Spatially Adaptive Interpolation of Digital Images Using Fuzzy Inference

Hou-Chun Ting and Hsueh-Ming Hang

Dept. of Electronics Engineering
National Chiao Tung University
Hsinchu, Taiwan 30010, ROC

ABSTRACT

This paper presents a novel adaptive interpolation method for digital images. This new method can reduce dramatically the blurring and jaggedness artifacts on the high-contrast edges, which are generally found in the interpolated images using conventional methods. This high performance is achieved via two proposed operators: a fuzzy-inference based edge preserving interpolator and an edge-shifted matching scheme. The former synthesizes the interpolated pixel to match the image local characteristics. Hence, edge integrity can be retained. However, due to its small footage, it does not work well on the sharply curved edges that have very sharp angles against one of the coordinates. Therefore, the edge-shifted matching technique is developed to identify precisely the orientation of sharply curved edges. By combining these two techniques, the subjective quality of the interpolated images is significantly improved, particularly along the high-contrast edges. Both synthesized images (such as letters) and natural scenes have been tested with very promising results.

Keywords: Image Interpolation, Fuzzy system, Nonlinear interpolator

1. INTRODUCTION

Interpolation is an important technique in multi-rate image signal processing such as pyramid coding, multi-resolution television broadcasting, and image zooming for viewing comfort. Linear and model-based interpolation methods are commonly used.[1–3] According to sampling theory, an ideal lowpass filter is needed to remove the replicated copies of the subsampled signal spectra located at higher frequencies. Therefore, all the interpolator should behave like a lowpass filter. It is clear that the conventional spatial invariant linear interpolator is designed to be a lowpass filter. Model based or surface fitting methods[2,3] also have lowpass nature although their processes may be nonlinear. Typically, these methods patch pieces of continuous spline functions to match the given (known) pixels, and, thus, all the pixels in between can be synthesized by superimposing the overlapped splines.

Blurring and jaggedness are the most visible defects that appear in images using the conventional interpolation methods. Theses artifacts are caused by the lowpass filter used to remove the unwanted highpass replica of the zoomed images in the frequency domain. Because it is not possible to implement the ideal low pass filter in practice, non-ideal filters such as zero-order hold (nearest neighbor) and first-order hold (bilinear) are often used to filter out the high pass replica. These non-ideal lowpass filters suppress low frequency components and introduce high frequency component aliasing. Low frequency suppression reduces the spatial resolution of the interpolated images (blurring) and the undesired high frequency aliasing produces broken edges (jaggedness).

If we look into the image interpolation problem deeper, the ordinary sampling theory does not offer us the most satisfactory solution. Although the ideal low pass filter provides the most faithful frequency spectrum of the interpolated signals in theory, it does not provide the most natural interpolated pictures because the original (high resolution image) high frequency components were eliminated in the subsampling process. In other words, a *better looking* picture can be produced by inserting properly the missing high frequency components. This is clear when we examine the

 0-8194-2103-0/96/$6.00

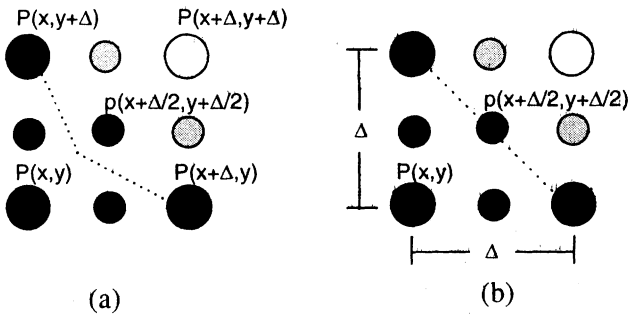


Figure 1: Utilization of the diagonal correlation in image interpolation.

frequency spectrum of the ideally interpolated signals. Their high frequency components are set to zero because in theory, the lost information (high frequency components) cannot be recovered. (We cannot create something we do not know.) However, for most typical pictures, the information contained in the subsampled image usually provides clues about the lost components. For example, if it 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 or spatially adaptive interpolators could potentially produce better looking images because they *create* additional information. Our approach in this paper is another attempt to regenerate the *original* high resolution images based on our prior knowledge of *typical* images. Thus, our methods contain two elements: i) decide the local image characteristics (flat regions, edges, etc.) and ii) generate the interpolated pixels by properly weighted averaging.

