Adaptively Quadratic (AQua) Image Interpolation

D. Darian Muresan, *Member, IEEE,* and Thomas W. Parks, *Fellow, IEEE*

***Abstract*—Image interpolation is a key aspect of digital image processing. This paper presents a novel interpolation method based on optimal recovery and adaptively determining the quadratic signal class from the local image behavior. The advantages of the new interpolation method are the ability to interpolate directly by any factor and to model properties of the data acquisition system into the algorithm itself. Through comparisons with other algorithms it is shown that the new interpolation is not only mathematically optimal with respect to the underlying image model, but visually it is very efficient at reducing jagged edges, a place where most other interpolation algorithms fail.**

***Index Terms*—Image modeling, interpolation, quadratic classes.**

I. Introduction

With the advent and proliferation of low resolution digital cameras, such as those found in today's cell phones, there is a dire need for good image interpolation techniques. The main focus of this paper is the introduction of a novel image interpolation technique that is based on optimal recovery and adaptively determining the local quadratic signal class.

This paper is organized as follows. Section II discusses several published image interpolation algorithms. Section III details the adaptively quadratic (AQua) image interpolation algorithm. Section IV compares the performance of the adaptively quadratic interpolation against four other interpolation techniques. Section V concludes with final remarks and future research work in this area. Finally, the Appendix reviews the theory of optimal recovery [1], [2], which is key to the problem of interpolating missing samples in a quadratic signal class.

II. Review

In the area of image interpolation by far the most well known and widely used techniques are those of polynomial or Lagrange interpolation and splines. These image models are based on the assumption that locally each image behaves like an n^{th} degree polynomial. Whether separable or nonseparable, these methods can be efficiently implemented using an up-sampler followed by a filter. Examples of such interpolation algorithms are cubic [3] and other spline based methods [4]. These polynomial image models have the advantage of being fast but tend to introduce serious jaggedness (the staircase effect) and blur. They are the choice of image manipulation programs such as Adobe Photoshop and GIMP.

There have been many attempts at improving the local polynomial image models in order to enhance edges and the overall image sharpness. In [5] the author introduces the concept of warped distances to adaptively adjust the bi-cubic interpolation filter. By assuming a different relative location (a warped distance) of the four known samples with respect to the interpolated sample, the filter coefficients can be changed in order to sharpen edges. It is not clear how, or even if, this method removes the staircase affect in curved edges. The image models of [6], [7] use splines for image resizing. Their methods are especially useful for down-sampling, while up-sampling is similar to spline interpolation.

Other attempts at modifying the polynomial image model in order to reduce jaggedness and sharpen edges are those of [8] and [9]. Around edges the authors of [8] map a 3 by 3 neighborhood about each pixel in the low-resolution image to a best-fit continuous space step edge and then re-sample it at the higher density. Images interpolated with this method look sharper than bi-cubic interpolation but often look too much like drawings, especially for zoom factors of four or more. In [9] the authors use an iterative rendering and correction step for edge directed interpolation and claim that this produces sharper images.

A second class of image interpolation algorithms are those based on multi-resolution analysis. The authors of [10] model the wavelet coefficients of a dyadic (nondecimated) wavelet transform using exponential decay. In particular, they show that the local maximum of the dyadic wavelet coefficients decreases exponentially from coarse to fine scales. The authors of [11], [12] apply the wavelet exponential decay model to image interpolation by posing the interpolation problem as one of estimating the fine detail wavelet coefficients. Exponential decay is estimated from the coarse scale coefficients. If an exponential decay is detected, an estimate of the fine scale wavelet coefficient is made based on the estimated decay, otherwise the estimate is zero.

The super-resolution image models of [13], [14] can also be described as interpolation methods based on adding image details in the wavelet domain, although the details are added based on training data and not wavelet decay. Using high resolution training data, images are decimated and table lookups of low-resolution/high-resolution patches are built. These methods produce sharper images and can work well, given decent training data, but their main drawback is the potential for introducing artifacts when the table lookup procedure chooses a wrong high resolution patch. In addition, the image models may not work as well for data containing artifacts, such as JPEG compressed images, as the methods tend to consider artifacts as part of the

Manuscript received February 3, 2003; revised October 13, 2003. This work was supported by Kodak, NSF, and TI. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Thierry Blu.

D. D. Muresan is with Digital Multi-Media Design, Arlington, VA 22209 USA (e-mail: darian@dmmd.net).

T. W. Parks is with Electrical and Computer Engineering Department, Cornell University, Ithaca, NY 14853 USA (e-mail: parks@ece.cornell.edu).

Digital Object Identifier 10.1109/TIP.2004.826097

1057-7149/04$20.00 © 2004 IEEE

image and may enhance the artifacts more than the image itself. The image models of [15] are similar in nature.

The authors of [16] model the wavelet coefficients using Gaussian mixtures and apply their models to image denoising [17]. In [18] the image model of [16] is extended to image interpolation. For this interpolation method the interpolation results are comparable with bi-cubic interpolation. One particular feature of this approach is that the detail wavelet coefficients are realizations of estimated Gaussian mixtures. Hence, every run of the algorithm produces a different result. The weakness of this model, in image interpolation, lies primarily in the determination of the Gaussian mixture parameters.

The authors of [19] propose a maximum a posteriori (MAP) pixel estimation technique which results in the optimization of convex functionals. Their nonlinear image expansion technique sharpens edges but the interpolation technique does not eliminate jagged edges. The work of [20] also uses a MAP framework for estimating a high resolution image from a sequence of under-sampled images. Their expansion method could be used for a single image by assuming only one frame. Applying the algorithm to a single frame results in images very similar to those of [19].

In [21] the authors present a least squares edge directed interpolation method. The method assumes that each pixel is a linear combination of its neighboring pixels. Further, it is assumed that locally the weights are constant. A linear system of equations is solved in order to find the local weights. This method works for interpolation by factors of two and performs well around edges, but performs poorly in high frequency regions, sometimes introducing undesired artifacts. In [22] it is shown how this solution can be reformulated using optimal recovery which allows for additional assumptions about the local derivatives in addition to the known local pixels to be used in the interpolation process.

The work of [23] is related to our image interpolation approach. The authors pose the image interpolation problem as one where the image belongs to a fixed quadratic image class. To solve the interpolation problem the authors add some known linear partial differential constraints. The solution to their interpolation problem is similar to the solution of our interpolation problem. The difference is that this paper develops adaptive quadratic signal classes to better model the local image behavior.

This paper presents a novel approach to image interpolation based on optimal recovery (Appendix) and the adaptive determination of the local quadratic signal class. The new interpolation method generalizes well to interpolation by any factor, removes jagged edges, and can easily incorporate a model for the camera lens in order to produce sharper results.

