NEW IMAGE INTERPOLATION TECHNIQUES

D. Darian Muresan and Thomas W. Parks

School of Electrical and Computer Engineering
Cornell University, Ithaca NY. 14853
darian, parks@ee.cornell.edu

ABSTRACT

Image interpolation is an important tool in digital image processing. In this paper we introduce and review four new approaches to image interpolation. Two of the methods are wavelet domain interpolation and two are data adaptive, directional interpolation.

1. INTRODUCTION

As digital photography becomes ever more popular with consumer electronics, the need for good interpolation techniques is essential. From printing large images to zooming and cropping, interpolation is at the heart of it all. Not only that, but good insight into interpolation techniques also helps in developing better tools in other areas of image processing, such as image denosing and image encoding.

2. DATA ADAPTIVE DIRECTIONAL INTERPOLATION

Data adaptive interpolation is based on the fact that interpolation works best when it's done along edges and not across. In this category, we present two approaches. The first method is based on work done by Kimmel [3]. We have taken their method, used for Color Filter Array (CFA) interpolation and adapted it to gray scale image interpolation. The second directional approach described in this paper is identical to the one presented by Li [2].

In our first directional interpolation method, we calculate the likelihood that each pixel belongs to an edge based on the directional derivatives. Then interpolation is such that more weight is given to the pixels lined up along the edge than the pixels across the edge. For example, in Fig. 1, with E a directional edge measure, we would interpolate pixel $P_{(m,n)}$ as follows:

$$P_{(m,n)} = \frac{\sum_{(i,j)\in D} P_{(m+i,n+j)} E_{(m+i,n+j)}}{\sum_{(i,j)\in D} E_{(m+i,n+j)}} \qquad (1)$$

This work was supported by Kodak and NSF MIP9705349

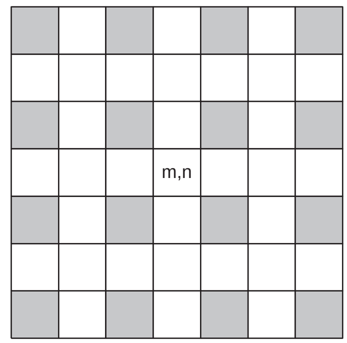

Figure 1: Interpolate pixel $P_{(m,n)}$.

with $D = \{(-1,-1),(-1,1),(1,-1),(1,1)\}$ when the corner pixels are known; and

$$P_{(m,n)} = \frac{\sum_{(i,j)\in S} P_{(m+i,n+j)} E_{(m+i,n+j)}}{\sum_{(i,j)\in S} E_{(m+i,n+j)}} \qquad (2)$$

with $S = \{(-1,0),(0,-1),(0,1),(1,0)\}$ when the side pixels are known. To define our edge measure, we need some notation. From Fig. 1 we define our gradients at pixel $P_{(m,n)}$, in the $x, y, x-diagonal$, and $y-diagonal$ directions as follows:

$$\begin{aligned} D_x(P_{(m,n)}) &= \frac{P_{(m,n-1)} - P_{(m,n+1)}}{2} \\ D_y(P_{(m,n)}) &= \frac{P_{(m-1,n)} - P_{(m+1,n)}}{2} \\ D_{xd}(P_{(m,n)}) &= \frac{P_{(m-1,n+1)} - P_{(m+1,n-1)}}{2\sqrt{2}} \\ D_{yd}(P_{(m,n)}) &= \frac{P_{(m-1,n-1)} - P_{(m+1,n+1)}}{2\sqrt{2}} \end{aligned}$$

With the above gradients, E, is defined as:

$$E_{(m+i,n+j)} = \left(\sqrt{1 + D(P_{(m,n)})^2 + D(P_{(m+i,n+j)})^2}\right)^{-1}, \qquad (3)$$

where D is the difference in the direction of $P_{(m+i,n+j)}$.

For example,

$$E_{(m-1,n-1)} = \sqrt{1 + D_{yd}(P_{(m,n)})^2 + D_{yd}(P_{(m-2,n-2)})^2}^{\,-1}.$$

Notice that we used $D_{yd}(P_{(m-2,n-2)})^2$ since $D_{yd}(P_{(m-1,n-1)})^2$ is not known and we are assuming that the two are equal.

