A NEW METHOD FOR DIRECTIONAL IMAGE INTERPOLATION

Sadik D. Bayrakeri *Russell M. Mersereau*

Digital Signal Processing Laboratory
School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, GA 30332-0250
Phone: (404) 853-0160 Fax: (404) 894-8363 E-mail: rmm@eedsp.gatech.edu

ABSTRACT

A nonlinear method for image interpolation is presented based on spatial domain directional interpolation. Existing directional interpolation algorithms, which only consider edge regions, are extended to the whole image by interpolating in multiple directions. The interpolated values along various directions are combined using directional weights, which depend on the variation in that direction. The interpolation value for each direction is assigned based on the magnitude of its directional derivative.

1. INTRODUCTION

In the context of pyramid coding schemes, shown in Figure 1, nonlinear interpolation techniques often give superior results compared to linear interpolation. Using a residual error minimization criterion, the high frequency components, which are lost in linear interpolation systems, are partially recovered and a better approximation of the original image is obtained.

A variety of nonlinear techniques have been used to perform image interpolation in a way that preserves sharp edges. One approach uses median filtering [1,2]. An alternative is the set of directional interpolation methods that use *direction estimation* and *directional interpolation* to provide improved performance while adding flexibility to the design criteria. Variations of directional interpolation algorithms come from using different edge models, interpolation strategies, and decision rules for different types of regions. The directional interpolation schemes in both [3] and [4] classify blocks as edgy or smooth. In [3], a continuous-space bilevel edge fit together with an exponential nonlinearity is used for pixel assignment. However, continuous edge fitting is not a precise operation due to the uncertainty and lack of information in regions of high activity and sharp edges. In addition, the exponential edge model-based pixel assignment and unlimited number of directions considered for small window sizes limit the practical application of the algorithm. In [4], bimodal continuous-space edge fitting is replaced by the determination of isointensity lines, orthogonal to the direction of the gradient, but the idea of an unlimited number of directions is retained. The exponential edge model is replaced with a linear mapping from the nearest pixels to the isointensity line. The approach in [5] uses a limited number of interpolation directions.

Previous directional interpolation algorithms consider only edgy regions in a single direction and simple averaging is performed for other regions, which are considered as smooth. In this paper, we present a new scheme that extends the application of directional interpolation to other types of regions. The method uses multiple directions and is discussed in the following sections.

2. LINEAR VERSUS NONLINEAR INTERPOLATION

Linear decimation and linear interpolation are the two basic operators that are used in multirate signal processing. When an antialiasing filter is missing from the decimator, *aliasing* of the high frequency regions occurs, and if the interpolator is to produce a close approximation to the original input, it must deal with the aliasing problem. When an antialiasing filter with a sharp frequency response is used, the overall high frequencies are lost in the decimation stage and the interpolator must try to predict the lost high frequency components. Based on experiments with linear decimators and interpolators for image interpolation by a factor of two in both the horizontal and vertical directions, the minimum interpolation error is observed with

This work was supported in part by the Joint Services Electronics Program under Contract DAAH-04-93-G-0027.

sharp frequency response antialiasing and interpolation filters with cutoff frequencies at $\pi/2$. Changing the cutoff frequencies independently for both filters increases the interpolation error. This simple experiment reveals that aliasing and left over distortions cannot be eliminated by linear decimaton and interpolation. This suggests that *nonlinear* or *spatially varying* interpolation might be preferred. One approach is to use directional interpolation techniques.

The term *directional interpolation* applies to a large number of interpolation schemes. The common feature is the interpolation along the *minimum variation* direction, which is also the locally low frequency direction. The reasoning behind the approach is that along the low frequency direction, the pixel values are more similar to each other. Isotropic averaging might incorporate pixels that deviate substantially from the interpolated pixel value and smooth edges.