In addition, most existing interpolation methods are separable operators along two axes. The separable operations result in jaggedness on particularly diagonal edges. Figure 1 is used to illustrate the potential advantage of using diagonal correlations. Figure 1(a) shows the interpolation result generated by a separable bilinear interpolation filter. It is clear that the interpolated pixel $p(x+\Delta/2, y+\Delta/2) = \frac{1}{4}(P(x,y)+P(x+\Delta,y)+P(x,y+\Delta)+P(x+\Delta,y+\Delta))$ does not give the desirable value, where $P(x,y)$, $P(x+\Delta,y)$, $P(x,y+\Delta)$ and $P(x+\Delta,y+\Delta)$ are the four given pixels. If we consider the diagonal correlation, since 3 out of the 4 given pixels have similar intensity values, it seems more natural to set the interpolated center pixel the same as the 3 dominated pixels as shown by Fig. 1(b). 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 are usually interfered by camera noise; also, the surfaces of a flat object do not produce absolutely the same light intensity captured by the camera. How to compute the directional correlations of the original pixels from the sampled pixel and use them properly to generate interpolated pixels is one of the key issues in this paper. Four directional correlation indices can be defined on the four neighboring pixels located on the rectangular sampling grid as shown in Fig. 1. These four indices are the horizontal, the vertical, the diagonal (upper-left to lower-right) and the anti-diagonal (upper-right to lower-left). Instead of defining and calculating these four indices explicitly, a fuzzy inference method is adopted here to interpolate images without having to calculate these indices explicitly. This is our first contribution.

Because edges of an image convey most essential perceptible information to the human eyes,* it is important to preserve the edge integrity and to enhance edge shape to compensate for the artifacts due to the lost high frequency components in the subsampling process. In the previous works, local edge operator is often used to extract the edges. However, it is observed that the *sharply curved edges*† in natural images often can not be extracted by the local edge operator. If the sharply curved edges can not be accurately identified, the edge jaggedness in the interpolated images cannot be avoided even when an edge-preserving interpolator is in use. In order to identify the orientations of the

*It is reported that[4] the very first step in the human perceptual system is to convert images into nearly edge-only signals.

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

sharply curved edges, an edge-shifted matching method is proposed in this paper. This method detects the edge orientation by matching the image segments across the edge borders in the horizontal and vertical directions. A few criteria are suggested to evaluate the matching results and to decide whether an edge exists or not. By locating the edges quite accurately, the jaggedness can be reduced significantly. This is our second contribution.

This paper is organized as follows. An edge-preserving interpolation filter based on fuzzy inference is described in Section 2. Because it is difficult to interpolate the sharply curved edges with local edge detector only, an edge-shifted matching method is proposed in Section 3. With the help of certain screening criteria, this method locates the sharply curved edges quite successfully. Experimental results of the proposed methods are provided in Section 4 to demonstrate their performance. A brief summary is presented in Section 5.

2. FUZZY-INFERENCE BASED INTERPOLATION

The simplest interpolation method is pixel replication (nearest neighbor). From the viewpoint of filtering, pixel replication is a zero-order hold operator, which is a rather poor lowpass filter. Similar to pixel replication method, bilinear interpolation is a low-pass filter implemented by a first-order hold operator. Bilinear interpolation introduces less high pass error than pixel replication as their spectra indicate. More complicated linear interpolation methods which attempt to improve the performance by including more known samples have been proposed.[2] However, they all suffer from the drawbacks we discussed in the previous section. Hence, we propose a nonlinear 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 multi-dimensional cases without the drawback of separable operations.

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 and Δ be the sampling interval.
By Taylor series expansion,

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

where R_1 represents all the remaining higher order terms (similar definitions for R_2 and R_3 in the following equations). Similarly, we have

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

Substituting x for $x + \Delta$ in (2), we have

$$f(x + \frac{\Delta}{2}) = f(x + \Delta) - f'(x + \Delta)\frac{\Delta}{2} + f''(x + \Delta)\frac{\Delta^2}{8} + R_3. \tag{3}$$

Averaging (1) and (3), we obtain

$$f(x + \frac{\Delta}{2}) = \frac{1}{2}\left[f(x) + f(x + \Delta) + \frac{f'(x) - f'(x + \Delta)}{2}\Delta + \frac{f''(x) + f''(x + \Delta)}{8}\Delta^2 + R_1 + R_3\right]. \tag{4}$$

It is ready to see that bilinear interpolation can be deduced from (4) by assuming that the first-order differential of $f(x)$ is constant and the higher order terms are negligible for all x. Though only the middle point is expressed in (4), expressions for all the point between the given samples can be derived similarly. If the second-order differential becomes significant and is assumed to be constant, cubic spline interpolation with parameter $a = -0.5$ is thus deduced as described in Parker et al.[2] In this 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 taken into account. From signal processing viewpoint, we are designing a higher order filter to approximate the ideal lowpass filter.