This method requires two passes: in the first pass we interpolate the pixels that have known values at their corners and in the second pass we interpolate the pixels that have known values at their sides. This type of interpolation smoothes along edges and tends to introduce some patchiness around textured areas. The results are not included in this paper, as the printing process diminishes the ability to observe differences. Results may be found on line at www.ee.cornell.edu/˜splab.

The second directional method is the method described in [2]. Just like our previous method, it too tries to interpolate along edges and not across them. Equations (1) and (2) are simplified to:

$$P_{(m,n)} = \sum_{(i,j)\in D} P_{(m+i,n+j)} E_{(m+i,n+j)} \qquad (4)$$

when the corner pixels are known; and

$$P_{(m,n)} = \sum_{(i,j)\in S} P_{(m+i,n+j)} E_{(m+i,n+j)} \qquad (5)$$

when the side pixels are known. The weights $E_{(m+i,n+j)}$ are derived from a least square problem formulation as follows:

1. Assume that within a fixed, small region (call it M), $E_{(m+i,n+j)}$ is constant for each (i, j).

2. Using the **known** pixel values set up a linear system of equations based on the above assumption to find $E_{(m+i,n+j)}$ via a mean square error solution.

3. Use the $E_{(m+i,n+j)}$ values found in step 2 to interpolate for the missing values via equation (4) or (5).

For example, let's assume that we want to interpolate pixel $P_{(m,n)}$ using the four corner neighboring pixels. For our step (1) we assume that our neighborhood M is 5 pixels by 5 pixels, as shown in Fig. 1. The known pixels are all the gray pixels in Fig. 1. For our second step, the linear system of equation that would be set up from the known values would then be equation (6). Solving equation (6) gives us the E weights, which can then be used in equation (4) to find the missing pixel. In this method, there are a few things to be noted:

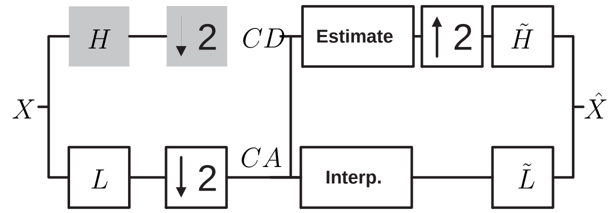

Figure 2: Problem model: given CA estimate CD.

1. Matrix C has more (or equal) rows than columns.

2. Matrix C may become singular in uniform regions of the image. In that case, take $E = 1/4$. If C is full rank, equation (6) can be solved via Least Squares: $E = (C^T C)^{-1} C^T P$.

This interpolation gives similar results to the first directional approach. In the textured regions the results look as if they were smoothed using the anisotropic approach from [3]. The results are also similar to a known commercial interpolation package by Altamira, Genuine Fractals. The details of Altamira's interpolation are not known at this time. Again, images are included on the web site mentioned above.

3. WAVELET BASED, SCALE INTERPOLATION

In this second part of our paper, we shift gears from directional interpolation to wavelet based detail prediction over scale. In the wavelet based interpolation, our problem model is as in Fig. 2. We assume that the original image has been passed through a low pass filter, such as the camera lens, and that the only available data is the low passed and decimated data CA. We also assume that the filter bank is a perfect reconstruction filter bank, although that is not a requirement of the interpolation algorithm. Similar work has been done in [4, 5]. Both [4, 5] estimated the detail coefficients based on the undecimated wavelet extrema of the CA coefficients.