Directions that are parallel to edges are low frequency directions. However, downsampling in edgy regions introduces uncertainty about the direction and the exact location of the edge in the original image. As the location of the edge cannot be identified with certainty, a large interpolation error will be observed if the pixel to be interpolated is assigned to the wrong side of the edge. Sharper edges introduce larger potential interpolation error. Smoothing across the edge decreases the peak error magnitude but increases the average interpolation error of the edgy region. The conclusion is that even for an edge region, a precise interpolation direction cannot be defined, but multiple directions can be combined to yield better estimates. Based on local regional variations, there might exist more than one low frequency direction centered on the pixel being interpolated. Combination of these multiple interpolation estimates underlies the proposed multidirectional interpolation scheme.

3. MULTIDIRECTIONAL INTERPOLATION SCHEME

Figure 2 shows the block diagram of the multidirectional interpolation scheme. To each downsampled pixel, eight directional weights are assigned by directional derivative operators D1...D8. (Note that only four operators are shown in the figure for simplicity.) Directional derivative operators are employed to test the variation of the region in a particular direction. For each downsampled pixel, within a 5×5 rectangular window, pixel differences in each direction are evaluated. Inverting and normalizing the absolute difference yields the normalized weights *W1,..., W8*. The higher the variation in a particular direction, the lower the weight that is assigned and vice versa. As the next step, the directional weights obtained for the downsampled image pixels are interpolated to a higher resolution lattice. The estimated directional weights for the two downsampled pixels that straddle the pixel to be interpolated in the appropriate direction are averaged to set the higher resolution weights *WA1,..., WA8.* Figure 3 illustrates the interpolation directions and the combinations used for *weight interpolation* for each position of the interpolated pixel. Notice that we need to consider three different cases, because for image interpolation by a factor of two in the horizontal and vertical directions, three different cases exist depending on whether the pixel to be interpolated lies on a new row and a new column (type A), an established column and a new row (type B), or an established row and a new column (type C).

The main intention of the method is to combine various directional interpolation values in an appropriate way. The directional weights in the previous step were evaluated for this reason. First, interpolation along each possible direction is obtained for the three different types of pixels. Figure 3 also shows the neighboring pixels used for interpolation along each direction. Based on the neighboring masks, six directions are employed for pixels of type A and five directions for pixels of type B and C.

Directional weight dependent interpolation is performed for each direction based on the following reasoning: Considering Figure 3 for pixels of type A, only the *nw* and *sw* directions intersect the nearest neighbors a, b , c, d. Our experimental results show that when there is low directional variation, better results are achieved by using only the nearest neighbors but for more directional variation, it becomes possible to use more distant pixels in the same direction. Two different sets of interpolation values are used depending on whether the directional variation is low or high. These are determined by using a threshold value, T, that is image independent.

The directional interpolation values for pixel type A are given below and for B and C type of pixels, a similar approach is followed.

if WAi < T,

$$\begin{aligned} nnw &\Leftarrow \frac{(a+d)}{2} \\ nw &\Leftarrow \frac{(a+d)}{2} \\ wnw &\Leftarrow \frac{(a+d)}{2} \\ wsw &\Leftarrow \frac{(b+c)}{2} \\ sw &\Leftarrow \frac{(b+c)}{2} \\ ssw &\Leftarrow \frac{(b+c)}{2} \end{aligned}$$

if WAi > T,

$$\begin{aligned} nnw &\Leftarrow \tfrac{(g+r)}{2} \\ nw &\Leftarrow \tfrac{(a+d)}{2} \\ wnw &\Leftarrow \tfrac{(e+t)}{2} \\ wsw &\Leftarrow \tfrac{(f+s)}{2} \\ sw &\Leftarrow \tfrac{(b+c)}{2} \\ ssw &\Leftarrow \tfrac{(p+h)}{2} \end{aligned}$$

In the above formulation, WAi represents one of the six directional weights considered for pixels of type A. For example, to evaluate the *nnw* directional assignment, WAi along the nnw direction is compared to the threshold. For the other directions, the same procedure is repeated.