III. Adaptively Quadratic (AQua) Interpolation

The first challenge of using optimal recovery for image interpolation is determining the quadratic signal class K:

$$K = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{x}^T\mathbf{Q}\mathbf{x} \leq \epsilon\}$$

from a set of training data. The training data is usually taken from the local features of the image and selecting a proper training set is discussed at the end of this section. For now assume that a training set of patches $\mathcal{S} = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$ representative of the local data is given for estimating the local quadratic signal class. The $\mathbf{Q}$ for which the ellipsoid

$$\mathbf{x}^T\mathbf{Q}\mathbf{x} \leq \epsilon, \tag{1}$$

for some constant ϵ must be representative of the training set $\mathcal{S}$. In other words $\mathbf{Q}$ must be a matrix such that when an image patch $\mathbf{y}$ is similar to the vectors in $\mathcal{S}$ then (1) holds for $\mathbf{y}$. Let matrix S be formed by arranging the image patches in $\mathcal{S}$ as columns:

$$S = (\mathbf{x}_1, \ldots, \mathbf{x}_m) \tag{2}$$

and consider the equation relating the image patch $\mathbf{y}$ to the training set $\mathcal{S}$ using a column of m weights, $\mathbf{a}$:

$$S\mathbf{a} = \mathbf{y}. \tag{3}$$

Vector $\mathbf{y}$ is similar to the vectors in $\mathcal{S}$ when $\mathbf{a}$ has small energy. Using standard notation for singular value decomposition of S, (3) can be rewritten as:

$$U\Lambda V^T\mathbf{a} = \mathbf{y}$$

The weight vector $\mathbf{a}$ is given by

$$\mathbf{a} = V\Lambda^{-1}U^T\mathbf{y}$$

and the sum of the squares, or the energy of $\mathbf{a}$ is:

$$\mathbf{a}^T\mathbf{a} = \mathbf{y}^T U\Lambda^{-2}U^T\mathbf{y}.$$

Since

$$SS^T = U\Lambda^2U^T$$

it follows that

$$\begin{aligned}\mathbf{a}^T\mathbf{a} &= \mathbf{y}^T(SS^T)^{-1}\mathbf{y} \\ &= \mathbf{y}^T\mathbf{Q}\mathbf{y}\end{aligned}$$

where $\mathbf{Q}$ is the pseudo inverse of SS^T. If $\mathbf{y}$ is very similar to the training set $\mathcal{S}$ then

$$\mathbf{a}^T\mathbf{a} \leq \epsilon,$$

and with the new $\mathbf{Q}$ it follows that

$$\mathbf{y}^T\mathbf{Q}\mathbf{y} \leq \epsilon.$$

This is the desired form for our ellipsoidal signal class. It results from an "Occam's razor" type of assumption that small weights are used to represent vectors that are similar to our training set $\mathcal{S}$.

Given the formulation of the quadratic signal class from a training set $\mathcal{S}$, or equivalently the formulation of $\mathbf{Q}$, the next challenge is determining the training set $\mathcal{S}$. One direct approach of selecting the vectors in $\mathcal{S}$ is based on the proximity of their locations to the position of the vector being modeled. In this case, patches are generated from the local neighborhood. A second approach is to use patches from other high density images. This works well when interpolating images that belong to a certain predetermined class. For example in [24] AQua interpolation is

applied to face interpolation and the training set is determined from other high resolution faces. A third approach is to adaptively search for training patches in other high resolution images. This paper uses the first approach where training patches are selected from the local neighborhood. For example, in Fig. 1 to model the quadratic signal class that the center patch

$$\mathbf{x} = \left[x_{(2,2)},\, x_{(2,3)},\, x_{(2,4)},\, x_{(2,5)},\, x_{(3,2)},\, \ldots,\, x_{(5,5)}\right]^T \quad (4)$$

belongs to, let

$$\mathcal{S} = \left\{ \begin{pmatrix} x_{(0,0)} \\ x_{(0,1)} \\ \vdots \\ x_{(3,3)} \end{pmatrix}, \ldots, \begin{pmatrix} x_{(4,4)} \\ x_{(4,5)} \\ \vdots \\ x_{(7,7)} \end{pmatrix} \right\}, \quad (5)$$

where $\mathcal{S}$ is formed by choosing all the possible 4×4 image blocks in the 8×8 region of Fig. 1.

A. Interpolation

Adaptively quadratic interpolation is performed in three steps. A block diagram is shown in Fig. 2 and the detailed steps are described next.

1) Determine High Density Class K: The training set used for determining the local quadratic signal class of the high density image (i.e. the interpolated image) is obtained by taking image patches from the local neighborhood. In the case of image interpolation the neighboring image patches contain missing samples. There are several approaches to handle this situation. The first approach is to use patches from the decimated image. For example, in Fig. 1 where the interpolation factor is $2\times$, instead of forming $\mathcal{S}$ as in (5), the set is formed using decimated patches:

$$\mathcal{S} = \left\{ \begin{pmatrix} x_{(0,0)} \\ x_{(0,2)} \\ \vdots \\ x_{(6,6)} \end{pmatrix}, \ldots \right\}. \quad (6)$$

While this approach works well for small interpolation factors, the approach quickly deteriorates when interpolating by larger factors. For larger interpolation factors the size of training patches is larger and using decimated patches quickly takes away from the locality of the method. Instead of using decimated training patches an alternative method is to interpolate the training patches before forming $\mathcal{S}$. Using interpolated patches in determining the quadratic signal class K is *almost* as good as using original high density patches. This can be explained as follows. Let matrix $\mathbf{Q}_o$ be formed using original high density patches, and matrix $\mathbf{Q}_i$ be formed using interpolated patches. To compare the two quadratic classes we must compare[1] $\mathbf{Q}_o^{-1}$ and $\mathbf{Q}_i^{-1}$. A direct way is to look at the norm[2] of the difference between $\mathbf{Q}_o^{-1}/\|\mathbf{Q}_o^{-1}\|$ and $\mathbf{Q}_i^{-1}/\|\mathbf{Q}_i^{-1}\|$. Table I shows the norms of the error matrix for three different patches of *lena* when interpolation is pixel replication and the interpolation factors are 2, 4, and 8. In all cases the norm of the error matrix is less than about two percent, suggesting a good fit between the quadratic signal classes generated by $\mathbf{Q}_o$ and $\mathbf{Q}_i$.

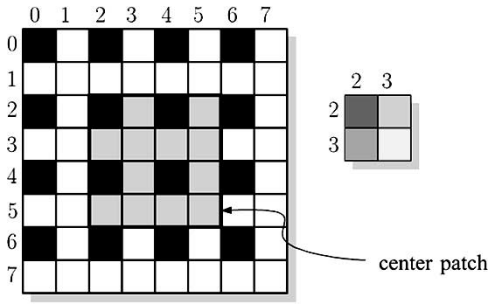


Fig. 1. Local high density image used for selecting $\mathcal{S}$ to estimate the quadratic class for the center 4×4 patch (dark pixels are part of the decimated image). To model the image acquisition system our assumption is that the high density image has been filtered by a low pass filter before down-sampling. For example, the pixel at location (2,2) is the average of the four high density pixels shown on the right.

