High-Quality Image Resizing Using Oblique Projection Operators

Chulhee Lee, *Member, IEEE,* Murray Eden, *Life Fellow, IEEE,* and Michael Unser, *Senior Member, IEEE*

***Abstract*— The standard interpolation approach to image resizing is to fit the original picture with a continuous model and resample the function at the desired rate. However, one can obtain more accurate results if one applies a filter prior to sampling, a fact well known from sampling theory. The optimal solution corresponds to an orthogonal projection onto the underlying continuous signal space. Unfortunately, the optimal projection prefilter is difficult to implement when sinc or high order spline functions are used. In this paper, we propose to resize the image using an oblique rather than an orthogonal projection operator in order to make use of faster, simpler, and more general algorithms. We show that we can achieve almost the same result as with the orthogonal projection provided that we use the same approximation space. The main advantage is that it becomes perfectly feasible to use higher order models (e.g, splines of degree $n \geq 3$). We develop the theoretical background and present a simple and practical implementation procedure using B-splines. Our experiments show that the proposed algorithm consistently outperforms the standard interpolation methods and that it provides essentially the same performance as the optimal procedure (least squares solution) with considerably fewer computations. The method works for arbitrary scaling factors and is applicable to both image enlargement and reduction.**

***Index Terms*— B-spline models, image scaling (magnification and reduction), oblique projection, orthogonal projection, resampling.**

I. Introduction

IMAGE scaling (magnification or reduction) is a basic operation in digital image processing [1], [2]. It is required on a routine basis for medical imaging, multimedia, and digital photography. The standard interpolation procedure is to fit the data with a continuous image model and then resample this function on the grid appropriate to the scaling desired. What differentiates among the available methods is the interpolation model chosen [3]–[5]. In fact, it is well known in approximation theory that the performance of an algorithm depends primarily on the model's ability to reproduce polynomials up to a specified degree n [6], [7]. Lower order methods are simplest and most rapid to implement but they produce artifacts. Nearest neighbor $(n = 0)$ interpolation is extremely cheap computationally, but typically introduces very noticeable blocklike artifacts. Bilinear interpolation $(n = 1)$, which calculates each new pixel value from its the four closest neighbors, is somewhat better but tends to blur small image details. Higher order interpolation methods—splines, in particular—produce much better outcomes but require much more computation. In the extreme, one approaches the ideal sinc-interpolator which can also be interpreted as a spline of infinite degree $(n \rightarrow +\infty)$ [8]. This latter option is almost never used in practice because of the very slow decay of $\text{sinc}(x) = O(1/|x|)$. It also gives rise to Gibbs oscillations. For the special case in which the scaling factor is a power of two, the resizing (decimation or interpolation) can be implemented by digital filtering using the tools developed in the context of the wavelet transform [9]–[11]. There have also been some recent proposals of nonlinear extrapolation techniques but these are typically restricted to image magnification by a factor of two [12].

Even though higher order interpolation methods (typically $n \geq 3$) work very well for image magnification and rotation, their use is more questionable for image reduction because of potential aliasing problems. To deal with those limitations, Unser et al. recently proposed a resizing procedure that uses the same type of spline signal representation as before, but applies a continuous prefilter prior to sampling in order to minimize artifacts [13]. This algorithm performs the orthogonal projection of the scaled image onto the sampling space and provides an approximation that is optimal in the least squares sense. Note that the method is equally applicable for image reduction or magnification by an arbitrary, not necessarily integer scaling factor. The only practical limitation is that it is difficult to perform an exact numerical implementation of the optimal prefilter for higher order splines $(n \geq 3)$.

In this paper, we propose to extend this previous approach by considering more general approximation schemes using oblique rather than orthogonal projectors. Interestingly, these projectors are of the same type as those encountered in wavelet theory for the decomposition of functions in terms of biorthogonal wavelet bases [14]–[16], except that we consider arbitrary scaling factors instead of limiting the scaling to powers of two. Our primary motivation here is to retain as much freedom as possible in the design of the analog prefilter that is to be applied to the rescaled signal. Computationally, this prefiltering step is the most difficult part of the least squares resizing algorithm. By choosing very simple filters (for example, the box function), we can hope in this way to obtain faster, simpler, and possibly more general algorithms.

Manuscript received January 25, 1996; revised June 12, 1997. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Ping Wah Wong.

C. Lee is with the Department of Electronic Engineering, Yonsei University, Seoul, Korea (e-mail: chulhee@bubble.yonsei.ac.kr).

M. Eden is with the National Institutes of Health, BEIP/NCRR, Bethesda, MD 20892, USA (e-mail: edenm@helix.nih.gov).

M. Unser is with Biomedical Image Processing, EPFL, DMT/IOA, Swiss Federal Institute of Technology, CH-1015 Lausanne, Switzerland (e-mail: Michael.Unser@epfl.ch).

Publisher Item Identifier S 1057-7149(98)03077-2.

In addition, we have very strong theoretical reasons to expect that this can be achieved with no sacrifice in quality, provided that we use the same approximation space as before [17], [18]. One goal of this paper will be to demonstrate that this is indeed the case. Moreover, we will see how we can use oblique projections to compute much higher order spline approximations than is practicable in the orthogonal case.

In essence, the method that we propose is a resampling approach rather than an interpolation in the conventional sense. As such, it is suited primarily for image reduction, especially for the case of noninteger factors that cannot be dealt with by using conventional decimation techniques. Since the algorithm is fast and works for arbitrary scaling factors, it is also applicable to image enlargements; in fact, our new method is equivalent to a spline interpolation when the magnification factor is an integer, and leads to better performance otherwise.

The paper is organized as follows. First, we provide definitions of some special notations and operators. In Section II, we present the basic principle of the algorithm and introduce the relevant signal subspaces $V(\varphi)$. We then discuss and compare the orthogonal (least squares) and oblique projection operators used to approximate the rescaled signal in $V(\varphi)$. In Section III, we derive a polynomial spline version of the oblique projection algorithm and propose a simple, practical implementation. In Section IV, we describe some experiments and discuss the results. In particular, we compare the proposed algorithm with the conventional methods.

A. Notation and Operators

L_2 is the vector space of measurable, square-integrable functions $g(x), x \in R$. L_2 is a Hilbert space whose metric $\|\cdot\|$ (the L_2-norm) is derived from the inner product

$$\langle g, h\rangle = \int_{-\infty}^{+\infty} g(x)h(x)\,dx$$
$$\|g\| = \sqrt{\langle g, g\rangle} = \left(\int_{-\infty}^{+\infty} |g(x)|^2\,dx\right)^{1/2}. \tag{1}$$

$\beta^n(x)$ represents the centered B-spline of degree n, which is obtained from the $(n+1)$-fold convolution of a unit rectangular pulse. An equivalent recursive definition is

$$\beta^n(x) = \beta^0(x)^*\beta^{n-1}(x) \tag{2}$$

where

$$\beta^0(x) = \begin{cases} 1, & -\frac{1}{2} \le x < +\frac{1}{2} \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

The B-spline is a symmetrical compactly supported, piecewise polynomial of degree n. We use the corresponding roman symbol to denote the discrete B-spline kernel, which is obtained by sampling the B-spline at the integers:

$$b^n(k) := \beta^n(x)|_{x=k}. \tag{4}$$

The sequence $(b^n)^{-1}(k)$ represents the impulse response of the corresponding inverse filter, which is stable for any n. It is the so-called direct B-spline filter of degree n [19].

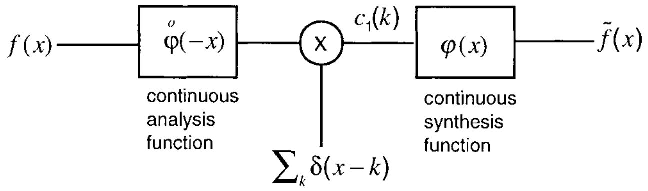


Fig. 1. Orthogonal projection. The signal is first convolved with the optimal prefilter and sampled thereafter. The sampling is modeled by a multiplication with a sequence of Dirac impulses. The synthesis filter corresponds to the generating function φ.

II. Projection-Based Image Resizing

A. Basic Principle of the Algorithm

As mentioned in the introduction, the standard approach to image resizing uses interpolation. Since we consider scaling only along the coordinate axes (without rotation), the processing can be performed in a completely separable fashion provided that we use tensor product basis functions. In fact, the same is also true if we use more sophisticated projection techniques; a justification can be found in [13]. In other words, we can re-size an image (or volume) by successive one-dimensional (1-D) processing along the several dimensions of the data. Thus, the mathematical problem reduces to that of resizing a 1-D signal $s(k)$, $k = 0, \ldots N-1$. For our purposes, it is advantageous to think of this process in terms of the following paradigm.

1) The discrete 1-D data set $s(k)$ is fitted with a continuously defined function $g(x)$ that provides an exact interpolation of the data points; that is, such that $g(x)|_{x=k} = s(k)$.
2) The scaling transformation, which is a mapping from L_2 into itself, is applied to the function $g(x)$. This yields the continuously-defined rescaled function $f(x) = g(x/a)$ where a is the scaling factor.
3) The function $f(x)$ is resampled at the integers (standard approach), or alternatively, is represented by an appropriate approximation in a given sampling space (new approach).

We note that this particular interpretation is the reverse of the classical one in which the interpolated function $g(x)$ is kept fixed and it is the sampling grid that is transformed accordingly. In any case, this is just a matter of preference because both formulations are mathematically equivalent. Here, we will adopt this new paradigm because it lends itself to a more concise mathematical treatment (compare it with the derivations in [13] which were done using the traditional framework). The present formulation also has certain advantages if one wants to extend the approach for more general classes of transformations.