Our approach is slightly different. First, we went back a step and assumed that we had the undecimated CA coefficients. Using the theory of optimal recovery [1, 6] we estimated the undecimated CD coefficients from the undecimated CA coefficients. Excellent results were obtained not only for the large CD coefficients around edges, but also in smooth regions, as shown in Fig. 3. Notice that in the middle of the array, where filtering blurred the image, the estimation properly recovered the lost details, whereas unsharp masking only slightly sharpened the image.

$$\underbrace{\begin{bmatrix} P_{(m-1,n-1)} \\ P_{(m-1,n+1)} \\ P_{(m+1,n-1)} \\ P_{(m+1,n+1)} \end{bmatrix}}_{P} = \underbrace{\begin{bmatrix} P_{(m-3,n-3)} & P_{(m-3,n+1)} & P_{(m+1,n+1)} & P_{(m+1,n-3)} \\ \vdots & \vdots & \vdots & \vdots \\ P_{(m-1,n-1)} & P_{(m-1,n+3)} & P_{(m+3,n+3)} & P_{(m+3,n-1)} \end{bmatrix}}_{C} \underbrace{\begin{bmatrix} E_{(m-1,n-1)} \\ E_{(m-1,n+1)} \\ E_{(m+1,n-1)} \\ E_{(m+1,n+1)} \end{bmatrix}}_{E} \quad (6)$$

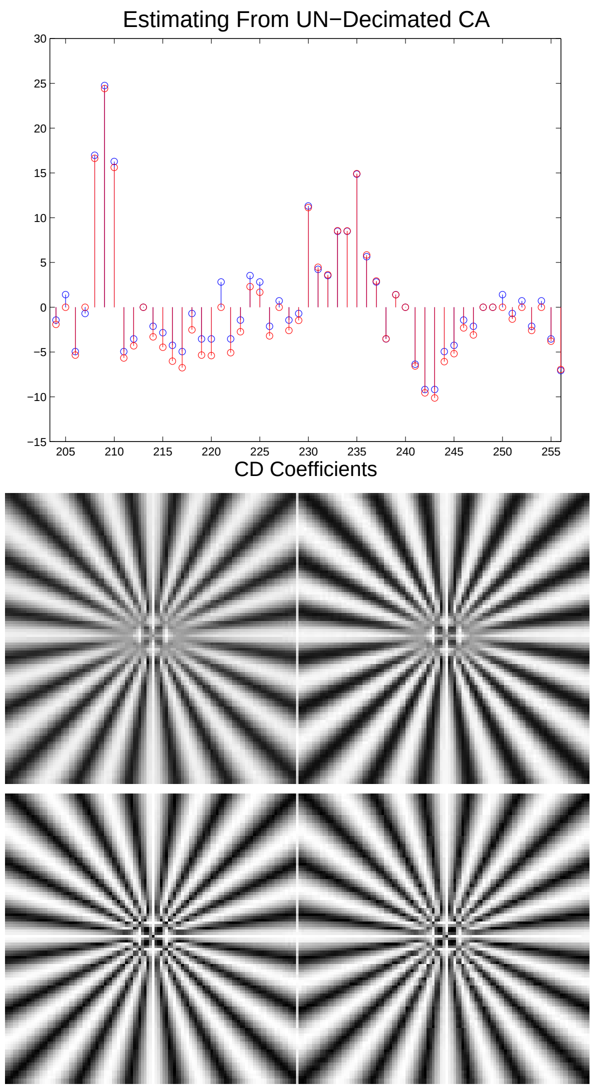

Figure 3: Plot of the CD and estimated $\hat{CD}$ coefficients using **undecimated** CA (blue are the actual CD coefficients) (top). Filtered image and sharpened image using unsharp masking (middle). Original image and reconstructed image using the estimated $\hat{CD}$ from undecimated CA (bottom).

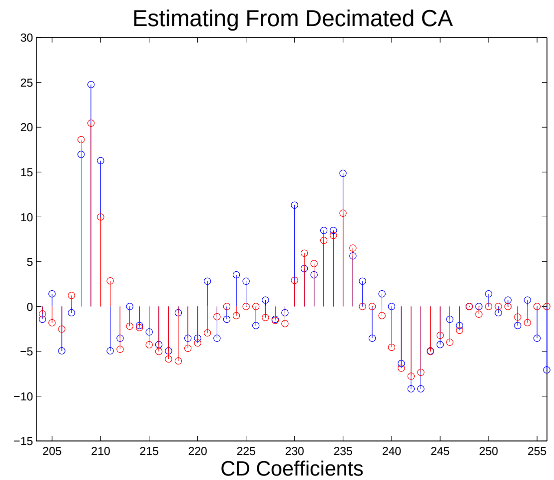

Figure 4: Coefficients CD and the estimated $\hat{CD}$ coefficients using the CA coefficients (blue are the true CD coefficients).