A second method of comparing the quadratic classes generated by $\mathbf{Q}_o$ and $\mathbf{Q}_i$ is to look at the correlation coefficients between the eigenvectors corresponding to the smallest eigenvalues of $\mathbf{Q}_o$ and $\mathbf{Q}_i$ (the largest eigenvalues of $\mathbf{Q}_i^{-1}$ and $\mathbf{Q}_o^{-1}$). If the correlation is close to one then the quadratic signal classes are *almost* the same. Table II depicts the first three correlation coefficients for three different patches of *lena*. In all cases the first eigenvector of $\mathbf{Q}_i$ (the vector corresponding to the largest eigenvalue of $\mathbf{Q}_i^{-1}$) correlates very strongly with the first eigenvector of $\mathbf{Q}_o$. Although the correlation coefficient is smaller for the second and third eigenvectors, the second and third eigenvalues are two orders of magnitude larger than the first. This suggests that the quadratic classes generated by $\mathbf{Q}_i$ and $\mathbf{Q}_o$ are very similar.

Table II also depicts two other trends. First, as expected, the correlation coefficient decreases as the scaling factor increases. This behavior does not depend solely on scaling factors but it also depends on the level of detail present in the decimated image. Starting with the 512×512 image of *lena* and interpolating by factors of 8 (horizontally and vertically) produces naturally good visual results. Decimating the 512×512 image by 8 and then interpolating back to 512×512 produces bad visual results, while the image continues to maintain nonjagged edges. Second, the correlation coefficient is smaller for textured regions than for well defined edge regions, suggesting that in textured regions more error is introduced. On the other hand, textured regions are also more forgiving from a visual score since human eyes are not as sensitive to errors made in these regions.

The correlation results shown in Table II are for the case when the initial interpolation step of Fig. 2 is pixel replication. The results are slightly better for cubic and linear interpolation. As it will be shown in the Results Section, if the iteration step of Fig. 2

[1]In optimal recovery $\mathbf{Q}^{-1}$ is used for interpolation (see Appendix). Alternatively, the quadratic class K is stretched in the direction of the eigenvector corresponding to the largest eigenvalue of $\mathbf{Q}^{-1}$. This can be understood by diagonalizing matrix $\mathbf{Q}$. That is $\mathbf{x}^T\mathbf{Q}\mathbf{x} = \epsilon$ becomes $\mathbf{x}^T V \Lambda V^T \mathbf{x} = \epsilon$. When $\mathbf{x}$ is in the direction of eigenvector $\mathbf{v}_i$ (i.e. $\mathbf{x} = \alpha_i \mathbf{v}_i$) its squared norm is: $\alpha_i^2 = \epsilon/\lambda_i$. Therefore, a small λ_i (i.e. a large eigenvalue of $\mathbf{Q}^{-1}$) causes a large stretch in the ellipsoid $\mathbf{x}^T\mathbf{Q}\mathbf{x} = \epsilon$.

[2]The norm of a matrix is its largest singular value [25].

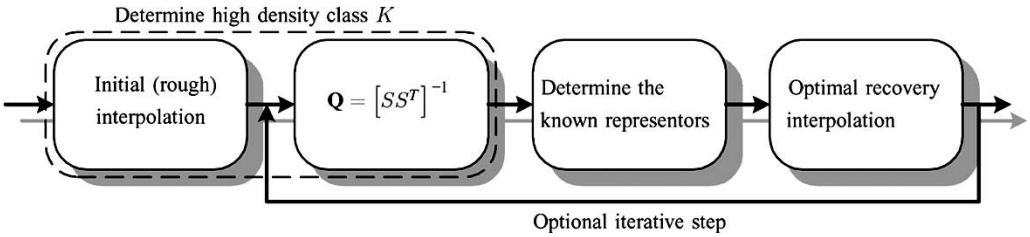


Fig. 2. Interpolation steps performed for each local region.

TABLE I

THREE HIGH DENSITY PATCHES (36 × 36) FROM THE *lena* IMAGE DEPICT: AN EDGE, A STRIPE, AND A TEXTURED REGION. THE HIGH DENSITY PATCH IS DECIMATED BY 2, 4, AND 8 AND THEN INTERPOLATED USING PIXEL REPLICATION. USING 4 × 4 PATCHES, MATRICES $\mathbf{Q}_i$ AND $\mathbf{Q}_o$ ARE BUILT. SHOWN IS THE MATRIX NORM OF THE DIFFERENCE OF THE NORMALIZED MATRICES: $\|(1/\|\mathbf{Q}_o^{-1}\|)\mathbf{Q}_o^{-1} - (1/\|\mathbf{Q}_i^{-1}\|)\mathbf{Q}_i^{-1}\|$

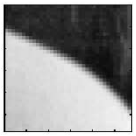
	$\|\frac{1}{\|\mathbf{Q}_o^{-1}\|}\mathbf{Q}_o^{-1} - \frac{1}{\|\mathbf{Q}_i^{-1}\|}\mathbf{Q}_i^{-1}\|$		
High Density Patch	Pix. Rep. Int. 2X	Pix. Rep. Int. 4X	Pix. Rep. Int. 8X
	0.0013	0.0035	0.0087
	0.0041	0.0111	0.0147
	0.0062	0.0155	0.0267

TABLE II

THREE HIGH DENSITY PATCHES (36 × 36) FROM THE *lena* IMAGE DEPICT: AN EDGE, A STRIPE, AND A TEXTURED REGION. THE HIGH DENSITY PATCH IS DECIMATED BY 2, 4, AND 8 AND THEN INTERPOLATED USING PIXEL REPLICATION. USING 4 × 4 PATCHES, MATRICES $\mathbf{Q}_i$ AND $\mathbf{Q}_o$ ARE BUILT. SHOWN IS THE CORRELATION COEFFICIENT (ρ) BETWEEN THE EIGENVECTORS CORRESPONDING TO THE SMALLEST EIGENVALUES OF $\mathbf{Q}_i$ AND $\mathbf{Q}_o$, AND THE INVERSE OF THE SMALLEST EIGENVALUES OF $\mathbf{Q}_i$ (DENOTED BY e_i^{-1}) AND $\mathbf{Q}_o$ (DENOTED BY e_o^{-1})

High Density Patch	Pix. Rep. Int. 2X			Pix. Rep. Int. 4X			Pix. Rep. Int. 8X		
	ρ	e_i^{-1}	e_o^{-1}	ρ	e_i^{-1}	e_o^{-1}	ρ	e_i^{-1}	e_o^{-1}
	0.99	4.03	4.08	0.99	3.93	4.08	0.99	4.77	4.08
	0.99	0.03	0.04	0.98	0.04	0.04	0.93	0.05	0.04
	0.98	0.01	0.01	0.47	0.01	0.01	0.00	0.03	0.01
	ρ	e_i^{-1}	e_o^{-1}	ρ	e_i^{-1}	e_o^{-1}	ρ	e_i^{-1}	e_o^{-1}
	0.99	2.44	2.51	0.96	2.32	2.51	0.93	2.67	2.51
	0.99	0.05	0.06	0.98	0.04	0.06	0.97	0.04	0.06
	0.98	0.01	0.01	0.03	0.02	0.01	0.00	0.03	0.01
	ρ	e_i^{-1}	e_o^{-1}	ρ	e_i^{-1}	e_o^{-1}	ρ	e_i^{-1}	e_o^{-1}
	0.95	1.33	1.34	0.88	1.36	1.34	0.87	1.74	1.34
	0.99	0.03	0.04	0.99	0.02	0.04	0.99	0.05	0.04
	0.97	0.01	0.01	0.08	0.01	0.01	0.07	0.03	0.01

is not used the interpolation output is dependent on the quality of the initial interpolation step. However, using the iterative step two times (i.e. use the interpolated image to re-determine the quadratic signal class and then re-apply optimal recovery twice)