In order to be applicable, we need make the above paradigm more precise by specifying the interpolation model used. Without loss of generality, we can assume that the interpolation function lies in a certain Hilbert space, $V(\varphi)$, generated by our interpolation function φ (a more precise definition will be given in the next section). The important point is that the chosen Hilbert space $V(\varphi)$ assures that any function

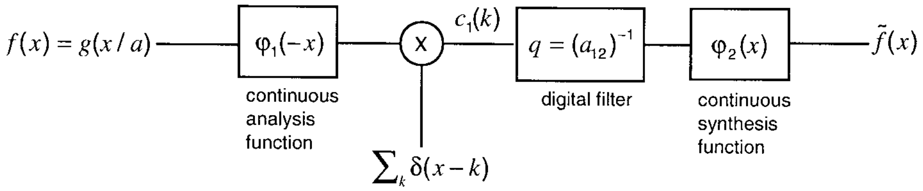


Fig. 2. Oblique projection. The block diagram is similar to Fig. 1 except that it includes an additional digital correction filter q. The prefilter, on the other hand, can be arbitrary.

$s(x) \in V(\varphi)$ is uniquely determined by its sample values $s(k)$ (its discrete representation). In other words, there is a unique function $g(x) \in V(\varphi)$ that provides an exact interpolation of the discrete input signal $s(k)$. The re-sizing method that we propose here differs from the standard interpolation approach only by the way in which we implement step 3. Instead of a straightforward resampling, we will approximate $f(x)$ by its projection $\tilde{f}(x)$ into the Hilbert space $V(\varphi)$. The resized digital signal will then correspond to the samples of $\tilde{f}(x)$; this projected function may be thought off as the "alias-free" version of $f(x)$.

B. Continuous/Discrete Signal Representation

We choose to represent our signals in the space $V(\varphi) \subset L_2$ which is defined as

$$V(\varphi) = \left\{ s(x) = \sum_{k \in Z} c(k)\varphi(x-k) \mid c \in l_2 \right\} \tag{5}$$

where l_2 is the vector space of square-summable sequences. The only restriction on the choice of the *generating* function $\varphi(x)$ is that the set $\{\varphi(x-k)\}_{k \in Z}$ is a Riesz basis of $V(\varphi)$; this is equivalent to the condition

$$0 < A \leq \hat{a}_\varphi(\omega) := \sum_{k \in Z} |\hat{\varphi}(\omega + 2\pi k)|^2 \leq B < +\infty \text{ a.e.} \tag{6}$$

where $\hat{\varphi}(\omega)$ is the Fourier transform of $\varphi(x)$, and where the constants A and B are the so-called frame (or Riesz) bounds [20]. This constraint ensures that the integer shifts of φ are linearly independent and that each function $s(x)$ in $V(\varphi)$ is uniquely characterized by the sequence of its coefficients $c(k)$. Note that $\hat{a}_\varphi(\omega)$ in (6) is also the Fourier transform of the autocorrelation sequence $a_\varphi(k) = \langle \varphi(x), \varphi(x+k) \rangle$.

The class of function spaces that can be specified in this way is quite general. It covers all interpolation models mentioned in the introduction, as well as the various multiresolution spaces associated with the wavelet transform. For the particular choice, $\varphi(x) = \beta^n(x)$ (B-spline of degree n), $V(\varphi)$ represents the space of polynomial splines of degree n with equally spaced knots at the integers (n odd) or between the integers for n even [21], [22].

So far, we have imposed no constraint on φ other then the stability condition (6). If we want the coefficients $c(k)$ in (5) to coincide with the samples of $s(x)$, it is necessary that the generating function satisfy the additional interpolation condition

$$\varphi_{\text{int}}(k) = \delta[k]$$

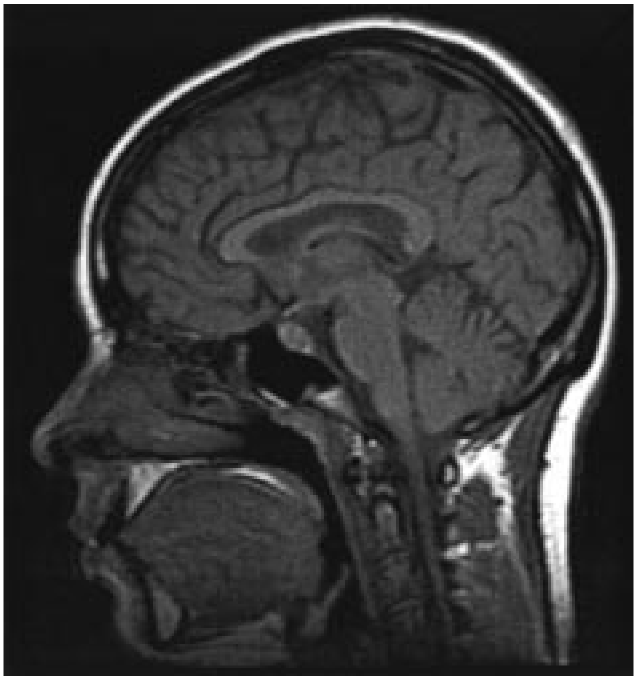


Fig. 3. Original magnetic resonance image of brain.

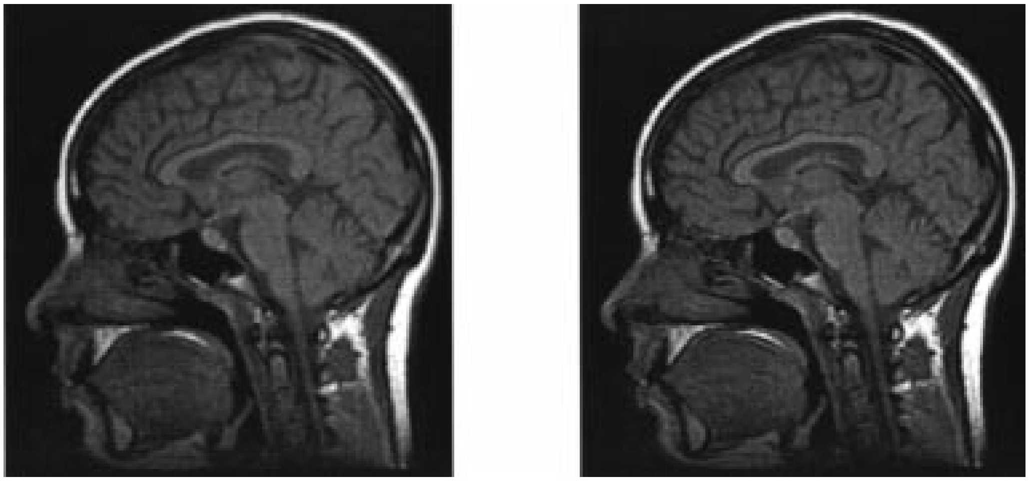
TABLE I
ANGLE BETWEEN THE SPLINE SPACES $V(\beta^n)$ AND $V(\beta^0)$ FOR $n = 0, \cdots, 9$

n	$\cos\theta_{12}$ (worst case)	$\overline{\cos\theta_{12}}$ (average spectral coherence)
0	1	1
1	0.866025	0.926420
2	0.872872	0.930323
3	0.836154	0.916853
4	0.816242	0.911748
5	0.797844	0.906829
6	0.783740	0.903457
7	0.771842	0.900722
8	0.761928	0.898563
9	0.753454	0.896791

where $\delta[k]$ denotes the discrete unit impulse at the origin. For a given representation space $V(\varphi)$, there is generally a unique interpolation kernel $\varphi_{\text{int}} \in V(\varphi)$, given by

$$\varphi_{\text{int}}(x) = \sum_{k \in Z} (b)^{-1}(k)\varphi(x-k)$$

where $(b)^{-1}$ denotes the convolution inverse of the sequence $b(k) := \varphi(x)|_{x=k}$, the sampled version of the generating function. For the particular case of polynomial splines, $\varphi^n_{\text{int}}(x)$ is the so-called cardinal (or fundamental) spline which can be



(a) (b)

Fig. 4. (a) Standard interpolation (B-spline of degree one), reduced by $\sqrt{2}$. (b) Oblique projection (B-spline of degree one), reduced by $\sqrt{2}$.

written as

$$\varphi_{\text{int}}^n(x) = \sum_{k\in Z} (b^n)^{-1}(k)\beta^n(x-k) \tag{7}$$

where $\beta^n(x)$ and b^n are defined by (3) and (4), respectively; the transfer functions of the spline filters $(b^n)^{-1}$ for $n = 0$ to 11 can be found in Table II. Although the cardinal spline function has infinite support, it provides a very convenient representation of a spline in terms of its samples at the integers. If we compare the cardinal spline representation of signal $s(x) = \sum_{k\in Z} s(k)\varphi_{\text{int}}^n(x-k)$ (spline interpolant) with its equivalent B-spline expansion $s(x) = \sum_{k\in Z} c(k)\beta^n(n-k)$, we find that

$$c(k) = ((b^n)^{-1} * s)(k). \tag{8}$$

This demonstrates that the conversion between these representations can be done by digital filtering. This equation is the basis for the fast spline interpolation algorithm described in [19], which computes the B-spline coefficients recursively using a cascade of causal and anti-causal exponential filters.