It seems that including more 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 discrete grids. Approximating differentials using differences at the known samples has a limited accuracy; the approximation error increases as the differentiation

order grows. Similar problem occurs when extending the above approach to the two-dimensional (2-D) cases. It gets even worse in 2-D because the cross terms — the second-order differentials with respect to the horizontal and the vertical axes simultaneously — are not well-defined if only the discrete samples are available. In addition, although in theory, a higher order filter can approximate the ideal lowpass filter better, in reality, images characteristics vary from one local region to the other; therefore, high order filters may not produce a superior 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 expressed by (4). Assume that the differentials with order 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 sample points. If $|f'(x)| > |f'(x+\Delta)|$, we have

$$\text{either} \quad f'(x) > |f'(x+\Delta)| \geq 0 \quad \text{or} \quad f'(x) < -|f'(x+\Delta)| \leq 0. \tag{5}$$

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

$$f(x+\frac{\Delta}{2}) = \frac{1}{2}\left[f(x) + f(x+\Delta) + \frac{f'(x) - f'(x+\Delta)}{4}\Delta\right]. \tag{6}$$

Because $f'(x) - f'(x+\Delta) > 0$, we conclude from (6) that

$$f(x+\frac{\Delta}{2}) > \frac{1}{2}\left[f(x) + f(x+\Delta)\right]. \tag{7}$$

Note that $f(x) < f(x+\Delta)$ because $f'(x) > 0$. Equation (7) shows that the middle interpolated sample should have a value closer to $f(x+\Delta)$, the sample with a lower gradient. Similar statement can be concluded for the second case in (5).

2.2. Edge preserving interpolation based on fuzzy inference

Conventional interpolation methods often blur (smooth) the edges (step functions) in the interpolated images (signals) because lowpass-like operators are used to remove the unwanted highpass replica. As a result, no information above half of the Nyquist frequency of the original signals is available in the interpolated signals. This effect results in blurred edge in the interpolated images. Suggested by the observation in the previous sub-section, we propose an edge preserving interpolation method based on fuzzy inference[9] to overcome partially the above problem. We synthesize the interpolated sample using

$$f(x+\frac{\Delta}{2}) = w(|f'(x)|) \cdot f(x) + w(|f'(x+\Delta)|) \cdot f(x+\Delta), \tag{8}$$

in which

$$w(|f'(x)|) + w(|f'(x+\Delta)|) = 1 \tag{9}$$

with $w(|f'(x)|), w(|f'(x+\Delta)|) \geq 0$. Based on the reasoning in the previous paragraph, $w(\cdot)$ should be a non-increasing function; that is, the value of $w(a)$ is smaller when a gets larger. Here we adopt fuzzy logic to implement (8) instead of defining the weighting function $w(\cdot)$ explicitly.

In its general form, there are four basic components in a fuzzy logic system: fuzzy rule base, fuzzifier, fuzzy inference engine and defuzzifier. A fuzzy rule base is a collection of IF-THEN rules that have the following structure:

$$Rule(k): IF \ r_1 \ is \ F_{k1} \ and \ \cdots \ and \ r_n \ is \ F_{kn}, \ THEN \ z \ is \ G_k \tag{10}$$

where F_{ki} and G_k are fuzzy sets defined on $U_i \subset R$ and $V \subset R$, respectively, and $\underline{r} = (r_1, \cdots, r_n)^T \in U_1 \times \cdots \times U_n$ and $z \in V$ are input and output linguistic variables. To convert (8) into a fuzzy logic system, a simple fuzzy rule base with 2 rules is constructed as follows:

$$\begin{array}{l} Rule(1): IF \quad r_1 \ is \ near \ x \quad THEN \ f(r_1) \ is \ f(x) \\ Rule(2): IF \quad r_1 \ is \ near \ x+\Delta \quad THEN \ f(r_1) \ is \ f(x+\Delta) \end{array} \tag{11}$$

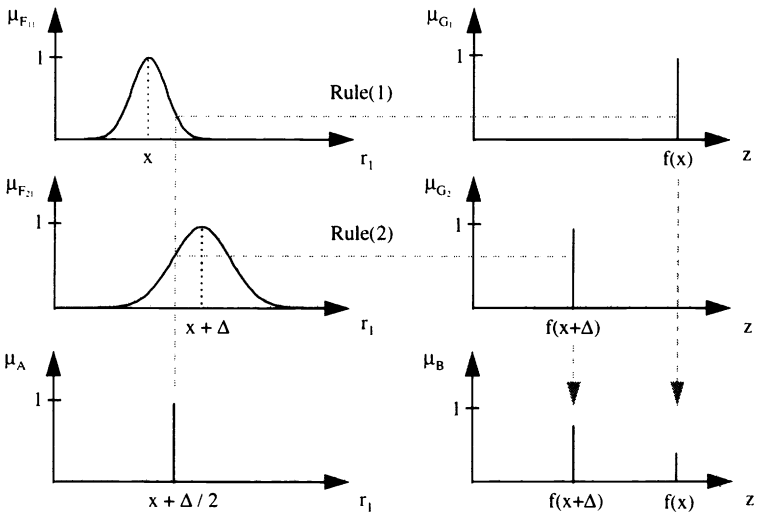


Figure 2: Fuzzy inference structure for interpolation.