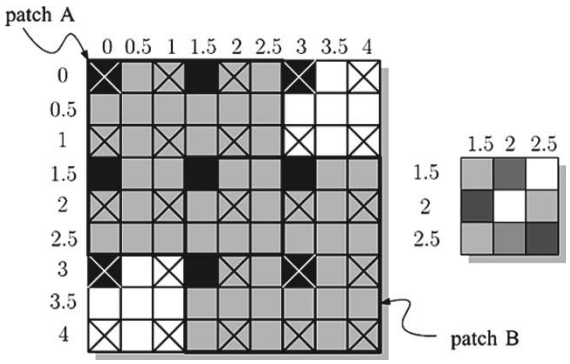


Fig. 3. Local image and two patches of 6 × 6 (dark pixels are part of the decimated image). To model the image acquisition system the assumption is that the high density image has been filtered by a low pass filter before down-sampling. For example, the pixel at location (1.5,1.5) is a weighted average of the nine high density pixels $x_{(1.5,1.5)}, \ldots, x_{(2.5,2.5)}$, shown on the right.

generates visual results that are very similar regardless of the initial interpolation step.

2) Determine the Known Representors: First, a decision must be taken about the known linear functionals of the patch that needs to be interpolated. (A linear functional of $\mathbf{x}$, denoted by $F(\mathbf{x})$, is any single valued linear function of $\mathbf{x}$, such as: samples, derivatives, integrals, etc.) In the large square of Fig. 1 assume that the center patch ($\mathbf{x}$ of (4)) belongs to a predetermined quadratic signal class K. To estimate the samples of the center patch $\mathbf{x}$ a decision must be made about what linear functionals of $\mathbf{x}$ are known. One approach is to assume that the known linear functionals of $\mathbf{x}$ (the dark pixels) are the samples of $\mathbf{x}$. In that case:

$$F_1(\mathbf{x}) = x_{(2,2)}$$
$$F_2(\mathbf{x}) = x_{(2,4)}$$
$$F_3(\mathbf{x}) = x_{(4,2)}$$
$$F_4(\mathbf{x}) = x_{(4,4)}$$

Once linear functionals are known the representors are vectors ϕ_i in $\mathbb{R}^n$ for which:

$$F_i(\mathbf{x}) = (\phi_i, \mathbf{x})_{\mathbf{Q}}.$$

The representors of the known functionals are products between $\mathbf{Q}^{-1}$ and vectors with 1 in the location of the known values and zero everywhere else (i.e. for F_1 the vector would have 1 at location (2,2)). A second alternative is to assume that the original high density pixels have been averaged first, before decimation by two. (For example, this would be the case in a digital camera where one pixel in a CCD sensor records the average light energy detected in the area of the pixel.) In this second case the known linear functionals are:

$$F_1(\mathbf{x}) = \frac{\left(x_{(2,2)} + x_{(2,3)} + x_{(3,2)} + x_{(3,3)}\right)}{4}$$
$$F_2(\mathbf{x}) = \frac{\left(x_{(2,4)} + x_{(2,5)} + x_{(3,4)} + x_{(3,5)}\right)}{4}$$
$$F_3(\mathbf{x}) = \frac{\left(x_{(4,2)} + x_{(4,3)} + x_{(5,2)} + x_{(5,3)}\right)}{4}$$
$$F_4(\mathbf{x}) = \frac{\left(x_{(4,4)} + x_{(4,5)} + x_{(5,4)} + x_{(5,5)}\right)}{4}$$

The interpolation problem is estimating $x_{(2,2)}$, $x_{(2,3)}$, $x_{(3,2)}$, and $x_{(3,3)}$ on the right of Fig. 1 from knowing F_1, F_2, F_3, and F_4 which are the high density averages at $x_{(2,2)}$, $x_{(2,4)}$, $x_{(4,2)}$, and $x_{(4,2)}$ in the big square of Fig. 1. The representors of the known functionals are now products between $\mathbf{Q}^{-1}$ and vectors that average the four samples (i.e. for F_1 the vector has 1/4 at $x_{(2,2)}, x_{(2,3)}, x_{(3,2)}, x_{(3,3)}$ and zero everywhere else). As it will be shown in the Results Section making the first assumption about the known functionals tends to produce smoother images, while using average functionals images tend to look sharper.

3) Optimal Recovery Interpolation: Once the quadratic signal class is determined from step 1 the optimal recovery solution (reviewed in the Appendix) is vector $\bar{\mathbf{u}}$ which is a linear combination of the representors found in the previous step. The estimates of the missing samples are the samples of $\bar{\mathbf{u}}$. The optimal estimates for the samples of $\mathbf{x}$ are weighted averages of the known functionals F_i and depend on both the quadratic signal class K and the representors of F_i.

To clarify the details of the interpolation steps and to exemplify the extension of AQua interpolation to any factor this paper goes through the mechanics of interpolating by a factor of 1.5. In the large square of Fig. 3 the dark pixels are the known samples from the decimated image and the pixels marked with "X" are the pixels that need to be estimated when the interpolation factor is 1.5. If the pixels of the decimated image are at indices $0, 1, 2, \ldots, m$ and the interpolation factor is $k > 1$ then in the high density image the pixels of the decimated image are at locations $0, k, 2k, \ldots, km$ and interpolation is done by estimating the missing samples at the integer locations. In our case the decimated pixels are at locations 0, 1.5, 3, 4.5, ..., etc. and the interpolation estimates the pixels at location 0, 1, 2, 3, ..., etc. The interpolation algorithm for $1.5\times$ is as follows:

1) **Determine high density class** K.
   In Fig. 3 two different training patches are patch A and B. For a general interpolation factor k the size of the training patches should be chosen such that each patch contains at least four of the known decimated pixels. This is so that the estimation of any pixel is based on at least four closest known functionals. The training patches are first estimated using cubic or other initial interpolation.
2) **Determine the known representors**.
   As previously mentioned selecting the known functionals can be based on the assumption that the decimated pixels are the same in the high density image as in the decimated image, or that the pixels in the decimated image are some weighted average of the high density pixels. For example, in the large square of Fig. 3 the value of pixel $x_{(1.5,1.5)}$ can be taken as the average of the pixels on the right side of Fig. 3. The representors are determined similarly to the $2\times$ interpolation example.
3) **Optimal recovery interpolation**.
   Again, the optimal recovery solution is vector $\bar{\mathbf{u}}$ which is a linear combination of the representors in step 2 and the estimated pixels are samples of vector $\bar{\mathbf{u}}$. The samples of $\bar{\mathbf{u}}$ can be estimated directly, without finding $\bar{\mathbf{u}}$ first. For patch A of Fig. 3 this means that the estimated samples are the pixels at location $x_{(0,0)}$, $x_{(0,1)}$, $x_{(1,0)}$, and $x_{(1,1)}$.