C. Least Square Approximation

Given an arbitrary function $f \in L_2$ (in our case, $f(x) = g(x/a)$) where g is a function that interpolates the input signal and a the desired scaling factor), the orthogonal projection of f onto $V(\varphi)$ is given by (cf. [20])

$$(P_\varphi f)(x) = \sum_{k\in Z} \langle f(x), \overset{o}{\varphi}(x-k)\rangle \varphi(x-k)$$

where $\overset{o}{\varphi} \in V(\varphi)$ is the dual of φ and is defined as

$$\overset{o}{\varphi} = \sum_{k\in Z} (a_\varphi)^{-1}(k)\varphi(x-k).$$

$(a_\varphi)^{-1}$ represents the convolution inverse of the autocorrelation sequence $a_\varphi := \langle \varphi(x), \varphi(x-k)\rangle$. Note that $P_\varphi f$ provides the minimum L_2-error approximation of f in $V(\varphi)$ (least squares solution). This approximation process is illustrated in Fig. 1.

When we combine this approximation process with steps 1 and 2 in our general paradigm, we end up with a generalized version of the least squares resizing procedure described in [13], which was derived in the more restricted spline framework. In practice, this approach works extremely well but is difficult to implement exactly for splines of degree greater than one. The complication arises because the dual analysis function $\overset{o}{\varphi}$ (optimal prefilter) is determined uniquely by the choice of the synthesis function φ. Therefore, depending on the approximation space used, it may be extremely difficult to perform the appropriate prefiltering which needs to be applied in a continuous fashion. In order to gain in design flexibility, we can consider the more general approximation scheme described next.

D. Oblique Projection

Fig. 2 illustrates the generalized sampling procedure initially proposed in [23]. In this approach, there is no restriction on the prefilter φ_1: it can be chosen independent of the generating function $\varphi_2 = \varphi$. To compensate for a possible mismatch, the system includes a digital correction filter ensuring that the input signal $f(x)$ and its approximation $\tilde{f}(x)$ are consistent in the sense that they yield the same measurement:

$$c_1(k) := \langle f(x), \varphi_1(x-k)\rangle = \langle \tilde{f}(x), \varphi_1(x-k)\rangle.$$

Under these conditions, the approximation $\tilde{f}(x)$ corresponds to the projection of $f(x)$ onto $V(\varphi_2)$ perpendicular to the analysis space $V(\varphi_1)$. It is given by

$$\tilde{f}(x) = P_{2\perp 1} f(x) = \sum_{k\in Z} (q^* c_1)(k)\varphi_2(x-k) \tag{9}$$

where $q = (a_{12})^{-1}$ is the convolution inverse of the cross-correlation sequence $a_{12}(k) := \langle \varphi_1(x-k), \varphi_2(x)\rangle$. This projector has the special property that the approximation error is perpendicular to $V(\varphi_1)$. Therefore, unless $\varphi_1 \in V(\varphi_2)$ (or equivalently $V(\varphi_1) = V(\varphi_2)$), the projection error is not orthogonal to the approximation space $V(\varphi_2)$; hence the term "oblique."

TABLE II
DIRECT B-SPLINE FILTERS FOR $n = 0$ UP TO 11

Impulse response	Transfer function
$(b^1)^{-1}(k)$	1
$(b^2)^{-1}(k)$	$\frac{8}{z+6+z^{-1}}$
$(b^3)^{-1}(k)$	$\frac{6}{z+4+z^{-1}}$
$(b^4)^{-1}(k)$	$\frac{384}{z^2+76z+230+76z^{-1}+z^{-2}}$
$(b^5)^{-1}(k)$	$\frac{120}{z^2+26z+66+26z^{-1}+z^{-2}}$
$(b^6)^{-1}(k)$	$\frac{46080}{z^3+722z^2+10543z+23548+10543z^{-1}+722z^{-2}+z^{-3}}$
$(b^7)^{-1}(k)$	$\frac{5040}{z^3+120z^2+1191z+2416+1191z^{-1}+120z^{-2}+z^{-3}}$
$(b^8)^{-1}(k)$	$\frac{10321920}{\begin{matrix}z^4+6552z^3+331612z^2+2485288z+4675014\\+2485288z^{-1}+331612z^{-2}+6552z^{-3}+z^{-4}\end{matrix}}$
$(b^9)^{-1}(k)$	$\frac{362880}{z^4+502z^3+14608z^2+88234z+156190+8823z^{-1}+14608z^{-2}+502z^{-3}+z^{-4}}$
$(b^{10})^{-1}(k)$	$\frac{3715891200}{\begin{matrix}z^5+59038z^4+9116141z^3+178300904z^2+906923282z+1527092468\\+906923282z^{-1}+178300904z^{-2}+9116141z^{-3}+59038z^{-4}+z^{-5}\end{matrix}}$
$(b^{11})^{-1}(k)$	$\frac{39916800}{\begin{matrix}z^5+2036z^4+152637z^3+2203488z^2+9738114z+15724248\\+9738114z^{-1}+2203488z^{-2}+1526371z^{-3}+2036z^{-4}+z^{-5}\end{matrix}}$

When compared to the least squares solution from the previous section with $\varphi = \varphi_2$, the oblique projection will in general not be optimal. However, we can expect its loss of performance to be negligible under most circumstances. In fact, there are two mathematical results that partially support this prediction.

First, we have the following error bound (cf. [23], Theorem 3)

$$\forall f \in L_2, \|f - P_2 f\| \leq \|f - P_{2\perp 1} f\| \leq \frac{1}{\cos\theta_{12}} \|f - P_2 f\| \tag{10}$$

where $P_{2\perp 1}f$ is defined by (9) and $P_2 f$ denotes the orthogonal projection of f onto $V(\varphi_2)$; θ_{12} is the (largest) angle between the subspace $V(\varphi_1)$ and $V(\varphi_2)$. In other words, the approximation errors must follow each other rather closely suggesting that both methods are nearly equivalent. Of special interest for our purpose are the angles between the space of piecewise constant splines ($\varphi_1 = \beta^0$) and the splines of degree n ($\varphi_2 = \beta^n$). These quantities are given in Table I. The first value corresponds to the worst case (maximum angle) while the second is an average measure integrated over all frequencies. The mathematical details of this analysis can be found in [23] (Sections IV and V-D). In any case, we should note that the upper bound in (10) corresponds to a worst case scenario and that the agreement between the two errors is in general much tighter than the factor $1/\cos\theta_{12}$.

Second, we have shown recently that both approximation methods have the same asymptotic behavior as the sampling step goes to zero (cf. [18], Theorem 4.1). The only requirement for this asymptotic equivalence is that the analysis function satisfies the following "partition of unity" condition

$$\sum \varphi_1(x-k) = 1.$$

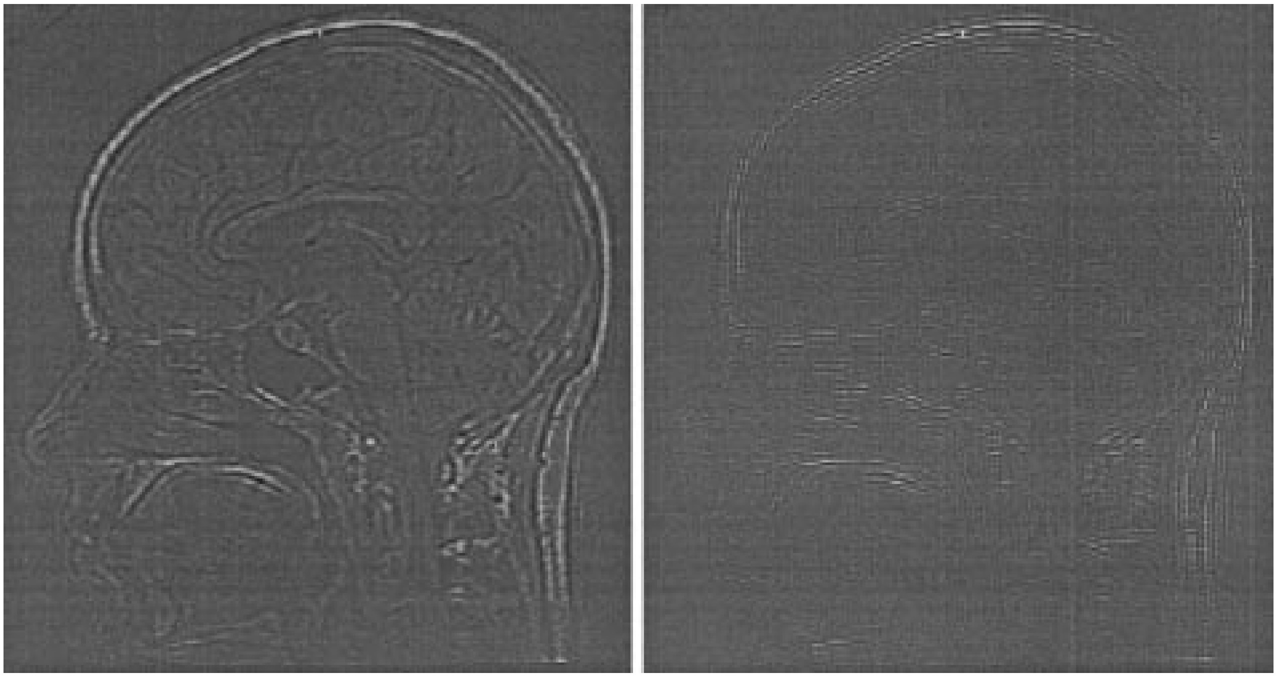
In practice, any approximation procedure will reach its asymptotic regime as soon as the function to be approximated becomes sufficiently smooth with respect to the sampling grid. In the present context, this means that the eventual difference between oblique and orthogonal projection methods should vanish as the scaling factor increases. More precisely, if we define $f_a(x) = g(x/a)$, we have that

$$\|f_a(x) - P_2 f_a\| = C \cdot \left\|f_1^{(L)}\right\| \cdot a^{-L} + O(a^{-(L+1)})$$

and

$$\|f_a(x) - P_{2\perp 1} f_a\| = C \cdot \left\|f_1^{(L)}\right\| \cdot a^{-L} + O(a^{-(L+1)}) \tag{11}$$

with the same constant C in each case. The integer $L = n+1$ represents the order of the representation (i.e., the model has the ability to reproduce polynomials of degree n), and $\|f_1^{(L)}\|$ is the norm of the L-th derivative of $f_1 = g$. Polynomial splines of degree n have an order of approximation $L = n+1$; the corresponding value of the constant C has also been determined explicitly [18].