At first glance, equation (11) does not seem to bear any further information. However, it should be noted that the meaning of "near" in these two rules may be different from each other and this becomes our focus; that is, how to define a proper membership function for this fuzzy word. Indeed, determining the shape of membership functions is quite subjective and depends on applications. Here we choose Gaussian function for fuzzy sets of the antecedent part in (11):

$$\mu_{F_{ki}}(r_i) = a_{ki} \exp[-(\frac{r_i - \bar{r}_{ki}}{\sigma_{ki}})^2], \tag{12}$$

where a_{ki}, $\bar{r}_{ki}$ and σ_{ki} are adjustable parameters. In our use, a_{ki} is set to 1 for all k and i, and $\bar{r}_{11}$ is x and $\bar{r}_{21}$ is $x + \Delta$, respectively. Note that $i = 1$ for 1-D input signal. According to the observations we have at the end of the previous sub-section, we make σ_{k1} depending on the absolute value of the "gradient" (first-order difference). We assume a simple relation between σ_{k1} and the gradient $D(\cdot)$:

$$\sigma_{k1} = \alpha \cdot D(\bar{r}_{k1}) + \beta, \tag{13}$$

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

$$D(\bar{r}_{k1}) \equiv \frac{|f'(\bar{r}_{k1})|}{f'_{max}}, \tag{14}$$

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.

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 inferred by $Rule(k)$ in (10). The output membership function is described by

$$\mu_{B_k}(z) = \sup_{\underline{r} \in U}[\mu_{F_{k1} \times \cdots \times F_{kn} - G_k}(\underline{r}, z) \star \mu_A(\underline{r})]. \tag{15}$$

The fuzzy implication $F_{k1} \times \cdots \times F_{kn} - G_k$ can be defined in several different ways. Here we adopt the product-operation rule[8]; that is,

$$\mu_{F_{k1} \times \cdots \times F_{kn} - G_k}(\underline{r}, z) = \prod_{i=1}^{n} \mu_{F_{ki}}(\underline{r}) \cdot \mu_{G_k}(z). \tag{16}$$

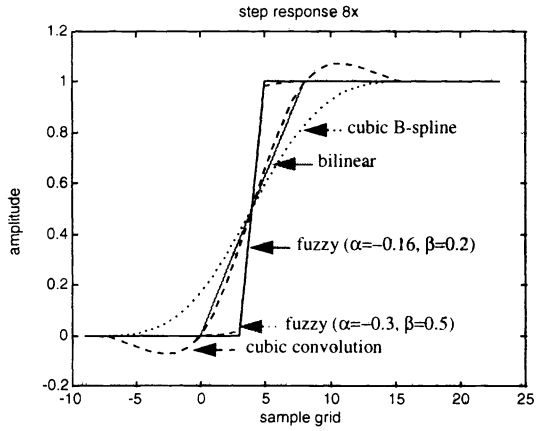


Figure 3: Step response comparison of different interpolation methods.

Also, the $\star$ is chosen to be the algebraic product operator which is one type of the operators in a class of t-norm intersection operations.[8]

As described earlier, the input of a fuzzy logic system should be a fuzzy set. In the case of interpolation, the input is a data location, a crisp point. We have to map this crisp input to a fuzzy set. For simplicity, singleton fuzzifier is used. At the end of this inference process, a defuzzifier maps the output fuzzy set to a crisp point, which is an interpolated pixel in our case. In this paper, we choose a center averaging defuzzifier which is defined by

$$z = \frac{\sum_{k=1}^{K} \bar{z}_k(\mu_{B_k}(\bar{z}_k))}{\sum_{k=1}^{K}(\mu_{B_k}(\bar{z}_k))}, \tag{17}$$

where $\bar{z}_k$ represents the point at which the fuzzy membership function μ_{B_k} achieves its maximum value. Note that the structure of (17) matches the requirement of (8) and (9). Figure 2 shows 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. We compare the step response of this method with those of the conventional interpolation methods in Fig. 3. This figure shows an example of interpolation by a zooming factor of 8. It is found that for fuzzy interpolation, a better result is obtained by iterating the middle point interpolation three times to produce the 7 interpolated points instead of producing these 7 interpolated points directly. It can be seen that the high-contrast jump is preserved by the fuzzy inference method. We notice that the parameters α and β in (13) control the shapes of the Gaussian membership functions and thus have an impact on the final results. Furthermore, the pair, (α,β), should be chosen to keep $\sigma_{k1} > 0$.

2.3. Digital image interpolation

Extending the one-dimensional (1-D) interpolation method described in the previous section to the two-dimensional case, a fuzzy-inference based interpolation method for digital images is thus devised. Our fuzzy image interpolation method consists of two steps: 1) use smoothed gradient (Prewitt) or Sobel gradient operators[10] to compute the spatially varying gradient of images; 2) adjust the parameters σ_{ki} of the fuzzy membership functions according to the calculated gradient and then the interpolated pixels are synthesized accordingly. The rule base consists of 4 rules as follows:

$$\begin{array}{llllllll}
Rule(1): IF & r_1 \ is \ near \ x & and & r_2 \ is \ near \ y & THEN & p(r_1,r_2) \ is & P(x,y) \\
Rule(2): IF & r_1 \ is \ near \ x+\Delta & and & r_2 \ is \ near \ y & THEN & p(r_1,r_2) \ is & P(x+\Delta,y) \\
Rule(3): IF & r_1 \ is \ near \ x & and & r_2 \ is \ near \ y+\Delta & THEN & p(r_1,r_2) \ is & P(x,y+\Delta) \\
Rule(4): IF & r_1 \ is \ near \ x+\Delta & and & r_2 \ is \ near \ y+\Delta & THEN & p(r_1,r_2) \ is & P(x+\Delta,y+\Delta).
\end{array} \tag{18}$$