TABLE III
COMPARISON OF DIFFERENT 2× INTERPOLATION ALGORITHMS USING PSNR VALUES ($10\log_{10}(255^2/MSE)$). THE ORIGINAL IMAGE IS LOW-PASSED BEFORE DECIMATION. FOR AQUA INTERPOLATION THE FIRST NUMBER REPRESENTS THE ORDER OF THE POLYNOMIAL USED FOR THE INITIAL INTERPOLATION AND THE SECOND NUMBER REPRESENTS THE NUMBER OF ITERATIONS. NUMBER (3,2) MEANS A CUBIC INITIAL INTERPOLATION AND TWO ITERATIONS. THE LAST TWO COLUMNS REPRESENT THE RESULTS OF AQUA WHEN USING THE HIGH DENSITY IMAGE AS THE INITIAL ESTIMATE

Image	Cubic	Sub-Pix	Bayesian	Edge Dir	AQua					
					(3,0)	(3,2)	(0,0)	(0,2)	(HD,0)	(HD,2)
Rings	22.41	21.14	19.40	21.65	27.21	**33.26**	19.31	28.88	35.11	33.59
Barbra	30.86	30.29	29.11	27.12	30.39	29.58	**31.03**	30.20	35.27	34.72
Lena	**39.69**	36.18	35.48	38.13	39.34	39.45	37.46	39.47	39.70	39.56
Mandrill	30.21	29.28	29.18	29.55	30.16	**30.34**	29.59	30.25	30.62	30.52
Peppers	40.12	36.16	38.86	39.40	40.04	**40.24**	38.35	40.21	40.39	40.27

IV. INTERPOLATION RESULTS

AQua interpolation is compared against four other interpolation methods. These methods are: sub-pixel edge localization [8], edge directed interpolation [21], bi-cubic, and Bayesian interpolation [19]. The sub-pixel edge localization interpolation is our own implementation of the algorithm described in [8], the edge directed interpolation was obtained directly from the author, bi-cubic interpolation is Matlab's "INTERP2" function, and the Bayesian interpolation algorithm was obtained from Simon Baker [14]. Interpolation is applied to five different test images: *rings, barbra, lena, mandrill*, and *peppers*. The *rings* image is 256 × 256 and consists of concentric circles that get closer and closer to each other as they move outward, away from the origin. The *rings* image is suggested by [26] for visualizing the results of applying different interpolation filters. Images *barbra, lena, mandrill*, and *peppers* are 512 × 512 gray scale images available from [27]. All images are gray scale images with 8 bits per pixel. In all cases interpolation is performed by first filtering the original image using a 2 × 2 averaging filter and then decimating by two. The decimated images were then interpolated using the five different algorithms and these results are presented next. The interested reader can view and download full copies of these interpolated image from [28].

Peak signal-to-noise-ratio ($10\log_{10}(255^2/MSE)$) between the 2× interpolated images and the high density filtered images (just before down-sampling by two) are presented in Table III. Without iterations the AQua algorithm performs slightly worse than cubic interpolation, and when using two iterations the AQua algorithm slightly outperforms bi-cubic. Table III shows a few other interesting trends. First, for the more natural images of *lena, mandrill* and *peppers* using AQua with 2 iterations produces similar results regardless of the initial interpolation image. Down-sampling *barbra* and *rings* introduces strong aliasing and for these two images PSNR values are not as meaningful. None the less for the *rings* image AQua performs much better. This is due to the very structured nature of the *rings* image. For the *rings* image AQua is able to lock in on the structure of the local edge and better reconstruct the local image. Second, using the high density image as the initial interpolation estimate produces the best PSNR values.

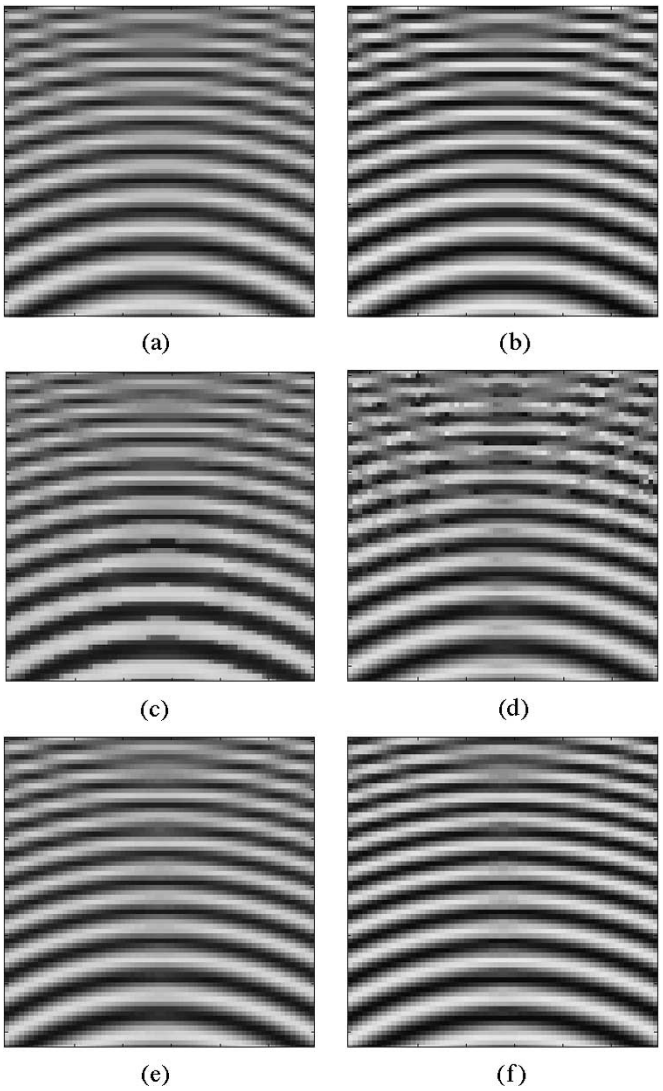


Fig. 4. Rings 2× interpolation (from left to right, top to bottom): bi-cubic (a), Bayesian (b), sub-pixel edge localization (c), edge-directed (d), AQua with samples as functionals (e), and AQua with averages as functionals (f).

It is interesting to note that in this case using two iterations deteriorates the PSNR values, as AQua tries to make the edges smoother than in the original image.

Although PSNR methods are the most common methods for measuring the quality of images, their inadequacies have long

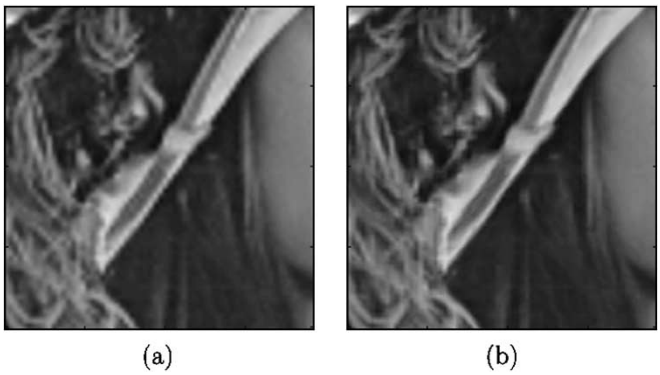


Fig. 5. Lena's hat $3\times$ interpolation (from left to right): bi-cubic (a), AQua with samples as functionals (b).

