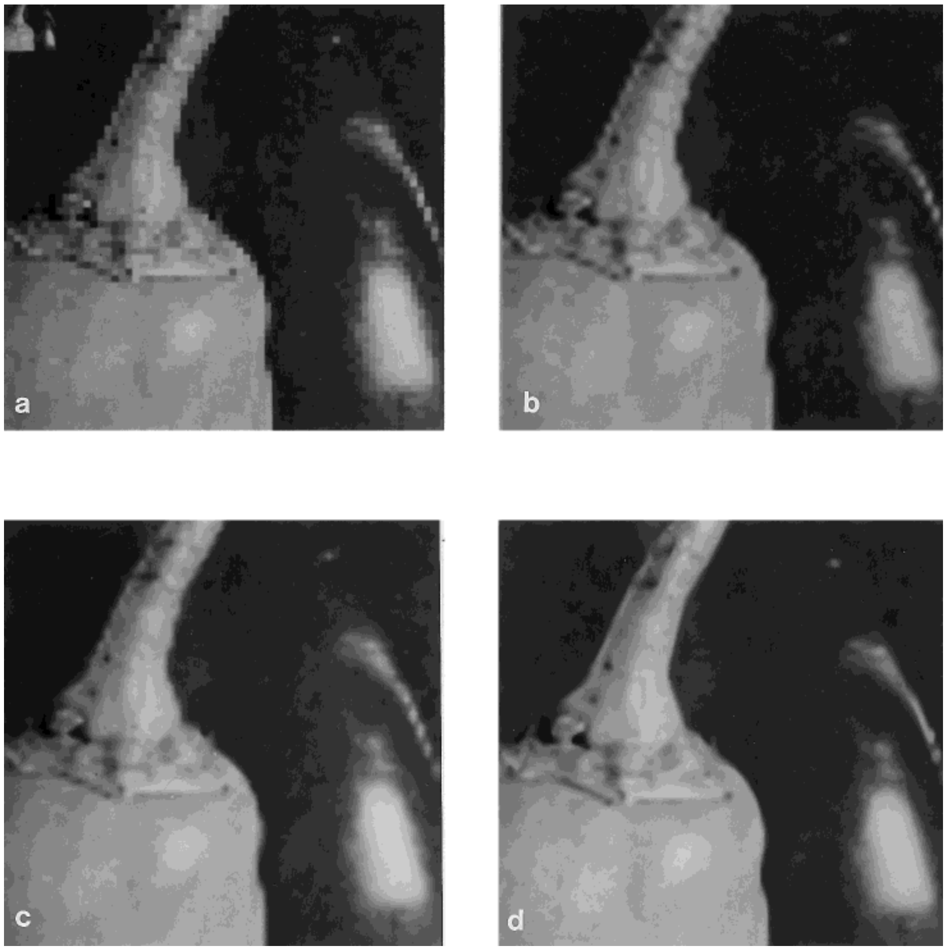


FIG. 17. Aliased "pepper," $z = 8$: (a) clipped image (64×64); (b) cubic spline; (c) median; and (d) our method.

high contrast textures. The original image size is 64×64 pixels. As shown in Figs. 13b–c, zooming artifacts are clearly observed in the interpolated images produced by the first two methods with a zooming factor $z = 8$. On the other hand, the intensity contrast and the curvature of the objects are well preserved by our method in Fig. 13d. The nearly only noticeable distortion produced by our method is the eroded right-angle corners in the letter "B." This is due to the assumption that if three pixels of a quadrant set have similar intensity, the middle interpolated has an intensity close to the majority. As a result, the middle interpolated pixel is always assigned the gray color and thus the right-angle corners are eroded. If synthesized images are our target images for interpolation, techniques can easily be developed to detect right-angle edges and this problem can be fixed. However, this situation does not occur for most natural images that have shadowed edges.

5.2. *Natural Images with/without Anti-aliasing Filtering*

Because of the Nyquist sampling criterion and the lowpass nature of image capture devices, the edge shadowing effect is almost always observed on the high contrast edges of the natural images. Two natural images are used to compare the performance of the aforementioned interpolation methods. The first test image is "Lena" with a resolution of 256×256 pixels. This image is produced by subsampling the original "Lena" that has the resolution of 512×512 pixels. Before subsampling, the original image is filtered by a lowpass FIR spatial filter to reduce aliasing effect. The tap coefficients and the frequency response of this FIR filter are shown in Fig. 14. This lowpass filter is widely used in MPEG-1 [2] and MPEG-2 [3] specifications for image size conversion.