(a) (b)

Fig. 5. (a) Approximation error for Fig. 4(a), $\mathrm{SNR} = 33.65$ dB. (b) Approximation error for Fig. 4(b), $\mathrm{SNR} = 39.02$ dB.

The only potential problem with an oblique projection is that it may increase noise. In our case, this effect will be minimal because the norm of the projector will remain close to one. In fact, there is a simple relation between the angle and the norm of this projector: $\|P_{2\perp 1}\| = 1/\cos\theta_{12}$ (cf. Theorem 4, in [24]). Thus, the quantity $1/\cos\theta_{12}$ also represents the worst case amplification factor for the noise.

Thus, the idea that we will explore next is to modify the resizing algorithm proposed in [13] by using such an oblique projection to simplify the approximation procedure in step 3.

III. Implementation Using Splines

A. Derivation of the Algorithm

In our implementation, we select the synthesis function $\varphi_2 = \varphi_{\text{int}}^n$ to be the cardinal spline of any degree n (cf. (7)). As for the analysis function, we try to choose the simplest φ_1 with the narrowest support such that $\sum_{k\in Z} \varphi_1(x-k) = 1$. The first choice that comes to mind is the B-spline of degree zero: $\varphi_1 = \beta^0$. A clear advantage of using β^0 is that the required inner products can be computed as simple integrals:

$$\begin{aligned} c_1(k) &= \langle g(x/a), \beta^0(x-k)\rangle \\ &= \int g(x/a)\beta^0(x-k)\,dx \\ &= \int_{k-1/2}^{k+1/2} g(x/a)\,dx. \end{aligned} \tag{12}$$

TABLE III
Poles of the Digital Filters of Table II

Impulse Response	Transfer Function	
$(b^1)^{-1}(k)$	-	
$(b^2)^{-1}(k)$	z_1= -0.171573	
$(b^3)^{-1}(k)$	z_1= -0.267949	
$(b^4)^{-1}(k)$	z_1= -0.361341	z_2= -0.0137254
$(b^5)^{-1}(k)$	z_1= -0.430575	z_2= -0.0430963
$(b^6)^{-1}(k)$	z_1=-0.488295 z_3= -0.00141415	z_2= -0.0816793
$(b^7)^{-1}(k)$	z_1= -0.53528 z_3= -0.00914869	z_2= -0.122555
$(b^8)^{-1}(k)$	z_1= -0.574687 z_3= -0.0236323	z_2= -0.163035 z_4= -0.000153821
$(b^9)^{-1}(k)$	z_1= -0.607997 z_3= -0.0432226	z_2= -0.201751 z_4= -0.00212131
$(b^{10})^{-1}(k)$	z_1= -0.636551 z_3= -0.065727 z_5= -0.0000169828	z_2= -0.238183 z_4= -0.00752819
$(b^{11})^{-1}(k)$	z_1= -0.661266 z_3= -0.0897596 z_5= -0.000510558	z_2= -0.27218 z_4= -0.0166696

Now, if we assume that the signal interpolant $g(x)$ obtained in step 1 is in the spline space, i.e., $g(x) = \sum_k c(k)\beta^n(x-k)$, then the integral of $g(x)$ can be expressed as

$$\int_{-\infty}^{x} g(x)\,dx = \sum_k c(k) \int_{-\infty}^{x} \beta^n(x-k)\,dx. \tag{13}$$

In order calculate this expression explicitly, we use the following proposition, the proof of which is provided in Appendix.

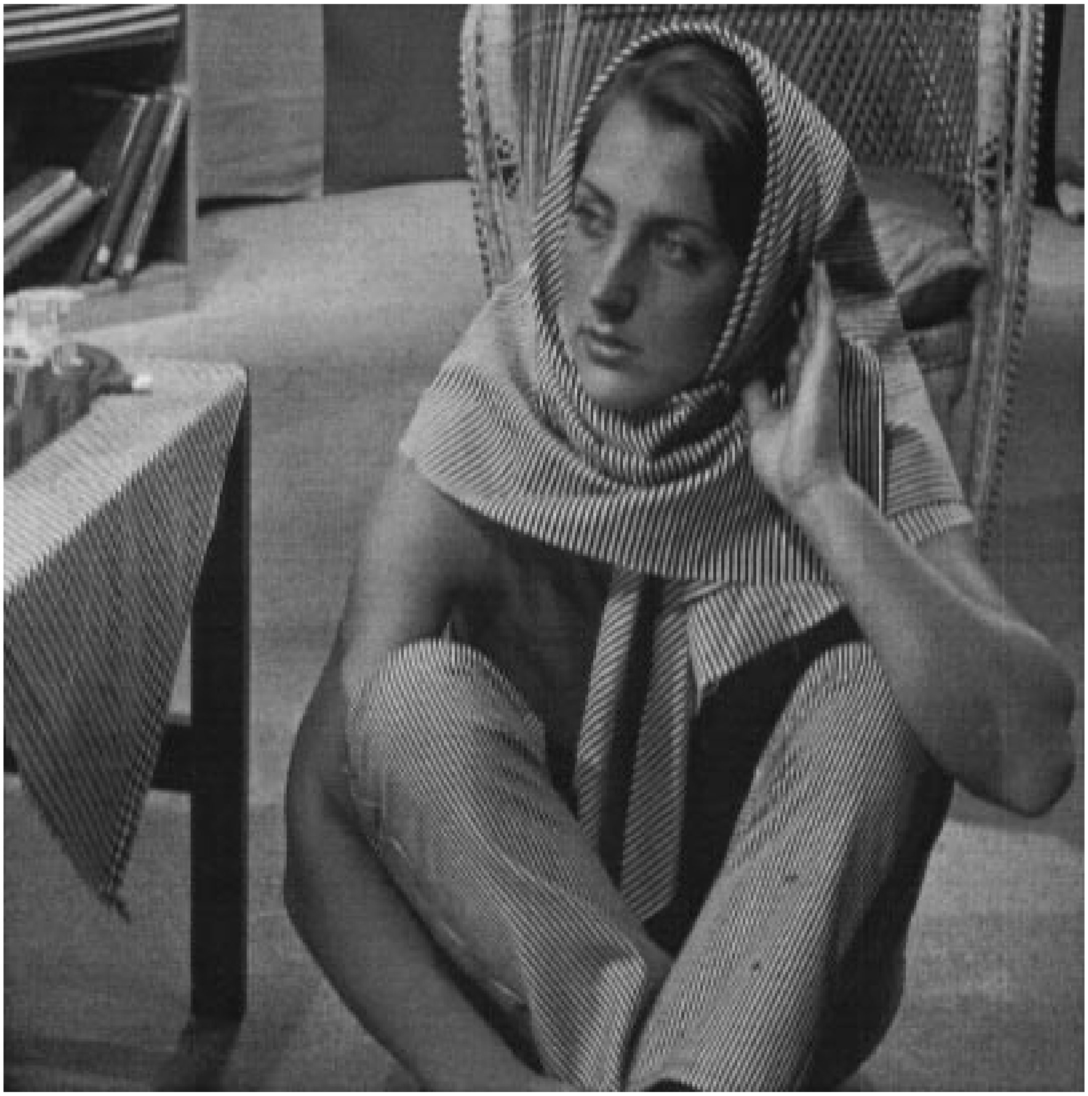


Fig. 6. Original Barbara image (512 × 512).