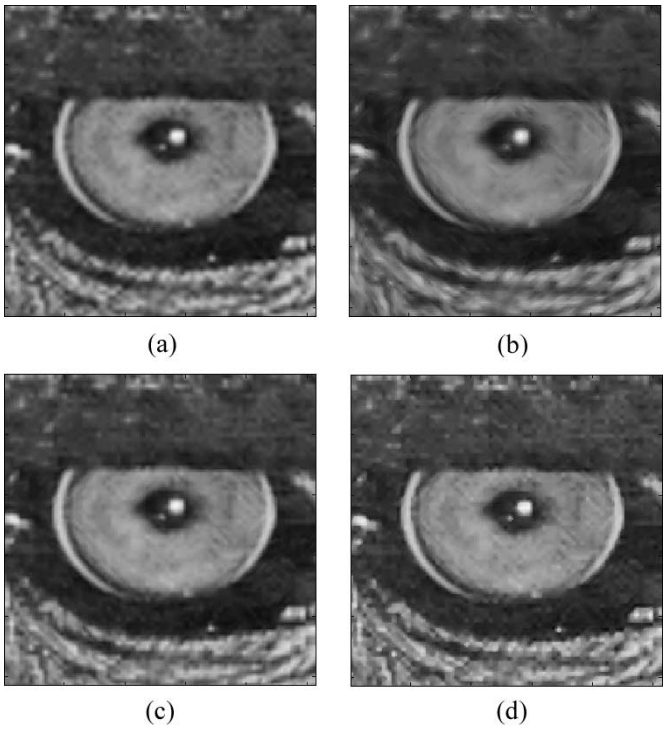


Fig. 6. Mandrill eye $4\times$ interpolation (from left to right, top to bottom): bi-cubic (a), edge-directed (b), AQua with samples as functionals (c), AQua with averages as functionals (d).

been recognized. For example, PSNR values do not take into consideration edge integrity and reconstructed images with low PSNR values can still have a very high visual quality score, as it is the case with AQua interpolation. For visual comparison and a more subjective evaluation, our results are also presented using images of interpolated results. The interpolation method uses cubic interpolation as the initial step and no iterations. From a visual point of view this method produces results that are almost indistinguishable from using any other initial interpolation and two or more iterations.

Our first image is a 256×256 *rings* image, a 1-D chirp signal rotated around 360 degrees, filtered with an averaging filter and down-sampled to 128×128. The down-sampling process introduces slight aliasing artifacts that manifest themselves as extra gray *rings*. In Fig. 4 the $2\times$ interpolation results using six different interpolation methods are presented. In this test image AQua interpolation does the best job at removing the aliasing artifacts introduced by the down-sampling process. Cubic interpolation tends to be the most blurry of all the images, while the edge directed interpolation tends to introduce slight artifacts. Also notice the extra sharpness in the AQua result that uses averages (instead of samples) for the known functionals.

The second image is *lena* (Fig. 5). To show that AQua can be applied to any interpolation factor this image has been increased $3\times$ in size using bi-cubic interpolation and AQua. (The other interpolation methods are based on interpolation by factors of $2\times$ and therefore are not compared here.) Notice how bi-cubic produces a more jagged edge in *lena*'s hat, while AQua maintains a cleaner edge.

Our final image is the $4\times$ interpolation of *mandrill*'s eye, as shown in Fig. 6. Using bi-cubic interpolation the edges around the eye tend to be somewhat jagged. The edge-directed algorithm does a much better job of maintaining edge integrity but the image tends to look a bit overly smooth. AQua interpolation with samples as functionals maintains fairly straight edges around the eye while making the image look somewhat more natural than the edge directed interpolation. Also notice the extra sharpness added to the image when functionals for AQua interpolation are averages.

In conclusion bi-cubic's main weaknesses are jagged and blurry edges. Bayesian interpolation [19] generates sharper edges but retains jagged edges. Sub-pixel edge localization interpolation [8] performs well at keeping sharp edges, but images tend to be less natural and flat. The edge directed interpolation [21] performs well at maintaining edge integrity but it performs less desirably in high frequency regions. AQua performs well compared to all the reviewed algorithms.

V. Conclusion

This paper presented a novel method for adaptive image interpolation. The algorithm first determines the local quadratic signal class from local image patches and then applies optimal recovery to estimate the missing samples. Additionally, the new interpolation algorithm allows for integrating knowledge about the lens acquisition system into the interpolation itself by using weighted averages as functionals. This tends to produce somewhat sharper images. The general theory of optimal recovery allows estimation of *any* linear functional of the image. Arbitrary interpolation factors can be used and samples on any lattice can be estimated directly. The focus here has been the interpolation of images by arbitrary factors. Through visual examples this paper has shown that AQua interpolation performs better than several other published interpolation algorithms, especially in structured images and around edges where most other algorithms introduce objectionable artifacts or jaggedness. An extension of AQua interpolation is image rotation where the samples of the rotated image can be estimated directly on the rotated grid.

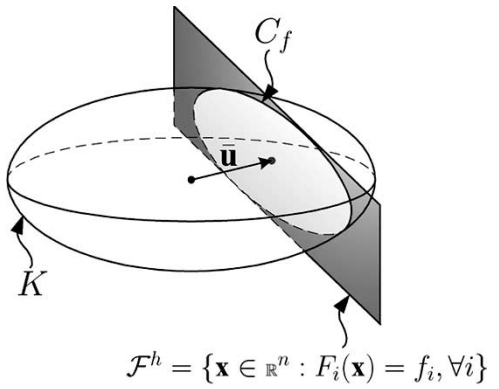


Fig. 7. Intersection of K with hyper-plane $\mathcal{F}^h$.

APPENDIX

OPTIMAL RECOVERY

The theory of optimal recovery is detailed in [1], [2]. Using the notation of [1] this appendix reviews optimal recovery as it applies to the problem of image interpolation.

The basic problem of image interpolation is that of approximating an unknown function $\mathbf{x}$ at pixel x_0 in terms of its known values at pixels $x_1, \ldots, x_k$, with the additional assumption that $\mathbf{x}$ is an element of a known linear space V. More generally, the problem is to approximate a linear functional $F(\mathbf{x})$ in terms of other known linear functionals $F_1(\mathbf{x}), \ldots, F_k(\mathbf{x})$. (A linear functional $F(\mathbf{x})$ can be any linear function of $\mathbf{x}$, such as: samples, derivatives, integrals, etc.) Further, there is the assumption that F_i are linearly independent.