For sharp edges, the directional weightings around the edge direction dominates and interpolation is performed only parallel to the direction of the edge. For smooth regions, the weights are comparable and the algorithm reduces to simple averaging. For regions with moderate edginess a compromise between these extremes is obtained.

4. EXPERIMENTAL RESULTS

Performance evaluation is based on the relative error between the linear decimation-interpolation model and the nonlinear filter under test, using seven 256×256 images with different features. The absolute error for each scheme is found by evaluating the mean square error between the interpolation and the original image and then the relative error is calculated as given below:

$$\text{Relative Error} = 10\log\left(\frac{NLerror}{Referror}\right) \text{ db}$$

The linear reference model with 25 tap FIR filter and the absolute interpolation errors for seven images are shown in Figure 4. Images with different features are chosen to test the performance of the algorithms for different types of regions. Image CLOWN has fair number of vertical and horizontal edges and corners. LEHAR is a very high frequency image with hardly predictible, repetitive sharp diagonal edges. BRIDGE is a good example of a texture image. FRUIT represents a very lowpass image with very smooth edges. LENNA is an image with small amount of texture and fairly sharp edges. CAMERAMAN is an image with very sharp edges. TOOLS is a simple image with 3 fairly sharp edges.

Considering Figure 5, the largest reference model error is observed for the image LEHAR. Comparable error magnitudes are achieved for the images BRIDGE and CAMERAMAN. Less error is given by the LENNA, CLOWN, FRUIT and TOOLS images in decreasing order. Except for the image FRUIT, which is a very low frequency image with smooth edges and no texture, linear interpolation performance decreases very severely when the antialiasing filter is not present. This is an expected outcome as linear interpolation, which is only the prediction of lowpass component of the original image, cannot eliminate aliasing. The opposite case is observed for directional interpolation, where the antialiasing filter increases the prediction error significantly for the images CLOWN, FRUIT, LENNA, CAMERAMAN and TOOLS while decreasing the error for the barely predictable images LEHAR and BRIDGE. Our experimental results showed that directional interpolation is successfully applied except in highly textured regions where linear interpolation with an antialiasing filter gives the best performance.

Based on experiments when no antialiasing filter is used, directional interpolation outperforms the linear scheme and the median-based schemes [1][2] for the image LENNA, CAMERAMAN, and TOOLS. The same performance is obtained for image FRUIT, in which case the errors are very small and most of the algorithms have similar error magnitudes. For LEHAR and BRIDGE, which are very hard to interpolate, the reference model outperforms all the other algorithms although directional interpolation performs slightly better than median-based interpolation.

The results are summarized as follows:

- When no antialiasing filter is used, directional interpolation performs better than linear interpolation for both smooth and highly detailed images. (1.5 – 3 db relative error reduction)
- When an antialiasing filter is used, linear interpolation performance is better for very high detailed images(up to 0.6 db) whereas the directional filter is better for other cases. (0.5 – 3 db)
- If the relative errors between the above cases are considered, the best strategy is to use linear interpolation with an antialiasing filter for unpredictable and highly detailed texture images and to use directional interpolation without an antialiasing filter for the other cases.
- For sharp edges, directional interpolation performs much better than median-based algorithms [1][2]. (0.6 – 0.8 db)

5. CONCLUSION

We have described a multidirectional interpolation scheme. By using multiple directions, directional interpolation algorithms are extended to treat regions other than smooth areas and simple edge. Based on our earlier experimental results, it would appear that an effective method for reducing interpolation error would be a combination of nonlinear decimation and the directional interpolation. However, although an average 1 - 1.25 db percent relative error reduction has been achieved, it remains unclear whether the increased computational complexity justifies improvement over the linear scheme.