Proposition 1: The integral of a B-spline of degree n is given by

$$\int_{-\infty}^{x} \beta^n(x)\,dx = \sum_{k=0}^{\infty} \beta^{n+1}\left(x - \frac{1}{2} - k\right). \tag{14}$$

By substituting (14) in (13) and using the unit step $u(k) = 1$ for $k \geq 0$ and $u(k) = 0$ otherwise, we obtain

$$\begin{aligned} G(x) = \int_{-\infty}^{x} g(x)\,dx &= \sum_{k=-\infty}^{\infty} c(k) \sum_{l=k}^{\infty} \beta^{n+1}\left(x - \frac{1}{2} - l\right) \\ &= \sum_{l=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} c(k)u(l-k)\beta^{n+1}\left(x - \frac{1}{2} - l\right) \\ &= \sum_{k=-\infty}^{\infty} (u * c)(k)\beta^{n+1}\left(x - \frac{1}{2} - k\right). \end{aligned} \tag{15}$$

In other words, the integral is a polynomial spline of degree $n + 1$:

$$G(x) = \sum_{k \in Z} C(k)\beta^{n+1}\left(x - \frac{1}{2} - k\right) \tag{16}$$

whose B-spline coefficients are given by

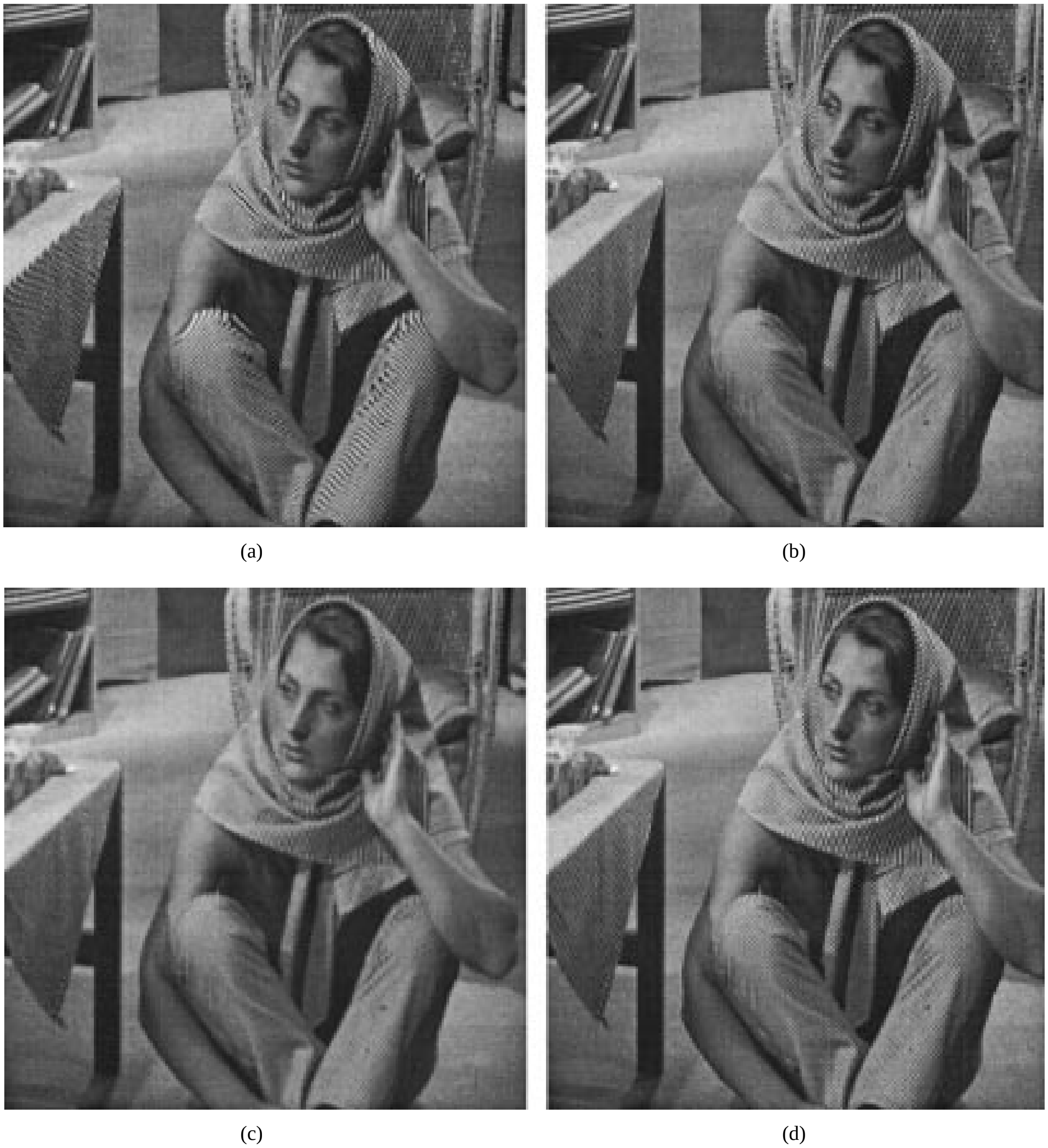
$$C(k) = (u * c)(k) = \sum_{l=-\infty}^{k} c(l). \tag{17}$$

Equation (12) can therefore be calculated as

$$\begin{aligned} c_1(k) &= \langle g(x/a), \beta^0(x-k) \rangle \\ &= aG\left(\frac{k}{a} + \frac{1}{2a}\right) - aG\left(\frac{k}{a} - \frac{1}{2a}\right) \end{aligned} \tag{18}$$

which is an expression that requires the evaluation of the integral (16) at two particular points. Since we need to scan through all the values of k successively, we can take advantage of previous calculations by rewriting the expression of the next coefficient $c_1(k+1)$ as follows:

$$\begin{aligned} c_1(k+1) &= aG\left(\frac{k+1}{a} + \frac{1}{2a}\right) - aG\left(\frac{k+1}{a} - \frac{1}{2a}\right) \\ &= aG\left(\frac{k+1}{a} + \frac{1}{2a}\right) - aG\left(\frac{k}{a} + \frac{1}{2a}\right). \end{aligned} \tag{19}$$



(a)

(b)

(c)

(d)

Fig. 7. Comparison of reduction procedures for $a = 1/2$. (a) Result of a simple decimation by a factor of two. (b) Least squares approximation with $n = 1$ $(\mathrm{SNR} = 24.71)$. (c) Oblique projection with $n = 0$ $(\mathrm{SNR} = 24.12)$. (d) Oblique projection with $n = 1$ $(\mathrm{SNR} = 24.75)$.

Since the second term of (19) is the same as the first term of (18), we therefore only need to perform one calculation of $G(x)$ per coefficient except at $k = 0$. The evaluation of $G(x)$, which is equivalent to a spline interpolation of degree $n+1$, is clearly the most time consuming part of the whole procedure. We have therefore effectively reduced the computing time by half.

The last aspect that needs to be dealt with is the derivation of the corresponding correction filter in Fig. 2. For our particular choice of φ_1 and φ_2, the cross-correlation function can be written as

$$\begin{aligned} a_{12}(k) &= \left\langle \beta^0(x-k), \sum_{l \in Z} (b^n)^{-1}(l) \beta^n(x-l) \right\rangle \\ &= \sum_{l \in Z} (b^n)^{-1}(l) \langle \beta^0(x-k), \beta^n(x-l) \rangle. \end{aligned}$$

Using the symmetry and the convolution property of B-splines, this simplifies to

$$a_{12}(k) = \sum_{l \in Z} (b^n)^{-1}(l) b^{n+1}(l-k) = ((b^n)^{-1} * b^{n+1})(k).$$

Hence, the required digital correction filter is

$$q(k) = ((b^{n+1})^{-1} * b^n)(k). \tag{20}$$

B. Practical Implementation

Now we propose the following procedure for image scaling using oblique projection.

Image Scaling Using Oblique Projection:

Step 1: Find the B-spline representation of the input signal $s(k), k = 0, \ldots, N-1$, i.e. $g(x) = \sum_k c(k) \beta^n(x-k)$. This is done efficiently by digital filtering (cf. (8)).

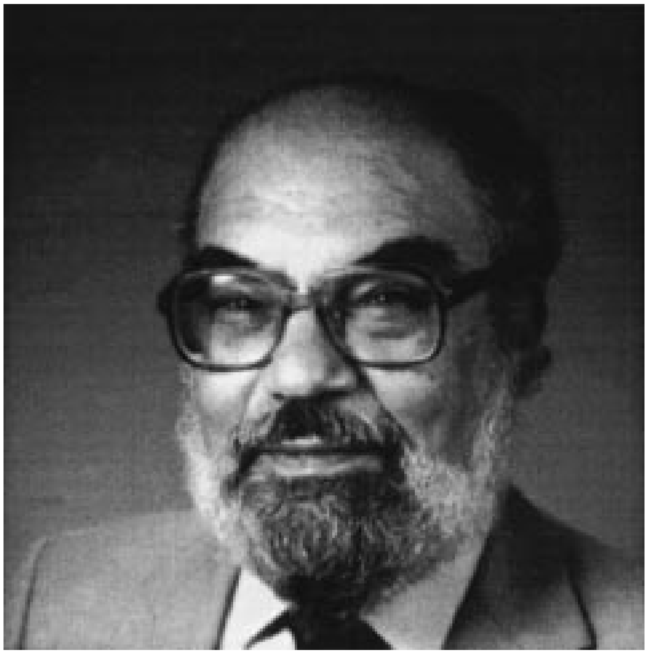


Fig. 8. Murray image.

Step 2: Compute the B-spline coefficients of the integral of $g(x)$ in Step 1 (cf. (17)).

Step 3: Calculate the coefficients of the re-scaled function $f(x) = g(x/a)$ (cf. (18)).

Step 4: Apply the appropriate postfilter to compute $\tilde{f}(k)$ [cf. (20) and (9)].

Steps 1, 2 and 4 are pure digital filtering operations. The filter in Step 1 is the direct B-spline filter $(b^n)^{-1}$ which has an infinite impulse response (IIR). The filter in Step 2 is a simple running sum updated recursively. Finally, the postfilter in Step 4 consists of an all pole component—the IIR filter $(b^{n+1})^{-1}$, and a finite impulse response (FIR) kernel b^n. In all cases, we use standard mirror symmetric boundary conditions. The two IIR filters in Steps 1 and 4 are implemented using the recursive algorithm described in [19]. Computationally, the most expensive part of the algorithm is Step 3. It is equivalent to a spline interpolation of degree $n+1$. This calculation requires a routine[1] that returns the values of $\beta^{n+1}(x)$ at a given point x_0; explicit B-spline formulas can be found in [25]. Note that the summation (16) only involves a small number of terms (typically, $n+1$) because of the compact support of the basis functions.