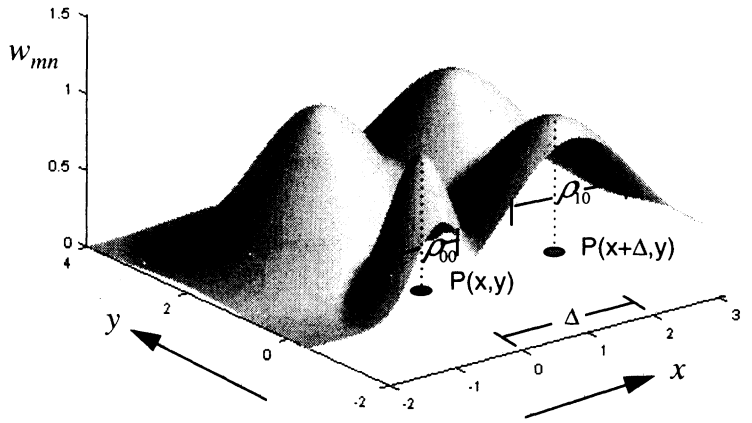


Figure 4: Two dimensional Gaussian shape membership functions.

A fuzzy inference engine similar to that in Fig. 2 can thus be constructed. To determine σ_{ki} of the membership functions for the two antecedent parts, instead of setting them separately, we have

$$\sigma_k = \alpha \cdot D(\bar{r}_{k1}, \bar{r}_{k2}) + \beta, \tag{19}$$

where $D(\bar{r}_{k1}, \bar{r}_{k2})$ is the average magnitude of the two directional gradients at $P(\bar{r}_{k1}, \bar{r}_{k2})$ along the horizontal and the vertical axes. Similar to the 1-D case, the product-operation rule is adopted in the fuzzy implication. An example of the 2-D fuzzy membership functions is shown in Fig. 4.

Based on the choices we have made earlier, the interpolated pixel $p(x+\delta x, y+\delta y)$ is computed by the defuzzification process:[8]

$$\begin{aligned} p(x+\delta_x, y+\delta_y) &= \frac{\sum_{m=0}^{1}\sum_{n=0}^{1} P(x+m\Delta, y+n\Delta)\cdot w_{mn}(x+\delta_x, y+\delta_y)}{\sum_{m=0}^{1}\sum_{n=0}^{1} w_{mn}(x+\delta_x, y+\delta_y)}, \\ & 0 \le \delta_x, \delta_y \le \Delta, \end{aligned} \tag{20}$$

where $P(x+m\Delta, y+n\Delta)$ is the given pixel on the sampling grid before interpolation. The function $w_{mn}(x,y)$ is the 2-D membership function as described below.

$$w_{mn}(x,y) = \exp\left(-\frac{(x-m\Delta)^2+(y-n\Delta)^2}{\rho_{mn}^2}\right) \quad m,n = 0,1; \quad x,y \in Z, \tag{21}$$

where $\rho_{00} = \sigma_1$, $\rho_{10} = \sigma_2$, $\rho_{01} = \sigma_3$ and $\rho_{11} = \sigma_4$; σ_k is the parameter derived from (19) and Z is the 2-D coordinates of the given pixels.

3. EDGE-SHIFTED MATCHING METHOD FOR IDENTIFYING SHARPLY CURVED EDGES

Though the proposed interpolation method based on fuzzy inference preserves edges in the interpolated images, it is only locally effective. Since only 4 neighboring pixels are used in computing the interpolated pixels (as shown in (20)) and a square region of 4x4 pixels are involved in computing the four gradients (when a 3x3 gradient operator is in use), it does not have sufficient information to interpolate the sharply curved edges well for lack of their orientations. For the same reason, the ordinary small size edge detectors such as the 3x3 Sobel operators also fail to identify their correct orientations. Examining the characteristics of the sharply curved edges in natural images, we develop