With the sharpening idea in mind, we return to the original problem. Assuming that only the decimated CA coefficients are given, we first interpolate to obtain the undecimated CA coefficients and then use optimal estimation to determine the CD coefficients. Simply using a linear interpolation filter for interpolating CA generated reasonable results in estimating the CD coefficients, as shown in Fig. 4. For the most part, the images interpolated using this method seem to be sharper and visually closer to the original image than the directional algorithms were. Around edges, small ringing artifacts may sometimes be noticed. Again, visual images are provided on the above mentioned web site.

Our final wavelet based interpolation technique is again based on the model of Fig. 2. Unlike our previous method, where we tried to estimate the actual CD coefficients, in this approach, we only try to guess what CD coefficients would give us sharp edges. The approach is actually very simple, as depicted in Fig. 5. First step is to enlarge the image using pixel replication, or any other edge enhancing/preserving type interpolation. The reason for this step is so that edges do not get blurred and on the contrary, may even be enhanced. After the image is enlarged using pixel replica-

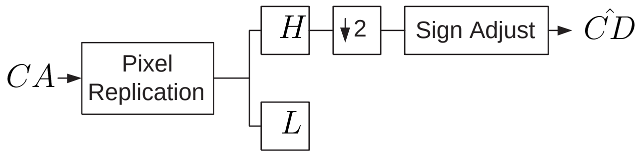

Figure 5: Generating $\hat{CD}$ coefficients that enhance edges.

tion, we do a wavelet decomposition to obtain the detail coefficients. We keep only the large detail coefficients, since these are indications of edges. Experimentally, we noticed that while the $\hat{CD}$ wavelet coefficients are of the right magnitude and at the right location, the sign is not always correct. Using a simple non-linear sign adjust procedure we adjust the signs of the $\hat{CD}$ coefficients so as to eliminate ripples in the output signal. The idea is that a large detail coefficient of wrong sign, will generate a ripple in the edge and we try to minimize that.

This interpolation procedure does sharpen up edges and it does very well around objects that have straight or vertical lines. However, since this interpolation approach is separable, curved edges tend not to do as well. Again, the results are available on line.

4. CONCLUSION

In this paper, we reviewed four image interpolation techniques, which seem to generate reasonably high quality images, which seem to be an improvement over bi-cubic interpolation. The wavelet based interpolation approaches were implemented using separable 1-D processing. It is our belief that these results can be further improved, especially around curved edges, by looking at 2-D, non-separable processing.

5. REFERENCES

[1] D. Darian Muresan and Thomas W. Parks, “Prediction of Image Detail,” *IEEE ICIP,* Vancouver, Sept. 2000.

[2] Xin Li and Michael T. Orchard, “New Edge Directed Interpolation,” *IEEE International Conference on Image Processing,* Vancouver, Sept. 2000.

[3] Ron Kimmel, “Demosaicing: Image Reconstruction from Color CCD Samples,” *IEEE Transactions on Image Processing,* Vol. 8, No. 9, (September 1999), pp. 1221-1228.

[4] W. Knox Carey, Daniel B. Chuang and Sheila S. Hemami, “Regularity-Preserving Image Interpolation,” *IEEE Transactions on Image Processing,* Vol. 8, No. 9 (Sept. 1999), 1293-1297.

[5] S. Grace Chang, Zoran Cvetkovic, and Martin Vetterli, “Resolution Enhancement of Images Using Wavelet Transform Extrema Interpolation,” *IEEE ICASSP,* (May 8-12, 1995), pp. 2379-2382.

[6] M. Golomb and H. F. Weinberger, “Optimal approximation and Error Bounds,” *On Numerical Approximation,* R. E. Langer ed., The University of Wisconsin Press, Maddison, pp. 117-190, 1959.