As already mentioned in the introduction, full image resizing is obtained by applying the 1-D procedure to the rows and the columns in succession (separable algorithm).

Note that the conventional interpolation approach can be implemented by skipping Step 2 and 4 and replacing the calculation in Step 3 by an interpolation that uses the basis function $\beta^n(x)$ instead of $\beta^{n+1}(x)$. Since the cost of digital filtering is negligible compared to Step 3, the complexity of our new oblique projection algorithm is not significantly different from that of a conventional interpolation with one more order of approximation.

So far in the literature, spline functions of relatively low orders have been used. Since the rescaling algorithm based on the oblique projection is easily applicable to higher order splines, we were motivated to calculate the relevant filter formulas and parameters for splines up to degree 11. Table II shows the transfer functions of the spline interpolation filters (conversion from cardinal to B-spline representation). Table III provides the causal poles of the transfer functions; these are the only parameters required by the recursive filtering algorithm [5], [19]. The additional FIR filter b^n for Step 4 can be obtained from the inverse of the transfer functions in Table II.

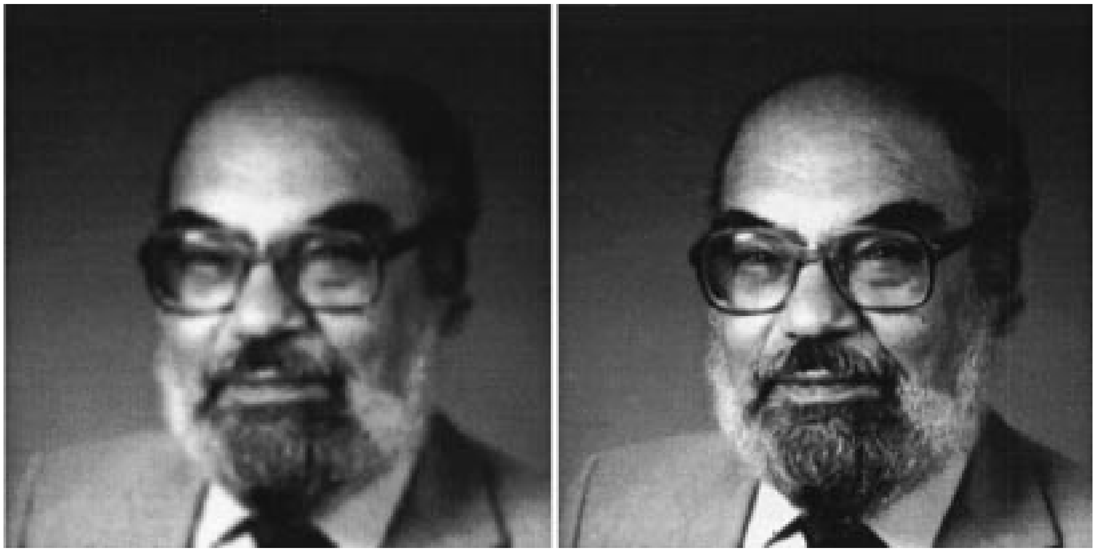
IV. Results and Discussion

A. Experiments

In order to evaluate the proposed algorithm, we performed a succession of complementary image reductions and magnifications, and vice versa. We tested two other algorithms for comparison. The first one is the standard algorithm which fits the image with a spline and then resamples this functions at the specified rate. It is one of the most widely used algorithms, at least for lower order splines. For $n = 1$, the approach is equivalent to bilinear interpolation. The second is the optimal method proposed in [13]. It applies the proper prefilter prior to sampling and provides the minimum error approximation. Note that these two methods also fit into the general theoretical framework that is presented in Section II. Specifically, if $\varphi_1 \in V(\varphi_2)$ then we get the least squares solution (orthogonal projection). Likewise, for $\varphi_1(x) = \delta(x)$ (Dirac), we obtain the standard interpolation procedure—with the important restriction that the angle θ_{12} between the input and output spaces is no longer defined (i.e., no upper error bound in (10)). The global loss of information was measured by the relative mean square difference between the approximation and the initial digital image, expressed in decibels.

In a first series of experiments, we first reduced the linear dimension of a brain magnetic resonance image (MRI) by $\sqrt{2}$ and then magnified it by $\sqrt{2}$. Table IV shows the signal-to-noise ratio (SNR) as a function of the degree n. As expected, the proposed algorithm consistently performed better than the standard interpolation method while requiring comparable processing time. In some cases, the differences are quite dramatic as illustrated in Figs. 3–5. Compared to the least square method, the oblique projection performed comparably with significant decrease in processing time. For the piecewise constant case (splines of degree zero), both algorithms are rigorously equivalent. Because an exact closed form formula with the least squares algorithm is extremely difficult to obtain for $n > 1$, the implemented version of the optimal algorithm for $n = 3$ is not numerically exact; it uses a Gaussian approximation instead of a kernel that is in fact the convolution of two B-splines of different size [13]. This lack of exactness may explain why the SNR of the optimal algorithm is not as good as the one of the oblique algorithm.

[1] We are ready to provide the corresponding C code on request for B-splines of degree up to ten, as well as the short 1-D subroutine that performs the recursive filtering.



(a) (b)

Fig. 9. Murray images after ten successive enlargements by $\sqrt{3}$ and reductions by $1/\sqrt{3}$. (a) Standard method (B-spline of degree one, $\mathrm{SNR} = 30.02$ dB). (b) Oblique interpolation (B-spline of degree one, $\mathrm{SNR} = 55.49$ dB).

TABLE IV
MAGNETIC RESONANCE IMAGE (FIRST REDUCED BY $\sqrt{2}$ AND MAGNIFIED BY $\sqrt{2}$)

order	standard	optimal	oblique
0	27.19 dB	32.73 dB	32.73 dB
1	33.65 dB	39.25 dB	39.02 dB
3	41.54 dB	41.74 dB	42.15 dB
5	42.20 dB		42.70 dB
7	42.21 dB		42.82 dB

Tables V and VI provide similar performance comparisons for two other test images (Murray and mandrill images), which are shown in Figs. 8 and 10. We have arbitrarily chosen a set of irrational scaling factors. The images are first reduced (enlarged) and then enlarged (reduced) back using the same scaling factor a. As can be seen from those results, the performance of the oblique projection is essentially the same as that of the optimal solution for $n \leq 3$, which confirms our expectations. When the images are first reduced and then enlarged [Tables V and VI(a)–(d)], the performance of the oblique projection is better than that of the standard algorithm by 1–2 dB. For larger reduction factors, this is true irrespective of the degree n. It appears that the primary reason for the lesser performance of the standard interpolation methods is aliasing. This effect is more prominent for larger reduction factors or when the images contain many high-frequency details (e.g., the mandrill). In the case of image reduction using the oblique projection operator, there is a tendency to saturation for $n \geq 3$, sometimes with a very slight loss of performance for $n > 3$. In principle, higher order approximations should be better, at least for the least squares case [26]. For $n > 3$, the benefit of having a higher order of approximation appears to be offset by the fact that the angle θ_{12} between the input and output spaces increases with n (cf. (10)). In this sense, the proposed cubic spline algorithm appears to provide the best compromise for image reduction.

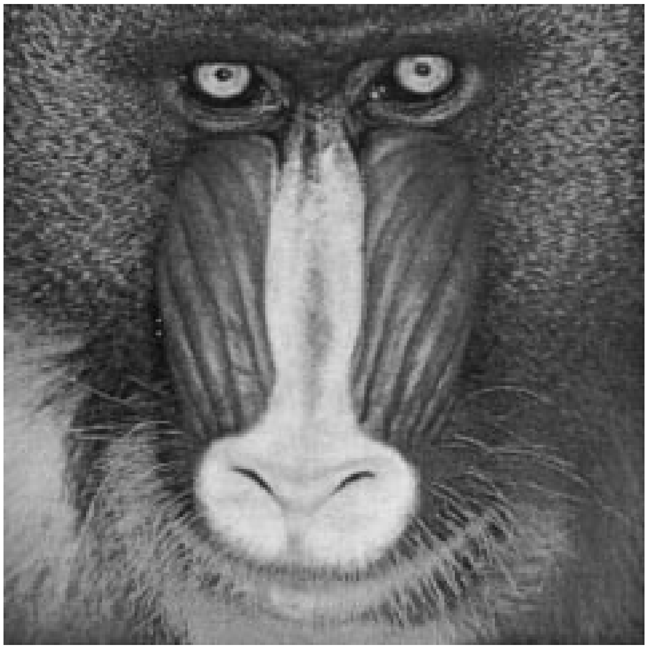


Fig. 10. Mandrill image.