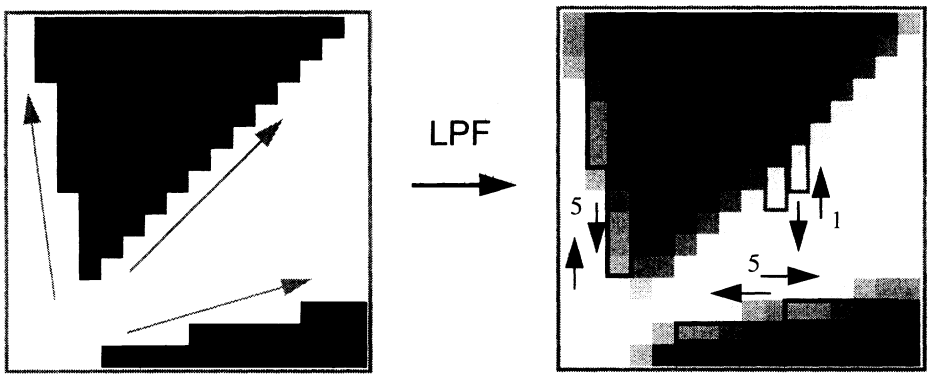


Figure 5: Edge shadowing effect caused by low pass filtering.

a matching algorithm in spatial domain to identify the edge orientation. After the edge orientation being identified, the interpolated pixels can then be synthesized along its orientation. Combining this technique together with fuzzy-inference based interpolation, sharply curved edges can be recreated without noticeable artifacts.

3.1. Edge shadowing effect caused by lowpass filtering

Because of the bandwidth limitation, it is impossible to precisely transmit or store an analog image with continuous 2-D coordinates. To reduce the transmission bandwidth or storage capacity, sampling is essential to generate the digital images. An anti-aliasing lowpass filter is usually applied to images before sampling to avoid aliasing. As a result, the original high-contrast edges (with discontinuous intensity) are blurred by lowpass filters.

Figure 5 shows the smoothing effect due to low pass filtering on high-contrast edges with different orientations. In the figure on the right-hand side, we also demonstrate an orientation identification method for sharply curved edges. Edges are divided into two classes according to their orientations: the horizontally dominant and the vertically dominant classes. For example, as shown in the left figure in Fig. 5, the angle of the lower right edge is less than 45 degrees and thus this edge is classified to the horizontally dominant. It is shown that two horizontal line segments 5 pixels apart horizontally have matched intensity profiles. The orientation of the intensity profile in this case is approximately aligned with the edge orientation when the edge angle is very sharp. Another example is the upper left edge, whose angle is greater than 45 degrees and thus it is a vertically dominant edge. The third edge in the middle is a diagonal edge with a nearly 45-degree angle and thus can be classified as either horizontally or vertically dominant but in either case it would be irrelevant to the operations described in the next sub-section.

3.2. Line mask shift matching

Most edge preserving interpolation methods consist of two step: i) edge detection and tracing, and ii) interpolation along the orthogonal direction to the estimated edge line with a contrast enhancement scheme.[3,5–7] At the first step, sharply curved edges are usually difficult to detect using only local edge detectors. Though an operator with a larger window size can be used; however, the number of pixels involved increases significantly and the detection algorithm becomes very complicated. We thus propose a simple but effective edge orientation identification method for, particularly, the sharply curved edges.

First, we must determine whether an edge exists or not and then identify the dominant orientation if an edge indeed exists. Local operators such as smoothed gradient (Prewitt) or Sobel gradient operators can be used to detect edges. Note that this edge detection operation is already a part of the fuzzy-inference interpolation. For example, the calculated gradient vector is $\vec{g}(x,y) = (g_h(x,y), g_v(x,y))$, $x, y \in Z$. An edge is detected if the gradient

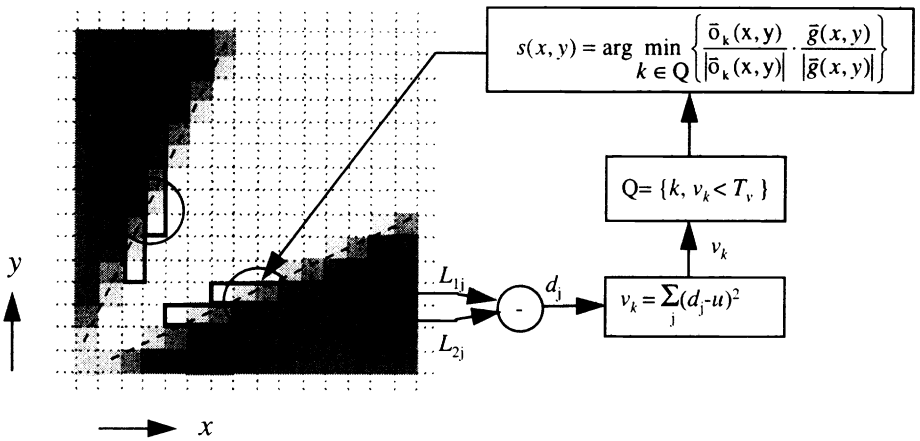


Figure 6: Edge-shifted matching method.

magnitude $|\vec{g}(x,y)| > T_g$, where T_g is a pre-selected threshold value. We then determine the edge orientation. If $|g_h(x,y)| > |g_v(x,y)|$, then the edge is vertically dominant; otherwise, it is horizontally dominant. Now, we start to identify the edge orientation by shifting two line-masks around the detected edge and compare their profiles.