6. REFERENCES

[1] Zeng B. , Venetsanopoulos A. N., "A Comparative Study of Several Nonlinear Image Interpolation Schemes", *Proceedings of SPIE v 1818*, 1992, pp 21-29

[2] Lehtonen A. , Renfors M., "Nonlinear Quincunx Interpolation Filtering", *Proceedings of SPIE v 1360*, 1990, pp 135-142 142

[3] Jensen K., Anastassiou D., "Spatial Resolution Enhancement of Images using Nonlinear Interpolation", *Proc. ICASSP*, 1990, pp 2045-2048

[4] Algazi V. R., Gary E. F., "Directional Interpolation of Images based on Visual Properties and Rank Order Filtering", *Proc. ICASSP*, 1991, pp 3005-3008

[5] Ho Y. S., Gersho A., "Variable Rate Contour based Interpolative Vector Quantization for Image Coding", *IEEE Global Telecommunications Conference and Exhibition*, 1988, pp 750-754

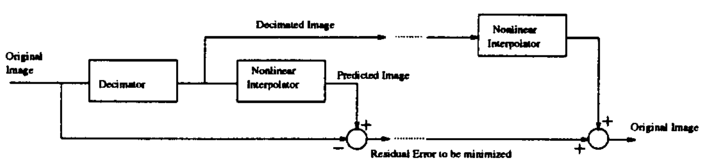


Figure 1: One stage pyramid coder

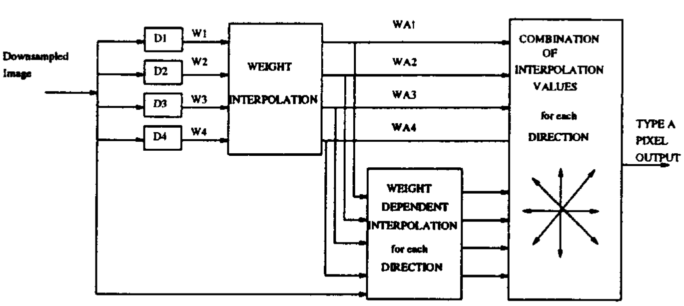


Figure 2: Multidirectional interpolation scheme

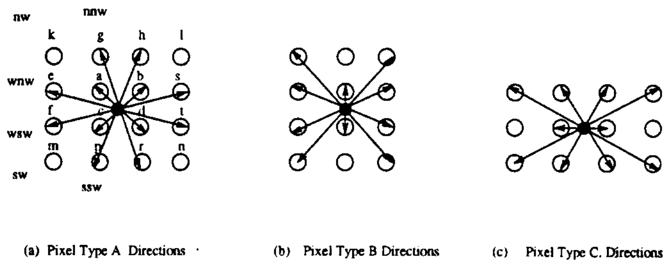
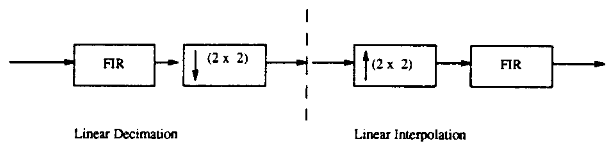


Figure 3: Combinations used for weight interpolation



Reference Model

CLOWN	LEHAR	BRIDGE	FRUIT	LENNA	CAMERA	TOOLS
4.1 e+6	9.96 e+7	1.08 e +7	3.62 e+6	5.43 e+6	1.09 e+7	1.02 e+6

Reference Model Error Magnitudes

Figure 4: Reference model and error magnitudes

CLOWN	LEHAR	BRIDGE	FRUIT	LENNA	CAMERA	TOOLS
1.08	2.75	1.89	-0.12	1.39	1.55	0.56

Decimation without antialias filter and linear interpolation with 25 tap FIR filter

CLOWN	LEHAR	BRIDGE	FRUIT	LENNA	CAMERA	TOOLS
0.38	0.27	0.50	-2..93	-0.45	0.067	-0.91

Decimation with antialias filter and directional interpolation

CLOWN	LEHAR	BRIDGE	FRUIT	LENNA	CAMERA	TOOLS
-0.08	2.3	0.71	-4.43	-0.91	-0.17	-1.93

Decimation without antialias filter and directional interpolation

Figure 5: Relative errors in db related to antialias filter affects