The percentages and the number of the detected edge pixels are summarized in Table 3. The percentages of highly oblique edge pixels are relatively small. We list the mean squared errors (MSE) of the three methods in Table 4. Although both the cubic spline and the median based interpolation produce less mean squared error than our method, the jaggy and the blurring artifacts in their images are much more visible, particularly along highly oblique edges. It is well known that MSE does not completely agree with human subjective evaluation. In this specific case, our method often produces good visual quality edges but their locations may be shifted slightly. Hence, there may be fewer pixels in error, but the differences (error values) are larger comparing to those of the cubic spline interpolation. The MSE measure gives a much higher penalty on large errors. Hence, although the cubic spline and the median based interpolation may make mistakes in many pixels but the average differences are smaller. Therefore, our scheme has a slightly higher MSE value. To have a close-up examination of the image quality, we clip a portion of the original image and interpolate this clipped image (64×64 pixels) with $z = 8$. The results are shown in Figs. 15b–d. It is clear that the edges are preserved quite well by our method even when the zooming factor is as large as 8 in each direction.

A second image, "baboon," of size 256×256 is examined. This image is derived from filtering and subsampling the original 512×512 image. It contains rather complex texture patterns. We also clip a portion of the original image and interpolate this clipped image (64×64 pixels) with $z = 8$. A long and thin hair crosses this clipped image. The simulation results are shown in Figs. 16b–d. Again, our approach clearly preserves the edge sharpness and integrity better than the cubic and the median based methods.

To further verify the robustness of our method, we interpolate the images which were subsampled without using an anti-aliasing filter. As we know, aliasing appears if the original signal contains frequency components higher than half of the subsampling rate. The aliasing effect is visible around the high contrast edges. In general, these aliasing components degrade the interpolated image quality quite significantly when the conventional methods are in use. For illustration purpose, we first subsample the 512×512 "pepper" image down to 256×256 pixels without prefiltering. We also clipped a portion of the aliased image (64×64 pixels) and interpolate it with $z = 8$. It is again clear from Figs. 17b–d that our method preserves the shape and intensity contrast of the peppers better. This result indicates that our method is still effective even when the source images contain aliased components.

Finally, we want to comment on the average computational complexity of our scheme. Roughly speaking, if the computing time of the cubic spline interpolation is denoted by u (the bilinear interpolation has a similar value), the gradient computation, quadrant set classification, and fuzzy inference require totally around $3u$ to $4u$ computing time. The segment matching and the highly oblique edge interpolation further consume about $2u$ to $5u$ computing time, depending on the image content. For example, image "baboon" contains a lot of sharp edges and thus costs about $5u$ computing time. It is possible, however, to investigate methods that could reduce the computation of our scheme.

6. CONCLUSIONS

The ordinary image interpolators often try to preserve faithfully the frequency spectrum of the subsampled images. Hence, the interpolated images lack high frequency components and appear blurred. In addition, their interpolation process is often split into two independent subprocesses: one along the horizontal axis and the other, the vertical axis. The resultant off-axis edges are thus jagged. There are two major contributions in this paper. The first one is the fuzzy-inference based method that includes image diagonal correlation and matches image local characteristics. Therefore, it can recreate the high-contrast edges in the interpolated images. However, the fuzzy-inference interpolation still cannot remove edge jaggedness at the highly oblique edges. Our second contribution is to develop a segment matching technique that identifies the correct orientation of highly oblique edges and performs interpolation along the edge orientations. Combining these two techniques, we can improve the interpolated image subjective quality dramatically because the most evident improvement comes from the high-contrast edges that are sensitive to our eyes.

REFERENCES

1. P. J. Burt and E. H. Adelson, The Laplacian pyramid as a compact image code, *IEEE Trans. Commun.* **31,** 1983, 532–540.
2. ISO CD 11172-2, Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s, 1991.
3. ISO-IEC/JTC1/SC29/WG11, Test Model 2, Document MPEG92/No. 245, 1992.
4. R. E. Crochiere and L. R. Rabiner, Interpolation and decimation of digital signals: A tutorial review, *Proc. IEEE* **69,** 1981, 300–331.
5. J. A. Parker, R. V. Kenyon, and D. E. Troxel, Comparison of interpolating methods for image resampling, *IEEE Trans. Med. Imag.* **2,** 1983, 31–39.
6. H. S. Hou and H. C. Andrews, Cubic splines for image interpolation and digital filtering, *IEEE Trans. Acoustics, Speech and Signal Processing* **26,** 1978, 508–517.
7. R. G. Keys, Cubic convolution interpolation for digital image processing, *IEEE Trans. Acoustics, Speech and Signal Processing* **29,** 1981, 1153–1160.
8. Y. Wang and S. K. Mitra, Edge preserved image zooming, in *Proceedings, European Signal Processing Conference*, 1988, pp. 1445–1448.
9. K. Xue, A. Winans, and E. Walowit, An edge-restricted spatial interpolation algorithm, *J. Electron. Imag.* **1,** 1992, 152–161.
10. S. Thurnhofer, M. Lightstone and S. K. Mitra, Adaptive interpolation of images with application to interlaced-to-progressive conversion, in *Proceedings, SPIE Visual Communications and Image Processing*, 1993, 614–625.
11. M. Unser, A. Aldroubi, and M. Eden, Enlargement or reduction of digital images with minimum loss of information, *IEEE Trans. Image Processing* **4,** 1995, 247–258.
12. K. Jensen and D. Anastassiou, Subpixel edge localization and the interpolation of still images, *IEEE Trans. Image Processing* **4,** 1995, 285–295.