If an edge is horizontally dominant, we create two line-masks: $L_1 = \{P(x+(k_1-M)\Delta, y), P(x+(k_1-M+1)\Delta, y), \cdots, P(x+(k_1+M)\Delta, y)\}$, and $L_2 = \{P(x+(-k_2-M)\Delta, y+\Delta), P(x+(-k_2-M+1)\Delta, y+\Delta), \cdots, P(x+(-k_2+M)\Delta, y+\Delta)\}$, in which k_1 and k_2 indicate the relative locations of these two segments and their initial value decides the search range $(=(k_1+k_2)$ and M determines the mask size, $2M+1$. Similarly, vertical line-masks can be created as shown in Fig. 6, if the edge is decided to be vertically dominant. Let L_{ij} represents the jth components in the line-mask L_i. We first compute the line difference $d_j = L_{1j} - L_{2j}$, and its variance

$$v_{k=(k_1+k_2)} = \sum_{j=1}^{2M+1} (d_j - u)^2 \tag{22}$$

where u is the mean of $\{d_j, j = 1, ..., 2M+1\}$. If $v_{k=(k_1+k_2)} < T_v$, we add the index $k = (k_1+k_2)$ to the candidate set Q. Threshold T_v controls the matching criterion. Repeat the above process by shifting the line-mask 1 pixel inward alternately in opposite direction; that is, one of k_1 or k_2 is decreased by 1 at one iteration.

To determine the correct edge orientation, we check all the items in the candidate set Q. If Q is an empty set, it means that no significant edge orientation presents. Otherwise, for a horizontally dominant edge, we create an orientation vector $\vec{o}_k(x,y) = (k,1)$ for each element in Q. Similarly, vector $\vec{o}_k(x,y) = (1,k)$ is created for a vertically dominant edge. Ideally, the orientation vector should be perpendicular to the gradient vector at an edge; that is, their inner product should be zero. However, this criterion is too strict and thus is not suitable for natural images. Instead, we choose the orientation vector that achieves the minimum inner product as shown below:

$$s(x,y) = \arg\min_{k \in Q} \left\{ \frac{\vec{o}_k(x,y)}{|\vec{o}_k(x,y)|} \cdot \frac{\vec{g}(x,y)}{|\vec{g}(x,y)|} \right\}. \tag{23}$$

Based on this expression, the search range can be narrowed down to reduce computation by estimating an initial orientation vector that has inner product near zero.

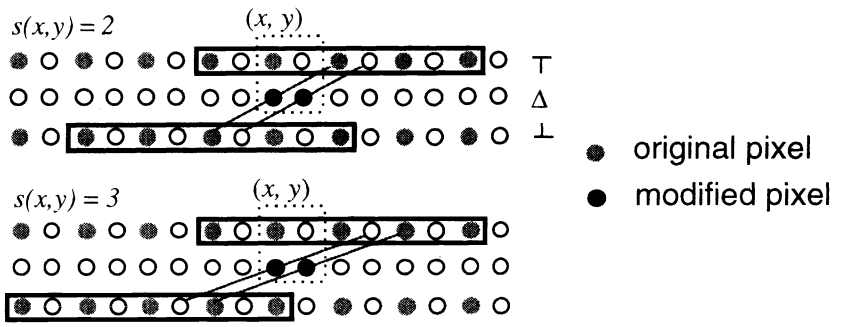
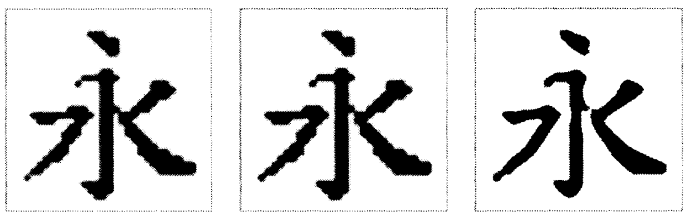


Figure 7: Edge-shifted interpolation.



(a) (b) (c)

Figure 8: "Chinese Character", 4x interpolated. (a) bilinear; (b) cubic spline; (c) adaptive fuzzy inference

3.3. Sharply curved edge interpolation

After determining the orientation vector for a detected edge, we then synthesize the interpolated pixels along the orientation vector to replace the ones generated by the fuzzy-inference method. Here, we simply use bilinear interpolation to modify the interpolated pixels around a given pixel $P(x, y)$. Because the gradient along the edge orientation is small, bilinear interpolation does not introduce noticeable blurring. For a horizontally dominant edge, according to (23) its orientation vector is $(s(x, y), 1)$. The interpolated pixels are then modified to

$$p(x, y + \frac{\Delta}{2}) = \frac{1}{2}[p(x + \frac{s(x, y)}{2}\Delta, y) + p(x - \frac{s(x, y)}{2}\Delta, y + \Delta)]; \tag{24}$$

$$p(x + \frac{\Delta}{2}, y + \frac{\Delta}{2}) = \frac{1}{2}[p(x + \frac{s(x, y) + 1}{2}\Delta, y) + p(x - \frac{s(x, y) - 1}{2}\Delta, y + \Delta)], \tag{25}$$

where $p(x, y)$ on the right-hand side of the above equations represents the interpolated pixels using the fuzzy inference interpolation. Figure 7 illustrates the pixel locations used in (24) and(25) for a zooming factor of 2. The upper portion is an example that $s(x, y)$ 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.