The image is modeled as belonging to a certain ellipsoidal signal class K

$$K = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{x}^T \mathbf{Q} \mathbf{x} \leq \epsilon\} \tag{7}$$

where $\mathbf{Q}$ is a positive definite matrix. Noting the values of the functionals F_i by f_i, the unknown function $\mathbf{x}$ lies in the set C_f

$$C_f = \left\{\mathbf{x} \in \mathbb{R}^n : \mathbf{x}^T \mathbf{Q} \mathbf{x} \leq \epsilon, F_i(\mathbf{x}) = f_i \text{ for } i = 1, \ldots, k\right\}. \tag{8}$$

That is $\mathbf{x}$ lies in C_f, the hyper-circle defined by the intersection of the hyper-plane $\mathcal{F}^h = \{\mathbf{x} \in \mathbb{R}^n : F_i(\mathbf{x}) = f_i, \forall i\}$ with the ellipsoid signal class K, as shown in Fig. 7. With $\mathbf{Q}$-norm of $\mathbf{x}$ defined as $\|\mathbf{x}\|_{\mathbf{Q}} = \mathbf{x}^T \mathbf{Q} \mathbf{x}$, let $\bar{\mathbf{u}}$ be the minimum $\mathbf{Q}$-norm signal in C_f

$$\|\bar{\mathbf{u}}\|_{\mathbf{Q}} = \inf_{F_i(\mathbf{X}) = f_i} \|\mathbf{x}\|_{\mathbf{Q}} \tag{9}$$

and $\mathcal{F}$ be the subspaces parallel to the hyper-circle C_f:

$$\mathcal{F} = \{\mathbf{x} \in \mathbb{R}^n : F_1(\mathbf{x}) = \ldots = F_{\mathbf{k}}(\mathbf{x}) = 0\}. \tag{10}$$

Then $F(\bar{\mathbf{u}})$ is the best approximation to the value of $F(\mathbf{x})$. That is $F(\bar{\mathbf{u}})$ is the Chebyshev center [29] of $F(\mathbf{x})$ on C_f:

$$\sup_{\mathbf{X} \in C_f} |F(\bar{\mathbf{u}}) - F(\mathbf{x})| = \inf_{\mathbf{U} \in C_f} \sup_{\mathbf{X} \in C_f} |F(\mathbf{u}) - F(\mathbf{x})| \tag{11}$$

Next, let $\bar{\mathbf{y}}$ be the unit norm element in $\mathcal{F}$ for which the functional F attains its least upper bound

$$F(\bar{\mathbf{y}}) = \sup_{\mathbf{X} \in \mathcal{F}, \|\mathbf{X}\|_{\mathbf{Q}} = 1} |F(\mathbf{x})|. \tag{12}$$

Then, the bounds on the error of $F(\mathbf{x})$ are

$$F(\bar{\mathbf{u}}) - F(\bar{\mathbf{y}}) \left(\epsilon - \|\bar{\mathbf{u}}\|_{\mathbf{Q}}\right)^{\frac{1}{2}} \leq F(\mathbf{x}) \leq F(\bar{\mathbf{u}}) + F(\bar{\mathbf{y}}) \left(\epsilon - \|\bar{\mathbf{u}}\|_{\mathbf{Q}}\right)^{\frac{1}{2}} \tag{13}$$

and these bounds are attained for the functions $\mathbf{x} \in C_f$

$$\mathbf{x} = \bar{\mathbf{u}} \pm \left(\epsilon - \|\bar{\mathbf{u}}\|_{\mathbf{Q}}\right)^{\frac{1}{2}} \bar{\mathbf{y}} \tag{14}$$

which are vectors on the boundary of the hyper-circle C_f.

Calculation of $\bar{\mathbf{u}}$, $F(\bar{\mathbf{u}})$, and $\bar{\mathbf{y}}$ is done using representors. By the Riesz representation theorem [30] there are elements $\phi, \phi_1, \ldots, \phi_k$ in $\mathbb{R}^n$ such that

$$F(\mathbf{x}) = (\phi, \mathbf{x})_{\mathbf{Q}},\ F_i(\mathbf{x}) = (\phi_i, \mathbf{x})_{\mathbf{Q}},\ \forall i \tag{15}$$

for all $\mathbf{x} \in K$. Vectors $\phi, \phi_1, \ldots, \phi_k$ are linearly independent since $F, F_1, \ldots, F_k$ are assumed to be linearly independent. Functionals $F_i(\mathbf{x}) = (\phi_i, \mathbf{x})_{\mathbf{Q}}$ remain constant for all $\mathbf{x} \in C_f$. That means subspace $\mathcal{F}$ is the set of all vectors in $\mathbb{R}^n$ orthogonal to the representors $\phi_i, \forall i$. Equivalently, $\phi_1, \ldots, \phi_k$ is a basis for $\mathcal{F}^{\perp}$. With $\bar{\mathbf{u}} \in \mathcal{F}^{\perp}$ it follows that $\bar{\mathbf{u}}$ is a linear combination of the representors ϕ_i. Similarly, $\bar{\mathbf{y}}$ is a linear combination of the representors $\phi, \phi_1, \ldots, \phi_k$

$$\bar{\mathbf{u}} = \sum_i c_i \phi_i \text{ and } \bar{\mathbf{y}} = d\phi + \sum_i d_i \phi_i. \tag{16}$$

Constants c_i are found by forcing $\bar{\mathbf{u}}$ to satisfy the given functionals

$$F_i(\bar{\mathbf{u}}) = (\phi_i, \bar{\mathbf{u}})_{\mathbf{Q}} \tag{17}$$

$$= \left(\phi_i, \sum_j c_j \phi_j\right)_{\mathbf{Q}}. \tag{18}$$

In matrix form, this is equivalent to solving

$$\begin{bmatrix} F_1(\bar{\mathbf{u}}) \\ \vdots \\ F_k(\bar{\mathbf{u}}) \end{bmatrix} = \begin{bmatrix} (\phi_1, \phi_1)_{\mathbf{Q}} & (\phi_1, \phi_2)_{\mathbf{Q}} & \cdots & (\phi_1, \phi_k)_{\mathbf{Q}} \\ (\phi_1, \phi_2)_{\mathbf{Q}} & (\phi_2, \phi_2)_{\mathbf{Q}} & \cdots & (\phi_2, \phi_k)_{\mathbf{Q}} \\ \vdots & \vdots & \vdots & \vdots \\ (\phi_1, \phi_k)_{\mathbf{Q}} & \vdots & \vdots & (\phi_k, \phi_k)_{\mathbf{Q}} \end{bmatrix} \begin{bmatrix} c_1 \\ \vdots \\ c_k \end{bmatrix} \tag{19}$$

for c_i. Constants d and d_i are found similarly. Using the fact that $\bar{\mathbf{y}} \in \mathcal{F}$ it follows that $\bar{\mathbf{y}}$ is perpendicular to the representors ϕ_i, $\forall i$:

$$d \begin{bmatrix} (\phi, \phi_1)_{\mathbf{Q}} \\ \vdots \\ (\phi, \phi_k)_{\mathbf{Q}} \end{bmatrix} + \begin{bmatrix} (\phi_1, \phi_1)_{\mathbf{Q}} & (\phi_1, \phi_2)_{\mathbf{Q}} & \cdots & (\phi_1, \phi_k)_{\mathbf{Q}} \\ (\phi_1, \phi_2)_{\mathbf{Q}} & (\phi_2, \phi_2)_{\mathbf{Q}} & \cdots & (\phi_2, \phi_k)_{\mathbf{Q}} \\ \vdots & \vdots & \vdots & \vdots \\ (\phi_1, \phi_k)_{\mathbf{Q}} & \vdots & \vdots & (\phi_k, \phi_k)_{\mathbf{Q}} \end{bmatrix} \begin{bmatrix} d_1 \\ \vdots \\ d_k \end{bmatrix} = 0 \tag{20}$$

From (20) find d_i as a function of d. Vector $\bar{\mathbf{y}}$ from (16) is now defined as a function of only one unknown constant d. Constant d is found from the restriction that $\|\bar{\mathbf{y}}\|_{\mathbf{Q}} = 1$.