13. H. C. Ting and H. M. Hang, Spatially adaptive interpolation of digital images using fuzzy inference, in *Proceedings, SPIE Visual Communications and Image Processing, 1996,* pp. 1206–1217.
14. L. Kitchen and A. Rosenfeld, Edge evaluation using local edge coherence, *IEEE Trans. System, Man and Cybernetics* **9,** 1981, 597–605.
15. V. R. Algazi, G. E. Ford and R. Potharlanka, Directional interpolation of image based on visual properties and rank order filtering, in *Proceedings, IEEE Intl. Conf. on Acoustics, Speech and Signal Processing*, 1991, 3005–3008.
16. D. H. Hubel, *Eyes, Brain, and Vision*, Sci. Amer. Library, New York, 1988.
17. K. P. Hong, J. K. Paik, H. J. Kim and C. H. Lee, An edge-preserving image interpolation system for a digital camcorder, *IEEE Trans. Consumer Electronics,* **42,** 1996, 279–284.
18. S. Carrato, G. Ramponi, and S. Marsi, A simple edge-sensitive image interpolation filter, in *Proceedings, IEEE International Conf. on Image Processing*, 1996, 711–714.
19. R. R. Yager and D. P. Filev, *Essentials of Fuzzy Modeling and Control*, John Wiley & Sons, New York, 1994.
20. R. R. Yager and L. A. Zadeh, Eds., *An Introduction to Fuzzy Logic Applications in Intelligent Systems*, Kluwer, Boston, 1992.
21. H. J. Zimmermann, *Fuzzy Sets, Decision Making, and Expert Systems*, Kluwer, Boston, 1986.
22. L. X. Wang, *Adaptive Fuzzy Systems and Controls*, Prentice-Hall, New Jersey, 1994.
23. A. K. Jain, *Fundamentals of Digital Image Processing*, Prentice-Hall, New Jersey, 1989.
24. R. N. Bracewell, *Two-dimensional Imaging*, Prentice-Hall, New Jersey, 1995.
25. K. Sauer, Postprocessing of edge enhancement in low bit rate coded images, in *Proceedings, SPIE Visual Communications and Image Processing*, 1988, 658–665.
26. M. Okutomi and T. Kanade, A locally adaptive window for signal matching, *J. Comput. Vision* **7,** 1992, 143–162.
27. E. P. Lyvers and O. R. Mitchell, Precision edge contrast and orientation estimation, *IEEE Trans. Pattern Analysis and Machine Intelligence* **10,** 1988, 927–937.
28. R. M. Haralick and L. G. Shapiro, *Computer and Robot Vision: Volume I*, Addison-Wesley, Massachusetts, 1992.
29. P. H. Gregson, Using angular dispersion of gradient direction for detecting edge ribbons, *IEEE Trans. Pattern Analysis and Machine Intelligence* **15,** 1993, 682–696.
30. B. Zeng and N. A. Venetsanopoulos, A comparative study of several nonlinear image interpolation schemes, in *Proceedings, SPIE Visual Communications and Image Processing*, 1992, 21–29.

HOU-CHUN TING was born in Changhua, Taiwan, R.O.C., in 1964. He received the B.S. and M.S. degrees in Electrical Engineering from National Tsing Hua University in 1986 and 1988, respectively. He is currently working toward the Ph.D. degree at National Chiao Tung University. His research interests include image processing and video coding.

HSUEH-MING HANG received the B.S. and M.S. degrees from National Chiao Tung University, Hsinchu, Taiwan in 1978 and 1980, respectively, and the Ph.D. in Electrical Engineering from Rensselaer Polytechnic Institute, Troy, NY, in 1984. From 1984 to 1991, he was with AT&T Bell Laboratories, Holmdel, NJ, He joined the Electronics Engineering Department of National Chiao Tung University, Hsinchu, Taiwan, in December 1991. He was a conference co-chair of the Symposium on Visual Communications and Image Processing (VCIP), 1993, and the Program Chair of the same conference in 1995. He guest co-edited two *Optical Engineering* special issues on Visual Communications and Image Processing in July 1991 and July 1993. He was an associate editor of *IEEE Transactions on Image Processing* from 1992 to 1994 and a co-editor of the book *Handbook of Visual Communications* (Academic Press, 1995). He is currently an associate editor of *IEEE Transactions on Circuits and Systems for Video Technology* and an editor of *Journal of Visual Communication and Image Representation*, Academic Press. He is a senior member of IEEE and a member of Sigma Xi.