To further highlight the differences between the algorithms, we also performed some experiments with integer reduction factors. This is a somewhat special situation in which all interpolation algorithms give the same results and are equivalent to a simple decimation without prefiltering. Some comparative examples of image reductions by a factor of two are shown in Fig. 7. For the purpose of demonstration, we used the Barbara image which has many structured high frequency patterns (Fig. 6). The aliasing is quite visible in Fig. 7(a) (standard interpolation); in this case, the SNR depends on the underlying spline model: $\mathrm{SNR} = 21.18$ for $n = 0$, $\mathrm{SNR} = 23.89$ for $n = 1$, and $\mathrm{SNR} = 22.75$ for $n = 3$. The results are obviously much more satisfactory if we use our oblique projection algorithm, and there is not much difference with the least squares solution displayed in Fig 7(b). The fact that there

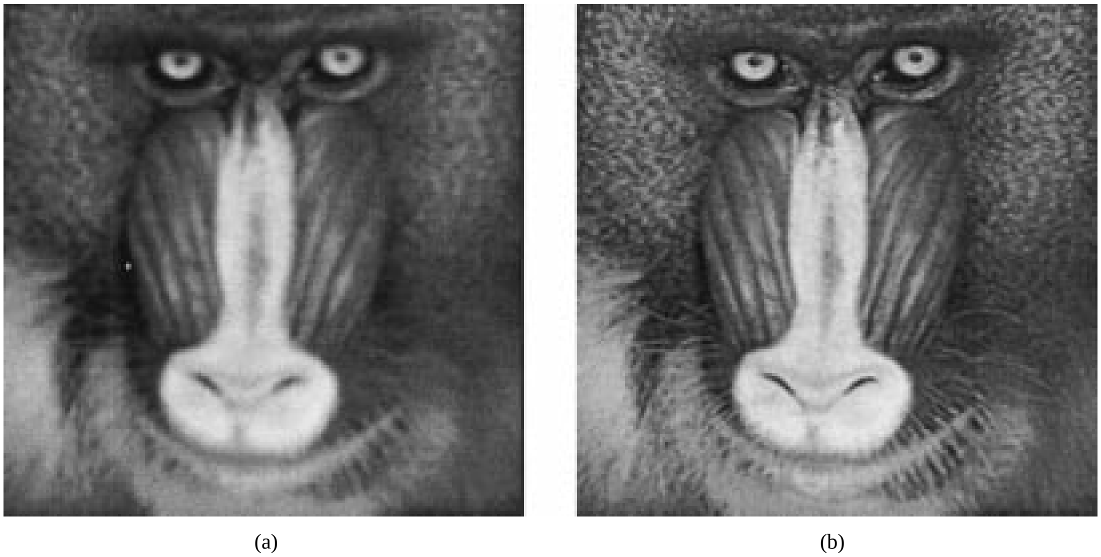

(a) (b)

Fig. 11. (a) Accumulated distortion after ten iterations of the standard interpolation algorithm (degree $= 1$, scale $= 0.97$). (b) Accumulated distortion after ten iterations of the oblique projection algorithm (degree $= 1$, scale $= 0.97$).

is a better suppression of aliasing is also reflected by the higher values of the SNR ($\mathrm{SNR} \cong 24.7$ dB for $n = 1$) computed using the same methodology as before. Note that when the reduction factor is an integer, there are also more standard filtering/decimation techniques with the same kind of performance as our new algorithm [19]. These methods tend to be more efficient computationally but they require a prefilter that needs to be designed specifically for each reduction factor m.

Our resizing algorithm is obviously also applicable to image magnification. If we do the same type of back and forth experiment as before with an integer magnification, we find that the error is zero, at least for the cases where n is odd (for n even, this is only true for odd magnification factors). The reason for this special behavior is that the underlying spline spaces are nested so that the projector can reproduce the signal in $V(\varphi_2)$ without error. Another way to understand this property is that a spline stretched by an integer factor m can still be viewed as a spline with finer knots at the integers. Hence, for integer scaling factors, our method is equivalent to a standard spline interpolation of degree n, a fact that we have verified experimentally. We do not present any examples here because the performance of spline interpolators in well documented in the literature [5]. However, as soon as we attempt a nonintegral magnification [cf. Tables V and VI, (e)–(g)], the oblique projection performs considerably better than the corresponding interpolation. If high order splines $(n \geq 3)$ are used, the improvement in SNR's is often more than 10 dB. For linear splines, the improvement is as much as 20 dB. This finding is in agreement with the fact that the oblique projection is asymptotically optimal [cf. (11)], while the standard interpolation is usually biased asymptotically [27]. For image enlargement in general, the oblique projection using high-order splines outperforms all other methods. This finding is also consistent with the theoretical results presented in Section II-C. In particular, for a very large scaling factor $(\sqrt{26} \approx 5.10)$, the oblique projection produces no error. It is noted that when an image is enlarged and reduced to the original size using the standard interpolation of degree zero (nearest neighbor interpolation), the error will be zero. The SNR in this case is meaningless as the enlarged image appears very blocky.

Fig. 9 displays the result of ten successive enlargement by $\sqrt{3}$ and reduction by $1/\sqrt{3}$ using linear splines, for the standard interpolation method and the oblique projection. The SNR's are 30.02 dB and 55.49 dB, respectively. The oblique projection provides more faithful and sharper image. Fig. 11 displays the final results of ten cumulative image reductions by a factor of 0.97 using a linear spline signal representation. Again, the oblique projection provides a much higher quality result with better preservation of small image details. Of course, these are artificial experiments designed to accumulate errors so that the differences are more striking visually.

V. Conclusion

We have introduced a new image scaling algorithm based on an oblique projection. By using an oblique rather than an orthogonal projection, we have obtained faster, simpler, and more general algorithms. In particular, the new approach allows great flexibility in the design of the prefilter. For a simple and easy implementation, we have proposed using the B-spline of degree zero as prefilter so that the required inner products can be computed by straightforward integration of the input signal. We have derived the formulas required and have discussed some techniques for fast implementation. In all our test experiments, the proposed algorithm outperformed the standard interpolation method and showed comparable performance to the optimum algorithm while reducing computing time substantially with respect to the latter. The oblique projection that we have proposed using splines of degree n had approximately the same computational complexity as a spline interpolation of degree $n+1$; but its performance was usually better.

TABLE V

MURRAY. (a) $a = 1/\sqrt{26} \approx 0.196$. (b) $a = 1/\sqrt{8} \approx 0.354$. (c) $a = 0.5$. (d) $a = 1/\sqrt{3} \approx 0.577$. (e) $a = 1/\sqrt{3} \approx 1.732$. (f) $a = 1/\sqrt{5} \approx 2.236$. (g) $a = 1/\sqrt{26} \approx 5.10$

order	standard	optimal	oblique
0	22.97	24.68	24.67
1	25.31	26.17	26.24
3	24.90	26.09	26.46
5	24.65		26.40
7	24.52		26.34

(a)

order	standard	optimal	oblique
0	26.43	28.65	28.64
1	29.47	30.61	30.58
3	29.49	31.13	30.97
5	29.17		30.93
7	29.00		30.88

(b)

order	standard	optimal	oblique
0	27.14	31.31	31.30
1	31.74	32.55	32.71
3	31.77	33.03	33.34
5	31.56		33.25
7	31.43		33.18

(c)

order	standard	optimal	oblique
0	27.65	32.06	32.05
1	32.69	34.15	34.25
3	33.76	34.72	34.67
5	33.33		34.52
7	33.05		34.39

(d)

order	standard	optimal	oblique
0	-	40.67	40.63
1	40.73	63.00	63.46
3	58.66	63.94	69.74
5	61.76		70.25
7	61.24		70.28

(e)

order	standard	optimal	oblique
0	-	42.64	42.58
1	42.89	66.54	68.07
3	62.42	59.91	75.16
5	62.92		77.84
7	62.29		78.97

(f)

order	standard	optimal	oblique
0	-	49.72	49.54
1	49.77	87.91	81.67
3	66.83	63.96	no error
5	64.93		no error
7	64.02		no error

(g)

TABLE VI

MANDRILL. (a) $a = 1/\sqrt{26} \approx 0.196$. (b) $a = 1/\sqrt{8} \approx 0.354$. (c) $a = 0.5$. (d) $a = 1/\sqrt{3} \approx 0.577$. (e) $a = 1/\sqrt{3} \approx 1.732$. (f) $a = 1/\sqrt{5} \approx 2.236$. (g) $a = 1/\sqrt{26} \approx 5.10$

order	standard	optimal	oblique
0	19.28	21.84	21.84
1	21.44	22.44	22.39
3	20.45	22.53	22.46
5	20.14		22.44
7	20.00		22.41

(a)

order	standard	optimal	oblique
0	21.01	23.68	23.67
1	23.41	24.18	24.13
3	22.42	24.29	24.13
5	22.04		24.05
7	21.85		24.00

(b)

order	standard	optimal	oblique
0	21.91	24.79	24.78
1	24.25	25.51	25.55
3	23.64	25.72	25.56
5	23.40		25.41
7	23.27		25.31

(c)

order	standard	optimal	oblique
0	22.53	25.60	25.59
1	25.47	26.42	26.47
3	25.35	26.61	26.40
5	24.86		26.20
7	24.57		26.06

(d)

order	standard	optimal	oblique
0	-	33.08	33.06
1	33.07	54.78	57.15
3	52.45	66.58	67.69
5	60.40		68.51
7	60.30		68.89

(e)

order	standard	optimal	oblique
0	-	35.11	35.08
1	35.09	60.53	62.88
3	56.69	62.22	70.48
5	61.36		75.82
7	60.61		76.63

(f)

order	standard	optimal	oblique
0	-	41.95	41.86
1	41.94	80.99	95.82
3	63.62	65.99	no error
5	61.76		no error
7	61.04		no error

(g)

As far as the new algorithm is concerned, the best results for image reduction were obtained for $n = 3$ (cubic spline model). For image enlargement, the performance of the algorithm could be improved almost arbitrarily by using higher order splines. Considering that the optimal least squares solution is very difficult to implement for $n > 1$, the oblique projection can provide a very attractive solution whenever high-quality results are required.

Our method can deal with arbitrary scaling factors; it shows its full strength when noninteger factors are desired.

It is particularly well suited for image reduction because of its built-in antialiasing mechanism. It can also handle image magnification and is most indicated for nonintegral magnification factors because of its improved performance. When the magnification factor is an integer, it provides the same results as a spline interpolation of degree n.