ACKNOWLEDGMENT

Many thanks go to X. Li (Princeton University) and Simon Baker (Carnegie Mellon University) for providing working code for two of the interpolation algorithms. Thanks also go to R. Song, member of the DSP Lab, for implementing the algorithm of [8]. The authors are also grateful to Prof. S. Hemami (Cornell University) and the journal reviewers for their excellent comments.

REFERENCES

[1] M. Golomb and H. F. Weinberger, *On Numerical Approximation*, R. E. Langer, Ed: The University of Wisconsin Press, 1959, ch. Optimal Approximation and Error Bounds.
[2] C. A. Michelli and T. J. Rivlin, *Optimal Estimation in Approximation Theory*. New York: Plenum, 1976, ch. A Survey of Optimal Recovery.
[3] E. Maeland, "On the comparison of interpolation methods," *IEEE Trans. Medical Imaging*, vol. 7, pp. 213–217, 1988.
[4] T. M. Lehmann, C. Gönner, and K. Spitzer, "Survey: Interpolation methods in medical image processing," *IEEE Trans. Medical Imaging*, vol. 18, pp. 1049–1075, 1999.
[5] G. Ramponi, "Warped distance for space-variant linear image interpolation," *IEEE Trans. Image Processing*, vol. 8, pp. 629–639, 1999.
[6] M. Unser, A. Aldroubi, and M. Eden, "Enlargement or reduction of digital images with minimum loss of information," *IEEE Trans. Image Processing*, vol. 4, pp. 247–258, 1995.
[7] L. Chulhee, M. Eden, and M. Unser, "High-quality image resizing using oblique projection operators," *IEEE Trans. Image Processing*, vol. 7, pp. 679–692, 1998.
[8] K. Jensen and D. Anastassio, "Subpixel edge localization and the interpolation of still images," *IEEE Trans. Image Processing*, vol. 4, pp. 285–295, 1995.
[9] J. Allebach and P. W. Wong, "Edge-directed interpolation," in *IEEE Proc. ICIP*, 1996, pp. 707–710.
[10] S. Mallat and S. Zhong, "Characterization of signals from multiscale edges," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 14, pp. 710–732, 1992.
[11] S. G. Chang, Z. Cvetkovic, and M. Vetterli, "Resolution enhancement of images using wavelet transform extrema interpolation," in *IEEE Proc. ICASSP*, 1995, pp. 2379–2382.
[12] W. K. Carey, D. B. Chuang, and S. S. Hemami, "Regularity-preserving image interpolation," *IEEE Trans. Image Processing*, vol. 8, pp. 1293–1297, 1999.
[13] W. T. Freeman, T. R. Jones, and E. C. Pasztor, "Example-Based Super-Resolution," MERL—Mitsubishi Electronic Research Laboratory, 2001.
[14] S. Baker and T. Kanade, "Limits on super-resolution and how to break them," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, no. 9, pp. 1167–1183, 2002.
[15] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin, "Image analogies," in *Proc. SIGGRAPH 2001*, 2001.
[16] J. K. Romberg, H. Choi, and R. G. Baraniuk, "Bayesian tree-structured image modeling using wavelet-domain hidden Markov models," *IEEE Trans. on Image Processing*, vol. 10, no. 7, pp. 1056–1068, 2001.
[17] ——, "Shift-invariant denoising using wavelet-domain hidden Markov trees," in *Conference Record of the Thirty-Third Asilomar Conference on Signals, Systems, and Computers*, vol. 2, 1999, pp. 1277–1281.
[18] K. Kinebuchi, D. D. Muresan, and T. W. Parks, "Image interpolation using wavelet-based hidden Markov trees," in *IEEE Proc. ICASSP*, 2001.
[19] R. Schultz and R. Stevenson, "A bayesian approach to image expansion for improved definition," *IEEE Trans. on Image Processing*, vol. 3, no. 3, pp. 233–242, 1994.
[20] R. C. Hardie, K. J. Barnard, and E. E. Armstrong, "Joint map registration and high-resolution image estimation using a sequence of undersampled images," *IEEE Trans. Image Processing*, vol. 6, no. 12, pp. 1621–1633, 1997.
[21] X. Li, "Edge Directed Statistical Inference With Applications to Image Processing," Ph.D. dissertation, Princeton University, Princeton, NJ, 2000.
[22] D. D. Muresan and T. W. Parks, "Adaptive, optimal-recovery image interpolation," in *IEEE Proc. ICASSP*, 2001.
[23] G. Chen and R. J. P. de Figueiredo, "A unified approach to optimal image interpolation problems based on linear partial differential equation models," *IEEE Transactions on Image Processing*, vol. 2, no. 1, 1993.
[24] D. D. Muresan and T. W. Parks, "Optimal face reconstruction using training," in *IEEE Proc. ICIP*, 2002.
[25] L. N. Trefethen and D. Bau III, *Numerical Linear Algebra*: SIAM, 1997.
[26] J. Bernd, *Digital Image Processing*: Springer-Verlag, 1995.
[27] Image Database (2002). *http://links.uwaterloo.ca/bragzone.base.html* [Online]
[28] The DSP Website (2003). *http://dsplab.ece.cornell.edu/papers/results/orinterp/* [Online]
[29] R. G. Shenoy and T. W. Parks, "An optimal recovery approach to interpolation," *IEEE Trans. Signal Processing*, vol. 40, no. 8, pp. 1987–1996, 1992.
[30] W. Rudin, *Principles of Mathematical Analysis*: McGraw-Hill, Inc., 1976.

D. Darian Muresan (M'03) received the B.S. degree in mathematics and electrical engineering from the University of Washington, Seattle, and the M.Eng. and Ph.D. degrees in electrical and computer engineering from Cornell University, Ithaca, NY.

His interests include image and signal processing and hardware design. He is the co-inventor of an analog-to-digital converter and has several other patents pending in image processing. He interned with Hewlett-Packard as a Hardware Design Engineer and is the founder of Digital Multi-Media Design, Arlington, VA (http://www.dmmd.net).

Thomas W. Parks (F'82) received the B.E.E., M.S., and Ph.D. degrees from Cornell University, Ithaca, NY.

From 1967 to 1986, he was on the Electrical Engineering faculty at Rice University, Houston, TX. In 1986, he joined Cornell as a Professor of electrical engineering. He has co-authored a number of books on digital signal processing. His research interests are signal theory and digital signal processing.

Dr. Parks is a recipient of the IEEE Third Millennium Medal. He received the Humboldt Foundation Senior Scientist Award, and has been a Senior Fulbright Fellow.