(a) (b) (c)

Figure 9: "Lena", 4x interpolated. (a) bilinear; (b) cubic spline; (c) adaptive fuzzy inference

4. EXPERIMENTAL RESULTS

Several types of images have been tested using the proposed spatially adaptive fuzzy interpolator combined with the edge-shifted matching technique. Digital images can roughly be divided into two categories: i) images synthesized by computers, and 2) natural images captured by camera. For computer synthesized images, the edge intensity can be discontinuous and free from edge shadowing effect. On the other hand, edges in most natural images have shadowing effect.

The interpolated "Chinese Character" (originally bi-level), "Lena" and "aliased Pepper" (subsampled without pre-filtering) images with a 4x zooming factors using (a) bilinear interpolation; (b) cubic spline interpolation[2] with $a = -0.5$; and (c) adaptive fuzzy method ($\alpha = -0.16, \beta = 0.2$) combined with edge-shifted matching technique ($M = 2, K = 5, T_g = 40, T_v = 5000$) are shown in Figs. 8-10. It can be observed that jaggedness and blurring on the sharp edges are clearly visible in the cases of bilinear and cubic spline interpolation. These subjective artifacts are greatly reduced in the fuzzy-interpolated images.

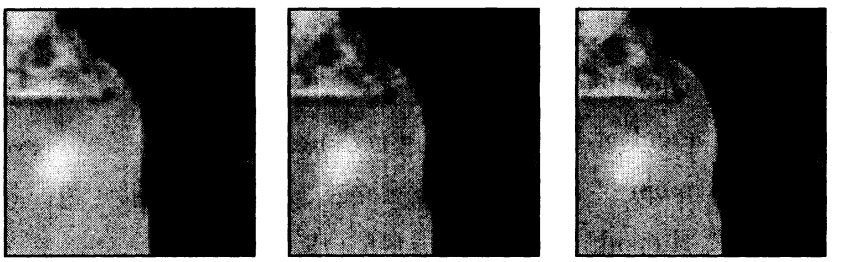
5. CONCLUSIONS

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 broken into two independent sub-processes: one along horizontal axis and the other, 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 preserve the high-contrast edges. However, like most small-size local edge detectors which fail to correctly identify the orientation of sharply curved edges, the fuzzy-inference interpolation does not work well on the sharply curved edges. Our second contribution is to develop an edge-shifted matching technique that identifies the correct orientation of sharply curved edges and performs interpolation along these edges properly. 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 most sensitive to our eyes.

6. REFERENCES

[1] R. E. Crochiere and L. R. Rabiner, "Interpolation and Decimation of Digital Signals — A Tutorial Review," *Proc. IEEE*, Vol. 69, pp. 300-331, Mar. 1981.

[2] J. A. Parker, R. V. Kenyon and D. E. Troxel, "Comparison of Interpolating Methods for Image Resampling," *IEEE Trans. on Medical Imaging*, Vol. MI-2, No.1, pp.31-39, Mar. 1983.



(a) (b) (c)

Figure 10: "Aliased Pepper", 4x interpolated. (a) bilinear; (b) cubic spline; (c) adaptive fuzzy inference

[3] K. Jensen and D. Anastassiou, "Subpixel Edge Localization and the Interpolation of Still Images," *IEEE Trans. on Image Processing*, Vol.4, No.3, pp.285-295, Mar. 1995.

[4] D.H. Hubel, *Eyes, Brain, and Vision*, New York, N.Y.: Scientific American Library, 1988.

[5] K. Xue, A. Winans and E. Walowit, "An Edge-restricted Spatial Interpolation Algorithm," *J. of Electronic Imaging*, Vol.1(2), pp.152-161, Apr. 1992.

[6] Y. Wang and S. K. Mitra, "Edge Preserved Image Zooming," *Proc. of European Signal Processing Conference*, pp.1445-1448, 1988.

[7] K. Sauer, "Postprocessing of Edge Enhancement in Low Bit Rate Coded Images," *Proc. of SPIE, Visual Communications and Image Processing*, pp.658-665, 1988.

[8] L. X. Wang, *Adaptive Fuzzy Systems and Controls*, Prentice-Hall: Englewood Cliffs, NJ, 1994.

[9] R. R. Yager and D. P. Filev, *Essentials of Fuzzy Modeling and Control*, John Wiley & Sons, NY, 1994.

[10] A. K. Jain, *Fundamentals of Digital Image Processing*, Prentice-Hall: Englewood Cliffs, NJ, 1989.