APPENDIX

Proof of Proposition 1: A form equivalent to the definition of B-spline in (2) is

$$\beta^n(u) = \int_{x-\frac{1}{2}}^{x+\frac{1}{2}} \beta^{n-1}(u)\,du. \tag{A1}$$

By substituting x by $x - \frac{1}{2} - k$ in (A1), we find that

$$\begin{aligned}\beta^n\left(x - \frac{1}{2} - k\right) &= \int_{x-1-k}^{x-k} \beta^{n-1}(u)\,du \\ &= \int_{-\infty}^{x-k} \beta^{n-1}(u)\,du \\ &\quad - \int_{-\infty}^{x-1-k} \beta^{n-1}(u)\,du. \end{aligned} \tag{A2}$$

Now we sum (A2) over k (from 0 to infinite) and obtain

$$\begin{aligned}&\sum_{k=0}^{\infty} \beta^n\left(x - \frac{1}{2} - k\right) \\ &\quad = \sum_{k=0}^{\infty}\left(\int_{-\infty}^{x-k} \beta^{n-1}(u)\,du - \int_{-\infty}^{x-1-k} \beta^{n-1}(u)\,du\right) \\ &\quad = \int_{-\infty}^{x} \beta^{n-1}(u)\,du - \int_{-\infty}^{x-1-\infty} \beta^{n-1}(u)\,du \\ &\quad = \int_{-\infty}^{x} \beta^{n-1}(u)\,du.\end{aligned}$$

Therefore, we have shown that the integral of the B-spline of degree $n - 1$ is given by

$$\int_{-\infty}^{x} \beta^{n-1}(x)\,dx = \sum_{k=0}^{\infty} \beta^n\left(x - \frac{1}{2} - k\right). \qquad \square$$

REFERENCES

[1] W. K. Pratt, *Digital Image Processing*. New York: Wiley, 1978.

[2] A. Rosenfeld and A. C. Kak, *Digital Picture Processing*. New York: Academic, 1982.

[3] R. G. Keys, "Cubic convolution interpolation for digital image processing," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-29, pp. 1153–1160, 1981.

[4] H. S. Hou and H. C. Andrews, "Cubic splines for image interpolation and digital filtering," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-26, pp. 508–517, 1978.

[5] M. Unser, A. Aldroubi, and M. Eden, "Fast B-spline transforms for continuous image representation and interpolation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 13, pp. 277–285, 1991.

[6] G. Strang and G. Fix, "A Fourier analysis of the finite element variational method," in *Constructive Aspect of Functional Analysis*, Rome, Italy: Edizioni Cremonese, 1971, pp. 796–830.

[7] C. de Boor, "Quasiinterpolation and approximation power of multivariate splines," in *Computation of Curves and Surfaces*, W. Dahmen, Ed. Boston, MA: Kluwer, pp. 313–345, 1990.

[8] A. Aldroubi, M. Unser, and M. Eden, "Cardinal spline filters: Stability and convergence to the ideal sinc interpolator," *Signal Process.*, vol. 28, pp. 127–138, Aug. 1992.

[9] S. G. Mallat, "A theory of multiresolution signal decomposition: The wavelet representation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, pp. 674–693, 1989.

[10] M. Unser, A. Aldroubi, and M. Eden, "The L_2 polynomial spline pyramid," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 15, pp. 364–379, 1993.

[11] A. Aldroubi, M. Unser, and M. Eden, "Discrete spline filters for multiresolutions and wavelets of l_2," *SIAM J. Math. Anal.*, vol. 25, pp. 1412–1432, Sept. 1994.

[12] H. Greenspan and C. H. Anderson, "Image enhancement by nonlinear extrapolation in frequency space," in *Proc. SPIE, Image and Video Processing II*, 1994, vol. 2182, pp. 2–13.

[13] 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.

[14] M. Vetterli and J. Kovacevic, *Wavelets and Subband Coding*. Englewood Cliffs, NJ: Prentice-Hall, 1995.

[15] M. Vetterli and C. Herley, "Wavelets and filter banks: Theory and design," *IEEE Trans. Signal Processing*, vol. 40, pp. 2207–2232, Sept. 1992.

[16] A. Cohen, I. Daubechies, and J. C. Feauveau, "Biorthogonal bases of compactly supported wavelets," *Commun. Pure Appl. Math.*, vol. 45, pp. 485–560, 1992.

[17] W. Sweldens and R. Piessens, "Asymptotic error expansions for wavelet approximations of smooth functions II," *Numer. Math.*, vol. 68, pp. 377–401, 1994.

[18] M. Unser, "Approximation power of biorthogonal wavelet expansions," *IEEE Trans. Signal Processing*, vol. 44, pp. 519–527, Mar. 1996.

[19] M. Unser, A. Aldroubi, and M. Eden, "B-spline signal processing, Part II: Efficient design and applications," *IEEE Trans. Signal Processing*, vol. 41, pp. 834–848, Feb. 1993.

[20] A. Aldroubi and M. Unser, "Sampling procedures in function spaces and asymptotic equivalence with Shannon's sampling theory," *Numer. Funct. Anal. Optimiz.*, vol. 15, pp. 1–21, Feb. 1994.

[21] I. J. Schoenberg, *Cardinal Spline Interpolation*. Philadelphia, PA: SIAM, 1973.

[22] ——, "Cardinal interpolation and spline functions," *J. Approx. Theory*, vol. 2, pp. 167–206, 1969.

[23] M. Unser and A. Aldroubi, "A general sampling theory for nonideal acquisition devices," *IEEE Trans. Signal Processing*, vol. 42, pp. 2915–2925, 1994.

[24] M. Unser and J. Zerubia, "Generalized sampling: Stability and performance analysis," *IEEE Trans. Signal Processing*, vol. 45, pp. 2941–2950, Dec. 1997.

[25] M. Unser, A. Aldroubi, and M. Eden, "B-spline signal processing. Part I: Theory," *IEEE Trans. Signal Processing*, vol. 41, pp. 821–833, Feb. 1993.

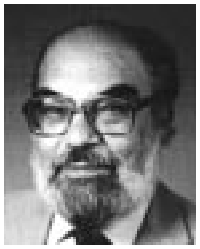
[26] M. Unser, "On the optimality of ideal filters for pyramid and wavelet signal approximation," *IEEE Trans. Signal Processing*, vol. 41, pp. 3591–3596, Dec. 1993.

[27] M. Unser and I. Daubechies, "On the approximation power of convolution-based least squares vs. interpolation," *IEEE Trans. Signal Processing*, vol. 45, pp. 1697–1711, July 1997.

Chulhee Lee (S'89–M'91) received the B.S. and M.S degrees in electronics engineering from Seoul National University, Seoul, Korea, in 1984 and 1986, respectively, and the Ph.D. degree in electrical engineering from Purdue University, West Lafayette, IN, in 1992.

From 1986 to 1987, he was a researcher in the Acoustic Laboratory, Technical University of Denmark (DTU), Lyngby. From 1993 to 1996, he worked with National Institutes of Health, Bethesda, MD. In 1996, he joined the faculty of the Department of Electronic Engineering, Yonsei University, Seoul, Korea. His research interests include image/signal processing, pattern recognition, and neural networks.

Dr. Lee is a member of Tau Beta Pi, Eta Kappa Nu, and KSEA.



Murray Eden (M'60–F'73–LF'91) was born in Brooklyn, New York, on August 17, 1920. He received the B.S. degree from City College of New York in 1939 and the Ph.D. from the University of Maryland, College Park, in 1951.

He is currently Professor of Electrical Engineering Emeritus at the Massachusetts Institute of Technology, Cambridge, and Scientist Emeritus with the National Institutes of Health (NIH), Bethesda, MD. He was the former director of the Biomedical Engineering and Instrumentation Program, National Center for Research Resources, NIH. His research interests include pattern recognition, analytical uses of image processing and models for perception.

Michael Unser (M'89–SM'94) was born in Zug, Switzerland, on April 9, 1958. He received the M.S. (summa cum laude) and Ph.D. degrees in electrical engineering in 1981 and 1984, respectively, from the Swiss Federal Institute of Technology, Lausanne, Switzerland.

From 1985 to 1997, he was with the Biomedical Engineering and Instrumentation Program, National Institutes of Health, Bethesda, MD, where he headed the Image Processing Group. He is now Professor of Biomedical Image Processing at the Swiss Federal Institute of Technology. His research centers around the numerical aspects of biomedical imaging. He has a strong interest in sampling theories, multiresolution algorithms, wavelets, and the use of splines for image processing. He is the author of over 70 published journal papers in these areas.

Dr. Unser serves as an Associate Editor for the IEEE SIGNAL PROCESSING LETTERS, and is a former Associate Editor (1992–1995) for the IEEE TRANSACTIONS ON IMAGE PROCESSING. He is a member of the Image and Multidimensional Signal Processing Committee of the IEEE Signal Processing Society. He is also on the editorial boards of *Signal Processing* and *Pattern Recognition*. He co-organized the 1994 IEEE-EMBS Workshop on Wavelets in Medicine and Biology, and serves as conference chair for SPIE's Wavelet Applications in Signal and Image Processing, which has been held annually since 1993. He received the Dommer Prize for Excellence from the Swiss Federal Institute of Technology in 1981, the Research Prize of the Brown–Boweri Corporation (Switzerland) for his thesis in 1984, and the IEEE Signal Processing Society's 1995 Best Paper Award (with A. Aldroubi and M. Eden) in an IMDSP technical area for a TRANSACTIONS paper on B-spline signal processing.